REJ09B0452-0100

Everywhere you imagine. RENESAS

16

H8S/2117R Group

Hardware Manual

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

H8S/2117R R4F2117R

Hardware Manual

All information contained in this material, including products and product specifications at the time of publication of this material, is subject to change by Renesas Technology Corp. without notice. Please review the latest information published by Renesas Technology Corp. through various means, including the Renesas Technology Corp. website (http://www.renesas.com).

Rev.1.00
Revision Date: Apr. 28, 2008

Renesas Technology
www.renesas.com

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, 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 products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life

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

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

1. Handling of Unused Pins
   Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.
   — The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions may occur due to the false recognition of the pin state as an input signal. Unused pins should be handled as described under Handling of Unused Pins in the 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 register 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, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.
   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the 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 functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.
4. Clock Signals
   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.
   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization 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 progress, 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 part number, confirm that the change will not lead to problems.
   — The characteristics of MPU/MCU in the same group but having different part numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different part numbers, implement a system-evaluation test for each of the products.

How to Use This Manual

1. Objective and Target Users

   This manual was written to explain the hardware functions and electrical characteristics of this LSI to the target users, i.e. those who will be using this LSI in the design of application systems. Target users are expected to understand the fundamentals of electrical circuits, logic circuits, and microcomputers.

   This manual is organized in the following items: an overview of the product, descriptions of the CPU, system control functions, and peripheral functions, electrical characteristics of the device, and usage notes.

   When designing an application system that includes this LSI, take all points to note into account. Points to note are given in their contexts and at the final part of each section, and in the section giving usage notes.

   The list of revisions is a summary of major points of revision or addition for earlier versions. It does not cover all revised items. For details on the revised points, see the actual locations in the manual.

   The following documents have been prepared for the H8S/2117R Group. Before using any of the documents, please visit our web site to verify that you have the most up-to-date available version of the document.

Document Type	Contents	Document Title	Document No.
Data Sheet	Overview of hardware and electrical characteristics	—	—
Hardware Manual	Hardware specifications (pin assignments, memory maps, peripheral specifications, electrical characteristics, and timing charts) and descriptions of operation	H8S/2117R Group Hardware Manual	This manual
Software Manual	Detailed descriptions of the CPU and instruction set	H8S/2600 Series H8S/2000 Series Software Manual	REJ09B0139
Application Note	Examples of applications and sample programs	The latest versions are available from our web site.	
Renesas Technical Update	Preliminary report on the specifications of a product, document, etc.		

2. Description of Numbers and Symbols

Aspects of the notations for register names, bit names, numbers, and symbolic names in this manual are explained below.

(1) Overall notation

In descriptions involving the names of bits and bit fields within this manual, the modules and registers to which the bits belong may be clarified by giving the names in the forms "module name"."register name"."bit name" or "register name"."bit name".

(2) Register notation

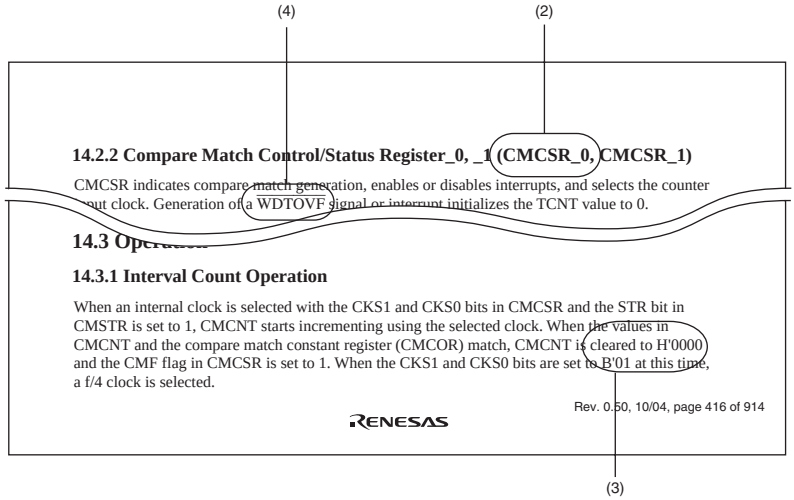
The style "register name"_"instance number" is used in cases where there is more than one instance of the same function or similar functions.
[Example] CMCSR_0: Indicates the CMCSR register for the compare-match timer of channel 0.

(3) Number notation

Binary numbers are given as B'nnnn (B' may be omitted if the number is obviously binary), hexadecimal numbers are given as H'nnnn or 0xnnnn, and decimal numbers are given as nnnn.
[Examples] Binary: B'11 or 11
Hexadecimal: H'EFA0 or 0xEFA0
Decimal: 1234

(4) Notation for active-low

An overbar on the name indicates that a signal or pin is active-low.
[Example] $\overline{\text{WDTOVF}}$

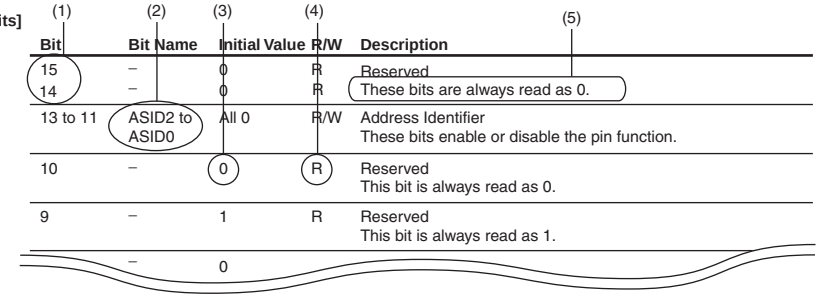


Note: The bit names and sentences in the above figure are examples and have nothing to do with the contents of this manual.

3. Description of Registers

Each register description includes a bit chart, illustrating the arrangement of bits, and a table of bits, describing the meanings of the bit settings. The standard format and notation for bit charts and tables are described below.

[Table of Bits]



Bit	Bit Name	Initial Value	R/W	Description
15	–	0	R	Reserved
14	–	0	R	These bits are always read as 0.
13 to 11	ASID2 to ASID0	All 0	R/W	Address Identifier These bits enable or disable the pin function.
10	–	0	R	Reserved This bit is always read as 0.
9	–	1	R	Reserved This bit is always read as 1.
	–	0		

Note: The bit names and sentences in the above figure are examples, and have nothing to do with the contents of this manual.

(1) Bit
Indicates the bit number or numbers.
In the case of a 32-bit register, the bits are arranged in order from 31 to 0. In the case of a 16-bit register, the bits are arranged in order from 15 to 0.

(2) Bit name
Indicates the name of the bit or bit field.
When the number of bits has to be clearly indicated in the field, appropriate notation is included (e.g., ASID[3:0]).
A reserved bit is indicated by "–".
Certain kinds of bits, such as those of timer counters, are not assigned bit names. In such cases, the entry under Bit Name is blank.

(3) Initial value
Indicates the value of each bit immediately after a power-on reset, i.e., the initial value.
0: The initial value is 0
1: The initial value is 1
–: The initial value is undefined

(4) R/W
For each bit and bit field, this entry indicates whether the bit or field is readable or writable, or both writing to and reading from the bit or field are impossible.
The notation is as follows:

R/W: The bit or field is readable and writable.
R/(W): The bit or field is readable and writable.
However, writing is only performed to flag clearing.
R: The bit or field is readable.
"R" is indicated for all reserved bits. When writing to the register, write the value under Initial Value in the bit chart to reserved bits or fields.
W: The bit or field is writable.

(5) Description
Describes the function of the bit or field and specifies the values for writing.

4. Description of Abbreviations

The abbreviations used in this manual are listed below.

- Abbreviations specific to this product

Abbreviation	Description
BSC	Bus controller
CPG	Clock pulse generator
INT	Interrupt controller
SCI	Serial communication interface
TMR	8-bit timer
TPU	16-bit timer pulse unit
WDT	Watchdog timer

- Abbreviations other than those listed above

Abbreviation	Description
ACIA	Asynchronous communication interface adapter
bps	Bits per second
CRC	Cyclic redundancy check
DMA	Direct memory access
DMAC	Direct memory access controller
GSM	Global System for Mobile Communications
Hi-Z	High impedance
IEBus	Inter Equipment Bus (IEBus is a trademark of NEC Electronics Corporation.)
I/O	Input/output
IrDA	Infrared Data Association
LSB	Least significant bit
MSB	Most significant bit
NC	No connection
PLL	Phase-locked loop
PWM	Pulse width modulation
SFR	Special function register
SIM	Subscriber Identity Module
UART	Universal asynchronous receiver/transmitter
VCO	Voltage-controlled oscillator

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

Contents

Section 1 Overview 1
1.1 Features 1
1.1.1 Applications 1
1.1.2 Overview of Functions 2
1.2 List of Products 7
1.3 Block Diagram 8
1.4 Pin Descriptions 9
1.4.1 Pin Assignments 9
1.4.2 Pin Assignment in Each Operating Mode 12
1.4.3 Pin Functions 19

Section 2 CPU 29
2.1 Features 29
2.1.1 Differences between H8S/2600 CPU and H8S/2000 CPU 30
2.1.2 Differences from H8/300 CPU 31
2.1.3 Differences from H8/300H CPU 31
2.2 CPU Operating Modes 32
2.2.1 Normal Mode 32
2.2.2 Advanced Mode 34
2.3 Address Space 36
2.4 Registers 37
2.4.1 General Registers 38
2.4.2 Program Counter (PC) 39
2.4.3 Extended Control Register (EXR) 39
2.4.4 Condition-Code Register (CCR) 40
2.4.5 Multiply-Accumulate Register (MAC) 41
2.4.6 Initial Values of CPU Registers 41
2.5 Data Formats 42
2.5.1 General Register Data Formats 42
2.5.2 Memory Data Formats 44
2.6 Instruction Set 45
2.6.1 Table of Instructions Classified by Function 46
2.6.2 Basic Instruction Formats 56
2.7 Addressing Modes and Effective Address Calculation 57
2.7.1 Register Direct—Rn 57
2.7.2 Register Indirect—@ERn 57

2.7.3 Register Indirect with Displacement—@(d:16, ERn) or @(d:32, ERn) ... 58
2.7.4 Register Indirect with Post-Increment or Pre-Decrement—@ERn+ or @-ERn ... 58
2.7.5 Absolute Address—@aa:8, @aa:16, @aa:24, or @aa:32 ... 58
2.7.6 Immediate—#xx:8, #xx:16, or #xx:32 ... 59
2.7.7 Program-Counter Relative—@(d:8, PC) or @(d:16, PC) ... 59
2.7.8 Memory Indirect—@@aa:8 ... 60
2.7.9 Effective Address Calculation ... 61
2.8 Processing States ... 63
2.9 Usage Note ... 65
2.9.1 Notes on Using the Bit Operation Instruction ... 65

Section 3 MCU Operating Modes ... 67
3.1 Operating Mode Selection ... 67
3.2 Register Descriptions ... 68
3.2.1 Mode Control Register (MDCR) ... 68
3.2.2 System Control Register (SYSCR) ... 69
3.2.3 Serial Timer Control Register (STCR) ... 71
3.2.4 System Control Register 3 (SYSCR3) ... 73
3.3 Operating Mode Descriptions ... 73
3.3.1 Mode 2 ... 73
3.4 Address Map ... 74

Section 4 Exception Handling ... 75
4.1 Exception Handling Types and Priority ... 75
4.2 Exception Sources and Exception Vector Table ... 76
4.3 Reset ... 79
4.3.1 Reset Exception Handling ... 79
4.3.2 Interrupts Immediately after Reset ... 80
4.3.3 On-Chip Peripheral Modules after Reset is Cancelled ... 80
4.4 Interrupt Exception Handling ... 81
4.5 Trap Instruction Exception Handling ... 81
4.6 Exception Handling by Illegal Instruction ... 82
4.7 Stack Status after Exception Handling ... 83
4.8 Usage Note ... 84

Section 5 Interrupt Controller 85
5.1 Features 85
5.2 Input/Output Pins 87
5.3 Register Descriptions 88
5.3.1 Interrupt Control Registers A to D (ICRA to ICRD) 89
5.3.2 Address Break Control Register (ABRKCR) 91
5.3.3 Break Address Registers A to C (BARA to BARC) 92
5.3.4 IRQ Sense Control Registers (ISCR16H, ISCR16L, ISCRH, ISCRL) 93
5.3.5 IRQ Enable Registers (IER16, IER) 96
5.3.6 IRQ Status Registers (ISR16, ISR) 97
5.3.7 Keyboard Matrix Interrupt Mask Registers (KMIMRA KMIMR) Wake-Up Event Interrupt Mask Registers (WUEMR) 99
5.3.8 IRQ Sense Port Select Register 16 (ISSR16) IRQ Sense Port Select Register (ISSR) 103
5.3.9 Wake-Up Sense Control Register (WUESCR) Wake-Up Input Interrupt Status Register (WUESR) Wake-Up Enable Register (WER) 104
5.4 Interrupt Sources 105
5.4.1 External Interrupt Sources 105
5.4.2 Internal Interrupt Sources 108
5.5 Interrupt Exception Handling Vector Tables 108
5.6 Interrupt Control Modes and Interrupt Operation 117
5.6.1 Interrupt Control Mode 0 119
5.6.2 Interrupt Control Mode 1 121
5.6.3 Interrupt Exception Handling Sequence 124
5.6.4 Interrupt Response Times 125
5.7 Address Breaks 126
5.7.1 Features 126
5.7.2 Block Diagram 126
5.7.3 Operation 127
5.7.4 Usage Notes 127
5.8 Usage Notes 129
5.8.1 Conflict between Interrupt Generation and Disabling 129
5.8.2 Instructions for Disabling Interrupts 130
5.8.3 Interrupts during Execution of EEPMOV Instruction 130
5.8.4 Vector Address Switching 130
5.8.5 External Interrupt Pin in Software Standby Mode and Watch Mode 131
5.8.6 Noise Canceller Switching 131
5.8.7 IRQ Status Register (ISR) 131

Section 6 Bus Controller (BSC) 133
6.1 Register Descriptions 133
6.1.1 Bus Control Register (BCR) 133
6.1.2 Wait State Control Register (WSCR) 134

Section 7 I/O Ports 135
7.1 Register Descriptions 143
7.1.1 Data Direction Register (PnDDR) (n = 1 to 6, 8, 9, A to D, and F to J) 144
7.1.2 Data Register (PnDR) (n = 1 to 6, 8, and 9) 145
7.1.3 Input Data Register (PnPIN) (n = 1 to 9 and A to J) 145
7.1.4 Pull-Up MOS Control Register (PnPCR) (n = 1 to 3, 9, B to D, F, H, and J)
Pull-Up MOS Control Register (KMPCR) (Port 6) 146
7.1.5 Output Data Register (PnODR) (n = A to D and F to J) 149
7.1.6 Noise Canceler Enable Register (PnNCE) (n = 6, C, and G) 149
7.1.7 Noise Canceler Decision Control Register (PnNCMC) (n = 6, C, and G) 150
7.1.8 Noise Cancel Cycle Setting Register (PnNCCS) (n = 6, C, and G) 150
7.1.9 Port Nch-OD Control Register (PnNOCR) (n = C, D, and F to J) 152
7.1.10 Pin Functions 153
7.2 Output Buffer Control 154
7.2.1 Port 1 154
7.2.2 Port 2 154
7.2.3 Port 3 155
7.2.4 Port 4 155
7.2.5 Port 5 159
7.2.6 Port 6 161
7.2.7 Port 7 162
7.2.8 Port 8 163
7.2.9 Port 9 166
7.2.10 Port A 168
7.2.11 Port B 169
7.2.12 Port C 173
7.2.13 Port D 177
7.2.14 Port E 177
7.2.15 Port F 178
7.2.16 Port G 181
7.2.17 Port H 185
7.2.18 Port I 187
7.2.19 Port J 187
7.3 Change of Peripheral Function Pins 194
7.3.1 Port Control Register 0 (PTCNT0) 194

7.3.2 Port Control Register 1 (PTCNT1) 195
7.3.3 Port Control Register 2 (PTCNT2) 196

Section 8 8-Bit PWM Timer (PWMU) 197
8.1 Features 197
8.2 Input/Output Pins 199
8.3 Register Descriptions 200
8.3.1 PWM Control Register A (PWMCONA) 202
8.3.2 PWM Control Register B (PWMCONB) 202
8.3.3 PWM Control Register C (PWMCONC) 205
8.3.4 PWM Control Register D (PWMCOND) 206
8.3.5 PWM Prescaler Registers 0 to 5 (PWMPRE0 to PWMPRE5) 207
8.3.6 PWM Duty Setting Registers 0 to 5 (PWMREG0 to PWMREG5) 209
8.4 Operation 210
8.4.1 Single-Pulse Mode (8 Bits, 16 Bits) 210
8.4.2 Pulse Division Mode 214
8.5 Usage Note 217
8.5.1 Setting Module Stop Mode 217
8.5.2 Note on Using 16-Bit Single-Pulse PWM Timer 217

Section 9 14-Bit PWM Timer (PWMX) 219
9.1 Features 219
9.2 Input/Output Pins 220
9.3 Register Descriptions 220
9.3.1 PWMX (D/A) Counter (DACNT) 221
9.3.2 PWMX (D/A) Data Registers A and B (DADRA and DADRB) 222
9.3.3 PWMX (D/A) Control Register (DACR) 224
9.3.4 Peripheral Clock Select Register (PCSR) 225
9.4 Bus Master Interface 226
9.5 Operation 229
9.6 Usage Notes 236
9.6.1 Module Stop Mode Setting 236

Section 10 16-Bit Timer Pulse Unit (TPU) 237
10.1 Features 237
10.2 Input/Output Pins 241
10.3 Register Descriptions 242
10.3.1 Timer Control Register (TCR) 243
10.3.2 Timer Mode Register (TMDR) 247
10.3.3 Timer I/O Control Register (TIOR) 249

10.3.4 Timer Interrupt Enable Register (TIER) ... 258
10.3.5 Timer Status Register (TSR) ... 260
10.3.6 Timer Counter (TCNT) ... 263
10.3.7 Timer General Register (TGR) ... 263
10.3.8 Timer Start Register (TSTR) ... 263
10.3.9 Timer Synchro Register (TSYR) ... 264
10.4 Interface to Bus Master ... 265
10.4.1 16-Bit Registers ... 265
10.4.2 8-Bit Registers ... 265
10.5 Operation ... 267
10.5.1 Basic Functions ... 267
10.5.2 Synchronous Operation ... 273
10.5.3 Buffer Operation ... 275
10.5.4 PWM Modes ... 279
10.5.5 Phase Counting Mode ... 283
10.6 Interrupts ... 288
10.6.1 Interrupt Source and Priority ... 288
10.6.2 A/D Converter Activation ... 289
10.7 Operation Timing ... 290
10.7.1 Input/Output Timing ... 290
10.7.2 Interrupt Signal Timing ... 294
10.8 Usage Notes ... 297
10.8.1 Input Clock Restrictions ... 297
10.8.2 Caution on Period Setting ... 297
10.8.3 Conflict between TCNT Write and Clear Operations ... 298
10.8.4 Conflict between TCNT Write and Increment Operations ... 298
10.8.5 Conflict between TGR Write and Compare Match ... 299
10.8.6 Conflict between Buffer Register Write and Compare Match ... 299
10.8.7 Conflict between TGR Read and Input Capture ... 300
10.8.8 Conflict between TGR Write and Input Capture ... 301
10.8.9 Conflict between Buffer Register Write and Input Capture ... 301
10.8.10 Conflict between Overflow/Underflow and Counter Clearing ... 302
10.8.11 Conflict between TCNT Write and Overflow/Underflow ... 303
10.8.12 Multiplexing of I/O Pins ... 303
10.8.13 Module Stop Mode Setting ... 303

Section 11 16-Bit Cycle Measurement Timer (TCM) 305
11.1 Features 305
11.2 Input/Output Pins 307
11.3 Register Descriptions 307
11.3.1 TCM Timer Counter (TCMCNT) 309
11.3.2 TCM Cycle Upper Limit Register (TCMMLCM) 309
11.3.3 TCM Cycle Lower Limit Register (TCMMINCM) 310
11.3.4 TCM Input Capture Register (TCMICR) 310
11.3.5 TCM Input Capture Buffer Register (TCMICRF) 310
11.3.6 TCM Status Register (TCMCSR) 311
11.3.7 TCM Control Register (TCMCR) 313
11.3.8 TCM Interrupt Enable Register (TCMIER) 315
11.4 Operation 317
11.4.1 Timer Mode 317
11.4.2 Cycle Measurement Mode 319
11.5 Interrupt Sources 324
11.6 Usage Notes 325
11.6.1 Conflict between TCMCNT Write and Count-Up Operation 325
11.6.2 Conflict between TCMMLCM Write and Compare Match 325
11.6.3 Conflict between TCMICR Read and Input Capture 326
11.6.4 Conflict between Edge Detection in Cycle Measurement Mode and Writing to TCMMLCM or TCMMINCM 326
11.6.5 Conflict between Edge Detection in Cycle Measurement Mode and Clearing of TCMMDS Bit in TCMCR 327
11.6.6 Settings of TCMCKI and TCMMCI 327
11.6.7 Setting for Module Stop Mode 327

Section 12 16-Bit Duty Period Measurement Timer (TDP) 329
12.1 Features 329
12.2 Input/Output Pins 331
12.3 Register Descriptions 331
12.3.1 TDP Timer Counter (TDPCNT) 333
12.3.2 TDP Pulse Width Upper Limit Register (TDPWDMX) 333
12.3.3 TDP Pulse Width Lower Limit Register (TDPWDMN) 334
12.3.4 TDP Cycle Upper Limit Register (TDPPDMX) 334
12.3.5 TDP Cycle Lower Limit Register (TDPPDMN) 334
12.3.6 TDP Input Capture Register (TDPICR) 335
12.3.7 TDP Input Capture Buffer Register (TDPICRF) 335
12.3.8 TDP Status Register (TDPCSR) 335

12.3.9 TDP Control Register 1 (TDPCR1) 338
12.3.10 TDP Control Register 2 (TDPCR2) 340
12.3.11 TDP Interrupt Enable Register (TDPIER) 341
12.4 Operation 343
12.4.1 Timer Mode 343
12.4.2 Cycle Measurement Mode 345
12.5 Interrupt Sources 350
12.6 Usage Notes 351
12.6.1 Conflict between TDPCNT Write and Count-Up Operation 351
12.6.2 Conflict between TDPPDMX Write and Compare Match 351
12.6.3 Conflict between Input Capture and TDPICR Read 352
12.6.4 Conflict between Edge Detection in Cycle Measurement Mode and Writing to the Upper Limit or Lower Limit Register 352
12.6.5 Conflict between Edge Detection in Cycle Measurement Mode and TDPMDS Bit Clearing 353
12.6.6 Settings for TDPCKI and TDPMCI 353
12.6.7 Setting for Module Stop Mode 353

Section 13 8-Bit Timer (TMR) 355
13.1 Features 355
13.2 Input/Output Pins 358
13.3 Register Descriptions 359
13.3.1 Timer Counter (TCNT) 360
13.3.2 Time Constant Register A (TCORA) 361
13.3.3 Time Constant Register B (TCORB) 361
13.3.4 Timer Control Register (TCR) 362
13.3.5 Timer Control/Status Register (TCSR) 367
13.3.6 Time Constant Register C (TCORC) 372
13.3.7 Input Capture Registers R and F (TICRR and TICRF) 372
13.3.8 Timer Connection Register I (TCONRI) 373
13.3.9 Timer Connection Register S (TCONRS) 373
13.3.10 Timer XY Control Register (TCRXY) 374
13.4 Operation 375
13.4.1 Pulse Output 375
13.5 Operation Timing 376
13.5.1 TCNT Count Timing 376
13.5.2 Timing of CMFA and CMFB Setting at Compare-Match 377
13.5.3 Timing of Timer Output at Compare-Match 377
13.5.4 Timing of Counter Clear at Compare-Match 378
13.5.5 TCNT External Reset Timing 378

13.5.6 Timing of Overflow Flag (OVF) Setting 379
13.6 TMR_0 and TMR_1 Cascaded Connection 380
13.6.1 16-Bit Count Mode 380
13.6.2 Compare-Match Count Mode 380
13.7 TMR_Y and TMR_X Cascaded Connection 381
13.7.1 16-Bit Count Mode 381
13.7.2 Compare-Match Count Mode 381
13.7.3 Input Capture Operation 382
13.8 Interrupt Sources 384
13.9 Usage Notes 385
13.9.1 Conflict between TCNT Write and Counter Clear 385
13.9.2 Conflict between TCNT Write and Count-Up 385
13.9.3 Conflict between TCOR Write and Compare-Match 386
13.9.4 Conflict between Compare-Matches A and B 386
13.9.5 Switching of Internal Clocks and TCNT Operation 387
13.9.6 Mode Setting with Cascaded Connection 389
13.9.7 Module Stop Mode Setting 389

Section 14 Watchdog Timer (WDT) 391
14.1 Features 391
14.2 Input/Output Pins 393
14.3 Register Descriptions 393
14.3.1 Timer Counter (TCNT) 394
14.3.2 Timer Control/Status Register (TCSR) 394
14.4 Operation 398
14.4.1 Watchdog Timer Mode 398
14.4.2 Interval Timer Mode 399
14.5 Interrupt Sources 400
14.6 Usage Notes 400
14.6.1 Notes on Register Access 400
14.6.2 Conflict between Timer Counter (TCNT) Write and Increment 401
14.6.3 Changing Values of CKS2 to CKS0 Bits 402
14.6.4 Changing Value of PSS Bit 402
14.6.5 Switching between Watchdog Timer Mode and Interval Timer Mode 402

Section 15 Serial Communication Interface (SCI) 403
15.1 Features 403
15.2 Input/Output Pins 405
15.3 Register Descriptions 406
15.3.1 Receive Shift Register (RSR) 407
15.3.2 Receive Data Register (RDR) 407
15.3.3 Transmit Data Register (TDR) 407
15.3.4 Transmit Shift Register (TSR) 407
15.3.5 Serial Mode Register (SMR) 408
15.3.6 Serial Control Register (SCR) 411
15.3.7 Serial Status Register (SSR) 414
15.3.8 Smart Card Mode Register (SCMR) 418
15.3.9 Bit Rate Register (BRR) 419
15.4 Operation in Asynchronous Mode 424
15.4.1 Data Transfer Format 425
15.4.2 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode 426
15.4.3 Clock 427
15.4.4 SCI Initialization (Asynchronous Mode) 428
15.4.5 Serial Data Transmission (Asynchronous Mode) 429
15.4.6 Serial Data Reception (Asynchronous Mode) 431
15.5 Multiprocessor Communication Function 435
15.5.1 Multiprocessor Serial Data Transmission 437
15.5.2 Multiprocessor Serial Data Reception 438
15.6 Operation in Clocked Synchronous Mode 441
15.6.1 Clock 441
15.6.2 SCI Initialization (Clocked Synchronous Mode) 442
15.6.3 Serial Data Transmission (Clocked Synchronous Mode) 443
15.6.4 Serial Data Reception (Clocked Synchronous Mode) 445
15.6.5 Simultaneous Serial Data Transmission and Reception (Clocked Synchronous Mode) 447
15.7 Smart Card Interface Description 449
15.7.1 Sample Connection 449
15.7.2 Data Format (Except in Block Transfer Mode) 450
15.7.3 Block Transfer Mode 451
15.7.4 Receive Data Sampling Timing and Reception Margin 452
15.7.5 Initialization 453
15.7.6 Serial Data Transmission (Except in Block Transfer Mode) 453
15.7.7 Serial Data Reception (Except in Block Transfer Mode) 456

15.7.8 Clock Output Control ... 458
15.8 Interrupt Sources ... 460
15.8.1 Interrupts in Normal Serial Communication Interface Mode ... 460
15.8.2 Interrupts in Smart Card Interface Mode ... 461
15.9 Usage Notes ... 462
15.9.1 Module Stop Mode Setting ... 462
15.9.2 Break Detection and Processing ... 462
15.9.3 Mark State and Break Sending ... 462
15.9.4 Receive Error Flags and Transmit Operations (Clocked Synchronous Mode Only) ... 462
15.9.5 Relation between Writing to TDR and TDRE Flag ... 462
15.9.6 SCI Operations during Mode Transitions ... 463
15.9.7 Notes on Switching from SCK Pins to Port Pins ... 466
15.9.8 Note on Writing to Registers in Transmission, Reception, and Simultaneous Transmission and Reception ... 467

Section 16 CIR Interface ... 469
16.1 Features ... 469
16.2 Input Pins ... 471
16.3 Register Description ... 471
16.3.1 Receive Control Register 1 (CCR1) ... 472
16.3.2 Receive Control Register 2 (CCR2) ... 473
16.3.3 Receive Status Register (CSTR) ... 474
16.3.4 Interrupt Enable Register (CEIR) ... 476
16.3.5 Bit Rate Register (BRR) ... 477
16.3.6 Receive Data Register 0 to 17 (CIRRDR0 to CIRRDR17) ... 478
16.3.7 Header Minimum/Maximum High-Level Period Register (HHMIN and HHMAX) ... 478
16.3.8 Header Minimum/Maximum Low-Level Period Register (HLMIN/HLMAX) ... 480
16.3.9 Data Level 1 Minimum/Maximum Period Register (DT1MIN/DT1MAX) ... 480
16.3.10 Data Level 0 Minimum/Maximum Period Register (DT0MIN/DT0MAX) ... 481
16.3.11 Repeat Header Minimum/Maximum Low-Level Period Register (RMIN/RMAX) ... 481
16.4 Operation ... 482
16.4.1 Determination of Signal Type by Low/High-Level Period ... 484
16.4.2 Operation of FIFO Register ... 486
16.4.3 Operation in Watch Mode ... 487

16.4.4 Switching between System Clock and Sub Clock 487
16.5 Noise Canceler Circuit 488
16.6 Reset Conditions 490
16.7 Interrupt Sources 490
16.8 Usage Note 491

Section 17 Serial Communication Interface with FIFO (SCIF) 493
17.1 Features 493
17.2 Input/Output Pins 495
17.3 Register Descriptions 496
17.3.1 Receive Shift Register (FRSR) 497
17.3.2 Receive Buffer Register (FRBR) 497
17.3.3 Transmitter Shift Register (FTSR) 498
17.3.4 Transmitter Holding Register (FTHR) 498
17.3.5 Divisor Latch H, L (FDLH, FDLL) 498
17.3.6 Interrupt Enable Register (FIER) 499
17.3.7 Interrupt Identification Register (FIIR) 500
17.3.8 FIFO Control Register (FFCR) 502
17.3.9 Line Control Register (FLCR) 503
17.3.10 Modem Control Register (FMCR) 504
17.3.11 Line Status Register (FLSR) 506
17.3.12 Modem Status Register (FMSR) 510
17.3.13 Scratch Pad Register (FSCR) 511
17.3.14 SCIF Control Register (SCIFCR) 512
17.4 Operation 514
17.4.1 Baud Rate 514
17.4.2 Operation in Asynchronous Communication 515
17.4.3 Initialization of the SCIF 516
17.4.4 Data Transmission/Reception with Flow Control 519
17.4.5 Data Transmission/Reception Through the LPC Interface 525
17.5 Interrupt Sources 528
17.6 Usage Note 528
17.6.1 Power-Down Mode When LCLK is Selected for SCLK 528

Section 18 I^2C Bus Interface (IIC) 529
18.1 Features 529
18.2 Input/Output Pins 533
18.3 Register Descriptions 534
18.3.1 I^2C Bus Data Register (ICDR) 536
18.3.2 Slave Address Register (SAR) 537

18.3.3 Second Slave Address Register (SARX) 538
18.3.4 I²C Bus Mode Register (ICMR) 540
18.3.5 I²C Bus Control Register (ICCR) 543
18.3.6 I²C Bus Status Register (ICSR) 551
18.3.7 I²C Bus Control Initialization Register (ICRES) 555
18.3.8 I²C Bus Extended Control Register (ICXR) 556
18.4 Operation 560
18.4.1 I²C Bus Data Format 560
18.4.2 Initialization 562
18.4.3 Master Transmit Operation 562
18.4.4 Master Receive Operation 567
18.4.5 Slave Receive Operation 570
18.4.6 Slave Transmit Operation 574
18.4.7 IRIC Setting Timing and SCL Control 577
18.4.8 Noise Canceler 579
18.4.9 Initialization of Internal State 579
18.5 Interrupt Sources 581
18.6 Usage Notes 582
18.6.1 Module Stop Mode Setting 585

Section 19 Keyboard Buffer Control Unit (PS2) 587
19.1 Features 587
19.2 Input/Output Pins 589
19.3 Register Descriptions 590
19.3.1 Keyboard Control Register 1 (KBCR1) 591
19.3.2 Keyboard Buffer Control Register 2 (KBCR2) 593
19.3.3 Keyboard Control Register H (KBCRH) 594
19.3.4 Keyboard Control Register L (KBCRL) 596
19.3.5 Keyboard Data Buffer Register (KBBR) 598
19.3.6 Keyboard Buffer Transmit Data Register (KBTR) 598
19.4 Operation 599
19.4.1 Receive Operation 599
19.4.2 Transmit Operation 601
19.4.3 Receive Abort 602
19.4.4 KCLKI and KDI Read Timing 605
19.4.5 KCLKO and KDO Write Timing 605
19.4.6 KBF Setting Timing and KCLK Control 606
19.4.7 Receive Timing 607
19.4.8 Operation during Data Reception 607
19.4.9 KCLK Fall Interrupt Operation 608

19.4.10 First KCLK Falling Interrupt 609
19.5 Usage Notes 613
19.5.1 KBIOE Setting and KCLK Falling Edge Detection 613
19.5.2 KD Output by KDO bit (KBCRL) and by Automatic Transmission 614
19.5.3 Module Stop Mode Setting 614
19.5.4 Medium-Speed Mode 614
19.5.5 Transmit Completion Flag (KBTE) 614

Section 20 LPC Interface (LPC) 615
20.1 Features 615
20.2 Input/Output Pins 617
20.3 Register Descriptions 618
20.3.1 Host Interface Control Registers 0 and 1 (HICR0 and HICR1) 620
20.3.2 Host Interface Control Registers 2 and 3 (HICR2 and HICR3) 626
20.3.3 Host Interface Control Register 4 (HICR4) 629
20.3.4 Host Interface Control Register 5 (HICR5) 630
20.3.5 LPC Channel 1 Address Registers H and L (LADR1H and LADR1L) 631
20.3.6 LPC Channel 2 Address Registers H and L (LADR2H and LADR2L) 632
20.3.7 LPC Channel 3 Address Registers H and L (LADR3H and LADR3L) 634
20.3.8 LPC Channel 4 Address Registers H and L (LADR4H and LADR4L) 636
20.3.9 Input Data Registers 1 to 4 (IDR1 to IDR4) 637
20.3.10 Output Data Registers 1 to 4 (ODR1 to ODR4) 637
20.3.11 Bidirectional Data Registers 0 to 15 (TWR0 to TWR15) 638
20.3.12 Status Registers 1 to 4 (STR1 to STR4) 638
20.3.13 SERIRQ Control Register 0 (SIRQCR0) 645
20.3.14 SERIRQ Control Register 1 (SIRQCR1) 649
20.3.15 SERIRQ Control Register 2 (SIRQCR2) 653
20.3.16 SERIRQ Control Register 3 (SIRQCR3) 656
20.3.17 SERIRQ Control Register 4 (SIRQCR4) 657
20.3.18 SCIF Address Register (SCIFADRH, SCIFADRL) 658
20.3.19 Host Interface Select Register (HISEL) 659
20.4 Operation 660
20.4.1 LPC interface Activation 660
20.4.2 LPC I/O Cycles 661
20.4.3 Gate A20 664
20.4.4 LPC Interface Shutdown Function (LPCPD) 667
20.4.5 LPC Interface Serialized Interrupt Operation (SERIRQ) 671
20.4.6 LPC Interface Clock Start Request 673
20.4.7 SCIF Control from LPC Interface 673
20.5 Interrupt Sources 674

20.5.1 IBFI1, IBFI2, IBFI3, IBFI4, OBEI, and ERRI 674
20.5.2 SMI, HIRQ1, HIRQ3, HIRQ4, HIRQ5, HIRQ6, HIRQ7, HIRQ8, HIRQ9, HIRQ10, HIRQ11, HIRQ12, HIRQ13, HIRQ14, and HIRQ15 675
20.6 Usage Note 678
20.6.1 Data Conflict 678

Section 21 FSI Interface 681
21.1 Features 681
21.2 Input/Output Pins 683
21.3 Register Description 684
21.3.1 FSI Control Register 1 (FSICR1) 686
21.3.2 FSI Control Register 2 (FSICR2) 688
21.3.3 FSI Byte Count Register (FSIBNR) 689
21.3.4 FSI Instruction Register (FSIINS) 690
21.3.5 FSI Instruction Register (FSIRDINS) 691
21.3.6 FSI Program Instruction Register (FSIPPINS) 691
21.3.7 FSI Status Register (FSISTR) 691
21.3.8 FSI Transmit Data Registers 0 to 7 (FSITDR0 to FSITDR7) 693
21.3.9 FSI Receive Data Register (FSIRDR) 693
21.3.10 FSI Access Host Base Address Registers H and L (FSIHBARH and FSIHBARL) 694
21.3.11 FSI Flash Memory Size Register (FSISR) 694
21.3.12 FSI Command Host Base Address Registers H and L (CMDHBARH and CMDHBARL) 695
21.3.13 FSI Command Register (FSICMDR) 696
21.3.14 FSI LPC Command Status Register 1 (FSILSTR1) 696
21.3.15 FSI LPC Command Status Register 2 (FSILSTR2) 698
21.3.16 FSI General-Purpose Registers 1 to F (FSIGPR1 to FSIGPRF) 699
21.3.17 FSI LPC Control Register (SLCR) 699
21.3.18 FSI Address Registers H, M, and L (FSIARH, FSIARM, and FSIARL) 700
21.3.19 FSI Write Data Registers HH, HL, LH, and LL (FSIWDRHH, FSIWDRHL, FSIWDRLH, and FSIWDRLL) 701
21.4 Operation 703
21.4.1 LPC/FW Memory Cycles 703
21.4.2 SPI Flash Memory Transfer 705
21.4.3 Flash Memory Instructions 706
21.4.4 FSI Memory Cycle (Direct Transfer between LPC and SPI) 707
21.4.5 FSI Memory Cycle (LPC-SPI Command Transfer) 714
21.4.6 SPI Flash Memory Write Operation Mode 722
21.5 Reset Conditions 723

21.6 Interrupt Sources 725
21.7 Usage Note 725
21.7.1 Longword Transfer in FW Memory Write Cycles 725

Section 22 A/D Converter 727
22.1 Features 727
22.2 Input/Output Pins 729
22.3 Register Descriptions 730
22.3.1 A/D Data Registers A to H (ADDRA to ADDRH) 731
22.3.2 A/D Control/Status Register (ADCSR) 732
22.3.3 A/D Control Register (ADCR) 734
22.4 Operation 735
22.4.1 Single Mode 735
22.4.2 Scan Mode 736
22.4.3 Input Sampling and A/D Conversion Time 737
22.5 Interrupt Source 739
22.6 A/D Conversion Accuracy Definitions 739
22.7 Usage Notes 741
22.7.1 Module Stop Mode Setting 741
22.7.2 Permissible Signal Source Impedance 741
22.7.3 Influences on Absolute Accuracy 742
22.7.4 Setting Range of Analog Power Supply and Other Pins 742
22.7.5 Notes on Board Design 742
22.7.6 Notes on Noise Countermeasures 743
22.7.7 Module Stop Mode Setting 744
22.7.8 Note on activation of the A/D converter by an external trigger 745

Section 23 RAM 747

Section 24 Flash Memory 749
24.1 Features 749
24.2 Mode Transition Diagram 750
24.3 Flash Memory MAT Configuration 752
24.4 Block Structure 753
24.5 Programming/Erasing Interface 754
24.6 Input/Output Pins 756
24.7 Register Descriptions 757
24.7.1 Programming/Erasing Interface Registers 759
24.7.2 Programming/Erasing Interface Parameters 765
24.8 On-Board Programming Mode 776

24.8.1 Boot Mode 776
24.8.2 User Program Mode 780
24.8.3 User Boot Mode 789
24.8.4 Storable Areas for On-Chip Program and Program Data 793
24.9 Protection 798
24.9.1 Hardware Protection 798
24.9.2 Software Protection 799
24.9.3 Error Protection 799
24.10 Switching between User MAT and User Boot MAT 801
24.11 Programmer Mode 802
24.12 Standard Serial Communication Interface Specifications for Boot Mode 802
24.13 Usage Notes 831

Section 25 Clock Pulse Generator 833
25.1 Oscillator 834
25.1.1 Connecting Crystal Resonator 834
25.1.2 External Clock Input Method 835
25.2 Duty Correction Circuit 838
25.3 Subclock Input Circuit 838
25.4 Subclock Waveform Forming Circuit 839
25.5 Clock Select Circuit 839
25.6 Usage Notes 840
25.6.1 Notes on Resonator 840
25.6.2 Notes on Board Design 840

Section 26 Power-Down Modes 841
26.1 Register Descriptions 841
26.1.1 Standby Control Register (SBYCR) 842
26.1.2 Low-Power Control Register (LPWRCR) 844
26.1.3 Module Stop Control Registers H, L, A, and B (MSTPCRH, MSTPCRL, MSTPCRA, MSTPCRB) 845
26.2 Mode Transitions and LSI States 848
26.3 Medium-Speed Mode 850
26.4 Sleep Mode 851
26.5 Software Standby Mode 851
26.6 Watch Mode 853
26.7 Module Stop Mode 854
26.8 Usage Notes 854
26.8.1 I/O Port Status 854
26.8.2 Current Consumption when Waiting for Oscillation Stabilization 854

Section 27 List of Registers 855
27.1 Register Addresses (Address Order) 856
27.2 Register Bits 878
27.3 Register States in Each Operating Mode 896
27.4 Register Selection Condition 912
27.5 Register Addresses (Classification by Type of Module) 934

Section 28 Electrical Characteristics 957
28.1 Absolute Maximum Ratings 957
28.2 DC Characteristics 958
28.3 AC Characteristics 963
28.3.1 Clock Timing 964
28.3.2 Control Signal Timing 966
28.3.3 Timing of On-Chip Peripheral Modules 968
28.4 A/D Conversion Characteristics 979
28.5 Flash Memory Characteristics 980
28.6 Usage Notes 981

Appendix 983
A. I/O Port States in Each Pin State 983
B. Product Lineup 984
C. Package Dimensions 985
D. Treatment of Unused Pins 988

Index 989

Section 1 Overview

1.1 Features

The core of each product in the H8S/2117R Group of CISC (complex instruction set computer) microcomputers is an H8S/2600 CPU, which has an internal 16-bit architecture. The H8S/2600 CPU provides upward-compatibility with the CPUs of other Renesas Technology-original microcomputers; H8/300, H8/300H, and H8S.

As peripheral functions, each LSI of the group includes a serial communication interface with FIFO, an I^2C bus interface, an A/D converter, and various types of timers. Together, the modules realize low-cost system configurations. The power consumption of these modules is kept down dynamically by power-down modes. The on-chip ROM is a flash memory (F-ZTAT™*) with a capacity of 160 Kbytes.

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

1.1.1 Applications

Examples of the applications of this LSI include PC peripheral equipment, office automation equipment, and industrial equipment.

1.1.2 Overview of Functions

Table 1.1 lists the functions of this LSI in outline.

Table 1.1 Overview of Functions

Classification	Module/ Function	Description
Memory	ROM	• ROM lineup: Flash memory version H8S/2117R: 160 Kbytes
	RAM	• RAM capacity: 8 Kbytes
CPU	CPU	• 16-bit high-speed H8S/2600 CPU (CISC type) Upward-compatibility with H8/300, H8/300H, and H8S CPUs at object level • General-register architecture (sixteen 16-bit general registers) • Eight addressing modes • 4-Gbyte address space Program: 4 Gbytes available Data: 4 Gbytes available • 69 basic instructions (bit arithmetic and logic instructions, multiply and divide instructions, bit manipulation instructions, multiply-and-accumulate instructions, and others) • Minimum instruction execution time: 50.0 ns (for an ADD instruction while system clock ϕ = 20 MHz and V_{CC} = 3.0 to 3.6 V) • On-chip multiplier (16 × 16 → 32 bits) • Supports multiply-and-accumulate instructions (16 × 16 + 32 → 32 bits)
	Operating mode	• Advanced and single-chip modes

Classification	Module/ Function	Description
CPU	MCU operating mode	Mode 2: Single-chip mode (selected by driving the MD2 and MD0 pins low and MD1 pin high) Mode 4: Boot mode (selected by driving the MD2 high and MD1 and MD0 pins low) Mode 6: On-chip emulation mode (selected by driving the MD2 and MD1 pins high and the MD0 pin low) Note: MD0 is not available as a pin and is internally fixed to 0. • Power-down state (transition to the power-down state made by the SLEEP instruction)
Interrupt (source)	Interrupt controller	• 41 external interrupt pins (NMI, $\overline{\text{IRQ15}}$ to $\overline{\text{IRQ0}}$ ($\overline{\text{ExIRQ15}}$ to $\overline{\text{ExIRQ6}}$), $\overline{\text{KIN15}}$ to $\overline{\text{KIN0}}$, and $\overline{\text{WUE15}}$ to $\overline{\text{WUE8}}$) • 66 internal interrupt sources • Two interrupt control modes (specified by the system control register) • Two levels of interrupt priority orders specifiable (by setting the interrupt control register) • Independent vector addresses
Clock	Clock pulse generator (CPG)	• Two clock generation circuits • Clock pulse generator and subclock input circuit System clock (ϕ) synchronization: 8 to 20 MHz • Five power-down modes: Medium-speed mode, sleep mode, watch mode, software standby mode, and module stop mode
A/D converter	A/D converter (ADC)	• 10-bit resolution × 16 input channels • Sample and hold function included • Conversion time: 4 μs per channel (with A/D conversion clock ADCLK at 10 MHz operation) • Two operating modes: single mode and scan mode • Three methods to start A/D conversion: software and two timer (TPU/TMR) triggers

Classification	Module/ Function	Description
Timer	8-bit PWM timer (PWMU)	• 8-bit timers A/B × six channels • Selectable from four clock sources • Cycle selectable for each channel • Supports 8-bit single pulse mode, 16-bit single pulse mode, and 8-bit pulse division mode.
	14-bit PWM timer (PWMX)	• 14 bits × two channels • Pulse division method • Selectable from sixteen operation clocks by combination of eight system clock cycles and two base cycles
	16-bit timer pulse unit (TPU)	• 16 bits × three channels • Selectable from eight counter input clocks for each channel • Maximum 8-pulse inputs/outputs • The following operations can be set. — Counter clear operation — Multiple timer counters (TCNT) can be written to simultaneously. — Simultaneous clearing by compare match and input capture possible — Register simultaneous input/output possible by counter synchronous operation — Maximum of 7-phase PWM output possible by combination with synchronous operation • Supports buffer operation and phase counting mode (two-phase encoder input) for some channels • Supports input capture function • Supports output compare function (waveform output at compare match)
	16-bit cycle measurement timer (TCM)	• 16 bits × four channels • Selectable from seven clocks: six internal clocks and one external clock • Capable of measuring the periods of input waveforms
	16-bit duty period measurement timer (TDP)	• 16 bits × three channels • Selectable from seven clocks: six internal clocks and one external clock • Capable of measuring the periods and pulse width of input waveforms

Classification	Module/ Function	Description
Timer	8-bit timer (TMR)	• 8 bits × four channels (also works as 16 bits × two channels) • Selectable from seven clocks: six internal clocks and one external clock • Pulse output or PWM output with an arbitrary duty cycle
Watchdog timer	Watchdog timer (WDT)	• 8 bits × two channels (selectable from eight counter input clocks) • Switchable between watchdog timer mode and interval timer mode
Serial interface	Serial communi-cation interface with FIFO (SCIF)	• One channel (asynchronous mode) • 16-stage FIFO buffers for transmission and reception • Full-duplex communication capability • On-chip baud rate generator allows any bit rate to be selected • Direct control from the LPC host
	Serial communi-cation interface (SCI)	• Two channels (choice of asynchronous or clocked synchronous serial communication mode) • Full-duplex communication capability • Selection of the desired bit rate and LSB-first or MSB-first transfer
Smart card/ SIM		• The SCI module supports a smart card (SIM) interface.
High-performance communication	I^2C bus interface (IIC)	• Three channels (two channels are switchable between input pin and output pin) • Two types of communication formats • I^2C bus format: addressing format with an acknowledge bit, for master/slave operation • Clocked synchronous serial format: non-addressing format without an acknowledge bit, for master operation only
	Keyboard buffer control unit (PS2)	• Four channels • Conforms to PS/2 interface specifications • Direct bus drive • Interrupt and error detection
	LPC interface (LPC)	• Four channels • Serial transfer of cycle type, address, and data in synchronization with the PCI clock • Supports LPC interface I/O read and I/O write cycles • Supports the shutdown function (LPCPD) of the LPC interface

Classification	Module/ Function	Description
High-performance communication	FSI interface (FSI)	• One channel • Supports communications between this LCI and SPI flash memory • Capable of operating as a master • Supports LPC reset and LPC shut-down
	CIR interface (CIR)	• One channel • Selectable from four sampling clocks: Three internal clocks and subclock • 18-byte FIFO incorporated
I/O ports		• Input-only pins: 13 pins • Input/output pins: 112 pins (TFP-144V and TLP-145V) 128 pins (BP-176V) • 76 pull-up resistors for TFP-144V and TLP-145V, and 84 pull-up resistors for BP-176V • 40 pins with LED drive capability • 24 on-chip noise cancellers
Package		• 144-pin thin QFP package (PTQP0144LC-A) (old code: TFP-144V, package dimensions: 16 × 16 mm, pin pitch: 0.40 mm) • 176-pin BGA package (PLBG0176GA-A) (old code: BP-176V, package dimensions: 13 × 13 mm, pin pitch: 0.80 mm) • 145-pin TLP package (PTLG0145JB-A) (package dimensions: 9 × 9 mm, pin pitch: 0.65 mm) • Lead- (Pb-) free version
Operating frequency/ Power supply voltage		• Operating frequency: 8 to 20 MHz • Power supply voltage: Vcc = 3.0 to 3.6 V, AVcc = 3.0 to 3.6 V • Supply current: 25 mA (typ.) (Vcc = 3.3 V, AVcc = 3.3 V, ϕ = 20 MHz)
Operating peripheral temperature (°C)		• –20 to +75°C (regular specifications)

1.2 List of Products

Table 1.2 is the list of products, and figure 1.1 shows how to read the product name code.

Table 1.2 List of Products

Part No.	ROM Capacity	RAM Capacity	Package	Remarks
R4F2117R	160 Kbytes	8 Kbytes	PTQP0144LC-A PLBG0176GA-A PTLG0145JB-A	Flash memory version

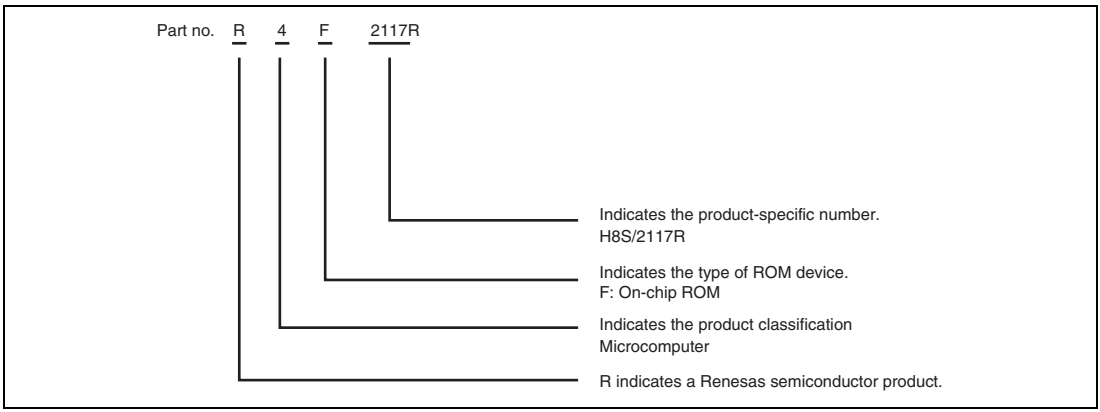
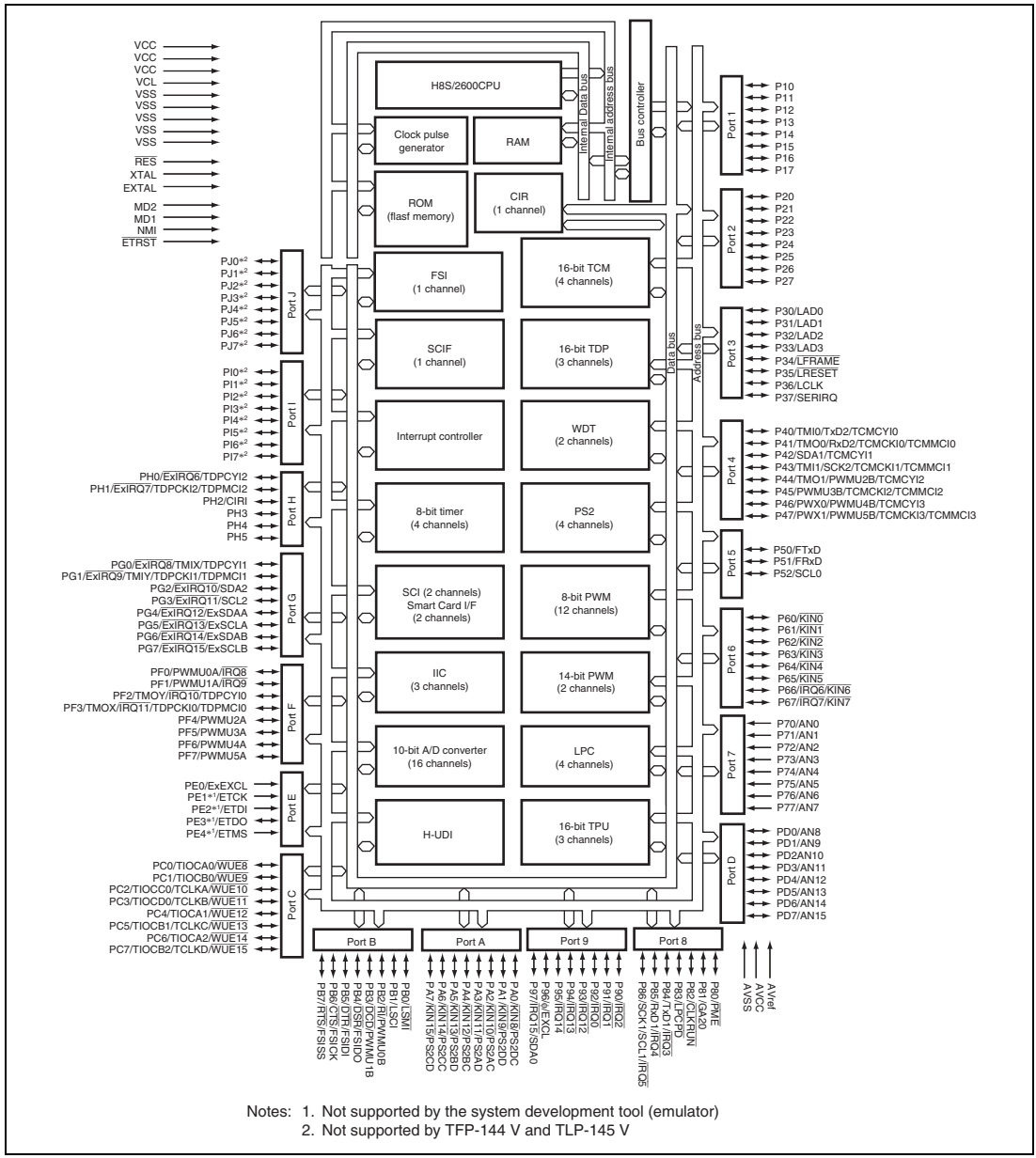


Figure 1.1 How to Read the Product Name Code

1.3 Block Diagram



Notes: 1. Not supported by the system development tool (emulator)
2. Not supported by TFP-144 V and TLP-145 V

Figure 1.2 Internal Block Diagram

1.4 Pin Descriptions

1.4.1 Pin Assignments

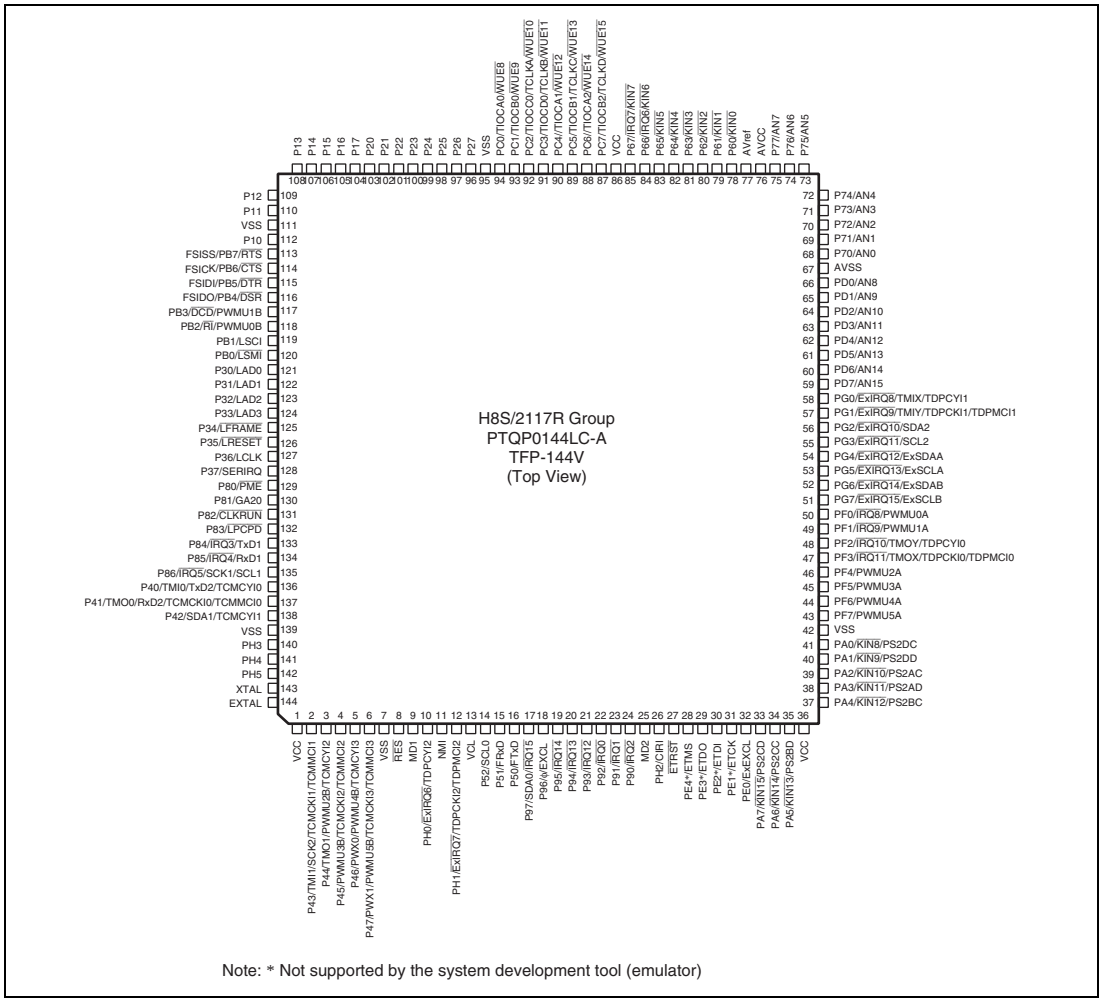


Figure 1.3 Pin Assignments (TFP-144V)

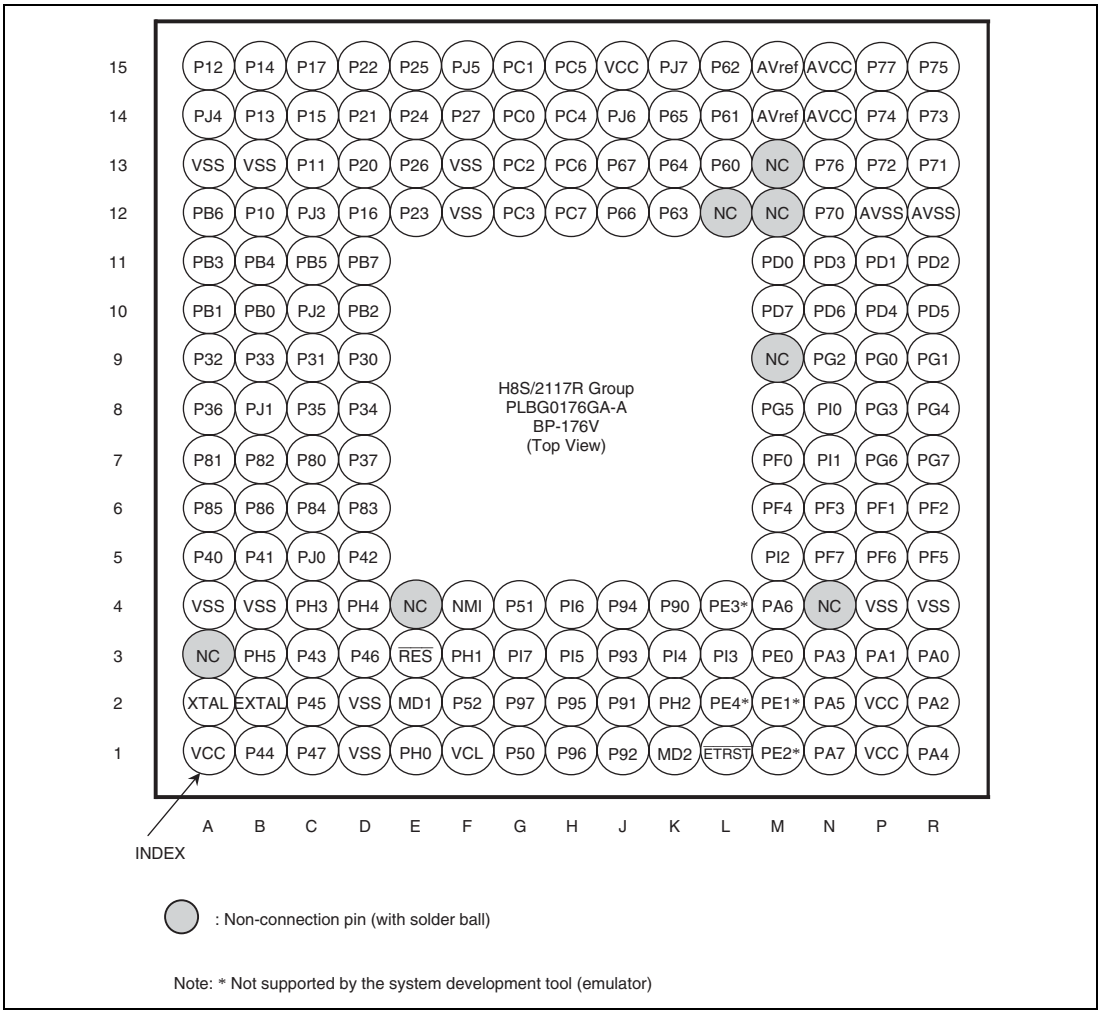


Figure 1.4 Pin Assignments (BP-176V)

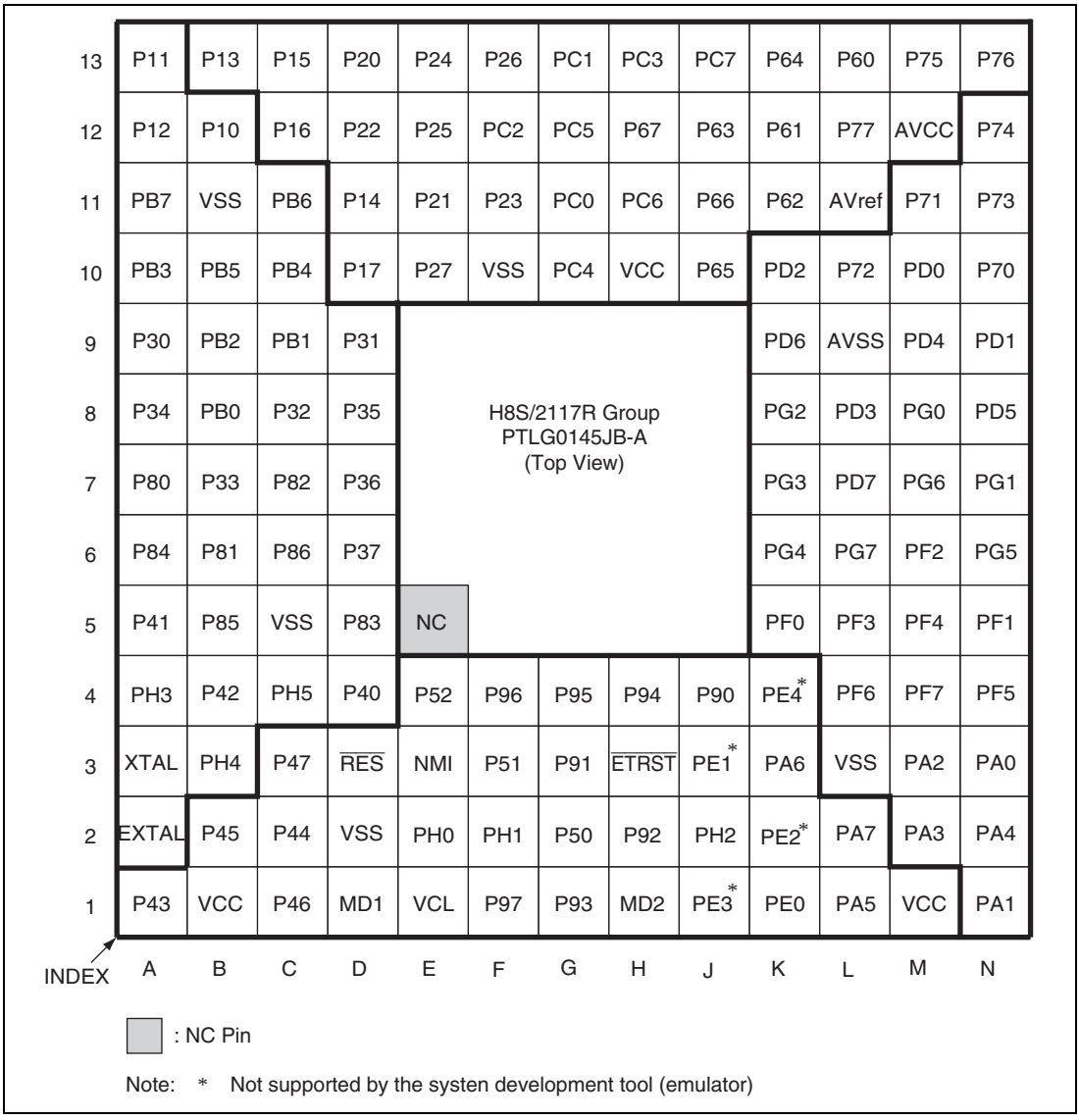


Figure 1.5 Pin Assignments (TLP-145V)

1.4.2 Pin Assignment in Each Operating Mode

Table 1.3 H8S/2117R Group Pin Assignment in Each Operating Mode

Pin No.			Pin Name
TFP-144V	BP-176V	TLP-145V	Single-Chip Mode
			Mode 2 (EXPE = 0)
1	A1	B1	VCC
2	C3	A1	P43/TMI1/SCK2/TCMCKI1/TCMMCI1
3	B1	C2	P44/TMO1/PWMU2B/TCMCYI2
4	C2	B2	P45/PWMU3B/TCMCKI2/TCMMCI2
5	D3	C1	P46/PWX0/PWMU4B/TCMCYI3
6	C1	C3	P47/PWX1/PWMU5B/TCMCKI3/TCMMCI3
7	D2	D2	VSS
—	E4	—	NC
8	E3	D3	$\overline{\text{RES}}$
—	D1	—	VSS
9	E2	D1	MD1
10	E1	E2	PH0/$\overline{\text{ExIRQ6}}$/TDPCYI2
11	F4	E3	NMI
12	F3	F2	PH1/$\overline{\text{ExIRQ7}}$/TDPCKI2/TDPMCI2
13	F1	E1	VCL
14 (N)	F2 (N)	E4 (N)	P52/SCL0
15	G4	F3	P51/FRxD
—	G3 (N)	—	PI7
16	G1	G2	P50/FTxD
17 (N)	G2 (N)	F1 (N)	P97/SDA0/$\overline{\text{IRQ15}}$
—	H4 (N)	—	PI6
—	H3 (N)	—	PI5
18	H1	F4	P96/ϕ/EXCL
19	H2	G4	P95/$\overline{\text{IRQ14}}$
20	J4	H4	P94/$\overline{\text{IRQ13}}$
21	J3	G1	P93/$\overline{\text{IRQ12}}$

Pin No.			Pin Name
TFP-144V	BP-176V	TLP-145V	Single-Chip Mode
			Mode 2 (EXPE = 0)
22	J1	H2	P92/$\overline{\text{IRQ0}}$
23	J2	G3	P91/$\overline{\text{IRQ1}}$
24	K4	J4	P90/$\overline{\text{IRQ2}}$
—	K3 (N)	—	PI4
25	K1	H1	MD2
26	K2	J2	PH2/CIRI
—	L3 (N)	—	PI3
27	L1	H3	$\overline{\text{ETRST}}$
28 (T)	L2 (T)	K4 (T)	PE4*/ETMS
29	L4	J1	PE3*/ETDO
30 (T)	M1 (T)	K2 (T)	PE2*/ETDI
31 (T)	M2 (T)	J3 (T)	PE1*/ETCK
32 (T)	M3 (T)	K1 (T)	PE0/ExEXCL
33 (N)	N1 (N)	L2 (N)	PA7/$\overline{\text{KIN15}}$/PS2CD
34 (N)	M4 (N)	K3 (N)	PA6/$\overline{\text{KIN14}}$/PS2CC
35 (N)	N2 (N)	L1 (N)	PA5/$\overline{\text{KIN13}}$/PS2BD
36	P1	M1	VCC
—	P2	—	VCC
37 (N)	R1 (N)	N2 (N)	PA4/$\overline{\text{KIN12}}$/PS2BC
38 (N)	N3 (N)	M2 (N)	PA3/$\overline{\text{KIN11}}$/PS2AD
39 (N)	R2 (N)	M3 (N)	PA2/$\overline{\text{KIN10}}$/PS2AC
40 (N)	P3 (N)	N1 (N)	PA1/$\overline{\text{KIN9}}$/PA2DD
—	N4	—	NC
41 (N)	R3 (N)	N3 (N)	PA0/$\overline{\text{KIN8}}$/PA2DC
42	P4	L3	VSS
—	M5 (N)	—	PI2
—	R4	—	VSS
43	N5	M4	PF7/PWMU5A
44	P5	L4	PF6/PWMU4A

Pin No.			Pin Name
TFP-144V	BP-176V	TLP-145V	Single-Chip Mode
			Mode 2 (EXPE = 0)
45	R5	N4	PF5/PWMU3A
46	M6	M5	PF4/PWMU2A
47	N6	L5	PF3/$\overline{\text{IRQ11}}$/TMOX/TDPCKI0/TDPMCI0
48	R6	M6	PF2/$\overline{\text{IRQ10}}$/TMOY/TDPCYI0
49	P6	N5	PF1/$\overline{\text{IRQ9}}$/PWMU1A
50	M7	K5	PF0/$\overline{\text{IRQ8}}$/PWMU0A
—	N7 (N)	—	PI1
51 (N)	R7 (N)	L6 (N)	PG7/$\overline{\text{ExIRQ15}}$/ExSCLB
52 (N)	P7 (N)	M7 (N)	PG6/$\overline{\text{ExIRQ14}}$/ExSDAB
53 (N)	M8 (N)	N6 (N)	PG5/$\overline{\text{ExIRQ13}}$/ExSCLA
—	N8 (N)	—	PI0
54 (N)	R8 (N)	K6 (N)	PG4/$\overline{\text{ExIRQ12}}$/ExSDAA
55 (N)	P8 (N)	K7 (N)	PG3/$\overline{\text{ExIRQ11}}$/SCL2
—	M9 (N)	—	NC
56 (N)	N9 (N)	K8 (N)	PG2/$\overline{\text{ExIRQ10}}$/SDA2
57 (N)	R9 (N)	N7 (N)	PG1/$\overline{\text{ExIRQ9}}$/TMIY/TDPCKI1/TDPMCI1
58 (N)	P9 (N)	M8 (N)	PG0/$\overline{\text{ExIRQ8}}$/TMIX/TDPCYI1
59	M10	L7	PD7/AN15
60	N10	K9	PD6/AN14
61	R10	N8	PD5/AN13
62	P10	M9	PD4/AN12
63	N11	L8	PD3/AN11
64	R11	K10	PD2/AN10
65	P11	N9	PD1/AN9
66	M11	M10	PD0/AN8
67	R12	L9	AVSS
—	P12	—	AVSS
68	N12	N10	P70/AN0
69	R13	M11	P71/AN1

Pin No.			Pin Name
TFP-144V	BP-176V	TLP-145V	Single-Chip Mode
			Mode 2 (EXPE = 0)
—	M12	—	NC
70	P13	L10	P72/AN2
71	R14	N11	P73/AN3
72	P14	N12	P74/AN4
73	R15	M13	P75/AN5
74	N13	N13	P76/AN6
75	P15	L12	P77/AN7
76	N14	M12	AVCC
—	M13	—	NC
—	N15	—	AVCC
77	M14	L11	AVref
—	L12	E5	NC
—	M15	—	AVref
78	L13	L13	P60/$\overline{\text{KIN0}}$
79	L14	K12	P61/$\overline{\text{KIN1}}$
80	L15	K11	P62/$\overline{\text{KIN2}}$
81	K12	J12	P63/$\overline{\text{KIN3}}$
82	K13	K13	P64/$\overline{\text{KIN4}}$
—	K15	—	PJ7
83	K14	J10	P65/$\overline{\text{KIN5}}$
84	J12	J11	P66/$\overline{\text{IRQ6}}$/$\overline{\text{KIN6}}$
85	J13	H12	P67/$\overline{\text{IRQ7}}$/$\overline{\text{KIN7}}$
86	J15	H10	VCC
—	J14	—	PJ6
87	H12	J13	PC7/TIOCB2/TCLKD/$\overline{\text{WUE15}}$
88	H13	H11	PC6/TIOCA2/$\overline{\text{WUE14}}$
89	H15	G12	PC5/TIOCB1/TCLKC/$\overline{\text{WUE13}}$
90	H14	G10	PC4/TIOCA1/$\overline{\text{WUE12}}$
91	G12	H13	PC3/TIOCD0/TCLKB/$\overline{\text{WUE11}}$

Pin No.			Pin Name
TFP-144V	BP-176V	TLP-145V	Single-Chip Mode
			Mode 2 (EXPE = 0)
92	G13	F12	PC2/TIOCC0/TCLKA/$\overline{\text{WUE10}}$
93	G15	G13	PC1/TIOCB0/$\overline{\text{WUE9}}$
94	G14	G11	PC0/TIOCA0/$\overline{\text{WUE8}}$
95	F12	F10	VSS
—	F13	—	VSS
—	F15	—	PJ5
96	F14	E10	P27
97	E13	F13	P26
98	E15	E12	P25
99	E14	E13	P24
100	E12	F11	P23
101	D15	D12	P22
102	D14	E11	P21
103	D13	D13	P20
104	C15	D10	P17
105	D12	C12	P16
106	C14	C13	P15
107	B15	D11	P14
108	B14	B13	P13
109	A15	A12	P12
110	C13	A13	P11
—	A14	—	PJ4
111	B13	B11	VSS
—	C12	—	PJ3
—	A13	—	VSS
112	B12	B12	P10
113	D11	A11	PB7/$\overline{\text{RTS}}$/FSISS
114	A12	C11	PB6/$\overline{\text{CTS}}$/FSICK
115	C11	B10	PB5/$\overline{\text{DTR}}$/FSIDI

Pin No.			Pin Name
TFP-144V	BP-176V	TLP-145V	Single-Chip Mode Mode 2 (EXPE = 0)
116	B11	C10	PB4/$\overline{\text{DSR}}$/FSIDO
117	A11	A10	PB3/$\overline{\text{DCD}}$/PWMU1B
118	D10	B9	PB2/$\overline{\text{RI}}$/PWMU0B
—	C10	—	PJ2
119	A10	C9	PB1/LSCI
120	B10	B8	PB0/$\overline{\text{LSMI}}$
121	D9	A9	P30/LAD0
122	C9	D9	P31/LAD1
123	A9	C8	P32/LAD2
124	B9	B7	P33/LAD3
125	D8	A8	P34/$\overline{\text{LFRAME}}$
126	C8	D8	P35/$\overline{\text{LRESET}}$
127	A8	D7	P36/LCLK
—	B8	—	PJ1
128	D7	D6	P37/SERIRQ
129	C7	A7	P80/$\overline{\text{PME}}$
130	A7	B6	P81/GA20
131	B7	C7	P82/$\overline{\text{CLKRUN}}$
132	D6	D5	P83/$\overline{\text{LPCPD}}$
133	C6	A6	P84/$\overline{\text{IRQ3}}$/TxD1
134	A6	B5	P85/$\overline{\text{IRQ4}}$/RxD1
135 (N)	B6 (N)	C6 (N)	P86/$\overline{\text{IRQ5}}$/SCK1/SCL1
—	C5	—	PJ0
136	A5	D4	P40/TMI0/TxD2/TCMCYI0
137	B5	A5	P41/TMO0/RxD2/TCMCKI0/TCMMCI0
138 (N)	D5 (N)	B4 (N)	P42/SDA1/TCMCYI1
139	A4	C5	VSS
—	B4	—	VSS
140	C4	A4	PH3

Pin No.			Pin Name
TFP-144V	BP-176V	TLP-145V	Single-Chip Mode
			Mode 2 (EXPE = 0)
—	A3	—	NC
141	D4	B3	PH4
142	B3	C4	PH5
143	A2	A3	XTAL
144	B2	A2	EXTAL

Notes: (N) in Pin No. indicates the pin is driven by NMOS push-pull/open drain and has 5 V input tolerance.

(T) in Pin No. indicates the pin has 5 V input tolerance.

* This pin is not supported by the system development tool (emulator).

1.4.3 Pin Functions

Table 1.4 Pin Functions

Type	Symbol	Pin No. TFP-144V	BP-176V	TLP-145V	I/O	Name and Function
Power supply	VCC	1, 36, 86	A1, J15, P1, P2	B1, M1, H10	Input	Power supply pins. Connect all these pins to the system power supply. Connect the bypass capacitor between VCC and VSS (that is located near these pins).
	VCL	13	F1	E1	Input	External capacitance pin for internal step-down power. Connect this pin to VSS through an external capacitor (that is located near this pin) to stabilize internal step-down power.
	VSS	7, 42, 95, 111, 139	D1, D2, P4, R4, F12, F13, B13, A13, A4, B4	D2, L3, F10, B11, C5	Input	Ground pins. Connect all these pins to the system power supply (0 V).
Clock	XTAL	143	A2	A3	Input	For connection to a crystal resonator. An external clock can be supplied from the EXTAL pin. For an example of crystal resonator connection, see section 25, Clock Pulse Generator.
	EXTAL	144	B2	A2	Input	
	ϕ	18	H1	F4	Output	Supplies the system clock to external devices.
	EXCL	18	H1	F4	Input	32.768 kHz external sub clock should be supplied. To which pin the external clock is input can be selected from the EXCL or ExEXCL pin.
	ExEXCL	32	M3	K1	Input	
Operating mode control	MD2 MD1	25 9	K1 E2	H1 D1	Input	These pins set the operating mode. Inputs at these pins should not be changed during operation.
System control	$\overline{\text{RES}}$	8	E3	D3	Input	Reset pin. When this pin is low, the chip is reset.

Type	Symbol	Pin No. TFP-144V	BP-176V	TLP-145V	I/O	Name and Function
Interrupts	NMI	11	F4	E3	Input	Nonmaskable interrupt request input pin
	$\overline{\text{IRQ15}}$ to $\overline{\text{IRQ0}}$	17, 19 to 21, 47 to 50, 85, 84, 135 to 133, 24 to 22	G2, H2, J4, J3, N6, R6, P6, M7, J13, J12, B6, A6, C6, K4, J2, J1	F1, G4, H4, G1, L5, M6, N5, K5, H12, J11, C6, B5, A6, H2, G3, J4	Input	These pins request a maskable interrupt. To which pin an IRQ interrupt is input can be selected from the $\overline{\text{IRQn}}$ or $\overline{\text{ExIQRn}}$ pin. (n = 15 to 6)
	$\overline{\text{ExIRQ15}}$ to $\overline{\text{ExIRQ6}}$	51 to 58, 12, 10	R7, P7, M8, R8, P8, N9, R9, P9, F3, E1	L6, M7, N6, K6, K7, K8, N7, M8, F2, E2	Input	
H-UDI	$\overline{\text{ETRST}}$*[2]	27	L1	H3	Input	Interface pins for emulator
	ETMS	28	L2	K4	Input	Reset by holding the $\overline{\text{ETRST}}$ pin to low level regardless of the H-UDI activation. At this time, the $\overline{\text{ETRST}}$ pin should be held low level for 20 clocks of ETCK. Then, to activate the H-UDI, the $\overline{\text{ETRST}}$ pin should be set to high level and the pins ETCK, ETMS, and ETDI should be set appropriately. In the normal operation without activating the H-UDI, pins ETCK, ETMS, ETDI, and ETDO should be pulled up to high level. The $\overline{\text{ETRST}}$ pin is pulled up inside the chip.
	ETDO	29	L4	J1	Output	
	ETDI	30	M1	K2	Input	
	ETCK	31	M2	J3	Input	
8-bit timer (TMR_0, TMR_1, TMR_X, TMR_Y)	TMO0 TMO1 TMOX TMOY	137 3 47 48	B5 B1 N6 R6	A5 C2 L5 M6	Output	Waveform output pins with output compare function
	TMI0 TMI1 TMIX TMIY	136 2 58 57	A5 C3 P9 R9	D4 A1 M8 N7	Input	Counter event input and count reset input pins

Type	Symbol	Pin No. TFP-144V	BP-176V	TLP-145V	I/O	Name and Function
16-bit timer pulse unit (TPU)	TCLKA TCLKB TCLKC TCLKD	92 91 89 87	G13 G12 H15 H12	F12 H13 G12 J13	Input	Timer external clock input pins
	TIOCA0 TIOCB0 TIOCC0 TIOCD0	94 93 92 91	G14 G15 G13 G12	G11 G13 F12 H13	Input/ Output	Input capture input/output compare output/PWM output pins for TGRA_0 to TGRD_0
	TIOCA1 TIOCB1	90 89	H14 H15	G10 G12	Input/ Output	Input capture input/output compare output/PWM output pins for TGRA_1 and TGRB_1
	TIOCA2 TIOCB2	88 87	H13 H12	H11 J13	Input/ Output	Input capture input/output compare output/PWM output pins for TGRA_2 and TGRB_2
16-bit cycle measure-ment timer (TCM)	TCMCKI3 to TCMCKI0	6, 4, 2, 137	C1, C2, C3, B5	C3, B2, A1, A5	Input	Timer external clock input pins
	TCMMCI3 to TCMMCI0	6, 4, 2, 137	C1, C2, C3, B5	C3, B2, A1, A5	Input	Cycle measurement enable input pins
	TCMCYI3 to TCMCYI0	5, 3, 138, 136	D3, B1, D5, A5	C1, C2, B4, D4	Input	Timer input capture input pins
16-bit duty period measure-ment timer (TDP)	TDPCKI2 to TDPCKI0	12, 57, 47	F3, R9, N6	F2, N7, L5	Input	Timer external clock input pins
	TDPMCI2 to TDPMCI0	12, 57, 47	F3, R9, N6	F2, N7, L5	Input	Cycle measurement enable input pins
	TDPCYI2 to TDPCYI0	10, 58, 47	E1, P9, R6	E2, M8, M6	Input	Timer input capture input pins
8-bit PWM timer U (PWMU)	PWMU5A to PWMU0A PWMU5B to PWMU0B	43 to 46, 49, 50, 6 to 3, 117, 118	N5, P5, R5, M6, P6, M7, C1, D3, C2, B1, A11, D10	M4, L4, N4, M5, N5, K5, C3, C1, B2, C2, A10, B9	Output	PWM timer pulse output pins

Type	Symbol	Pin No. TFP-144V	BP-176V	TLP-145V	I/O	Name and Function
14-bit PWM timer (PWMX)	PWX0 PWX1	5 6	D3 C1	C1 C3	Output	PWM timer pulse output pins
Serial communi-cation interface (SCI_1, SCI_2)	TxD1 TxD2	133 136	C6 A5	A6 D4	Output	Transmit data output pins
	RxD1 RxD2	134 137	A6 B5	B5 A5	Input	Receive data input pins
	SCK1 SCK2	135 2	B6 C3	C6 A1	Input/ Output	Clock input/output pins Output type of SCK1 and SCK2 is NMOS push-pull
Keyboard buffer control unit (PS2)	PS2AC PS2BC PS2CC PS2DC	39 37 34 41	R2 R1 M4 R3	M3 N2 K3 N3	Input/ Output	Synchronous clock input/output pins for the keyboard buffer control unit
	PS2AD PS2BD PS2CD PS2DD	38 35 33 40	N3 N2 N1 P3	M2 L1 L2 N1	Input/ Output	Data input/output pins for the keyboard buffer control unit
Keyboard control	$\overline{\text{KIN15}}$ to $\overline{\text{KIN0}}$	33 to 35, 37 to 41, 85 to 78	N1, M4, N2, R1, N3, R2, P3, R3, J13, J12, K14, K13, K12, L15, L14, L13	L2, K3, L1, N2, M2, M3, N1, N3, H12, J11, J10, K13, J12, K11, K12, L13	Input	Input pins for matrix keyboard. Normally, $\overline{\text{KIN15}}$ to $\overline{\text{KIN0}}$ function as key scan inputs, and P17 to P10 and P27 to P20 function as key scan outputs. Thus, composed with a maximum of 16 outputs x 16 inputs, a 256-key matrix can be configured.
	$\overline{\text{WUE15}}$ to $\overline{\text{WUE8}}$	87 to 94	H12, H13, H15, H14, G12, G13, G15, G14	J13, H11, G12, G10, H13, F12, G13, G11	Input	Wake-up event input pins. Same wake up as key wake up can be performed with various sources.

Type	Symbol	Pin No. TFP-144V	BP-176V	TLP-145V	I/O	Name and Function
Serial communi-cation interface with FIFO (SCIF)	FTxD	16	G1	G2	Output	Transmit data output pin
	FRxD	15	G4	F3	Input	Receive data input pin
	$\overline{RI}$	118	D10	B9	Input	Ring indicator input pin
	$\overline{DCD}$	117	A11	A10	Input	Data carrier detect input pin
	$\overline{DSR}$	116	B11	C10	Input	Data set ready input pin
	$\overline{DTR}$	115	C11	B10	Output	Data terminal ready output pin
	$\overline{CTS}$	114	A12	C11	Input	Transmission permission input pin
	$\overline{RTS}$	113	D11	A11	Output	Transmission request output pin
LPC Interface (LPC)	LAD3 to LAD0	124 to 121	B9, A9, C9, D9	B7, C8, D9, A9	Input/ Output	LPC command, address, and data input/output pins
	$\overline{LFRAME}$	125	D8	A8	Input	Input pin indicating LPC cycle start and forced termination of an abnormal LPC cycle
	$\overline{LRESET}$	126	C8	D8	Input	Input pin indicating LPC reset
	LCLK	127	A8	D7	Input	LPC clock input pin
	SERIRQ	128	D7	D6	Input/ Output	LPC serial host interrupt (HIRQ1 to HIRQ15) input/output pin
	LSCI, $\overline{LSMI}$, $\overline{PME}$	119, 120, 129	A10, B10, C7	C9, B8, A7	Input/ Output	LPC auxiliary output pins. Functionally, they are general I/O ports.
	GA20	130	A7	B6	Input/ Output	GATE A20 control signal output pin. Output state monitoring input is possible.
	$\overline{CLKRUN}$	131	B7	C7	Input/ Output	Input/output pin that requests the start of LCLK operation when LCLK is stopped.
	$\overline{LPCPD}$	132	D6	D5	Input	Input pin that controls LPC module shutdown.

Type	Symbol	Pin No. TFP-144V	BP-176V	TLP-145V	I/O	Name and Function
FSI interface (FSI)	FSISS	113	D11	A11	Output	FSI slave select pin
	FSICK	114	A12	C11	Output	Clock output pin
	FSIDI	115	C11	B10	Input	Receive data input pin
	FSIDO	116	B11	C10	Output	Transmit data output pin
CIR interface (CIR)	CIRI	26	K2	J2	Input	Receive data input pin
A/D converter	AN15 to AN0	59 to 66, 75 to 68	M10, N10, R10, P10, N11, R11, P11, M11, P15, N13, R15, P14, R14, P13, R13, N12	L7, K9, N8, M9, L8, K10, N9, M10, L12, N13, M13, N12, N11, L10, M11, N10	Input	Analog input pins
	AVCC	76	N14, N15	M12	Input	Analog power supply pin for the A/D converter. When the A/D converter is not used, this pin should be connected to the system power supply (+3 V).
	AVref	77	M14, M15	L11	Input	Reference power supply pin for the A/D converter. When the A/D converter is not used, this pin should be connected to the system power supply (+3 V).
	AVSS	67	R12, P12	L9	Input	Ground pin for the A/D converter. This pin should be connected to the system power supply (0 V).

Type	Symbol	Pin No. TFP-144V	BP-176V	TLP-145V	I/O	Name and Function
I^2C bus interface (IIC)	SCL0 SCL1 SCL2 ExSCLA ExSCLB	14 135 55 53 51	F2 B6 P8 M8 R7	E4 C6 K7 N6 L6	Input/ Output	I^2C clock I/O pins. The output type is NMOS open-drain. To which pin the clock is input or output can be selected from the SCL0, SCL1, ExSCLA, and ExSCLB pins.
	SDA0 SDA1 SDA2 ExSDAA ExSDAB	17 138 56 54 52	G2 D5 N9 R8 P7	F1 B4 K8 K6 M7	Input/ Output	I^2C data I/O pins. The output type is NMOS open-drain. To which pin the clock is input or output can be selected from the SDA0, SDA1, ExSDAA, and ExSDAB pins.
I/O port	P17 to P10	104 to 110, 112	C15, D12, C14, B15, B14, A15, C13, B12	D10, C12, C13, D11, B13, A12, A13, B12	Input/ Output	8-bit input/output pins
	P27 to P20	96 to 103	F14, E13, E15, E14, E12, D15, D14, D13	E10, F13, E12, E13, F11, D12, E11, D13	Input/ Output	8-bit input/output pins
	P37 to P30	128 to 121	D7, A8, C8, D8, B9, A9, C9, D9	D6, D7, D8, A8, B7, C8, D9, A9	Input/ Output	8-bit input/output pins
	P47 to P40	6 to 2, 138 to 136	C1, D3, C2, B1, C3, D5, B5, A5	C3, C1, B2, C2, A1, B4, A5, D7	Input/ Output	8-bit input/output pins (The output type of P42 is NMOS push-pull.)
	P52 to P50	14 to 16	F2, G4, G1	E4, F3, G2	Input/ Output	3-bit input/output pins (The output type of P52 is NMOS push-pull.)
	P67 to P60	85 to 78	J13, J12, K14, K13, K12, L15, L14, L13	H12, J11, J10, K13, J12, K11, K12, L13	Input/ Output	8-bit input/output pins

Type	Symbol	Pin No. TFP-144V	Pin No. BP-176V	Pin No. TLP-145V	I/O	Name and Function
I/O port	P77 to P70	75 to 68	P15, N13, R15, P14, R14, P13, R13, N12	L12, N13, M13, N12, N11, L10, M11, N10	Input	8-bit input pins
	P86 to P80	135 to 129	B6, A6, C6, D6, B7, A7, C7	C6, B5, A6, D5, C7, B6, A7	Input/ Output	7-bit input/output pins (The output type of P86 is NMOS push-pull.)
	P97 to P90	17 to 24	G2, H1, H2, J4, J3, J1, J2, K4	F1, F4, G4, H4, G1, H2, G3, J4	Input/ Output	8-bit input/output pins (The output type of P97 is NMOS push-pull.)
	PA7 to PA0	33 to 35, 37 to 41	N1, M4, N2, R1, N3, R2, P3, R3	L2, K3, L1, N2, M2, M3, N1, N3	Input/ Output	8-bit input/output pins (The output type of PA7 to PA0 is NMOS push-pull.)
	PB7 to PB0	113 to 120	D11, A12, C11, B11, A11, D10, A10, B10	A11, C11, B10, C10, A10, B9, C9, B8	Input/ Output	8-bit input/output pins
	PC7 to PC0	87 to 94	H12, H13, H15, H14, G12, G13, G15, G14	J13, H11, G12, G10, H13, F12, G13, G11	Input/ Output	8-bit input/output pins
	PD7 to PD0	59 to 66	M10, N10, R10, P10, N11, R11, P11, M11	L7, K9, N8, M9, L8, K10, N9, M10	Input/ Output	8-bit input/output pins
	PE4 to PE0*[1]	28 to 32	L2, L4, M1, M2, M3	K4, J1, K2, J3, K1	Input	5-bit input pins
	PF7 to PF0	43 to 50	N5, P5, R5, M6, N6, R6, P6, M7	M4, L4, N4, M5, L5, M6, N5, K5	Input/ Output	8-bit input/output pins
	PG7 to PG0	51 to 58	R7, P7, M8, R8, P8, N9, R9, P9	L6, M7, N6, K6, K7, K8, N7, M8	Input/ Output	8-bit input/output pins (The output type of PG7 to PG0 is NMOS push-pull.)

Type	Symbol	Pin No. TFP-144V	Pin No. BP-176V	Pin No. TLP-145V	I/O	Name and Function
I/O port	PH5 to PH0	142 to 140, 26, 12, 10	B3, D4, C4, K2, F3, E1	C4, B3, A4, J2, F2, E2	Input/ Output	6-bit input/output pins
	PI7 to PI0	—	G3, H4, H3, K3, L5, M5, N7, N8	—	Input/ Output	8-bit input/output pins (The output type of PI7 to PI0 is NMOS push-pull.)
	PJ7 to PJ0	—	K15, J14, F15, A14, C12, C10, B8, C5	—	Input/ Output	8-bit input/output pins

Notes: 1. Pins PE4 to PE1 are not supported by the system development tool (emulator).

2. Following precautions are required on the power-on reset signal that is applied to the $\overline{\text{ETRST}}$ pin.
The reset signal should be applied on power supply.
Set apart the power-on reset circuit from this LSI to prevent the $\overline{\text{ETRST}}$ pin of the emulator from affecting the operation of this LSI.
Set apart the power-on reset circuit from this LSI to prevent the system reset of this LSI from affecting the $\overline{\text{ETRST}}$ pin of the emulator.

Section 2 CPU

The H8S/2600 CPU is a high-speed central processing unit with an internal 32-bit architecture that is upward-compatible with the H8/300 and H8/300H CPUs. The H8S/2600 CPU has sixteen 16-bit general registers, can address a 16-Mbyte linear address space, and is ideal for realtime control. This section describes the H8S/2600 CPU. The usable modes and address spaces differ depending on the product. For details on each product, refer to section 3, MCU Operating Modes.

2.1 Features

- 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-bit registers
- 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

— 16 ÷ 8-bit register-register divide: 12 states
— 16 × 16-bit register-register multiply: 3 states
— 32 ÷ 16-bit register-register divide: 20 states

- Two CPU operating modes
  — Normal mode*
  — Advanced mode
- Power-down state
  — Transition to power-down state by the SLEEP instruction
  — CPU clock speed selection

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 H8S/2600 CPU only.
- The number of execution states of the MULXU and MULXS instructions;

		Execution States	
Instruction	**Mnemonic**	**H8S/2600**	**H8S/2000**
MULXU	MULXU.B Rs, Rd	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 functions, and power-down modes, etc., depending on the model.

2.1.2 Differences from H8/300 CPU

In comparison to the H8/300 CPU, the H8S/2600 CPU has the following enhancements:

- More general registers and control registers
  - Eight 16-bit extended registers, and one 8-bit and two 32-bit control registers, have been added.
- Expanded address space
  - Normal mode supports the same 64-kbyte address space as the H8/300 CPU.
  - Advanced mode supports a maximum 16-Mbyte address space.
- Enhanced addressing
  - The addressing modes have been enhanced to make effective use of the 16-Mbyte address 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 enhancements:

- 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.

2.2 CPU Operating Modes

The H8S/2600 CPU has two operating modes: normal and advanced. Normal mode supports a maximum 64-kbyte address space. Advanced mode supports a maximum 16-Mbyte total address space. The mode is selected by the mode pins.

2.2.1 Normal Mode

The exception vector table and stack have the same structure as in the H8/300 CPU.

- Address Space

  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-bit segments of 32-bit registers. When En is used as a 16-bit register it can contain any value, even when the corresponding general register (Rn) is used as an address register. If the general register is referenced in the register indirect addressing mode with pre-decrement (@–Rn) or post-increment (@Rn+) and a carry or borrow occurs, however, the value in the corresponding 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 table. One branch address is stored per 16 bits. The exception vector table structure in normal mode is 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 instructions uses an 8-bit absolute address included in the instruction code to specify a memory operand 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'0000 to 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 PC, condition-code register (CCR), and extended control register (EXR) are pushed onto the stack in exception handling, they are stored as shown in figure 2.2. EXR is not pushed onto the stack in interrupt control mode 0. For details, see section 4, Exception Handling.

Note: Normal mode is not available in this LSI.

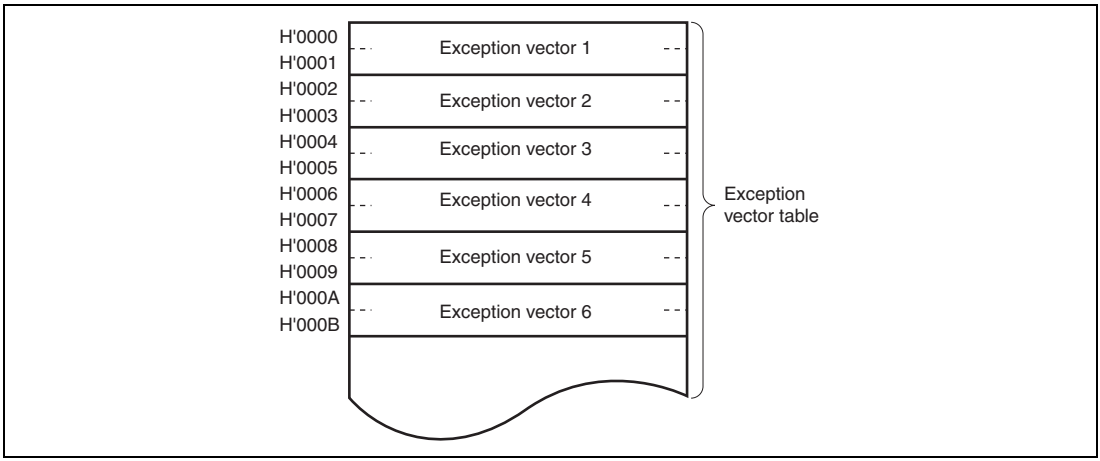
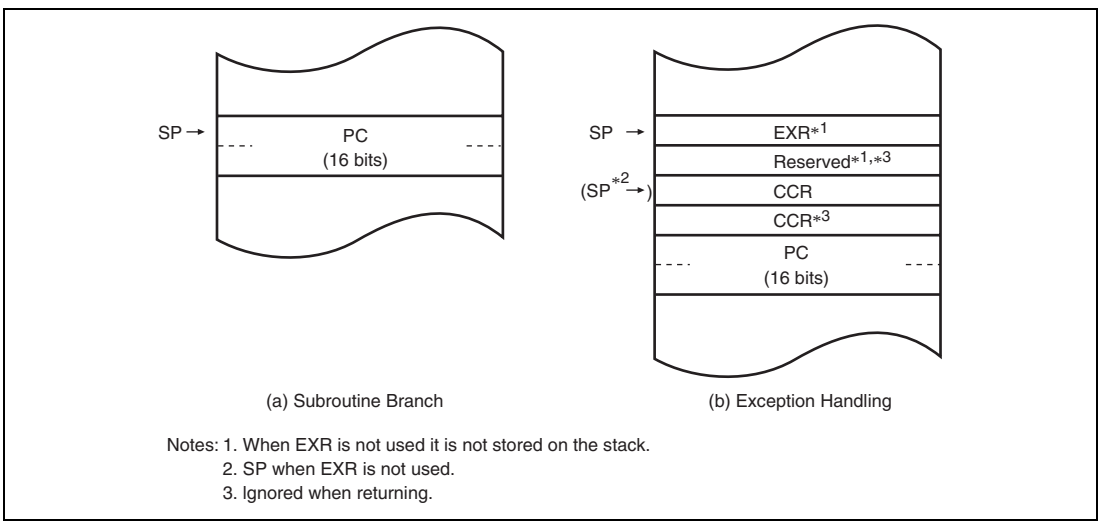


Figure 2.1 Exception Vector Table (Normal Mode)



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

Figure 2.2 Stack Structure in Normal Mode

2.2.2 Advanced Mode

- Address Space

  Linear access to a 16-Mbyte 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-bit segments of 32-bit registers or address registers.
- Instruction Set

  All instructions and addressing modes can be used.
- Exception Vector Table and Memory Indirect Branch Addresses

  In advanced mode, the top area starting at H'00000000 is allocated to the exception vector table in units of 32 bits. In each 32 bits, the upper 8 bits are ignored and a branch address is stored in the lower 24 bits (figure 2.3). For details of the exception vector table, see section 4, Exception Handling.

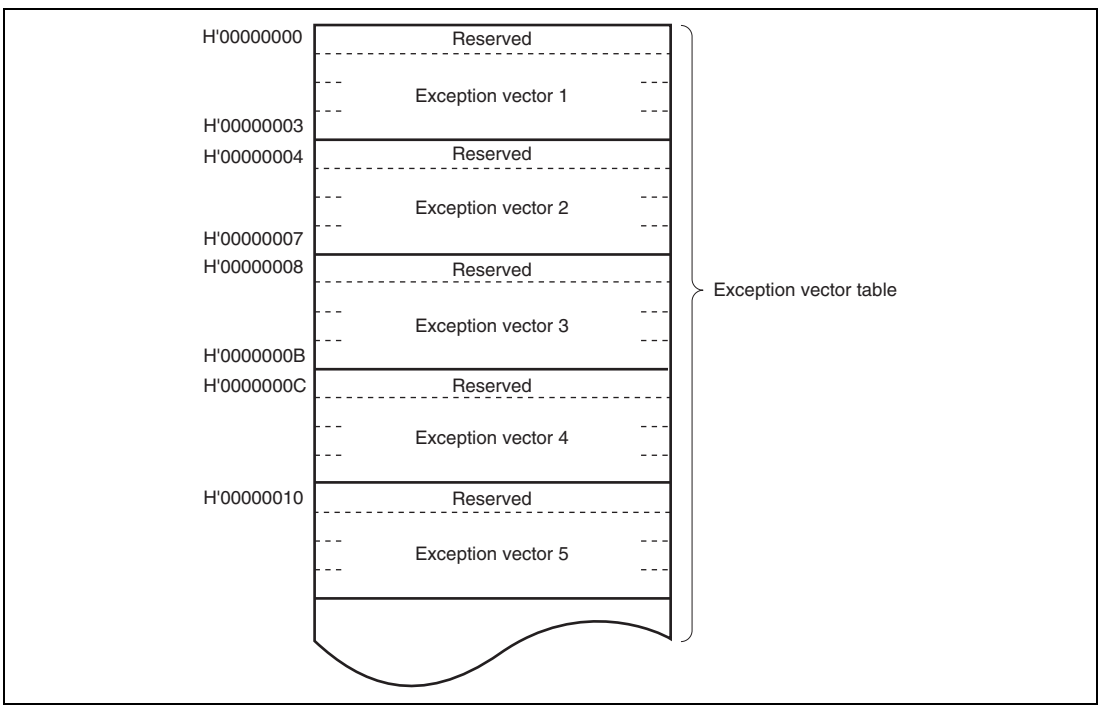
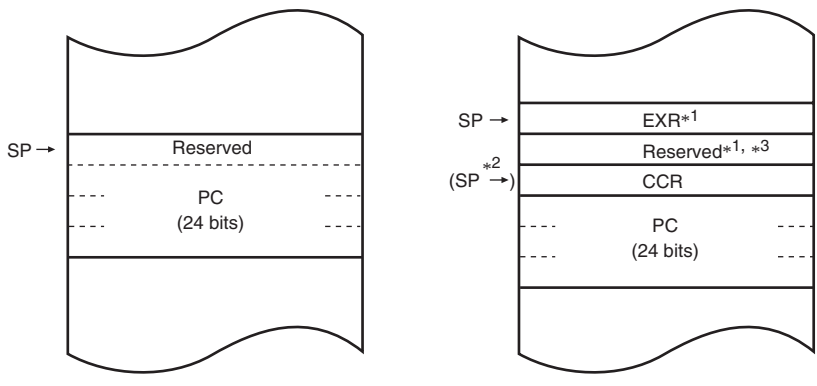


Figure 2.3 Exception Vector Table (Advanced Mode)

The memory indirect addressing mode (@@aa:8) employed in the JMP and JSR instructions uses an 8-bit absolute address included in the instruction code to specify a memory operand that contains a branch address. In advanced mode the operand is a 32-bit longword operand, providing a 32-bit branch address. The upper 8 bits of these 32 bits is a reserved area that is regarded as H'00. Branch addresses can be stored in the area from H'00000000 to H'000000FF. Note that the first part of this range is also used for the exception vector table.

- Stack Structure

  In advanced mode, when the program counter (PC) is pushed onto the stack in a subroutine call, and the PC, condition-code register (CCR), and extended control register (EXR) are pushed onto the stack in exception handling, they are stored as shown in figure 2.4. When EXR is not pushed onto the stack in interrupt control mode 0. For details, see section 4, Exception Handling.



(a) Subroutine Branch (b) Exception Handling

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

Figure 2.4 Stack Structure in Advanced Mode

2.3 Address Space

Figure 2.5 shows a memory map for the H8S/2600 CPU. The H8S/2600 CPU provides linear access to a maximum 64-kbyte address space in normal mode, and a maximum 16-Mbyte (architecturally 4-Gbyte) address space in advanced mode. The usable modes and address spaces differ depending on the product. For details on each product, refer to section 3, MCU Operating Modes.

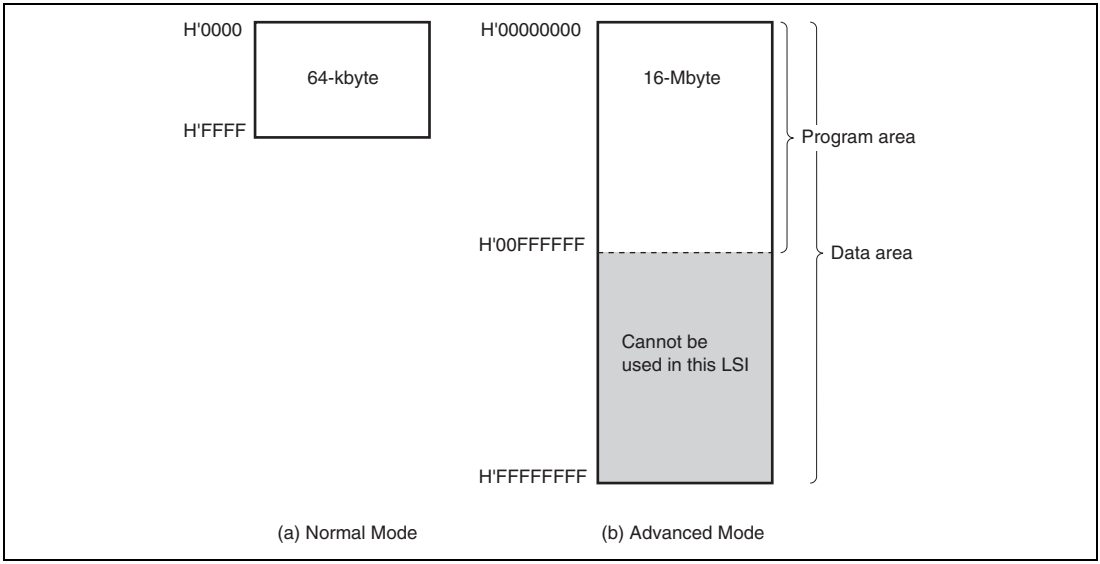


Figure 2.5 Memory Map

2.4 Registers

The H8S/2600 CPU has the internal registers shown in figure 2.6. There are two types of registers; general registers and control registers. The control registers are a 24-bit program counter (PC), an 8-bit extended control register (EXR), an 8-bit condition code register (CCR), and a 64-bit multiply-accumulate register (MAC).

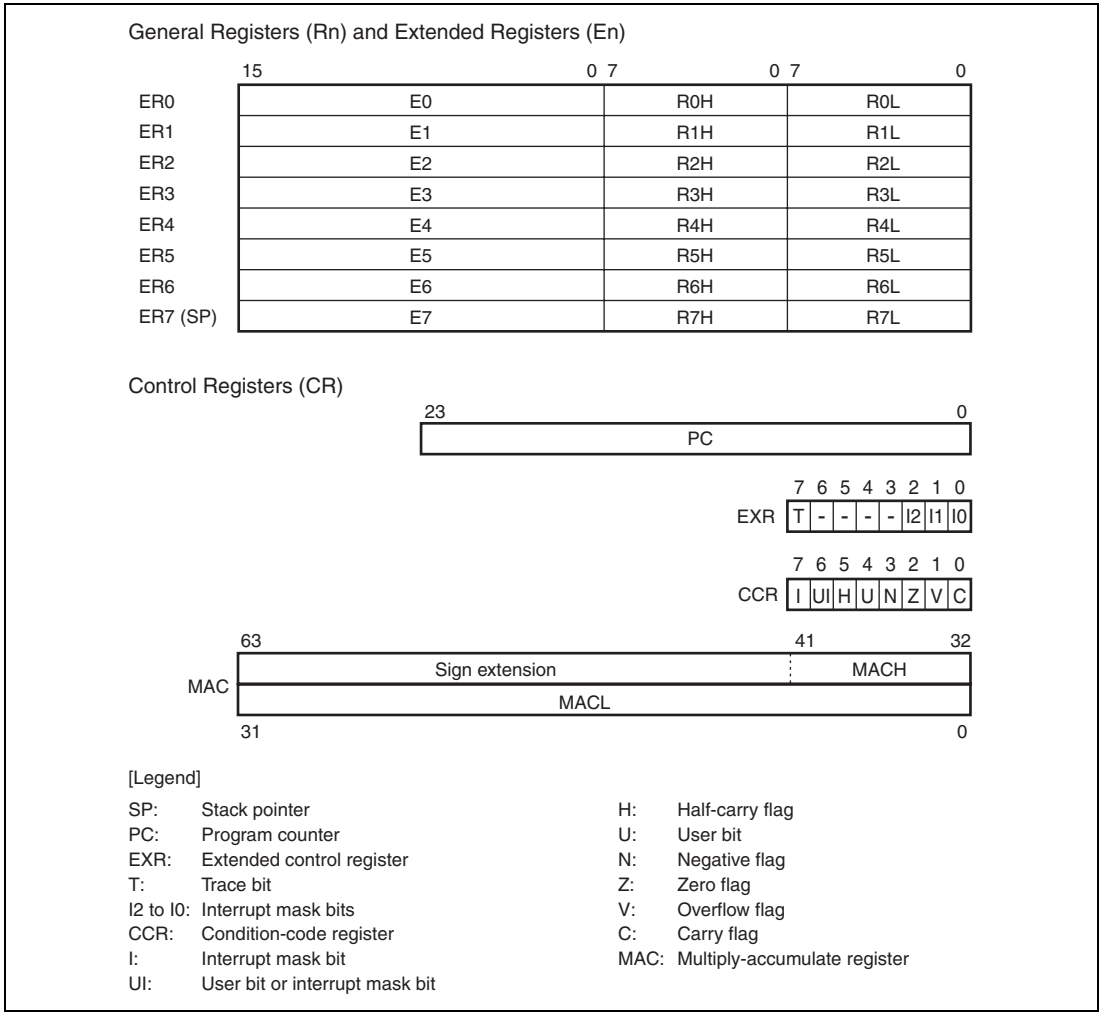


Figure 2.6 CPU Registers

2.4.1 General Registers

The H8S/2600 CPU has eight 32-bit general registers. These general registers are all functionally identical and can be used as both address registers and data registers. When a general register is used as a data register, it can be accessed as a 32-bit, 16-bit, or 8-bit register. Figure 2.7 illustrates the usage of the general registers. When the general registers are used as 32-bit registers or address registers, they are designated by the letters ER (ER0 to ER7).

The ER registers divide into 16-bit general registers designated by the letters E (E0 to E7) and R (R0 to R7). These registers are functionally equivalent, providing a maximum of sixteen 16-bit registers. The E registers (E0 to E7) are also referred to as extended registers.

The R registers divide into 8-bit general registers designated by the letters RH (R0H to R7H) and RL (R0L to R7L). These registers are functionally equivalent, providing a maximum of sixteen 8-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-register function, and is used implicitly in exception handling and subroutine calls. Figure 2.8 shows the stack.

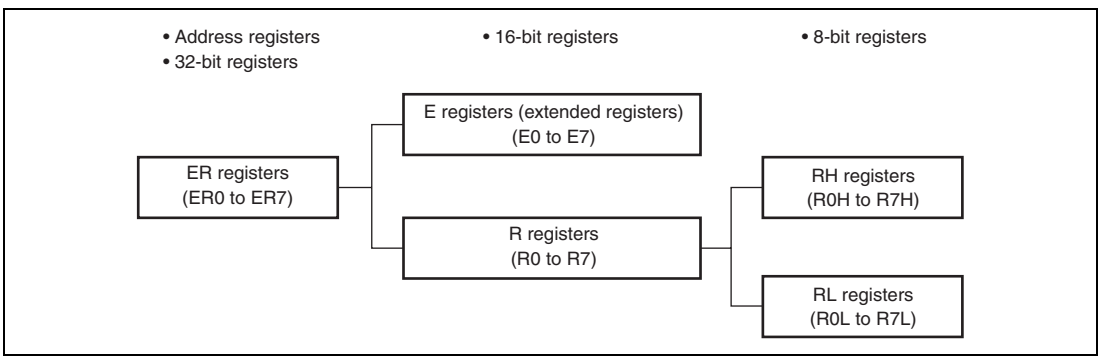


Figure 2.7 Usage of General Registers

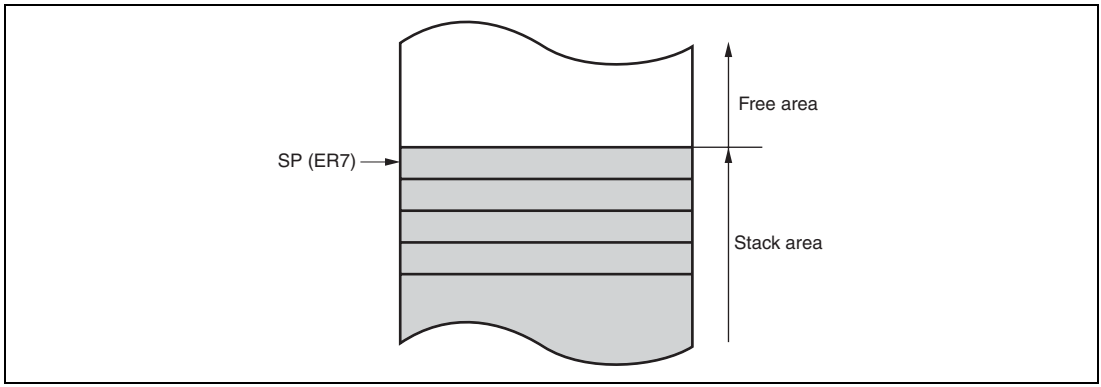


Figure 2.8 Stack

2.4.2 Program Counter (PC)

This 24-bit counter indicates the address of the next instruction the CPU will execute. The length of all CPU instructions is 2 bytes (one word), so the least significant PC bit is ignored. (When an 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 instructions. When these instructions, except for the STC instruction, are executed, all interrupts including NMI 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 generated each time an instruction is executed. When this bit is cleared to 0, instructions are executed in sequence.
6 to 3	—	All 1	—	Reserved These bits are always read as 1.
2	I2	1	R/W	These bits designate the interrupt mask level (0 to 7). For details, refer to section 5, Interrupt Controller.
1	I1	1	R/W	
0	I0	1	R/W	

2.4.4 Condition-Code Register (CCR)

This 8-bit register contains internal CPU status information, including an interrupt mask bit (I) and half-carry (H), negative (N), zero (Z), overflow (V), and carry (C) flags.

Operations can be performed on the CCR bits by the LDC, STC, ANDC, ORC, and XORC instructions. The N, Z, V, and C flags are used as branching conditions for conditional branch (Bcc) instructions.

Bit	Bit Name	Initial Value	R/W	Description
7	I	1	R/W	Interrupt Mask Bit Masks interrupts other than NMI when set to 1. NMI is accepted regardless of the I bit setting. The I bit is set to 1 at the start of an exception-handling sequence. For 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 LDC, STC, ANDC, ORC, and XORC instructions. This bit cannot be 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, CMP.B, or NEG.B instruction is executed, this flag is set to 1 if there is a carry or borrow at bit 3, and cleared to 0 otherwise. When the ADD.W, SUB.W, CMP.W, or NEG.W instruction is executed, the H flag is set to 1 if there is a carry or borrow at bit 11, and cleared to 0 otherwise. When the ADD.L, SUB.L, CMP.L, or NEG.L instruction is executed, the H flag is set to 1 if there is a carry or borrow at bit 27, and cleared to 0 otherwise.
4	U	Undefined	R/W	User Bit Can be read or written by software using the LDC, STC, ANDC, ORC, and XORC instructions.
3	N	Undefined	R/W	Negative Flag Stores the value of the most significant bit of data as a sign bit.
2	Z	Undefined	R/W	Zero Flag Set to 1 to indicate zero data, and cleared to 0 to indicate non-zero data.

Bit	Bit Name	Initial Value	R/W	Description
1	V	Undefined	R/W	Overflow Flag Set to 1 when an arithmetic overflow occurs, and cleared to 0 at other times.
0	C	Undefined	R/W	Carry Flag Set to 1 when a carry occurs, and cleared to 0 otherwise. Used by: • Add instructions, to indicate a carry • Subtract instructions, to indicate a borrow • Shift and rotate instructions, to indicate a carry The carry flag is also used as a bit accumulator by bit manipulation instructions.

2.4.5 Multiply-Accumulate Register (MAC)

This 64-bit register stores the results of multiply-and-accumulate operations. It consists of two 32-bit registers denoted MACH and MACL. The lower 10 bits of MACH are valid; the upper bits are 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, clears the trace bit in EXR to 0, and sets the interrupt mask bits in CCR and EXR to 1. The other CCR bits and the general registers are not initialized. In particular, the stack pointer (ER7) is not initialized. The stack pointer should therefore be initialized by an MOV.L instruction executed immediately after a reset.

2.5 Data Formats

The H8S/2600 CPU can process 1-bit, 4-bit (BCD), 8-bit (byte), 16-bit (word), and 32-bit (longword) data. Bit-manipulation instructions operate on 1-bit data by accessing bit n (n = 0, 1, 2, …, 7) of byte operand data. The DAA and DAS decimal-adjust instructions treat byte data as two digits of 4-bit BCD data.

2.5.1 General Register Data Formats

Figure 2.9 shows the data formats in general registers.

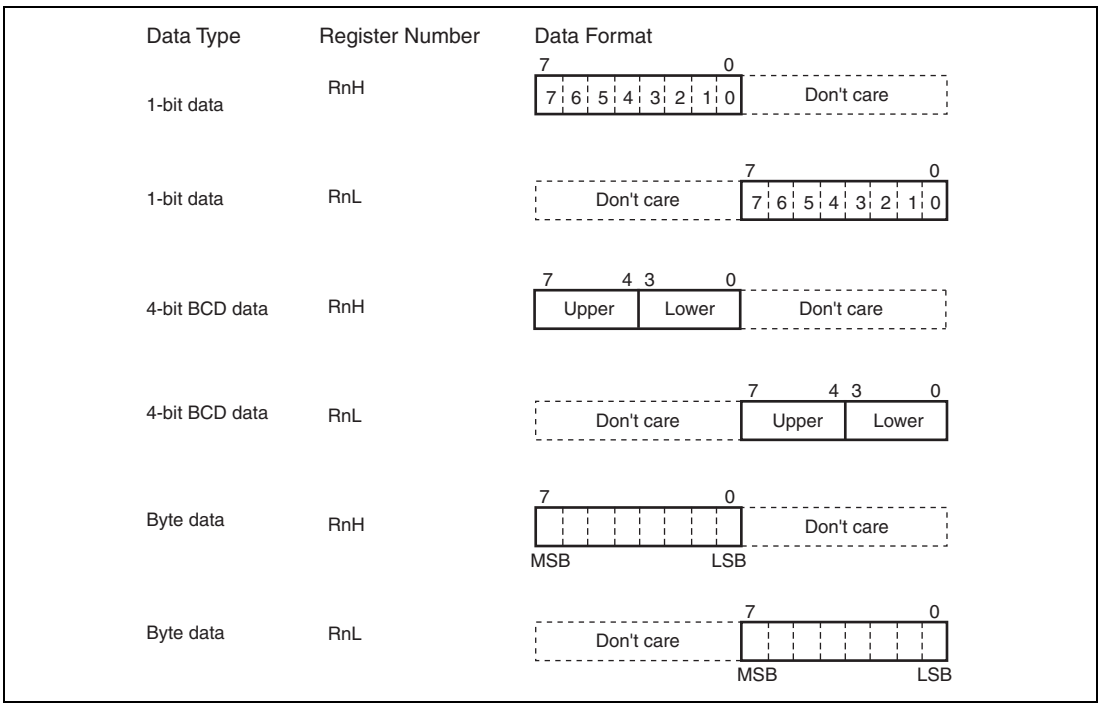


Figure 2.9 General Register Data Formats (1)

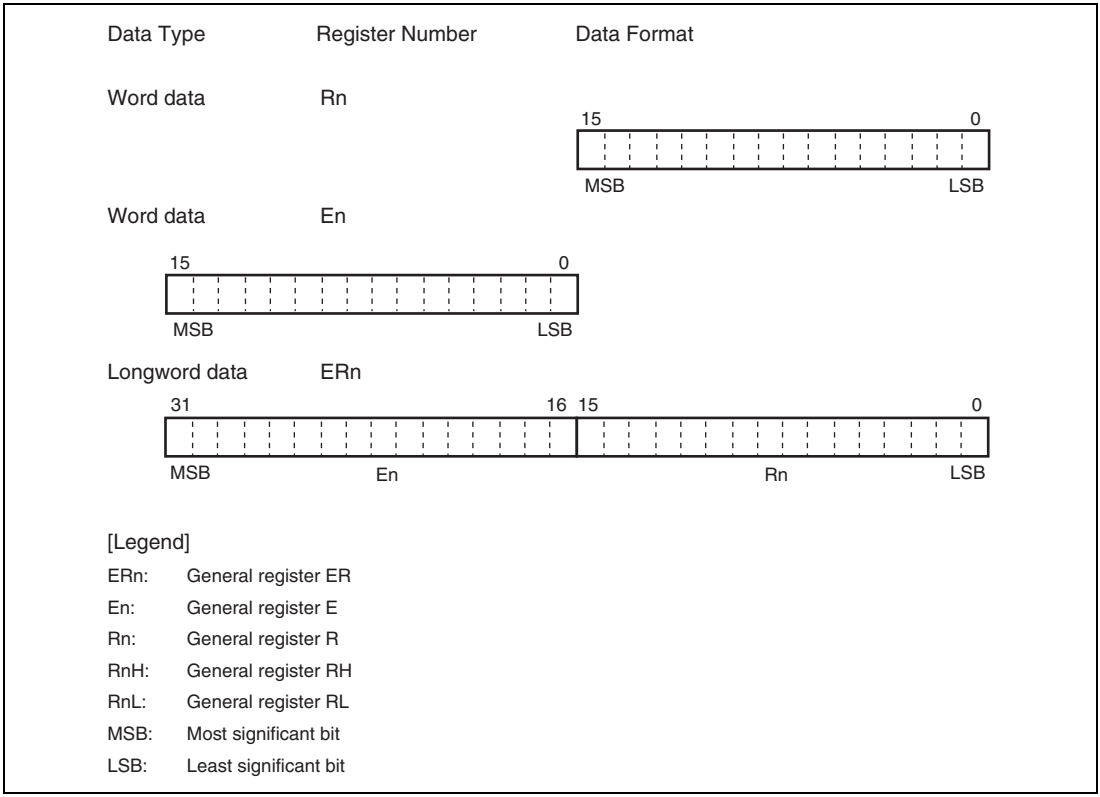


Figure 2.9 General Register Data Formats (2)

2.5.2 Memory Data Formats

Figure 2.10 shows the data formats in memory. The H8S/2600 CPU can access word data and longword data in memory, however word or longword data must begin at an even address. If an attempt is made to access word or longword data at an odd address, an address error does not occur, however the least significant bit of the address is regarded as 0, so access begins the preceding address. This also applies to instruction fetches.

When ER7 is used as an address register to access the stack, the operand size should be word or longword.

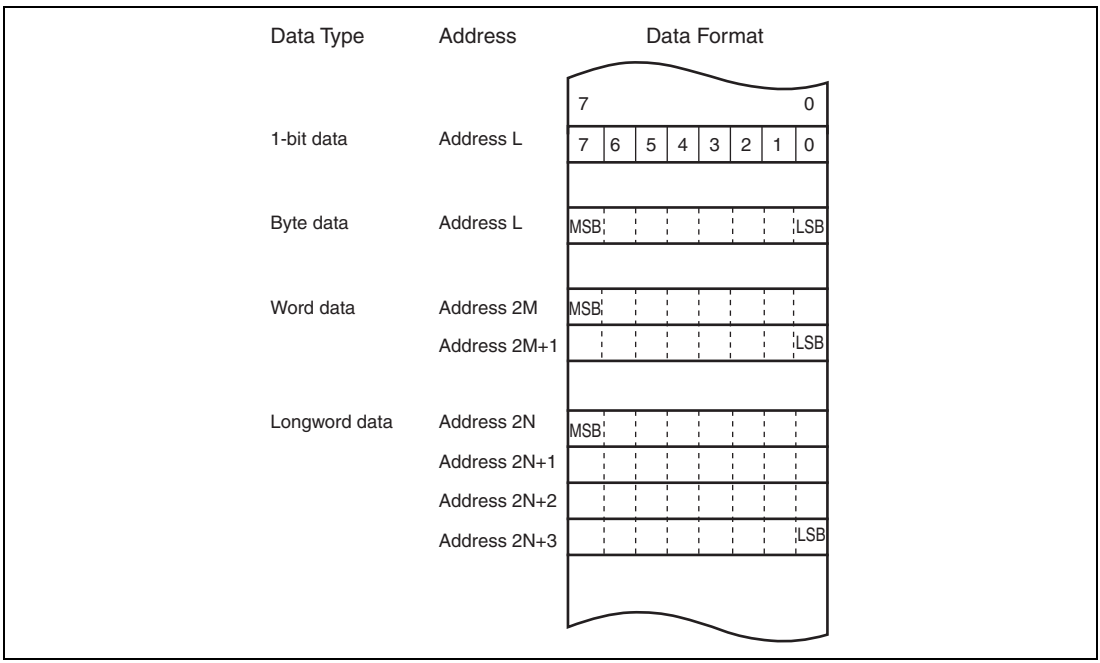


Figure 2.10 Memory Data Formats

2.6 Instruction Set

The H8S/2600 CPU has 69 instructions. The instructions are classified by function in table 2.1.

Table 2.1 Instruction Classification

Function	Instructions	Size	Types
Data transfer	MOV	B/W/L	5
	POP*[1], PUSH*[1]	W/L	
	LDM, STM	L	
	MOVFPE*[3], MOVTPE*[3]	B	
Arithmetic operation	ADD, SUB, CMP, NEG	B/W/L	23
	ADDX, SUBX, DAA, DAS	B	
	INC, DEC	B/W/L	
	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/L	4
Shift	SHAL, SHAR, SHLL, SHLR, ROTL, ROTR, ROTXL, ROTXR	B/W/L	8
Bit manipulation	BSET, BCLR, BNOT, BTST, BLD, BILD, BST, BIST, BAND, BIAND, BOR, BIOR, BXOR, BIXOR	B	14
Branch	Bcc*[2], JMP, BSR, JSR, RTS	—	5
System control	TRAPA, RTE, SLEEP, LDC, STC, ANDC, ORC, XORC, NOP	—	9
Block data transfer	EEPMOV	—	1
			Total: 69

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 Rn,@-SP. 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 instruction.

2.6.1 Table of Instructions Classified by Function

Tables 2.3 to 2.10 summarize the instructions in each functional category. The notation used in tables 2.3 to 2.10 is defined below.

Table 2.2 Operation Notation

Symbol	Description
Rd	General register (destination)*
Rs	General register (source)*
Rn	General register*
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)

Symbol	Description
:8/:16/:24/:32	8-, 16-, 24-, or 32-bit length

Note: * General registers include 8-bit registers (R0H to R7H, R0L to R7L), 16-bit registers (R0 to R7, E0 to E7), and 32-bit registers (ER0 to ER7).

Table 2.3 Data Transfer Instructions

Instruction	Size*	Function
MOV	B/W/L	(EAs) → Rd, Rs → (EAd) Moves data between two general registers or between a general register and memory, or moves immediate data to a general register.
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 to MOV.W @SP+, Rn. POP.L ERn is identical to MOV.L @SP+, ERn.
PUSH	W/L	Rn → @–SP Pushes a general register onto the stack. PUSH.W Rn is identical to MOV.W Rn, @–SP. PUSH.L ERn is identical to MOV.L ERn, @–SP.
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

Table 2.4 Arithmetic Operations Instructions (1)

Instruction	Size*	Function
ADD SUB	B/W/L	Rd ± Rs → Rd, Rd ± #IMM → Rd Performs addition or subtraction on data in two general registers, or on immediate data and data in a general register (immediate byte data cannot be subtracted from byte data in a general register. Use the SUBX or ADD instruction.)
ADDX SUBX	B	Rd ± Rs ± C → Rd, Rd ± #IMM ± C → Rd Performs addition or subtraction with carry on byte data in two general registers, or on immediate data and data in a general register.
INC DEC	B/W/L	Rd ± 1 → Rd, Rd ± 2 → Rd Increments or decrements a general register by 1 or 2. (Byte operands 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 register.
DAA DAS	B	Rd decimal adjust → Rd Decimal-adjusts an addition or subtraction result in a general register by referring to the CCR to produce 4-bit BCD data.
MULXU	B/W	Rd × Rs → Rd Performs unsigned multiplication on data in two general registers: either 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: either 8 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: either 16 bits ÷ 8 bits → 8-bit quotient and 8-bit remainder or 32 bits ÷ 16 bits → 16-bit quotient and 16-bit remainder.

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

Table 2.4 Arithmetic Operations Instructions (2)

Instruction	Size*[1]	Function
DIVXS	B/W	Rd ÷ Rs → Rd Performs signed division on data in two general registers: either 16 bits ÷ 8 bits → 8-bit quotient and 8-bit remainder or 32 bits ÷ 16 bits → 16-bit quotient and 16-bit remainder.
CMP	B/W/L	Rd – Rs, Rd – #IMM Compares data in a general register with data in another general register or with immediate data, and sets CCR bits according to the result.
NEG	B/W/L	0 – Rd → Rd Takes the two's complement (arithmetic complement) of data in a general register.
EXTU	W/L	Rd (zero extension) → Rd Extends the lower 8 bits of a 16-bit register to word size, or the lower 16 bits of a 32-bit register to longword size, by padding with zeros on the left.
EXTS	W/L	Rd (sign extension) → Rd Extends the lower 8 bits of a 16-bit register to word size, or the lower 16 bits of a 32-bit register to longword size, by extending the sign bit.
TAS*[2]	B	@ERd – 0, 1 → (<bit 7> of @ERd) Tests memory contents, and sets the most significant bit (bit 7) to 1.
MAC	—	(EAs) × (EAd) + MAC → MAC Performs signed multiplication on memory contents and adds the result to the multiply-accumulate register. The following operations can be 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-accumulate 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 instruction.

Table 2.5 Logic Operations Instructions

Instruction	Size*	Function
AND	B/W/L	Rd ∧ Rs → Rd, Rd ∧ #IMM → Rd Performs a logical AND operation on a general register and another general register or immediate data.
OR	B/W/L	Rd ∨ Rs → Rd, Rd ∨ #IMM → Rd Performs a logical OR operation on a general register and another general register or immediate data.
XOR	B/W/L	Rd ⊕ Rs → Rd, Rd ⊕ #IMM → Rd Performs a logical exclusive OR operation on a general register and another general register or immediate data.
NOT	B/W/L	~(Rd) → (Rd) Takes the one's complement (logical complement) of general register contents.

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

Table 2.6 Shift Instructions

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

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

Table 2.7 Bit Manipulation Instructions (1)

Instruction	Size*	Function
BSET	B	1 → (<bit-No.> of <EAd>) Sets a specified bit in a general register or memory operand to 1. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.
BCLR	B	0 → (<bit-No.> of <EAd>) Clears a specified bit in a general register or memory operand to 0. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.
BNOT	B	~(<bit-No.> of <EAd>) → (<bit-No.> of <EAd>) Inverts a specified bit in a general register or memory operand. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.
BTST	B	~(<bit-No.> of <EAd>) → Z Tests a specified bit in a general register or memory operand and sets or clears the Z flag accordingly. The bit number is specified by 3-bit 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 memory 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 general register or memory operand and stores the result in the carry flag. 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 memory 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 general register 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

Table 2.7 Bit Manipulation Instructions (2)

Instruction	Size*[1]	Function
BXOR	B	C ⊕ (<bit-No.> of <EAd>) → C XORs the carry flag with a specified bit in a general register or memory operand and stores the result in the carry flag.
BIXOR	B	C ⊕ [~(<bit-No.> of <EAd>)] → C XORs the carry flag with the inverse of a specified bit in a general register or memory operand and stores the result in the carry flag. The bit number is specified by 3-bit immediate data.
BLD	B	(<bit-No.> of <EAd>) → C Transfers a specified bit in a general register or memory operand to the carry flag.
BILD	B	~(<bit-No.> of <EAd>) → C Transfers the inverse of a specified bit in a general register or memory 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 register or memory operand.
BIST	B	~C → (<bit-No.> of <EAd>) Transfers the inverse of the carry flag value to a specified bit in a general register or memory operand. The bit number is specified by 3-bit immediate data.

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

Table 2.8 Branch Instructions

Instruction	Size	Function		
Bcc	—	Branches to a specified address if a specified condition is true. The branching conditions are listed below.		
		Mnemonic	**Description**	**Condition**
		BRA(BT)	Always (true)	Always
		BRN(BF)	Never (false)	Never
		BHI	High	$C \vee Z = 0$
		BLS	Low or same	$C \vee Z = 1$
		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		

Table 2.9 System Control Instructions

Instruction	Size*	Function
TRAPA	—	Starts trap-instruction exception handling.
RTE	—	Returns from an exception-handling routine.
SLEEP	—	Causes a transition to a power-down state.
LDC	B/W	(EAs) → CCR, (EAs) → EXR Moves general register or memory contents or immediate data to CCR or EXR. Although CCR and EXR are 8-bit registers, word-size transfers are performed between them and memory. The upper 8 bits are valid.
STC	B/W	CCR → (EAd), EXR → (EAd) Transfers CCR or EXR contents to a general register or memory. Although CCR and EXR are 8-bit registers, word-size transfers are performed between them and memory. The upper 8 bits are valid.
ANDC	B	CCR ∧ #IMM → CCR, EXR ∧ #IMM → EXR Logically ANDs the CCR or EXR contents with immediate data.
ORC	B	CCR ∨ #IMM → CCR, EXR ∨ #IMM → EXR Logically ORs the CCR or EXR contents with immediate data.
XORC	B	CCR ⊕ #IMM → CCR, EXR ⊕ #IMM → EXR Logically 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

Table 2.10 Block Data Transfer Instructions

Instruction	Size	Function
EEPMOV.B	—	if R4L ≠ 0 then Repeat @ER5+ → @ER6+ R4L-1 → R4L Until R4L = 0 else next;
EEPMOV.W	—	if R4 ≠ 0 then Repeat @ER5+ → @ER6+ R4-1 → R4 Until R4 = 0 else next;
		Transfers a data block. Starting from the address set in ER5, transfers data for the number of bytes set in R4L or R4 to the address location set in ER6. Execution of the next instruction begins as soon as the transfer is completed.

2.6.2 Basic Instruction Formats

The H8S/2600 CPU instructions consist of 2-byte (1-word) units. An instruction consists of an operation field (op field), a register field (r field), an effective address extension (EA field), and a condition field (cc).

Figure 2.11 shows examples of instruction formats.

- Operation Field

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

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

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

  Specifies the branching condition of Bcc instructions.

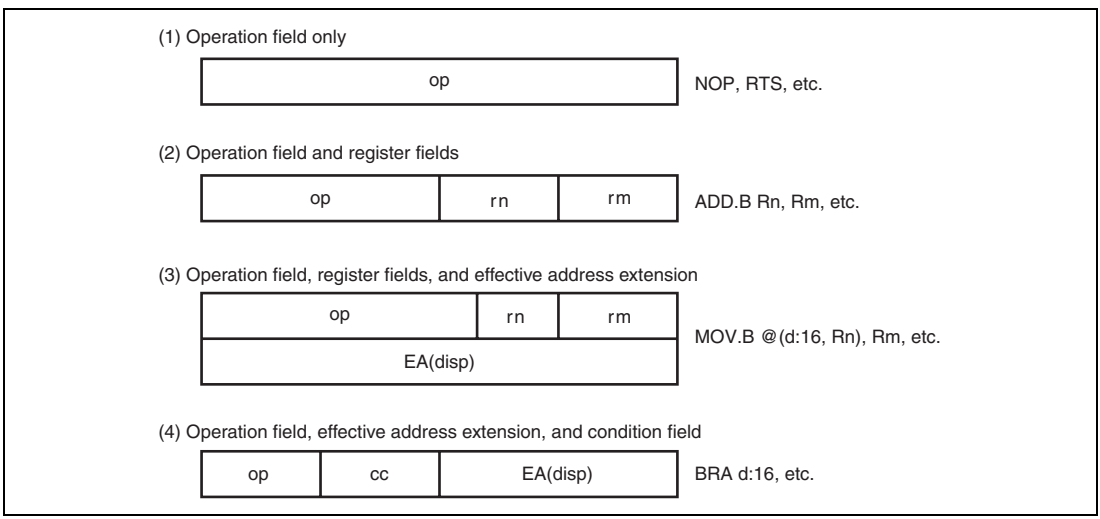


Figure 2.11 Instruction Formats (Examples)

2.7 Addressing Modes and Effective Address Calculation

The H8S/2600 CPU supports the eight addressing modes listed in table 2.11. Each instruction uses a subset of these addressing modes. Arithmetic and logic instructions can use the register direct and immediate modes. Data transfer instructions can use all addressing modes except program-counter relative and memory indirect. Bit manipulation instructions use register direct, register indirect, or the absolute addressing mode to specify an operand, and register direct (BSET, BCLR, BNOT, and BTST instructions) or immediate (3-bit) addressing mode to specify a bit number in the operand.

Table 2.11 Addressing Modes

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/@aa:32
6	Immediate	#xx:8/#xx:16/#xx:32
7	Program-counter relative	@(d:8,PC)/@(d:16,PC)
8	Memory indirect	@@aa:8

2.7.1 Register Direct—Rn

The register field of the instruction specifies an 8-, 16-, or 32-bit general register containing the operand. R0H to R7H and R0L to R7L can be specified as 8-bit registers. R0 to R7 and E0 to E7 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 contains the address of the operand on memory. If the address is a program instruction address, the lower 24 bits are valid and the upper 8 bits are all assumed to be 0 (H'00).

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

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

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

Register indirect with post-increment—@ERn+: The register field of the instruction code specifies an address register (ERn) which contains the address of a memory operand. After the operand is accessed, 1, 2, or 4 is added to the address register contents and the sum is stored in the address register. The value added 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 even.

Register indirect with pre-decrement—@-ERn: The value 1, 2, or 4 is subtracted from an address register (ERn) specified by the register field in the instruction code, and the result is the 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 even.

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

The instruction code contains the absolute address of a memory operand. The absolute address 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'FFFF). For a 16-bit absolute address the upper 16 bits are a sign extension. A 32-bit absolute address can access the entire address space.

A 24-bit absolute address (@aa:24) indicates the address of a program instruction. The upper 8 bits are all assumed to be 0 (H'00).

Table 2.12 Absolute Address Access Ranges

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

Note: Normal mode is not available in this LSI.

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

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

The ADDS, SUBS, INC, and DEC instructions contain immediate data implicitly. Some bit manipulation instructions contain 3-bit immediate data in the instruction code, specifying a bit number. The TRAPA instruction contains 2-bit immediate data in its instruction code, specifying a 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 contained in the instruction is sign-extended and added to the 24-bit PC contents to generate a branch address. Only the lower 24 bits of this branch address are valid; the upper 8 bits are all assumed to be 0 (H′00). The PC value to which the displacement is added is the address of the first byte of the next instruction, so the possible branching range is –126 to +128 bytes (–63 to +64 words) or –32766 to +32768 bytes (–16383 to +16384 words) from the branch instruction. The resulting value should be an even number.

2.7.8 Memory Indirect—@@aa:8

This mode can be used by the JMP and JSR instructions. The instruction code contains an 8-bit absolute address specifying a memory operand. This memory operand contains a branch address. The upper bits of the absolute address are all assumed to be 0, so the address range is 0 to 255 (H'0000 to H'00FF in normal mode, H'000000 to H'0000FF in advanced mode). In normal mode, the memory operand is a word operand and the branch address is 16 bits long. In advanced mode, the memory operand is a longword operand, the first byte of which is assumed to be 0 (H'00).

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

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

Note: Normal mode is not available in this LSI.

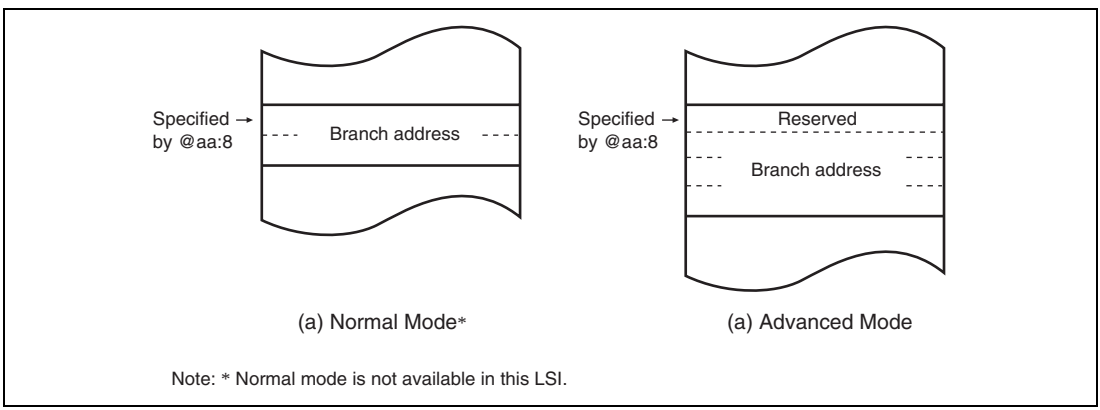


Figure 2.12 Branch Address Specification in Memory Indirect Mode

2.7.9 Effective Address Calculation

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

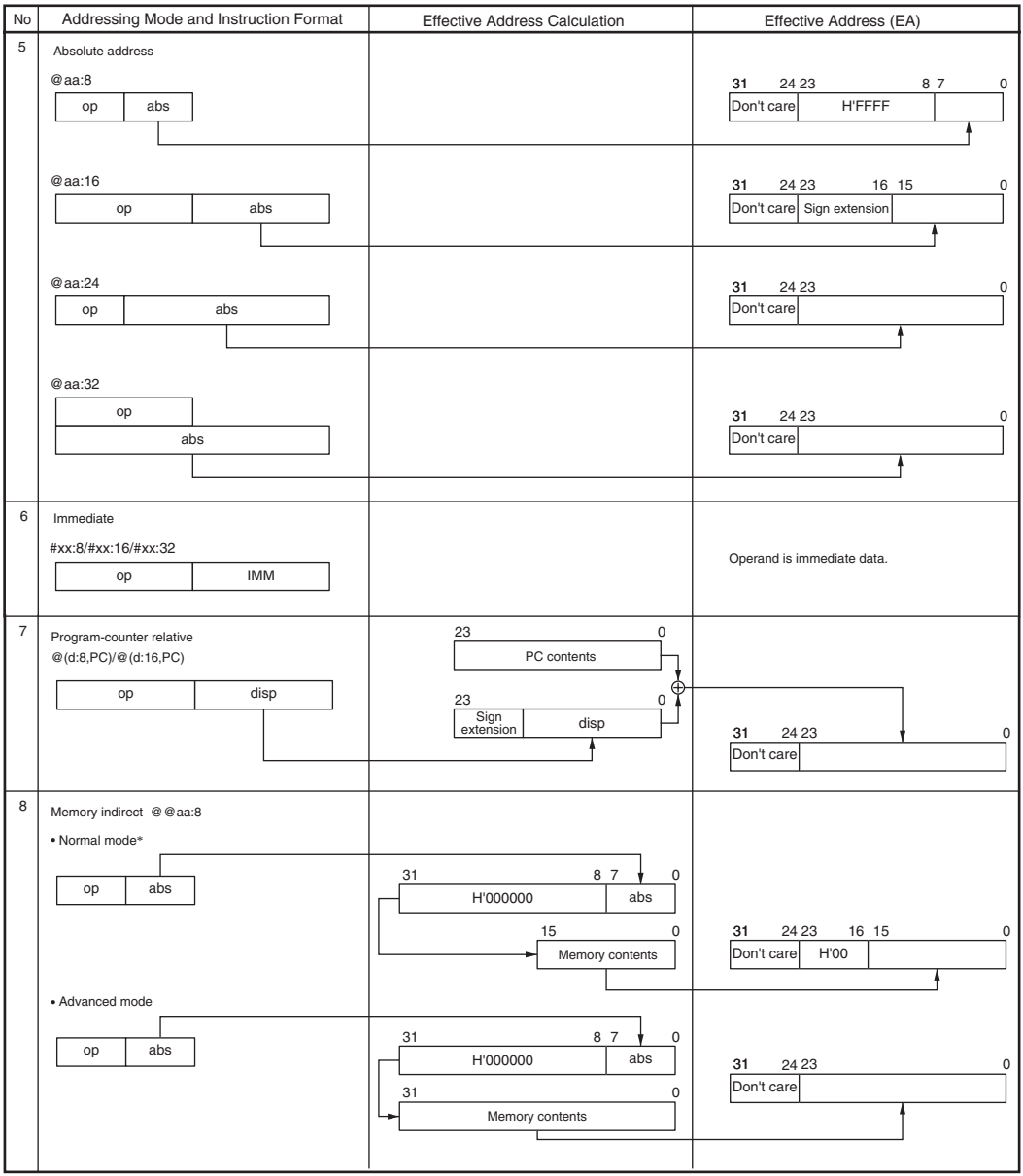
Note: Normal mode is not available in this LSI.

Table 2.13 Effective Address Calculation (1)

No	Addressing Mode and Instruction Format	Effective Address Calculation	Effective Address (EA)
1	Register direct(Rn) op rm rn		Operand is general register contents.
2	Register indirect(@ERn) op r	31 0 General register contents	31 24 23 0 Don't care
3	Register indirect with displacement @(d:16,ERn) or @(d:32,ERn) op r disp	31 0 General register contents 31 0 Sign extension disp	31 24 23 0 Don't care
4	Register indirect with post-increment or pre-decrement •Register indirect with post-increment @ERn+ op r	31 0 General register contents 1, 2, or 4	31 24 23 0 Don't care
	•Register indirect with pre-decrement @-ERn op r	31 0 General register contents 1, 2, or 4	31 24 23 0 Don't care

Operand Size	Offset
Byte	1
Word	2
Longword	4

Table 2.13 Effective Address Calculation (2)



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

2.8 Processing States

The H8S/2600 CPU has four main processing states: the reset state, exception handling state, program execution state and power-down state. Figure 2.13 indicates the state transitions.

- Reset State

  In this state, the CPU and all on-chip peripheral modules are initialized and not operating. When the $\overline{\text{RES}}$ input goes low, all current processing stops and the CPU enters the reset state. All interrupts are masked in the reset state. Reset exception handling starts when the $\overline{\text{RES}}$ signal changes from low to high. For details, refer to section 4, Exception Handling.

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

- Exception-Handling State

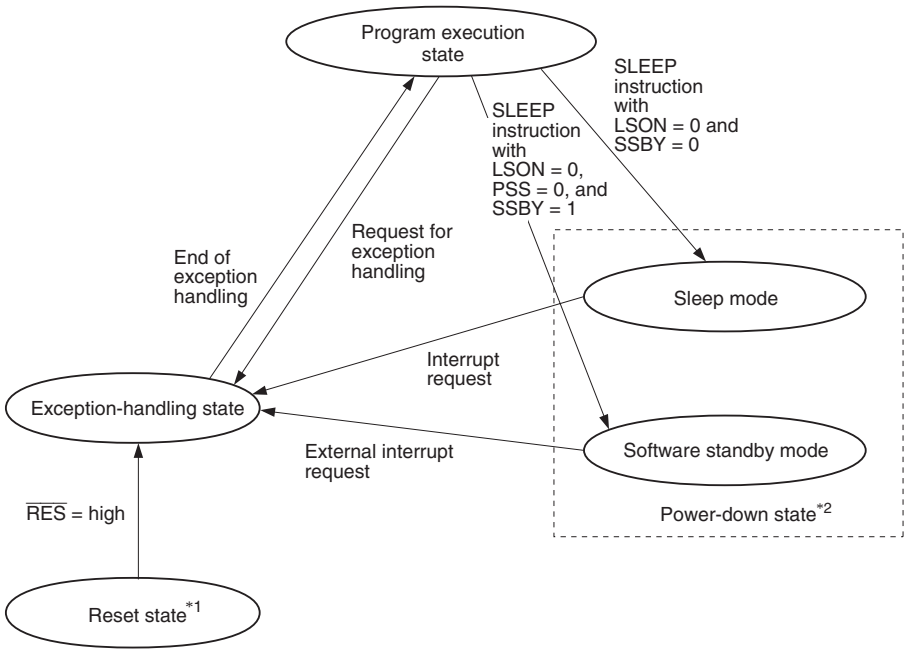
  The exception-handling state is a transient state that occurs when the CPU alters the normal processing flow due to an exception source, such as a reset, trace, interrupt, or trap instruction. The CPU fetches a start address (vector) from the exception vector table and branches to that address. For further 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 state occurs when a SLEEP instruction is executed or the CPU enters software standby mode. For further details, refer to section 26, Power-Down Modes.



Notes: 1. From any state, a transition to the reset state is made whenever the $\overline{RES}$ pin goes low. A transition can also be made to the reset state when the watchdog timer overflows.
2. The power-down state also includes watch mode. For details, refer to section 26, Power-Down Modes.

Figure 2.13 State Transitions

2.9 Usage Note

2.9.1 Notes on Using the Bit Operation Instruction

Instructions BSET, BCLR, BNOT, BST, and BIST read data in byte units, and write data in byte units after bit operation. Therefore, attention must be paid when these instructions are used for ports or registers including write-only bits.

Instruction BCLR can be used to clear the flag in the internal I/O register to 0. If it is obvious that the flag has been set to 1 by the interrupt processing routine, it is unnecessary to read the flag beforehand.

Section 3 MCU Operating Modes

3.1 Operating Mode Selection

This LSI supports three operating modes (modes 2, 4, and 6). The operating mode is determined by the setting of the mode pins (MD2 and MD1). Table 3.1 shows the MCU operating mode selection.

Table 3.1 MCU Operating Mode Selection

MCU Operating Mode	MD2	MD1	MD0*	CPU Operating Mode	Description	On-Chip ROM
2	0	1	0	Advanced	Single-chip mode	Enabled
4	1	0	0	—	Flash memory programming/erasing	—
6	1	1	0	Emulation	On-chip emulation mode	Enabled

Note: * MD0 is not available as a pin and is internally fixed to 0.

Modes 2 is single-chip mode.

Modes 0, 1, 3, 5 and 7 are not available in this LSI. Modes 4 and 6 are operating modes for a special purpose. Thus, mode pins should be set to enable mode 2 in the normal program execution state. Mode pin settings should not be changed during operation. After a reset is canceled, the mode pin inputs should be latched by reading MDCR.

Mode 4 is a boot mode for programming or erasing the flash memory. For details, see section 24, Flash Memory.

Mode 6 is an on-chip emulation mode. In this mode, this LSI is controlled by an on-chip emulator (E10A) via the JTAG, thus enabling on-chip emulation.

3.2 Register Descriptions

The following registers are related to the operating modes.

Table 3.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Data Bus Width
Mode control register	MDCR	R/W	—	H'FFC5	8
System control register	SYSCR	R/W	H'09	H'FFC4	8
Serial timer control register	STCR	R/W	H'00	H'FFC3	8
System control register 3	SYSCR3	R/W	H'60	H'FE7D	8

3.2.1 Mode Control Register (MDCR)

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

Bit	Bit Name	Initial Value	R/W	Description
7	EXPE	0	R/W	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 pins (MD2 and MD1) (the current operating mode). The MDS2 and MDS1 bits correspond to the MD2 and MD1 pins, respectively. These bits are read-only bits and cannot be written to. The input levels of the mode pins (MD2 and MD1) are latched into these bits when MDCR is read. These latches are canceled by a reset.
0	—	0	R	Reserved The initial value should not be changed.

Note: * The initial values are determined by the settings of the MD2 and MD1 pins.

3.2.2 System Control Register (SYSCR)

SYSCR monitors a reset source, selects the interrupt control mode and the detection edge for NMI, enables or disables access to the on-chip peripheral module registers, and enables or disables the on-chip RAM address space.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved The initial value should not be changed.
5	INTM1	0	R	Interrupt Control Select Mode 1 and 0
4	INTM0	0	R/W	These bits select the interrupt control mode of the interrupt controller. For details on the interrupt control modes, see section 5.6, Interrupt Control Modes and Interrupt Operation. 00: Interrupt control mode 0 01: Interrupt control mode 1 10: Setting prohibited 11: Setting prohibited
3	XRST	1	R	External Reset Indicates the reset source. A reset is caused by an external reset input, or when the watchdog timer overflows. 0: A reset is caused when the watchdog timer 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 input. 0: An interrupt is requested at the falling edge of NMI input 1: An interrupt is requested at the rising edge of NMI input

Bit	Bit Name	Initial Value	R/W	Description
1	KINWUE	0	R/W	Keyboard Control Register Access Enable When the RELOCATE bit is cleared to 0, this bit enables or disables CPU access for the keyboard matrix interrupt registers (KMIMRA and KMIMR), pull-up MOS control register (KMPCR), and registers (TCR_X/TCR_Y, TCSR_X/TCSR_Y, TICRR/TCORA_Y, TICRF/TCORB_Y, TCNT_X/TCNT_Y, TCORC, TCORA_X, TCORB_X, TCONRI, and TCONRS) of 8-bit timers (TMR_X and TMR_Y) 0: Enables CPU access for registers of TMR_X and TMR_Y in areas from H'(FF)FFF0 to H'(FF)FFF7 and from H'(FF)FFFC to H'(FF)FFFF 1: Enables CPU access for the keyboard matrix interrupt registers and input pull-up MOS control register in areas from H'(FF)FFF0 to H'(FF)FFF7 and from H'(FF)FFFC to H'(FF)FFFF When the RELOCATE bit is set to 1, this bit is disabled. For details, see section 3.2.4, System Control Register 3 (SYSCR3) and section 27, List of Registers.
0	RAME	1	R/W	RAM Enable Enables or disables on-chip RAM. 0: On-chip RAM is disabled 1: On-chip RAM is enabled

3.2.3 Serial Timer Control Register (STCR)

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

Bit	Bit Name	Initial Value	R/W	Description
7	IICX2	0	R/W	I^2C Transfer Rate Select 2 to 0
6	IICX1	0	R/W	These bits control the IIC operation. These bits select the transfer rate in master mode together with bits CKS2 to CKS0 in the I^2C bus mode register (ICMR). For details on the transfer rate, see table 18.4.
5	IICX0	0	R/W	
4	IICE	0	R/W	I^2C Master Enable When the RELOCATE bit is cleared to 0, enables or disables CPU access for IIC registers (ICCR, ICSR, ICDR/SARX, ICMR/SAR, and ICRES), PWMX registers (DADRAH/DACR, DADRAL, DADRBH/DACNTH, and DADRBL/DACNTL), and SCI registers (SMR, BRR, and SCMR). 0: SCI_1 registers are accessed in areas from H'(FF)FF88 to H'(FF)FF89 and from H'(FF)FF8E to H'(FF)FF8F. SCI_2 registers are accessed in areas from H'(FF)FFA0 to H'(FF)FFA1 and from H'(FF)FFA6 to H'(FF)FFA7. Access is prohibited in areas from H'(FF)FFD8 to H'(FF)FFD9 and from H'(FF)FFDE to H'(FF)FFDF. 1: IIC_1 registers are accessed in areas from H'(FF)FF88 to H'(FF)FF89 and from H'(FF)FF8E to H'(FF)FF8F. PWMX registers are accessed in areas from H'(FF)FFA0 to H'(FF)FFA1 and from H'(FF)FFA6 to H'(FF)FFA7. IIC_0 registers are accessed in areas from H'(FF)FFD8 to H'(FF)FFD9 and from H'(FF)FFDE to H'(FF)FFDF. ICRES is accessed in areas of H'(FF)FEE6 When the RELOCATE bit is set to 1, this bit is disabled. For details, see section 3.2.4, System Control Register 3 (SYSCR3) and section 27, List of Registers.

Bit	Bit Name	Initial Value	R/W	Description
3	FLSHE	0	R/W	Flash Memory Control Register Enable Enables or disables CPU access for flash memory registers (FCCS, FPCS, FECS, FKEY, FMATS, and FTDAR), power-down state control registers (SBYCR, LPWRCR, MSTPCRH, and MSTPCRL), and on-chip peripheral module control registers (PCSR). 0: When RELOCATE is 0, control registers of power-down state and peripheral modules are accessed in an area from H'(FF)FF80 to H'(FF)FF87. Area from H'(FF)FEA8 to H'(FF)FEAE is reserved. When RELOCATE is 1, control registers of power-down state and peripheral modules are accessed in an area from H'(FF)FF80 to H'(FF)FF87. Area from H'(FF)FEA8 to H'(FF)FEAE is reserved. 1: When RELOCATE is 0, control registers of flash memory are accessed in an area from H'(FF)FEA8 to H'(FF)FEAE. Area from H'(FF)FF80 to H'(FF)FF87 is reserved. When RELOCATE is 1, control registers of power-down state and peripheral modules are accessed in an area from H'(FF)FF80 to H'(FF)FF87. Control registers of flash memory are accessed in an area from H'(FF)FEA8 to H'(FF)FEAE.
2	IICS	0	R/(W)	I^2C Extra Buffer Select Specifies bits 7 to 4 of port A as output buffers similar to SLC and SDA. These pins are used to implement an I^2C interface only by software. 0: PA7 to PA4 are normal input/output pins. 1: PA7 to PA4 are input/output pins enabling bus driving.
1	ICKS1	0	R/W	Internal Clock Source Select 1 and 0
0	ICKS0	0	R/W	These bits select a clock to be input to the timer counter (TCNT) and a count condition together with bits CKS2 to CKS0 in the timer control register (TCR). For details, see section 13.3.4, Timer Control Register (TCR).

3.2.4 System Control Register 3 (SYSCR3)

SYSCR3 selects the register map and interrupt vector.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R/W	Reserved The initial value should not be changed.
6	EIVS*	1	R/W	Extended interrupt Vector Select* Selects compatible mode or extended mode for the interrupt vector table. 0: H8S/2140B Group compatible vector mode 1: Extended vector mode For details, see section 5, Interrupt Controller.
5	RELOCATE	1	R/W	Register Address Map Select Selects compatible mode or extended mode for the register map. When extended mode is selected for the register map, CPU access for registers can be controlled without using the KINWUE bit in SYSCR or the IICE bit in STCR to switch the registers to be accessed. 0: H8S/2140B Group compatible register map mode 1: Extended register map mode For details, see section 27, List of Registers.
4 to 0	—	All 0	R/W	Reserved The initial value should not be changed.

Note: * Switch the modes when an interrupt occurrence is disabled.

3.3 Operating Mode Descriptions

3.3.1 Mode 2

The CPU can access a 16-Mbyte address space in either advanced mode or single-chip mode. The on-chip ROM is enabled.

3.4 Address Map

Figures 3.1 shows the address map in each operating mode.

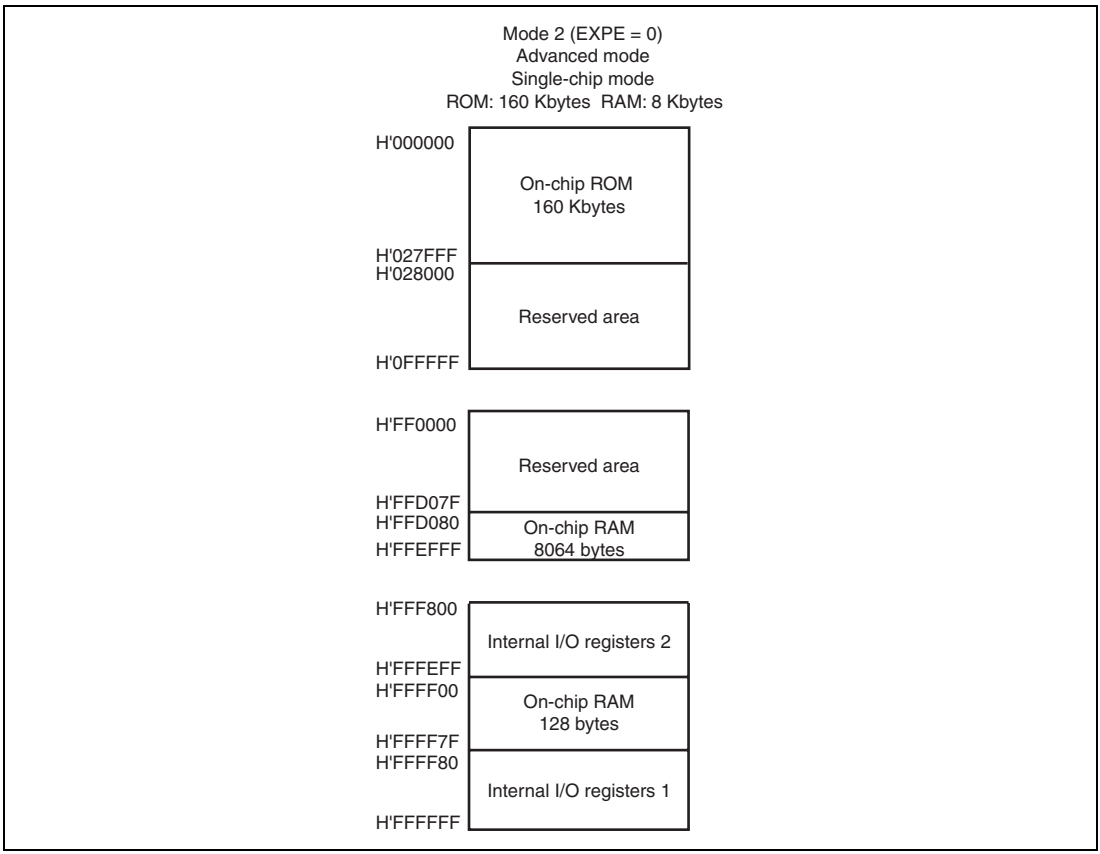


Figure 3.1 Address Map

Section 4 Exception Handling

4.1 Exception Handling Types and Priority

As table 4.1 indicates, exception handling may be caused by a reset, illegal instruction, interrupt, direct transition, or trap instruction. Exception handling is prioritized as shown in table 4.1. If two or more exceptions occur simultaneously, they are accepted and processed in order of priority.

Table 4.1 Exception Types and Priority

Priority	Exception Type	Start of Exception Handling
High ▲	Reset	Starts immediately after a low-to-high transition of the $\overline{\text{RES}}$ pin, or when the watchdog timer overflows.
	Illegal instruction	Exception handling starts when an undefined code is executed.
	Interrupt	Starts when execution of the current instruction or exception handling ends, if an interrupt request has been issued. Interrupt detection is not performed on completion of ANDC, ORC, XORC, or LDC instruction execution, or on completion of reset exception handling.
	Direct transition	Starts when a direct transition occurs as the result of SLEEP instruction execution.
	Trap instruction	Started by execution of a trap (TRAPA) instruction. Trap instruction exception handling requests are accepted at all times in the program execution state.
Low		

4.2 Exception Sources and Exception Vector Table

Different vector addresses are assigned to exception sources. Table 4.2 and table 4.3 list the exception sources and their vector addresses. The EIVS bit in the system control register 3 (SYSCR3) allows the selection of the H8S/2140B Group compatible vector mode or extended vector mode.

Table 4.2 Exception Handling Vector Table (H8S/2140B Group Compatible Vector Mode)

Exception Source		Vector Number	Vector Addresses Advanced Mode
Reset		0	H'000000 to H'000003
Reserved for system use		1 ⋮ 3	H'000004 to H'000007 ⋮ H'00000C to H'00000F
Illegal instruction		4	H'000010 to H'000013
Reserved for system use		5	H'000014 to H'000017
Direct transition		6	H'000018 to H'00001B
External interrupt (NMI)		7	H'00001C to H'00001F
Trap instruction (four sources)		8	H'000020 to H'000023
		9	H'000024 to H'000027
		10	H'000028 to H'00002B
		11	H'00002C to H'00002F
Reserved for system use		12 ⋮ 15	H'000030 to H'000033 ⋮ H'00003C to H'00003F
External interrupt	IRQ0	16	H'000040 to H'000043
	IRQ1	17	H'000044 to H'000047
	IRQ2	18	H'000048 to H'00004B
	IRQ3	19	H'00004C to H'00004F
	IRQ4	20	H'000050 to H'000053
	IRQ5	21	H'000054 to H'000057
	IRQ6, KIN7 to KIN0	22	H'000058 to H'00005B
	IRQ7, KIN15 to KIN8	23	H'00005C to H'00005F

Exception Source		Vector Number	Vector Addresses Advanced Mode
Internal interrupt*		24 | 29	H'000060 to H'000063 | H'000074 to H'000077
Reserved for system use		30	H'000078 to H'00007B
Reserved for system use		31	H'00007C to H'00007F
Reserved for system use		32	H'000080 to H'000083
External interrupt	WUE15 to WUE8	33	H'000084 to H'000087
Internal interrupt*		34 | 55	H'000088 to H'00008B | H'0000DC to H'0000DF
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 | 127	H'000100 to H'000103 | H'0001FC to H'0001FF

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

Table 4.3 Exception Handling Vector Table (Extended Vector Mode)

Exception Source		Vector Number	Vector Addresses Advanced Mode
Reset		0	H'000000 to H'000003
Reserved for system use		1 ⋮ 3	H'000004 to H'000007 ⋮ H'00000C to H'00000F
Illegal instruction		4	H'000010 to H'000013
Reserved for system use		5	H'000014 to H'000017
Direct transition		6	H'000018 to H'00001B
External interrupt (NMI)		7	H'00001C to H'00001F
Trap instruction (four sources)		8	H'000020 to H'000023
		9	H'000024 to H'000027
		10	H'000028 to H'00002B
		11	H'00002C to H'00002F
Reserved for system use		12 ⋮ 15	H'000030 to H'000033 ⋮ H'00003C to H'00003F
External interrupt	IRQ0	16	H'000040 to H'000043
	IRQ1	17	H'000044 to H'000047
	IRQ2	18	H'000048 to H'00004B
	IRQ3	19	H'00004C to H'00004F
	IRQ4	20	H'000050 to H'000053
	IRQ5	21	H'000054 to H'000057
	IRQ6	22	H'000058 to H'00005B
	IRQ7	23	H'00005C to H'00005F
Internal interrupt*		24 ⋮ 29	H'000060 to H'000063 ⋮ H'000074 to H'000077
External interrupt	KIN7 to KIN0	30	H'000078 to H'00007B
External interrupt	KIN15 to KIN8	31	H'00007C to H'00007F
Reserved for system use		32	H'000080 to H'000083
External interrupt	WUE15 to WUE8	33	H'000084 to H'000087

Exception Source		Vector Number	Vector Addresses Normal Mode
Internal interrupt*		34 ⎮ 55	H'000088 to H'00008B ⎮ H'0000DC to H'0000DF
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 ⎮ 127	H'000100 to H'000103 ⎮ H'0001FC to H'0001FF

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

4.3 Reset

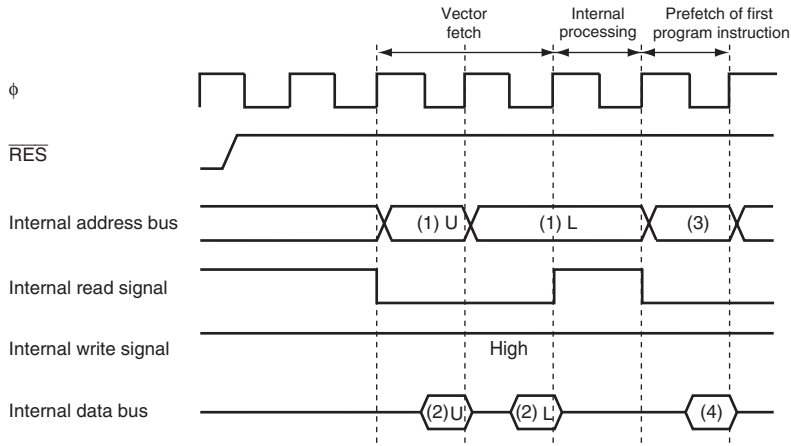
A reset has the highest exception priority. When the $\overline{\text{RES}}$ pin goes low, all processing halts and this LSI enters the reset state. To ensure that this LSI is reset, hold the $\overline{\text{RES}}$ pin low for at least 20 ms at power-on. To reset the chip during operation, hold the $\overline{\text{RES}}$ pin low for at least 20 states. A reset initializes the internal state of the CPU and the registers of on-chip peripheral modules. The chip can also be reset by overflow of the watchdog timer. For details, see section 14, Watchdog Timer (WDT).

4.3.1 Reset Exception Handling

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

1. The internal state of the CPU and the registers of the on-chip peripheral modules are initialized 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 then program 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 (Mode 2)

4.3.2 Interrupts Immediately after Reset

If an interrupt is accepted immediately after a reset and before the stack pointer (SP) is initialized, the PC and CCR will not be saved correctly, leading to a program crash. To prevent this, all interrupt requests, including NMI, are disabled immediately after a reset. Since the first instruction of a program is always executed immediately after a reset, make sure that this instruction initializes the SP (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 (MSTPCRH, MSTPCRL, MSTPCRA, MSTPCRB) are initialized, and all modules except the DTC operate in module stop mode. Therefore, the registers of on-chip peripheral modules cannot be read from or written to. To read from and write to these registers, clear module stop mode. For details on module stop mode, see section 26, Power-Down Modes.

4.4 Interrupt Exception Handling

Interrupts are controlled by the interrupt controller. The sources to start interrupt exception handling are external interrupt sources (NMI, IRQ15 to IRQ0, KIN15 to KIN0, and WUE15 to WUE8) and internal interrupt sources from the on-chip peripheral modules. NMI is an interrupt with the highest priority. For details, see section 5, Interrupt Controller.

Interrupt exception handling is conducted as follows:

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

4.5 Trap Instruction Exception Handling

Trap instruction exception handling starts when a TRAPA instruction is executed. Trap instruction 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 in the stack.
2. A vector address corresponding to the interrupt source is generated, the start address is loaded 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 corresponding to a vector number from 0 to 3, as specified in the instruction code.

Table 4.4 shows the status of CCR after execution of trap instruction exception handling.

Table 4.4 Status of CCR after Trap Instruction Exception Handling

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

4.6 Exception Handling by Illegal Instruction

The exception handling by the illegal instruction starts when an undefined code is executed. The exception handling by the illegal instruction is always executable in the program execution state.

The exception handling operates as follows:

1. The contents of the PC and CCR are saved in the stack.
2. The interrupt mask bit is updated.
3. An exception handling vector table address corresponding to the occurred exception is generated, the start address of the exception service routine is loaded from the vector table to the PC, and program execution starts from that address.

Table 4.5 shows the state of CCR after execution of illegal instruction exception handling.

Table 4.5 Status of CCR after Illegal Instruction Exception Handling

	CCR	
Interrupt Control Mode	**I**	**UI**
0	Set to 1	Retains the previous value
1	Set to 1	Set to 1

Illegal instruction code is not detected for fields that do not affect the definition of the instruction, such as an effective address extension (EA) and register fields. In addition, the instruction code of instructions consisting of multiple words are detected individually and are not detected as combinations of instruction codes.

Do not execute instruction codes that are not defined. The contents of general registers are not guaranteed after the execution of an undefined instruction code or exception handling by the illegal instruction. The value of the stack pointer at the time of exception handling by the illegal instruction and the saved contents of the PC are also not guaranteed.

4.7 Stack Status after Exception Handling

Figure 4.2 shows the stack after completion of trap instruction exception handling and interrupt exception handling.

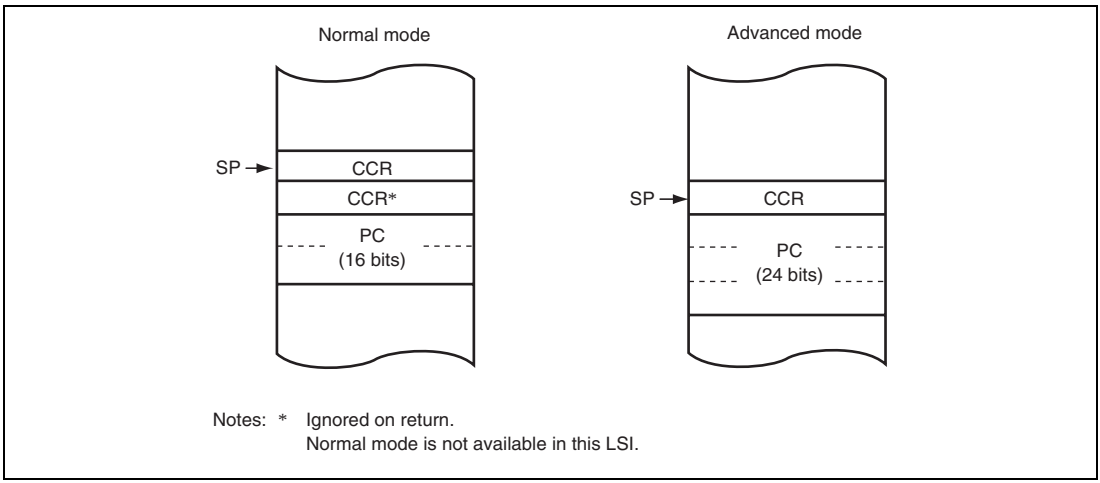


Figure 4.2 Stack Status after Exception Handling

4.8 Usage Note

When accessing word data or longword data, this LSI assumes that the lowest address bit is 0. The stack should always be accessed in words or longwords, and the value of the stack pointer (SP: ER7) should always be kept even.

Use the following instructions to save registers:

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

Use the following instructions to restore registers:

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

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

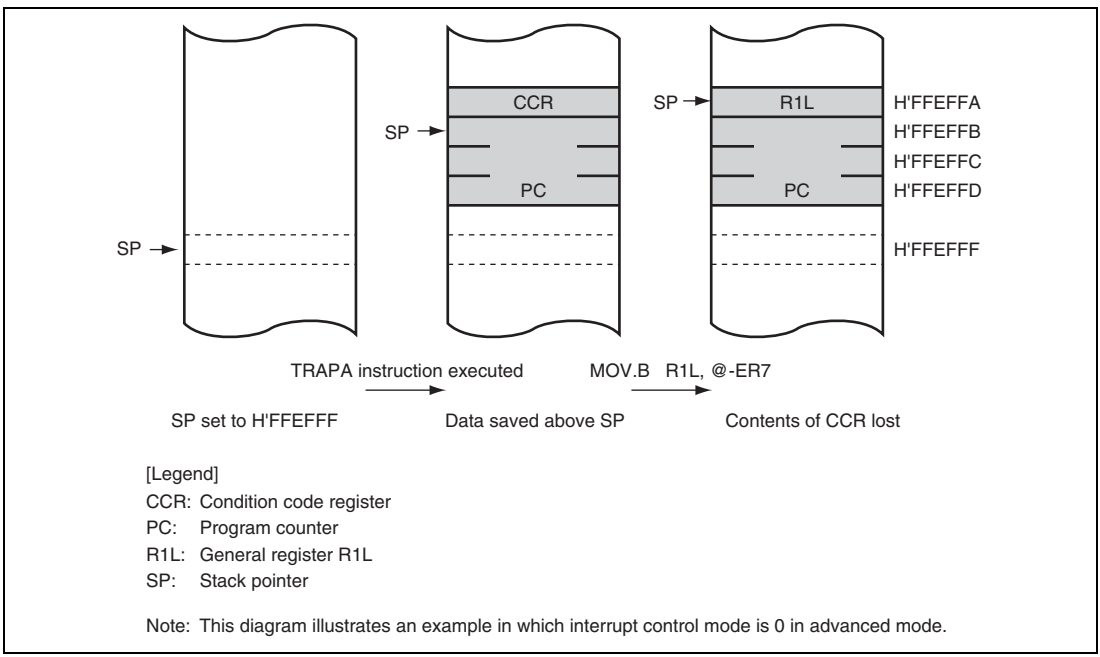


Figure 4.3 Operation when SP Value Is Odd

Section 5 Interrupt Controller

5.1 Features

- Two interrupt control modes

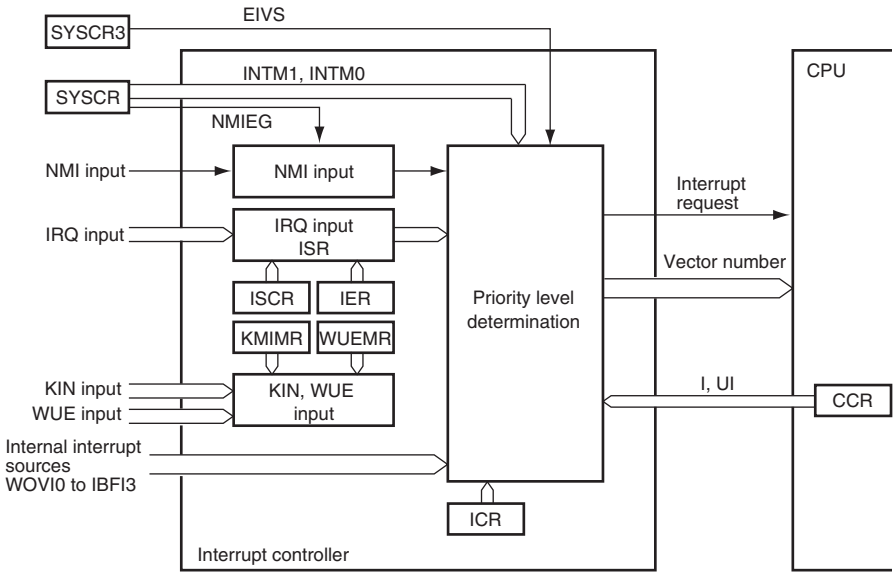
  Two interrupt control modes can be set by means of the INTM1 and INTM0 bits in the system control register (SYSCR).
- Priorities settable with ICR

  An interrupt control register (ICR) is provided for setting in each module interrupt priority levels for all interrupt requests excluding NMI and address breaks.
- Three-level interrupt mask control

  By means of the interrupt control mode, I and UI bits in CCR and ICR, 3-level interrupt mask control is performed.
- Forty-one external interrupt pins

  NMI is the highest-priority interrupt, and is accepted at all times. Rising edge or falling edge detection can be selected for NMI. Falling-edge, rising-edge, or both-edge detection, or level sensing, can be independently selected for $\overline{\text{IRQ15}}$ to $\overline{\text{IRQ0}}$. When the EIVS bit in the system control register 3 (SYSCR3) is cleared to 0, the IRQ6 interrupt is generated by $\overline{\text{IRQ6}}$ or $\overline{\text{KIN7}}$ to $\overline{\text{KIN0}}$. The IRQ7 interrupt is generated by $\overline{\text{IRQ7}}$ or $\overline{\text{KIN15}}$ to $\overline{\text{KIN8}}$. When the EIVS bit in the system control register 3 (SYSCR3) is set to 1, interrupts are requested on the falling edge of $\overline{\text{KIN15}}$ to $\overline{\text{KIN0}}$. For $\overline{\text{WUE15}}$ to $\overline{\text{WUE8}}$, either rising-edge or falling-edge detection can be selected individually for each pin regardless of the EIVS bit setting.
- Two interrupt vector addresses are selectable

  H8S/2140B Group compatible interrupt vector addresses or extended interrupt vector addresses are selected depending on the EIVS bit in system control register 3 (SYSCR3). In extended mode, independent vector addresses are assigned for the interrupt vector addresses of KIN7 to KIN0 or KIN15 to KIN8 interrupts.
- General ports for $\overline{\text{IRQ15}}$ to $\overline{\text{IRQ6}}$ input are selectable



[Legend]
ICR: Interrupt control register
ISCR: IRQ sense control register
IER: IRQ enable register
ISR: IRQ status register
KMIMR: Keyboard matrix interrupt mask register
WUEMR: Wake-up event interrupt mask register
SYSCR: System control register
SYSCR3: System control register 3

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

Pin Name	I/O	Function
NMI	Input	Nonmaskable external interrupt pin Rising edge or falling edge can be selected
$\overline{\text{IRQ15}}$ to $\overline{\text{IRQ0}}$, $\overline{\text{ExIRQ15}}$ to $\overline{\text{ExIRQ6}}$	Input	Maskable external interrupt pins Rising-edge, falling-edge, or both-edge detection, or level-sensing, can be selected individually for each pin. To which pin the IRQ15 to IRQ6 interrupt is input can be selected from the $\overline{\text{IRQm}}$ and $\overline{\text{ExIRQm}}$ pins. (n = 15 to 6)
$\overline{\text{KIN15}}$ to $\overline{\text{KIN0}}$	Input	Maskable external interrupt pins When EIVS = 0, falling-edge or level-sensing can be selected. When EIVS = 1, an interrupt is requested at the falling edge.
$\overline{\text{WUE15}}$ to $\overline{\text{WUE8}}$	Input	Maskable external interrupt pins Either rising edge or falling edge detection can be selected for each pin.

5.3 Register Descriptions

The interrupt controller has the following registers. For details on the system control register (SYSCR), see section 3.2.2, System Control Register (SYSCR). For details on system control register 3 (SYSCR3), see section 3.2.4, System Control Register 3 (SYSCR3).

Table 5.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Data Bus Width
Interrupt control registers A	ICRA	R/W	H'00	H'FEE8	8
Interrupt control registers B	ICRB	R/W	H'00	H'FEE9	8
Interrupt control registers C	ICRC	R/W	H'00	H'FEEA	8
Interrupt control registers D	ICRD	R/W	H'00	H'FE87	8
Address break control register	ABRKCR	R/W	—	H'FEF4	8
Break address registers A	BARA	R/W	H'00	H'FEF5	8
Break address registers B	BARB	R/W	H'00	H'FEF6	8
Break address registers C	BARC	R/W	H'00	H'FEF7	8
IRQ sense control register 16H	ISCR16H	R/W	H'00	H'FEFA	8
IRQ sense control register 16L	ISCR16L	R/W	H'00	H'FEFB	8
IRQ sense control register H	ISCRH	R/W	H'00	H'FEEC	8
IRQ sense control register L	ISCRL	R/W	H'00	H'FEED	8
IRQ enable register 16	IER16	R/W	H'00	H'FEF8	8
IRQ enable register	IER	R/W	H'00	H'FFC2	8
IRQ status register 16	ISR16	R/W	H'00	H'FEF9	8
IRQ status register	ISR	R/W	H'00	H'FEEB	8
Keyboard matrix interrupt mask register A	KMIMRA	R/W	H'FF	H'FFF3 H'FE83*[1]	8
Keyboard matrix interrupt mask register	KMIMR	R/W	H'BF H'FF*[2]	H'FFF1 H'FE81*[1]	8
Wake-up event interrupt mask registers	WUEMR	R/W	H'00	H'FE45	8
IRQ sense port select register 16	ISSR16	R/W	H'00	H'FEFC	8
IRQ sense port select register	ISSR	R/W	H'00	H'FEFD	8

Register Name	Abbreviation	R/W	Initial Value	Address	Data Bus Width
Wake-up sense control register	WUESCR	R/W	H'00	H'FE84	8
Wake-up input interrupt status register	WUESR	R/W	H'00	H'FE85	8
Wake-up enable register	WER	R/W	H'00	H'FE86	8

Note: 1. Address in the upper cell: when RELOCATE = 0, address in the lower cell: when RELOCATE = 1
2. Address in the upper cell: when EIVS = 0, address in the lower cell: when EIVS = 1

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 tables 5.2 and 5.3.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	ICRn7 to ICRn0	All 0	R/W	Interrupt Control Level 0: Corresponding interrupt source is interrupt control level 0 (no priority) 1: Corresponding interrupt source is interrupt control level 1 (priority)

Note: n: A to D

Table 5.3 Correspondence between Interrupt Source and ICR (H8S/2140B Group Compatible Vector Mode: EIVS = 0)

		Register			
Bit	**Bit Name**	**ICRA**	**ICRB**	**ICRC**	**ICRD**
7	ICRn7	IRQ0	A/D converter	SCIF	IRQ8 to IRQ11
6	ICRn6	IRQ1	TCM_0, TCM_1, TCM_2, TCM_3	SCI_1	IRQ12 to IRQ15
5	ICRn5	IRQ2, IRQ3	TDP_0, TDP_1, TDP_2	SCI_2	—
4	ICRn4	IRQ4, IRQ5	CIR	IIC_0	WUE8 to WUE15
3	ICRn3	IRQ6, IRQ7	TMR_0	IIC_1, IIC_2	TPU_0
2	ICRn2	—	TMR_1	FSI	TPU_1
1	ICRn1	WDT_0	TMR_X, TMR_Y	LPC, FSI	TPU_2
0	ICRn0	WDT_1	PS2	—	—

Note: n: A to D
—: Reserved. The initial value should not be changed.

Table 5.4 Correspondence between Interrupt Source and ICR (Extended Vector Mode: EIVS = 1)

		Register			
Bit	**Bit Name**	**ICRA**	**ICRB**	**ICRC**	**ICRD**
7	ICRn7	IRQ0	A/D converter	SCIF	IRQ8 to IRQ11
6	ICRn6	IRQ1	TCM_0, TCM_1, TCM_2, TCM_3	SCI_1	IRQ12 to IRQ15
5	ICRn5	IRQ2, IRQ3	TDP_0, TDP_1, TDP_2	SCI_2	KIN0 to KIN15
4	ICRn4	IRQ4, IRQ5	CIR	IIC_0	WUE8 to WUE15
3	ICRn3	IRQ6, IRQ7	TMR_0	IIC_1, IIC_2	TPU_0
2	ICRn2	—	TMR_1	FSI	TPU_1
1	ICRn1	WDT_0	TMR_X, TMR_Y	LPC, FSI	TPU_2
0	ICRn0	WDT_1	PS2	—	—

Note: n: A to D
—: Reserved. The initial value should not be changed.

5.3.2 Address Break Control Register (ABRKCR)

ABRKCR controls the address breaks. When both the CMF flag and BIE bit are set to 1, an 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 address specified by BARA to BARC is prefetched. [Clearing condition] When an exception handling is executed for an address break interrupt. [Setting condition] When an address specified by BARA to BARC is prefetched while the BIE bit is set to 1.
6 to 1	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
0	BIE	0	R/W	Break Interrupt Enable Enables or disables address break. 0: Disabled 1: Enabled

5.3.3 Break Address Registers A to C (BARA to BARC)

The BAR registers specify an address that is to be a break address. An address in which the first byte of an instruction exists should be set as a break address.

- BARA

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	A23 to A16	All 0	R/W	Addresses 23 to 16 The A23 to A16 bits are compared with A23 to A16 in the internal address bus.

- 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 in the internal address bus.

- BARC

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

5.3.4 IRQ Sense Control Registers (ISCR16H, ISCR16L, ISCRH, ISCRL)

The ISCR registers select the source that generates an interrupt request at pins $\overline{\text{IRQ15}}$ to $\overline{\text{IRQ0}}$ or pins $\overline{\text{ExIRQ15}}$ to $\overline{\text{ExIRQ6}}$.

- ISCR16H

Bit	Bit Name	Initial Value	R/W	Description
7	IRQ15SCB	0	R/W	IRQn Sense Control B
6	IRQ15SCA	0	R/W	IRQn Sense Control A BA 00: Interrupt request generated at low level of $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ input 01: Interrupt request generated at falling edge of $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ input 10: Interrupt request generated at rising edge of $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ input 11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ input (n = 15 to 12) Note: The $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ pin is selected by IRQ sense port select register 16 (ISSR16).
5	IRQ14SCB	0	R/W	
4	IRQ14SCA	0	R/W	
3	IRQ13SCB	0	R/W	
2	IRQ13SCA	0	R/W	
1	IRQ12SCB	0	R/W	
0	IRQ12SCA	0	R/W	

- ISCR16L

Bit	Bit Name	Initial Value	R/W	Description
7	IRQ11SCB	0	R/W	IRQn Sense Control B
6	IRQ11SCA	0	R/W	IRQn Sense Control A
5	IRQ10SCB	0	R/W	BA
4	IRQ10SCA	0	R/W	00: Interrupt request generated at low level of $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ input
3	IRQ9SCB	0	R/W	01: Interrupt request generated at falling edge of $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ input
2	IRQ9SCA	0	R/W	10: Interrupt request generated at rising edge of $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ input
1	IRQ8SCB	0	R/W	11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ input
0	IRQ8SCA	0	R/W	(n = 11 to 8)
				Note: The $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ pin is selected by IRQ sense port select register 16 (ISSR16).

- ISCRH

Bit	Bit Name	Initial Value	R/W	Description
7	IRQ7SCB	0	R/W	IRQn Sense Control B
6	IRQ7SCA	0	R/W	IRQn Sense Control A
5	IRQ6SCB	0	R/W	BA
4	IRQ6SCA	0	R/W	00: Interrupt request generated at low level of $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ input
3	IRQ5SCB	0	R/W	01: Interrupt request generated at falling edge of $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ input
2	IRQ5SCA	0	R/W	10: Interrupt request generated at rising edge of $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ input
1	IRQ4SCB	0	R/W	11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ input
0	IRQ4SCA	0	R/W	(n = 7 to 4)
				Note: The $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ pin is selected by the IRQ sense port select register (ISSR). The ExIRQ5 and ExIRQ4 pins are not supported.

- ISCRL

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

5.3.5 IRQ Enable Registers (IER16, IER)

The IER registers enable and disable interrupt requests IRQ15 to IRQ0.

- IER16

Bit	Bit Name	Initial Value	R/W	Description
7	IRQ15E	0	R/W	IRQn Enable
6	IRQ14E	0	R/W	The IRQn interrupt request is enabled when this bit is 1.
5	IRQ13E	0	R/W	
4	IRQ12E	0	R/W	(n = 15 to 8)
3	IRQ11E	0	R/W	
2	IRQ10E	0	R/W	
1	IRQ9E	0	R/W	
0	IRQ8E	0	R/W	

- IER

Bit	Bit Name	Initial Value	R/W	Description
7	IRQ7E	0	R/W	IRQn Enable
6	IRQ6E	0	R/W	The IRQn interrupt request is enabled when this bit is 1.
5	IRQ5E	0	R/W	
4	IRQ4E	0	R/W	(n = 7 to 0)
3	IRQ3E	0	R/W	
2	IRQ2E	0	R/W	
1	IRQ1E	0	R/W	
0	IRQ0E	0	R/W	

5.3.6 IRQ Status Registers (ISR16, ISR)

The ISR registers are flag registers that indicate the status of IRQ15 to IRQ0 interrupt requests.

- ISR16

Bit	Bit Name	Initial Value	R/W	Description
7	IRQ15F	0	R/(W)*	[Setting condition]
6	IRQ14F	0	R/(W)*	When the interrupt source selected by the ISCR16 registers occurs
5	IRQ13F	0	R/(W)*	
4	IRQ12F	0	R/(W)*	[Clearing conditions]
3	IRQ11F	0	R/(W)*	• When writing 0 to IRQnF flag after reading IRQnF = 1
2	IRQ10F	0	R/(W)*	
1	IRQ9F	0	R/(W)*	• When interrupt exception handling is executed when low-level detection is set and $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ input is high
0	IRQ8F	0	R/(W)*	• When IRQn interrupt exception handling is executed when falling-edge, rising-edge, or both-edge detection is set
				(n = 15 to 8)
				Note: The $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ pin is selected by the IRQ sense port select register 16 (ISSR16).

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

- ISR

Bit	Bit Name	Initial Value	R/W	Description
7	IRQ7F	0	R/(W)*	[Setting condition]
6	IRQ6F	0	R/(W)*	When the interrupt source selected by the ISCR registers occurs
5	IRQ5F	0	R/(W)*	[Clearing conditions]
4	IRQ4F	0	R/(W)*	• When writing 0 to IRQnF flag after reading IRQnF = 1
3	IRQ3F	0	R/(W)*	• When interrupt exception handling is executed when low-level detection is set and $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ input is high
2	IRQ2F	0	R/(W)*	• When IRQn interrupt exception handling is executed when falling-edge, rising-edge, or both-edge detection is set
1	IRQ1F	0	R/(W)*	(n = 7 to 0)
0	IRQ0F	0	R/(W)*	Note: The $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ pin is selected by the IRQ sense port select register (ISSR). The ExIRQ5 to ExIRQ0 pins are not supported.

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

5.3.7 Keyboard Matrix Interrupt Mask Registers (KMIMRA KMIMR) Wake-Up Event Interrupt Mask Registers (WUEMR)

The KMIMR and WUEMR registers enable or disable key-sensing interrupt inputs ($\overline{\text{KIN15}}$ to $\overline{\text{KIN0}}$) and wake-up event interrupt inputs ($\overline{\text{WUE15}}$ to $\overline{\text{WUE8}}$).

- KMIMRA

Bit	Bit Name	Initial Value	R/W	Description
7	KMIMR15	1	R/W	Keyboard Matrix Interrupt Mask
6	KMIMR14	1	R/W	These bits enable or disable a key-sensing input interrupt request (KIN15 to KIN8).
5	KMIMR13	1	R/W	
4	KMIMR12	1	R/W	0: Enables a key-sensing input interrupt request
3	KMIMR11	1	R/W	1: Disables a key-sensing input interrupt request
2	KMIMR10	1	R/W	
1	KMIMR9	1	R/W	
0	KMIMR8	1	R/W	

- KMIMR

Bit	Bit Name	Initial Value	R/W	Description
7	KMIMR7	1	R/W	Keyboard Matrix Interrupt Mask
6	KMIMR6	0/1*	R/W	These bits enable or disable a key-sensing input interrupt request (KIN7 to KIN0).
5	KMIMR5	1	R/W	
4	KMIMR4	1	R/W	0: Enables a key-sensing input interrupt request
3	KMIMR3	1	R/W	1: Disables a key-sensing input interrupt request
2	KMIMR2	1	R/W	When the EIVS bit in SYSCR3 is cleared to 0, the KMIMR6 bit also simultaneously controls enabling and disabling of the IRQ6 interrupt request. In this case, the initial value of the KMIMR6 bit is 0. When the EIVS bit is set to 1, the initial value of the KMIMR6 bit becomes 1.
1	KMIMR1	1	R/W	
0	KMIMR0	1	R/W	

Note: * The initial value is 0 when EIVS = 0 and the initial value is 1 when EIVS = 1.

- WUEMR

Bit	Bit Name	Initial Value	R/W	Description
7	WUEMR15	1	R/W	Wake-Up Event Interrupt Mask
6	WUEMR14	1	R/W	These bits enable or disable a wake-up event input interrupt request (WUE15 to WUE8).
5	WUEMR13	1	R/W	
4	WUEMR12	1	R/W	0: Enables a wake-up event input interrupt request
3	WUEMR11	1	R/W	1: Disables a wake-up event input interrupt request
2	WUEMR10	1	R/W	
1	WUEMR9	1	R/W	
0	WUEMR8	1	R/W	

Figure 5.2 shows the relation between the IRQ7 and IRQ6 interrupts, KMIMR, and KMIMRA in H8S/2140B Group compatible vector mode. The relation in extended vector mode is shown in figure 5.3.

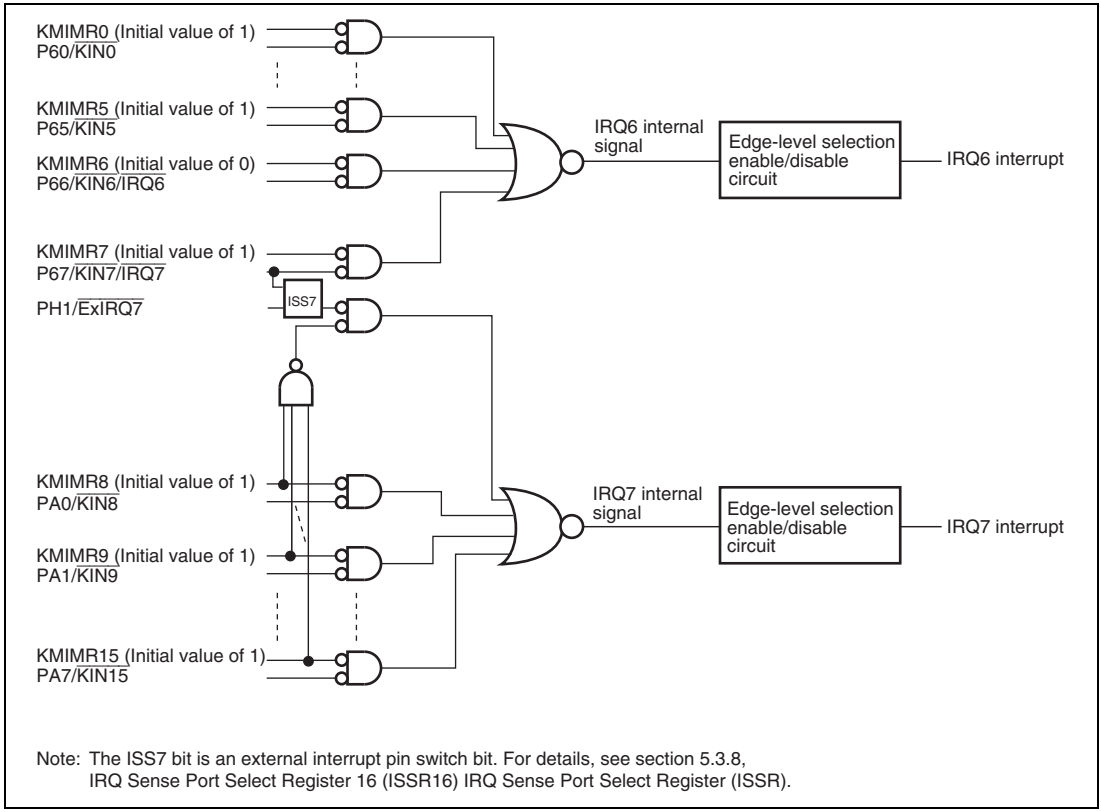
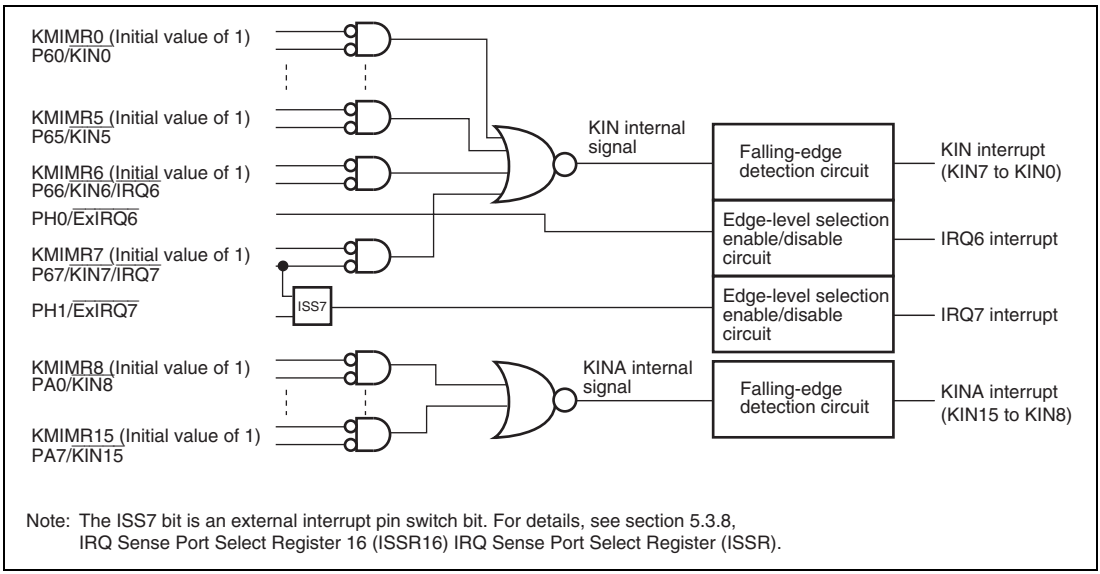


Figure 5.2 Relation between IRQ7/IRQ6 Interrupts and KIN15 to KIN0 Interrupts, KMIMR, and KMIMRA (H8S/2140B Group Compatible Vector Mode: EIVS = 0)

In H8S/2140B Group compatible vector mode, interrupt input from the $\overline{\text{IRQ7}}$ pin is ignored when even one of the KMIMR15 to KMIMR8 bits is cleared to 0. If the $\overline{\text{KIN7}}$ to $\overline{\text{KIN0}}$ pins or $\overline{\text{KIN15}}$ to $\overline{\text{KIN8}}$ pins are specified to be used as key-sensing interrupt input pins and wake-up event interrupt input pins, the interrupt sensing condition for the corresponding interrupt source (IRQ6 or IRQ7) must be set to low-level sensing or falling-edge sensing. Note that interrupt input cannot be made from the $\overline{\text{ExIRQ6}}$ pin.



Note: The ISS7 bit is an external interrupt pin switch bit. For details, see section 5.3.8, IRQ Sense Port Select Register 16 (ISSR16) IRQ Sense Port Select Register (ISSR).

Figure 5.3 Relation between IRQ7 and IRQ6 Interrupts, KIN15 to KIN0 Interrupts, KMIMR, and KMIMRA (Extended Vector Mode: EIVS = 1)

In extended vector mode, the initial value of the KMIMR6 bit is 1. Accordingly, it does not enable of disable the $\overline{\text{IRQ6}}$ pin interrupt. The interrupt input from the $\overline{\text{ExIRQ6}}$ pin becomes the IRQ6 interrupt request.

5.3.8 IRQ Sense Port Select Register 16 (ISSR16)
IRQ Sense Port Select Register (ISSR)

ISSR16 and ISSR select the IRQ15 to IRQ0 interrupt external input from the $\overline{\text{IRQ15}}$ to $\overline{\text{IRQ7}}$ pins and $\overline{\text{ExIRQ15}}$ to $\overline{\text{ExIRQ7}}$ pins.

- ISSR16

Bit	Bit Name	Initial Value	R/W	Description
7	ISS15	0	R/W	0: P97/$\overline{\text{IRQ15}}$ is selected 1: PG7/$\overline{\text{ExIRQ15}}$ is selected
6	ISS14	0	R/W	0: P95/$\overline{\text{IRQ14}}$ is selected 1: PG6/$\overline{\text{ExIRQ14}}$ is selected
5	ISS13	0	R/W	0: P94/$\overline{\text{IRQ13}}$ is selected 1: PG5/$\overline{\text{ExIRQ13}}$ is selected
4	ISS12	0	R/W	0: P93/$\overline{\text{IRQ12}}$ is selected 1: PG4/$\overline{\text{ExIRQ12}}$ is selected
3	ISS11	0	R/W	0: PF3/$\overline{\text{IRQ11}}$ is selected 1: PG3/$\overline{\text{ExIRQ11}}$ is selected
2	ISS10	0	R/W	0: PF2/$\overline{\text{IRQ10}}$ is selected 1: PG2/$\overline{\text{ExIRQ10}}$ is selected
1	ISS9	0	R/W	0: PF1/$\overline{\text{IRQ9}}$ is selected 1: PG1/$\overline{\text{ExIRQ9}}$ is selected
0	ISS8	0	R/W	0: PF0/$\overline{\text{IRQ8}}$ is selected 1: PG0/$\overline{\text{ExIRQ8}}$ is selected

- ISSR

Bit	Bit Name	Initial Value	R/W	Description
7	ISS7	0	R/W	0: P67/$\overline{\text{IRQ7}}$ is selected 1: PH1/$\overline{\text{ExIRQ7}}$ is selected
6 to 0	—	0	R/W	Reserved The initial values should not be changed.

5.3.9 Wake-Up Sense Control Register (WUESCR)
Wake-Up Input Interrupt Status Register (WUESR)
Wake-Up Enable Register (WER)

WUESCR selects the interrupt source of the wake-up event interrupt inputs (WUE15 to WUE8). WUESR is an interrupt request flag register. WER enables/disables interrupts.

- WUESCR

Bit	Bit Name	Initial Value	R/W	Description
7	WUE15SC	0	R/W	Wake-Up Event Interrupt Source Select
6	WUE14SC	0	R/W	These bits select the source that generates an interrupt request at wake-up event interrupt inputs (WUE15 to WUE8).
5	WUE13SC	0	R/W	
4	WUE12SC	0	R/W	0: Interrupt request generated at falling edge of $\overline{\text{WUEn}}$ input
3	WUE11SC	0	R/W	
2	WUE10SC	0	R/W	1: Interrupt request generated at rising edge of $\overline{\text{WUEn}}$ input
1	WUE9SC	0	R/W	
0	WUE8SC	0	R/W	(n = 15 to 8)

- WUESR

Bit	Bit Name	Initial Value	R/W	Description
7	WUE15F	0	R/(W)*	Wake-Up Input Interrupt (WUE15 to WUE8) Request Flag Register
6	WUE14F	0	R/(W)*	
5	WUE13F	0	R/(W)*	These bits are status flags that indicate that wake-up input interrupts (WUE15 to WUE8) are requested.
4	WUE12F	0	R/(W)*	
3	WUE11F	0	R/(W)*	[Setting condition]
2	WUE10F	0	R/(W)*	• When a wake-up input interrupt is generated
1	WUE9F	0	R/(W)*	[Clearing condition]
0	WUE8F	0	R/(W)*	• When 0 is written after reading 1

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

- WER

Bit	Bit Name	Initial Value	R/W	Description
7	WUEE	0	R/W	WUE Enable The WUE interrupt request is enabled when this bit is 1. 0: Wake-up input interrupt request is disabled 1: Wake-up input interrupt request is enabled
6 to 0	—	All 0	R/W	Reserved The initial values should not be changed.

5.4 Interrupt Sources

5.4.1 External Interrupt Sources

The interrupt sources of external interrupts are NMI, IRQ15 to IRQ0, KIN15 to KIN0 and WUE15 to WUE8. These interrupts can be used to restore this LSI from software standby mode.

(1) NMI Interrupt

The nonmaskable external interrupt NMI is the highest-priority interrupt, and is always accepted regardless of the interrupt control mode or the status of the CPU interrupt mask bits. The NMIEG bit in SYSCR can be used to select whether an interrupt is requested at a rising edge or falling edge on the NMI pin.

(2) IRQ15 to IRQ0 Interrupts:

Interrupts IRQ15 to IRQ0 are requested by an input signal at pins $\overline{\text{IRQ15}}$ to $\overline{\text{IRQ0}}$ or pins $\overline{\text{ExIRQ15}}$ to $\overline{\text{ExIRQ6}}$. Interrupts IRQ15 to IRQ0 have the following features:

- The interrupt exception handling for interrupt requests IRQ15 to IRQ0 can be started at an independent vector address.
- Using ISCR, it is possible to select whether an interrupt is generated by a low level, falling edge, rising edge, or both edges, at pins $\overline{\text{IRQ15}}$ to $\overline{\text{IRQ0}}$ or pins $\overline{\text{ExIRQ15}}$ to $\overline{\text{ExIRQ6}}$.
- 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 cleared to 0 by software.

When the interrupts are requested while IRQ15 to IRQ0 interrupt requests are generated at low level of $\overline{\text{IRQn}}$ input, hold the corresponding $\overline{\text{IRQ}}$ input at low level until the interrupt handling starts. Then put the relevant $\overline{\text{IRQ}}$ input back to high level within the interrupt handling routine and clear the IRQnF bit (n = 15 to 0) in ISR to 0. If the relevant $\overline{\text{IRQ}}$ input is put back to high level before the interrupt handling starts, the relevant interrupt may not be executed.

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

A block diagram of interrupts IRQ15 to IRQ0 is shown in figure 5.4.

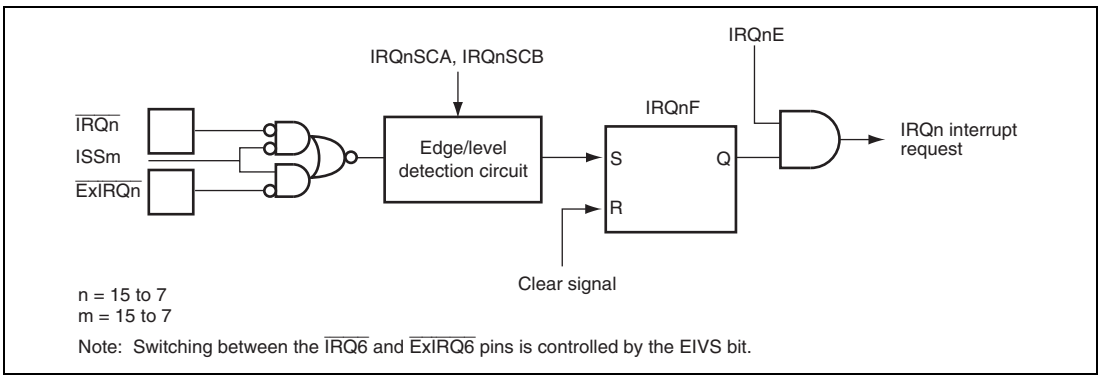


Figure 5.4 Block Diagram of Interrupts IRQ15 to IRQ0

(3) KIN15 to KIN0 Interrupts

Interrupts KIN15 to KIN0 are requested by the input signals on pins $\overline{\text{KIN15}}$ to $\overline{\text{KIN0}}$. Functions of interrupts KIN15 to KIN0 change as follows according to the setting of the EIVS bit in system control register 3 (SYSCR3).

- H8S/2140B Group compatible vector mode (EIVS = 0 in SYSCR3)
  - — Interrupts KIN15 to KIN8 correspond to interrupt IRQ7, and interrupts KIN7 to KIN0 correspond to interrupt IRQ6. The pin conditions for generating an interrupt request, whether the interrupt request is enabled, interrupt control level setting, and status of the interrupt request for the above interrupts are in accordance with the settings and status of the relevant interrupts IRQ7 and IRQ6.
  - — KIN15 to KIN0 interrupt requests can be masked by using KMIMRA and KMIMR.
  - — If the $\overline{\text{KIN7}}$ to $\overline{\text{KIN0}}$ pins are specified to be used as key-sensing interrupt input pins, the interrupt sensing condition for the corresponding interrupt source (IRQ6 or IRQ7) must be set to low-level sensing or falling-edge sensing.

— When using the $\overline{\text{IRQ6}}$ pin as the IRQ6 interrupt input pin, the KMIMR6 bit must be cleared to 0. When using the $\overline{\text{IRQ7}}$ pin as the IRQ7 interrupt input pin, the KMIMR15 to KMIMR8 bits must all be set to 1. If even one of these bits is cleared to 0, the IRQ7 interrupt input from the $\overline{\text{IRQ7}}$ pin is ignored.

- Extended vector mode (EIVS = 1 in SYSCR3)
  - — Interrupts KIN15 to KIN8 and KIN7 to KIN0, each form a group. The interrupt exception handling for an interrupt request from the same group is started at the same vector address.
  - — Interrupt requests are generated on the falling edge of pins $\overline{\text{KIN15}}$ to $\overline{\text{KIN0}}$.
  - — Interrupt requests KIN15 to KIN0 can be masked by using KMIMRA and KMIMR.
  - — The status of interrupt requests KIN15 to KIN0 are not indicated.

    An IRQ6 interrupt is enabled only by input to the $\overline{\text{ExIRQ6}}$ pin. The $\overline{\text{IRQ6}}$ pin is only available for a KIN interrupt input, and functions as the $\overline{\text{KIN6}}$ pin. The initial value of the KMIMR6 bit is 1. For the IRQ7 interrupt, either the $\overline{\text{IRQ7}}$ pin or $\overline{\text{ExIRQ7}}$ pin can be selected as the input pin using the ISS7 bit. The IRQ7 interrupt is not affected by the settings of bits KMIMR15 to KMIMR8. The detection of interrupts KIN15 to KIN0 does not depend on whether the relevant pin has been set for input or output. Therefore, when a pin is used as an external interrupt input pin, clear the DDR bit of the corresponding port to 0 so it is not used as an I/O pin for another function.

(4) WUE15 to WUE8 Interrupts

Interrupt requests WUE15 to WUE8 can be configured regardless of the setting of the EIVS bit in system control register 3 (SYSCR3).

A block diagram of interrupts WUE15 to WUE8 is shown in figure 5.5.

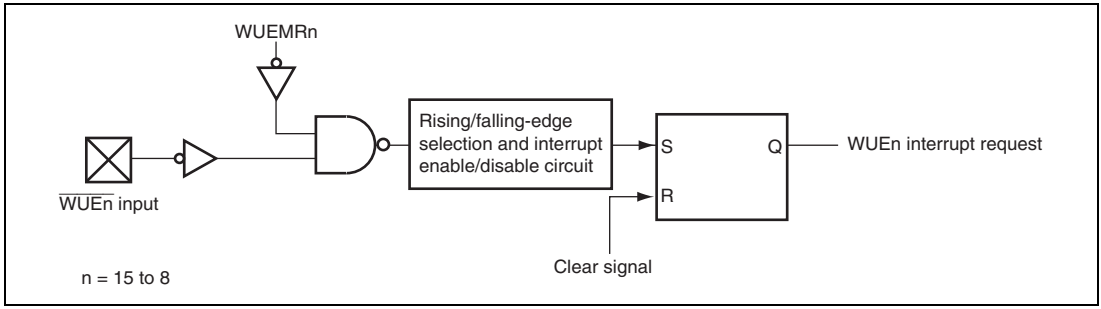


Figure 5.5 Block Diagram of Interrupts WUE15 to WUE8

5.4.2 Internal Interrupt Sources

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

- For each on-chip peripheral module there are flags that indicate the interrupt request status, and enable bits that individually select enabling or disabling of these interrupts. When the enable bit for a particular interrupt source is set to 1, an interrupt request is sent to the interrupt controller.
- The control level for each interrupt can be set by ICR.

5.5 Interrupt Exception Handling Vector Tables

Tables 5.4 and 5.5 list interrupt exception handling sources, vector addresses, and interrupt priorities. H8S/2140B Group compatible vector mode or extended vector mode can be selected for the vector addresses by the EIVS bit in system control register 3 (SYSCR3).

For default priorities, the lower the vector number, the higher the priority. Modules set at the same priority will conform to their default priorities. Priorities within a module are fixed.

An interrupt control level can be specified for a module to which an ICR bit is assigned. Interrupt requests from modules that are set to interrupt control level 1 (priority) by the interrupt control level and the I and UI bits in CCR are given priority and processed before interrupt requests from modules that are set to interrupt control level 0 (no priority).

Table 5.5 Interrupt Sources, Vector Addresses, and Interrupt Priorities (H8S/2140B Group Compatible Vector Mode)

Origin of Interrupt Source	Name	Vector Number	Vector Address: Advanced Mode	ICR	Priority
External pin	NMI	7	H'00001C	—	High
	IRQ0	16	H'000040	ICRA7	▲
	IRQ1	17	H'000044	ICRA6	
	IRQ2	18	H'000048	ICRA5	
	IRQ3	19	H'00004C		
	IRQ4	20	H'000050	ICRA4	
	IRQ5	21	H'000054		
	IRQ6, KIN7 to KIN0	22	H'000058	ICRA3	
	IRQ7, KIN15 to KIN8	23	H'00005C		Low

Origin of Interrupt Source	Name	Vector Number	Vector Address Advanced Mode	ICR	Priority
—	Reserved for system use	24	H'000060	—	High
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	
—	Reserved for system use	29 | 32	H'000074 | H'000080	—	
External pin	WUE15 to WUE8	33	H'000084	ICRD4	
TPU_0	TGI0A (TGR0A input capture/compare match) TGI0B (TGR0B input capture/compare match) TGI0C (TGR0C input capture/compare match) TGI0D (TGR0D input capture/compare match) TGI0V (Overflow 0)	34 35 36 37 38	H'000088 H'00008C H'000090 H'000094 H'000098	ICRD3	
TPU_1	TGI1A (TGR1A input capture/compare match) TGI1B (TGR1B input capture/compare match) TGI1V (Overflow 1) TGI1U (Underflow 1)	39 40 41 42	H'00009C H'0000A0 H'0000A4 H'0000A8	ICRD2	
TPU_2	TGI2A (TGR2A input capture/compare match) TGI2B (TGR2B input capture/compare match) TGI2V (Overflow 2) TGI2U (Underflow 2)	43 44 45 46	H'0000AC H'0000B0 H'0000B4 H'0000B8	ICRD1	
—	Reserved for system use	47	H'0000BC	—	
TCM_0	TICI0 (Input capture) TCMI0 (Compare match) TOVMI0 (Cycle overflow) TUDI0 (Cycle underflow) TOVI0 (Overflow)	48	H'0000C0	ICRB6	
TCM_1	TICI1 (Input capture) TCMI1 (Compare match) TOVMI1 (Cycle overflow) TUDI1 (Cycle underflow) TOVI1 (Overflow)	49	H'0000C4		Low

Origin of Interrupt Source	Name	Vector Number	Vector Address: Advanced Mode	ICR	Priority
TCM_2	TICI2 (Input capture) TCMI2 (Compare match) TOVMI2 (Cycle overflow) TUDI2 (Cycle underflow) TOVI2 (Overflow)	50	H'0000C8	ICRB6	High
TCM_3	TICI3 (Input capture) TCMI3 (Compare match) TOVMI3 (Cycle overflow) TUDI3 (Cycle underflow) TOVI3 (Overflow)	51	H'0000CC		
TDP_0	TICI0 (Input capture) TCMI0 (Compare match) TPDMXI0 (Cycle overflow) TPDMNI0 (Cycle underflow) TWDMNI0 (Pulse width lower limit underflow) TWDMXI0 (Pulse width upper limit overflow) TOVI0 (Overflow)	52	H'0000D0	ICRB5	
TDP_1	TICI1 (Input capture) TCMI1 (Compare match) TPDMXI1 (Cycle overflow) TPDMNI1 (Cycle underflow) TWDMNI1 (Pulse width lower limit underflow) TWDMXI1 (Pulse width upper limit overflow) TOVI1 (Overflow)	53	H'0000D4		
TDP_2	TICI2 (Input capture) TCMI2 (Compare match) TPDMXI2 (Cycle overflow) TPDMNI2 (Cycle underflow) TWDMNI2 (Pulse width lower limit underflow) TWDMXI2 (Pulse width upper limit overflow) TOVI2 (Overflow)	54	H'0000D8		
—	Reserved for system use	55	H'0000DC	—	
External pin	IRQ8 IRQ9 IRQ10 IRQ11	56 57 58 59	H'0000E0 H'0000E4 H'0000E8 H'0000EC	ICRD7	
	IRQ12 IRQ13 IRQ14 IRQ15	60 61 62 63	H'0000F0 H'0000F4 H'0000F8 H'0000FC	ICRD6	Low

Origin of Interrupt Source	Name	Vector Number	Vector Address Advanced Mode	ICR	Priority
TMR_0	CMIA0 (Compare match A) CMIB0 (Compare match B) OVI0 (Overflow)	64 65 66	H'000100 H'000104 H'000108	ICRB3	High
—	Reserved for system use	67	H'00010C	—	
TMR_1	CMIA1 (Compare match A) CMIB1 (Compare match B) OVI1 (Overflow)	68 69 70	H'000110 H'000114 H'000118	ICRB2	
—	Reserved for system use	71	H'00011C	—	
TMR_X TMR_Y	CMIAY (Compare match A) CMIBY (Compare match B) OVIY (Overflow) ICIX (Input capture) CMIAX (Compare match A) CMIBX (Compare match B) OVIX (Overflow)	72 73 74 75 76 77 78	H'000120 H'000124 H'000128 H'00012C H'000130 H'000134 H'000138	ICRB1	
FSI	FSII (Transmission/reception completion)	79	H'00013C	ICRC2	
—	Reserved for system use Reserved for system use	80 81	H'000140 H'000144	—	
SCIF	SCIF (SCIF interrupt)	82	H'000148	ICRC7	
—	Reserved for system use	83	H'00014C	—	
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	
SCI_2	ERI2 (Reception error) RXI2 (Reception completion) TXI2 (Transmission data empty 2) TEI2 (Transmission end 2)	88 89 90 91	H'000160 H'000164 H'000168 H'00016C	ICRC5	
IIC_0	IICI0 (1-byte transmission/reception completion)	92	H'000170	ICRC4	
CIR	RENDI (Reception end) OVEI (Overrun error) REPI (Repeat detection) FREI (Framing error) ABI (Abort) HEADFI (Header detection)	93	H'000174	ICRB4	
IIC_1	IICI1 (1-byte transmission/reception completion)	94	H'000178	ICRC3	
IIC_2	IICI2 (1-byte transmission/reception completion)	95	H'00017C		Low

Origin of Interrupt Source	Name	Vector Number	Vector Address: Advanced Mode	ICR	Priority
PS2	KBIA (Reception completion A)	96	H'000180	ICRB0	High
	KBIB (Reception completion B)	97	H'000184		
	KBIC (Reception completion C)	98	H'000188		
	KBTIA (Transmission completion A)/ KBCA (1st KCLKA)	99	H'00018C		
	KBTIB (Transmission completion B)/ KBCB (1st KCLKB)	100	H'000190		
	KBTIC (Transmission completion C)/ KBCC (1st KCLKC)	101	H'000194		
	KBID (Reception completion D)	102	H'000198		
	KBTID (Transmission completion D)/KBCD (1st KCLKD)	103	H'00019C		
FSI	LFSII (Command reception)/(Write reception)	104	H'0001A0	ICRC1	
—	Reserved for system use	105	H'0001A4	—	
LPC	OBEI (ODR1 to 4 transmission completion)	106	H'0001A8	ICRC1	
	IBFI4 (IDR4 reception completion)	107	H'0001AC		
	ERR1 (Transfer error, etc.)	108	H'0001B0		
	IBFI1 (IDR1 reception completion)	109	H'0001B4		
	IBFI2 (IDR2 reception completion)	110	H'0001B8		
	IBFI3 (IDR3 reception completion)	111	H'0001BC		
—	Reserved for system use	112	H'0001C0	—	
		127	H'0001FC		Low

Table 5.6 Interrupt Sources, Vector Addresses, and Interrupt Priorities (Extended Vector Mode)

Origin of Interrupt Source	Name	Vector Number	Vector Address: Advanced Mode	ICR	Priority
External pin	NMI	7	H'00001C	—	High
	IRQ0	16	H'000040	ICRA7	
	IRQ1	17	H'000044	ICRA6	
	IRQ2	18	H'000048	ICRA5	
	IRQ3	19	H'00004C		
	IRQ4	20	H'000050	ICRA4	
	IRQ5	21	H'000054		
	IRQ6	22	H'000058	ICRA3	
	IRQ7	23	H'00005C		
—	Reserved for system use	24	H'000060	—	
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	
—	Reserved for system use	29	H'000074	—	
External pin	KIN7 to KIN0	30	H'000078	ICRD5	
	KIN15 to KIN8	31	H'00007C		
—	Reserved for system use	32	H'000080	—	
External pin	WUE15 to WUE8	33	H'000084	ICRD4	
TPU_0	TGI0A (TGR0A input capture/compare match)	34	H'000088	ICRD3	
	TGI0B (TGR0B input capture/compare match)	35	H'00008C		
	TGI0C (TGR0C input capture/compare match)	36	H'000090		
	TGI0D (TGR0D input capture/compare match)	37	H'000094		
	TGI0V (Overflow 0)	38	H'000098		
TPU_1	TGI1A (TGR1A input capture/compare match)	39	H'00009C	ICRD2	
	TGI1B (TGR1B input capture/compare match)	40	H'0000A0		
	TGI1V (Overflow 1)	41	H'0000A4		
	TGI1U (Underflow 1)	42	H'0000A8		Low

Origin of Interrupt Source	Name	Vector Number	Vector Address: Advanced Mode	ICR	Priority
TPU_2	TGI2A (TGR2A input capture/compare match)	43	H'0000AC	ICRD1	High
	TGI2B (TGR2B input capture/compare match)	44	H'0000B0		
	TGI2V (Overflow 1)	45	H'0000B4		
	TGI2U (Underflow 2)	46	H'0000B8		
—	Reserved for system use	47	H'0000BC	—	
TCM_0	TICI0 (Input capture) TCMI0 (Compare match) TOVMI0 (Cycle overflow) TUDI0 (Cycle underflow) TOVI0 (Overflow)	48	H'0000C0	ICRB6	
TCM_1	TICI1 (Input capture) TCMI1 (Compare match) TOVMI1 (Cycle overflow) TUDI1 (Cycle underflow) TOVI1 (Overflow)	49	H'0000C4		
TCM_2	TICI2 (Input capture) TCMI2 (Compare match) TOVMI2 (Cycle overflow) TUDI2 (Cycle underflow) TOVI2 (Overflow)	50	H'0000C8		
TCM_3	TICI3 (Input capture) TCMI3 (Compare match) TOVMI3 (Cycle overflow) TUDI3 (Cycle underflow) TOVI3 (Overflow)	51	H'0000CC		
TDP_0	TICI0 (Input capture) TCMI0 (Compare match) TPDMXI0 (Cycle overflow) TPDMNI0 (Cycle underflow) TWDMNI0 (Pulse width lower limit underflow) TWDMXI0 (Pulse width upper limit overflow) TOVI0 (Overflow)	52	H'0000D0	ICRB5	
TDP_1	TICI1 (Input capture) TCMI1 (Compare match) TPDMXI1 (Cycle overflow) TPDMNI1 (Cycle underflow) TWDMNI1 (Pulse width lower limit underflow) TWDMXI1 (Pulse width upper limit overflow) TOVI1 (Overflow)	53	H'0000D4		Low

Origin of Interrupt Source	Name	Vector Number	Vector Address: Advanced Mode	ICR	Priority
TDP_2	TICI2 (Input capture) TCMI2 (Compare match) TPDMXI2 (Cycle overflow) TPDMNI2 (Cycle underflow) TWDMNI2 (Pulse width lower limit underflow) TWDMXI2 (Pulse width upper limit overflow) TOVI2 (Overflow)	54	H'0000D8	ICRB5	High
—	Reserved for system use	55	H'0000DC	—	
External pin	IRQ8	56	H'0000E0	ICRD7	
	IRQ9	57	H'0000E4		
	IRQ10	58	H'0000E8		
	IRQ11	59	H'0000EC		
	IRQ12	60	H'0000F0	ICRD6	
	IRQ13	61	H'0000F4		
	IRQ14	62	H'0000F8		
	IRQ15	63	H'0000FC		
TMR_0	CMIA0 (Compare match A)	64	H'000100	ICRB3	
	CMIB0 (Compare match B)	65	H'000104		
	OVI0 (Overflow)	66	H'000108		
—	Reserved for system use	67	H'00010C	—	
TMR_1	CMIA1 (Compare match A)	68	H'000110	ICRB2	
	CMIB1 (Compare match B)	69	H'000114		
	OVI1 (Overflow)	70	H'000118		
—	Reserved for system use	71	H'00011C	—	
TMR_X TMR_Y	CMIAY (Compare match A)	72	H'000120	ICRB1	
	CMIBY (Compare match B)	73	H'000124		
	OVIY (Overflow)	74	H'000128		
	ICIX (Input capture)	75	H'00012C		
	CMIAX (Compare match A)	76	H'000130		
	CMIBX (Compare match B)	77	H'000134		
	OVIX (Overflow)	78	H'000138		
FSI	FSII (Transmission/reception completion)	79	H'00013C	ICRC2	
—	Reserved for system use	80	H'000140	—	
	Reserved for system use	81	H'000144		
SCIF	SCIF (SCIF interrupt)	82	H'000148	ICRC7	
—	Reserved for system use	83	H'00014C	—	
SCI_1	ERI1 (Reception error 1)	84	H'000150	ICRC6	
	RXI1 (Reception completion 1)	85	H'000154		
	TXI1 (Transmission data empty 1)	86	H'000158		
	TEI1 (Transmission end 1)	87	H'00015C		Low

Origin of Interrupt Source	Name	Vector Number	Vector Address Advanced Mode	ICR	Priority
SCI_2	ERI2 (Reception error) RXI2 (Reception completion) TXI2 (Transmission data empty 2) TEI2 (Transmission end 2)	88 89 90 91	H'000160 H'000164 H'000168 H'00016C	ICRC5	High
IIC_0	IICI0 (1-byte transmission/reception completion)	92	H'000170	ICRC4	
CIR	RENDI (Reception end) OVEI (Overrun error) REPI (Repeat detection) FREI (Framing error) ABI (Abort) HEADFI (Header detection)	93	H'000174	ICRB4	
IIC_1	IICI1 (1-byte transmission/reception completion)	94	H'000178	ICRC3	
IIC_2	IICI2 (1-byte transmission/reception completion)	95	H'00017C		
PS2	KBIA (Reception completion A) KBIB (Reception completion B) KBIC (Reception completion C) KBTIA (Transmission completion A)/ KBCA (1st KCLKA) KBTIB (Transmission completion B)/ KBCB (1st KCLKB) KBTIC (Transmission completion C)/ KBCC (1st KCLKC) KBID (Reception completion D) KBTID (Transmission completion D)/KBCD (1st KCLKD)	96 97 98 99 100 101 102 103	H'000180 H'000184 H'000188 H'00018C H'000190 H'000194 H'000198 H'00019C	ICRB0	
FSI	LFSII (Command reception)/(Write reception)	104	H'0001A0	ICRC1	
—	Reserved for system use	105	H'0001A4	—	
LPC	OBEI (ODR1 to 4 transmission completion) IBFI4 (IDR4 reception completion) ERR1 (Transfer error, etc.) IBFI1 (IDR1 reception completion) IBFI2 (IDR2 reception completion) IBFI3 (IDR3 reception completion)	106 107 108 109 110 111	H'0001A8 H'0001AC H'0001B0 H'0001B4 H'0001B8 H'0001BC	ICRC1	
—	Reserved for system use	112 ⋮ 127	H'0001C0 ⋮ H'0001FC	—	Low

5.6 Interrupt Control Modes and Interrupt Operation

The interrupt controller has two modes: interrupt control mode 0 and interrupt control mode 1. Interrupt operations differ depending on the interrupt control mode. NMI and address break interrupts are always accepted except for in the reset state. The interrupt control mode is selected by SYSCR. Table 5.7 shows the interrupt control modes.

Table 5.7 Interrupt Control Modes

Interrupt Control Mode	SYSCR		Priority Setting Registers	Interrupt Mask Bits	Description
	INTM1	INTM0			
0	0	0	ICR	I	Interrupt mask control is performed by the I bit. Priority levels can be set with ICR.
1	0	1	ICR	I, UI	3-level interrupt mask control is performed by the I and UI bits. Priority levels can be set with ICR.

Figure 5.6 shows a block diagram of the priority determination circuit.

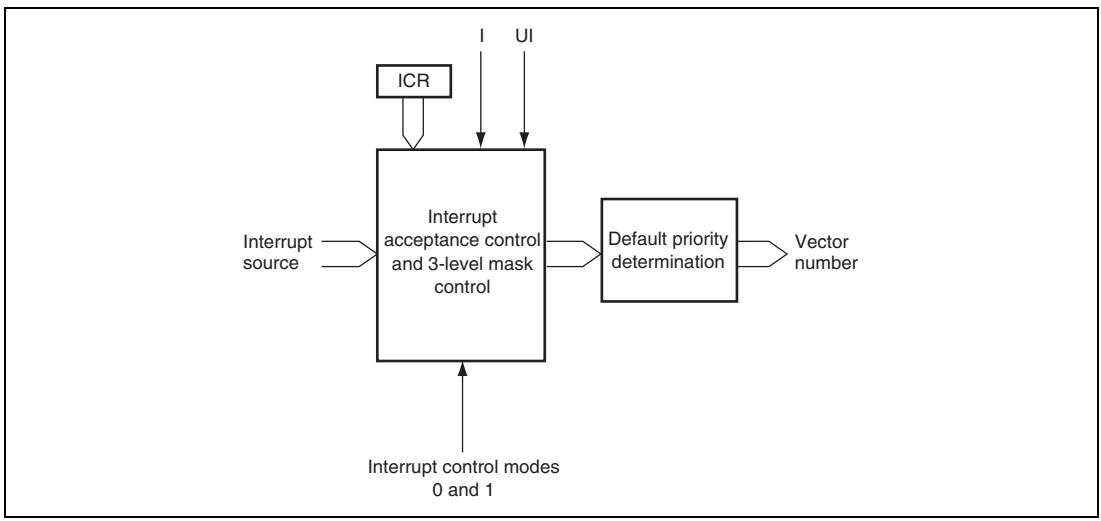


Figure 5.6 Block Diagram of Interrupt Control Operation

(1) Interrupt Acceptance Control and 3-Level Control

In interrupt control modes 0 and 1, interrupt acceptance control and 3-level mask control is performed by means of the I and UI bits in CCR and ICR (control level).

Table 5.8 shows the interrupts selected in each interrupt control mode.

Table 5.8 Interrupts Selected in Each Interrupt Control Mode

	Interrupt Mask Bits		
Interrupt Control Mode	**I**	**UI**	**Selected Interrupts**
0	0	*	All interrupts (interrupt control level 1 has priority)
	1	*	NMI and address break interrupts
1	0	*	All interrupts (interrupt control level 1 has priority)
	1	0	NMI, address break, and interrupt control level 1 interrupts
		1	NMI and address break interrupts

[Legend]
*: Don't care

(2) Default Priority Determination

The priority is determined for the selected interrupt, and a vector number is generated.

If the same value is set for ICR, acceptance of multiple interrupts is enabled, and so only the interrupt source with the highest priority according to the preset default priorities is selected and has a vector number generated.

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

Table 5.9 shows operations and control signal functions in each interrupt control mode.

Table 5.9 Operations and Control Signal Functions in Each Interrupt Control Mode

Interrupt Control Mode	Setting			Interrupt Acceptance Control 3-Level Control			Default Priority Determination
	INTM1	INTM0		I	UI	ICR	
0	0	0	O	IM	—	PR	O
1		1	O	IM	IM	PR	O

[Legend]

O: Interrupt operation control is performed

IM: Used as an interrupt mask bit

PR: Priority is set

—: Not used

5.6.1 Interrupt Control Mode 0

In interrupt control mode 0, interrupt requests other than NMI and address break are masked by ICR and the I bit of CCR in the CPU. Figure 5.7 shows a flowchart of the interrupt acceptance operation.

1. If an interrupt source occurs when the corresponding interrupt enable bit is set to 1, an interrupt request is sent to the interrupt controller.
2. According to the interrupt control level specified in ICR, the interrupt controller only accepts an interrupt request with interrupt control level 1 (priority), and holds pending an interrupt request with interrupt control level 0 (no priority). If several interrupt requests are issued, an interrupt request with the highest priority is accepted according to the priority order, an interrupt handling is requested to the CPU, and other interrupt requests are held pending.
3. If the I bit in CCR is set to 1, the interrupt controller holds pending interrupt requests other than NMI and address break. If the I bit is cleared to 0, any interrupt request is accepted.
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 saved on the stack shows the address of the first instruction to be executed after returning from the interrupt handling routine.
6. Next, the I bit in CCR is set to 1. This masks all interrupts except for NMI and address break interrupts.

7. The CPU generates a vector address for the accepted interrupt request and starts execution of the interrupt handling routine at the address indicated by the contents of the vector address in the vector table.

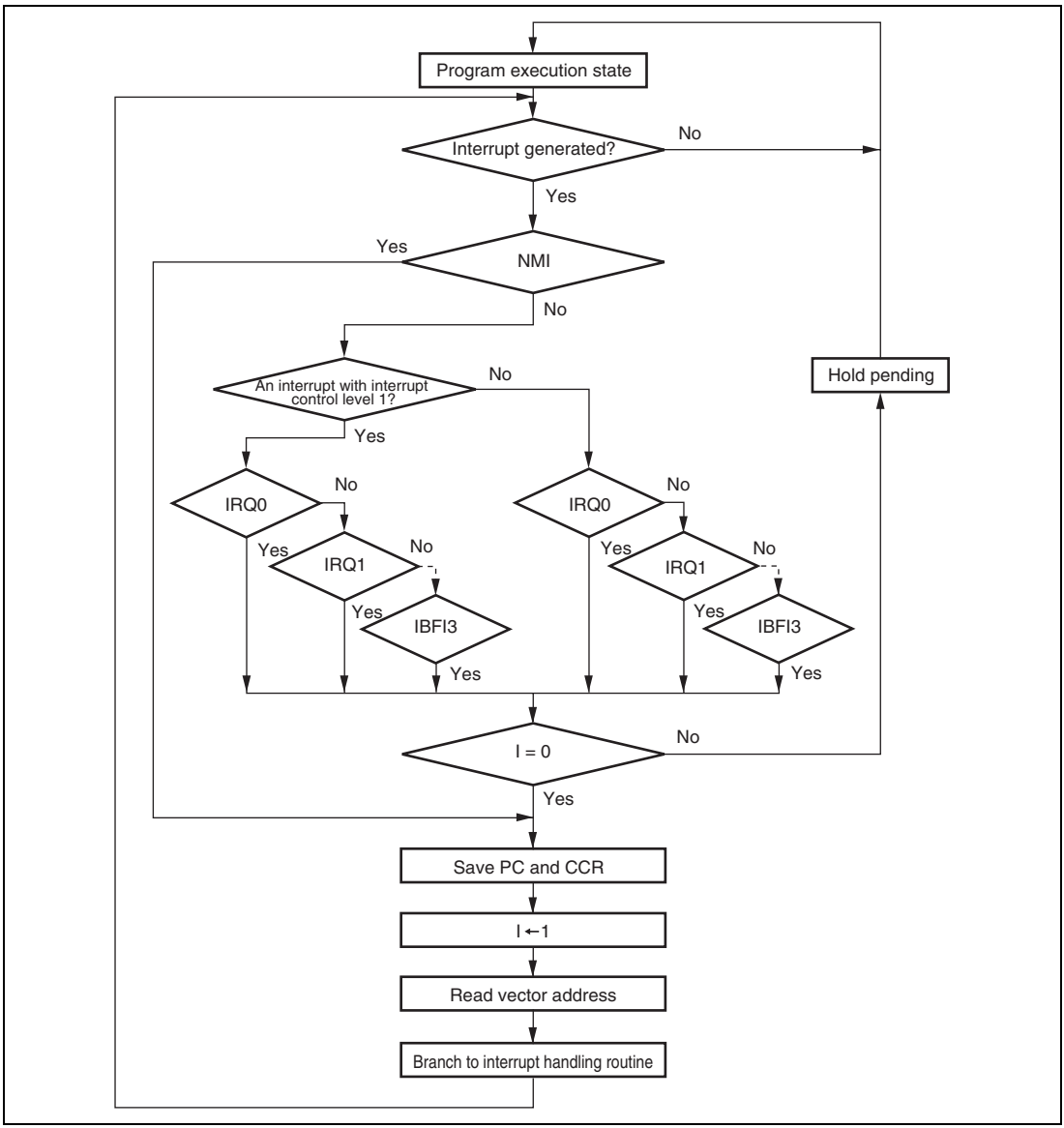


Figure 5.7 Flowchart of Procedure up to Interrupt Acceptance in Interrupt Control Mode 0

5.6.2 Interrupt Control Mode 1

In interrupt control mode 1, mask control is applied to three levels for interrupt requests other than NMI and address break by comparing the I and UI bits in CCR in the CPU, and the ICR setting.

- An interrupt request with interrupt control level 0 is accepted when the I bit in CCR is cleared to 0. When the I bit is set to 1, the interrupt request is held pending.
- An interrupt request with interrupt control level 1 is accepted when the I bit or UI bit in CCR is cleared to 0. When both the I and UI bits are set to 1, the interrupt request is held pending.

For instance, the state transition when the interrupt enable bit corresponding to each interrupt is set to 1, and ICRA to ICRD are set to H'20, H'00, H'00, and H'00, respectively (IRQ2 and IRQ3 interrupts are set to interrupt control level 1, and other interrupts are set to interrupt control level 0) is shown below. Figure 5.8 shows a state transition diagram.

- All interrupt requests are accepted when I = 0. (Priority order: NMI > IRQ2 > IRQ3 > address break > IRQ0 > IRQ1 …)
- Only NMI, IRQ2, IRQ3, and address break interrupt requests are accepted when I = 1 and UI = 0.
- Only NMI and address break interrupt requests are accepted when I = 1 and UI = 1.

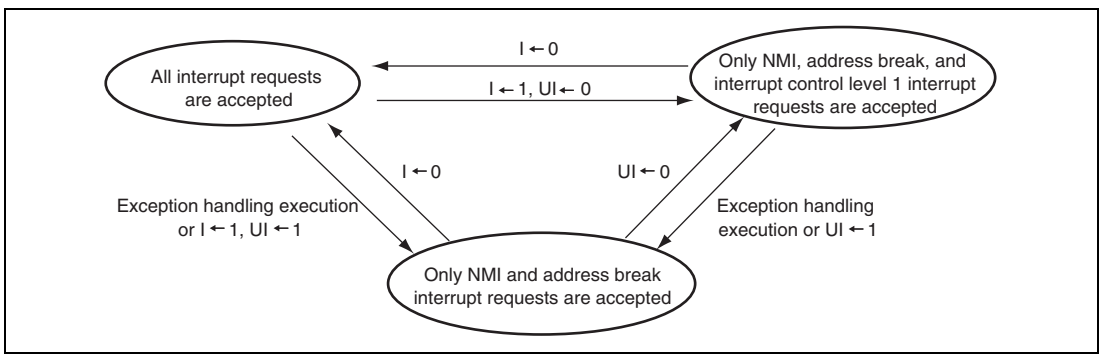


Figure 5.8 State Transition in Interrupt Control Mode 1

Figure 5.9 shows a flowchart of the interrupt acceptance operation.

1. If an interrupt source occurs when the corresponding interrupt enable bit is set to 1, an interrupt request is sent to the interrupt controller.
2. According to the interrupt control level specified in ICR, the interrupt controller only accepts an interrupt request with interrupt control level 1 (priority), and holds pending an interrupt request with interrupt control level 0 (no priority). If several interrupt requests are issued, an interrupt request with the highest priority is accepted according to the priority order, an interrupt handling is requested to the CPU, and other interrupt requests are held pending.
3. An interrupt request with interrupt control level 1 is accepted when the I bit is cleared to 0, or when the I bit is set to 1 while the UI bit is cleared to 0.

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

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

   When the I bit is cleared to 0, the UI bit does not affect acceptance of interrupt requests.
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 saved on the stack shows the address of the first instruction to be executed after returning from the interrupt handling routine.
6. The I and UI bits in CCR are set to 1. This masks all interrupts except for NMI and address break interrupts.
7. The CPU generates a vector address for the accepted interrupt request and starts execution of the interrupt handling routine at the address indicated by the contents of the vector address in the vector table.

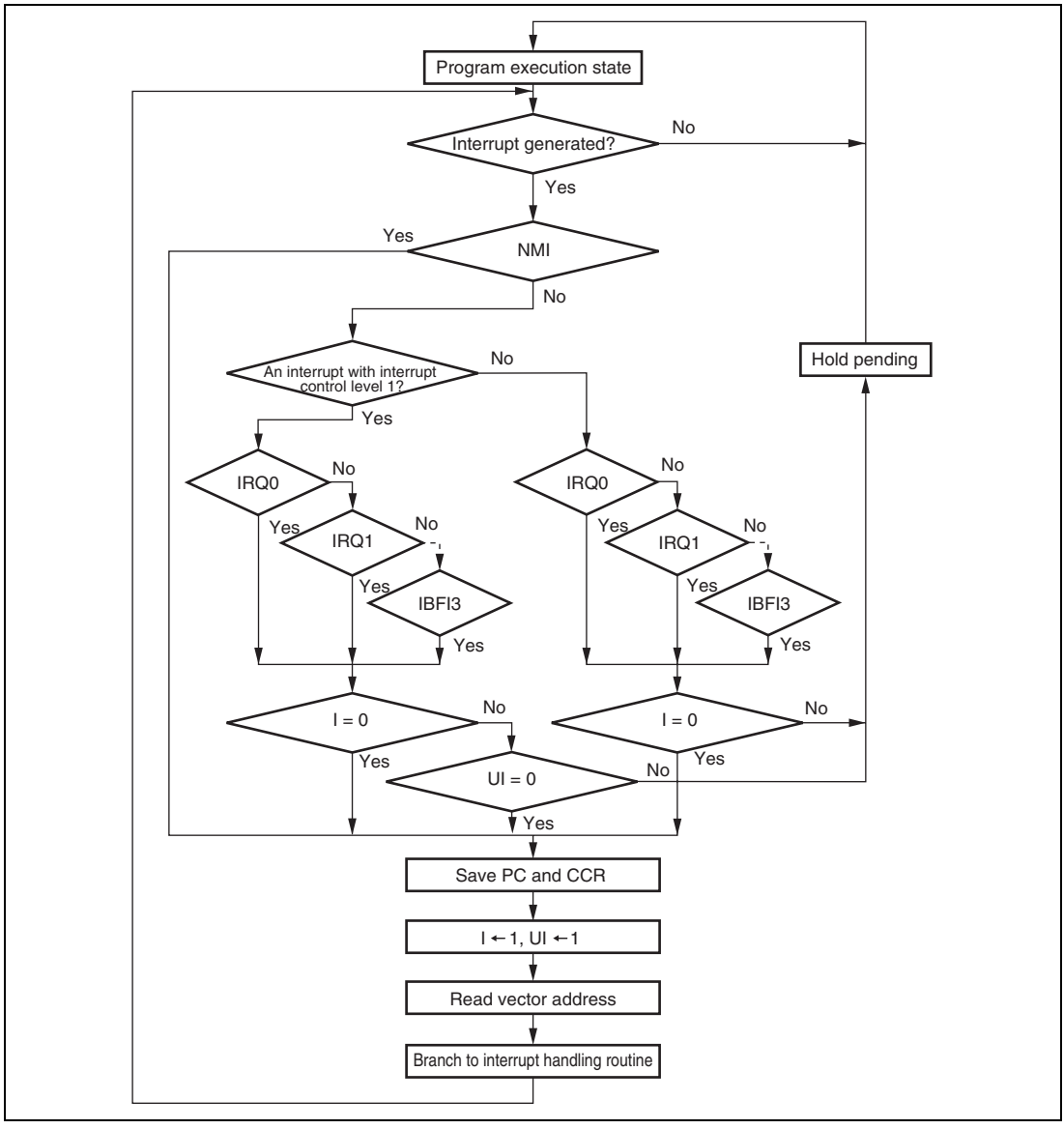


Figure 5.9 Flowchart of Procedure up to Interrupt Acceptance in Interrupt Control Mode 1

5.6.3 Interrupt Exception Handling Sequence

Figure 5.10 shows the interrupt exception handling sequence. The example shown is for the case where interrupt control mode 0 is set in advanced mode, and the program area and stack area are in on-chip memory.

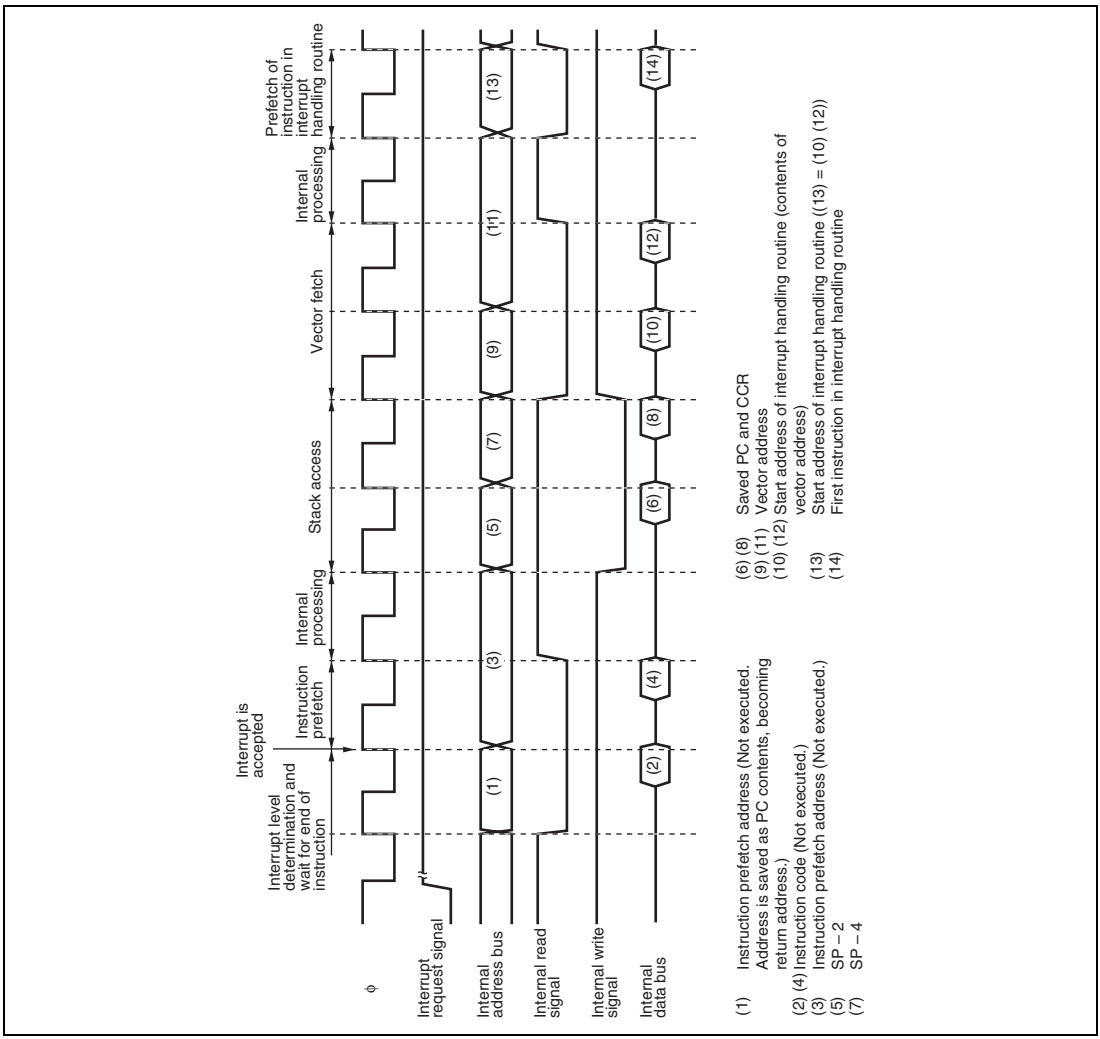


Figure 5.10 Interrupt Exception Handling

5.6.4 Interrupt Response Times

Table 5.10 shows interrupt response times – the intervals between generation of an interrupt request and execution of the first instruction in the interrupt handling routine.

Table 5.10 Interrupt Response Times

No.	Execution Status	Advanced Mode
1	Interrupt priority determination*[1]	3
2	Number of wait states until executing instruction ends*[2]	1 to 21
3	Saving of PC and CCR in stack	2
4	Vector fetch	2
5	Instruction fetch*[3]	2
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 vector fetch.

5.7 Address Breaks

5.7.1 Features

With this LSI, it is possible to identify the prefetch of a specific address by the CPU and generate an address break interrupt, using the ABRKCR and BAR registers. When an address break interrupt is generated, address break interrupt exception handling is executed.

This function can be used to detect the beginning of execution of a bug location in the program, and branch to a correction routine.

5.7.2 Block Diagram

Figure 5.11 shows a block diagram of the address break function.

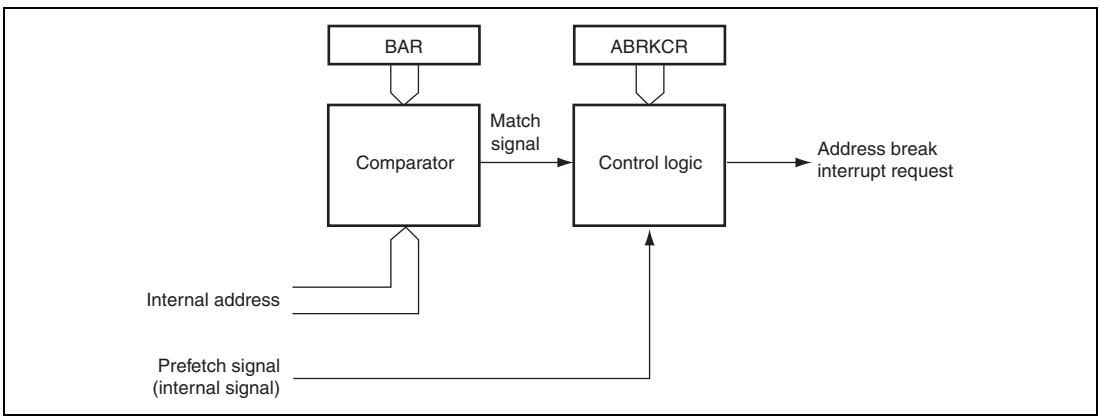


Figure 5.11 Block Diagram of Address Break Function

5.7.3 Operation

ABRKCR and BAR settings can be made so that an address break interrupt is generated when the CPU prefetches the address set in BAR. This address break function issues an interrupt request to the interrupt controller when the address is prefetched, and the interrupt controller determines the interrupt priority. When the interrupt is accepted, interrupt exception handling is started on completion of the currently executing instruction. With an address break interrupt, interrupt mask control by the I and UI bits in the CPU's CCR is ineffective.

The register settings when the address break function is used are as follows.

1. Set the break address in bits A23 to A1 in BAR.
2. Set the BIE bit in ABRKCR to 1 to enable address breaks. An address break will not be requested if the BIE bit is cleared to 0.

When the setting condition occurs, the CMF flag in ABRKCR is set to 1 and an interrupt is requested. If necessary, the source should be identified in the interrupt handling routine.

5.7.4 Usage Notes

- With the address break function, the address at which the first instruction byte is located should be specified as the break address. Occurrence of the address break condition may not be recognized for other addresses.
- If a branch instruction (Bcc, BSR) jump instruction (JMP, JSR), RTS instruction, or RTE instruction is located immediately before the address set in BAR, execution of this instruction will output a prefetch signal for that address, and an address break may be requested. This can be prevented by not making a break address setting for an address immediately following one of these instructions, or by determining within the interrupt handling routine whether interrupt handling was initiated by a genuine condition occurrence.
- As an address break interrupt is generated by a combination of the internal prefetch signal and address, the timing of the start of interrupt exception handling depends on the content and execution cycle of the instruction at the set address and the preceding instruction. Figure 5.12 shows some address timing examples.

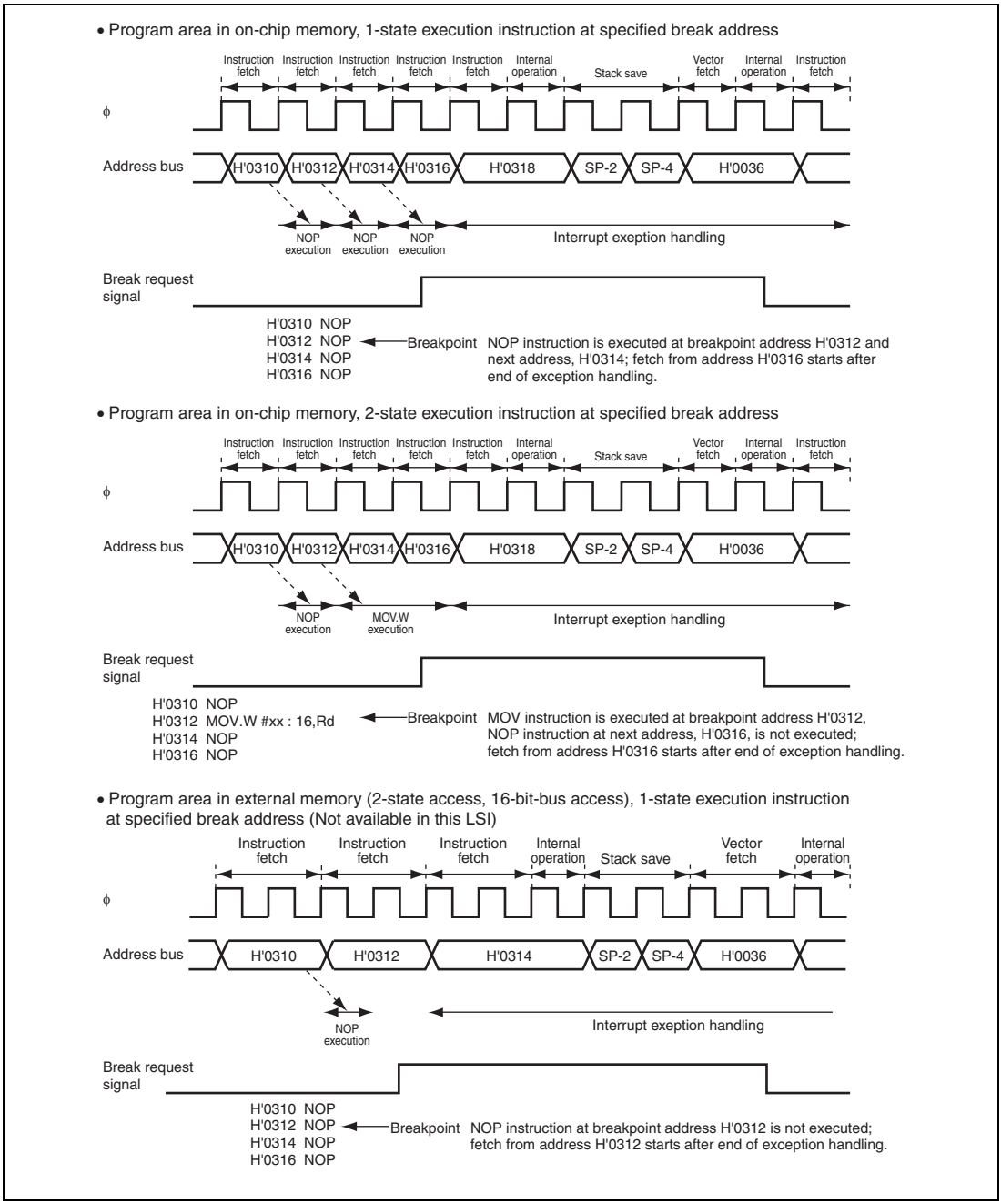


Figure 5.12 Examples of Address Break Timing

5.8 Usage Notes

5.8.1 Conflict between Interrupt Generation and Disabling

When an interrupt enable bit is cleared to 0 to disable interrupt requests, the disabling becomes effective after execution of the instruction. When an interrupt enable bit is cleared to 0 by an instruction such as BCLR or MOV, and if an interrupt is generated during execution of the instruction, the interrupt concerned will still be enabled on completion of the instruction, so interrupt exception handling for that interrupt will be executed on completion of the instruction. However, if there is an interrupt request of higher priority than that interrupt, interrupt exception handling will be executed for the higher-priority interrupt, and the lower-priority interrupt will be ignored. The same rule is also applied when an interrupt source flag is cleared to 0. Figure 5.13 shows an example where the CMIEA bit in TCR of the TMR is cleared to 0. The above conflict will not occur if an interrupt enable bit or interrupt source flag is cleared to 0 while the interrupt is disabled.

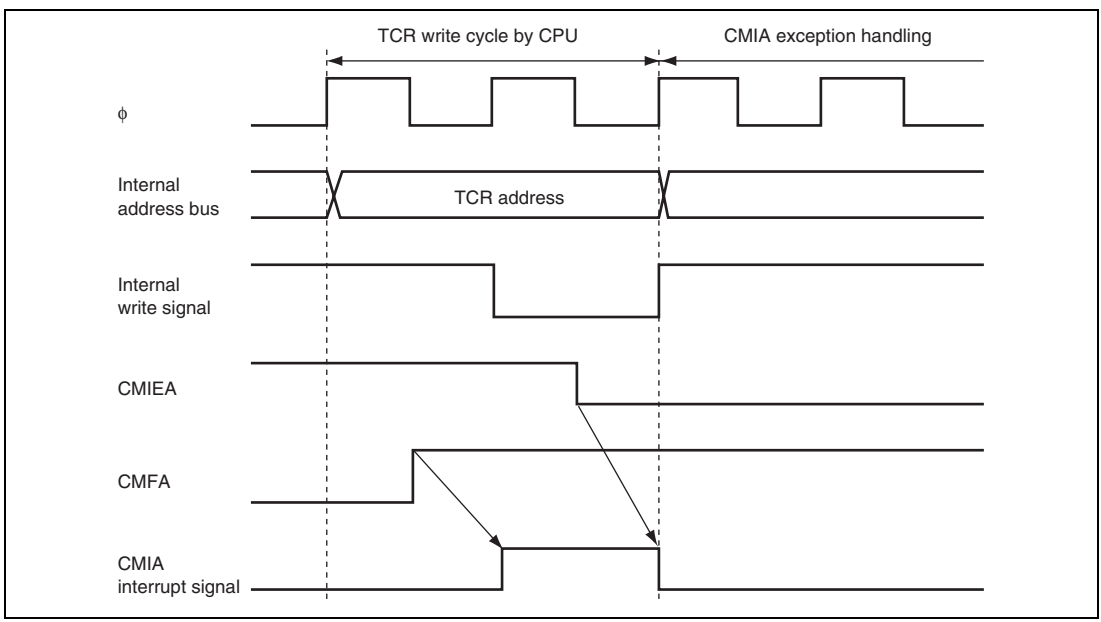


Figure 5.13 Conflict between Interrupt Generation and Disabling

5.8.2 Instructions for Disabling Interrupts

The instructions that disable interrupts are LDC, ANDC, ORC, and XORC. After any of these instructions are executed, all interrupts including NMI are disabled and the next instruction is always executed. When the I bit or UI bit is set by one of these instructions, the new value becomes valid two states after execution of the instruction ends.

5.8.3 Interrupts during Execution of EEPMOV Instruction

Interrupt operation differs between the EEPMOV.B instruction and the EEPMOV.W instruction.

With the EEPMOV.B instruction, an interrupt request including NMI issued during data transfer is not accepted until data transfer is completed.

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

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

5.8.4 Vector Address Switching

Switching between H8S/2140B Group compatible vector mode and extended vector mode must be done in a state with no interrupts occurring.

If the EIVS bit in SYSCR3 is changed from 0 to 1 when interrupt input is enabled because the $\overline{\text{KIN15}}$ to $\overline{\text{KIN0}}$ and $\overline{\text{WUE15}}$ to $\overline{\text{WUE8}}$ pins are set at low level, a falling edge is detected, thus causing an interrupt to be generated. The vector mode must be changed when interrupt input is disabled, that is the $\overline{\text{KIN15}}$ to $\overline{\text{KIN0}}$ and $\overline{\text{WUE15}}$ to $\overline{\text{WUE8}}$ pins are set at high level.

5.8.5 External Interrupt Pin in Software Standby Mode and Watch Mode

- When the pins ($\overline{\text{IRQ15}}$ to $\overline{\text{IRQ0}}$, $\overline{\text{ExIRQ15}}$ to $\overline{\text{ExIRQ6}}$, $\overline{\text{KIN15}}$ to $\overline{\text{KIN0}}$, and $\overline{\text{WUE15}}$ to $\overline{\text{WUE8}}$) are used as external input pins in software standby mode or watch mode, the pins should not be left floating.
- When the external interrupt pins ($\overline{\text{IRQ7}}$, $\overline{\text{IRQ6}}$, $\overline{\text{ExIRQ15}}$ to $\overline{\text{ExIRQ8}}$, $\overline{\text{KIN7}}$ to $\overline{\text{KIN0}}$, and $\overline{\text{WUE15}}$ to $\overline{\text{WUE8}}$) are used in software standby and watch modes, the noise canceller should be disabled.

5.8.6 Noise Canceller Switching

The noise canceller should be switched when the external input pins ($\overline{\text{IRQ7}}$, $\overline{\text{IRQ6}}$, $\overline{\text{ExIRQ15}}$ to $\overline{\text{ExIRQ8}}$, $\overline{\text{KIN7}}$ to $\overline{\text{KIN0}}$, and $\overline{\text{WUE15}}$ to $\overline{\text{WUE8}}$) are high.

5.8.7 IRQ Status Register (ISR)

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

Section 6 Bus Controller (BSC)

Since this LSI does not have an externally extended function, it does not have an on-chip bus controller (BSC). Considering the software compatibility with similar products, you must be careful to set appropriate values to the control registers for the bus controller.

6.1 Register Descriptions

The bus controller has the following registers.

Table 6.1 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Data Bus Width
Bus control register	BCR	R/W	H'D3	H'FFC6	8
Wait state control register	WSCR	R/W	H'F3	H'FFC7	8

6.1.1 Bus Control Register (BCR)

Bit	Bit Name	Initial Value	R/W	Description
7	—	1	R/W	Reserved The initial value should not be changed.
6	ICIS0	1	R/W	Idle Cycle Insertion The initial value should not be changed.
5	BRSTRM	0	R/W	Burst ROM Enable The initial value should not be changed.
4	BRSTS1	1	R/W	Burst Cycle Select 1 The initial value should not be changed.
3	BRSTS0	0	R/W	Burst Cycle Select 0 The initial value should not be changed.
2	—	0	R/W	Reserved The initial value should not be changed.
1	IOS1	1	R/W	IOS Select 1 and 0
0	IOS0	1	R/W	The initial value should not be changed.

6.1.2 Wait State Control Register (WSCR)

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 1	R/W	Reserved The initial value should not be changed.
5	ABW	1	R/W	Bus Width Control The initial value should not be changed.
4	AST	1	R/W	Access State Control The initial value should not be changed.
3	WMS1	0	R/W	Wait Mode Select 1 and 0
2	WMS0	0	R/W	The initial value should not be changed.
1	WC1	1	R/W	Wait Count 1 and 0
0	WC0	1	R/W	The initial value should not be changed.

Section 7 I/O Ports

Table 7.1 lists the port functions. The pins of each port also have other functions such as input/output pins of on-chip peripheral modules or interrupt input pins. Each I/O port includes a data direction register (DDR) that controls input/output, a data register (DR) that stores output data, and a port input data register (PIN) used to read the pin states. Port E does not have a DR or a DDR register.

Ports 1 to 3, 6, 9, B to D, F, H, and J have internal input pull-up MOSs and a pull-up MOS control register (PCR) that controls the on/off state of the input pull-up MOSs.

In addition, ports 1 to 3, C, and D can drive a LED (5 mA sink current). P52, P97, P86, P42, ports A, G, and I are NMOS push-pull outputs and 5-V tolerant inputs. PE4 and PE2 to PE0 are 5-V tolerant inputs.

Ports I and J are not supported by TFP-144V and TLP-145V.

Table 7.1 Port Functions

Port	Description	Bit	Function			Input Pull-up MOS Function	LED Drive Capability (5 mA Sink Current)	On-Chip Noise Canceler
			I/O	Input	Output			
Port 1	General I/O port	7	P17	—	—	O	O	—
		6	P16	—	—			
		5	P15	—	—			
		4	P14	—	—			
		3	P13	—	—			
		2	P12	—	—			
		1	P11	—	—			
		0	P10	—	—			
Port 2	General I/O port	7	P27	—	—	O	O	—
		6	P26	—	—			
		5	P25	—	—			
		4	P24	—	—			
		3	P23	—	—			
		2	P22	—	—			
		1	P21	—	—			
		0	P20	—	—			
Port 3	General I/O port also functioning as LPC input/output	7	P37/SERIRQ	—	—	O	O	—
		6	P36	LCLK	—			
		5	P35	$\overline{\text{LRESET}}$	—			
		4	P34	$\overline{\text{LFRAME}}$	—			
		3	P33/LAD3	—	—			
		2	P32/LAD2	—	—			
		1	P31/LAD1	—	—			
		0	P30/LAD0	—	—			

Port	Description	Bit	Function			Input Pull-up MOS Function	LED Drive Capability (5 mA Sink Current)	On-Chip Noise Canceler
			I/O	Input	Output			
Port 4	General I/O port also functioning as PWMX and PWMU_B outputs, TCM input, and TMR_0, TMR_1, IIC_1, and SCI_2 inputs/outputs	7	P47	TCMCKI3/ TCMMCI3	PWX1/PWMU5B	—	—	—
		6	P46	TCMCYI3	PWX0/PWMU4B			
		5	P45	TCMCKI2/ TCMMCI2	PWMU3B			
		4	P44	TCMCYI2	TMO1/PWMU2B			
		3	P43/SCK2	TMI1/TCMCKI1/ TCMMCI1	—			
		2	P42/SDA1	TCMCYI1				
		1	P41	RxD2/TCMCKI0/ TCMMCI0	TMO0			
		0	P40	TMI0/TCMCYI0	TxD2			
Port 5	General I/O port also functioning as IIC_0 and SCIF inputs/outputs	2	P52/SCL0	—	—	—	—	—
		1	P51	FRxD	—			
		0	P50	—	FTxD			
Port 6	General I/O port also functioning as interrupt input and keyboard input	7	P67	$\overline{\text{KIN7}}$/IRQ7	—	O	—	O
		6	P66	$\overline{\text{KIN6}}$/IRQ6	—			
		5	P65	$\overline{\text{KIN5}}$	—			
		4	P64	$\overline{\text{KIN4}}$	—			
		3	P63	$\overline{\text{KIN3}}$	—			
		2	P62	$\overline{\text{KIN2}}$	—			
		1	P61	$\overline{\text{KIN1}}$	—			
		0	P60	$\overline{\text{KIN0}}$	—			

Port	Description	Bit	Function			Input Pull-up MOS Function	LED Drive Capability (5 mA Sink Current)	On-Chip Noise Canceler
			I/O	Input	Output			
Port 7	General input port also functioning as A/D converter analog input	7	—	P77/AN7	—	—	—	—
		6	—	P76/AN6	—			
		5	—	P75/AN5	—			
		4	—	P74/AN4	—			
		3	—	P73/AN3	—			
		2	—	P72/AN2	—			
		1	—	P71/AN1	—			
		0	—	P70/AN0	—			
Port 8	General I/O port also functioning as interrupt input, IIC_1, SCI_1, and LPC inputs/outputs	6	P86/SCK1/SCL1	$\overline{\text{IRQ5}}$	—	—	—	—
		5	P85	RxD1/$\overline{\text{IRQ4}}$	—			
		4	P84	$\overline{\text{IRQ3}}$	TxD1			
		3	P83	$\overline{\text{LPCPD}}$	—			
		2	P82/$\overline{\text{CLKRUN}}$	—	—			
		1	P81/GA20	—	—			
		0	P80/$\overline{\text{PME}}$	—	—			
Port 9	General I/O port also functioning as external sub-clock, interrupt input, IIC_0 input/output, and system clock output	7	P97/SDA0	$\overline{\text{IRQ15}}$	—	O (P95 to P90)	—	—
		6	P96	EXCL	ϕ			
		5	P95	$\overline{\text{IRQ14}}$	—			
		4	P94	$\overline{\text{IRQ13}}$	—			
		3	P93	$\overline{\text{IRQ12}}$	—			
		2	P92	$\overline{\text{IRQ0}}$	—			
		1	P91	$\overline{\text{IRQ1}}$	—			
		0	P90	$\overline{\text{IRQ2}}$	—			

Port	Description	Bit	Function			Input Pull-up MOS Function	LED Drive Capability (5 mA Sink Current)	On-Chip Noise Canceler
			I/O	Input	Output			
Port A	General I/O port also functioning as keyboard input and PS2 input/output	7	PA7/PS2CD	$\overline{KIN15}$	—	—	—	—
		6	PA6/PS2CC	$\overline{KIN14}$	—			
		5	PA5/PS2BD	$\overline{KIN13}$	—			
		4	PA4/PS2BC	$\overline{KIN12}$	—			
		3	PA3/PS2AD	$\overline{KIN11}$	—			
		2	PA2/PS2AC	$\overline{KIN10}$	—			
		1	PA1/PS2DD	$\overline{KIN9}$	—			
		0	PA0/PS2DC	$\overline{KIN8}$	—			
Port B	General I/O port also functioning as LPC, SCIF and FSI inputs/outputs and PWMU output	7	PB7	—	$\overline{RTS}$/FSISS	O	—	—
		6	PB6	$\overline{CTS}$	FSICK			
		5	PB5	FSIDI	$\overline{DTR}$			
		4	PB4	$\overline{DSR}$	FSIDO			
		3	PB3	$\overline{DCD}$	PWMU1B			
		2	PB2	$\overline{RI}$	PWMU0B			
		1	PB1/LSCI	—	—			
		0	PB0/$\overline{LSMI}$	—	—			
Port C	General I/O port also functioning as wake-up input and TPU input/output	7	PC7/TIOCB2	TCLKD/$\overline{WUE15}$	—	O	O	O
		6	PC6/TIOCA2	$\overline{WUE14}$	—			
		5	PC5/TIOCB1	TCLKC/$\overline{WUE13}$	—			
		4	PC4/TIOCA1	$\overline{WUE12}$	—			
		3	PC3/TIOCD0	TCLKB/$\overline{WUE11}$	—			
		2	PC2/TIOCC0	TCLKA/$\overline{WUE10}$	—			
		1	PC1/TIOCB0	$\overline{WUE9}$	—			
		0	PC0/TIOCA0	$\overline{WUE8}$	—			

Port	Description	Bit	Function			Input Pull-up MOS Function	LED Drive Capability (5 mA Sink Current)	On-Chip Noise Canceler
			I/O	Input	Output			
Port D	General I/O port also functioning as A/D converter analog input	7	PD7	AN15	—	O	O	—
		6	PD6	AN14	—			
		5	PD5	AN13	—			
		4	PD4	AN12	—			
		3	PD3	AN11	—			
		2	PD2	AN10	—			
		1	PD1	AN9	—			
		0	PD0	AN8	—			
Port E	General input port also functioning as external sub-clock input, emulator input/output	4	—	PE4*[1]/ETMS	—	—	—	—
		3	—	PE3*[1]	ETDO			
		2	—	PE2*[1]/ETDI	—			
		1	—	PE1*[1] /ETCK	—			
		0	—	PE0/ExEXCL	—			
Port F	General I/O port also functioning as interrupt and TDP inputs, TMR_X, TMR_Y, and PWM outputs	7	PF7	—	PWMU5A	O	—	—
		6	PF6	—	PWMU4A			
		5	PF5	—	PWMU3A			
		4	PF4	—	PWMU2A			
		3	PF3	TDPCKI0/ TDPMCI0/$\overline{\text{IRQ11}}$	TMOX			
		2	PF2	TDPCYI0//$\overline{\text{IRQ10}}$	TMOY			
		1	PF1	$\overline{\text{IRQ9}}$	PWMU1A			
		0	PF0	$\overline{\text{IRQ8}}$	PWMU0A			

Port	Description	Bit	Function			Input Pull-up MOS Function	LED Drive Capability (5 mA Sink Current)	On-Chip Noise Canceler
			I/O	Input	Output			
Port G	General I/O port also functioning as interrupt and TDP inputs, TMR_X and TMR_Y inputs, and IIC0 to IIC2 inputs/outputs	7	PG7/ExSCLB	$\overline{\text{ExIRQ15}}$	—	—	—	O
		6	PG6/ExSDAB	$\overline{\text{ExIRQ14}}$	—			
		5	PG5/ExSCLA	$\overline{\text{ExIRQ13}}$	—			
		4	PG4/ExSDAA	$\overline{\text{ExIRQ12}}$	—			
		3	PG3/SCL2	$\overline{\text{ExIRQ11}}$	—			
		2	PG2/SDA2	$\overline{\text{ExIRQ10}}$	—			
		1	PG1	TMIY1/ TDPCKI1/TDPMCI1/ $\overline{\text{ExIRQ9}}$	—			
		0	PG0	TMIX/TDPCYI1 $\overline{\text{ExIRQ8}}$	—			
Port H	General I/O port also functioning as interrupt and TDP and CIR inputs	5	PH5	—	—	O	—	—
		4	PH4	—	—			
		3	PH3	—	—			
		2	PH2	CIRI	—			
		1	PH1	TDPCKI2/ TDPMCI1/ $\overline{\text{ExIRQ7}}$	—			
		0	PH0	TDPCYI2/ $\overline{\text{ExIRQ6}}$	—			
Port I	General I/O port	7	PI7*2	—	—	—	—	—
		6	PI6*2	—	—			
		5	PI5*2	—	—			
		4	PI4*2	—	—			
		3	PI3*2	—	—			
		2	PI2*2	—	—			
		1	PI1*2	—	—			
		0	PI0*2	—	—			

Port	Description	Bit	Function			Input Pull-up MOS Function	LED Drive Capability (5 mA Sink Current)	On-Chip Noise Canceler
			I/O	Input	Output			
Port J	General I/O port	7	PJ7*[2]	—	—	O	—	—
		6	PJ6*[2]	—	—			
		5	PJ5*[2]	—	—			
		4	PJ4*[2]	—	—			
		3	PJ3*[2]	—	—			
		2	PJ2*[2]	—	—			
		1	PJ1*[2]	—	—			
		0	PJ0*[2]	—	—			

Notes: 1. Not supported by the system development tool (emulator).
2. Not supported by TFP-144V and TLP-145V.

7.1 Register Descriptions

Table 7.2 lists each port registers.

Table 7.2 Register Configuration in Each Port

Port	Number of Pins	Registers									
		DDR	DR	PIN	PCR	KMPCR	ODR	NCE	NCMC	NCCS	NOCR
Port 1	8	O	O	O*[2]	O	—	—	—	—	—	—
Port 2	8	O	O	O*[2]	O	—	—	—	—	—	—
Port 3	8	O	O	O*[2]	O	—	—	—	—	—	—
Port 4	8	O	O	O*[2]	—	—	—	—	—	—	—
Port 5	3	O	O	O*[2]	—	—	—	—	—	—	—
Port 6	8	O	O	O*[2]	—	O	—	O	O	O	—
Port 7	8	—	—	O	—	—	—	—	—	—	—
Port 8	7	O	O	O*[2]	—	—	—	—	—	—	—
Port 9	8	O	O	O*[2]	O	—	—	—	—	—	—
Port A	8	O	—	O	—	—	O	—	—	—	—
Port B	8	O	—	O	O*[2]	—	O	—	—	—	—
Port C	8	O	—	O	O*[2]	—	O	O	O	O	O
Port D	8	O	—	O	O*[2]	—	O	—	—	—	O
Port E	5	—	—	O	—	—	—	—	—	—	—
Port F	8	O	—	O	O*[2]	—	O	—	—	—	O
Port G	8	O	—	O	—	—	O	O	O	O	O
Port H	6	O	—	O	O*[2]	—	O	—	—	—	O
Port I	8*[1]	O	—	O	—	—	O	—	—	—	O
Port J	8*[1]	O	—	O	O	—	O	—	—	—	O

[Legend]

O: Register exists

—: No register exists

Notes: 1. Not supported by TFP-144V and TLP-145V.

2. Valid only when the PORTS bit in the port control register 2 (PTCNT2) is 1.

7.1.1 Data Direction Register (PnDDR) (n = 1 to 6, 8, 9, A to D, and F to J)

DDR specifies the port input or output for each bit.

The upper five bits in P5DDR, the upper one bit in P8DDR, and the upper two bits in PHDDR are reserved.

(1) PORTS = 0

Bit	Bit Name	Initial Value	R/W	Description
7	Pn7DDR	0	W	The corresponding pins act as output ports when these bits are set to 1 and act as input ports when cleared to 0. Note: These bits cannot be set with bit manipulation instructions such as BSET and BCLR.
6	Pn6DDR	0	W	
5	Pn5DDR	0	W	
4	Pn4DDR	0	W	
3	Pn3DDR	0	W	
2	Pn2DDR	0	W	
1	Pn1DDR	0	W	
0	Pn0DDR	0	W	

(2) PORTS = 1

Bit	Bit Name	Initial Value	R/W	Description
7	Pn7DDR	0	R/W	The corresponding pins act as output ports when these bits are set to 1 and act as input ports when cleared to 0.
6	Pn6DDR	0	R/W	
5	Pn5DDR	0	R/W	
4	Pn4DDR	0	R/W	
3	Pn3DDR	0	R/W	
2	Pn2DDR	0	R/W	
1	Pn1DDR	0	R/W	
0	Pn0DDR	0	R/W	

7.1.2 Data Register (PnDR) (n = 1 to 6, 8, and 9)

DR is a register that stores output data of the pins to be used as the general output port. Since the P96DR bit is determined by the state of the P96 pin, the initial value is undefined. The upper five bits in P5DR and the upper one bit in P8DR are reserved.

Bit	Bit Name	Initial Value	R/W	Description
7	Pn7DR	0	R/W	PnDR stores output data for the pins that are used as the general output port. When the PORTS bit in PTCNT2 is 0, reading this register reads out the current settings of these bits for pins corresponding to PnDDR bits set to 1 and reads out the states of pins corresponding to PnDDR bits cleared to 0.
6	Pn6DR	0	R/W	
5	Pn5DR	0	R/W	
4	Pn4DR	0	R/W	
3	Pn3DR	0	R/W	
2	Pn2DR	0	R/W	
1	Pn1DR	0	R/W	
0	Pn0DR	0	R/W	

7.1.3 Input Data Register (PnPIN) (n = 1 to 9 and A to J)

PIN is an 8-bit read-only register that reflects the port pin state. A write to PIN is invalid. The upper five bits in P5PIN, the upper one bit in P8PIN, the upper three bits in PEPIN, and the upper two bits in PHPIN are reserved.

Bits P1PIN to P9PIN are valid only when PORTS in PTCNT2 is 1.

Bit	Bit Name	Initial Value	R/W	Description
7	Pn7PIN	Undefined*	R	When this register is read, the pin states are returned.
6	Pn6PIN	Undefined*	R	
5	Pn5PIN	Undefined*	R	
4	Pn4PIN	Undefined*	R	
3	Pn3PIN	Undefined*	R	
2	Pn2PIN	Undefined*	R	
1	Pn1PIN	Undefined*	R	
0	Pn0PIN	Undefined*	R	

Note: * The initial values of these pins are determined in accordance with the states of pins Pn7 to Pn0.

7.1.4 Pull-Up MOS Control Register (PnPCR) (n = 1 to 3, 9, B to D, F, H, and J) Pull-Up MOS Control Register (KMPCR) (Port 6)

PCR is a register that controls on/off of the port input pull-up MOS.

If a bit in PCR is set to 1 while the pin is in the input state, the input pull-up MOS corresponding to the bit in PCR is turned on. Table 7.3 shows the input pull-up MOS state. The upper two bits in P9PCR and the upper two bits in PHPCR are reserved.

PBPCR to PDPCR, PFPCR, and PHPCR are valid only when PORTS in PTCNT2 is 1.

Bit	Bit Name	Initial Value	R/W	Description
7	Pn7PCR	0	R/W	For pins in the input state corresponding to bits in this register that have been set to 1, the input pull-up MOSs are turned on.
6	Pn6PCR	0	R/W	
5	Pn5PCR	0	R/W	
4	Pn4PCR	0	R/W	
3	Pn3PCR	0	R/W	
2	Pn2PCR	0	R/W	
1	Pn1PCR	0	R/W	
0	Pn0PCR	0	R/W	

Table 7.3 Input Pull-Up MOS State (1)

- Port 1 to 3, 6, 9, and J

Port	Pin State	Reset	Software Standby Mode	Other Operation
Port 1	Port output	Off		
	Port input	Off	On/Off	
Port 2	Port output	Off		
	Port input	Off	On/Off	
Port 3	Port output	Off		
	Port input	Off	On/Off	
Port 6 (KMPCR)	Port output	Off		
	Port input	Off	On/Off	
Port 9	Port output	Off		
	Port input	Off	On/Off	
Port J	Port output	Off		
	Port input	Off	On/Off	

[Legend]

Off: The input pull-up MOS is always off.

On/Off: On when PnDDR = 0 and PnPCR = 1; otherwise off.

Table 7.3 Input Pull-Up MOS State (2)

- Port B to D, F, and H

Port	Pin State	Reset	Software Standby Mode	Other Operation
Port B	Port output	Off		
	Port input	Off	On/Off	
Port C	Port output	Off		
	Port input	Off	On/Off	
Port D	Port output	Off		
	Port input	Off	On/Off	
Port F	Port output	Off		
	Port input	Off	On/Off	
Port H	Port output	Off		
	Port input	Off	On/Off	

[Legend]

Off: The input pull-up MOS is always off.

On/Off: On when the pin is in the input state, PnDDR = 0, and PnODR = 1; otherwise off (when PORTS = 0).
On when the pin is in the input state, PnDDR = 0, and PnPCR = 1; otherwise off (when PORTS = 1).

7.1.5 Output Data Register (PnODR) (n = A to D and F to J)

ODR is a register that stores output data for ports. The upper two bits in PHODR are reserved.

Bit	Bit Name	Initial Value	R/W	Description
7	Pn7ODR	0	R/W	ODR stores the output data for the pins that are used as the general output port.
6	Pn6ODR	0	R/W	
5	Pn5ODR	0	R/W	
4	Pn4ODR	0	R/W	
3	Pn3ODR	0	R/W	
2	Pn2ODR	0	R/W	
1	Pn1ODR	0	R/W	
0	Pn0ODR	0	R/W	

7.1.6 Noise Canceler Enable Register (PnNCE) (n = 6, C, and G)

NCE enables or disables the noise cancel circuit at port n pins in bit units.

Bit	Bit Name	Initial Value	R/W	Description
7	Pn7NCE	0	R/W	Noise cancel circuit is enabled when a bit in this register is set to 1, and the pin setting state is fetched in P6DR or PnPIN in the sampling cycle set by the PnNCCS.
6	Pn6NCE	0	R/W	
5	Pn5NCE	0	R/W	
4	Pn4NCE	0	R/W	
3	Pn3NCE	0	R/W	
2	Pn2NCE	0	R/W	
1	Pn1NCE	0	R/W	
0	Pn0NCE	0	R/W	

7.1.7 Noise Canceler Decision Control Register (PnNCMC) (n = 6, C, and G)

NCMC controls whether 1 or 0 is expected for the input signal to port n pins in bit units.

Bit	Bit Name	Initial Value	R/W	Description
7	Pn7NCMC	0	R/W	1 expected: 1 is stored in the port data register when 1 is input stably. 0 expected: 0 is stored in the port data register when 0 is input stably.
6	Pn6NCMC	0	R/W	
5	Pn5NCMC	0	R/W	
4	Pn4NCMC	0	R/W	
3	Pn3NCMC	0	R/W	
2	Pn2NCMC	0	R/W	
1	Pn1NCMC	0	R/W	
0	Pn0NCMC	0	R/W	

7.1.8 Noise Cancel Cycle Setting Register (PnNCCS) (n = 6, C, and G)

NCCS controls the sampling cycles of the noise canceler.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	Undefined	R/W	Reserved The read value is undefined. The write value should always be 0.
2	PnNCCK2	0	R/W	These bits set the sampling cycles of the noise canceler. When ϕ is 10 MHz 000: 0.80 µs ϕ/2 001: 12.8 µs ϕ/32 010: 3.3 ms ϕ/8192 011: 6.6 ms ϕ/16384 100: 13.1 ms ϕ/32768 101: 26.2 ms ϕ/65536 110: 52.4 ms ϕ/131072 111: 104.9 ms ϕ/262144
1	PnNCCK1	0	R/W	
0	PnNCCK0	0	R/W	

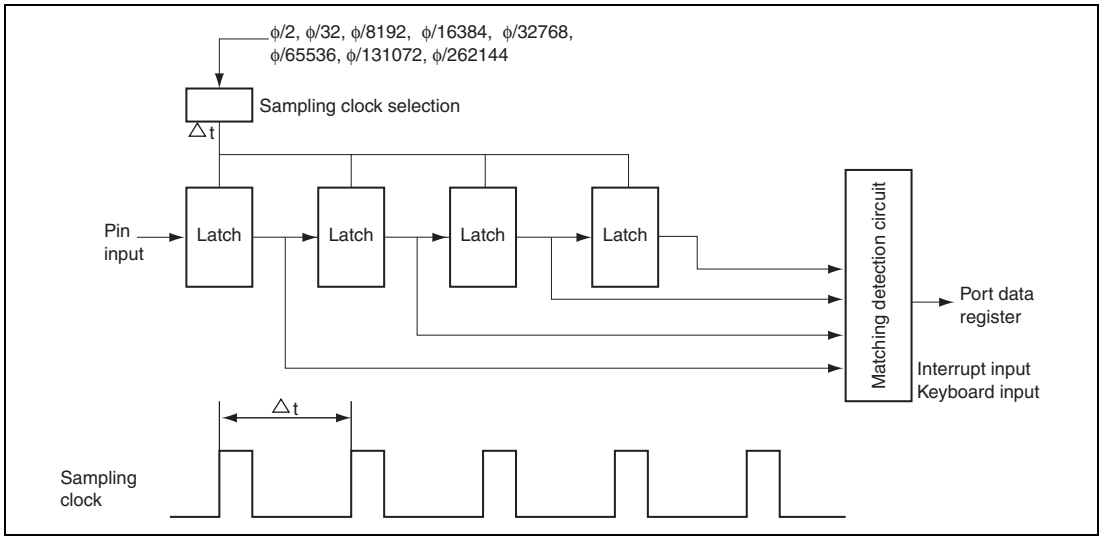


Figure 7.1 Noise Cancel Circuit

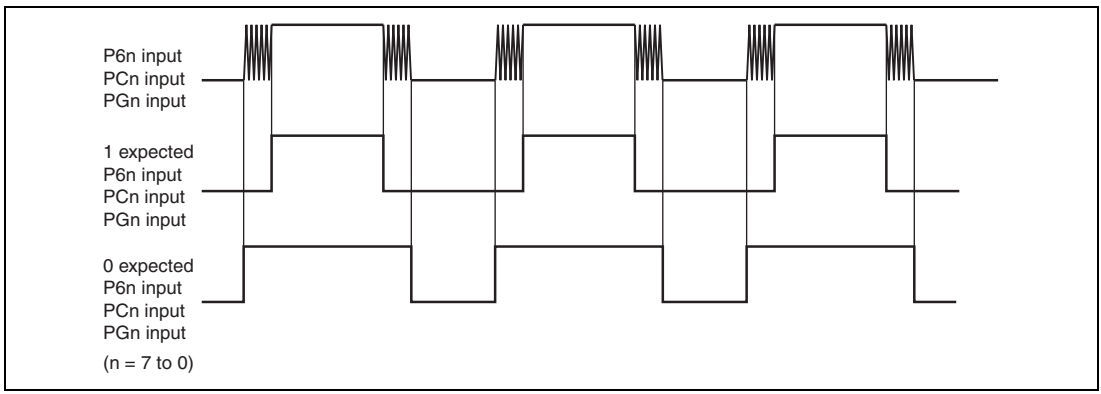


Figure 7.2 Schematic View of Noise Cancel Operation

7.1.9 Port Nch-OD Control Register (PnNOCR) (n = C, D, and F to J)

The individual bits of NOCR specify output driver type for the pins of port n that is specified as output. The upper two bits in PHNOCR are reserved.

Bit	Bit Name	Initial Value	R/W	Description
7	Pn7NOCR	0	R/W	0: CMOS (Ports G and I are NMOS push-pull output) (High level driver is enabled) 1: N channel open-drain (High level driver is disabled)
6	Pn6NOCR	0	R/W	
5	Pn5NOCR	0	R/W	
4	Pn4NOCR	0	R/W	
3	Pn3NOCR	0	R/W	
2	Pn2NOCR	0	R/W	
1	Pn1NOCR	0	R/W	
0	Pn0NOCR	0	R/W	

7.1.10 Pin Functions

The pin function is switched according to the setting of the PORTS bit in PTCNT2. (Ports B to D, F, and H)

(1) PORTS = 0

DDR	0		1			
NOCR	—		0		1	
ODR	0	1	0	1	0	1
N-ch. driver	Off		On	Off	On	Off
P-ch. driver	Off		Off	On	Off	
Input pull-up MOS	Off	On	Off			
Pin function	Input pin		Output pin			

(2) PORTS = 1

DDR	0		1			
NOCR	—		0		1	
ODR	—		0	1	0	1
PCR	0	1	—			
N-ch. Driver	Off		On	Off	On	Off
P-ch. Driver	Off		Off	On	Off	
Input pull-up MOS	Off	On	Off			
Pin Function	Input pin		Output pin			

7.2 Output Buffer Control

This section describes the output priority of each pin.

The name of each peripheral module pin is followed by "_OE". This (for example: TIOCA4_OE) indicates whether the output of the corresponding function is valid (1) or if another setting is specified (0). Table 7.4 lists each port output signal's valid setting. For details on the corresponding output signals, see the register description of each peripheral module.

7.2.1 Port 1

(1) P17 to P10

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

		Setting
		I/O Port
Module Name	Pin Function	P1nDDR
I/O port	P1n output	1
	P1n input (initial setting)	0

(n = 7 to 0)

7.2.2 Port 2

(1) P27 to P20

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

		Setting
		I/O Port
Module Name	Pin Function	P2nDDR
I/O port	P2n output	1
	P2n input (initial setting)	0

(n = 7 to 0)

7.2.3 Port 3

(1) P37/SERIRQ, P36/LCLK, P35/$\overline{\text{LRESET}}$, P34/$\overline{\text{LFRAME}}$, P33/LAD3, P32/LAD2, P31/LAD1, P30/LAD0

The pin function is switched as shown below according to the combination of the FSILIE bit in SLCR of FSI, the SCIFE bit in HICR5 and the LPC4E bit in HICR4 and LPC3E to LPC1E bits in HICR0 of the LPC, and the P3nDDR bit. LPCENABLE in the following table is expressed by the following logical expression.

LPCENABLE = 1: FSILIE + SCIFE + LPC4E + LPC3E + LPC2E + LPC1E

		Setting	
		Logical expression	I/O Port
Module Name	Pin Function	LPCENABLE	P3nDDR
LPC	LPC output	1	—
I/O port	P3n output	0	1
	P3n input (initial setting)	0	0

(n = 7 to 0)

7.2.4 Port 4

(1) P47/PWX1/PWMU5B/TCMCKI3/TCMMCI3

The pin function is switched as shown below according to the combination of the PWMX and PWMU and the P47DDR bit.

		Setting		
		PWMX	PWMU	I/O Port
Module Name	Pin Function	PWX1_OE	PWMU5B_OE	P47DDR
PWMX	PWX1 output	1	—	—
PWMU	PWMU5B output	0	1	1
I/O port	P47 output	0	0	1
	P47 input (initial setting)	0	—	0

(2) P46/PWX0/PWMU4B/TCMCYI3

The pin function is switched as shown below according to the combination of the PWMX and PWMU and the P46DDR bit.

		Setting		
Module Name	Pin Function	PWMX	PWMU	I/O Port
		PWX0_OE	PWMU4B_OE	P46DDR
PWMX	PWX0 output	1	—	—
PWMU	PWMU4B output	0	1	1
I/O port	P46 output	0	0	1
	P46 input (initial setting)	0	—	0

(3) P45/PWMU3B/TCMCKI2/TCMMCI2

The pin function is switched as shown below according to the combination of the PWMX and the P45DDR bit.

		Setting	
Module Name	Pin Function	PWMU	I/O Port
		PWMU3B_OE	P45DDR
PWMU	PWMU3B output	1	1
I/O port	P45 output	0	1
	P45 input (initial setting)	—	0

(4) P44/TMO1/PWMU2B/TCMCYI2

The pin function is switched as shown below according to the combination of the TMR and PWMU and the P44DDR bit.

		Setting		
		TMR	PWMU	I/O Port
Module Name	Pin Function	TMO1_OE	PWMU2B_OE	P44DDR
TMR	TMO1 output	1	—	—
PWMU	PWMU2B output	0	1	1
I/O port	P44 output	0	0	1
	P44 input (initial setting)	0	—	0

(5) P43/TMI1/SCK2/TCMCKI1/TCMMCI1

The pin function is switched as shown below according to the combination of the SCI and the P43DDR bit.

		Setting	
		SCI	I/O Port
Module Name	Pin Function	SCK2_OE	P43DDR
SCI	SCK2 input/output	1	—
I/O port	P43 output	0	1
	P43 input (initial setting)	0	0

(6) P42/SDA1/TCMCYI1

The pin function is switched as shown below according to the combination of the IIC1AS and IIC1BS bits in PTCNT1, ICE bit in ICCR of IIC_1, and the P42DDR bit. When the TCMIPE bit in TCMIER_1 of TCM_1 is set to 1, TCMCYI1 functions as an input pin.

		Setting	
		IIC_1	I/O Port
Module Name	Pin Function	SDA1_OE	P42DDR
IIC	SDA1 output	1	—
I/O port	P42 output	0	1
	P42 input (initial setting)	0	0

Note: To use this pin as SDA1, clear the IIC1AS and IIC1BS bits in PTCNT1 to 0. The output format for SDA1 is NMOS output only and direct bus drive is possible. When this pin is used as the P42 output pin, the output format is NMOS push-pull.

(7) P41/TMO0/RxD2/TCMCKI0/TCMMCI0

The pin function is switched as shown below according to the combination of the TMR and the P41DDR bit.

		Setting		
		TMR	SCI	I/O Port
Module Name	Pin Function	TMO0_OE	RE	P41DDR
TMR	TMO0 output	1	0	—
SCI	RxD2 input	0	1	—
I/O port	P41 output	0	0	1
	P41 input (initial setting)	0	0	0

Note: To use this pin as TMO0 output, clear the RE bit in SCR of the SCI2 to 0.

(8) P40/TMI0/TxD2/TCMCYI0

The pin function is switched as shown below according to the combination of the TMR and the SCI and the P40DDR bit.

		Setting	
		SCI	I/O Port
Module Name	Pin Function	TxD2_OE	P40DDR
SCI	TxD2 output	1	—
I/O port	P40 output	0	1
	P40 input (initial setting)	0	0

7.2.5 Port 5

(1) P52/SCL0

The pin function is switched as shown below according to the combination of the IIC0AS and IIC0BS bits in PTCNT1, ICE bit in ICCR of IIC_0, and the P52DDR bit.

		Setting	
		IIC_0	I/O Port
Module Name	Pin Function	SCL0_OE	P52DDR
IIC	SCL0 output	1	—
I/O port	P52 output	0	1
	P52 input (initial setting)	0	0

Note: To use this pin as SCL0, clear the IIC0AS and IIC0BS bits in PTCNT1 to 0. The output format for SCL0 is NMOS output only and direct bus drive is possible. When this pin is used as the P52 output pin, the output format is NMOS push-pull.

(2) P51/FRxD

The pin function is switched as shown below according to the SCIFOE1 bit in SCIFCR of the SCIF and the SCIFE bit in HICR5 of the SCIF, and the P51DDR bit.

SCIFENABLE = 1: SCIFOE1 + SCIFE

		Setting	
		Logical Expression	I/O Port
Module Name	Pin Function	SCIFENABLE	P51DDR
SCIF	FRxD input	1	—
I/O port	P51 output	0	1
	P51 input (initial setting)	0	0

(3) P50/FTxD

The pin function is switched as shown below according to the SCIFOE1 bit in SCIFCR of the SCIF, the SCIFE bit in HICR5, and the P50DDR bit.

SCIFENABLE = 1: SCIFOE1 + SCIFE

		Setting	
		Logical Expression	I/O Port
Module Name	Pin Function	SCIFENABLE	P50DDR
SCIF	FTxD output	1	—
I/O port	P50 output	0	1
	P50 input (initial setting)	0	0

7.2.6 Port 6

(1) P67/$\overline{\text{KIN7}}$/$\overline{\text{IRQ7}}$

When the KMIM7 bit in KMIMR of the interrupt controller is cleared to 0, this pin can be used as the $\overline{\text{KIN7}}$ input pin. When the ISS7 bit in ISSR is cleared to 0 and the IRQ7E bit in IER of the interrupt controller is set to 1, this pin can be used as the $\overline{\text{IRQ7}}$ input pin.

The pin function is switched as shown below according to the state of the P67DDR bit.

		Setting
		I/O Port
Module Name	Pin Function	P67DDR
I/O port	P67 output	1
	P67 input (initial setting)	0

(2) P66/$\overline{\text{KIN6}}$/$\overline{\text{IRQ6}}$

When the KMIM6 bit in KMIMR of the interrupt controller is cleared to 0, this pin can be used as the $\overline{\text{KIN6}}$ input pin. When the EIVS bit in SYSCR3 is cleared to 0 and the IRQ6E bit in IER of the interrupt controller is set to 1, this pin can be used as the $\overline{\text{IRQ6}}$ input pin.

The pin function is switched as shown below according to the state of the P66DDR bit.

		Setting
		I/O Port
Module Name	Pin Function	P66DDR
I/O port	P66 output	1
	P66 input (initial setting)	0

(3) P65/$\overline{\text{KIN5}}$, P64/$\overline{\text{KIN4}}$, P63/$\overline{\text{KIN3}}$, P62/$\overline{\text{KIN2}}$, P61/$\overline{\text{KIN1}}$, P60/$\overline{\text{KIN0}}$

When the KMIMn bit in KMIMR of the interrupt controller is cleared to 0, this pin can be used as the $\overline{\text{KIN}}$n input pin.

The pin function is switched as shown below according to the state of the P6nDDR bit.

		Setting
		I/O Port
Module Name	Pin Function	P6nDDR
I/O port	P6n output	1
	P6n input (initial setting)	0

(n = 5 to 0)

7.2.7 Port 7

(1) P77/AN7, P76/AN6, P75/AN5, P74/AN4, P73/AN3, P72/AN2, P71/AN1, P70/AN0

Module Name	Pin Function
A/D converter	ANn/P7n input

(n = 7 to 0)

7.2.8 Port 8

(1) P86/$\overline{\text{IRQ5}}$/SCK1/SCL1

The pin function is switched as shown below according to the combination of the C/$\overline{\text{A}}$ bit in SMR of SCI_1, CKE0 and CKE1 bits in SCR, IIC1AS and IIC1BS bits in PTCNT1, ICE bit in ICCR of IIC_1, and the P86DDR bit. When the ISS5 bit in ISSR is cleared to 0 and the IRQ5E bit in IER of the interrupt controller is set to 1, this pin can be used as the $\overline{\text{IRQ5}}$ input pin.

		Setting		
		SCI	IIC	I/O Port
Module Name	Pin Function	SCK1_OE	SCL1_OE	P86DDR
SCI	SCK1 input/output	1	0	—
IIC	SCL1 input/output	0	1	—
I/O port	P86 output	0	0	1
	P86 input (initial setting)	0	0	0

Note: To use this pin as SCL1 input/output, be sure that SCK1_OE is 0. To use this pin as SCL1, the IIC1AS and IIC1BS bits in PTCNT1 must be cleared to 0. The output format for SCL1 is NMOS output only and direct bus drive is possible. When this pin is used as the P86 output pin or SCK1 output pin, the output format is NMOS push-pull.

(2) P85/$\overline{\text{IRQ4}}$/RxD1

The pin function is switched as shown below according to the state of the P85DDR bit.

		Setting	
		SCI	I/O Port
Module Name	Pin Function	RE	P85DDR
SCI	RxD1 input	1	—
I/O port	P85 output	0	1
	P85 input (initial setting)	0	0

(3) P84/$\overline{\text{IRQ3}}$/TxD1

The pin function is switched as shown below according to the combination of the register setting of the SCI and the P84DDR bit.

Module Name	Pin Function	Setting: SCI — TxD1_OE	Setting: I/O Port — P84DDR
SCI	TxD1 output	1	—
I/O port	P84 output	0	1
	P84 input (initial setting)	0	0

(4) P83/$\overline{\text{LPCPD}}$

The pin function is switched as shown below according to the combination of the FSILIE bit in SLCR of FSI, the SCIFE bit in HICR5 and the LPC4E bit in HICR4 of the LPC, LPC3E to LPC1E bits in HICR0, and the P83DDR bit. LPCENABLE in the following table is expressed by the following logical expression.

LPCENABLE = 1 : FSILIE + SCIFE + LPC4E + LPC3E + LPC2E + LPC1E

Module Name	Pin Function	Setting: Logical Expression — LPCENABLE	Setting: I/O Port — P83DDR
LPC	$\overline{\text{LPCPD}}$ input	1	—
I/O port	P83 output	0	1
	P83 input (initial setting)	0	0

(5) P82/$\overline{\text{CLKRUN}}$

The pin function is switched as shown below according to the combination of the FSILIE bit in SLCR of FSI, the SCIFE bit in HICR5 and the LPC4E bit in HICR4 of the LPC, LPC3E to LPC1E bits in HICR0, and the P82DDR bit. LPCENABLE in the following table is expressed by the following logical expression.

LPCENABLE = 1 : FSILIE + SCIFE + LPC4E + LPC3E + LPC2E + LPC1E

		Setting	
		Logical Expression	I/O Port
Module Name	Pin Function	LPCENABLE	P82DDR
LPC	$\overline{\text{CLKRUN}}$ output	1	—
I/O port	P82 output	0	1
	P82 input (initial setting)	0	0

(6) P81/GA20

The pin function is switched as shown below according to the combination of the register setting of the LPC and the P81DDR bit.

		Setting	
		LPC	I/O Port
Module Name	Pin Function	GA20_OE	P81DDR
LPC	GA20 output	1	—
I/O port	P81 output	0	1
	P81 input (initial setting)	0	0

(7) P80/$\overline{\text{PME}}$

The pin function is switched as shown below according to the combination of the register setting of the LPC and the P80DDR bit.

		Setting	
		LPC	I/O Port
Module Name	Pin Function	$\overline{\text{PME}}$_OE	P80DDR
LPC	$\overline{\text{PME}}$ output	1	—
I/O port	P80 output	0	1
	P80 input (initial setting)	0	0

7.2.9 Port 9

(1) P97/$\overline{\text{IRQ15}}$/SDA0

The pin function is switched as shown below according to the combination of the IIC0AS and IIC0BS bits in PTCNT1, ICE bit in ICCR of IIC_0, and the P97DDR bit. When the ISS15 bit in ISSR16 is cleared to 0 and the IRQ15E bit in IER16 of the interrupt controller is set to 1, this pin can be used as the $\overline{\text{IRQ15}}$ input pin.

		Setting	
		IIC_0	I/O Port
Module Name	Pin Function	SDA0_OE	P97DDR
IIC	SDA0 input/output	1	—
I/O port	P97 output	0	1
	P97 input (initial setting)	0	0

Note: To use this pin as SDA0, clear the IIC0AS and IIC0BS bits in PTCNT1 to 0. The output format for SDA0 is NMOS output only and direct bus drive is possible. When this pin is used as the P97 output pin, the output format is NMOS push-pull.

(2) P96/ϕ/EXCL

The pin function is switched as shown below according to the combination of the register settings of the EXCLS bit in PTCNT0, EXCLE bit in LPWRCR, and the P96DDR bit.

Module Name	Pin Function	Setting I/O Port P96DDR
Clock	ϕ output	1
I/O port	P96 input (initial setting)	0

(3) P95/$\overline{\text{IRQ14}}$, P94/$\overline{\text{IRQ13}}$, P93/$\overline{\text{IRQ12}}$, P92/$\overline{\text{IRQ0}}$, P91/$\overline{\text{IRQ1}}$, P90/$\overline{\text{IRQ2}}$

The pin function is switched as shown below according to the state of the P9nDDR bit. When the ISSm bit in ISSR (ISSR16) is cleared to 0 and the IRQmE bit in IER (IER16) of the interrupt controller is set to 1, this pin can be used as the $\overline{\text{IRQm}}$ input pin.

Module Name	Pin Function	Setting I/O Port P9nDDR
I/O port	P9n output	1
	P9n input (initial setting)	0

(n = 5 to 0)

(m = 14 to 12, 2 to 0)

7.2.10 Port A

(1) PA7/$\overline{\text{KIN15}}$/PS2CD, PA6/$\overline{\text{KIN14}}$/PS2CC, PA5/$\overline{\text{KIN13}}$/PS2BD, PA4/$\overline{\text{KIN12}}$/PS2BC, PA3/$\overline{\text{KIN11}}$/PS2AD, PA2/$\overline{\text{KIN10}}$/PS2AC, PA1/$\overline{\text{KIN9}}$/PS2DD, PA0/$\overline{\text{KIN8}}$/PS2DC

The pin function is switched according to the combination of the register setting of PS2 and the PAnDDR bit. When the KMIMRm bit in KMIMRA of the interrupt controller is cleared to 0, this pin can be used as the $\overline{\text{KIN}}$m input pin.

		Setting	
		PS2	I/O Port
Module Name	Pin Function	PS2_OE	PAnDDR
PS2	PS2 input/output	1	—
I/O port	PAn output	0	1
	PAn input (initial setting)	0	0

(n = 7 to 0, m = 15 to 8)

Note: When the KBIOE bit is set to 1, this pin functions as an NMOS open-drain output, and direct bus drive is possible.

When the IICS bit in STCR is set to 1, the output format for PA7 to PA4 is NMOS open-drain, and direct bus drive is possible.

7.2.11 Port B

(1) PB7/$\overline{\text{RTS}}$/FSISS

The pin function is switched as shown below according to the combination of the SCIFE bit in HICR5 of LPC, the FSIE bit in FSICR1 of FSI and the PB7DDR bit.

		Setting		
		FSI	SCIF	I/O Port
Module Name	Pin Function	FSISS_OE	$\overline{\text{RTS}}$_OE	PB7DDR
FSI	FSISS output	1	—	—
SCIF	$\overline{\text{RTS}}$ output	0	1	—
I/O port	PB7 output	0	0	1
	PB7 input (initial setting)	0	0	0

(2) PB6/$\overline{\text{CTS}}$/FSICK

The pin function is switched as shown below according to the FSIE bit in FSICR1 of FSI and the PB6DDR bit.

		Setting	
		FSI	I/O Port
Module Name	Pin Function	FSICK_OE	PB6DDR
FSI	FSICK output	1	—
I/O port	PB6 output	0	1
	PB6 input (initial setting)	0	0

(3) PB5/$\overline{\text{DTR}}$/FSIDI

The pin function is switched as shown below according to the combination of the SCIFE bit in HICR5 of LPC, the FSIE bit in FSICR1of FSI and the PB5DDR bit.

		Setting		
		FSI	SCIF	I/O Port
Module Name	Pin Function	FSIDI	$\overline{\text{DTR}}$_OE	PB5DDR
FSI	FSIDI input	1	—	—
SCIF	$\overline{\text{DTR}}$ output	0	1	—
I/O port	PB5 output	0	0	1
	PB5 input (initial setting)	0	0	0

(4) PB4/$\overline{\text{DSR}}$/FSIDO

The pin function is switched as shown below according to the state of the FSIE bit in FSICR1 of FSI and the PB4DDR bit.

		Setting	
		FSI	I/O Port
Module Name	Pin Function	FSIDO_OE	PB4DDR
FSI	FSIDO output	1	—
I/O port	PB4 output	0	1
	PB4 input (initial setting)	0	0

(5) PB3/$\overline{\text{DCD}}$/PWMU1B

The pin function is switched as shown below according to the combination of the register setting of the PWMU and the PB3DDR bit.

		Setting	
		PWMU	I/O Port
Module Name	Pin Function	PWMU1B_OE	PB3DDR
PWMU	PWMU1B output	1	1
I/O port	PB3 output	0	1
	PB3 input (initial setting)	—	0

(6) PB2/$\overline{\text{RI}}$/PWMU0B

The pin function is switched as shown below according to the combination of the register setting of the PWMU and the PB2DDR bit.

		Setting	
		PWMU	I/O Port
Module Name	Pin Function	PWMU0B_OE	PB2DDR
PWMU	PWMU0B output	1	1
I/O port	PB2 output	0	1
	PB2 input (initial setting)	—	0

(7) PB1/LSCI

The pin function is switched as shown below according to the combination of the register setting of the LPC and the PB1DDR bit.

		Setting	
		LPC	I/O Port
Module Name	Pin Function	LSCI_OE	PB1DDR
LPC	LSCI output	1	—
I/O port	PB1 output	0	1
	PB1 input (initial setting)	0	0

(8) PB0/$\overline{\text{LSMI}}$

The pin function is switched as shown below according to the combination of the register setting of the LPC and the PB0DDR bit.

		Setting	
		LPC	I/O Port
Module Name	Pin Function	$\overline{\text{LSMI}}$_OE	PB0DDR
LPC	$\overline{\text{LSMI}}$ output	1	—
I/O port	PB0 output	0	1
	PB0 input (initial setting)	0	0

7.2.12 Port C

(1) PC7/$\overline{\text{WUE15}}$/TIOCB2/TCLKD

The pin function is switched as shown below according to the combination of the register setting of the TPU and the PC7DDR bit. When the WUEMR15 bit in WUEMR of the interrupt controller is cleared to 0, this pin can be used as the $\overline{\text{WUE15}}$ input pin.

This pin functions as the TCLKD input when TPSC2 to TPSC0 in TCR_0 is B'111. Also, when channel 2 is set to phase counting mode, this pin functions as the TCLKD input. This pin functions as TIOCB2 input when TPU channel 2 timer operating mode is set to normal operation or phase counting mode and IOB3 in TIOR_2 is set to 1.

		Setting	
		TPU	I/O Port
Module Name	Pin Function	TIOCB2_OE	PC7DDR
TPU	TIOCB2 output	1	—
I/O port	PC7 output	0	1
	PC7 input (initial setting)	0	0

(2) PC6/$\overline{\text{WUE14}}$/TIOCA2

The pin function is switched as shown below according to the combination of the register setting of the TPU and the PC6DDR bit. When the WUEMR14 bit in WUEMR of the interrupt controller is cleared to 0, this pin can be used as the $\overline{\text{WUE14}}$ input pin.

This pin functions as TIOCA2 input when TPU channel 2 timer operating mode is set to normal operation or phase counting mode and IOA3 in TIOR_2 is set to 1.

		Setting	
		TPU	I/O Port
Module Name	Pin Function	TIOCA2_OE	PC6DDR
TPU	TIOCA2 output	1	—
I/O port	PC6 output	0	1
	PC6 input (initial setting)	0	0

(3) PC5/$\overline{\text{WUE13}}$/TIOCB1/TCLKC

The pin function is switched as shown below according to the combination of the register setting of the TPU and the PC5DDR bit. When the WUEMR13 bit in WUEMR of the interrupt controller is cleared to 0, this pin can be used as the $\overline{\text{WUE13}}$ input pin.

This pin functions as TCLKC input when TPSC2 to TPSC0 in TCR_0 or TCR_2 is set to B'110 or when channel 2 is set to phase counting mode.

This pin functions as TIOCB1 input when TPU channel 1 timer operating mode is set to normal operation or phase counting mode and IOB3 to IOB0 in TIOR_1 are set to B'10xx. (x: Don't care.)

		Setting	
		TPU	I/O Port
Module Name	Pin Function	TIOCB1_OE	PC5DDR
TPU	TIOCB1 output	1	—
I/O port	PC5 output	0	1
	PC5 input (initial setting)	0	0

(4) PC4/$\overline{\text{WUE12}}$/TIOCA1

The pin function is switched as shown below according to the combination of the register setting of the TPU and the PC4DDR bit. When the WUEMR12 bit in WUEMR of the interrupt controller is cleared to 0, this pin can be used as the $\overline{\text{WUE12}}$ input pin.

This pin functions as TIOCA1 input when TPU channel 1 timer operating mode is set to normal operation or phase counting mode and IOA3 to IOA0 in TIOR_1 are set to B'10xx. (x: Don't care.)

		Setting	
		TPU	I/O Port
Module Name	Pin Function	TIOCA1_OE	PC4DDR
TPU	TIOCA1 output	1	—
I/O port	PC4 output	0	1
	PC4 input (initial setting)	0	0

(5) PC3/$\overline{\text{WUE11}}$/TIOCD0/TCLKB

The pin function is switched as shown below according to the combination of the register setting of the TPU and the PC3DDR bit. When the WUEMR11 bit in WUEMR of the interrupt controller is cleared to 0, this pin can be used as the $\overline{\text{WUE11}}$ input pin.

This pin functions as TCLKB input when TPSC2 to TPSC0 in any of TCR_0 to TCR_2 are set to B'101 or when channel 1 is set to phase counting mode.

This pin functions as TIOCD1 input when TPU channel 0 timer operating mode is set to normal operation or phase counting mode and IOD3 to IOD0 in TIOR_0 are set to B'10xx. (x: Don't care.)

		Setting	
		TPU	I/O Port
Module Name	Pin Function	TIOCD0_OE	PC3DDR
TPU	TIOCD0 output	1	—
I/O port	PC3 output	0	1
	PC3 input (initial setting)	0	0

(6) PC2/$\overline{\text{WUE10}}$/TIOCC0/TCLKA

The pin function is switched as shown below according to the combination of the register setting of the TPU and the PC2DDR bit. When the WUEMR10 bit in WUEMR of the interrupt controller is cleared to 0, this pin can be used as the $\overline{\text{WUE10}}$ input pin.

This pin functions as TCLKA input when TPSC2 to TPSC0 in any of TCR_0 to TCR_2 are set to B'100 or when channel 1 is set to phase counting mode.

This pin functions as TIOCC0 input when TPU channel 0 timer operating mode is set to normal operation or phase counting mode and IOC3 to IOC0 in TIOR_0 are set to B'10xx. (x: Don't care.)

		Setting	
		TPU	I/O Port
Module Name	Pin Function	TIOCC0_OE	PC2DDR
TPU	TIOCC0 output	1	—
I/O port	PC2 output	0	1
	PC2 input (initial setting)	0	0

(7) PC1/$\overline{\text{WUE9}}$/TIOCB0

The pin function is switched as shown below according to the combination of the register setting of the TPU and the PC1DDR bit. When the WUEMR9 bit in WUEMR of the interrupt controller is cleared to 0, this pin can be used as the $\overline{\text{WUE9}}$ input pin.

This pin functions as TIOCB0 input when TPU channel 0 timer operating mode is set to normal operation or phase counting mode and IOB3 to IOB0 in TIORH_0 are set to B'10xx. (x: Don't care.)

		Setting	
		TPU	I/O Port
Module Name	Pin Function	TIOCB0_OE	PC1DDR
TPU	TIOCB0 output	1	—
I/O port	PC1 output	0	1
	PC1 input (initial setting)	0	0

(8) PC0/$\overline{\text{WUE8}}$/TIOCA0

The pin function is switched as shown below according to the combination of the register setting of the TPU and the PC0DDR bit. When the WUEMR8 bit in WUEMR of the interrupt controller is cleared to 0, this pin can be used as the $\overline{\text{WUE8}}$ input pin.

This pin functions as TIOCA0 input when TPU channel 0 timer operating mode is set to normal operation or phase counting mode and IOA3 to IOA0 in TIORH_0 are set to B'10xx. (x: Don't care.)

		Setting	
		TPU	I/O Port
Module Name	Pin Function	TIOCA0_OE	PC0DDR
TPU	TIOCA0 output	1	—
I/O port	PC0 output	0	1
	PC0 input (initial setting)	0	0

7.2.13 Port D

(1) PD7/AN15, PD6/AN14, PD5/AN13, PD4/AN12, PD3/AN11, PD2/AN10, PD1/AN9, PD0/AN8

The pin function is switched as shown below according to the state of the PDnDDR bit.

When this pin is used as an analog input pin, do not set the pin as output.

		Setting
		I/O Port
Module Name	Pin Function	PDnDDR
I/O port	PDn output	1
	PDn input (initial setting)	0

(n = 7 to 0)

7.2.14 Port E

(1) PE4/ETMS, PE3/ETDO, PE2/ETDI, PE1/ETCK

The pin function is switched as shown below according to the operating mode.

		Setting	
		On-Chip Emulation Mode	Single-Chip Mode
Module Name	Pin Function	Emulator Input/Output	PEn input
Operating mode	On-chip emulation mode	1	—
	Single-chip mode	0	1

Note: These pins are not supported by the system development tool (emulator).

(2) PE0/ExEXCL

The pin function is switched as shown below according to the combination of the EXCLS bit in PTCNT0 and EXCLE bit in LPWRCR. When the EXCLS bit in PTCNT0 and EXCLE bit in LPWRCR are set to 1, this pin can be used as the ExEXCL input pin.

		Setting
		I/O Port
Module Name	Pin Function	ExEXCL
I/O port	PE0 input (initial setting)	0

7.2.15 Port F

(1) PF7/PWMU5, APF6/PWMU4A, PF5/PWMU3A, PF4/PWMU2A

The pin function is switched as shown below according to the combination of the register setting of the PWMU and the PFnDDR bit.

		Setting	
		PWMU	I/O Port
Module Name	Pin Function	PWMUmA_OE	PFnDDR
PWMU	PWMUmA output	1	1
I/O port	PFn output	0	1
	PFn input (initial setting)	—	0

(n = 5 to 2, m = 7 to 4)

(2) PF3/TMOX/$\overline{\text{IRQ11}}$/TDPCKI0/TDPMCI0

The pin function is switched as shown below according to the combination of the register setting of the TMR and the PF3DDR bit. When the PMMS bit in TDPCR2_0 of TDP0 is set to 1, this pin can be used as the TDPMCI0 input pin. When the external clock is selected by the CKS3 to CKS0 bits in TDPCR1_0 of TDP0, this pin is used as the TDPCKI0 input pin. Do not set input of TDPCKI0 and TDPMCI0 at the same time. When the ISS11 bit in ISSR16 is cleared to 0 and the IRQ11E bit in IER16 of the interrupt controller is set to 1, this pin can be used as the $\overline{\text{IRQ11}}$ input pin.

		Setting	
		TMR	I/O Port
Module Name	Pin Function	TMOX_OE	PF3DDR
TMR	TMOX output	1	—
I/O port	PF3 output	0	1
	PF3 input (initial setting)	0	0

(3) PF2/TMOY/$\overline{\text{IRQ10}}$/TDPCYI0

The pin function is switched as shown below according to the combination of the register setting of the TMR and the PF2DDR bit. When the TDPIPE bit in TDPIER_0 of TDP0 is set to 1, this pin can be used as the TDPCYI0 input pin. When the ISS10 bit in ISSR16 is cleared to 0 and the IRQ10E bit in IER16 of the interrupt controller is set to 1, this pin can be used as the $\overline{\text{IRQ10}}$ input pin.

		Setting	
		TMR	I/O Port
Module Name	Pin Function	TMOY_OE	PF2DDR
TMR	TMOY output	1	—
I/O port	PF2 output	0	1
	PF2 input (initial setting)	0	0

(4) PF1/$\overline{\text{IRQ9}}$/PWMU1A

The pin function is switched as shown below according to the combination of the register setting of the PWMU and the PF1DDR bit. When the ISS9 bit in ISSR16 is cleared to 0 and the IRQ9E bit in IER16 of the interrupt controller is set to 1, this pin can be used as the $\overline{\text{IRQ9}}$ input pin.

		Setting	
		PWMU	I/O Port
Module Name	Pin Function	PWMU1A_OE	PF1DDR
PWMU	PWMU1A output	1	1
I/O port	PF1 output	0	1
	PF1 input (initial setting)	—	0

(5) PF0/$\overline{\text{IRQ8}}$/PWMU0A

The pin function is switched as shown below according to the combination of the register setting of the PWMU and the PF1DDR bit. When the ISS8 bit in ISSR16 is cleared to 0 and the IRQ8E bit in IER16 of the interrupt controller is set to 1, this pin can be used as the $\overline{\text{IRQ8}}$ input pin.

		Setting	
		PWMU	I/O Port
Module Name	Pin Function	PWMU0A_OE	PF0DDR
PWMU	PWMU0A output	1	1
I/O port	PF0 output	0	1
	PF0 input (initial setting)	—	0

7.2.16 Port G

(1) PG7/ExSCLB/$\overline{\text{ExIRQ15}}$

The pin function is switched as shown below according to the combination of the register setting of PTCNT1 and the PG7DDR bit. When the ISS15 bit in ISSR16 is set to 1 and the IRQ15E bit in IER16 of the interrupt controller is set to 1, this pin can be used as the $\overline{\text{ExIRQ15}}$ input pin.

		Setting	
		PTCNT1	I/O Port
Module Name	Pin Function	ExSCLB_OE	PG7DDR
PTCNT1	ExSCLB input/output	1	—
I/O port	PG7 output	0	1
	PG7 input (initial setting)	0	0

Note: The output format for ExSCLB is NMOS output only, and direct bus drive is possible. When this pin is used as the PG7 output pin, the output format is NMOS push-pull.

(2) PG6/ExSDAB/$\overline{\text{ExIRQ14}}$

The pin function is switched as shown below according to the combination of the register setting of PTCNT1 and the PG6DDR bit. When the ISS14 bit in ISSR16 is set to 1 and the IRQ14E bit in IER16 of the interrupt controller is set to 1, this pin can be used as the $\overline{\text{ExIRQ14}}$ input pin.

		Setting	
		PTCNT1	I/O Port
Module Name	Pin Function	ExSDAB_OE	PG6DDR
PTCNT1	ExSDAB input/output	1	—
I/O port	PG6 output	0	1
	PG6 input (initial setting)	0	0

Note: The output format for ExSDAB is NMOS output only, and direct bus drive is possible. When this pin is used as the PG6 output pin, the output format is NMOS push-pull.

(3) PG5/ExSCLA/$\overline{\text{ExIRQ13}}$

The pin function is switched as shown below according to the combination of the register setting of PTCNT1 and the PG5DDR bit. When the ISS13 bit in ISSR16 is set to 1 and the IRQ13E bit in IER16 of the interrupt controller is set to 1, this pin can be used as the $\overline{\text{ExIRQ13}}$ input pin.

		Setting	
		PTCNT1	I/O Port
Module Name	Pin Function	ExSCLA_OE	PG5DDR
PTCNT1	ExSCLA input/output	1	—
I/O port	PG5 output	0	1
	PG5 input (initial setting)	0	0

Note: The output format for ExSCLA is NMOS output only, and direct bus drive is possible. When this pin is used as the PG5 output pin, the output format is NMOS push-pull.

(4) PG4/ExSDAA/$\overline{\text{ExIRQ12}}$

The pin function is switched as shown below according to the combination of the register setting of PTCNT1 and the PG4DDR bit. When the ISS12 bit in ISSR16 is set to 1 and the IRQ12E bit in IER16 of the interrupt controller is set to 1, this pin can be used as the $\overline{\text{ExIRQ12}}$ input pin.

		Setting	
		PTCNT1	I/O Port
Module Name	Pin Function	ExSDAA_OE	PG4DDR
PTCNT1	ExSDAA input/output	1	—
I/O port	PG4 output	0	1
	PG4 input (initial setting)	0	0

Note: The output format for ExSDAA is NMOS output only, and direct bus drive is possible. When this pin is used as the PG4 output pin, the output format is NMOS push-pull.

(5) PG3/SCL2/$\overline{\text{ExIRQ11}}$

The pin function is switched as shown below according to the combination of the register setting of the IIC and the PG3DDR bit. When the ISS11 bit in ISSR16 is set to 1 and the IRQ11E bit in IER16 of the interrupt controller is set to 1, this pin can be used as the $\overline{\text{ExIRQ11}}$ input pin.

		Setting	
		IIC	I/O Port
Module Name	Pin Function	SCL2_OE	PG3DDR
IIC	SCL2 input/output	1	—
I/O port	PG3 output	0	1
	PG3 input (initial setting)	0	0

Note: The output format for SCL2 is NMOS output only, and direct bus drive is possible. When this pin is used as the PG3 output pin, the output format is NMOS push-pull.

(6) PG2/SDA2/$\overline{\text{ExIRQ10}}$

The pin function is switched as shown below according to the combination of the register setting of the IIC and the PG2DDR bit. When the ISS10 bit in ISSR16 is set to 1 and the IRQ10E bit in IER16 of the interrupt controller is set to 1, this pin can be used as the $\overline{\text{ExIRQ10}}$ input pin.

		Setting	
		IIC	I/O Port
Module Name	Pin Function	SDA2_OE	PG2DDR
IIC	SDA2 input/output	1	—
I/O port	PG2 output	0	1
	PG2 input (initial setting)	0	0

Note: The output format for SDA2 is NMOS output only, and direct bus drive is possible. When this pin is used as the PG2 output pin, the output format is NMOS push-pull.

(7) PG1/$\overline{\text{ExIRQ9}}$/TMIY/TDPCKI1/TDPMCI1

The pin function is switched as shown below according to the state of the PG1DDR bit. When the PMMS bit in TDPCR2_1 of the TDP is set to 1, this pin is used as the TDPMCI1 input pin. When the external clock is selected by the CKS2 to CKS0 bits in TDPCR1_1 of the TDP, this pin is used as the TDPCKI1 input pin. Do not set input of TDPCKI1 and TDPMCI1 at the same time. When the ISS9 bit in ISSR16 is set to 1 and the IRQ9E bit in IER16 of the interrupt controller is set to 1, this pin can be used as the $\overline{\text{ExIRQ9}}$ input pin.

		Setting
		I/O Port
Module Name	Pin Function	PG1DDR
I/O port	PG1 output	1
	PG1 input (initial setting)	0

(8) PG0/$\overline{\text{ExIRQ8}}$/TMIX/TDPCYI1

The pin function is switched as shown below according to the state of the PG0DDR bit. When the TDPIPE bit in TDPIER_1 of the TDP is set to 1, this pin is used as the TDPCYI1 input pin. When the ISS8 bit in ISSR16 is set to 1 and the IRQ8E bit in IER16 of the interrupt controller is set to 1, this pin can be used as the $\overline{\text{ExIRQ8}}$ input pin.

		Setting
		I/O Port
Module Name	Pin Function	PG0DDR
I/O port	PG0 output	1
	PG0 input (initial setting)	0

7.2.17 Port H

(1) PH5, PH4, PH3

The pin function is switched as shown below according to the state of the PHnDDR bit.

		Setting
		I/O Port
Module Name	Pin Function	PHnDDR
I/O port	PHn output	1
	PHn input (initial setting)	0

(n = 5 to 3)

(2) PH2/CIRI

The pin function is switched as shown below according to the combination of the register setting of CIR and the PH2DDR bit.

		Setting	
		CIR	I/O Port
Module Name	Pin Function	CIRI	PH2DDR
CIR	CIRI input	1	—
I/O port	PH2 output	0	1
	PH2 input (initial setting)	0	0

(3) PH1/$\overline{\text{ExIRQ7}}$/TDPCKI2/TDPMCI2

The pin function is switched as shown below according to the state of the PH1DDR bit. When the PMMS bit in TDPCR2_2 of the TDP is set to 1, this pin is used as the TDPMCI2 input pin. When the external clock is selected by the CKS2 to CKS0 bits in TDPCR1_2 of the TDP, this pin is used as the TDPCKI2 input pin. Do not set input of TDPCKI2 and TDPMCI2 at the same time. When the ISS7 bit in ISSR is set to 1 and the IRQ7E bit in IER of the interrupt controller is set to 1, this pin can be used as the $\overline{\text{ExIRQ7}}$ input pin.

		Setting
		I/O Port
Module Name	Pin Function	PH1DDR
I/O port	PH1 output	1
	PH1 input (initial setting)	0

(4) PH0/$\overline{\text{ExIRQ6}}$/TDPCYI2

The pin function is switched as shown below according to the state of the PH0DDR bit. When the TDPIPE bit in TDPIER_2 of the TDP is set to 1, this pin is used as the TDPCYI2 input pin. When the EIVS bit in SYSCR3 is set to 1 and the IRQ6E bit in IER of the interrupt controller is set to 1, this pin can be used as the $\overline{\text{ExIRQ6}}$ input pin.

		Setting
		I/O Port
Module Name	Pin Function	PH0DDR
I/O port	PH0 output	1
	PH0 input (initial setting)	0

7.2.18 Port I

(1) PI7, PI6, PI5, PI4, PI3, PI2, PI1, PI0

The pin function is switched as shown below according to the state of the PInDDR bit.

Module Name	Pin Function	Setting I/O Port PInDDR
I/O port	PIn output	1
	PIn input (initial setting)	0

(n = 7 to 0)

Note: The output format for PIn is NMOS push-pull.

7.2.19 Port J

(1) PJ7, PJ6, PJ5, PJ4, PJ3, PJ2, PJ1, PJ0

The pin function is switched as shown below according to the state of the PJnDDR bit.

Module Name	Pin Function	Setting I/O Port PJnDDR
I/O port	PJn output	1
	PJn input (initial setting)	0

(n = 7 to 0)

Table 7.4 Available Output Signals and Settings in Each Port

Port		Output Specification Signal Name	Output Signal Name	Signal Selection Register Settings	Internal Module Settings
P1	7	P17_OE	P17		
	6	P16_OE	P16		
	5	P15_OE	P15		
	4	P14_OE	P14		
	3	P13_OE	P13		
	2	P12_OE	P12		
	1	P11_OE	P11		
	0	P10_OE	P10		
P2	7	P27_OE	P27		
	6	P26_OE	P26		
	5	P25_OE	P25		
	4	P24_OE	P24		
	3	P23_OE	P23		
	2	P22_OE	P22		
	1	P21_OE	P21		
	0	P20_OE	P20		
P3	7	SERIRQ_OE	SERIRQ		FSI.SLCR.FSILIE,
	6	P36_OE	P36		LPC.HICR5.SCIFE, HICR4.LPC4E, HICR0.LPC[3E:1E]
	5	P35_OE	P35		
	4	P34_OE	P34		LPCENABLE = 1: FSILIE + SCIFE + LPC4E + LPC3E + LPC2E + LPC1E
	3	LAD3_OE	LAD3		
	2	LAD2_OE	LAD2		
	1	LAD1_OE	LAD1		
	0	LAD0_OE	LAD0		

Port		Output Specification Signal Name	Output Signal Name	Signal Selection Register Settings	Internal Module Settings
P4	7	PWX1_OE	PWX1		PWMX.DACR.OEB = 1
		PWMU5B_OE	PWMU5B		PWMU_B.PWMCONB.PWM5E = 1
	6	PWX0_OE	PWX0		PWMX.DACR.OEA = 1
		PWMU4B_OE	PWMU4B		PWMU_B.PWMCONB.PWM4E = 1
	5	PWMU3B_OE	PWMU3B		PWMU_B.PWMCONB.PWM3E = 1
	4	TMO1_OE	TMO1		Except TMR_1.TCSR.OS[3:0] = 0000
		PWMU2B_OE	PWMU2B		PWMU_B.PWMCONB.PWM2E = 1
	3	SCK2_OE	SCK2		SCI_2.SCR.CKE[1:0] = 01/10/11 + SMR.C/$\overline{A}$ = 1
	2	SDA1_OE	SDA1	PTCNT1.IIC1AS PTCNT1.IIC1BS	ICE•$\overline{IIC1AS}$•$\overline{IIC1BS}$ = 1
	1	TMO0_OE	TMO0		Except TMR_0.TCSR.OS[3:0] = 0000
	0	TxD2_OE	TxD2		SCI_2.SCR.TE = 1
P5	2	SCL0_OE	SCL0	PTCNT1.IIC1AS PTCNT1.IIC1BS	ICE•$\overline{IIC0AS}$•$\overline{IIC0BS}$ = 1
	1	P51_OE	P51		
	0	FTxD_OE	FTxD		SCIF.SCIFCR.SCIFOE1, LPC.HICR5.SCIFE SCIFENABLE = 1: SCIFOE1 + SCIFE
P6	7	P67_OE	P67		
	6	P66_OE	P66		
	5	P65_OE	P65		
	4	P64_OE	P64		
	3	P63_OE	P63		
	2	P62_OE	P62		
	1	P61_OE	P61		
	0	P60_OE	P60		

Port		Output Specification Signal Name	Output Signal Name	Signal Selection Register Settings	Internal Module Settings
P8	6	SCK1_OE	SCK1		SCI_1.SMR.C/$\overline{A}$ = 1 or SCI_1.SMR.C/$\overline{A}$ = 0, SCR.CKE[1:0] = 01/10/11
		SCL1_OE	SCL1	PTCNT1.IIC1AS PTCNT1.IIC1BS	ICE•$\overline{IIC1AS}$•$\overline{IIC1BS}$ = 1
	5	P85_OE	P85		
	4	TxD1_OE	TxD1		SCI_1.SCR.TE = 1
	3	P83_OE	P83		
	2	$\overline{CLKRUN}$_OE	$\overline{CLKRUN}$		FSI.SLCR.FSILIE, LPC.HICR5.SCIFE, HICR4.LPC4E, HICR0.LPC[3E:1E] LPCENABLE = 1: FSILIE + SCIFE + LPC4E + LPC3E + LPC2E + LPC1E
	1	GA20_OE	GA20		LPC.HICR0.FGA20E = 1
	0	$\overline{PME}$_OE	$\overline{PME}$		LPC.HICR0.PMEE = 1
P9	7	SDA0_OE	SDA	PTCNT1.IIC0AS PTCNT1.IIC0BS	ICE•$\overline{IIC0AS}$•$\overline{IIC0BS}$ = 1
	6	ϕ_OE	ϕ		
	5	P95_OE	P95		
	4	P94_OE	P94		
	3	P93_OE	P93		
	2	P92_OE	P92		
	1	P91_OE	P91		
	0	P90_OE	P90		
PA	7	PS2CD_OE	PS2CD		PS2_2.KBCRH.KBIOE = 1
	6	PS2CC_OE	PS2CC		PS2_2.KBCRH.KBIOE = 1
	5	PS2BD_OE	PS2BD		PS2_1.KBCRH.KBIOE = 1
	4	PS2BC_OE	PS2BC		PS2_1.KBCRH.KBIOE = 1
	3	PS2AD_OE	PS2AD		PS2_0.KBCRH.KBIOE = 1
	2	PS2AC_OE	PS2AC		PS2_0.KBCRH.KBIOE = 1
	1	PS2DD_OE	PS2DD		PS2_3.KBCRH.KBIOE = 1
	0	PS2DC_OE	PS2DC		PS2_3.KBCRH.KBIOE = 1

Port		Output Specification Signal Name	Output Signal Name	Signal Selection Register Settings	Internal Module Settings
PB	7	$\overline{\text{RTS}}$_OE	$\overline{\text{RTS}}$		LPC.HICR5.SCIFE, SCIFCR.SCIFOE1, SCIFOE0 SCIFOE = 1: ($\overline{\text{SCIFE}}$ • SCIFOE1 • $\overline{\text{SCIFOE0}}$ + SCIFE • $\overline{\text{SCIFOE0}}$)
		FSISS_OE	FSISS		FSI.FSICR1.FSIE = 1
	6	FSICK_OE	FSICK		FSI.FSICR1.FSIE = 1
	5	$\overline{\text{DTR}}$_OE	$\overline{\text{DTR}}$		LPC.HICR5.SCIFE, SCIFCR.SCIFOE1, SCIFOE0 SCIFOE=1: ($\overline{\text{SCIFE}}$ • SCIFOE1 • $\overline{\text{SCIFOE0}}$ + SCIFE • $\overline{\text{SCIFOE0}}$)
	4	FSIDO_OE	FSIDO		FSI.FSICR1.FSIE = 1
	3	PWMU1B_OE	PWMU1B		PWMU_B.PWMCONB.PWM1E = 1
	2	PWMU0B_OE	PWMU0B		PWMU_B.PWMCONB.PWM0E = 1
	1	LSCI_OE	LSCI		LPC.HICR0.LSCIE = 1
	0	$\overline{\text{LSMI}}$_OE	$\overline{\text{LSMI}}$		LPC.HICR0.LSMIE = 1
PC	7	TIOCB2_OE	TIOCB2		TPU.TIOR2.IOB3 = 0, TPU.TIOR2.IOB[1:0] = 01/10/11
	6	TIOCA2_OE	TIOCA2		TPU.TIOR2.IOA3 = 0, TPU.TIOR2.IOA[1:0] = 01/10/11
	5	TIOCB1_OE	TIOCB1		TPU.TIOR1.IOB3 = 0, TPU.TIOR1.IOB[1:0] = 01/10/11
	4	TIOCA1_OE	TIOCA1		TPU.TIOR1.IOA3 = 0, TPU.TIOR1.IOA[1:0] = 01/10/11
	3	TIOCD0_OE	TIOCD0		TPU.TIOR0.IOD3 = 0, TPU.TIOR0.IOD[1:0] = 01/10/11
	2	TIOCC0_OE	TIOCC0		TPU.TIOR0.IOC3 = 0, TPU.TIOR0.IOC[1:0] = 01/10/11
	1	TIOCB0_OE	TIOCB0		TPU.TIOR0.IOB3 = 0, TPU.TIOR0.IOB[1:0] = 01/10/11
	0	TIOCA0_OE	TIOCA0		TPU.TIOR0.IOA3 = 0, TPU.TIOR0.IOA[1:0] = 01/10/11

Port		Output Specification Signal Name	Output Signal Name	Signal Selection Register Settings	Internal Module Settings
PD	7	PD7_OE	PD7		
	6	PD6_OE	PD6		
	5	PD5_OE	PD5		
	4	PD4_OE	PD4		
	3	PD3_OE	PD3		
	2	PD2_OE	PD2		
	1	PD1_OE	PD1		
	0	PD0_OE	PD0		
PF	7	PWMU5A_OE	PWMU5A		PWMU_A.PWMCONB.PWM5E = 1
	6	PWMU4A_OE	PWMU4A		PWMU_A.PWMCONB.PWM4E = 1
	5	PWMU3A_OE	PWMU3A		PWMU_A.PWMCONB.PWM3E = 1
	4	PWMU2A_OE	PWMU2A		PWMU_A.PWMCONB.PWM2E = 1
	3	TMOX_OE	TMOX		Except TMR_X.TCSR.OS[3:0] = 0000
	2	TMOY_OE	TMOY		Except TMR_Y.TCSR.OS[3:0] = 0000
	1	PWMU1A_OE	PWMU1A		PWMU_A.PWMCONB.PWM1E = 1
	0	PWMU0A_OE	PWMU0A		PWMU_A.PWMCONB.PWM0E = 1
PG	7	ExSCLB_OE	ExSCLB		PTCNT1.IIC1BS or PTCNT1.IIC0BS is 1
	6	ExSDAB_OE	ExSDAB		PTCNT1.IIC1BS or PTCNT1.IIC0BS is 1
	5	ExSCLA_OE	ExSCLA		PTCNT1.IIC1AS or PTCNT1.IIC0AS is 1
	4	ExSDAA_OE	ExSDAA		PTCNT1.IIC1AS or PTCNT1.IIC0AS is 1
	3	SCL2_OE	SCL2		IIC_2.ICCR.ICE = 1
	2	SDA2_OE	SDA2		IIC_2.ICCR.ICE = 1
	1	PG1_OE	PG1		
	0	PG0_OE	PG0		
PH	5	PH5_OE	PH5		
	4	PH4_OE	PH4		
	3	PH3_OE	PH3		
	2	PH2_OE	PH2		
	1	PH1_OE	PH1		
	0	PH0_OE	PH0		

Port		Output Specification Signal Name	Output Signal Name	Signal Selection Register Settings	Internal Module Settings
PI	7	PI7_OE	PI7		
	6	PI6_OE	PI6		
	5	PI5_OE	PI5		
	4	PI4_OE	PI4		
	3	PI3_OE	PI3		
	2	PI2_OE	PI2		
	1	PI1_OE	PI1		
	0	PI0_OE	PI0		
PJ	7	PJ7_OE	PJ7		
	6	PJ6_OE	PJ6		
	5	PJ5_OE	PJ5		
	4	PJ4_OE	PJ4		
	3	PJ3_OE	PJ3		
	2	PJ2_OE	PJ2		
	1	PJ1_OE	PJ1		
	0	PJ0_OE	PJ0		

7.3 Change of Peripheral Function Pins

For the external sub-clock input and IIC input/output, the multi-function I/O ports can be changed. I/O ports that also function as the external interrupt pins are changed by the setting of ISSR16 and ISSR. I/O ports that also function as the external sub-clock input pin are changed by the setting of PTCNT0. For IIC input/output, change the setting of PTCNT1. The pin name of the peripheral function is indicated by adding 'Ex' at the head of the original pin name. In each peripheral function description, the original pin name is used.

The following registers are available as the port control register.

- Port control register 0 (PTCNT0)
- Port control register 1 (PTCNT1)
- Port control register 2 (PTCNT2)

7.3.1 Port Control Register 0 (PTCNT0)

PTCNT0 selects ports that also function as the external sub-clock input pin.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R/W	Reserved The initial value should not be changed.
0	EXCLS	0	R/W	0: P96/EXCL is selected 1: PH0/ExEXCL is selected

7.3.2 Port Control Register 1 (PTCNT1)

PTCNT1 selects ports that also function as IIC input/output pins.

Bit	Bit Name	Initial Value	R/W	Description		
7	IIC1BS	0	R/W	These bits select input/output pins for IIC_1		
6	IIC1AS	0	R/W	IIC1BS	IIC1AS	
				0	0:	Selects P86/SCL1 and P42/SDA1
				0	1:	Selects PG5/ExSCLA and PG4/ExSDAA
				1	0:	Selects PG7/ExSCLB and PG6/ExSDAB
				1	1:	Setting prohibited
4	—	0	R/W	Reserved		
5	—	0		The initial value should not be changed.		
3	IIC0BS	0	R/W	These bits select input/output pins for IIC_0		
2	IIC0AS	0	R/W	IIC0BS	IIC0AS	
				0	0:	Selects P52/SCL0 and P97/SDA0
				0	1:	Selects PG5/ExSCLA and PG4/ExSDAA
				1	0:	Selects PG7/ExSCLB and PG6/ExSDAB
				1	1:	Setting prohibited
1	—	0	R/W	Reserved		
0	—	0	R/W	The initial value should not be changed.		

Note: Do not set input/output of IIC_0 and IIC_1 for one pin at the same time.

7.3.3 Port Control Register 2 (PTCNT2)

PTCNT2 selects ports that also function as SCI input/output pins and controls the port specification.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R/W	Reserved The initial value should not be changed.
6	TxD2RS	0	R/W	0: TxD2 direct output 1: TxD2 inverted output
5	RxD2RS	0	R/W	0: RxD2 direct input 1: RxD2 inverted input
4	TxD1RS	0	R/W	0: TxD1 direct output 1: TxD1 inverted output
3	RxD1RS	0	R/W	0: RxD1 direct input 1: RxD1 inverted input
2	—	0	R/W	Reserved The initial value should not be changed.
1	PORTS	0	R/W	0: Existing port specification 1: New port specification
0	—	0	R/W	Reserved The initial value should not be changed.

Section 8 8-Bit PWM Timer (PWMU)

This LSI has two channels of 8-bit PWM timers, A and B (PWMU_A and PWMU_B). Each PWMU outputs 6 PWM waveforms. Each of the PWM channels of a PWMU can operate independently. A PWMU allows long-period PWM outputs for six channels in 8-bit single-pulse mode and for three channels in 16-bit single-pulse mode. In addition, PWM outputs at a high carrier frequency are available in 8-bit pulse division mode. Connecting a low-pass filter externally to the LSI allows the PWMU to be used as an 8-bit D/A converter.

8.1 Features

- Selectable from four types of counter input clock
  Selection of four internal clock signals (ϕ, ϕ/2, ϕ/4, and ϕ/8)
- Independent operation and variable cycle for each channel
  Cascaded connection of two channels is possible.
  Operation of channel 1 (higher order) and channel 0 (lower order) as a 16-bit single-pulse PWM timer
  Operation of channel 3 (higher order) and channel 2 (lower order) as a 16-bit single-pulse PWM timer
  Operation of channel 5 (higher order) and channel 4 (lower order) as a 16-bit single-pulse PWM timer
- 8-bit single pulse mode
  Operates at a maximum carrier frequency of 78.1 kHz (at 20 MHz operation)
  Pulse output settable with a duty cycle from 0/255 to 255/255
  PWM output enable/disable control, and selection of direct or inverted PWM output
- 16-bit single pulse mode
  Two channels are cascade-connected for operation in this mode.
  Operates at a maximum carrier frequency of 305.1 Hz (at 20 MHz operation)
  Pulse output settable with a duty cycle from 0/65535 to 65535/65535
  PWM output enable/disable control, and selection of direct or inverted PWM output
- 8-bit pulse division mode
  Operable at a maximum carrier frequency of 1.25 MHz (at 20 MHz operation)
  Pulse output settable with a duty cycle from 0/16 to 15/16
  PWM output enable/disable control, and selection of direct or inverted PWM output

Figure 8.1 shows a block diagram of the PWMU.

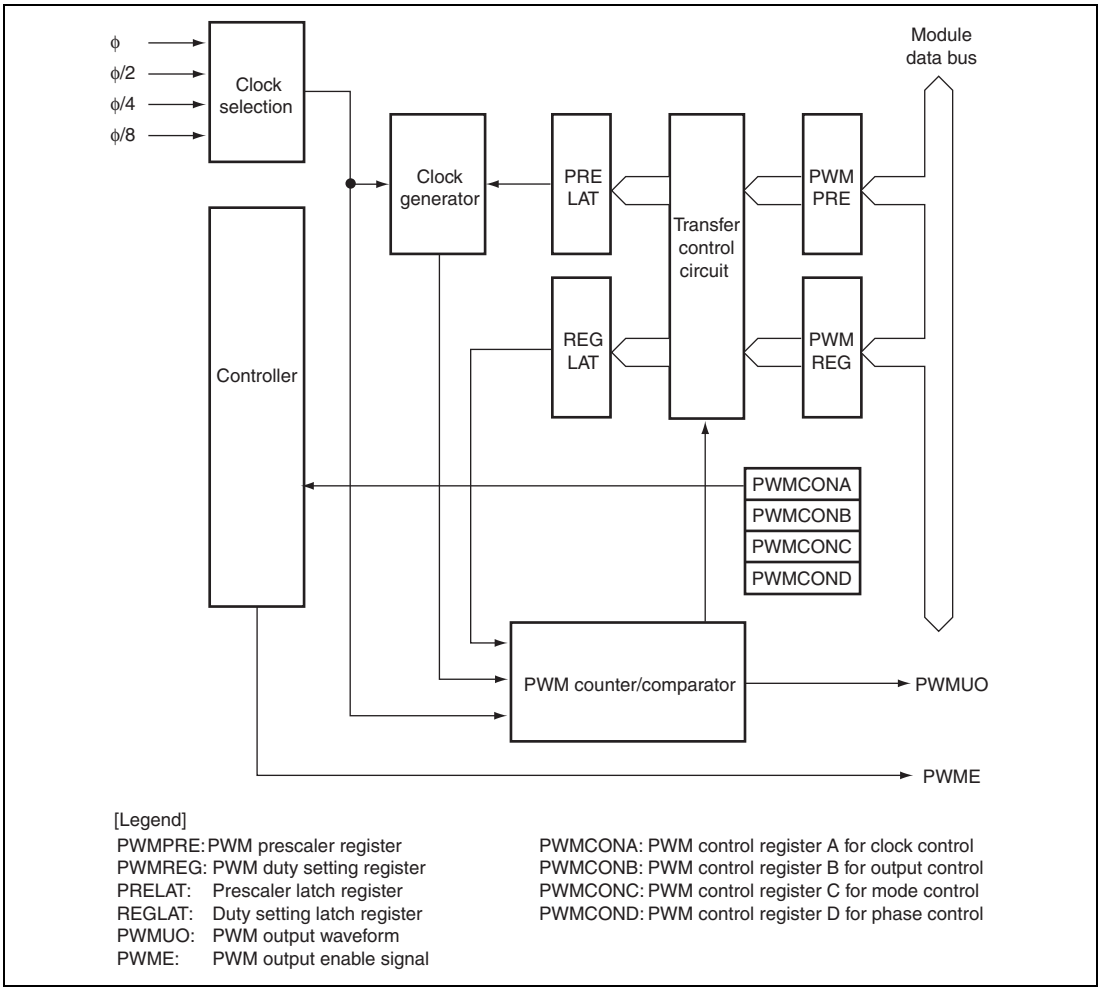


Figure 8.1 Block Diagram of PWMU Timer

8.2 Input/Output Pins

Table 8.1 shows the PWMU pin configuration.

Table 8.1 Pin Configuration

Channel		Pin Name	I/O	Function
Channel A	0	PWMU0A	Output	PWM pulse output (8-bit single pulse, 8-bit pulse division)
	1	PWMU1A	Output	PWM pulse output (8-bit single pulse, 16-bit single pulse, 8-bit pulse division)
	2	PWMU2A	Output	PWM pulse output (8-bit single pulse, 8-bit pulse division)
	3	PWMU3A	Output	PWM pulse output (8-bit single pulse, 16-bit single pulse, 8-bit pulse division)
	4	PWMU4A	Output	PWM pulse output (8-bit single pulse, 8-bit pulse division)
	5	PWMU5A	Output	PWM pulse output (8-bit single pulse, 16-bit single pulse, 8-bit pulse division)
Channel B	0	PWMU0B	Output	PWM pulse output (8-bit single pulse, 8-bit pulse division)
	1	PWMU1B	Output	PWM pulse output (8-bit single pulse, 16-bit single pulse, 8-bit pulse division)
	2	PWMU2B	Output	PWM pulse output (8-bit single pulse, 8-bit pulse division)
	3	PWMU3B	Output	PWM pulse output (8-bit single pulse, 16-bit single pulse, 8-bit pulse division)
	4	PWMU4B	Output	PWM pulse output (8-bit single pulse, 8-bit pulse division)
	5	PWMU5B	Output	PWM pulse output (8-bit single pulse, 16-bit single pulse, 8-bit pulse division)

8.3 Register Descriptions

The PWMU has the following registers.

Table 8.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Data Bus Width
Channel A	PWM control register A_A (for clock control)	PWMCONA_A	R/W	H'00	H'FD0C	8
	PWM control register B_A (for output control)	PWMCONB_A	R/W	H'00	H'FD0D	8
	PWM control register C_A (for mode control)	PWMCONC_A	R/W	H'00	H'FD0E	8
	PWM control register D_A (for phase control)	PWMCOND_A	R/W	H'00	H'FD0F	8
	PWM prescaler register 0_A	PWMPRE0_A	R/W	H'00	H'FD01	8
	PWM prescaler register 1_A	PWMPRE1_A	R/W	H'00	H'FD03	8
	PWM prescaler register 2_A	PWMPRE2_A	R/W	H'00	H'FD05	8
	PWM prescaler register 3_A	PWMPRE3_A	R/W	H'00	H'FD07	8
	PWM prescaler register 4_A	PWMPRE4_A	R/W	H'00	H'FD09	8
	PWM prescaler register 5_A	PWMPRE5_A	R/W	H'00	H'FD0B	8
	PWM duty setting register 0_A	PWMREG0_A	R/W	H'00	H'FD00	8
	PWM duty setting register 1_A	PWMREG1_A	R/W	H'00	H'FD02	8
	PWM duty setting register 2_A	PWMREG2_A	R/W	H'00	H'FD04	8
	PWM duty setting register 3_A	PWMREG3_A	R/W	H'00	H'FD06	8
	PWM duty setting register 4_A	PWMREG4_A	R/W	H'00	H'FD08	8
	PWM duty setting register 5_A	PWMREG5_A	R/W	H'00	H'FD0A	8

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Data Bus Width
Channel B	PWM control register A_B (for clock control)	PWMCONA_B	R/W	H'00	H'FD1C	8
	PWM control register B_B (for output control)	PWMCONB_B	R/W	H'00	H'FD1D	8
	PWM control register C_B (for mode control)	PWMCONC_B	R/W	H'00	H'FD1E	8
	PWM control register D_B (for phase control)	PWMCOND_B	R/W	H'00	H'FD1F	8
	PWM prescaler register 0_B	PWMPRE0_B	R/W	H'00	H'FD11	8
	PWM prescaler register 1_B	PWMPRE1_B	R/W	H'00	H'FD13	8
	PWM prescaler register 2_B	PWMPRE2_B	R/W	H'00	H'FD15	8
	PWM prescaler register 3_B	PWMPRE3_B	R/W	H'00	H'FD17	8
	PWM prescaler register 4_B	PWMPRE4_B	R/W	H'00	H'FD19	8
	PWM prescaler register 5_B	PWMPRE5_B	R/W	H'00	H'FD1B	8
	PWM duty setting register 0_B	PWMREG0_B	R/W	H'00	H'FD10	8
	PWM duty setting register 1_B	PWMREG1_B	R/W	H'00	H'FD12	8
	PWM duty setting register 2_B	PWMREG2_B	R/W	H'00	H'FD14	8
	PWM duty setting register 3_B	PWMREG3_B	R/W	H'00	H'FD16	8
	PWM duty setting register 4_B	PWMREG4_B	R/W	H'00	H'FD18	8
	PWM duty setting register 5_B	PWMREG5_B	R/W	H'00	H'FD1A	8

8.3.1 PWM Control Register A (PWMCONA)

PWMCONA selects the PWM clock source.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	CLK1, CLK0	All 0	R/W	Clock Select 1, 0 These bits select the PWM count clock source. CLK1 CLK0 0 0: Internal clock ϕ is selected 0 1: Internal clock $\phi/2$ is selected 1 0: Internal clock $\phi/4$ is selected 1 1: Internal clock $\phi/8$ is selected
5 to 0	–	All 0	R	Reserved These bits are always read as 0 and cannot be modified.

8.3.2 PWM Control Register B (PWMCONB)

PWMCONB controls enabling and disabling of the PWM output and counter operation of each channel.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R/W	Reserved The initial value should not be changed.
5	PWM5E	0	R/W	PWMU5 Output Enable 0: PWMU5 output and counter operation are disabled. 1: PWMU5 output and counter operation are enabled.

Bit	Bit Name	Initial Value	R/W	Description
4	PWM4E	0	R/W	PWMU4 Output Enable • 8-bit single-pulse/pulse-division mode 0: PWMU4 output and counter operation are disabled. 1: PWMU4 output and counter operation are enabled. • 16-bit single-pulse mode 0: PWMU4 output and counter operation are disabled. 1: PWMU4 output and counter operation are enabled.
3	PWM3E	0	R/W	PWMU3 Output Enable 0: PWMU3 output and counter operation are disabled. 1: PWMU3 output and counter operation are enabled.
2	PWM2E	0	R/W	PWMU2 Output Enable • 8-bit single-pulse/pulse division mode 0: PWMU2 output and counter operation are disabled. 1: PWMU2 output and counter operation are enabled. • 16-bit single-pulse mode 0: PWMU2 output and counter operation are disabled. 1: PWMU2 output and counter operation are enabled.
1	PWM1E	0	R/W	PWMU1 Output Enable 0: PWMU1 output and counter operation are disabled. 1: PWMU1 output and counter operation are enabled.

Bit	Bit Name	Initial Value	R/W	Description
0	PWM0E	0	R/W	PWMU0 Output Enable • 8-bit single-pulse/pulse division mode 0: PWMU0 output and counter operation are disabled. 1: PWMU0 output and counter operation are enabled. • 16-bit single-pulse mode 0: PWMU0 output and counter operation are disabled. 1: PWMU0 output and counter operation are enabled.

8.3.3 PWM Control Register C (PWMCONC)

PWMCONC selects the PWM count mode and operating mode for each channel.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R/W	Reserved The initial value should not be changed.
6	CNTMD01	0	R/W	Channels 0 and 1 Counter Select 0: Channels 0 and 1 are in 8-bit counter operation. 1: Channels 0 and 1 are in 16-bit counter operation (Upper: channel 1, lower: channel 0). Note: When the 16-bit counter is selected, specify single pulse mode.
5	PWMSL5	0	R/W	Channel 5 Operating Mode Select 0: Single-pulse mode 1: Pulse division mode (Specify 8-bit counter mode.)
4	PWMSL4	0	R/W	Channel 4 Operating Mode Select 0: Single pulse mode 1: Pulse division mode (Specify 8-bit counter mode.)
3	PWMSL3	0	R/W	Channel 3 Operating Mode Select 0: Single pulse mode 1: Pulse division mode (Specify 8-bit counter mode.)
2	PWMSL2	0	R/W	Channel 2 Operating Mode Select 0: Single pulse mode 1: Pulse division mode (Specify 8-bit counter mode.)
1	PWMSL1	0	R/W	Channel 1 Operating Mode Select 0: Single pulse mode 1: Pulse division mode (Specify 8-bit counter mode.)
0	PWMSL0	0	R/W	Channel 0 Operating Mode Select 0: Single pulse mode 1: Pulse division mode (Specify 8-bit counter mode.)

8.3.4 PWM Control Register D (PWMCOND)

PWMCOND selects the PWM count mode and output phase for each channel.

Bit	Bit Name	Initial Value	R/W	Description
7	PH5S	0	R/W	Channel 5 Output Phase Select 0: PWMU5 direct output 1: PWMU5 inverted output
6	PH4S	0	R/W	Channel 4 Output Phase Select 0: PWMU4 direct output 1: PWMU4 inverted output
5	PH3S	0	R/W	Channel 3 Output Phase Select 0: PWMU3 direct output 1: PWMU3 inverted output
4	PH2S	0	R/W	Channel 2 Output Phase Select 0: PWMU2 direct output 1: PWMU2 inverted output
3	PH1S	0	R/W	Channel 1 Output Phase Select 0: PWMU1 direct output 1: PWMU1 inverted output
2	PH0S	0	R/W	Channel 0 Output Phase Select 0: PWMU0 direct output 1: PWMU0 inverted output
1	CNTMD45	0	R/W	Channels 4 and 5 Counter Select 0: Channels 4 and 5 are in 8-bit counter operation. 1: Channels 4 and 5 are in 16-bit counter operation (Upper: channel 5, lower: channel 4). Note: When the 16-bit counter is selected, specify single pulse mode.
0	CNTMD23	0	R/W	Channels 2 and 3 Counter Select 0: Channels 2 and 3 are in 8-bit counter operation. 1: Channels 2 and 3 are in 16-bit counter operation (Upper: channel 2, lower: channel 3). Note: When the 16-bit counter is selected, specify single pulse mode.

8.3.5 PWM Prescaler Registers 0 to 5 (PWMPRE0 to PWMPRE5)

PWMPRE are 8-bit readable/writable registers used to set the PWM cycle. The initial value is H'00.

When the PWMPRE value is n, the PWM cycle is calculated as follows.

(1) 8-Bit Single Pulse Mode

$$\text{PWM cycle} = [255 \times (n + 1)] / \text{internal clock frequency} \quad (0 \leq n \leq 255)$$

Table 8.3 Resolution, PWM Conversion Period, and Carrier Frequency (8-Bit Counter Operation) when ϕ = 20 MHz

				Carrier Frequency	
		PWM Conversion Period		Single Pulse Mode	
Internal Clock Frequency	Resolution	Min.	Max.	Min.	Max.
ϕ	50 ns	12.8 µs	3.3 ms	306.4 Hz	78.4 kHz
ϕ/2	100 ns	25.5 µs	6.5 ms	153.2 Hz	39.2 kHz
ϕ/4	200 ns	51.2 µs	13.1 ms	76.6 Hz	19.6 kHz
ϕ/8	400 ns	102 µs	26.1 ms	38.3 Hz	9.8 kHz

(2) 16-Bit Single Pulse Mode

When 16-bit single pulse mode is selected, PWMPRE0, PWMPRE2, and PWMPRE4 are valid. The settings of PWMPRE1, PWMPRE3, and PWMPRE5 are invalid.

PWM cycle = [65535 × (n + 1)] / internal clock frequency (0 ≤ n ≤ 255)

Table 8.4 Resolution, PWM Conversion Period, and Carrier Frequency (16-Bit Counter Operation) when ϕ = 20 MHz

Internal Clock Frequency	Resolution	PWM Conversion Period		Carrier Frequency Single Pulse Mode	
		Min.	Max.	Min.	Max.
ϕ	50 ns	3.3 ms	838.9 ms	1.2 Hz	305.1 Hz
ϕ/2	100 ns	6.5 ms	1.7 s	0.6 Hz	152.6 Hz
ϕ/4	200 ns	13.1 ms	3.4 s	0.3 Hz	76.3 Hz
ϕ/8	400 ns	26.2 ms	6.7 s	0.15 Hz	38.1 Hz

(3) 8-Bit Pulse Division Mode

PWM cycle = [16 × (n + 1)] / internal clock frequency (0 ≤ n ≤ 255)

PWM conversion cycle = [256 × (n + 1)] / internal clock frequency (0 ≤ n ≤ 255)

Table 8.5 Resolution, PWM Conversion Period, and Carrier Frequency when ϕ = 20 MHz (at 8-bit counter operation)

Internal Clock Frequency	Resolution	PWM Conversion Period		Carrier Frequency (1/PWM cycle)	
		Min.	Max.	Min.	Max.
ϕ	50 ns	12.8 μs	3.3ms	4882.8Hz	1250.0 kHz
ϕ/2	100 ns	25.6 μs	6.6ms	2441.4Hz	625.0 kHz
ϕ/4	200 ns	51.2 μs	13.1ms	1220.7Hz	312.5 kHz
ϕ/8	400 ns	102.4 μs	26.2ms	610.4Hz	156.3 kHz

8.3.6 PWM Duty Setting Registers 0 to 5 (PWMREG0 to PWMREG5)

PWMREG0 to PWMREG5 are 8-bit readable/writable registers used to set the high period (duty) of the PWM output pulse. The initial value is H'00.

(1) 8-Bit Single Pulse Mode

Directly set the high period of the pulse for PWM output. With PWMREG registers, the duty cycle of the PWM output pulse is specified as a value from 0/255 to 255/255 with a resolution of 1/255.

When the PWMREG value is m, the high period of the output pulse is calculated as follows:

$$\text{Output pulse high period} = (\text{PWM cycle} \times m) / 255 \quad (0 \leq m \leq 255)$$

(2) 16-Bit Single Pulse Mode

Directly set the high period of the pulse for PWM output. With cascade-connected PWMREG registers, the duty cycle of the PWM output pulse is specified as a value from 0/65535 to 65535/65535.

When the PWMREG value is m, the high period of the output pulse is calculated as follows:

$$\text{Output pulse high period} = (\text{PWM cycle} \times m) / 65535 \quad (0 \leq m \leq 65535)$$

Set the respective high-level pulse periods by using the following register combinations (cascaded connection): PWMREG1 (higher order) and PWMREG0 (lower order), PWMREG3 (higher order) and PWMREG2 (lower order), and PWMREG5 (higher order) and PWMREG4 (lower order).

(3) 8-Bit Pulse Division Mode

Specify the basic pulse duty cycle and the number of additional pulses for PWM output. The higher-order four bits of the PWMREG setting specify the duty cycle of the basic pulse as 0/16 to 15/16 with a resolution of 1/16, and the lower-order four bits specify the number of pulses to be added within the conversion period comprising the basic pulses.

8.4 Operation

The PWMU operates in 8-bit single pulse mode, 16-bit single pulse mode, or 8-bit division pulse mode.

8.4.1 Single-Pulse Mode (8 Bits, 16 Bits)

Figure 8.2 shows a block diagram of 8-bit single pulse mode. Figure 8.3 shows a block diagram of 16-bit single pulse mode.

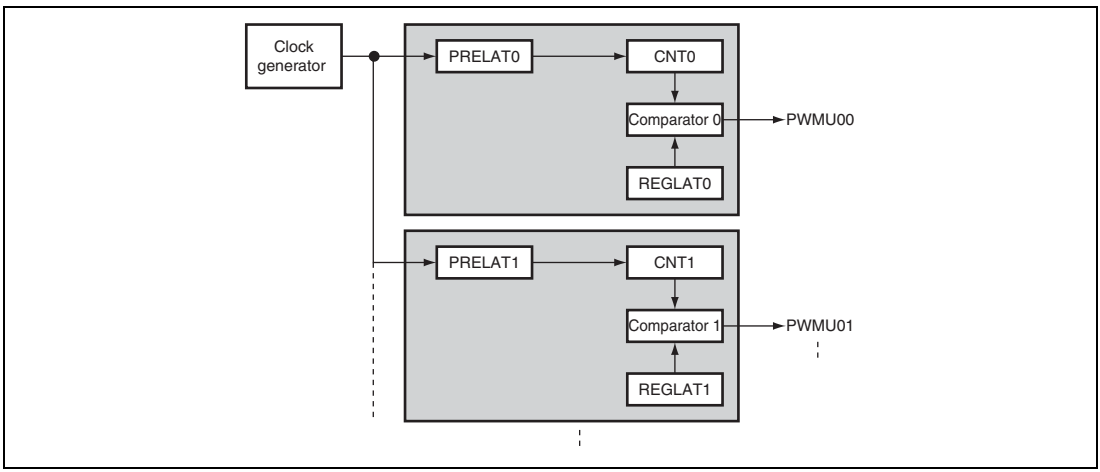


Figure 8.2 Block Diagram of 8-Bit Single Pulse Mode

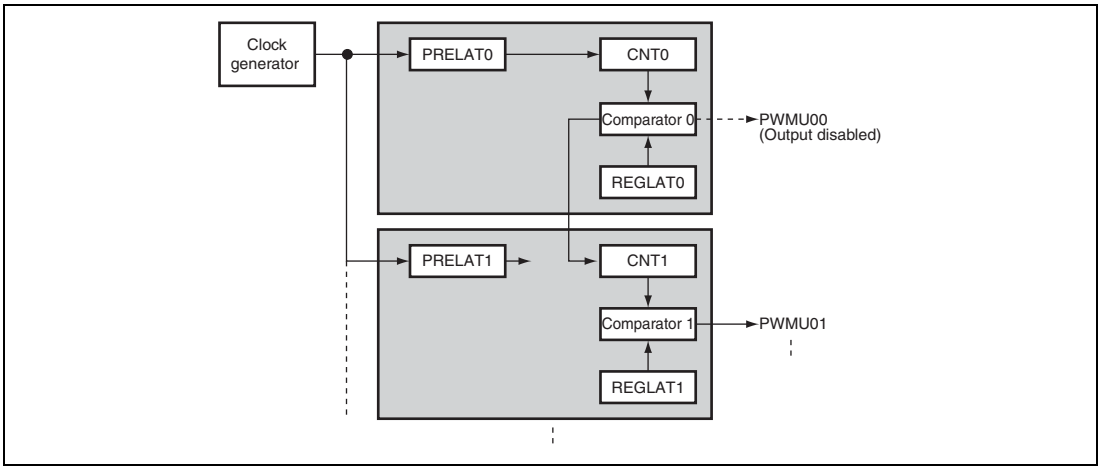


Figure 8.3 Block Diagram of 16-bit Single Pulse Mode

When the PWMnE bit (n = 0 to 5) in PWMCONB is set to 1, the PWMU outputs pulses that start with a high level. The updated PWMREG value is written in REGLAT, and the updated PWMPRE value is written in PRELAT.

When the REGLAT value is less than the duty counter value, the PWMU outputs a high level (when direct output is selected). At each PWM clock timing, the duty counter is incremented. When the clock generator counter is H'00, the PWM clock is generated by decrementing the PRELAT value.

Figure 8.4 shows an example of duty counter and clock generator counter operation.

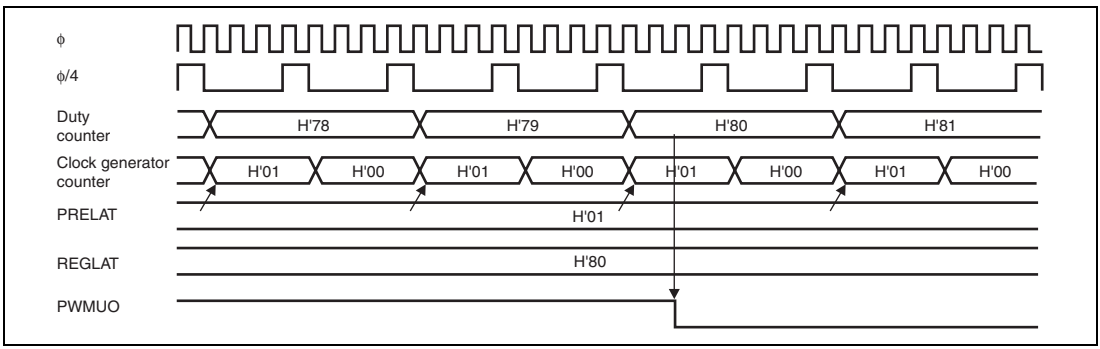


Figure 8.4 Example of Duty Counter and Clock Generator Counter Operation (When PWMPRE = H'01 and PWMREG = H'80 with φ/4 Selected as Count Clock Source)

The following shows the duty counter value and PWMU output timing.

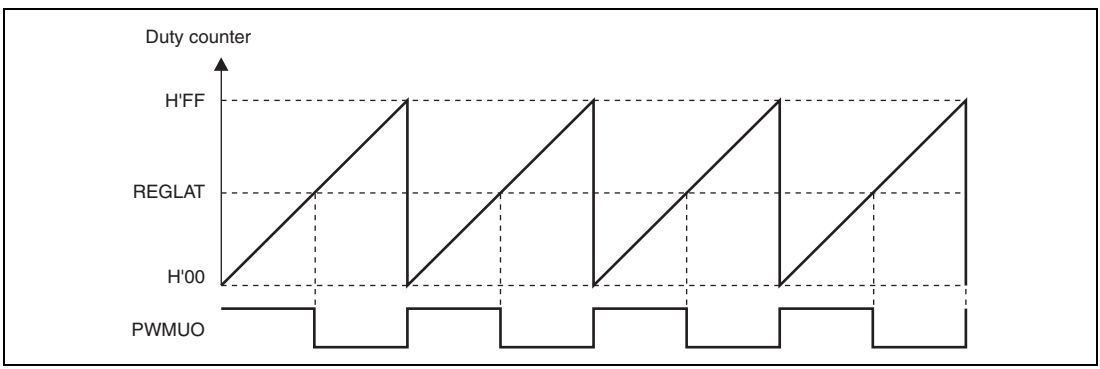


Figure 8.5 Duty Counter Value and PWMU Output Timing

If the PWMREG value is changed during PWM output, the PWMREG value is loaded into REGLAT when the duty counter overflows (at the beginning of the next PWM cycle). The following shows the PWMU output waveform when the PWMREG value is changed.

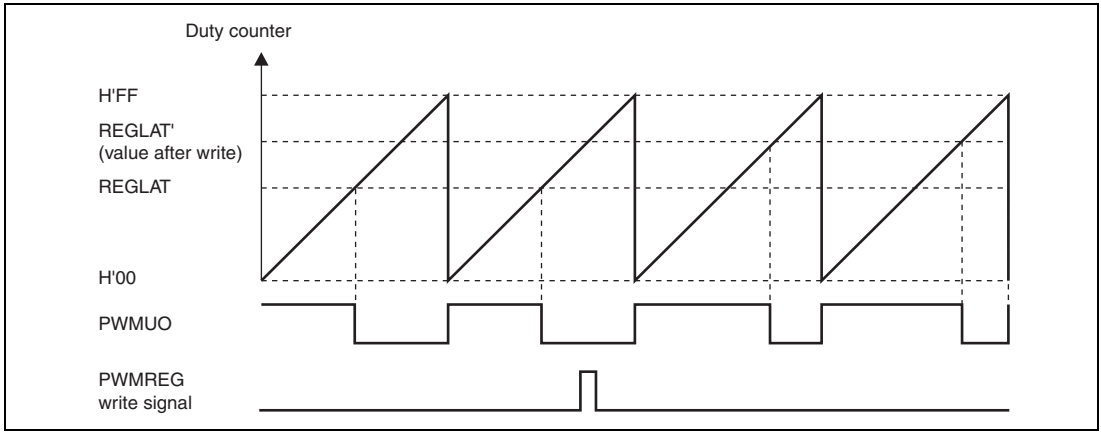


Figure 8.6 PWMU Output Waveform When PWMREG Value is Changed

When the PWMPRE value is changed during PWM output, the PWM cycle changes from the next cycle. When the clock generator counter underflows, the PWMPRE value is loaded into PRELAT. The following shows the PRELAT update timing when the PWMPRE value is changed.

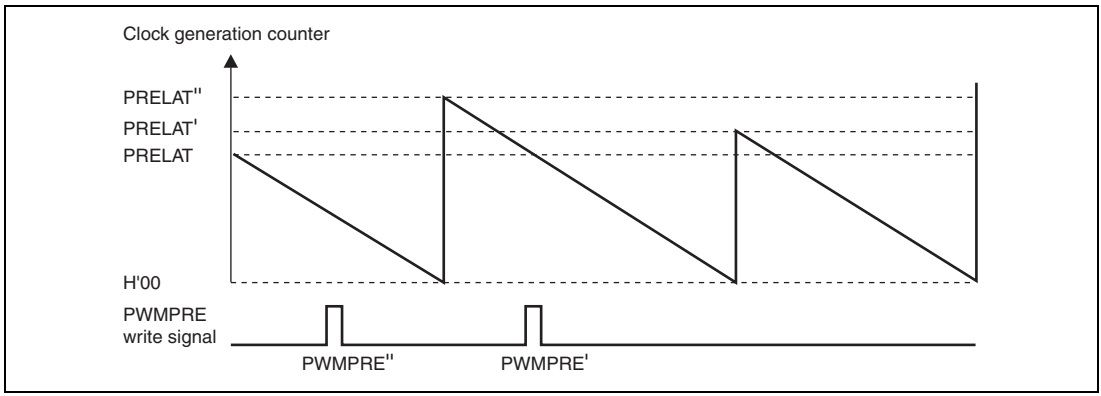


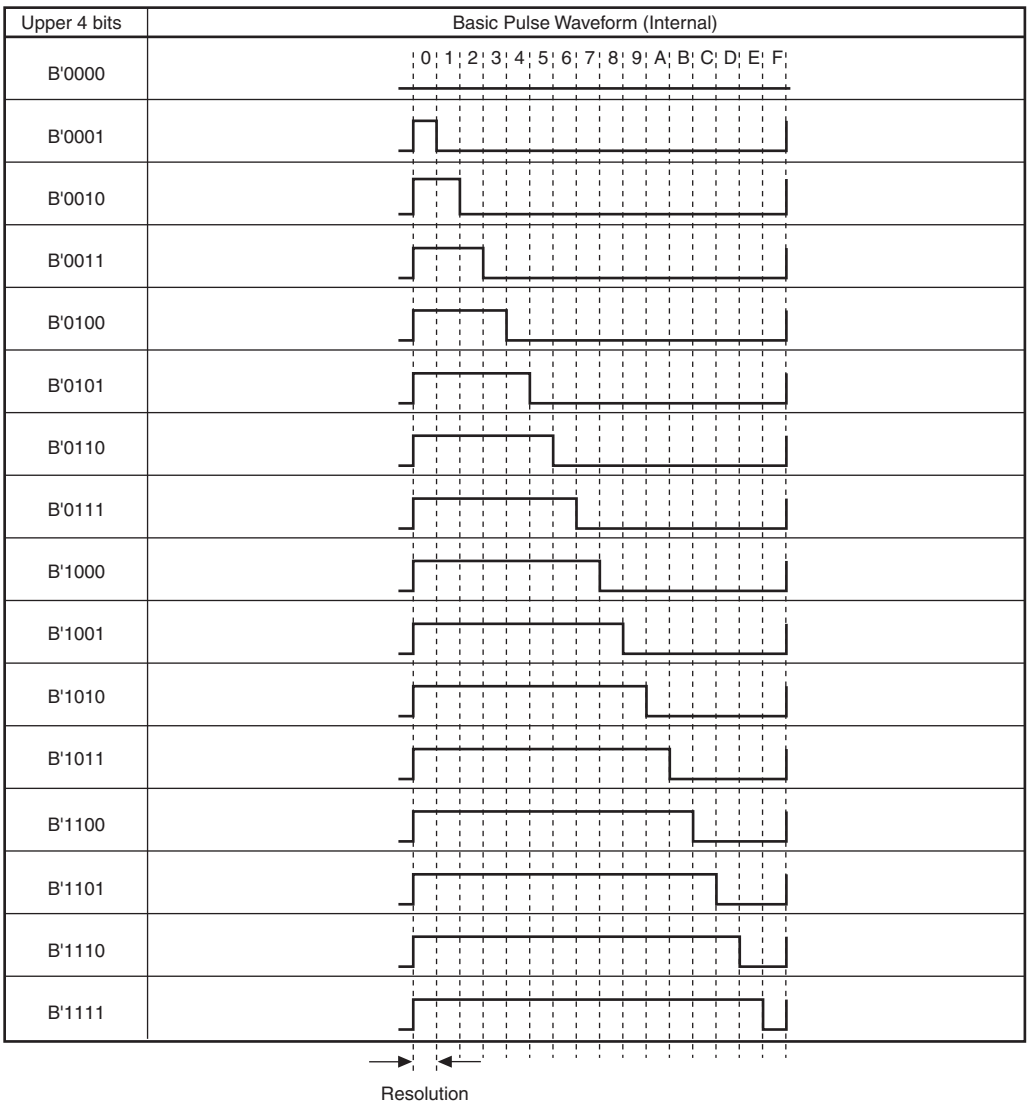
Figure 8.7 PRELAT Update Timing When PWMPRE Value is Changed

8.4.2 Pulse Division Mode

In pulse division mode, the higher-order four bits in PWMREG specify the duty cycle of the basic pulse as 0/16 to 15/16 with a resolution of 1/16. The following shows the duty cycle of the basic pulse.

Table 8.6 Basic Pulse Duty Cycle

Upper 4 bits	Basic Pulse Waveform (Internal)
B'0000	
B'0001	
B'0010	
B'0011	
B'0100	
B'0101	
B'0110	
B'0111	
B'1000	
B'1001	
B'1010	
B'1011	
B'1100	
B'1101	
B'1110	
B'1111	



The lower four bits in PWMREG specify the position of pulses added to the 16 basic pulses. The additional pulse adds a high period (when PHnS = 0) at the resolution width before the rising edge of the basic pulse. Although there is no rising edge of the basic pulse when the upper four bits in PWMREG is B'0000, the timing for adding pulses is the same. Table 8.7 shows the additional pulse positions corresponding to the basic pulses, and figure 8.8 shows an example of additional pulse timing.

Table 8.7 Additional Pulse Positions Corresponding to Basic Pulse

Lower 4 Bits	Basic Pulse Number															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
B'0000																
B'0001																O
B'0010								O								O
B'0011								O				O				O
B'0100				O				O				O				O
B'0101				O				O				O		O		O
B'0110				O		O		O				O		O		O
B'0111				O		O		O		O		O		O		O
B'1000		O		O		O		O		O		O		O		O
B'1001		O		O		O		O		O		O		O	O	O
B'1010		O		O		O	O	O		O		O		O	O	O
B'1011		O		O		O	O	O		O	O	O		O	O	O
B'1100		O	O	O		O	O	O		O	O	O		O	O	O
B'1101		O	O	O		O	O	O		O	O	O	O	O	O	O
B'1110		O	O	O	O	O	O	O		O	O	O	O	O	O	O
B'1111		O	O	O	O	O	O	O	O	O	O	O	O	O	O	O

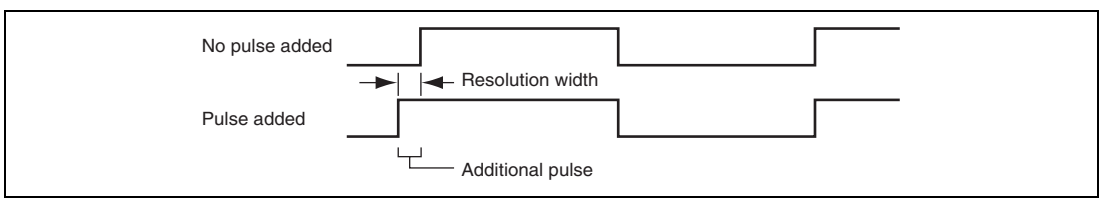


Figure 8.8 Example of Additional Pulse Timing (Upper 4 Bits in PWMREG = B'1000)

(1) Example of Setting

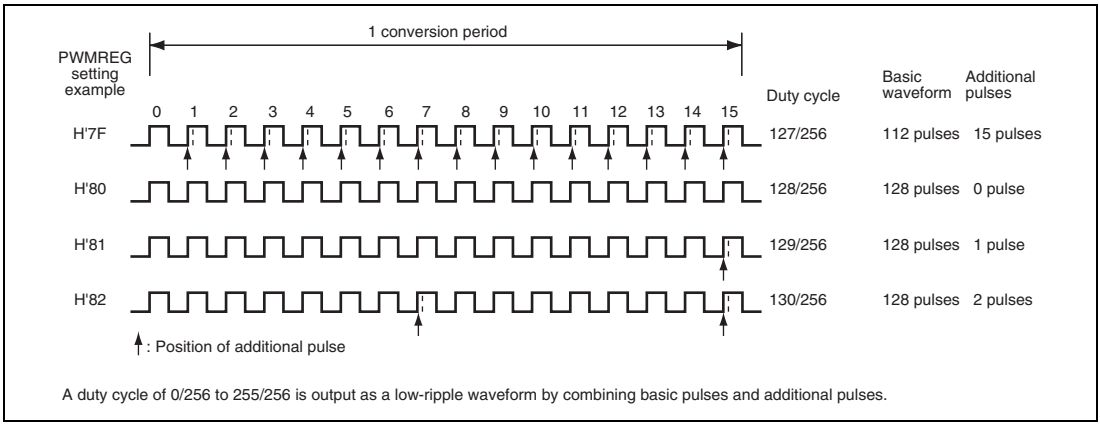


Figure 8.9 Example of WMU Setting

(2) Example of Circuit for Use as D/A Converter

The following shows an example of a circuit in which PWMU output pulses are used as a D/A converter. When a low-pass filter is connected externally to the LSI, low-ripple analog output can be generated. If pulse division mode is used, a D/A output with even less ripple is available.

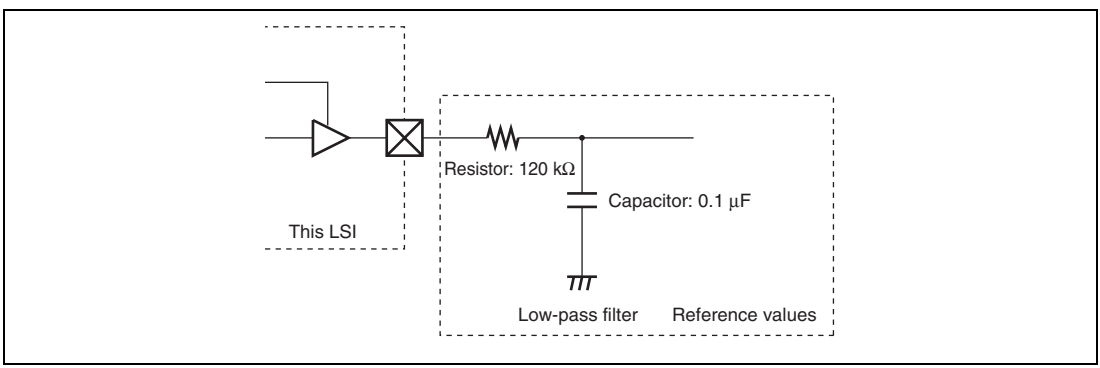


Figure 8.10 Example of Circuit for Use as a D/A Converter

8.5 Usage Note

8.5.1 Setting Module Stop Mode

The module stop control register can be used to enable or disable PWMU operation. The default setting disables PWMU operation. Clearing the module stop mode enables registers to be accessed. For details, see section 26, Power-Down Modes.

8.5.2 Note on Using 16-Bit Single-Pulse PWM Timer

When the duty cycle is to be changed in usage of a 16-bit single-pulse PWM timer, the higher- and lower-order eight bits must be individually written to the respective PWMREGn (n = 0 to 5) registers. There will thus be a time lag between the write operations, and this may lead to the output of a pulse waveform with a duty cycle other than the intended one during the corresponding period.

Also, care must be taken to ensure that there are no interrupts while writing to PWMREGn is in progress, since interrupt processing can lead to the continued output of pulses with a duty cycle other than the intended one.

Section 9 14-Bit PWM Timer (PWMX)

This LSI has an on-chip 14-bit pulse-width modulator (PWM) timer with two output channels. It can be connected to an external low-pass filter to operate as a 14-bit D/A converter.

9.1 Features

- Division of pulse into multiple base cycles to reduce ripple
- Eight resolution settings
  The resolution can be set to 1, 2, 64, 128, 256, 1024, 4096, or 16384 system clock cycles.
- 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 cycle settings)

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

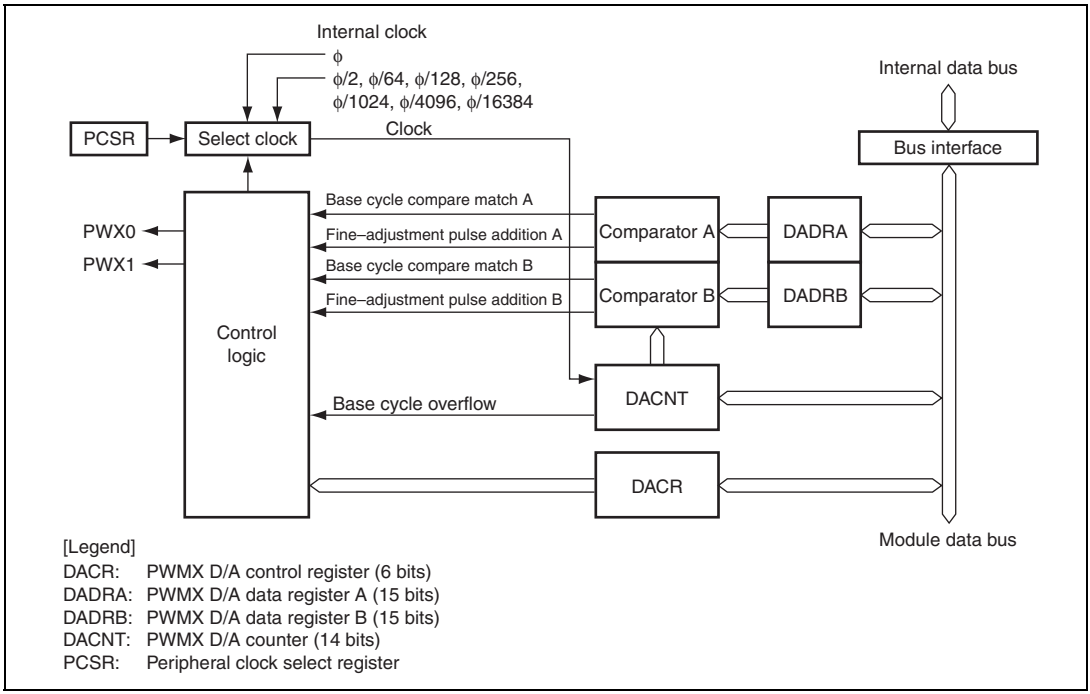


Figure 9.1 PWMX (D/A) Block Diagram

9.2 Input/Output Pins

Table 9.1 lists the PWMX (D/A) module input and output pins.

Table 9.1 Pin Configuration

Pin Name	Abbreviation	I/O	Function
PWMX output pin 0	PWX0	Output	PWMX output of channel A
PWMX output pin 1	PWX1	Output	PWMX output of channel B

9.3 Register Descriptions

The PWMX (D/A) module has the following registers. The PWMX (D/A) registers are assigned to the same addresses with other registers. The registers are selected by the IICE bit in the serial timer control register (STCR). For details on the module stop control register, see section 26.1.3, Module Stop Control Registers H, L, A, and B (MSTPCRH, MSTPCRL, MSTPCRA, MSTPCRB).

Table 9.2 Register Configuration

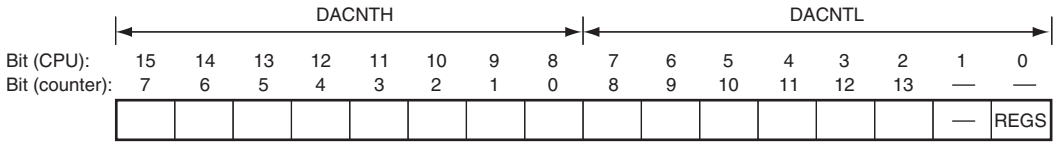
Register Name	Abbreviation	R/W	Initial Value	Address	Data Bus Width
PWMX (D/A) counter H	DACNTH	R/W	H'00	H'FFA6 H'FEA6*	8
PWMX (D/A) counter L	DACNTL	R/W	H'03	H'FFA7 H'FEA7*	8
PWMX (D/A) data register AH	DADRAH	R/W	H'FF	H'FFA0 H'FEA0*	8
PWMX (D/A) data register AL	DADRAL	R/W	H'FF	H'FFA1 H'FEA1*	8
PWMX (D/A) data register BH	DADRBH	R/W	H'FF	H'FFA6 H'FEA6*	8
PWMX (D/A) data register BL	DADRBL	R/W	H'FF	H'FFA7 H'FEA7*	8
PWMX (D/A) control register	DACR	R/W	H'30	H'FFA0 H'FEA0*	8
Peripheral clock select register	PCSR	R/W	H'00	H'FF82	8

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

* Upper address: when RELOCATE = 0
Lower address: when RELOCATE = 1

9.3.1 PWMX (D/A) Counter (DACNT)

DACNT is a 14-bit readable/writable up-counter. The input clock is selected by the clock select bit (CKS) in DACR. DACNT functions as the time base for both PWMX (D/A) channels. When a channel operates with 14-bit precision, it uses all DACNT bits. When a channel operates with 12-bit precision, it uses the lower 12 bits and ignores the upper 2-bit counter. As DACNT is 16 bits, data transfer between the CPU is performed through the temporary register (TEMP). For details, see section 9.4, Bus Master Interface.



- DACNTH

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	DACNT7 to DACNT0	All 0	R/W	Upper Up-Counter

- DACNTL

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	DACNT 8 to DACNT 13	All 0	R/W	Lower Up-Counter
1	—	1	R	Reserved Always read as 1 and cannot be modified.
0	REGS	1	R/W	Register Select DADRA and DACR, and DADRB and DACNT, are located at the same addresses. The REGS bit specifies which registers can be accessed. 0: DADRA and DADRB can be accessed 1: DACR and DACNT can be accessed

9.3.2 PWMX (D/A) Data Registers A and B (DADRA and DADRB)

DADRA corresponds to PWMX (D/A) channel A, and DADRB to PWMX (D/A) channel B. As DACNT is 16 bits, data transfer between the CPU is performed through the temporary register (TEMP). For details, see section 9.4, Bus Master Interface.

- DADRA

Bit	Bit Name	Initial Value	R/W	Description
15	DA13	1	R/W	D/A Data 13 to 0 These bits set a digital value to be converted to an analog value. In each base cycle, the DACNT value is continually compared with the DADR value to determine the duty cycle of the output waveform, and to decide whether to output a fine-adjustment pulse equal in width to the resolution. To enable this operation, this register must be set within a range that depends on the CFS bit. If the DADR value is outside this range, the PWM output is held constant. A channel can be operated with 12-bit precision by fixing DA0 and DA1 to 0. The two data bits are not compared with DACNT12 and DACNT13 of DACNT.
14	DA12	1	R/W	
13	DA11	1	R/W	
12	DA10	1	R/W	
11	DA9	1	R/W	
10	DA8	1	R/W	
9	DA7	1	R/W	
8	DA6	1	R/W	
7	DA5	1	R/W	
6	DA4	1	R/W	
5	DA3	1	R/W	
4	DA2	1	R/W	
3	DA1	1	R/W	
2	DA0	1	R/W	
1	CFS	1	R/W	Carrier Frequency Select 0: Base cycle = resolution (T) × 64 The range of DA13 to DA0: H'0100 to H'3FFF 1: Base cycle = resolution (T) × 256 The range of DA13 to DA0: H'0040 to H'3FFF
0	—	1	R	Reserved Always read as 1 and cannot be modified.

- DADRB

Bit	Bit Name	Initial Value	R/W	Description
15	DA13	1	R/W	D/A Data 13 to 0 These bits set a digital value to be converted to an analog value. In each base cycle, the DACNT value is continually compared with the DADR value to determine the duty cycle of the output waveform, and to decide whether to output a fine-adjustment pulse equal in width to the resolution. To enable this operation, this register must be set within a range that depends on the CFS bit. If the DADR value is outside this range, the PWM output is held constant. A channel can be operated with 12-bit precision by fixing DA0 and DA1 to 0. The two data bits are not compared with DACNT12 and DACNT13 of DACNT.
14	DA12	1	R/W	
13	DA11	1	R/W	
12	DA10	1	R/W	
11	DA9	1	R/W	
10	DA8	1	R/W	
9	DA7	1	R/W	
8	DA6	1	R/W	
7	DA5	1	R/W	
6	DA4	1	R/W	
5	DA3	1	R/W	
4	DA2	1	R/W	
3	DA1	1	R/W	
2	DA0	1	R/W	
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, are located at the same addresses. The REGS bit specifies which registers can be accessed. 0: DADRA and DADRB can be accessed 1: DACR and DACNT can be accessed

9.3.3 PWMX (D/A) Control Register (DACR)

DACR enables the PWM outputs, and selects the output phase and operating speed.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R/W`	Reserved The initial value should not be changed.
6	PWME	0	R/W	PWMX Enable Starts or stops the PWM D/A counter (DACNT). 0: DACNT operates as a 14-bit up-counter 1: DACNT halts at H'0003
5	—	1	R	Reserved
4	—	1	R	Always read as 1 and cannot be modified.
3	OEB	0	R/W	Output Enable B Enables or disables output on PWMX (D/A) channel B. 0: PWMX (D/A) channel B output (at the PWX1 output pin) is disabled 1: PWMX (D/A) channel B output (at the PWX1 output pin) is enabled
2	OEA	0	R/W	Output Enable A Enables or disables output on PWMX (D/A) channel A. 0: PWMX (D/A) channel A output (at the PWX0 output pin) is disabled 1: PWMX (D/A) channel A output (at the PWX0 output pin) 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 of resolution can be selected. 0: Operates at resolution (T) = system clock cycle time (t_{cyc}) 1: Operates at resolution (T) = system clock cycle time (t_{cyc}) × 2, × 64, × 128, × 256, × 1024, × 4096, and × 16384.

9.3.4 Peripheral Clock Select Register (PCSR)

PCSR and the CKS bit of DACR select the operating speed.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R/W	Reserved
6	—	0	R/W	The initial value should not be changed.
5	PWCKXB	0	R/W	PWMX clock select
4	PWCKXA	0	R/W	These bits select a clock cycle with the CKS bit of DACR of PWMX being 1. See table 9.3.
3 to 1	—	All 0	R/W	Reserved The initial value should not be changed.
0	PWCKXC	0	R/W	PWMX clock select This bit selects a clock cycle with the CKS bit of DACR of PWMX being 1. See table 9.3.

Table 9.3 Clock Select of PWMX

PWCKXC	PWCKXB	PWCKXA	Resolution (T)
0	0	0	Operates on the system clock cycle (t_{cyc}) x 2
0	0	1	Operates on the system clock cycle (t_{cyc}) x 64
0	1	0	Operates on the system clock cycle (t_{cyc}) x 128
0	1	1	Operates on the system clock cycle (t_{cyc}) x 256
1	0	0	Operates on the system clock cycle (t_{cyc}) x 1024
1	0	1	Operates on the system clock cycle (t_{cyc}) x 4096
1	1	0	Operates on the system clock cycle (t_{cyc}) x 16384
1	1	1	Setting prohibited

9.4 Bus Master Interface

DACNT, DADRA, and DADRB are 16-bit registers. The data bus linking the bus master and the on-chip peripheral modules, however, is only 8 bits wide. When the bus master accesses these registers, it therefore uses an 8-bit temporary register (TEMP).

These registers are written to and read from as follows.

- Write
  When the upper byte is written to, the upper-byte write data is stored in TEMP. Next, when the lower byte is written to, the lower-byte write data and TEMP value are combined, and the 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 the lower-byte value is transferred to TEMP. Next, when the lower byte is read from, the lower-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 the upper byte should always be accessed before the lower byte. Correct data will not be transferred if only the upper byte or only the lower byte is accessed. Also note that a bit manipulation instruction 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
```

Table 9.4 Reading/Writing to 16-bit Registers

	Read		Write	
Register	**Word**	**Byte**	**Word**	**Byte**
DADRA, DADRB	O	O	O	×
DACNT	O	×	O	×

[Legend]

O: Enabled access.

Word-unit access includes accessing byte sequentially, first upper byte, and then lower byte.

×: The result of the access in the unit cannot be guaranteed.

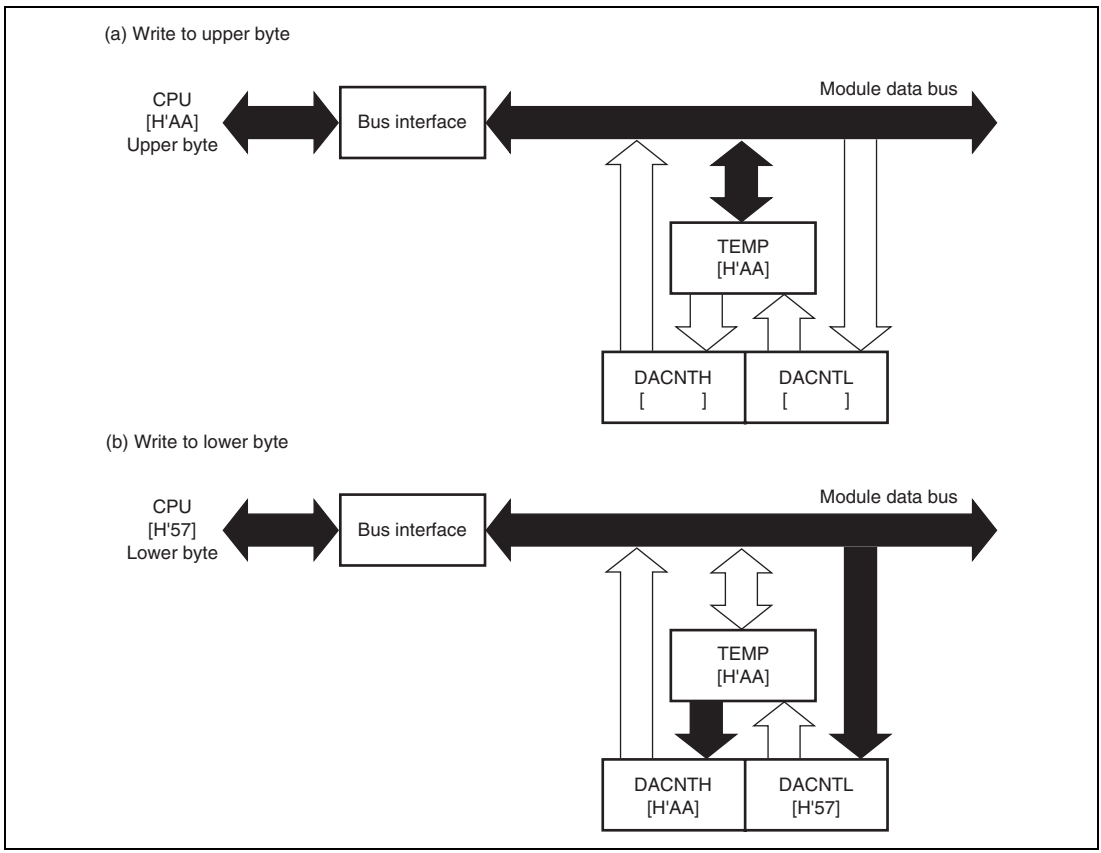


Figure 9.2 DACNT Access Operation (1) [CPU → DACNT (H'AA57) Writing]

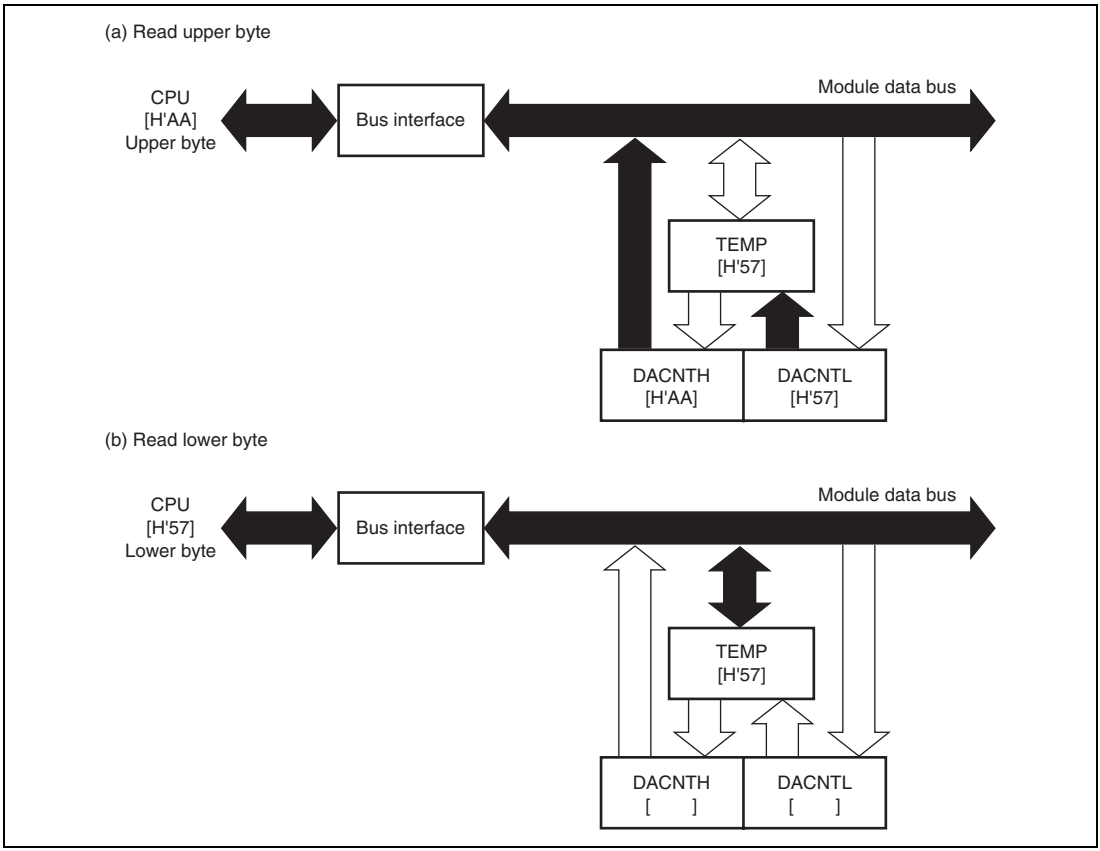


Figure 9.2 DACNT Access Operation (2) [DACNT → CPU (H'AA57) Reading]

9.5 Operation

A PWM waveform like the one shown in figure 9.3 is output from the PWMX pin. DA13 to DA0 in DADR corresponds to the total width (T_L) of the low (0) pulses output in one conversion cycle (256 pulses when CFS = 0, 64 pulses when CFS = 1). When OS = 0, this waveform is directly output. When OS = 1, the output waveform is inverted, and DA13 to DA0 in DADR value corresponds to the total width (T_H) of the high (1) output pulses. Figures 9.4 and 9.5 show the types of waveform output available.

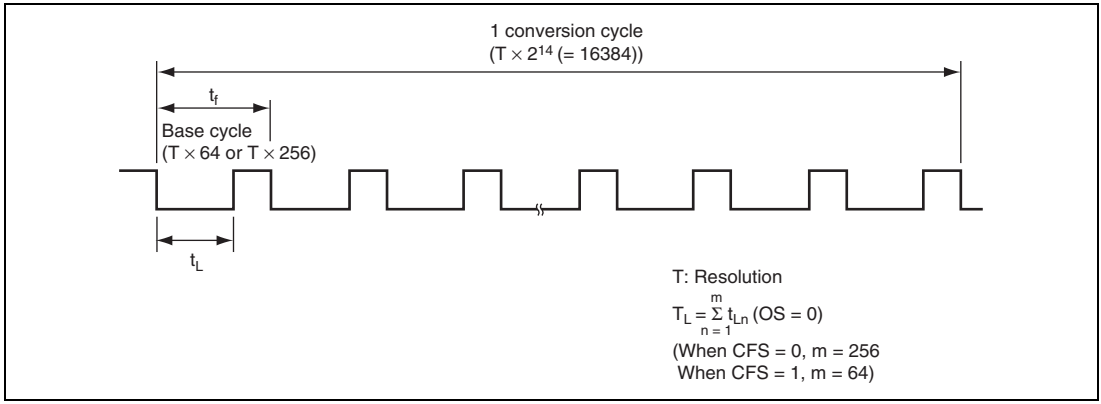


Figure 9.3 PWMX (D/A) Operation

Table 9.5 summarizes the relationships between the CKS and CFS bit settings and the resolution, base cycle, and conversion cycle. The PWM output remains fixed unless DA13 to DA0 in DADR contain at least a certain minimum value. The relationship between the OS bit and the output waveform is shown in figures 9.4 and 9.5.

Table 9.5 Settings and Operation (Examples when ϕ = 20 MHz)

PCSR									Fixed DADR Bits					
PWCKX0 PWCKX1			CK	Reso-lution T		Base	Conver-sion	TL/TH	Accuracy	Bit Data				Conversion
C	B	A	S	(μs)	CFS	Cycle	Cycle	(OS = 0/OS = 1)	(Bits)	DA3	DA2	DA1	DA0	Cycle*
—	—	—	0	0.05	0	3.2	819.2	Always low/high output	14					819.2 μs
						(μs)	(μs)	DA13 to 0 = H'0000 to H'00FF	12			0	0	204.8 μs
						/312.5kHz		(Data value) × T	10	0	0	0	0	51.2 μs
								DA13 to 0 = H'0100 to H'3FFF						
					1	12.8		Always low/high output	14					819.2 μs
						(μs)		DA13 to 0 = H'0000 to H'003F	12			0	0	204.8 μs
				(ϕ)		/78.1kHz		(Data value) × T	10	0	0	0	0	51.2 μs
								DA13 to 0 = H'0040 to H'3FFF						
0	0	0	1	0.1	0	6.4	1.64	Always low/high output	14					1638.4 μs
						(μs)	(ms)	DA13 to 0 = H'0000 to H'00FF	12			0	0	409.6 μs
						/156.2kHz		(Data value) × T	10	0	0	0	0	102.4 μs
								DA13 to 0 = H'0100 to H'3FFF						
					1	25.6		Always low/high output	14					1638.4 μs
						(μs)		DA13 to 0 = H'0000 to H'003F	12			0	0	409.6 μs
				(ϕ/2)		/39.1kHz		(Data value) × T	10	0	0	0	0	102.4 μs
								DA13 to 0 = H'0040 to H'3FFF						
0	0	1	1	3.2	0	204.8	52.4	Always low/high output	14					52.4 ms
						(μs)	(ms)	DA13 to 0 = H'0000 to H'00FF	12			0	0	13.1 ms
						/4.9kHz		(Data value) × T	10	0	0	0	0	3.3 ms
								DA13 to 0 = H'0100 to H'3FFF						
					1	819.2		Always low/high output	14					52.4 ms
						(μs)		DA13 to 0 = H'0000 to H'003F	12			0	0	13.1 ms
				(ϕ/64)		/1.2kHz		(Data value) × T	10	0	0	0	0	3.3 ms
								DA13 to 0 = H'0040 to H'3FFF						
0	1	0	1	6.4	0	409.6	104.9	Always low/high output	14					104.9 ms
						(μs)	(ms)	DA13 to 0 = H'0000 to H'00FF	12			0	0	26.2 ms
						/2.4kHz		(Data value) × T	10	0	0	0	0	6.6 ms
								DA13 to 0 = H'0100 to H'3FFF						
					1	1638.4		Always low/high output	14					104.9 ms
						(μs)		DA13 to 0 = H'0000 to H'003F	12			0	0	26.2 ms
				(ϕ/128)		/610.4kHz		(Data value) × T	10	0	0	0	0	6.6 ms
								DA13 to 0 = H'0040 to H'3FFF						

PCSR								Fixed DADR Bits					
PWCKX0 PWCKX1				Resolution T		Base	Conversion	TL/TH	Accuracy	Bit Data				Conversion
C	B	A	CKS	(μs)	CFS	Cycle	Cycle	(OS = 0/OS = 1)	(Bits)	DA3	DA2	DA1	DA0	Cycle*
0	1	1	1	12.8	0	819.2 (μs) /1.2kHz	209.7 (ms)	Always low/high output	14					209.7 ms
								DA13 to 0 = H'0000 to H'00FF (Data value) × T	12			0	0	52.4 ms
								DA13 to 0 = H'0100 to H'3FFF	10	0	0	0	0	13.1 ms
					1	3276.8 (μs) /305.2kHz		Always low/high output	14					209.7 ms
								DA13 to 0 = H'0000 to H'003F (Data value) × T	12			0	0	52.4 ms
				(ϕ/256)				DA13 to 0 = H'0040 to H'3FFF	10	0	0	0	0	13.1 ms
1	0	0	1	51.2	0	3.3 (ms) /305.2Hz	838.9 (ms)	Always low/high output	14					838.9 ms
								DA13 to 0 = H'0000 to H'00FF (Data value) × T	12			0	0	209.7 ms
								DA13 to 0 = H'0100 to H'3FFF	10	0	0	0	0	52.4 ms
					1	13.1 (ms) /76.3Hz		Always low/high output	14					838.9 ms
								DA13 to 0 = H'0000 to H'003F (Data value) × T	12			0	0	209.7 ms
				(ϕ/1024)				DA13 to 0 = H'0040 to H'3FFF	10	0	0	0	0	52.4 ms
1	0	1	1	204.8	0	13.1 (ms) /76.3Hz	3.4 (s)	Always low/high output	14					3.4 s
								DA13 to 0 = H'0000 to H'00FF (Data value) × T	12			0	0	838.9 ms
								DA13 to 0 = H'0100 to H'3FFF	10	0	0	0	0	209.7 ms
					1	52.4 (ms) /19.1Hz		Always low/high output	14					3.4 s
								DA13 to 0 = H'0000 to H'003F (Data value) × T	12			0	0	838.9 ms
				(ϕ/4096)				DA13 to 0 = H'0040 to H'3FFF	10	0	0	0	0	209.7 ms
1	1	0	1	819.2	0	52.4 (ms) /19.1Hz	13.4 (s)	Always low/high output	14					13.4 s
								DA13 to 0 = H'0000 to H'00FF (Data value) × T	12			0	0	3.4 s
								DA13 to 0 = H'0100 to H'3FFF	10	0	0	0	0	838.9 ms
					1	209.7 (ms) /4.8Hz		Always low/high output	14					13.4 s
								DA13 to 0 = H'0000 to H'003F (Data value) × T	12			0	0	3.4 s
				(ϕ/16384)				DA13 to 0 = H'0040 to H'3FFF	10	0	0	0	0	838.9 ms
1	1	1	1	Setting prohibited	—	—	—	—	—	—	—	—	—	—

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

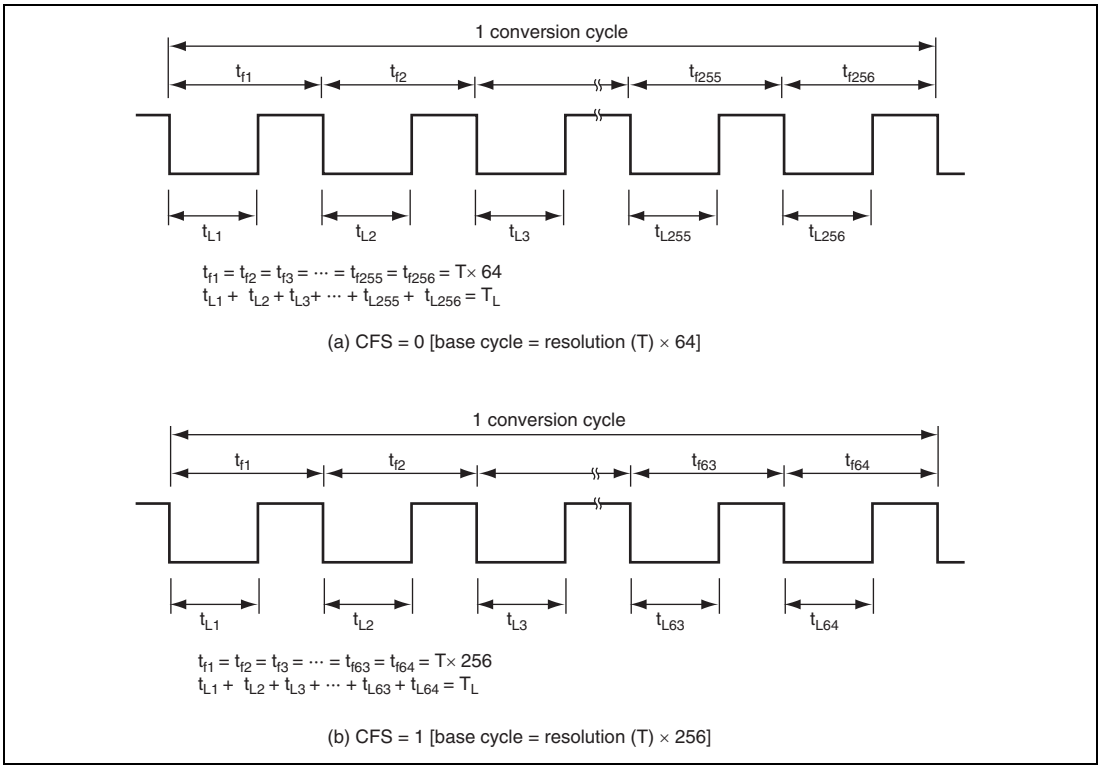


Figure 9.4 Output Waveform (OS = 0, DADR corresponds to T_L)

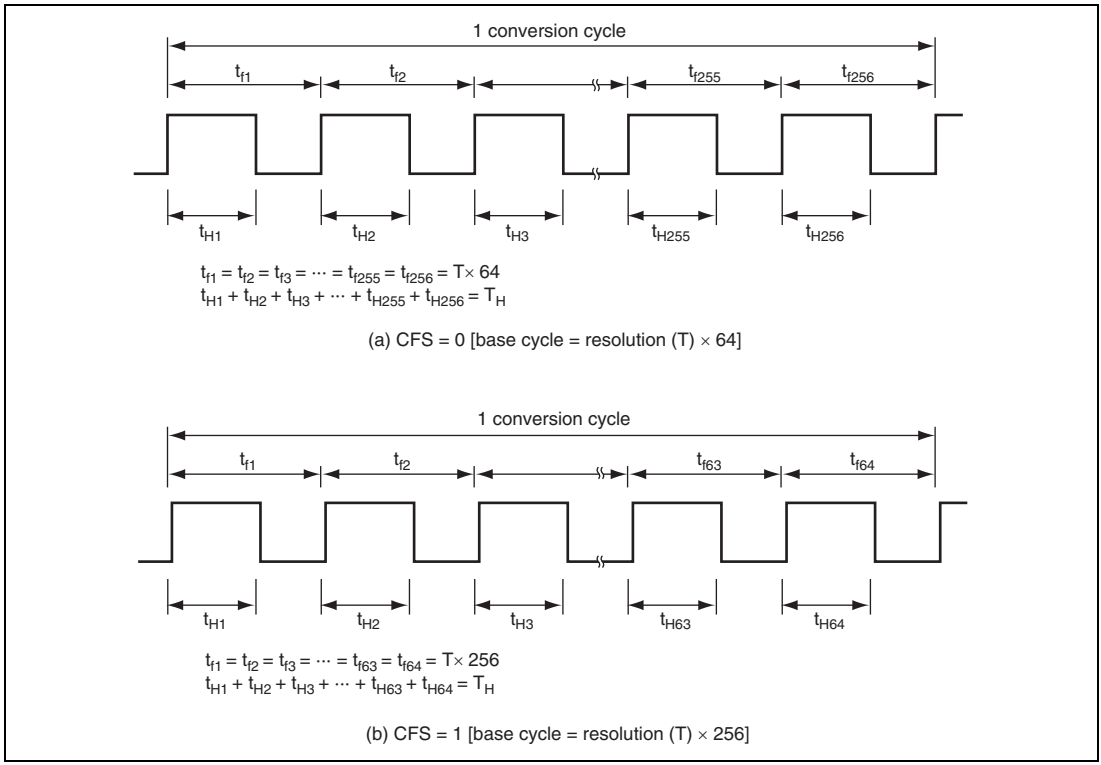


Figure 9.5 Output Waveform (OS = 1, DADR corresponds to T_H)

An example of the additional pulses when CFS = 1 (base cycle = resolution (T) × 256) and OS = 1 (inverted PWM output) is described below. When CFS = 1, the upper eight bits (DA13 to DA6) in DADR determine the duty cycle of the base pulse while the subsequent six bits (DA5 to DA0) determine the locations of the additional pulses as shown in figure 9.6.

Table 9.6 lists the locations of the additional pulses.

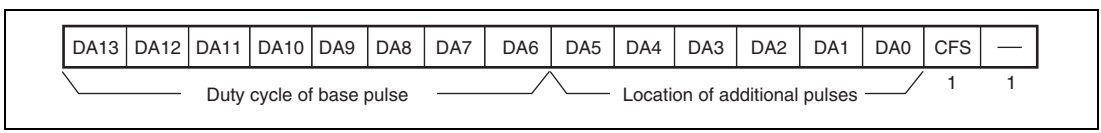


Figure 9.6 D/A Data Register Configuration when CFS = 1

In this example, DADR = H'0207 (B'0000 0010 0000 0111). The output waveform is shown in figure 9.7. Since CFS = 1 and the value of the upper eight bits is B'0000 0010, the high width of the base pulse duty cycle is 2/256 × (T).

Since the value of the subsequent six bits is B'0000 01, an additional pulse is output only at the location of base pulse No. 63 according to table 9.6. Thus, an additional pulse of 1/256 × (T) is to be added to the base pulse.

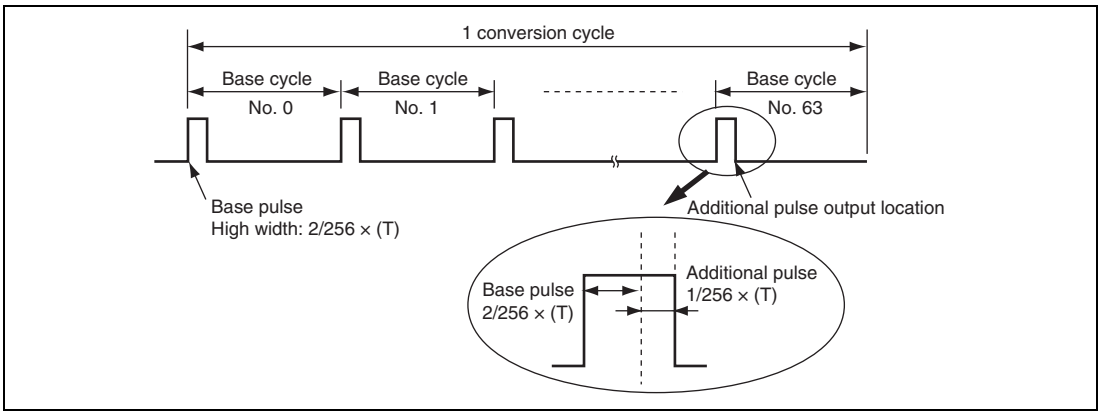
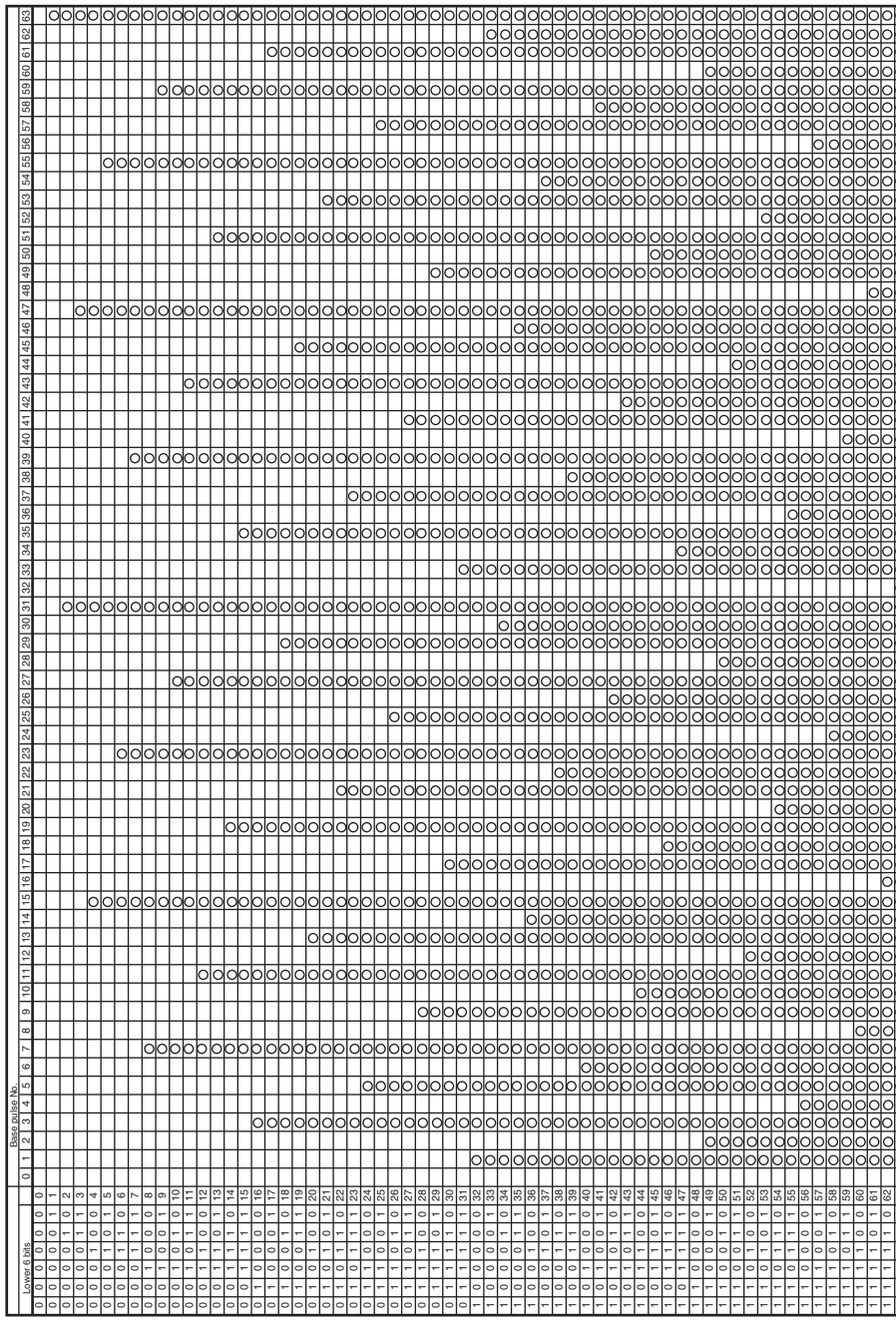


Figure 9.7 Output Waveform when DADR = H'0207 (OS = 1)

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

Table 9.6 Locations of Additional Pulses Added to Base Pulse (When CFS = 1)



9.6 Usage Notes

9.6.1 Module Stop Mode Setting

PWMX operation can be enabled or disabled by using the module stop control register. In the initial state, PWMX operation is disabled. Register access is enabled by clearing module stop mode. For details, see section 26, Power-Down Modes.

Section 10 16-Bit Timer Pulse Unit (TPU)

This LSI has an on-chip 16-bit timer pulse unit (TPU) that comprises three 16-bit timer channels. The function list of the 16-bit timer unit and its block diagram are shown in table 10.1 and figure 10.1, respectively.

10.1 Features

- Maximum 8-pulse input/output
- Selection of eight counter input clocks for channels 0 and 2, seven counter input clocks for channel 1
- The following operations can be set for each channel:
  - Waveform output at compare match
  - Input capture function
  - Counter clear operation
  - Multiple timer counters (TCNT) can be written to simultaneously
  - Simultaneous clearing by compare match and input capture possible
  - Register simultaneous input/output possible by counter synchronous operation
  - Maximum of 7-phase PWM output possible by combination with synchronous operation
- Buffer operation settable for channel 0
- Phase counting mode settable independently for each of channels 1 and 2
- Fast access via internal 16-bit bus
- 13 interrupt sources
- Automatic transfer of register data
- A/D converter conversion start trigger can be generated

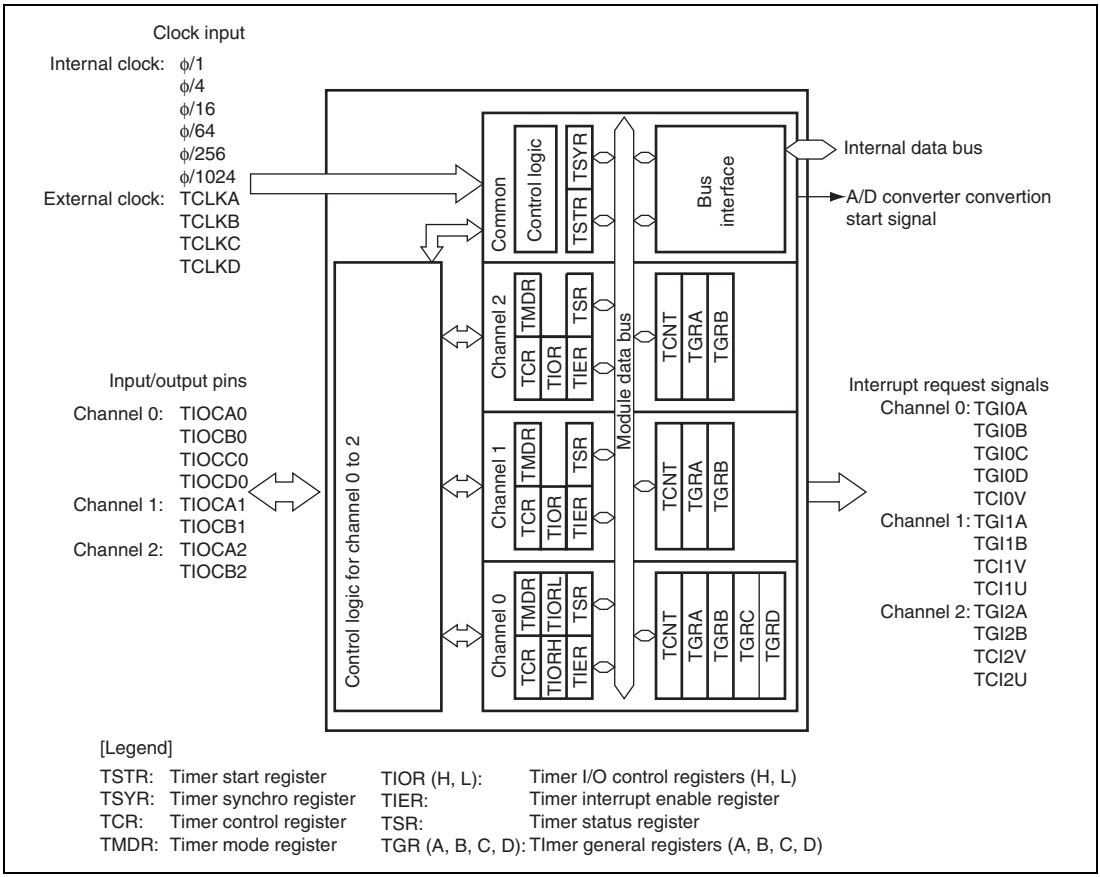


Figure 10.1 Block Diagram of TPU

Table 10.1 TPU Functions

Item		Channel 0	Channel 1	Channel 2
Count clock		φ/1	φ/1	φ/1
		φ/4	φ/4	φ/4
		φ/16	φ/16	φ/16
		φ/64	φ/64	φ/64
		TCLKA	φ/256	φ/1024
		TCLKB	TCLKA	TCLKA
		TCLKC	TCLKB	TCLKB
		TCLKD		TCLKC
General registers (TGR)		TGRA_0	TGRA_1	TGRA_2
		TGRB_0	TGRB_1	TGRB_2
General registers/buffer registers		TGRC_0	—	—
		TGRC_0		
I/O pins		TIOCA0	TIOCA1	TIOCA2
		TIOCB0	TIOCB1	TIOCB2
		TIOCC0		
		TIOCD0		
Counter clear function		TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture
Compare match output	0 output	O	O	O
	1 output	O	O	O
	Toggle output	O	O	O
Input capture function		O	O	O
Synchronous operation		O	O	O
PWM mode		O	O	O
Phase counting mode		—	O	O
Buffer operation		O	—	—

Item	Channel 0	Channel 1	Channel 2
A/D converter trigger	TGRA_0 compare match or input capture	TGRA_1 compare match or input capture	TGRA_2 compare match or input capture
Interrupt sources	5 sources • Compare match or input capture 0A • Compare match or input capture 0B • Compare match or input capture 0C • Compare match or input capture 0D • Overflow	4 sources • Compare match or input capture 1A • Compare match or input capture 1B • Overflow • Underflow	4 sources • Compare match or input capture 2A • Compare match or input capture 2B • Overflow • Underflow

[Legend]

O: Enable

—: Disable

10.2 Input/Output Pins

Table 10.2 Pin Configuration

Channel	Pin Name	I/O	Function
All	TCLKA	Input	External clock A input pin (Channel 1 phase counting mode A phase input)
	TCLKB	Input	External clock B input pin (Channel 1 phase counting mode B phase input)
	TCLKC	Input	External clock C input pin (Channel 2 phase counting mode A phase input)
	TCLKD	Input	External clock D input pin (Channel 2 phase counting mode B phase input)
0	TIOCA0	I/O	TGRA_0 input capture input/output compare output/PWM output pin
	TIOCB0	I/O	TGRB_0 input capture input/output compare output/PWM output pin
	TIOCC0	I/O	TGRC_0 input capture input/output compare output/PWM output pin
	TIOCD0	I/O	TGRD_0 input capture input/output compare output/PWM output pin
1	TIOCA1	I/O	TGRA_1 input capture input/output compare output/PWM output pin
	TIOCB1	I/O	TGRB_1 input capture input/output compare output/PWM output pin
2	TIOCA2	I/O	TGRA_2 input capture input/output compare output/PWM output pin
	TIOCB2	I/O	TGRA_2 input capture input/output compare output/PWM output pin

10.3 Register Descriptions

The TPU has the following registers.

Table 10.3 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Data Bus Width
Channel 0	Timer control register_0	TCR_0	R/W	H'00	H'FE50	8
	Timer mode register_0	TMDR_0	R/W	H'C0	H'FE51	8
	Timer I/O control register H_0	TIORH_0	R/W	H'00	H'FE52	8
	Timer I/O control register L_0	TIORL_0	R/W	H'00	H'FE53	8
	Timer interrupt enable register_0	TIER_0	R/W	H'40	H'FE54	8
	Timer status register_0	TSR_0	R/W	H'C0	H'FE55	8
	Timer counter_0	TCNT_0	R/W	H'0000	H'FE56	16
	Timer general register A_0	TGRA_0	R/W	H'FFFF	H'FE58	16
	Timer general register B_0	TGRB_0	R/W	H'FFFF	H'FE5A	16
	Timer general register C_0	TGRC_0	R/W	H'FFFF	H'FE5C	16
	Timer general register D_0	TGRD_0	R/W	H'FFFF	H'FE5E	16
Channel 1	Timer control register_1	TCR_1	R/W	H'00	H'FD40	8
	Timer mode register_1	TMDR_1	R/W	H'C0	H'FD41	8
	Timer I/O control register _1	TIOR_1	R/W	H'00	H'FD42	8
	Timer interrupt enable register_1	TIER_1	R/W	H'40	H'FD44	8
	Timer status register_1	TSR_1	R/W	H'C0	H'FD45	8
	Timer counter_1	TCNT_1	R/W	H'0000	H'FD46	16
	Timer general register A_1	TGRA_1	R/W	H'FFFF	H'FD48	16
	Timer general register B_1	TGRB_1	R/W	H'FFFF	H'FD4A	16
Channel 2	Timer control register_2	TCR_2	R/W	H'00	H'FE70	8
	Timer mode register_2	TMDR_2	R/W	H'C0	H'FE71	8
	Timer I/O control register_2	TIOR_2	R/W	H'00	H'FE72	8
	Timer interrupt enable register_2	TIER_2	R/W	H'40	H'FE74	8
	Timer status register_2	TSR_2	R/W	H'C0	H'FE75	8
	Timer counter_2	TCNT_2	R/W	H'0000	H'FE76	16

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Data Bus Width
Channel 2	Timer general register A_2	TGRA_2	R/W	H'FFFF	H'FE78	16
	Timer general register B_2	TGRB_2	R/W	H'FFFF	H'FE7A	16
Common	Timer start register	TSTR	R/W	H'00	H'FEB0	8
	Timer synchro register	TSYR	R/W	H'00	H'FEB1	8

10.3.1 Timer Control Register (TCR)

The TCR registers control the TCNT operation for each channel. The TPU has a total of three TCR registers, one for each channel (channel 0 to 2). TCR register settings should be made only when TCNT operation is stopped.

Bit	Bit Name	Initial value	R/W	Description
7	CCLR2	0	R/W	Counter Clear 2 to 0
6	CCLR1	0	R/W	These bits select the TCNT counter clearing source. See tables 10.4 and 10.5 for details.
5	CCLR0	0	R/W	
4	CKEG1	0	R/W	Clock Edge 1 and 0
3	CKEG0	0	R/W	These bits select the input clock edge. When the input clock is counted using both edges, the input clock cycle is divided in 2 (ϕ/4 both edges = ϕ/2 rising edge). If phase counting mode is used on channels 1, 2, 4, and 5, this setting is ignored and the phase counting mode setting has priority. Internal clock edge selection is valid when the input clock is ϕ/4 or slower. This setting is ignored if the input clock is ϕ/1 and rising edge count is selected. 00: Count at rising edge 01: Count at falling edge 1x: Count at both edges
2	TPSC2	0	R/W	Time Prescaler 2 to 0
1	TPSC1	0	R/W	These bits select the TCNT counter clock. The clock source can be selected independently for each channel. See tables 10.6 to 10.8 for details.
0	TPSC0	0	R/W	

[Legend]

x: Don't care

Table 10.4 CCLR2 to CCLR0 (channel 0)

Channel	Bit 7 CCLR2	Bit 6 CCLR1	Bit 5 CCLR0	Description
0	0	0	0	TCNT clearing disabled (Initial value)
			1	TCNT cleared by TGRA compare match/input capture
		1	0	TCNT cleared by TGRB compare match/input capture
			1	TCNT cleared by counter clearing for another channel performing synchronous/clearing synchronous operation*[1]
	1	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRC compare match/input capture*[2]
		1	0	TCNT cleared by TGRD compare match/input capture*[2]
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/synchronous operation*[1]

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

Table 10.5 CCLR2 to CCLR0 (channels 1 and 2)

Channel	Bit 7 Reserved*[2]	Bit 6 CCLR1	Bit 5 CCLR0	Description
1, 2	0	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRA compare match/input capture
		1	0	TCNT cleared by TGRB compare match/input capture
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/synchronous operation*[1]

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

Table 10.6 TPSC2 to TPSC0 (channel 0)

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

Table 10.7 TPSC2 to TPSC0 (channel 1)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
1	0	0	0	Internal clock: counts on ϕ
			1	Internal clock: counts on ϕ/4
		1	0	Internal clock: counts on ϕ/16
			1	Internal clock: counts on ϕ/64
	1	0	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKB pin input
		1	0	Internal clock: counts on ϕ/256
			1	Setting prohibited

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

Table 10.8 TPSC2 to TPSC0 (channel 2)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
2	0	0	0	Internal clock: counts on ϕ
			1	Internal clock: counts on ϕ/4
		1	0	Internal clock: counts on ϕ/16
			1	Internal clock: counts on ϕ/64
	1	0	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKB pin input
		1	0	External clock: counts on TCLKC pin input
			1	Internal clock: counts on ϕ/1024

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

10.3.2 Timer Mode Register (TMDR)

The TMDR registers are used to set the operating mode for each channel. The TPU has three TMDR registers, one for each channel. TMDR register settings should be made only when TCNT operation is stopped.

Bit	Bit Name	Initial value	R/W	Description
7	—	1	R	Reserved
6	—	1	R	These bits are always read as 1 and cannot be modified.
5	BFB	0	R/W	Buffer Operation B Specifies whether TGRB is to operate in the normal way, or TGRB and TGRD are to be used together for buffer operation. When TGRD is used as a buffer register. TGRD input capture/output compare is not generation. Because channels 1 and 2 have no TGRD, bit 5 is reserved. It is always read as 0 and cannot be modified. 0: TGRB operates normally 1: TGRB and TGRD used together for buffer operation
4	BFA	0	R/W	Buffer Operation A Specifies whether TGRA is to operate in the normal way, or TGRA and TGRC are to be used together for buffer operation. When TGRC is used as a buffer register, TGRC input capture/output compare is not generated. Because channels 1 and 2 have no TGRC, bit 4 is reserved. It is always read as 0 and cannot be modified. 0: TGRA operates normally 1: TGRA and TGRC used together for buffer operation
3	MD3	0	R/W	Modes 3 to 0 These bits are used to set the timer operating mode. MD3 is a reserved bit. In a write, the write value should always be 0. See table 10.9 for details.
2	MD2	0	R/W	
1	MD1	0	R/W	
0	MD0	0	R/W	

Table 10.9 MD3 to MD0

Bit 3 MD3*[1]	Bit2 MD2*[2]	Bit 1 MD1	Bit 0 MD0	Description
0	0	0	0	Normal operation
			1	Reserved
		1	0	PWM mode 1
			1	PWM mode 2
	1	0	0	Phase counting mode 1
			1	Phase counting mode 2
		1	0	Phase counting mode 3
			1	Phase counting mode 4
1	×	×	×	Setting prohibited

[Legend]

x: Don't care

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

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

10.3.3 Timer I/O Control Register (TIOR)

The TIOR registers control the TGR registers. The TPU has four TIOR registers, two each for channels 0, and one each for channels 1 and 2. Care is required since TIOR is affected by the TMDR setting. The initial output specified by TIOR is valid when the counter is stopped (the CST bit in TSTR is cleared to 0). Note also that, in PWM mode 2, the output at the point at which the counter is cleared to 0 is specified. When TGRC or TGRD is designated for buffer operation, this setting is invalid and the register operates as a buffer register.

- TIORH_0, TIOR_1, TIOR_2

Bit	Bit Name	Initial value	R/W	Description
7	IOB3	0	R/W	I/O Control B3 to B0
6	IOB2	0	R/W	Specify the function of TGRB.
5	IOB1	0	R/W	
4	IOB0	0	R/W	
3	IOA3	0	R/W	I/O Control A3 to A0
2	IOA2	0	R/W	Specify the function of TGRA.
1	IOA1	0	R/W	
0	IOA0	0	R/W	

- TIORL_0

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

Table 10.10 TIORH_0 (channel 0)

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_0 Function**	**TIOCB0 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCB0 pin Input capture at rising edge
			1		Capture input source is TIOCB0 pin Input capture at falling edge
		1	×		Capture input source is TIOCB0 pin Input capture at both edges
	1	×	×		Setting prohibited

[Legend]

×: Don't care

Table 10.11 TIORH_0 (channel 0)

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_0 Function**	**TIOCA0 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA0 pin Input capture at rising edge
			1		Capture input source is TIOCA0 pin Input capture at falling edge
		1	×		Capture input source is TIOCA0 pin Input capture at both edges
	1	×	×		Setting prohibited

[Legend]

×: Don't care

Table 10.12 TIORL_0 (channel 0)

				Description	
Bit 7 IOD3	**Bit 6 IOD2**	**Bit 5 IOD1**	**Bit 4 IOD0**	**TGRD_0 Function**	**TIOCD0 Pin Function**
0	0	0	0	Output Compare register*	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register*	Capture input source is TIOCD0 pin Input capture at rising edge
			1		Capture input source is TIOCD0 pin Input capture at falling edge
		1	×		Capture input source is TIOCD0 pin Input capture at both edges
	1	×	×		Setting prohibited

[Legend]

×: Don't care

Note: When the BFB bit in TMDR_0 is set to 1 and TGRD_0 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 10.13 TIORL_0 (channel 0)

				Description	
Bit 3 IOC3	**Bit 2 IOC2**	**Bit 1 IOC1**	**Bit 1 IOC0**	**TGRC_0 Function**	**TIOCA0 Pin Function**
0	0	0	0	Output compare register*	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register*	Capture input source is TIOCA0 pin Input capture at rising edge
			1		Capture input source is TIOCA0 pin Input capture at falling edge
		1	×		Capture input source is TIOCA0 pin Input capture at both edges
	1	×	×		Setting prohibited

[Legend]

×: Don't care

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

Table 10.14 TIOR_1 (channel 1)

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_1 Function**	**TIOCB1 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCB1 pin Input capture at rising edge
			1		Capture input source is TIOCB1 pin Input capture at falling edge
		1	×		Capture input source is TIOCB1 pin Input capture at both edges
	1	×	×		Setting prohibited

[Legend]

×: Don't care

Table 10.15 TIOR_1 (channel 1)

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_1 Function**	**TIOCA1 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA0 pin Input capture at rising edge
			1		Capture input source is TIOCA0 pin Input capture at falling edge
		1	×		Capture input source is TIOCA0 pin Input capture at both edges
	1	×	×		Setting prohibited

[Legend]

×: Don't care

Table 10.16 TIOR_2 (channel 2)

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_2 Function**	**TIOCB2 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	×	0	0	Input capture register	Capture input source is TIOCB2 pin Input capture at rising edge
			1		Capture input source is TIOCB2 pin Input capture at falling edge
		1	×		Capture input source is TIOCB2 pin Input capture at both edges

[Legend]

×: Don't care

Table 10.17 TIOR_2 (channel 2)

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_2 Function**	**TIOCA2 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	×	0	0	Input capture register	Capture input source is TIOCA2 pin Input capture at rising edge
			1		Capture input source is TIOCA2 pin Input capture at falling edge
		1	×		Capture input source is TIOCA2 pin Input capture at both edges

[Legend]

×: Don't care

10.3.4 Timer Interrupt Enable Register (TIER)

The TIER registers control enabling or disabling of interrupt requests for each channel. The TPU has three TIER registers, one for each channel.

Bit	Bit Name	Initial value	R/W	Description
7	TTGE	0	R/W	A/D Conversion Start Request Enable Enables or disables generation of A/D conversion start requests by TGRA input capture/compare match. 0: A/D conversion start request generation disabled 1: A/D conversion start request generation enabled
6	—	1	R	Reserved This bit is always read as 1 and cannot be modified.
5	TCIEU	0	R/W	Underflow Interrupt Enable Enables or disables interrupt requests (TCIU) by the TCFU flag when the TCFU flag in TSR is set to 1 in channels 1 and 2. In channel 0, bit 5 is reserved. 0: Interrupt requests (TCIU) by TCFU disabled 1: Interrupt requests (TCIU) by TCFU enabled
4	TCIEV	0	R/W	Overflow Interrupt Enable Enables or disables interrupt requests (TCIV) by the TCFV flag when the TCFV flag in TSR is set to 1. 0: Interrupt requests (TCIV) by TCFV disabled 1: Interrupt requests (TCIV) by TCFV enabled
3	TGIED	0	R/W	TGR Interrupt Enable D Enables or disables interrupt requests (TGID) by the TGFD bit when the TGFD bit in TSR is set to 1 in channel 0. In channels 1 and 2, bit 3 is reserved. It is always read as 0 and cannot be modified. 0: Interrupt requests (TGID) by TGFD disabled 1: Interrupt requests (TGID) by TGFD enabled.

Bit	Bit Name	Initial value	R/W	Description
2	TGIEC	0	R/W	TGR Interrupt Enable C Enables or disables interrupt requests (TGIC) by the TGFC bit when the TGFC bit in TSR is set to 1 in channel 0. In channels 1 and 2, bit 2 is reserved. It is always read as 0 and cannot be modified. 0: Interrupt requests (TGIC) by TGFC disabled 1: Interrupt requests (TGIC) by TGFC enabled
1	TGIEB	0	R/W	TGR Interrupt Enable B Enables or disables interrupt requests (TGIB) by the TGFB bit when the TGFB bit in TSR is set to 1. 0: Interrupt requests (TGIB) by TGFB disabled 1: Interrupt requests (TGIB) by TGFB enabled
0	TGIEA	0	R/W	TGR Interrupt Enable A Enables or disables interrupt requests (TGIA) by the TGFA bit when the TGFA bit in TSR is set to 1. 0: Interrupt requests (TGIA) by TGFA disabled 1: Interrupt requests (TGIA) by TGFA enabled

10.3.5 Timer Status Register (TSR)

The TSR registers indicate the status of each channel. The TPU has three TSR registers, one for each channel.

Bit	Bit Name	Initial value	R/W	Description
7	TCFD	1	R	Count Direction Flag Status flag that shows the direction in which TCNT counts in channel 1 and 2. In channel 0, bit 7 is reserved. It is always read as 0 and cannot be modified. 0: TCNT counts down 1: TCNT counts up
6	—	1	R	Reserved This bit is always read as 1 and cannot be modified.
5	TCFU	0	R/(W)*	Underflow Flag Status flag that indicates that TCNT underflow has occurred when channels 1 and 2 are set to phase counting mode. In channel 0, bit 5 is reserved. It is always read as 0 and cannot be modified. [Setting condition] When the TCNT value underflows (change from H'0000 to H'FFFF) [Clearing condition] When 0 is written to TCFU after reading TCFU = 1
4	TCFV	0	R/(W) *	Overflow Flag Status flag that indicates that TCNT overflow has occurred. [Setting condition] When the TCNT value overflows (change from H'FFFF to H'0000) [Clearing condition] When 0 is written to TCFV after reading TCFV = 1

Bit	Bit Name	Initial value	R/W	Description
3	TGFD	0	R/(W)*	Input Capture/Output Compare Flag D Status flag that indicates the occurrence of TGRD input capture or compare match in channel 0. In channels 1 and 2, bit 3 is reserved. It is always read as 0 and cannot be modified. [Setting conditions] • When TCNT = TGRD while TGRD is functioning as output compare register • When TCNT value is transferred to TGRD by input capture signal while TGRD is functioning as input capture register [Clearing condition] When 0 is written to TGFD after reading TGFD = 1
2	TGFC	0	R/(W)*	Input Capture/Output Compare Flag C Status flag that indicates the occurrence of TGRC input capture or compare match in channel 0. In channels 1 and 2, bit 2 is reserved. It is always read as 0 and cannot be modified. [Setting conditions] • When the TCNT = TGRC while TGRC is functioning as output compare register • When TCNT value is transferred to TGRC by input capture signal while TGRC is functioning as input capture register [Clearing condition] When 0 is written to TGFC after reading TGFC = 1

Bit	Bit Name	Initial value	R/W	Description
1	TGFB	0	R/(W)*	Input Capture/Output Compare Flag B Status flag that indicates the occurrence of TGRB input capture or compare match. [Setting conditions] • When TCNT = TGRB while TGRB is functioning as output compare register • When TCNT value is transferred to TGRB by input capture signal while TGRB is functioning as input capture register [Clearing condition] When 0 is written to TGFB after reading TGFB = 1
0	TGFA	0	R/(W)*	Input Capture/Output Compare Flag A Status flag that indicates the occurrence of TGRA input capture or compare match. The write value should always be 0 to clear this flag. [Setting conditions] • When TCNT = TGRA while TGRA is functioning as output compare register • When TCNT value is transferred to TGRA by input capture signal while TGRA is functioning as input capture register [Clearing condition] When 0 is written to TGFA after reading TGFA = 1

Note: * The write value should always be 0 to clear the flag.

10.3.6 Timer Counter (TCNT)

The TCNT registers are 16-bit counters. The TPU has three TCNT counters, one for each channel. The TCNT counters are initialized to H'0000 by a reset. The TCNT counters cannot be accessed in 8-bit units; they must always be accessed as a 16-bit unit.

10.3.7 Timer General Register (TGR)

The TGR registers are 16-bit registers with a dual function as output compare and input capture registers. The TPU has 16 TGR registers, four for channel 0 and two each for channels 1 and 2. TGRC and TGRD for channel 0 can also be designated for operation as buffer registers. The TGR registers are initialized to H'FFFF by a reset. The TGR registers cannot be accessed in 8-bit units; they must always be accessed as a 16-bit unit. TGR buffer register combinations are TGRA—TGRC and TGRB—TGRD.

10.3.8 Timer Start Register (TSTR)

TSTR is an 8-bit readable/writable register that selects operation/stoppage for channels 0 to 2. TCNT of a channel performs counting when the corresponding bit in TSTR is set to 1. When setting the operating mode in TMDR or setting the count clock in TCR, first stop the TCNT counter.

Bit	Bit Name	Initial value	R/W	Description
7 to 3	—	0	R	Reserved The initial value should not be changed.
2	CST2	0	R/W	Counter Start 2 to 0 (CST2 to CST0)
1	CST1	0	R/W	These bits select operation or stoppage for TCNT.
0	CST0	0	R/W	If 0 is written to the CST bit during operation with the TIOC pin designated for output, the counter stops but the TIOC pin output compare output level is retained. If TIOR is written to when the CST bit is cleared to 0, the pin output level will be changed to the set initial output value. 0: TCNT_n count operation is stopped 1: TCNT_n performs count operation (n = 2 to 0)

10.3.9 Timer Synchro Register (TSYR)

TSYR selects independent operation or synchronous operation for the channel 0 to 2 TCNT counters. A channel performs synchronous operation when the corresponding bit in TSYR is set to 1.

Bit	Bit Name	Initial value	R/W	Description
7 to 3	—	0	R/W	Reserved The initial value should not be changed.
2	SYNC2	0	R/W	Timer Synchro 2 to 0
1	SYNC 1	0	R/W	These bits select whether operation is independent of or synchronized with other channels. When synchronous operation is selected, synchronous presetting of multiple channels, and synchronous clearing through counter clearing on another channel are possible. To set synchronous operation, the SYNC bits for at least two channels must be set to 1. To set synchronous clearing, in addition to the SYNC bit, the TCNT clearing source must also be set by means of bits CCLR2 to CCLR0 in TCR. 0: TCNT_n operates independently (TCNT presetting /clearing is unrelated to other channels) 1: TCNT_n performs synchronous operation TCNT synchronous presetting/synchronous clearing is possible (n = 2 to 0)
0	SYNC 0	0	R/W	

10.4 Interface to Bus Master

10.4.1 16-Bit Registers

TCNT and TGR are 16-bit registers. As the data bus to the bus master is 16 bits wide, these registers can be read and written to in 16-bit units.

These registers cannot be read from or written to in 8-bit units; 16-bit access must always be used.

An example of 16-bit register access operation is shown in figure 10.2.

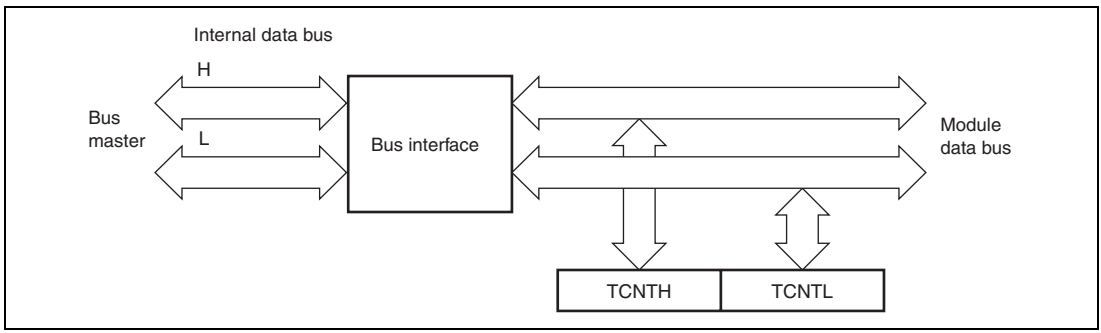


Figure 10.2 16-Bit Register Access Operation [Bus Master ↔ TCNT (16 Bits)]

10.4.2 8-Bit Registers

Registers other than TCNT and TGR are 8-bit. As the data bus to the CPU is 16 bits wide, these registers can be read and written to in 16-bit units. They can also be read and written to in 8-bit units.

Examples of 8-bit register access operation are shown in figures 10.3, 10.4, and 10.5.

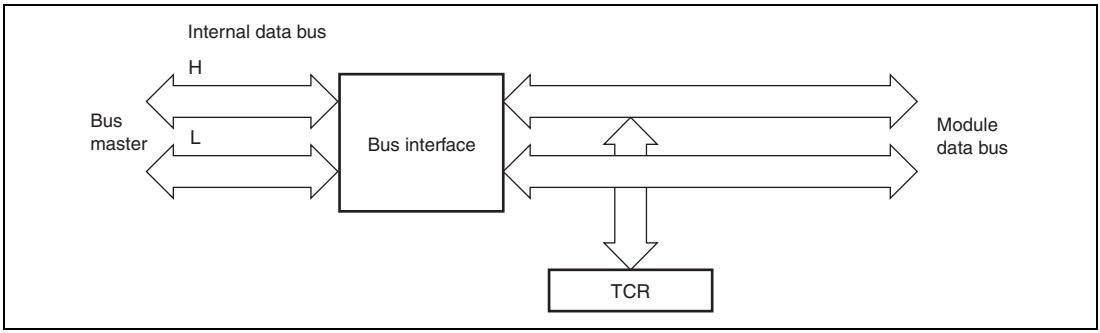


Figure 10.3 8-Bit Register Access Operation [Bus Master ↔ TCR (Upper 8 Bits)]

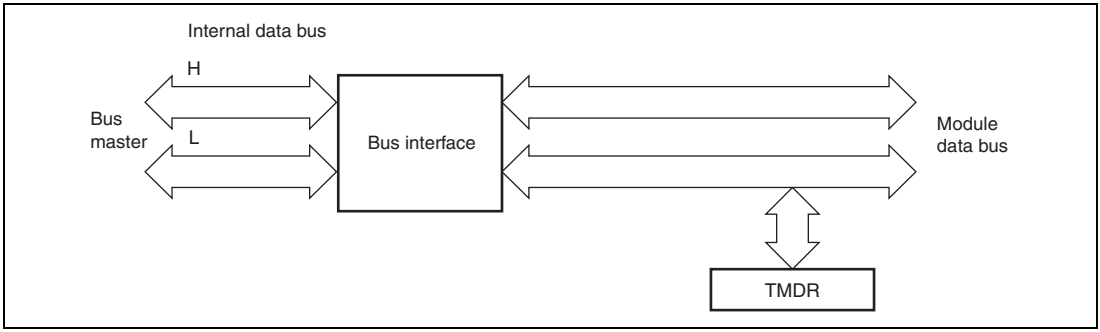


Figure 10.4 8-Bit Register Access Operation [Bus Master ↔ TMDR (Lower 8 Bits)]

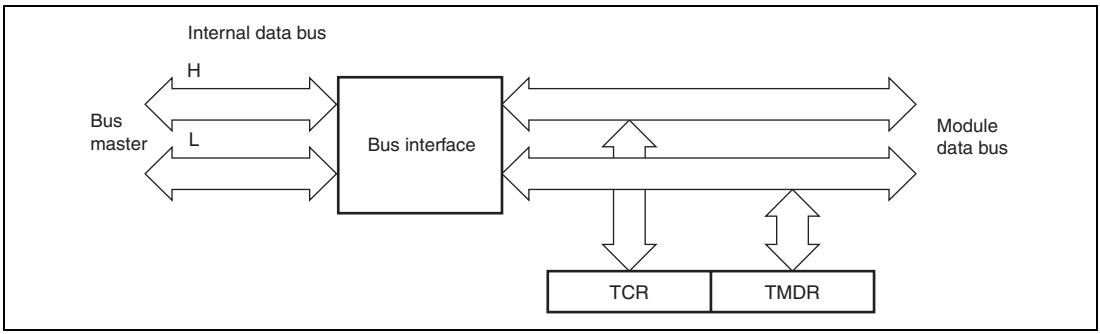


Figure 10.5 8-Bit Register Access Operation [Bus Master ↔ TCR and TMDR (16 Bits)]

10.5 Operation

10.5.1 Basic Functions

Each channel has a TCNT and TGR. TCNT performs up-counting, and is also capable of free-running operation, synchronous counting, and external event counting. Each TGR can be used as an input capture register or output compare register.

(1) Counter Operation

When one of bits CST0 to CST2 is set to 1 in TSTR, the TCNT counter for the corresponding channel starts counting. TCNT can operate as a free-running counter, periodic counter, and so on.

(a) Example of count operation setting procedure

Figure 10.6 shows an example of the count operation setting procedure.

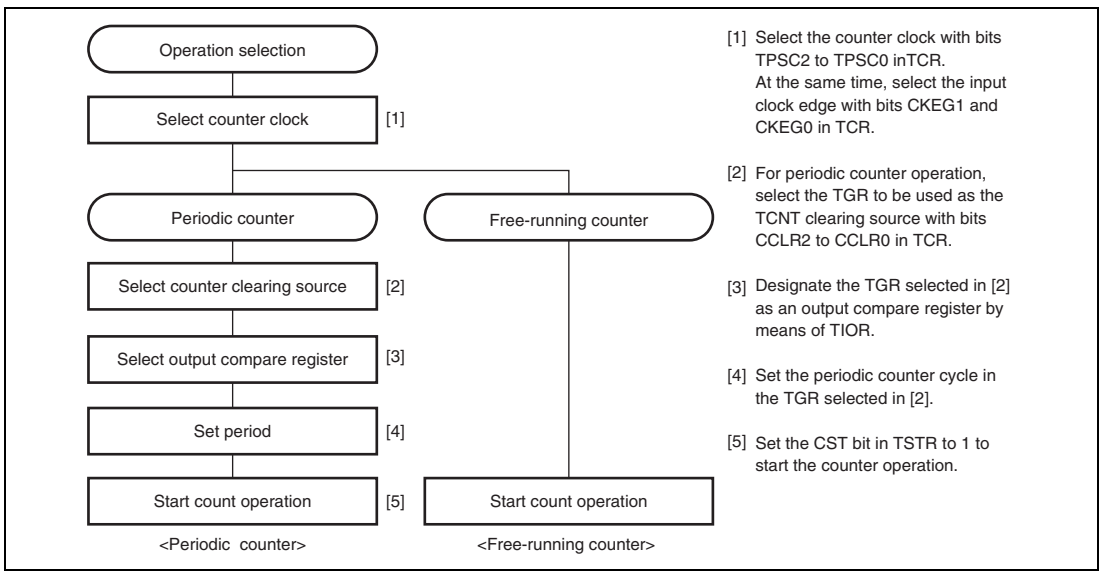


Figure 10.6 Example of Counter Operation Setting Procedure

(b) Free-running count operation and periodic count operation

Immediately after a reset, the TPU's TCNT counters are all designated as free-running counters. When the relevant bit in TSTR is set to 1 the corresponding TCNT counter starts up-count operation as a free-running counter. When TCNT overflows (from H'FFFF to H'0000), the TCFV bit in TSR is set to 1. If the value of the corresponding TCIEV bit in TIER is 1 at this point, the TPU requests an interrupt. After overflow, TCNT starts counting up again from H'0000. Figure 10.7 illustrates free-running counter operation.

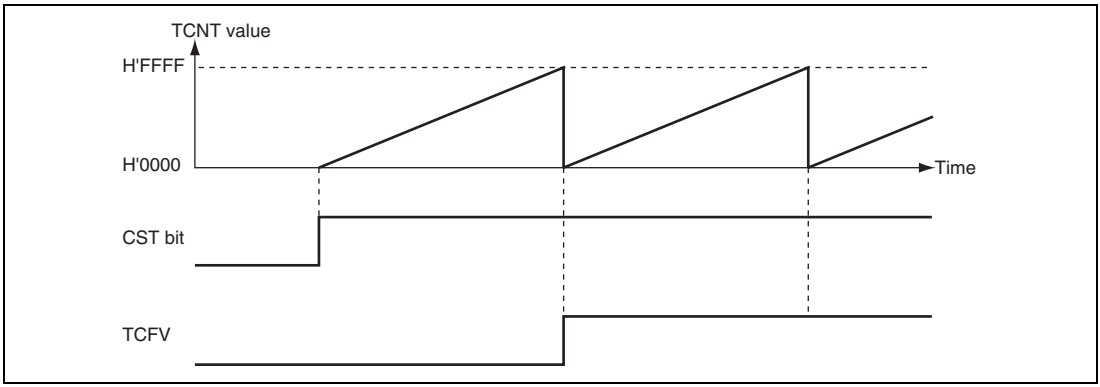


Figure 10.7 Free-Running Counter Operation

When compare match is selected as the TCNT clearing source, the TCNT counter for the relevant channel performs periodic count operation. The TGR register for setting the period is designated as an output compare register, and counter clearing by compare match is selected by means of bits CCLR2 to CCLR0 in TCR. After the settings have been made, TCNT starts up-count operation as periodic counter when the corresponding bit in TSTR is set to 1. When the count value matches the value in TGR, the TGF bit in TSR is set to 1 and TCNT is cleared to H'0000. If the value of the corresponding TGIE bit in TIER is 1 at this point, the TPU requests an interrupt. After a compare match, TCNT starts counting up again from H'0000. Figure 10.8 illustrates periodic counter operation.

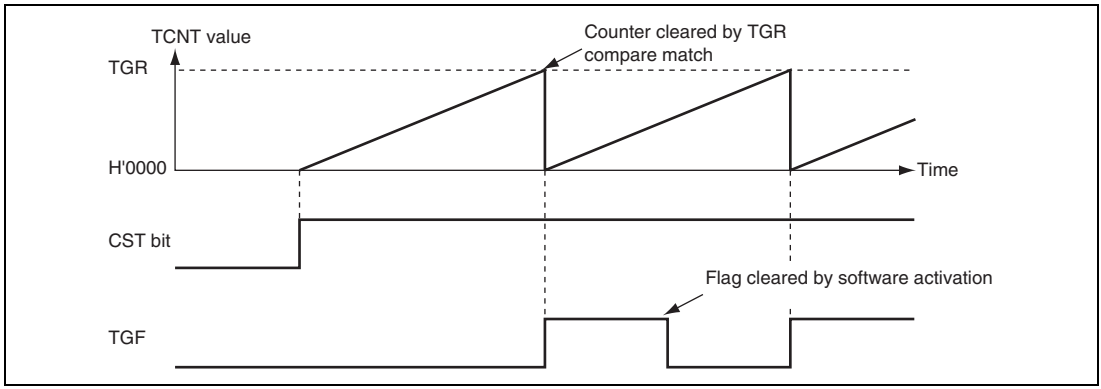


Figure 10.8 Periodic Counter Operation

(2) Waveform Output by Compare Match

The TPU can perform 0, 1, or toggle output from the corresponding output pin using compare match.

(a) Example of setting procedure for waveform output by compare match

Figure 10.9 shows an example of the setting procedure for waveform output by compare match.

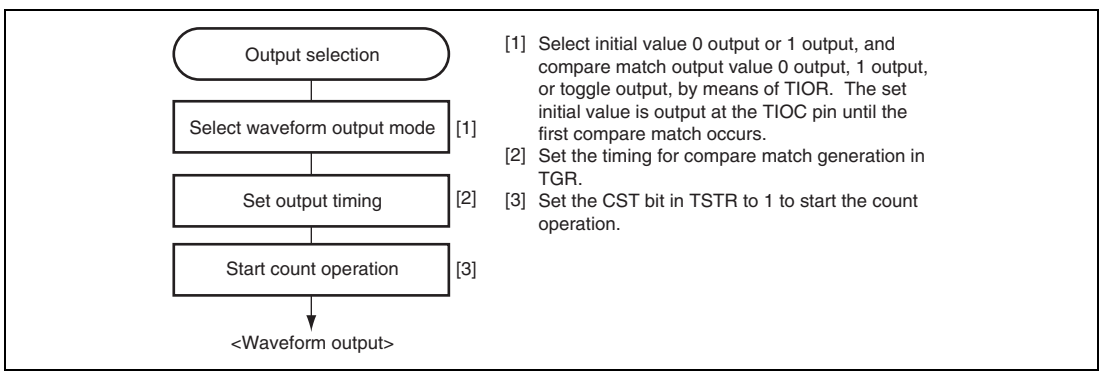


Figure 10.9 Example of Setting Procedure for Waveform Output by Compare Match

(b) Examples of waveform output operation

Figure 10.10 shows an example of 0 output/1 output. In this example TCNT has been designated as a free-running counter, and settings have been made so that 1 is output by compare match A, and 0 is output by compare match B. When the set level and the pin level coincide, the pin level does not change.

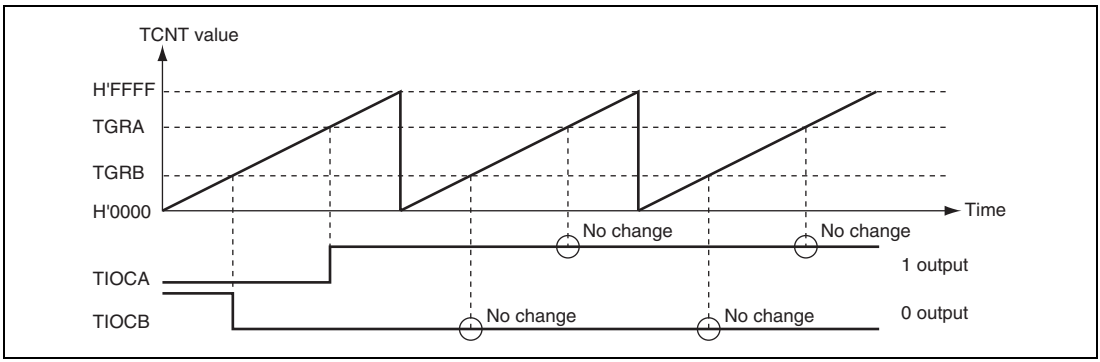


Figure 10.10 Example of 0 Output/1 Output Operation

Figure 10.11 shows an example of toggle output.

In this example TCNT has been designated as a periodic counter (with counter clearing performed by compare match B), and settings have been made so that output is toggled by both compare match A and compare match B.

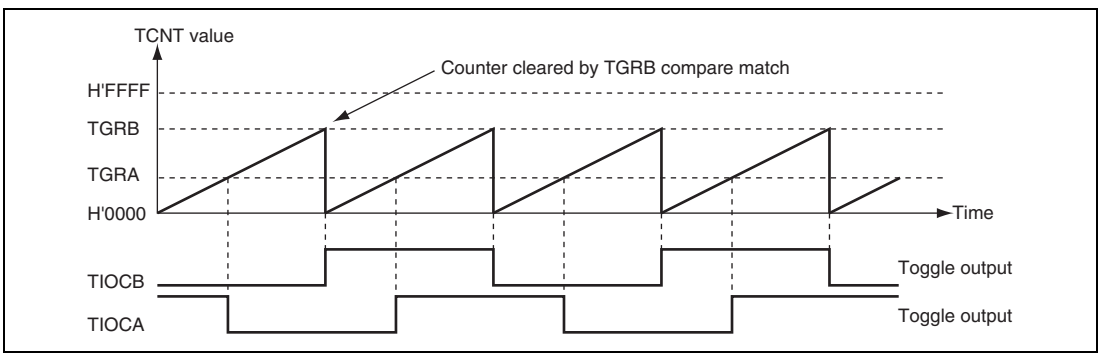


Figure 10.11 Example of Toggle Output Operation

(3) Input Capture Function

The TCNT value can be transferred to TGR on detection of the TIOC pin input edge. Rising edge, falling edge, or both edges can be selected as the detected edge.

(a) Example of input capture operation setting procedure

Figure 10.12 shows an example of the input capture operation setting procedure.

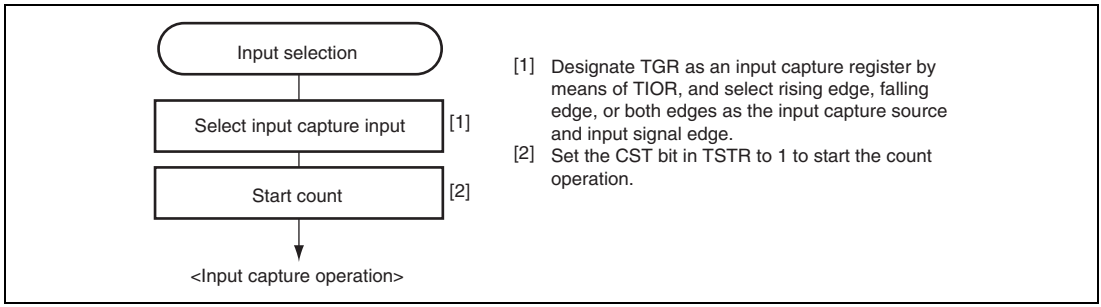


Figure 10.12 Example of Input Capture Operation Setting Procedure

(b) Example of input capture operation

Figure 10.13 shows an example of input capture operation. In this example both rising and falling edges have been selected as the TIOCA pin input capture input edge, falling edge has been selected as the TIOCB pin input capture input edge, and counter clearing by TGRB input capture has been designated for TCNT.

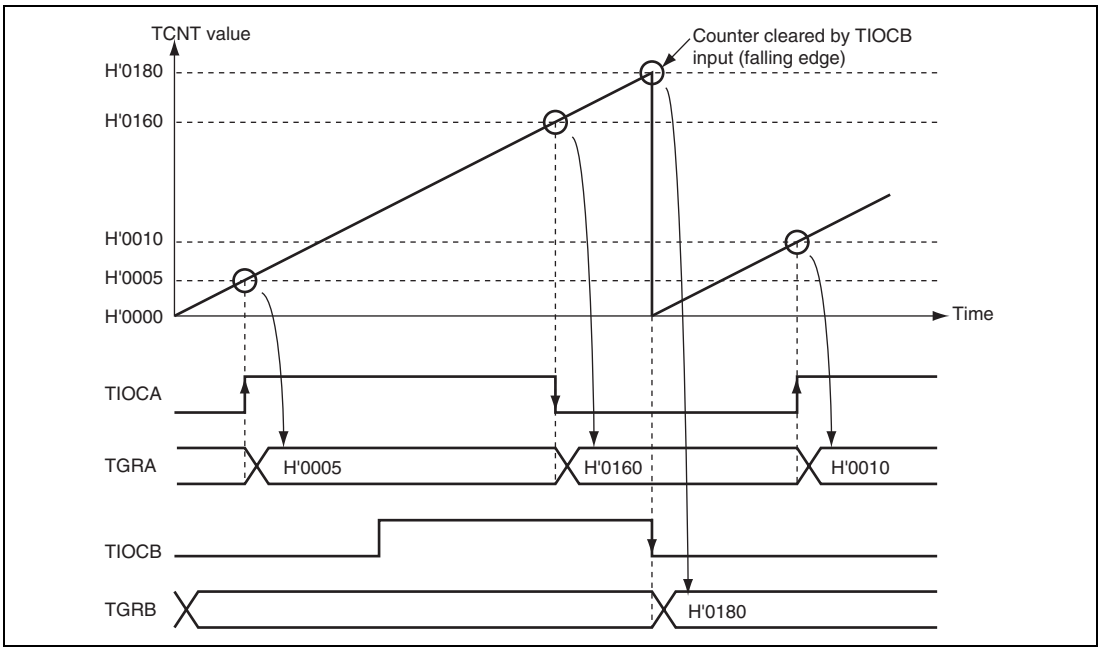


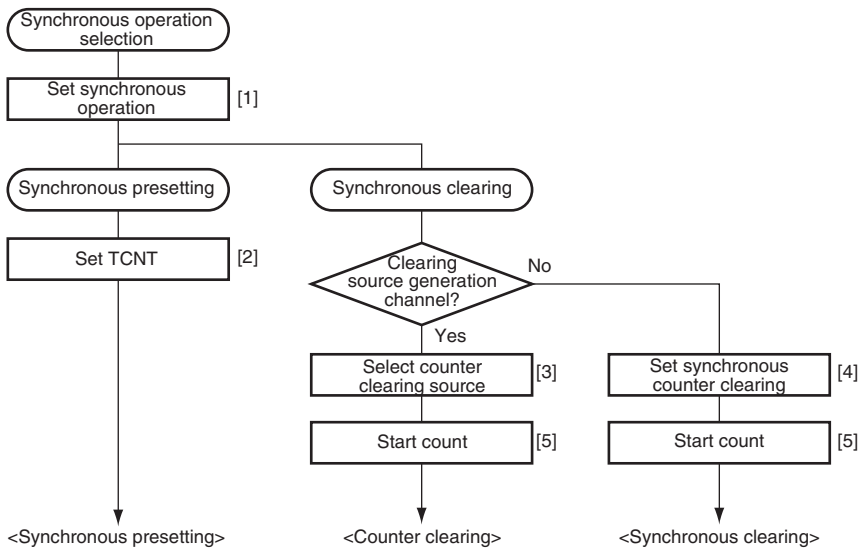
Figure 10.13 Example of Input Capture Operation

10.5.2 Synchronous Operation

In synchronous operation, the values in a number of TCNT counters can be rewritten simultaneously (synchronous presetting). Also, a number of TCNT counters can be cleared simultaneously by making the appropriate setting in TCR (synchronous clearing). Synchronous operation enables TGR to be incremented with respect to a single time base. Channels 0 to 2 can all be designated for synchronous operation.

(1) Example of Synchronous Operation Setting Procedure

Figure 10.14 shows an example of the synchronous operation setting procedure.



[1] Set to 1 the SYNC bits in TSYR corresponding to the channels to be designated for synchronous operation.
[2] When the TCNT counter of any of the channels designated for synchronous operation is written to, the same value is simultaneously written to the other TCNT counters.
[3] Use bits CCLR2 to CCLR0 in TCR to specify TCNT clearing by input capture/output compare, etc.
[4] Use bits CCLR2 to CCLR0 in TCR to designate synchronous clearing for the counter clearing source.
[5] Set to 1 the CST bits in TSTR for the relevant channels, to start the count operation.

Figure 10.14 Example of Synchronous Operation Setting Procedure

(2) Example of Synchronous Operation

Figure 10.15 shows an example of synchronous operation.

In this example, synchronous operation and PWM mode 1 have been designated for channels 0 to 2, TGRB_0 compare match has been set as the channel 0 counter clearing source, and synchronous clearing has been set for the channel 1 and 2 counter clearing source. Three-phase PWM waveforms are output from pins TIOC0A, TIOC1A, and TIOC2A. At this time, synchronous presetting, and synchronous clearing by TGRB_0 compare match, is performed for channel 0 to 2 TCNT counters, and the data set in TGRB_0 is used as the PWM cycle. For details of PWM modes, see section 10.5.4, PWM Modes.

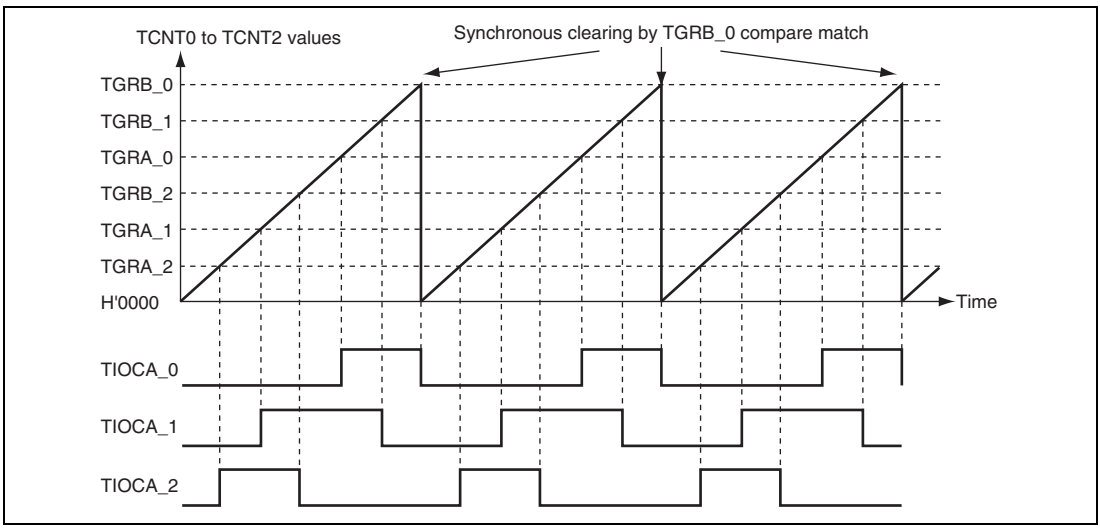


Figure 10.15 Example of Synchronous Operation

10.5.3 Buffer Operation

Buffer operation, provided for channels 0 and 3, enables TGRC and TGRD to be used as buffer registers. Buffer operation differs depending on whether TGR has been designated as an input capture register or as a compare match register. Table 10.18 shows the register combinations used in buffer operation.

Table 10.18 Register Combinations in Buffer Operation

Channel	Timer General Register	Buffer Register
0	TGRA_0	TGRC_0
	TGRB_0	TGRD_0

- When TGR is an output compare register

  When a compare match occurs, the value in the buffer register for the corresponding channel is transferred to the timer general register. This operation is illustrated in figure 10.16.

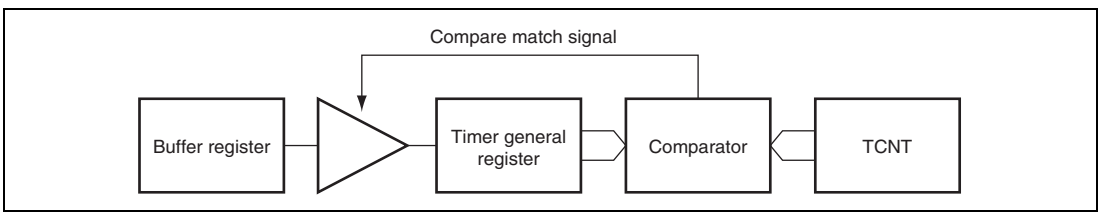


Figure 10.16 Compare Match Buffer Operation

- When TGR is an input capture register

  When input capture occurs, the value in TCNT is transferred to TGR and the value previously held in the timer general register is transferred to the buffer register. This operation is illustrated in figure 10.17.

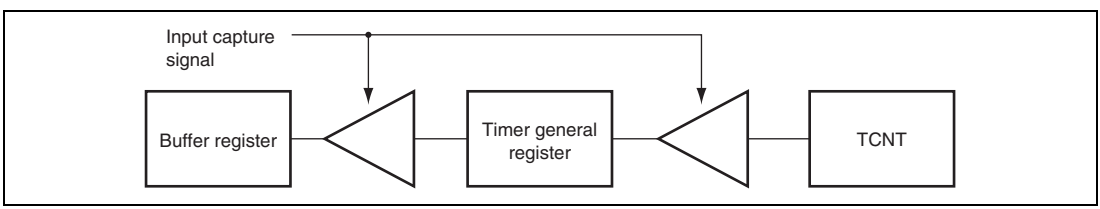


Figure 10.17 Input Capture Buffer Operation

(1) Example of Buffer Operation Setting Procedure

Figure 10.18 shows an example of the buffer operation setting procedure.

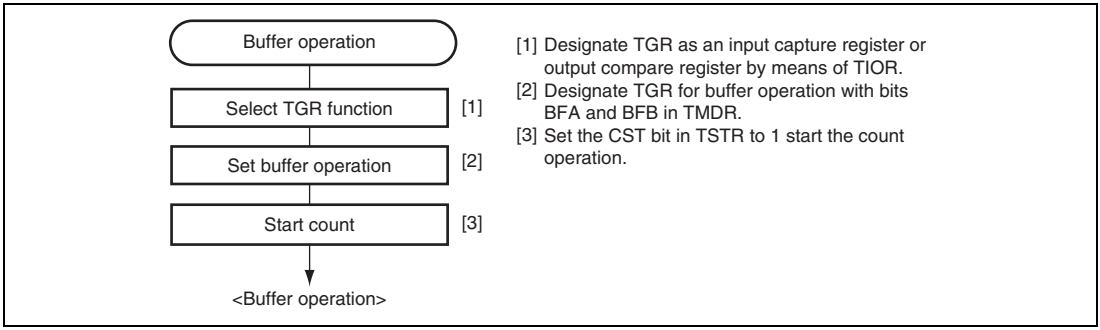


Figure 10.18 Example of Buffer Operation Setting Procedure

(2) Examples of Buffer Operation

(a) When TGR is an output compare register

Figure 10.19 shows an operation example in which PWM mode 1 has been designated for channel 0, and buffer operation has been designated for TGRA and TGRC. The settings used in this example are TCNT clearing by compare match B, 1 output at compare match A, and 0 output at compare match B. As buffer operation has been set, when compare match A occurs the output changes and the value in buffer register TGRC is simultaneously transferred to timer general register TGRA. This operation is repeated each time compare match A occurs. For details of PWM modes, see section 10.5.4, PWM Modes.

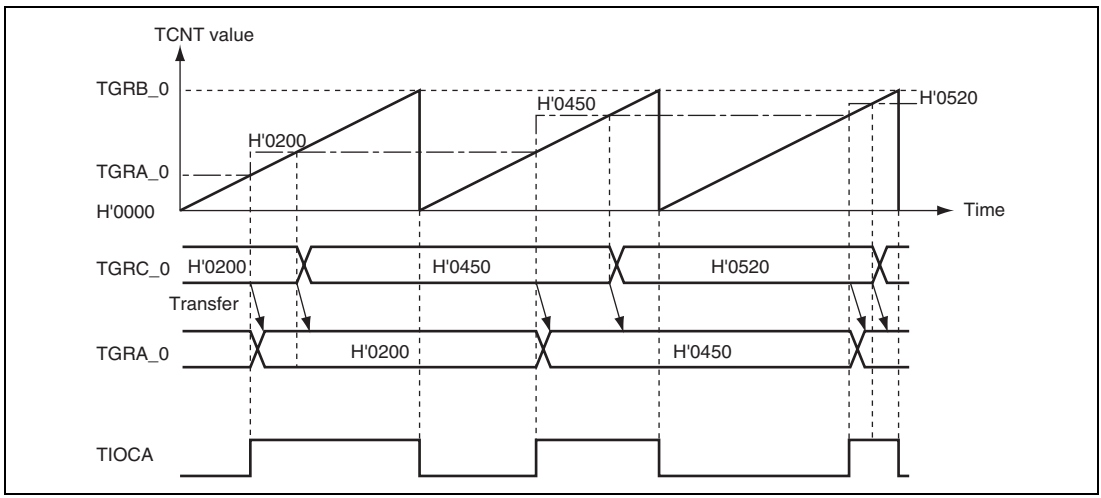


Figure 10.19 Example of Buffer Operation (1)

(b) When TGR is an input capture register

Figure 10.20 shows an operation example in which TGRA has been designated as an input capture register, and buffer operation has been designated for TGRA and TGRC. Counter clearing by TGRA input capture has been set for TCNT, and both rising and falling edges have been selected as the TIOCA pin input capture input edge. As buffer operation has been set, when the TCNT value is stored in TGRA upon occurrence of input capture A, the value previously stored in TGRA is simultaneously transferred to TGRC.

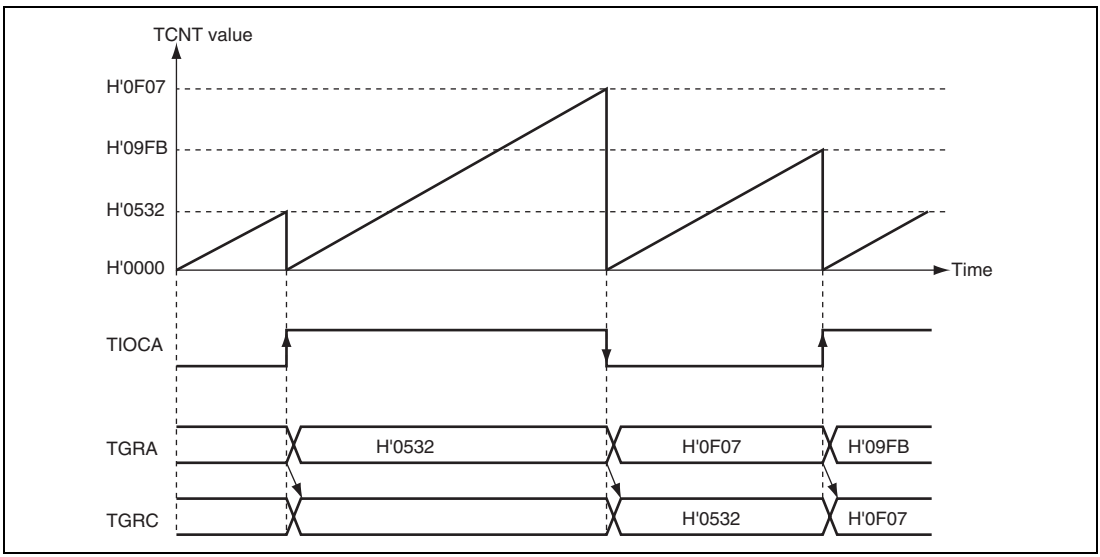


Figure 10.20 Example of Buffer Operation (2)

10.5.4 PWM Modes

In PWM mode, PWM waveforms are output from the output pins. 0, 1, or toggle output can be selected as the output level in response to compare match of each TGR. Settings of TGR registers can output a PWM waveform in the range of 0 % to 100 % duty. Designating TGR compare match as the counter clearing source enables the period to be set in that register. All channels can be designated for PWM mode independently. Synchronous operation is also possible. There are two PWM modes, as described below.

- PWM mode 1

  PWM output is generated from the TIOCA and TIOCC pins by pairing TGRA with TGRB and TGRC with TGRD. The output specified by bits IOA3 to IOA0 and IOC3 to IOC0 in TIOR is output from the TIOCA and TIOCC pins at compare matches A and C, and the output specified by bits IOB3 to IOB0 and IOD3 to IOD0 in TIOR is output at compare matches B and D. The initial output value is the value set in TGRA or TGRC. If the set values of paired TGRs are identical, the output value does not change when a compare match occurs. In PWM mode 1, a maximum 4-phase PWM output is possible.
- PWM mode 2

  PWM output is generated using one TGR as the cycle register and the others as duty registers. The output specified in TIOR is performed by means of compare matches. Upon counter clearing by a synchronization register compare match, the output value of each pin is the initial value set in TIOR. If the set values of the cycle and duty registers are identical, the output value does not change when a compare match occurs. In PWM mode 2, a maximum 7-phase PWM output is possible by combined use with synchronous operation. The correspondence between PWM output pins and registers is shown in table 10.19.

Table 10.19 PWM Output Registers and Output Pins

		Output Pins	
Channel	**Registers**	**PWM Mode 1**	**PWM Mode 2**
0	TGRA_0	TIOCA0	TIOCA0
	TGRB_0		TIOCB0
	TGRC_0	TIOCC0	TIOCC0
	TGRD_0		TIOCD0
1	TGRA_1	TIOCA1	TIOCA1
	TGRB_1		TIOCB1
2	TGRA_2	TIOCA2	TIOCA2
	TGRB_2		TIOCB2

Note: In PWM mode 2, PWM output is not possible for the TGR register in which the period is set.

(1) Example of PWM Mode Setting Procedure

Figure 10.21 shows an example of the PWM mode setting procedure.

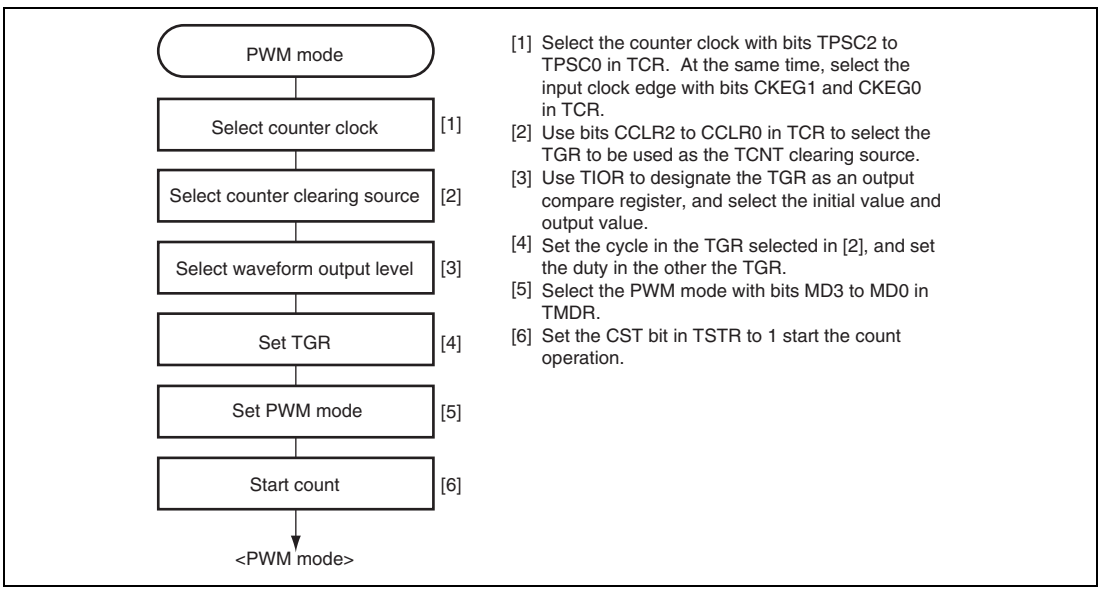


Figure 10.21 Example of PWM Mode Setting Procedure

(2) Examples of PWM Mode Operation

Figure 10.22 shows an example of PWM mode 1 operation.

In this example, TGRA compare match is set as the TCNT clearing source, 0 is set for the TGRA initial output value and output value, and 1 is set as the TGRB output value. In this case, the value set in TGRA is used as the period, and the values set in TGRB registers as the duty.

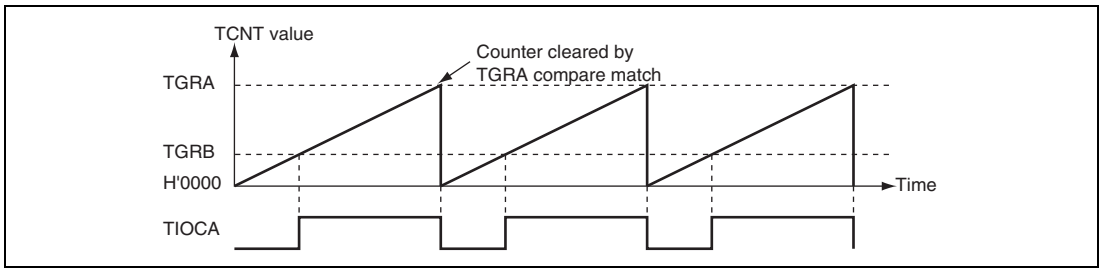


Figure 10.22 Example of PWM Mode Operation (1)

Figure 10.23 shows an example of PWM mode 2 operation. In this example, synchronous operation is designated for channels 0 and 1, TGRB_1 compare match is set as the TCNT clearing source, and 0 is set for the initial output value and 1 for the output value of the other TGR registers (TGRA_0 to TGRD_0, TGRA_1), to output a 5-phase PWM waveform. In this case, the value set in TGRB_1 is used as the cycle, and the values set in the other TGRs as the duty.

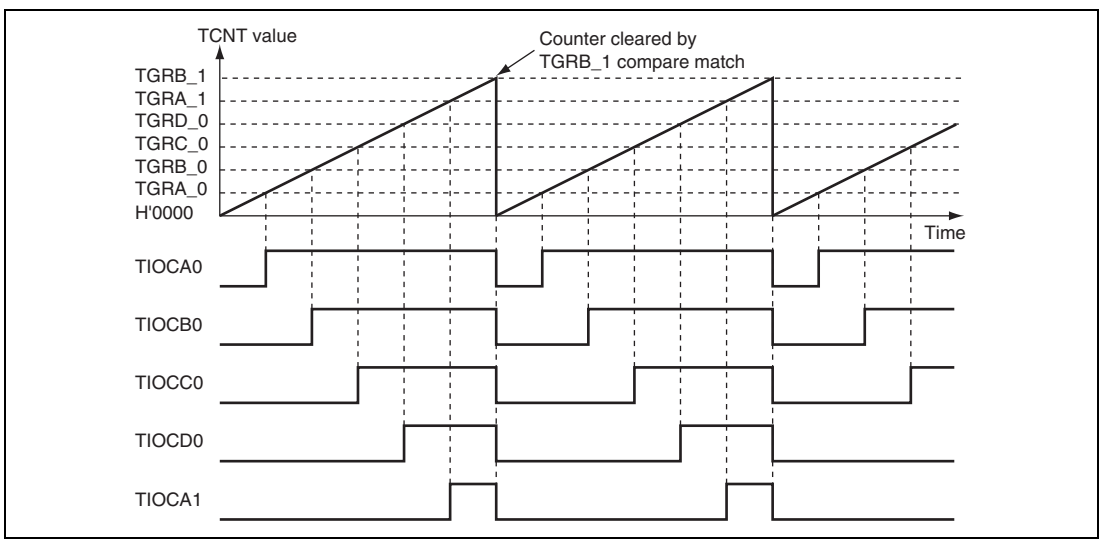


Figure 10.23 Example of PWM Mode Operation (2)

Figure 10.24 shows examples of PWM waveform output with 0% duty and 100% duty in PWM mode.

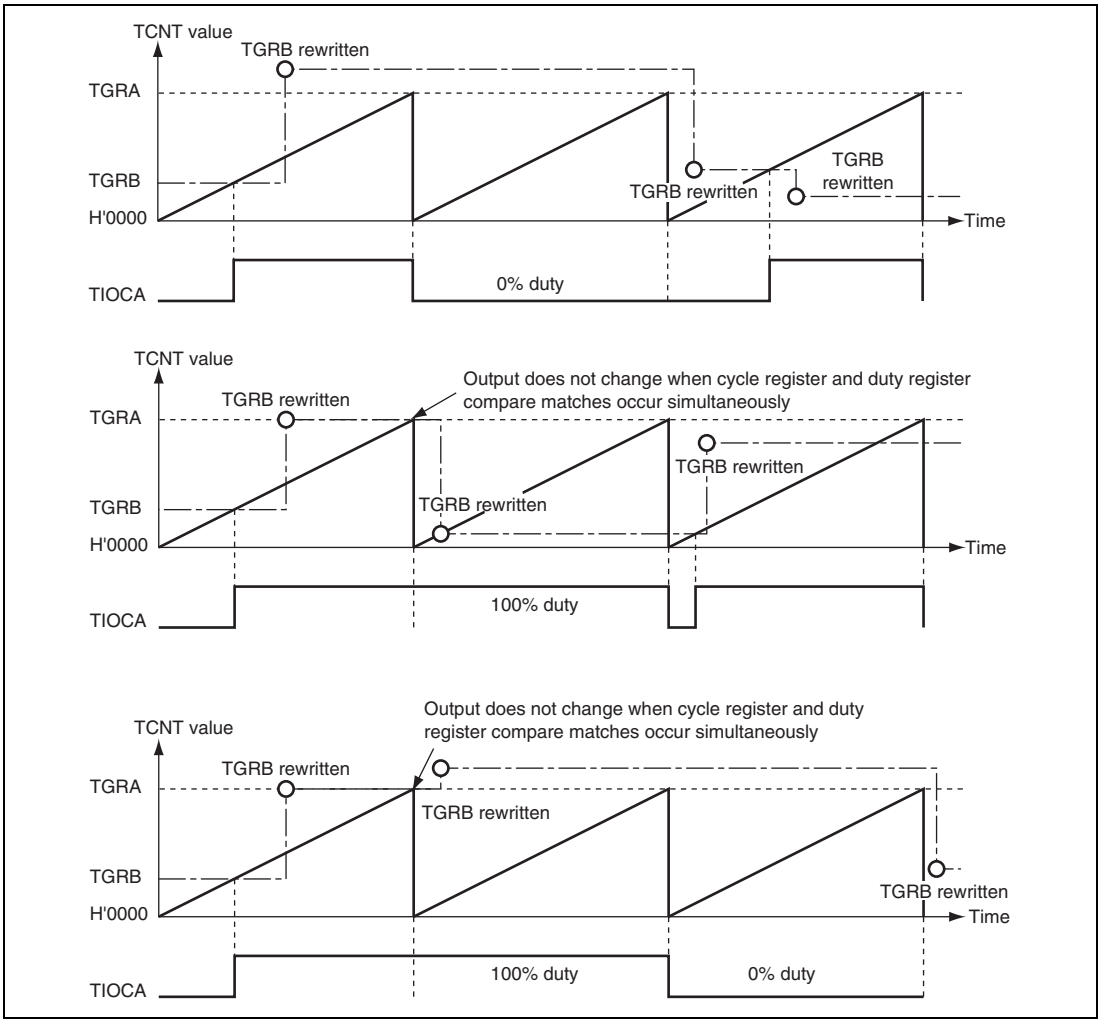


Figure 10.24 Example of PWM Mode Operation (3)

10.5.5 Phase Counting Mode

In phase counting mode, the phase difference between two external clock inputs is detected and TCNT is incremented/decremented accordingly. This mode can be set for channels 1 and 2. When phase counting mode is set, an external clock is selected as the counter input clock and TCNT operates as an up/down-counter regardless of the setting of bits TPSC2 to TPSC0 and bits CKEG1 and CKEG0 in TCR. However, the functions of bits CCLR1 and CCLR0 in TCR, and of TIOR, TIER, and TGR are valid, and input capture/compare match and interrupt functions can be used. This can be used for two-phase encoder pulse input. When overflow occurs while TCNT is counting up, the TCFV flag in TSR is set; when underflow occurs while TCNT is counting down, the TCFU flag is set. The TCFD bit in TSR is the count direction flag. Reading the TCFD flag provides an indication of whether TCNT is counting up or down. Table 10.20 shows the correspondence between external clock pins and channels.

Table 10.20 Phase Counting Mode Clock Input Pins

	External Clock Pins	
Channels	**A-Phase**	**B-Phase**
When channel 1 is set to phase counting mode	TCLKA	TCLKB
When channel 2 is set to phase counting mode	TCLKC	TCLKD

(1) Example of Phase Counting Mode Setting Procedure

Figure 10.25 shows an example of the phase counting mode setting procedure.

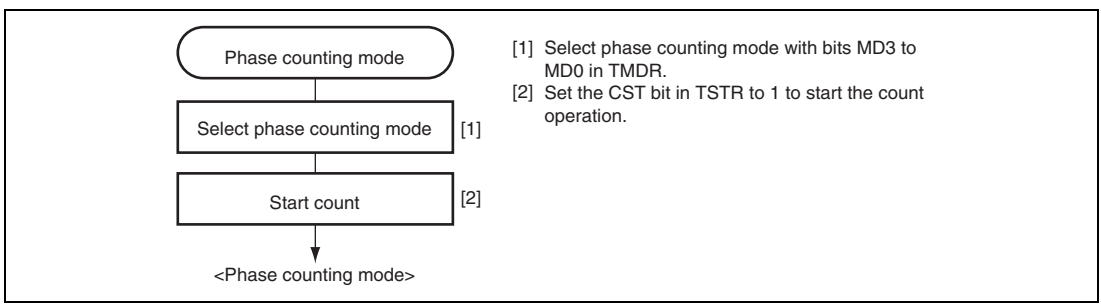


Figure 10.25 Example of Phase Counting Mode Setting Procedure

(2) Examples of Phase Counting Mode Operation

In phase counting mode, TCNT counts up or down according to the phase difference between two external clocks. There are four modes, according to the count conditions.

(a) Phase counting mode 1

Figure 10.26 shows an example of phase counting mode 1 operation, and table 10.21 summarizes the TCNT up/down-count conditions.

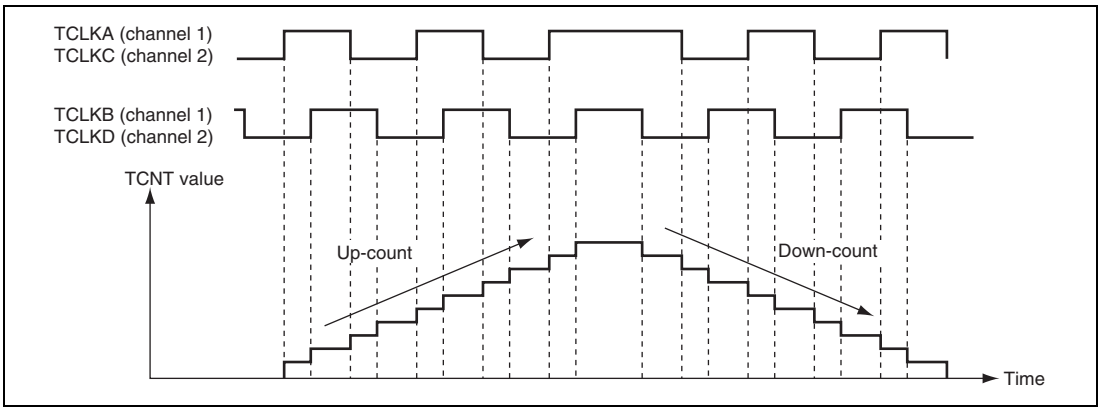


Figure 10.26 Example of Phase Counting Mode 1 Operation

Table 10.21 Up/Down-Count Conditions in Phase Counting Mode 1

TCLKA (Channel 1) TCLKC (Channel 2)	TCLKB (Channel 1) TCLKD (Channel 2)	Operation
High level	⌐↑	Up-count
Low level	⌐↓	
⌐↑	Low level	
⌐↓	High level	
High level	⌐↓	Down-count
Low level	⌐↑	
⌐↑	High level	
⌐↓	Low level	

[Legend]
⌐↑: Rising edge
⌐↓: Falling edge

(b) Phase counting mode 2

Figure 10.27 shows an example of phase counting mode 2 operation, and table 10.22 summarizes the TCNT up/down-count conditions.

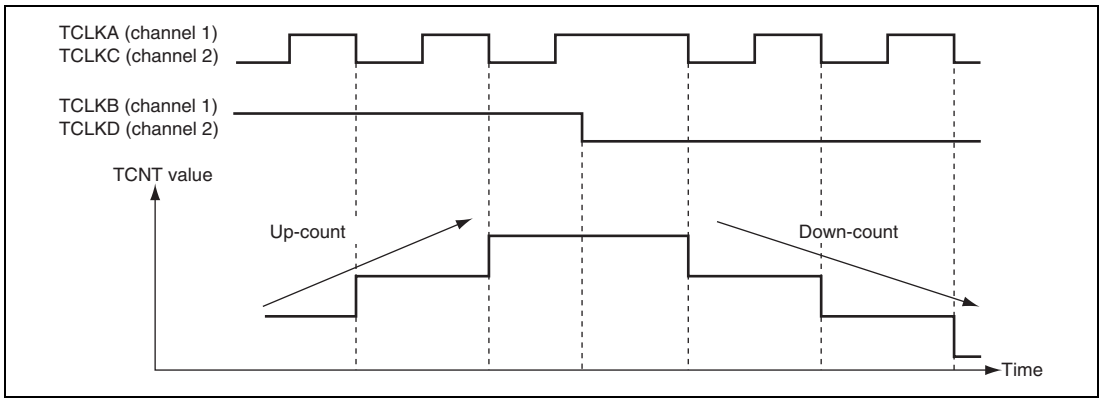


Figure 10.27 Example of Phase Counting Mode 2 Operation

Table 10.22 Up/Down-Count Conditions in Phase Counting Mode 2

TCLKA (Channel 1) TCLKC (Channel 2)	TCLKB (Channel 1) TCLKD (Channel 2)	Operation
High level	↑	Don't care
Low level	↓	Don't care
↑	Low level	Don't care
↓	High level	Up-count
High level	↓	Don't care
Low level	↑	Don't care
↑	High level	Don't care
↓	Low level	Down-count

[Legend]
↑: Rising edge
↓: Falling edge

(c) Phase counting mode 3

Figure 10.28 shows an example of phase counting mode 3 operation, and table 10.23 summarizes the TCNT up/down-count conditions.

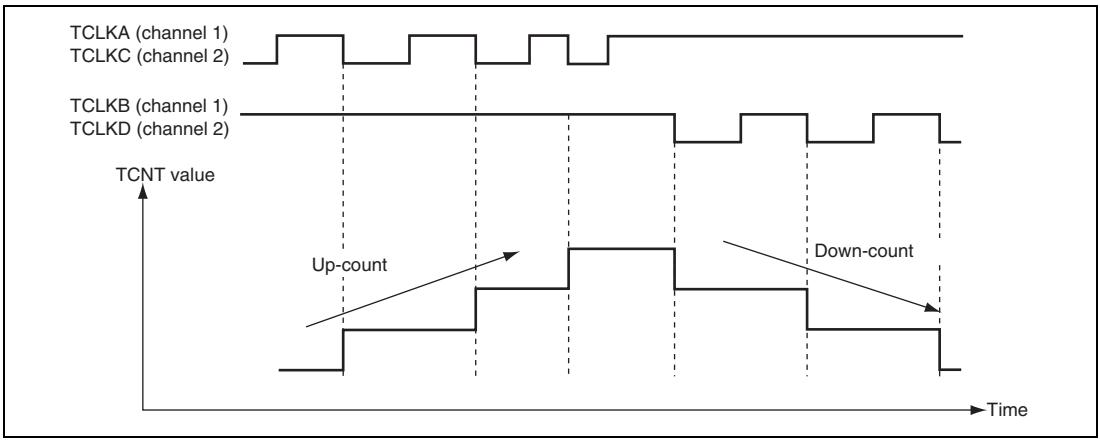


Figure 10.28 Example of Phase Counting Mode 3 Operation

Table 10.23 Up/Down-Count Conditions in Phase Counting Mode 3

TCLKA (Channel 1) TCLKC (Channel 2)	TCLKB (Channel 1) TCLKD (Channel 2)	Operation
High level	⌐↑	Don't care
Low level	⌐↓	Don't care
⌐↑	Low level	Don't care
⌐↓	High level	Up-count
High level	⌐↓	Down-count
Low level	⌐↑	Don't care
⌐↑	High level	Don't care
⌐↓	Low level	Don't care

[Legend]
⌐↑: Rising edge
⌐↓: Falling edge

(d) Phase counting mode 4

Figure 10.29 shows an example of phase counting mode 4 operation, and table 10.24 summarizes the TCNT up/down-count conditions.

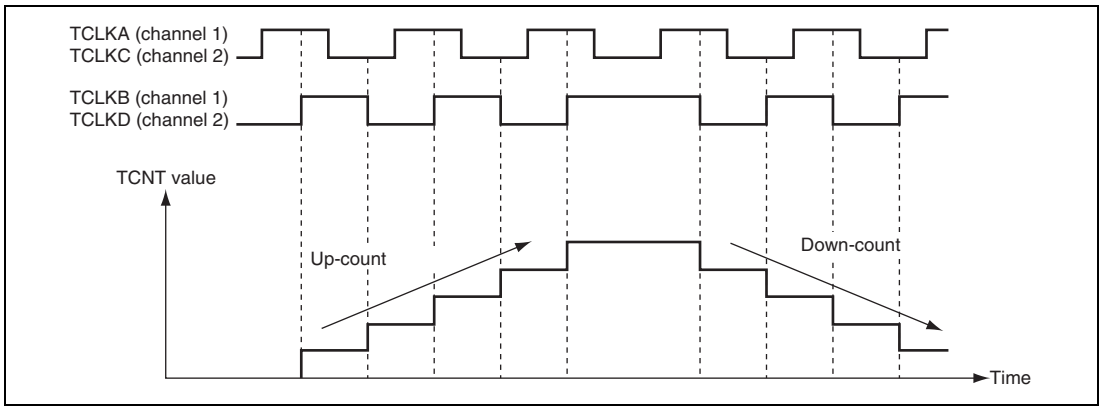


Figure 10.29 Example of Phase Counting Mode 4 Operation

Table 10.24 Up/Down-Count Conditions in Phase Counting Mode 4

TCLKA (Channel 1) / TCLKC (Channel 2)	TCLKB (Channel 1) / TCLKD (Channel 2)	Operation
High level	↑	Up-count
Low level	↓	
↑	Low level	Don't care
↓	High level	
High level	↓	Down-count
Low level	↑	
↑	High level	Don't care
↓	Low level	

[Legend]
↑: Rising edge
↓: Falling edge

10.6 Interrupts

10.6.1 Interrupt Source and Priority

There are three kinds of TPU interrupt source: TGR input capture/compare match, TCNT overflow, and TCNT underflow. Each interrupt source has its own status flag and enable/disabled bit, allowing generation of interrupt request signals to be enabled or disabled individually. When an interrupt request is generated, the corresponding status flag in TSR is set to 1. If the corresponding enable/disable bit in TIER is set to 1 at this time, an interrupt is requested. The interrupt request is cleared by clearing the status flag to 0. Relative channel priorities can be changed by the interrupt controller, but the priority order within a channel is fixed. For details, see section 5, Interrupt Controller. Table 10.25 lists the TPU interrupt sources.

Table 10.25 TPU Interrupts

Channel	Name	Interrupt Source	Interrupt Flag	Priority*
0	TGI0A	TGRA_0 input capture/compare match	TGFA	High
	TGI0B	TGRB_0 input capture/compare match	TGFB	
	TGI0C	TGRC_0 input capture/compare match	TGFC	
	TGI0D	TGRD_0 input capture/compare match	TGFD	
	TCI0V	TCNT_0 overflow	TCFV	
1	TGI1A	TGRA_1 input capture/compare match	TGFA	
	TGI1B	TGRB_1 input capture/compare match	TGFB	
	TCI1V	TCNT_1 overflow	TCFV	
	TCI1U	TCNT_1 underflow	TCFU	
2	TGI2A	TGRA_2 input capture/compare match	TGFA	
	TGI2B	TGRB_2 input capture/compare match	TGFB	
	TCI2V	TCNT_2 overflow	TCFV	
	TCI2U	TCNT_2 underflow	TCFU	Low

Note: * This table shows the initial state immediately after a reset. The relative channel priorities can be changed by the interrupt controller.

(1) Input Capture/Compare Match Interrupt

An interrupt is requested if the TGIE bit in TIER is set to 1 when the TGF flag in TSR is set to 1 by the occurrence of a TGR input capture/compare match on a particular channel. The interrupt request is cleared by clearing the TGF flag to 0. The TPU has 16 input capture/compare match interrupts, four each for channel 0, and two each for channels 1 and 2.

(2) Overflow Interrupt

An interrupt is requested if the TCIEV bit in TIER is set to 1 when the TCFV flag in TSR is set to 1 by the occurrence of TCNT overflow on a channel. The interrupt request is cleared by clearing the TCFV flag to 0. The TPU has three overflow interrupts, one for each channel.

(3) Underflow Interrupt

An interrupt is requested if the TCIEU bit in TIER is set to 1 when the TCFU flag in TSR is set to 1 by the occurrence of TCNT underflow on a channel. The interrupt request is cleared by clearing the TCFU flag to 0. The TPU has two underflow interrupts, one each for channels 1 and 2.

10.6.2 A/D Converter Activation

The A/D converter can be activated by the TGRA input capture/compare match for a channel. If the TTGE bit in TIER is set to 1 when the TGFA flag in TSR is set to 1 by the occurrence of a TGRA input capture/compare match on a particular channel, a request to start A/D conversion is sent to the A/D converter. If the TPU conversion start trigger has been selected on the A/D converter side at this time, A/D conversion is started. In the TPU, a total of three TGRA input capture/compare match interrupts can be used as A/D converter conversion start sources, one for each channel.

10.7 Operation Timing

10.7.1 Input/Output Timing

(1) TCNT Count Timing

Figure 10.30 shows TCNT count timing in internal clock operation, and figure 10.31 shows TCNT count timing in external clock operation.

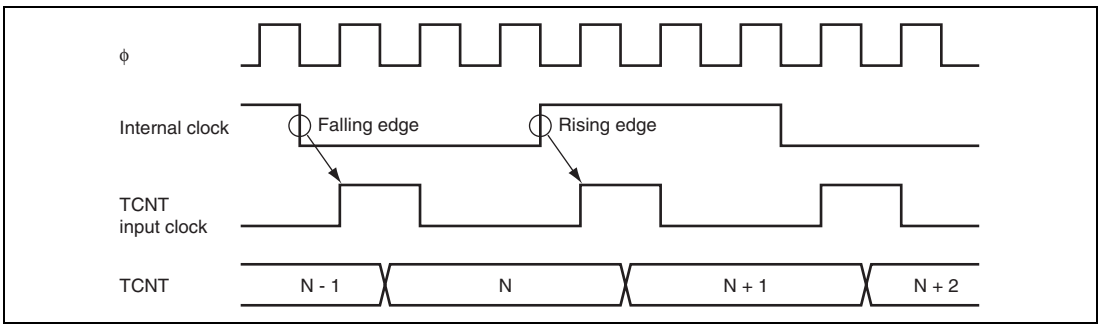


Figure 10.30 Count Timing in Internal Clock Operation

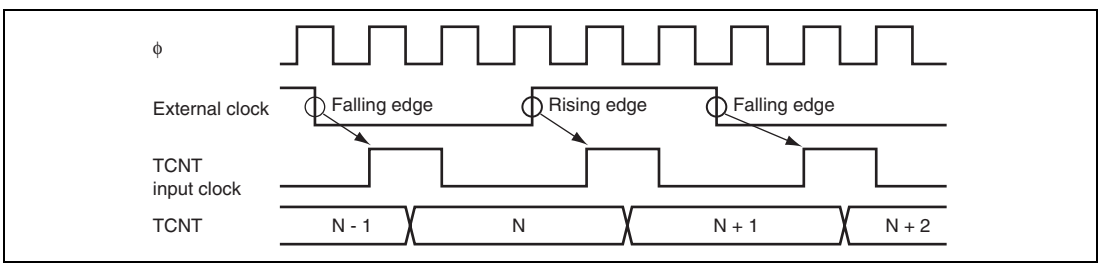


Figure 10.31 Count Timing in External Clock Operation

(2) Output Compare Output Timing

A compare match signal is generated in the final state in which TCNT and TGR match (the point at which the count value matched by TCNT is updated). When a compare match signal is generated, the output value set in TIOR is output at the output compare output pin (TIOC pin). After a match between TCNT and TGR, the compare match signal is not generated until the TCNT input clock is generated. Figure 10.32 shows output compare output timing.

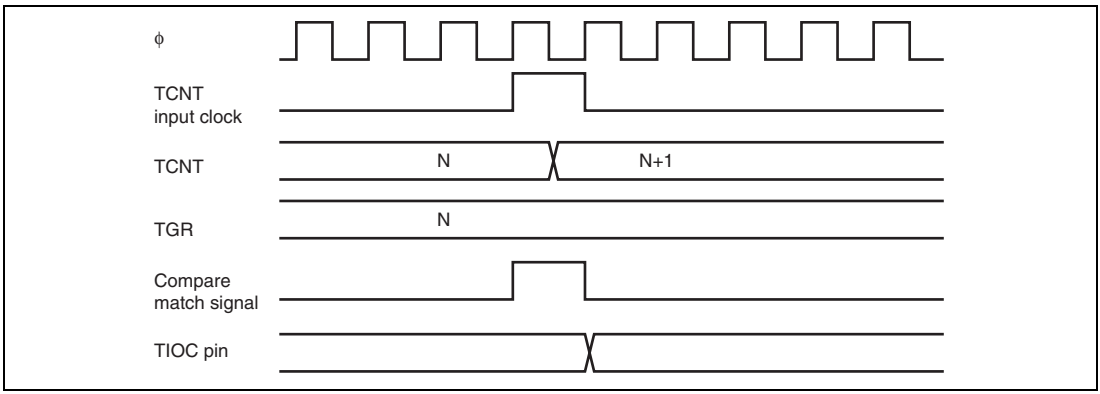


Figure 10.32 Output Compare Output Timing

Input Capture Signal Timing: Figure 10.33 shows input capture signal timing.

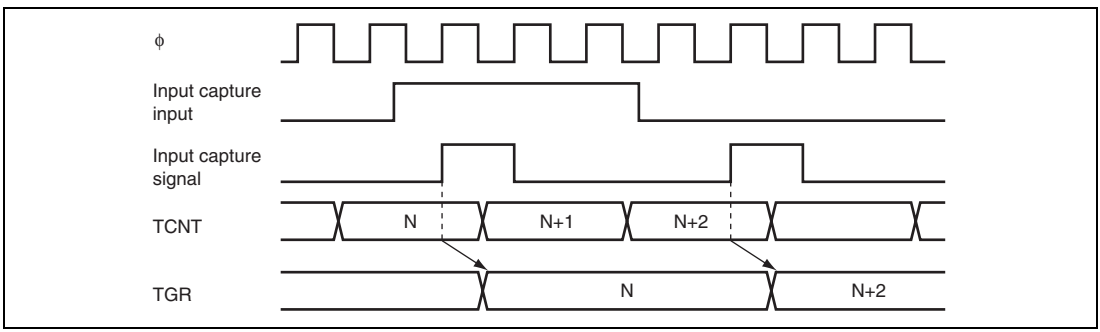


Figure 10.33 Input Capture Input Signal Timing

(3) Timing for Counter Clearing by Compare Match/Input Capture

Figure 10.34 shows the timing when counter clearing by compare match occurrence is specified, and figure 10.35 shows the timing when counter clearing by input capture occurrence is specified.

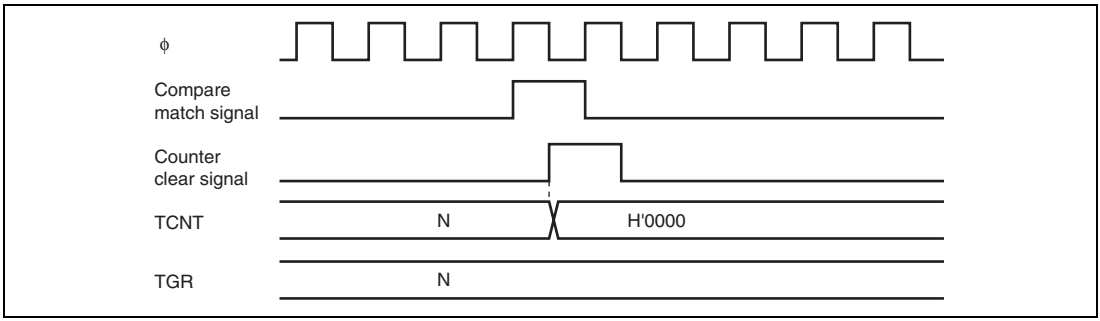


Figure 10.34 Counter Clear Timing (Compare Match)

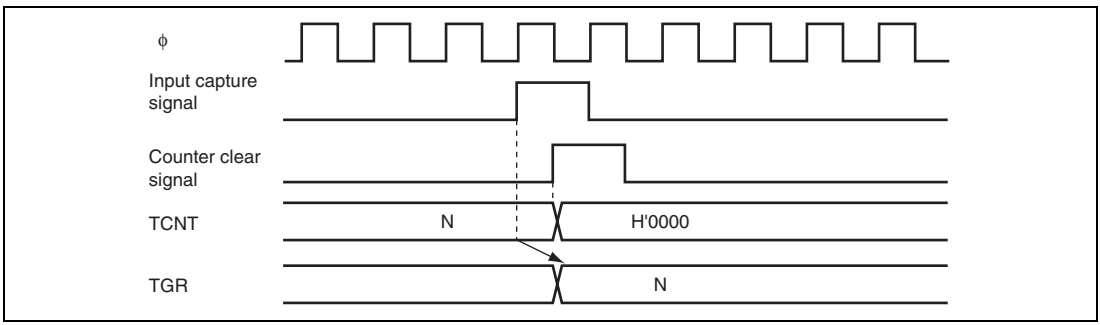


Figure 10.35 Counter Clear Timing (Input Capture)

(4) Buffer Operation Timing

Figures 10.36 and 10.37 show the timing in buffer operation.

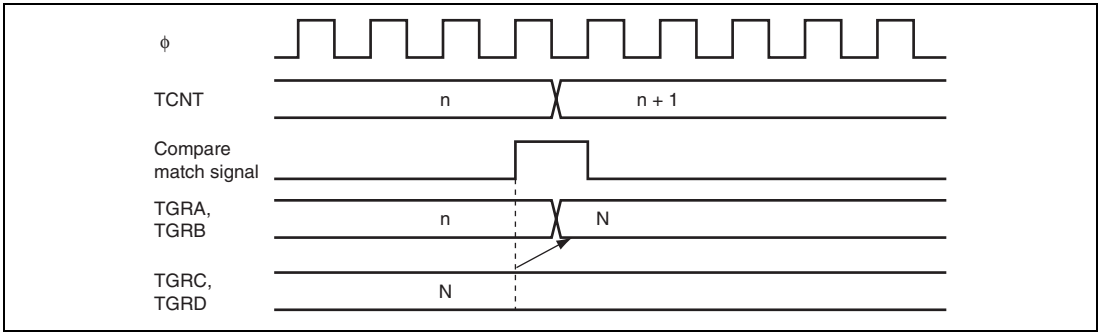


Figure 10.36 Buffer Operation Timing (Compare Match)

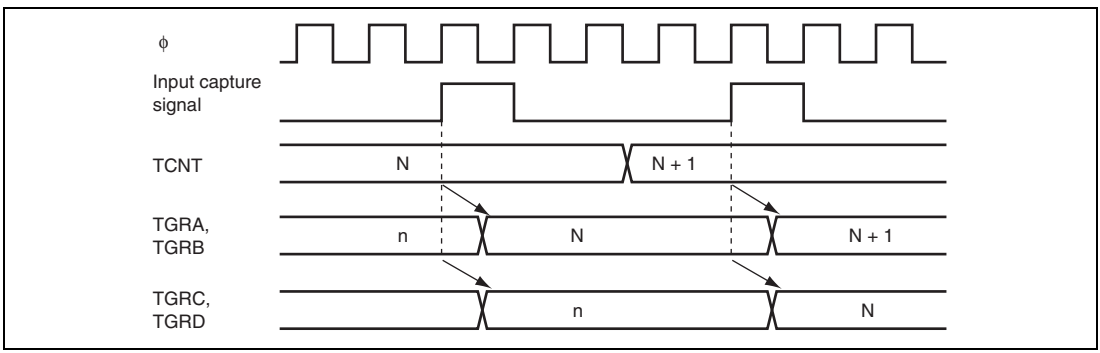


Figure 10.37 Buffer Operation Timing (Input Capture)

10.7.2 Interrupt Signal Timing

(1) TGF Flag Setting Timing in Case of Compare Match

Figure 10.38 shows the timing for setting of the TGF flag in TSR by compare match occurrence, and TGI interrupt request signal timing.

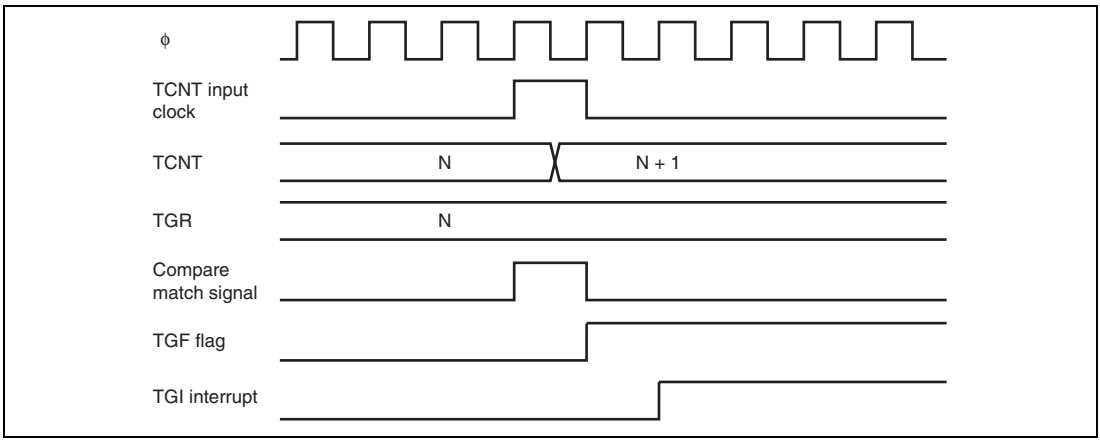


Figure 10.38 TGI Interrupt Timing (Compare Match)

(2) TGF Flag Setting Timing in Case of Input Capture

Figure 10.39 shows the timing for setting of the TGF flag in TSR by input capture occurrence, and TGI interrupt request signal timing.

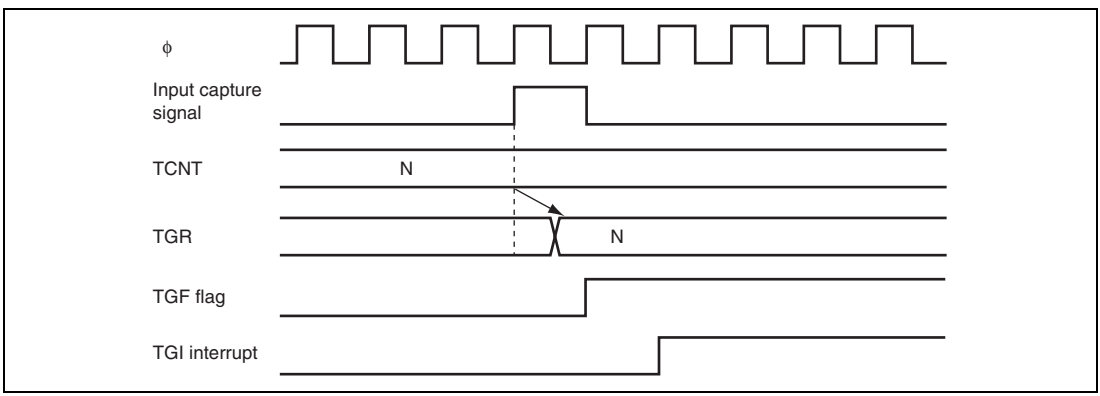


Figure 10.39 TGI Interrupt Timing (Input Capture)

(3) TCFV Flag/TCFU Flag Setting Timing

Figure 10.40 shows the timing for setting of the TCFV flag in TSR by overflow occurrence, and TCIV interrupt request signal timing. Figure 10.41 shows the timing for setting of the TCFU flag in TSR by underflow occurrence, and TCIU interrupt request signal timing.

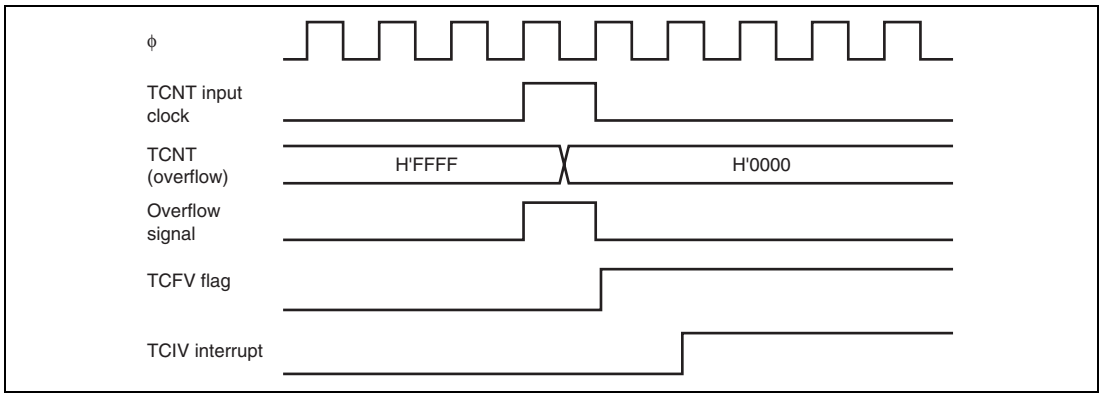


Figure 10.40 TCIV Interrupt Setting Timing

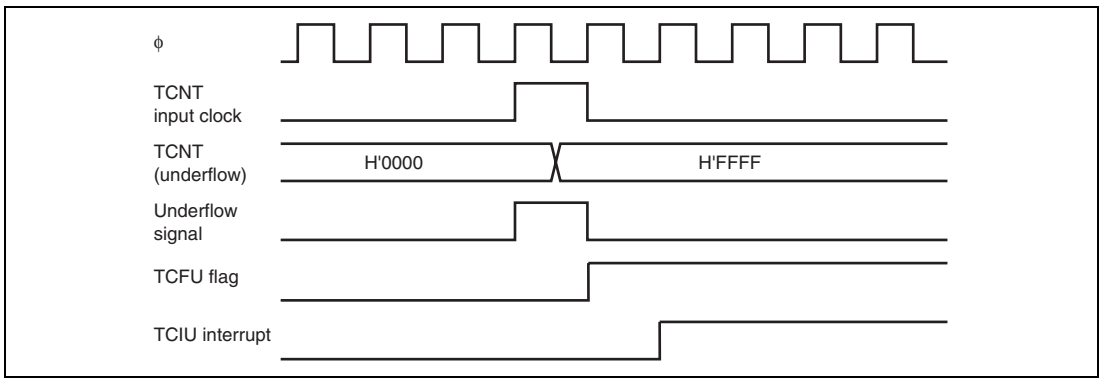


Figure 10.41 TCIU Interrupt Setting Timing

(4) Status Flag Clearing Timing

After a status flag is read as 1 by the CPU, it is cleared by writing 0 to it. Figure 10.42 shows the timing for status flag clearing by the CPU.

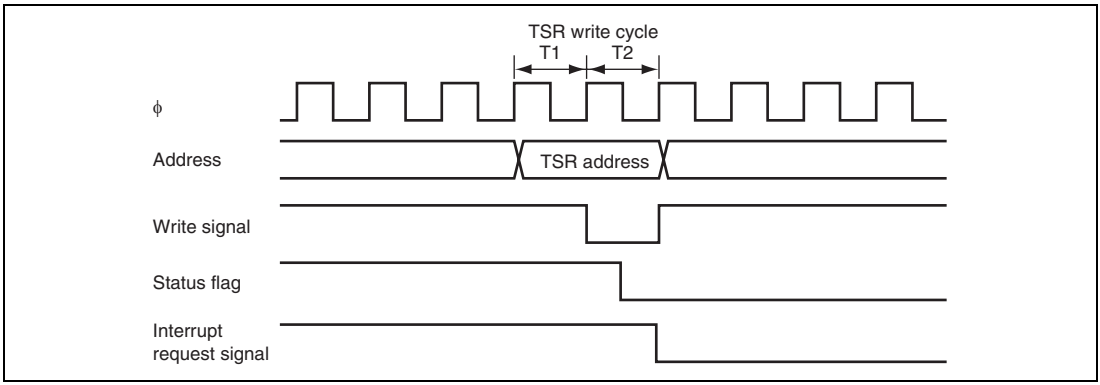


Figure 10.42 Timing for Status Flag Clearing by CPU

10.8 Usage Notes

10.8.1 Input Clock Restrictions

The input clock pulse width must be at least 1.5 states in the case of single-edge detection, and at least 2.5 states in the case of both-edge detection. The TPU will not operate properly with a narrower pulse width. In phase counting mode, the phase difference and overlap between the two input clocks must be at least 1.5 states, and the pulse width must be at least 2.5 states. Figure 10.43 shows the input clock conditions in phase counting mode.

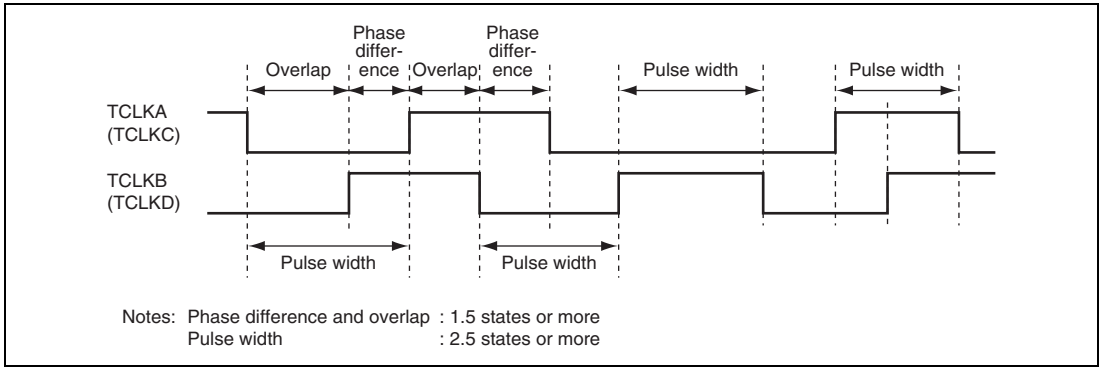


Figure 10.43 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

10.8.2 Caution on Period Setting

When counter clearing by compare match is set, TCNT is cleared in the final state in which it matches the TGR value (the point at which the count value matched by TCNT is updated). Consequently, the actual counter frequency is given by the following formula:

$$f = \frac{\phi}{(N + 1)}$$

Where f: Counter frequency
ϕ: Operating frequency
N: TGR set value

10.8.3 Conflict between TCNT Write and Clear Operations

If the counter clear signal is generated in the T_2 state of a TCNT write cycle, TCNT clearing takes precedence and the TCNT write is not performed. Figure 10.44 shows the timing in this case.

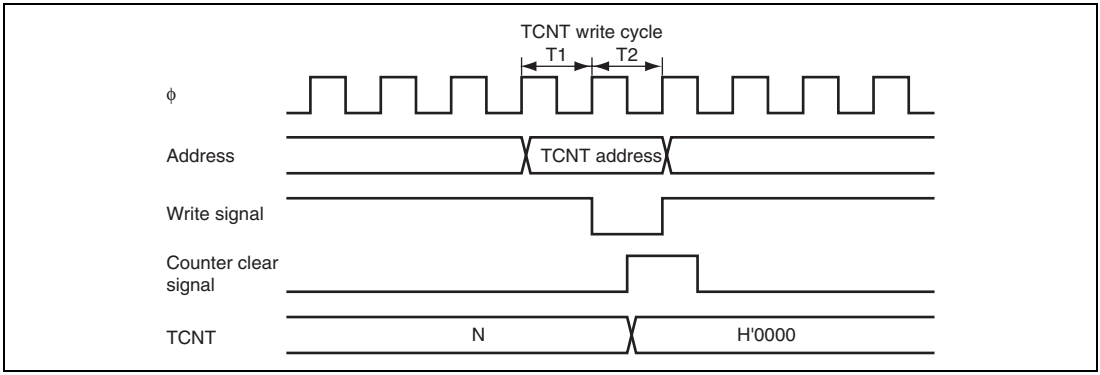


Figure 10.44 Conflict between TCNT Write and Clear Operations

10.8.4 Conflict between TCNT Write and Increment Operations

If incrementing occurs in the T_2 state of a TCNT write cycle, the TCNT write takes precedence and TCNT is not incremented. Figure 10.45 shows the timing in this case.

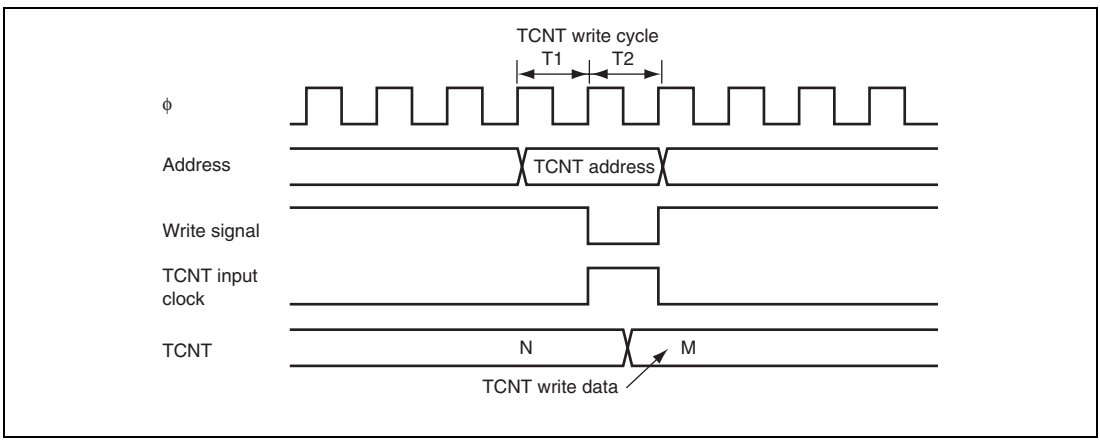


Figure 10.45 Conflict between TCNT Write and Increment Operations

10.8.5 Conflict between TGR Write and Compare Match

If a compare match occurs in the T_2 state of a TGR write cycle, the TGR write takes precedence and the compare match signal is inhibited. A compare match does not occur even if the same value as before is written. Figure 10.46 shows the timing in this case.

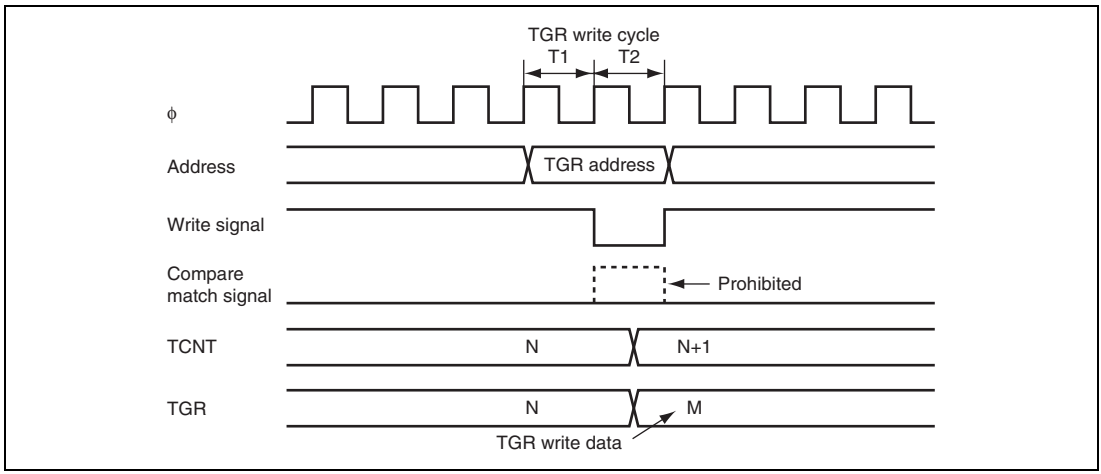


Figure 10.46 Conflict between TGR Write and Compare Match

10.8.6 Conflict between Buffer Register Write and Compare Match

If a compare match occurs in the T_2 state of a TGR write cycle, the data transferred to TGR by the buffer operation will be the data prior to the write. Figure 10.47 shows the timing in this case.

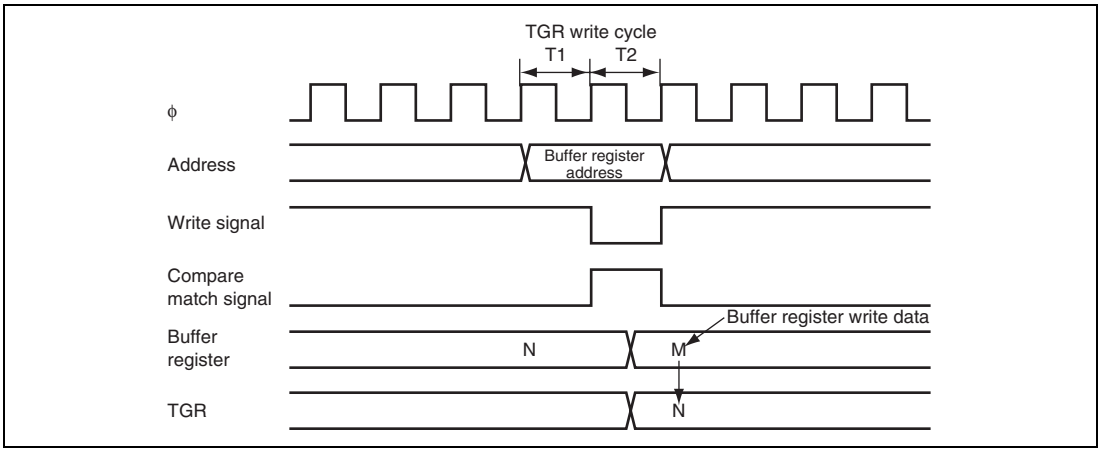


Figure 10.47 Conflict between Buffer Register Write and Compare Match

10.8.7 Conflict between TGR Read and Input Capture

If the input capture signal is generated in the T1 state of a TGR read cycle, the data that is read will be the data after input capture transfer. Figure 10.48 shows the timing in this case.

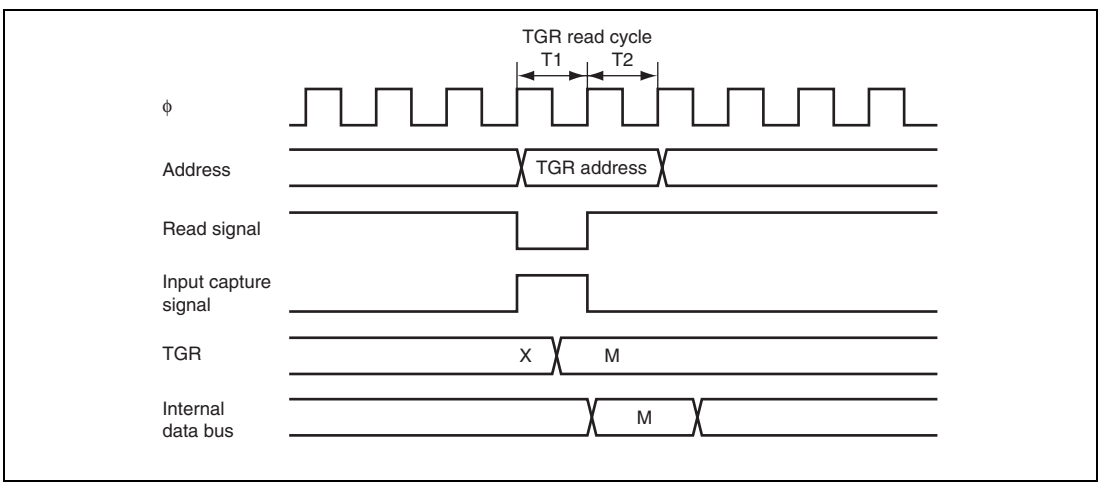


Figure 10.48 Conflict between TGR Read and Input Capture

10.8.8 Conflict between TGR Write and Input Capture

If the input capture signal is generated in the T_2 state of a TGR write cycle, the input capture operation takes precedence and the write to TGR is not performed. Figure 10.49 shows the timing in this case.

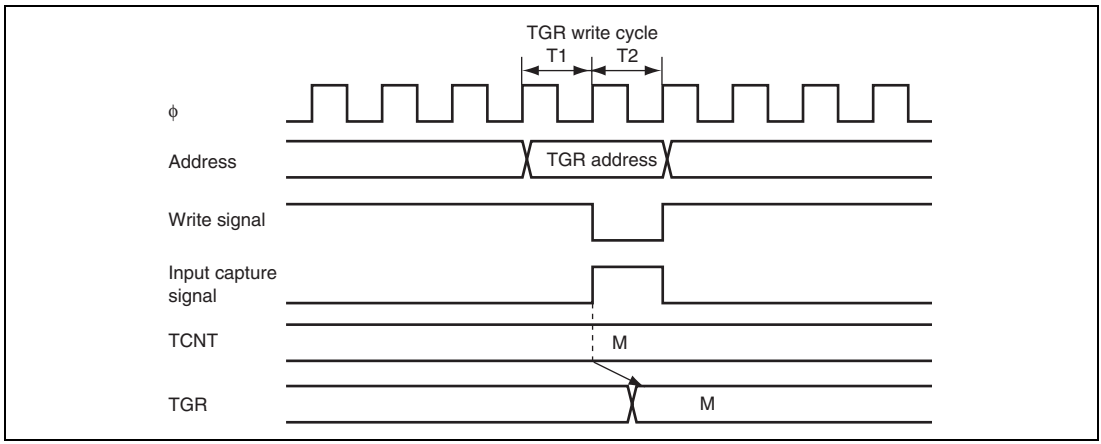


Figure 10.49 Conflict between TGR Write and Input Capture

10.8.9 Conflict between Buffer Register Write and Input Capture

If the input capture signal is generated in the T_2 state of a buffer register write cycle, the buffer operation takes precedence and the write to the buffer register is not performed. Figure 10.50 shows the timing in this case.

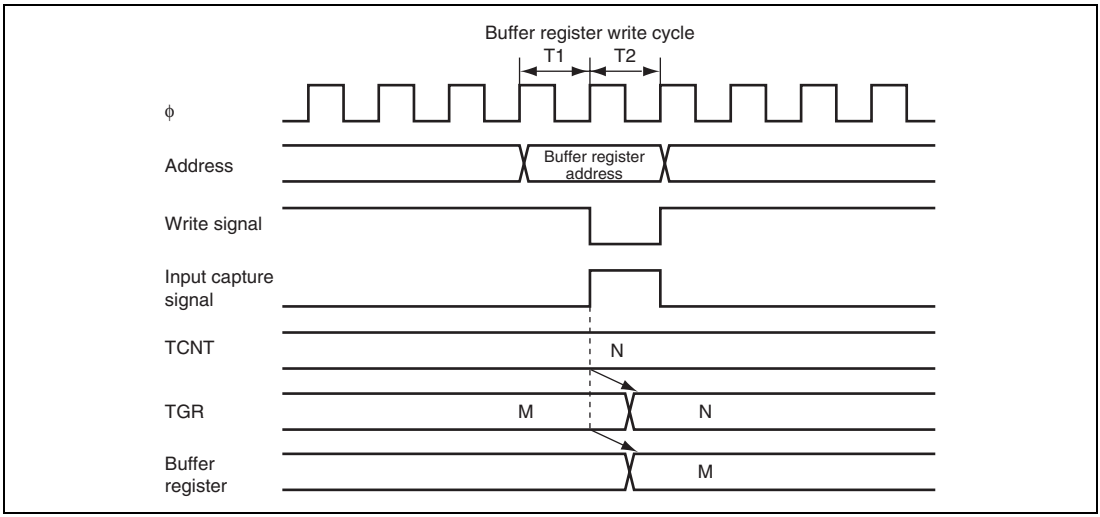


Figure 10.50 Conflict between Buffer Register Write and Input Capture

10.8.10 Conflict between Overflow/Underflow and Counter Clearing

If overflow/underflow and counter clearing occur simultaneously, the TCFV/TCFU flag in TSR is not set and TCNT clearing takes precedence. Figure 10.51 shows the operation timing when a TGR compare match is specified as the clearing source, and H'FFFF is set in TGR.

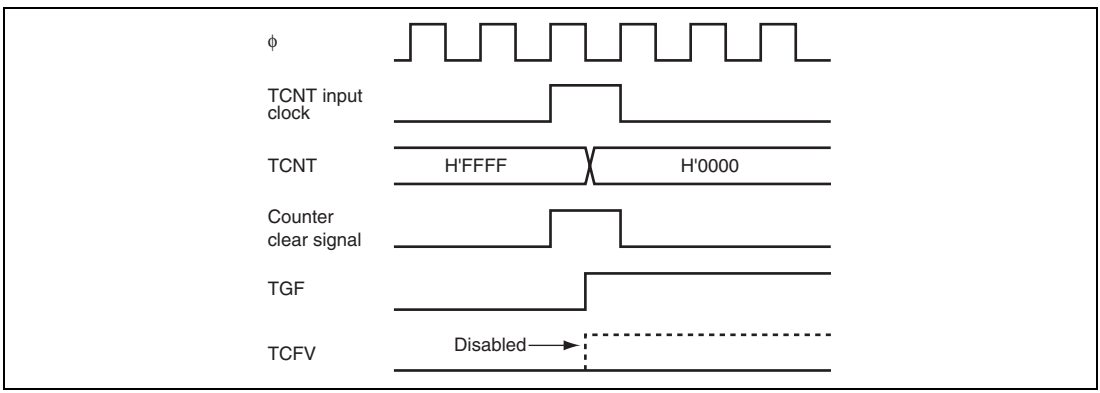


Figure 10.51 Conflict between Overflow and Counter Clearing

10.8.11 Conflict between TCNT Write and Overflow/Underflow

If there is an up-count or down-count in the T_2 state of a TCNT write cycle, and overflow/underflow occurs, the TCNT write takes precedence and the TCFV/TCFU flag in TSR is not set. Figure 10.52 shows the operation timing when there is conflict between TCNT write and overflow.

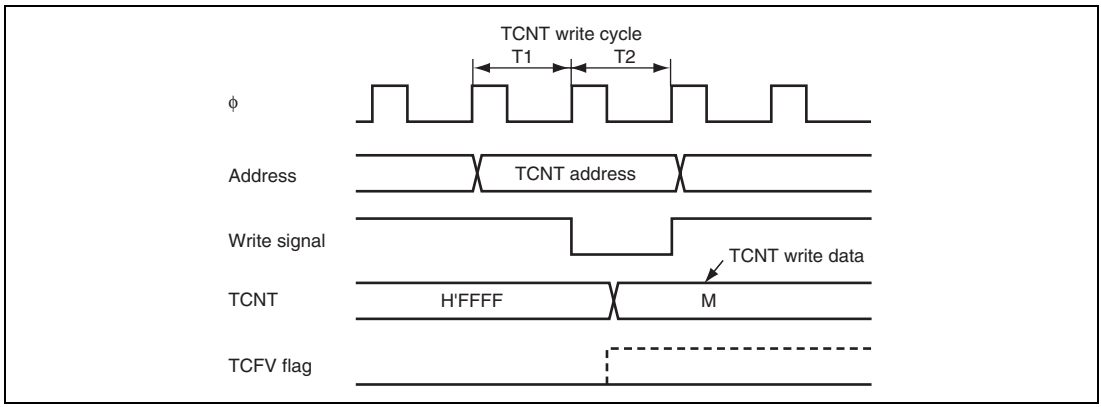


Figure 10.52 Conflict between TCNT Write and Overflow

10.8.12 Multiplexing of I/O Pins

In this LSI, the TCLKA input pin is multiplexed with the TIOCC0 I/O pin, the TCLKB input pin with the TIOCD0 I/O pin, the TCLKC input pin with the TIOCB1 I/O pin, and the TCLKD input pin with the TIOCB2 I/O pin. When an external clock is input, compare match output should not be performed from a multiplexed pin.

10.8.13 Module Stop Mode Setting

TPU operation can be enabled or disabled by the module stop control register. In the initial state, TPU operation is disabled. Access to TPU registers is enabled when module stop mode is cancelled. For details, see section 26, Power-Down Modes.

Section 11 16-Bit Cycle Measurement Timer (TCM)

This LSI has four channels on-chip 16-bit cycle measurement timers (TCM). Each TCM has a 16-bit counter that provides the basis for measuring the periods of input waveforms.

11.1 Features

- Capable of measuring the periods of input waveforms
- Sensed edge is selectable
- 16-bit compare match
- 16-bit resolution
- Selectable counter clock
  - Any of seven internal clocks or an external clock
- Five interrupt sources
  - Counter overflow
  - Cycle upper limit overflow
  - Cycle lower limit underflow
  - Compare match
  - Triggering of input capture

Figure 11.1 is a block diagram of the TCM.

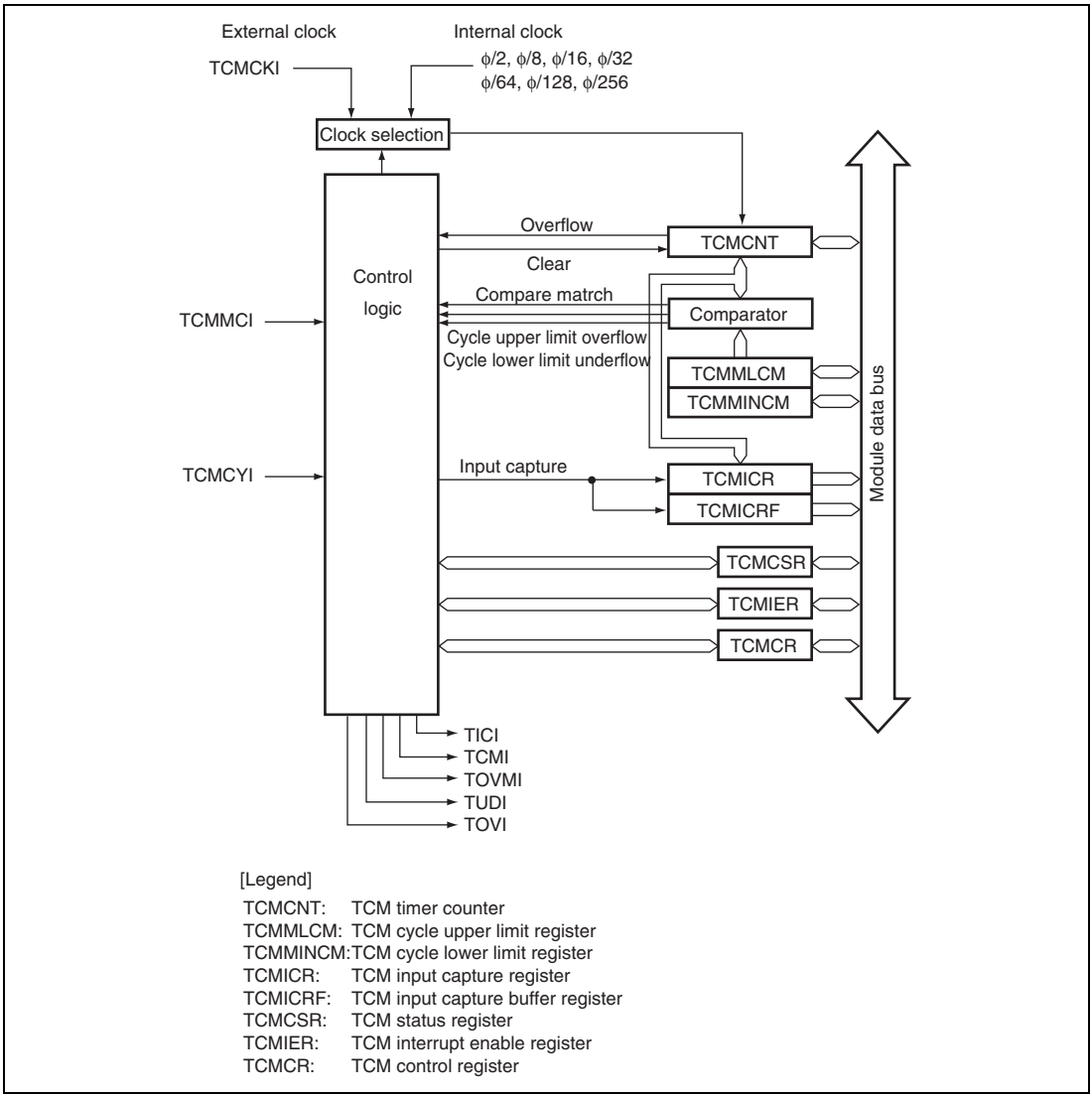


Figure 11.1 Block Diagram of the TCM

11.2 Input/Output Pins

Table 11.1 lists the input and output pins for the TCMs.

Table 11.1 Pin Configuration

Channel	Pin Name	I/O	Function
0	TCMCKI0	Input	External counter clock input
	(TCMMCI0)		Cycle measurement control input
	TCMCYI0	Input	External event input
1	TCMCKI1	Input	External counter clock input
	(TCMMCI1)		Cycle measurement control input
	TCMCYI1	Input	External event input
2	TCMCKI2	Input	External counter clock input
	(TCMMCI2)		Cycle measurement control input
	TCMCYI2	Input	External event input
3	TCMCKI3	Input	External counter clock input
	(TCMMCI3)		Cycle measurement control input
	TCMCYI3	Input	External event input

11.3 Register Descriptions

The TCMs have the following registers.

Table 11.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Data Bus Width
Channel 0	TCM timer counter_0	TCMCNT_0	R/W	H'0000	H'FBC0	16
	TCM cycle upper limit register_0	TCMMLCM_0	R/W	H'FFFF	H'FBC2	16
	TCM cycle lower limit register_0	TCMMINCM_0	R/W	H'0000	H'FBCC	16
	TCM input capture register_0	TCMICR_0	R	H'0000	H'FBC4	16
	TCM input capture buffer register_0	TCMICRF_0	R	H'0000	H'FBC6	16
	TCM status register_0	TCMCSR_0	R/W	H'00	H'FBC8	8
	TCM control register_0	TCMCR_0	R/W	H'00	H'FBC9	8
	TCM interrupt enable register_0	TCMIER_0	R/W	H'00	H'FBCA	8

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Data Bus Width
Channel 1	TCM timer counter_1	TCMCNT_1	R/W	H'0000	H'FBD0	16
	TCM cycle upper limit register_1	TCMMLCM_1	R/W	H'FFFF	H'FBD2	16
	TCM cycle lower limit register_1	TCMMINCM_1	R/W	H'0000	H'FBDC	16
	TCM input capture register_1	TCMICR_1	R	H'0000	H'FBD4	16
	TCM input capture buffer register_1	TCMICRF_1	R	H'0000	H'FBD6	16
	TCM status register_1	TCMCSR_1	R/W	H'00	H'FBD8	8
	TCM control register_1	TCMCR_1	R/W	H'00	H'FBD9	8
	TCM interrupt enable register_1	TCMIER_1	R/W	H'00	H'FBDA	8
Channel 2	TCM timer counter_2	TCMCNT_2	R/W	H'0000	H'FBE0	16
	TCM cycle upper limit register_2	TCMMLCM_2	R/W	H'FFFF	H'FBE2	16
	TCM cycle lower limit register_2	TCMMINCM_2	R/W	H'0000	H'FBEC	16
	TCM input capture register_2	TCMICR_2	R	H'0000	H'FBE4	16
	TCM input capture buffer register_2	TCMICRF_2	R	H'0000	H'FBE6	16
	TCM status register_2	TCMCSR_2	R/W	H'00	H'FBE8	8
	TCM control register_2	TCMCR_2	R/W	H'00	H'FBE9	8
	TCM interrupt enable register_2	TCMIER_2	R/W	H'00	H'FBEA	8
Channel 3	TCM timer counter_3	TCMCNT_3	R/W	H'0000	H'FBF0	16
	TCM cycle upper limit register_3	TCMMLCM_3	R/W	H'FFFF	H'FBF2	16
	TCM cycle lower limit register_3	TCMMINCM_3	R/W	H'0000	H'FBFC	16
	TCM input capture register_3	TCMICR_3	R	H'0000	H'FBF4	16
	TCM input capture buffer register_3	TCMICRF_3	R	H'0000	H'FBF6	16
	TCM status register_3	TCMCSR_3	R/W	H'00	H'FBF8	8
	TCM control register_3	TCMCR_3	R/W	H'00	H'FBF9	8
	TCM interrupt enable register_3	TCMIER_3	R/W	H'00	H'FBFA	8

11.3.1 TCM Timer Counter (TCMCNT)

TCMCNT is a 16-bit readable/writable up-counter. The input clock is selected by the bits CKS2 to CKS0 in TCMCR. When CKS2 to CKS0 are set to B'111, the external clock is selected. In this case, the rising or falling edge is selected by CKSEG in TCMCR.

When TCMCNT overflows (counting changes the value from H'FFFF to H'0000), OVF in TCMCSR is set to 1. When the CST bit in TCMCR is cleared in timer mode, TCMCR is initialized to H'0000. In cycle measurement mode, TCMCNT is cleared by detection of the first edge (the edge selected with the IEDG bit in TCMCR) of the measurement period (one period of the input waveform forms one measurement period).

In timer mode, TCMCNT is always writable. TCMCNT cannot be modified in cycle measurement mode. TCMCNT should always be accessed in 16-bit units and cannot be accessed in 8-bit units. TCMCNT is initialized to H'0000.

11.3.2 TCM Cycle Upper Limit Register (TCMMLCM)

TCMMLCM is a 16-bit readable/writable register. TCMMLCM is available as a compare match register when the TCMMDS bit in TCMCR is cleared (operation is in timer mode). TCMMLCM is available as a cycle upper limit register when the TCMMDS bit in TCMCR is set to 1 (operation is in cycle measurement mode).

In timer mode, the value in TCMMLCM is constantly compared with that in TCMCNT, when the values match, CMF in TCMCSR is set to 1. However, comparison is disabled in the second half of a cycle of writing to TCMMLCM.

In cycle measurement mode, a value that sets an upper limit on the measurement period can be set in TCMMLCM. When the second edge (first edge of the following cycle) of the measurement period is detected, the value in TCMCNT is transferred to TCMICR. At this time, the values in TCMICR and TCMMLCM are compared. The MAXOVF flag in TCMCSR is set to 1 if the value in TCMICR is greater than that in TCMMLCM. TCMMLCM should always be accessed in 16-bit units and cannot be accessed in 8-bit units. TCMMLCM is initialized to H'FFFF.

11.3.3 TCM Cycle Lower Limit Register (TCMMINCM)

TCMMINCM is a 16-bit readable/writable register. TCMMINCM is available as a cycle lower limit register when the TCMMDS bit in TCMCR is set to 1 (operation is in cycle measurement mode).

In cycle measurement mode, a value that sets a lower limit on the measurement period can be set in TCMMINCM. When the second edge (selectable with the IEDG bit in TCMCR) of the measurement period is detected, the value in TCMCNT is transferred to TCMICR. At this time, the values in TCMICR and TCMMINCM are compared. The MINUDF flag in TCMCSR is set to 1 if the value in TCMICR is smaller than that in TCMMINCM. TCMMLCM should always be accessed in 16-bit units and cannot be accessed in 8-bit units. TCMMINCM is initialized to H'0000.

11.3.4 TCM Input Capture Register (TCMICR)

TCMICR is a 16-bit read-only register. In timer mode, the value in TCMCNT is transferred to TCMICR on the edge selected by the IEDG bit in TCMCR. At the same time, the ICPF flag in TCMCSR is set to 1. In cycle measurement mode, the value in TCMCNT is transferred to TCMICR on detection of the second edge of the measurement period. At this time, the ICPF flag in TCMCSR is set to 1. TCMICR should always be accessed in 16-bit units and cannot be accessed in 8-bit units. TCMICR is initialized to H'0000.

11.3.5 TCM Input Capture Buffer Register (TCMICRF)

TCMICRF is a 16-bit read only register. TCMICRF can be used as TCMICR buffer register. When input capture is generated, the value in TCMICR is transferred to TCMICRF.

TCMICR and TCMICRF should always be accessed in 16-bit units and cannot be accessed in 8-bit units. TCMICRF is initialized to H'0000.

11.3.6 TCM Status Register (TCMCSR)

TCMCSR is an 8-bit readable/writable register that controls operation of the interrupt sources.

Bit	Bit Name	Initial Value	R/W	Description
7	OVF	0	R/(W)*	Timer Overflow This flag indicates that the TCMCNT has overflowed. [Setting condition] Overflow of TCMCNT (change in value from H'FFFF to H'0000) [Clearing condition] Reading OVF when OVF = 1 and then writing 0 to OVF.
6	MAXOVF	0	R/(W)*	Measurement Period Upper Limit Overflow This flag indicates that the measured number of cycles of the waveform for measurement in cycle measurement mode has reached the upper limit set in TCMMLCM, causing an overflow. [Setting condition] A greater value for TCMICR than TCMMLCM [Clearing condition] Reading MAXOVF when MAXOVF = 1 and then writing 0 to MAXOVF
5	CMF	0	R/(W)*	Compare Match Flag (only valid in timer mode) [Setting condition] When the values in TCMCNT and TCMMLCM match. [Clearing condition] Reading CMF when CMF = 1 and then writing 0 to CMF Note: CMF is not set in cycle measurement mode, even when the values in TCMCNT and TCMMLCM match.
4	CKSEG	0	R/W	External Clock Edge Select When bits CKS2 to CKS0 in TCMCR are set to B'111, this bit selects the edge for counting of external count clock edge. 0: Count falling edges of the external clock. 1: Count rising edges of the external clock.

Bit	Bit Name	Initial Value	R/W	Description
3	ICPF	0	R/(W)*	Input Capture Generation Timer mode: The flag indicates that the value in TCMCNT has been transferred to TCMICR on generation of an input capture signal. This flag is set when the input capture signal is generated, i.e. on detection of the edge selected by the IEDGD bit on the TCMCYI input pin. Cycle measurement mode: The flag indicates that the value in TCMCNT has been transferred to TCMICR on detection of the second edge (rising or falling as determined by the IEDG bit in TCMCR) during the measurement period. [Setting condition] Generation of the input capture signal [Clearing condition] Reading ICPF when ICPF = 1 and then writing 0 to ICPF
2	MINUDF	0	R/(W)*	Measurement Period Lower Limit Underflow This flag indicates that the measured number of cycles of the waveform for measurement in cycle measurement mode has reached the lower limit set in TCMMINCM, causing an underflow. [Setting condition] A smaller value for TCMICR than TCMMINCM [Clearing condition] Reading MINUDF when MINUDF = 1 and then writing 0 to MINUDF
1	MCICTL	0	R/W	TCMMCI Input Polarity Inversion 0: TCMMCI input is inverted for use. 1: TCMMCI input is directly used. Note: Change this bit when CST = 0 and TCMMDS = 0
2 to 0	—	All 0	R/W	Reserved The initial value should not be changed.

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

11.3.7 TCM Control Register (TCMCR)

TCMCR is an 8-bit readable/writable register. TCMCR selects input capture input edge, counter start, and counter clock, and controls operation mode.

Bit	Bit Name	Initial Value	R/W	Description
7	CST	0	R/W	Counter Start In timer mode, setting this bit to 1 starts counting by TCMCNT; clearing this bit stops counting by TCMCNT. Then, the counter is initialized to H'0000, and input-capture operation stops. Clear this bit and thus return TCMCNT to H'0000 in initialization for cycle measurement mode.
6	POCTL	0	R/W	TCMCYI Input Polarity Reversal 0: Use the TCMCYI input directly 1: Use the inverted TCMCYI input Note: Modify this bit while CST = 0 and TCMMDS = 0
5	CPSPE	0	R/W	Input Capture Stop Enable Controls whether or not counting up by TCMCNT and input-capture operation stop or continue when either of MAXOVF or MINUDF is set to 1 in cycle measurement mode. The bit does not affect operation in timer mode. 0: Counting up and input-capture operation continue when the flag is set to 1. 1: Counting up and input-capture operation are disabled when the flag is set to 1.

Bit	Bit Name	Initial Value	R/W	Description
4	IEDG	0	R/W	Input Edge Select In timer mode, selects the falling or rising edge of the TCMCYI input for use in input capture, in combination with the value of the POCTL bit. In cycle measurement mode, selects the falling or rising edge of the TCMCYI input for use in measurement, in combination with the value of the POCTL bit. POCTL = 0 0: Selects the rising edge of the TCMCYI input 1: Selects the falling edge of the TCMCYI input POCTL = 1 0: Selects the falling edge of the TCMCYI input 1: Selects the rising edge of the TCMCYI input
3	TCMMDS	0	R/W	TCM Mode Select Selects the TCM operating mode. 0: Timer mode The TCM provides compare match and input capture facilities. 1: Cycle measurement mode Setting this bit to 1 starts counting by TCMCNT. TCMCNT should be initialized to H'0000. Clear the CST in TCMCR to 0 before setting to cycle measurement mode.
2	CKS2	0	R/W	Clock Select 2, 1, 0 Selects the clock signal for input to TCMCNT. Note: Modify this bit when CST = 0 and TCMMDS = 0 000: Count ϕ/2 internal clock 001: Count ϕ/8 internal clock 010: Count ϕ/16 internal clock 011: Count ϕ/32 internal clock 100: Count ϕ/64 internal clock 101: Count ϕ/128 internal clock 110: Count ϕ/256 internal clock 111: Count external clock (select the external clock edge with CKSEG in TCMCSR.)
1	CKS1	0	R/W	
0	CKS0	0	R/W	

11.3.8 TCM Interrupt Enable Register (TCMIER)

TCMIER is an 8-bit readable/writable register that enables or disables interrupt requests.

Bit	Bit Name	Initial Value	R/W	Description
7	OVIE	0	R/W	Counter Overflow Interrupt Enable Enables or disables the issuing of interrupt requests on setting of the OVF flag in TCMCSR to 1. 0: Disable interrupt requests by OVF 1: Enable interrupt requests by OVF
6	MAXOVIE	0	R/W	Cycle Upper Limit Overflow Interrupt Enable Enables or disables the issuing of interrupt requests on setting of the MAXOVF flag in TCMCSR to 1. 0: Disable interrupt requests by MAXOVF 1: Enable interrupt requests by MAXOVF
5	CMIE	0	R/W	Compare Match Interrupt Enable Enables or disables the issuing of interrupt requests when the CMF bit in TCMCSR is set to 1. 0: Disable interrupt requests by CMF 1: Enable interrupt requests by CMF
4	TCMIPE	0	R/W	Input Capture Input Enable Enables or disables input to the pin. When using interrupt capture mode and cycle measurement mode, set this bit to 1. 0: Disable input 1: Enable input Note: Modify this bit when CST = 0 and TCMMDS = 0.
3	ICPIE	0	R/W	Input Capture Interrupt Enable Enables or disables interrupt requests when the ICPF flag in TCMCSR is set to 1. 0: Disable interrupt requests by ICPF 1: Enable interrupt requests by ICPF

Bit	Bit Name	Initial Value	R/W	Description
2	MINUDIE	0	R/W	Cycle Lower Limit Underflow Interrupt Enable Enables or disables the issuing of the TUDI interrupt requests when the MINUDF flag in TCMCSR is set to 1. 0: Disable interrupt requests by MINUDF 1: Enable interrupt requests by MINUDF
1	CMMS	0	R/W	Cycle Measurement Mode Selection Selects use of the TCMMCI signal in cycle measurement mode. 0: The TCMMCI signal is not used (cycle measurement is always performed). 1: The TCMMCI signal is used. When MCICTL in TCMCSR is 0, cycle measurement is performed only while TCMMCI is low. When MCICTL is 1, cycle measurement is performed only while TCMMCI is high. Note: Change this bit when CST = 0 and TCMMDS = 0.
0	—	0	R	Reserved This bit is always read as 0 and cannot be modified.

11.4 Operation

The TCM operates in timer mode or cycle measurement mode. TCM is in timer mode after a reset.

11.4.1 Timer Mode

When the TCMMDS bit in TCMCR is cleared to 0, TCM operates in timer mode.

(1) Counter Operation

TCMCNT operates as a free running counter in timer mode. TCMCNT starts counting up when the CST bit in TCMCR is set to 1. When TCMCNT overflows (the value changes from H'FFFF to H'0000), the OVF bit in TCMCSR is set to 1 and an interrupt request is generated if the OVIE bit in TCMIER is 1. Figure 11.2 shows an example of free running counter operation. In addition, figure 11.3 shows TCMCNT count timing of external clock operation. The external clock should have a pulse width of no less than 1.5 cycles. The counter will not operate correctly if the pulses are narrower than this.

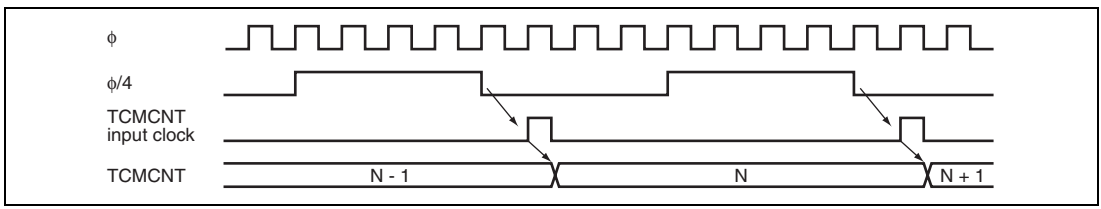


Figure 11.2 Example of Free Running Counter Operation

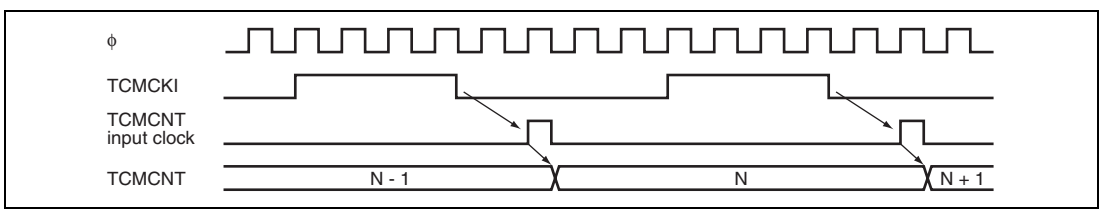


Figure 11.3 Count Timing of External Clock Operation (Falling Edges)

(2) Input Capture

The value in TCMCNT is transferred to TCMICR by detecting input edge of TCMCYI pin in timer mode. At this time, the ICPF flag in TCMCSR is set. Detection of rising or falling edges is selectable with the setting of the IEDG bit in TCMCR. Figure 11.4 shows an example of the timing of input capture operations and figure 11.5 shows buffer operation of input capture.

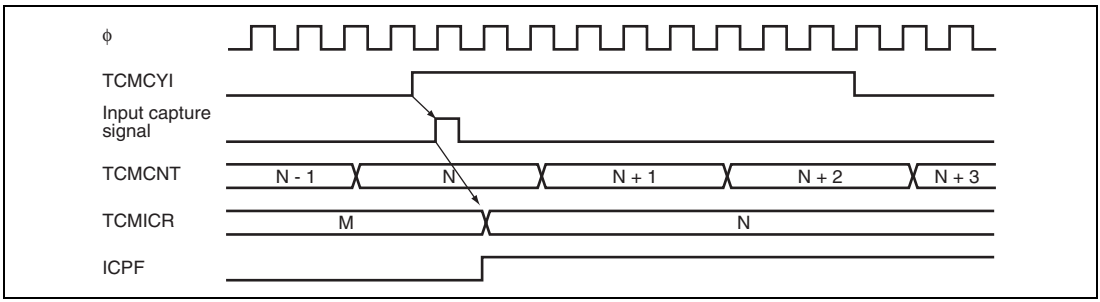


Figure 11.4 Input Capture Operation Timing (Sensing of Rising Edges)

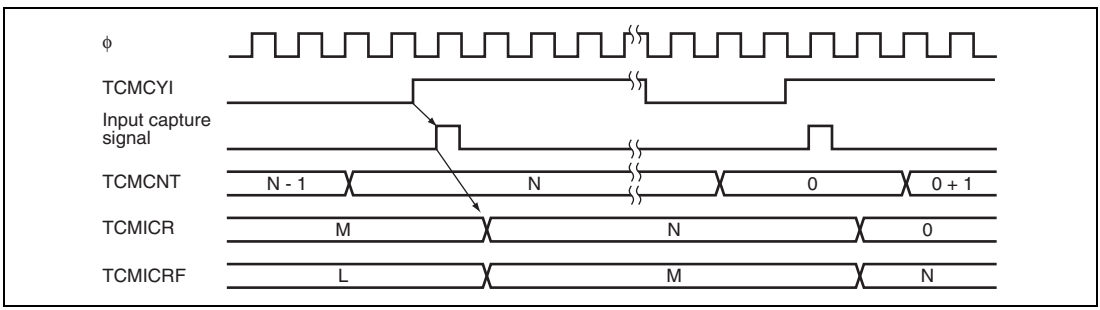


Figure 11.5 Buffer Operation of Input Capture

(3) CMF Set Timing when a Compare Match occurs

The CMF flag in TCMCSR is set in the last state where the values in TCMCNT and TCMMLCM match in timer mode. Therefore, a compare match signal is not generated until a further cycle of the TCMCNT input clock is generated after a match between the values in TCMCNT and TCMMLCM. For details, see section 11.6.2, Conflict between TCMMLCM Write and Compare Match. Figure 11.6 shows the timing with which the CMF flag is set.

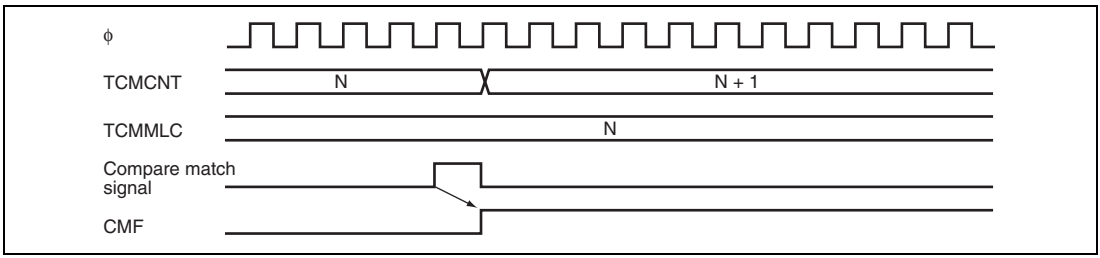


Figure 11.6 Timing of CMF Flag Setting on a Compare Match

11.4.2 Cycle Measurement Mode

When the TCMMDS bit in TCMCR is set to 1, the TCM operates in cycle measurement mode.

(1) Counter Operation

Setting the TCMMDS bit in TCMCR to 1 selects cycle measurement mode, in which counting up proceeds regardless of the setting of the CST bit in TCMCR. TCMCNT is cleared to H'0000 on detection of the first edge in the measurement period and counts up from there. Figure 11.7 shows an example of counter operation in cycle measurement mode.

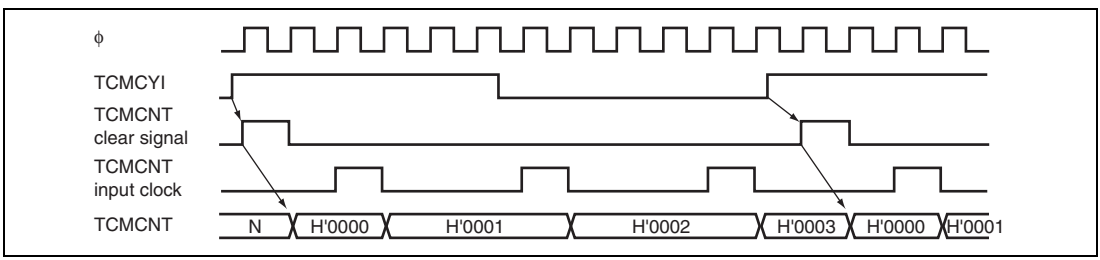


Figure 11.7 Example of Counter Operation in Cycle Measurement Mode

(2) Measuring a Cycle

In cycle measurement mode, one cycle of the input waveform for TCM form one measurement cycle. Start by setting TCMMDS = 0 and then set CST = 0, which clears TCMCNT to H'0000. After that, set an upper or lower limit on the measurement cycle in the TCMMLCM/TCMMINCM register. Finally, place the timer in cycle measurement mode by setting the TCMMDS bit in TCMCR to 1. TCMCNT will count cycles of the selected clock. On detection of the first edge (either rising or falling as selected with the IEDG bit in TCMCR) of the measurement cycle, TCMCNT is automatically cleared to H'0000. On detection of the second edge, the value in TCMCNT is transferred to TCMICR. At this time, the value in TCMICR is compared with the value in TCMMLCM/TCMMINCM. If TCMICR is larger than TCMMLCM, the MAXOVF bit in TCMCSR is set to 1. If TCMICR is smaller than TCMMINCM, the MINUDF bit in TCMCSR is set to 1. If generation of the corresponding interrupt request is enabled by the setting in TCMIER, the request is generated. In addition, on detection of the third edge, TCMCNT is cleared to H'0000, and the next round of measurement starts.

When the CPSPE bit in TCMCR has been cleared to 0, the next round of cycle measurement will start, even if the MAXOVF/MINUDF flag is set to 1.

If the MAXOVF/MINUDF flag is set to 1 while the CPSPE bit in TCMCR is set to 1, counting up by TCMCNT stops and so does cycle measurement. Subsequently clearing MAXOVF/MINUDF to 0 automatically clears TCMCNT to H'0000, and counting up for cycle measurement is then restarted.

Figure 11.8 shows an example of timing in speed measurement.

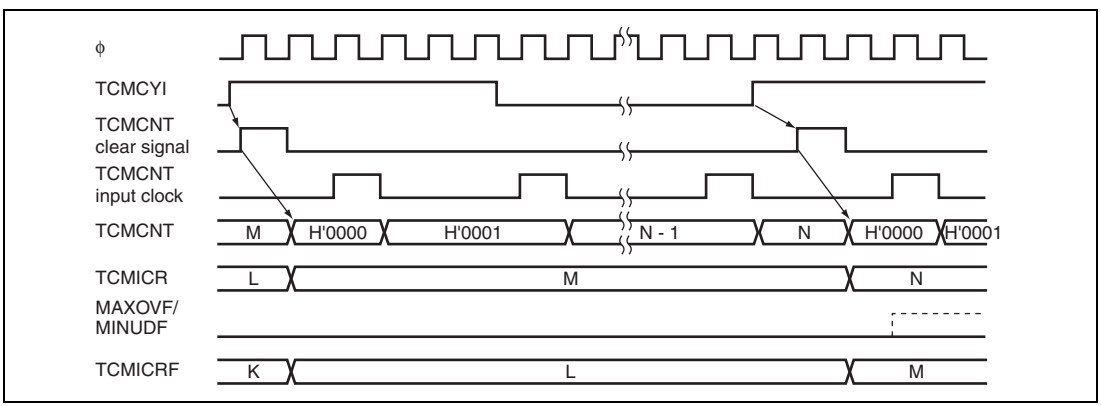


Figure 11.8 Example of Timing in Cycle Measurement

When the CMMS bit in TCMIER is set to 1, cycle measurement is performed only while the TCMMCI signal is high (MCICTL in TCMCSR is 0). Figure 11.9 shows an example of timing in cycle measurement when the CMMS bit is set to 1.

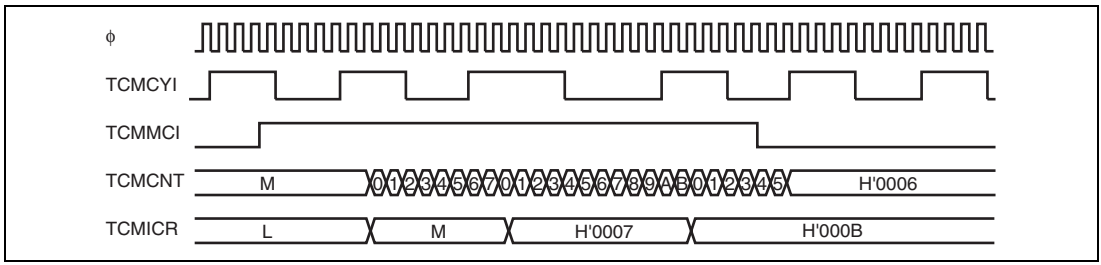


Figure 11.9 Example of Timing in Cycle Measurement when the CMMS Bit is Set to 1

(3) Determination of External Event (TCMCYI) Stoppage

The timer overflow flag can be used to determine the external event (TCMCYI) stopped state. Either of two sets of conditions represents the external event stopped state.

The external event can be considered to have stopped when a timer overflow is generated within the period from the start of cycle measurement mode to detection of the first edge (rising or falling as selected with the IEDG bit in TCMCR).

Figure 11.10 shows an example of the timing of the external event stopped state (1).

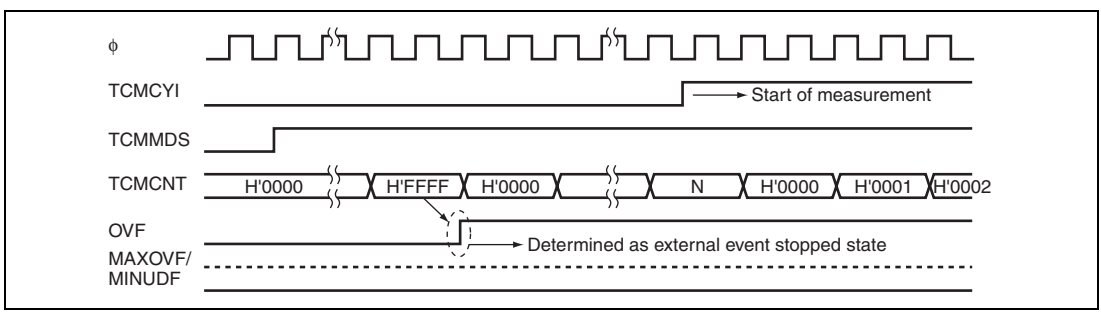


Figure 11.10 Example of Timing in External Event Stopped State (1)

Cycle measurement stops if MAXOVF/MINUDF is set to 1 while the CPSPE bit in TCMCR is set to 1. Subsequently clearing MAXOVF/MINUDF to 0 restarts cycle measurement. In this case, the external event can be considered to have stopped if a timer overflow is generated before detection of the first edge.

Figure 11.11 shows an example of the timing of the external event stopped state (2).

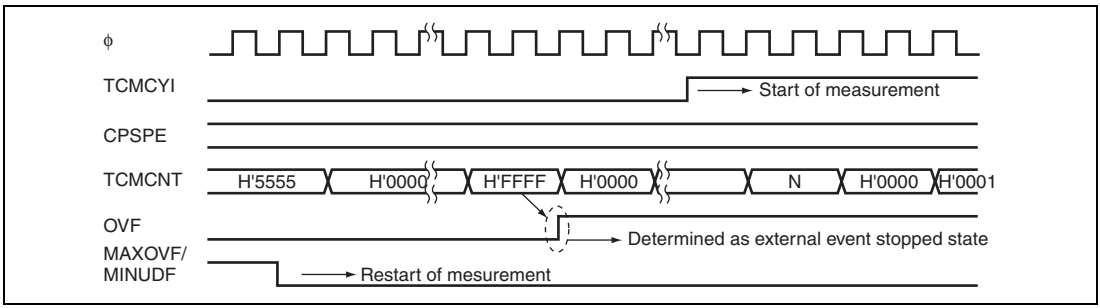


Figure 11.11 Example of Timing in External Event Stopped State (2)

(4) Example of Settings for Cycle Measurement Mode

Figure 11.12 shows an example of the flow when cycle measurement mode is to be used.

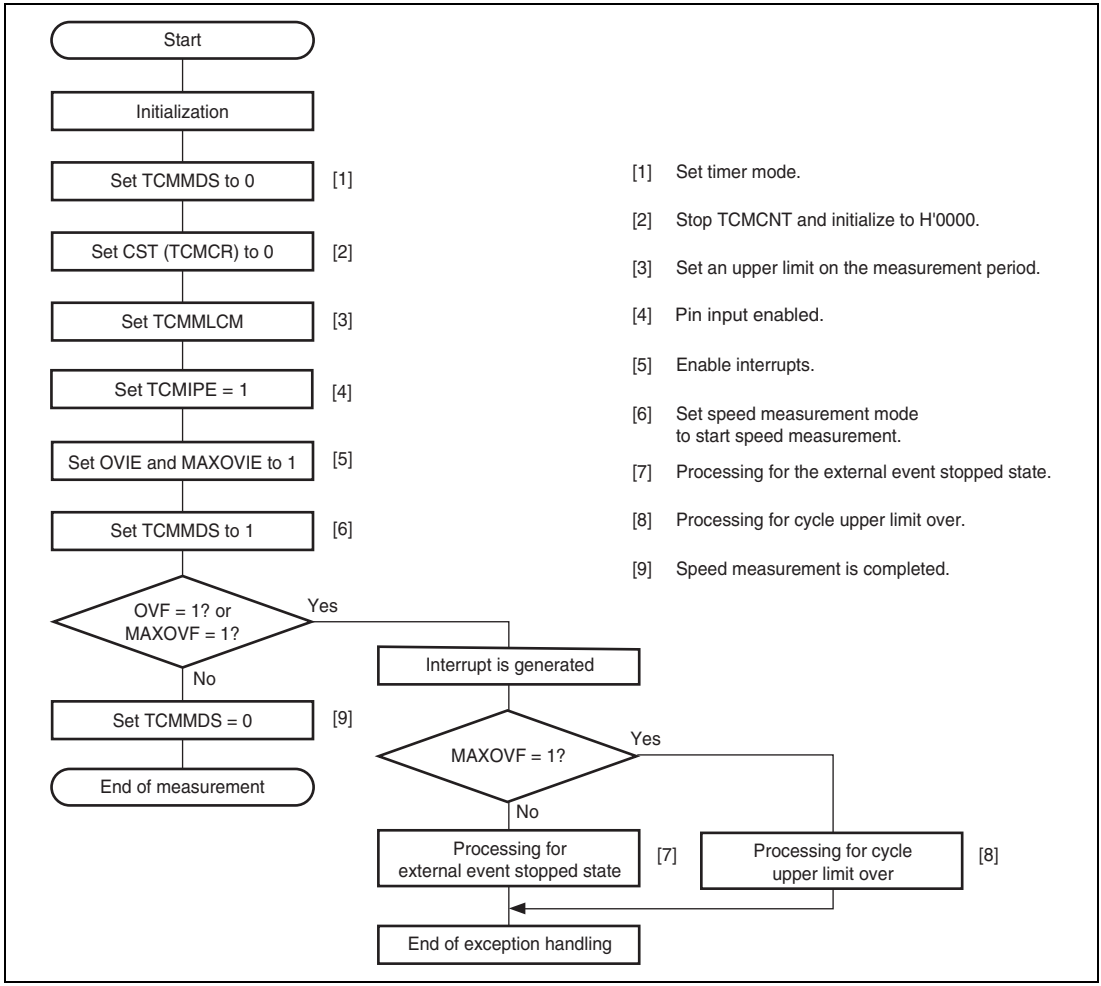


Figure 11.12 Example of Cycle Measurement Mode Settings

11.5 Interrupt Sources

TCM has five interrupt sources: TICI, TCMI, TOVMI, TUDI, and TOVI. Each interrupt source is either enabled or disabled by the corresponding interrupt enable bit in TCMIER and independently transferred to the interrupt controller. Since a single vector address is allocated for each type of interrupt source from all channels, the flags must be used to discriminate between the sources.

Table 11.3 lists the interrupt sources in priority order.

Table 11.3 TCM Interrupt Sources

Channel	Name	Interrupt Source	Interrupt Flag	Priority
TCM_0	TICI0	TCMICR_0 input capture	ICPF_0	High
	TCMI0	TCMMLCM_0 compare match	CMF_0	
	TOVMI0	TCMMLCM_0 overflow	MAXOVF_0	
	TUDI0	TCMMINCM_0 underflow	MINUDF_0	
	TOVI0	TCMCNT_0 overflow	OVF_0	
TCM_1	TICI1	TCMICR_1 input capture	ICPF_1	
	TCMI1	TCMMLCM_1 compare match	CMF_1	
	TOVMI1	TCMMLCM_1 overflow	MAXOVF_1	
	TUDI1	TCMMINCM_1 underflow	MINUDF_1	
	TOVI1	TCMCNT_1 overflow	OVF_1	
TCM_2	TICI2	TCMICR_2 input capture	ICPF_2	
	TCMI2	TCMMLCM_2 compare match	CMF_2	
	TOVMI2	TCMMLCM_2 overflow	MAXOVF_2	
	TUDI2	TCMMINCM_2 underflow	MINUDF_2	
	TOVI2	TCMCNT_2 overflow	OVF_2	
TCM_3	TICI3	TCMICR_3 input capture	ICPF_3	
	TCMI3	TCMMLCM_3 compare match	CMF_3	
	TOVMI3	TCMMLCM_3 overflow	MAXOVF_3	
	TUDI3	TCMMINCM_3 underflow	MINUDF_3	
	TOVI3	TCMCNT_3 overflow	OVF_3	Low

11.6 Usage Notes

11.6.1 Conflict between TCMCNT Write and Count-Up Operation

When a conflict between TCMCNT write and count-up operation occurs in the second half of the TCMCNT write cycle, TCMCNT is not incremented and writing to TCMCNT takes priority. Figure 11.13 shows the timing of this conflict.

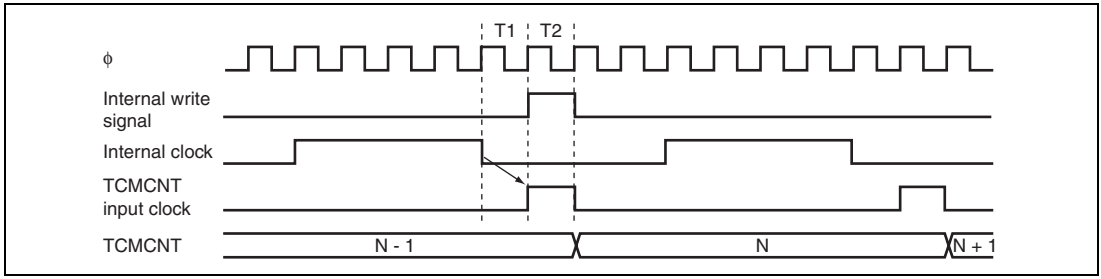


Figure 11.13 Conflict between TCMCNT Write and Count-Up Operation

11.6.2 Conflict between TCMMLCM Write and Compare Match

When a conflict between TCMMLCM write and a compare match should occur in the second half of a cycle of writing to TCMMLCM, writing to TCMMLCM takes priority and the compare match signal is inhibited. Figure 11.14 shows the timing of this conflict.

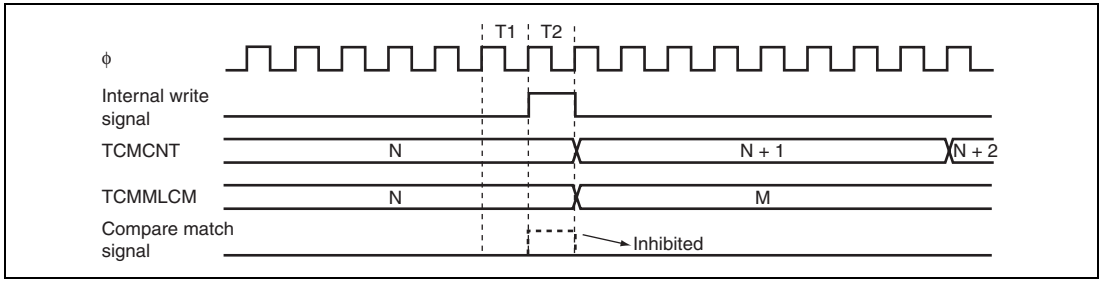


Figure 11.14 Conflict between TCMMLCM Write and Compare Match

11.6.3 Conflict between TCMICR Read and Input Capture

When operation is in timer mode and the corresponding input capture signal is detected during reading of TCMICR, the input capture signal is delayed by one system clock (ϕ). Figure 11.15 shows the timing of this conflict.

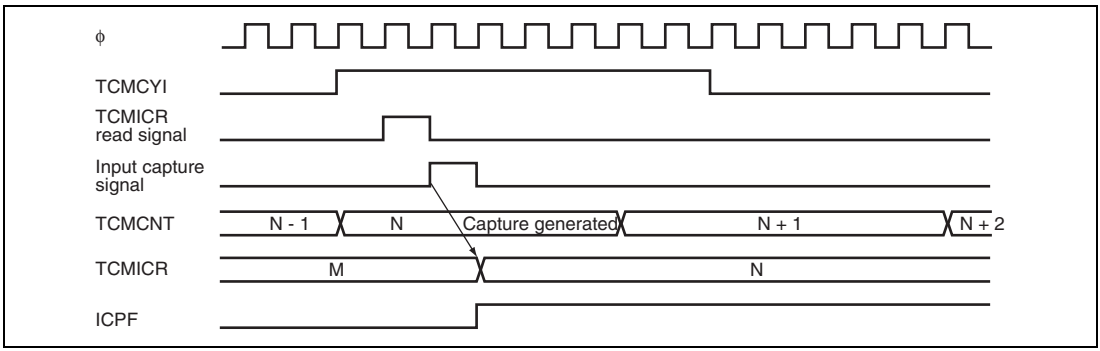


Figure 11.15 Conflict between TCMICR Read and Input Capture

11.6.4 Conflict between Edge Detection in Cycle Measurement Mode and Writing to TCMMLCM or TCMMINCM

If the selected edge of TCMCYI is detected in the second half of a cycle of writing to the register (TCMMLCM or TCMMINCM) in cycle measurement mode, the detected edge signal is delayed by one cycle of the system clock (ϕ).

Figure 11.16 shows the timing of this conflict.

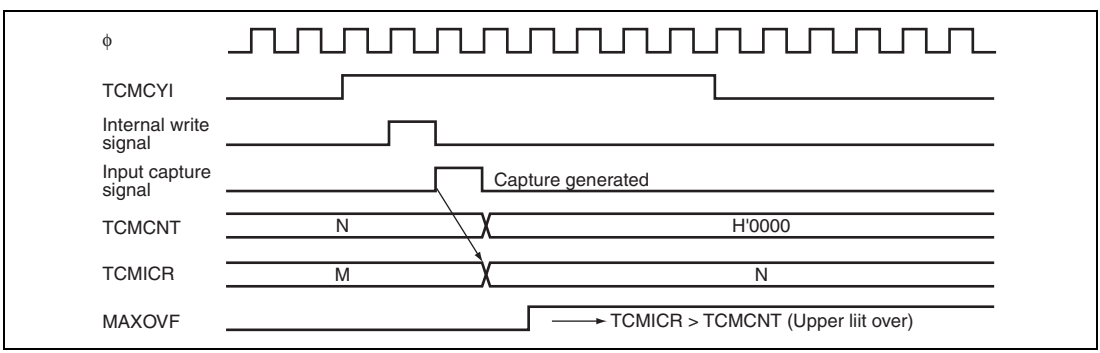


Figure 11.16 Conflict between Edge Detection and Register Write (Cycle Measurement Mode)

11.6.5 Conflict between Edge Detection in Cycle Measurement Mode and Clearing of TCMMDS Bit in TCMCR

If the CST bit in TCMCR is set to 1 in cycle measurement mode, and the TCMMDS bit in TCMCR is cleared, but the selected edge from TCMCYI is detected at the same time, detection of the selected edge will cause the timer to continue to operate in cycle measurement mode. The timer will not make the transition to timer mode until the next detection of the selected edge. Thus, ensure that the CST bit is cleared to 0 in cycle measurement mode.

Figure 11.17 shows the timing of this conflict.

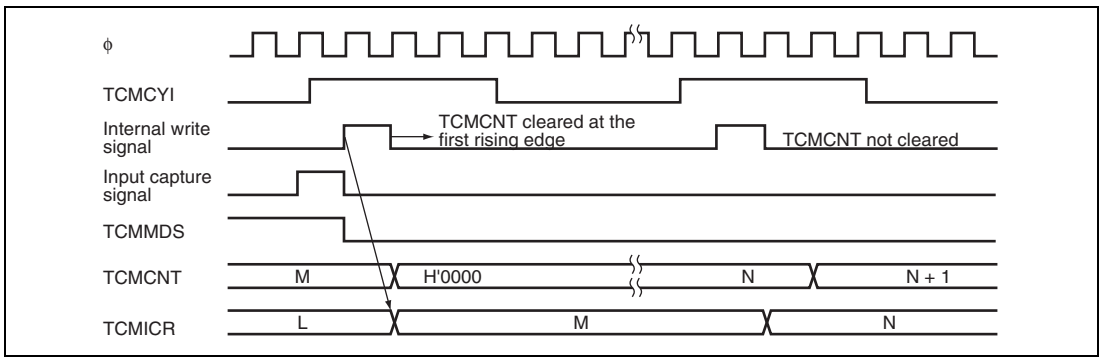


Figure 11.17 Conflict between Edge Detection and Clearing of TCMMDS (to Switch from Cycle Measurement Mode to Timer Mode)

11.6.6 Settings of TCMCKI and TCMMCI

TCMCKI and TCMMCI are multiplexed on the same pin of this LSI. Therefore, the selected external clock and the TCMMCI signal cannot be used at the same time. Do not make the settings CKS2 to CKS0 = B'111 and CMMS = B'1.

11.6.7 Setting for Module Stop Mode

The module-stop control register can be used to select either continuation or stoppage of TCM operation in module-stopped mode. The default setting is for TCM operation to stop. TCM registers become accessible on release from module stop mode. For details, see section 26, Power-Down Modes.

Section 12 16-Bit Duty Period Measurement Timer (TDP)

This LSI has a three-channel, 16-bit duty period measurement timer (TDP). The TDP uses a 16-bit counter as the basis for measuring input waveforms and the pulse width.

12.1 Features

- Capable of measuring input-waveform periods and the pulse width
- Selectable measured edge
- 16-bit compare match
- 16-bit resolution
- Selectable counter clock
  Any of seven internal clocks or an external clock can be selected.
- Seven interrupt sources
  Counter overflow
  Cycle upper limit overflow
  Cycle lower limit underflow
  Pulse width upper limit overflow
  Pulse width lower limit underflow
  Compare match
  Generation of input capture

Figure 12.1 shows a block diagram of the TDP.

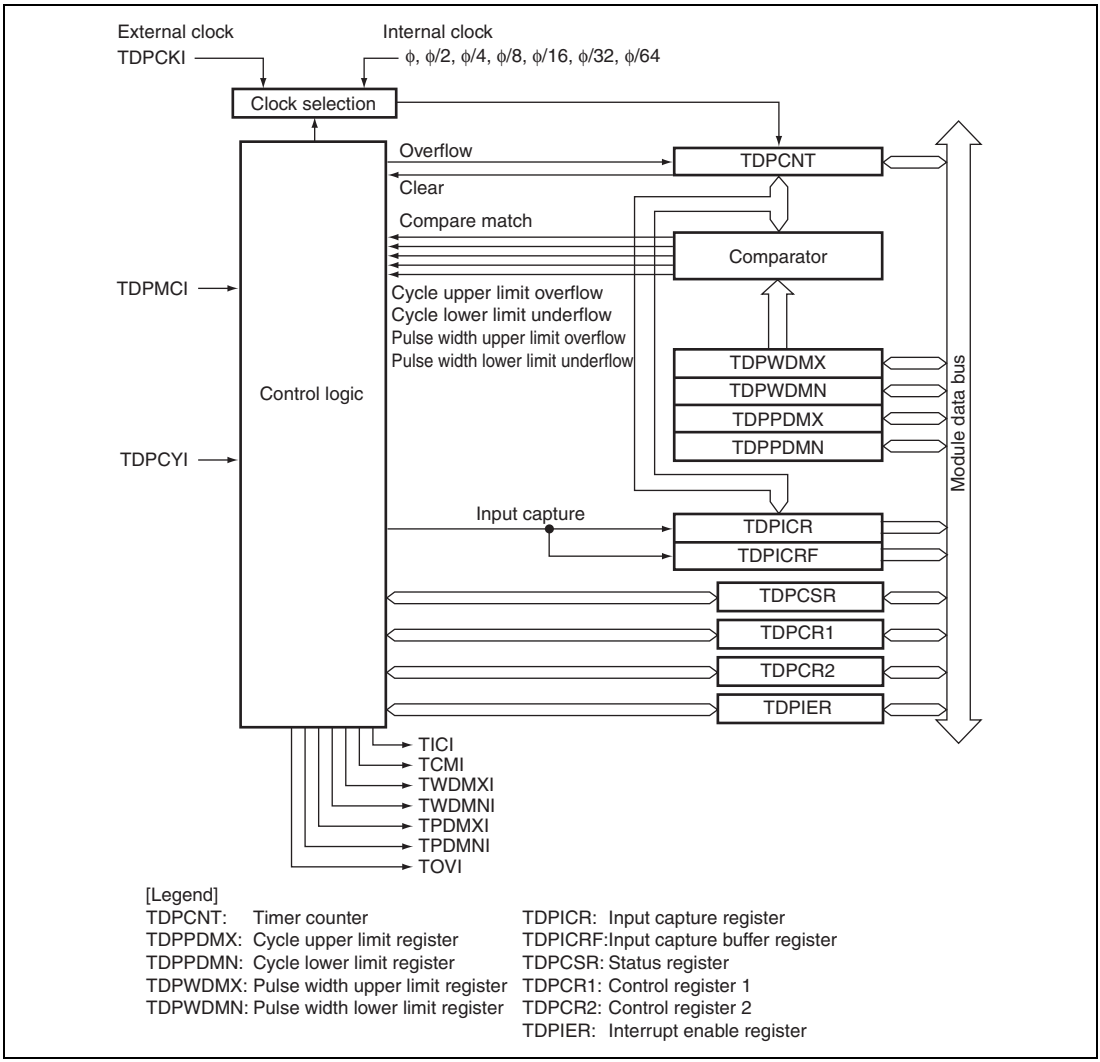


Figure 12.1 Block Diagram of TDP

12.2 Input/Output Pins

Table 12.1 lists the pin configuration of the TDP.

Table 12.1 Pin Configuration

Channel	Pin Name	I/O	Function
0	TDPCKI0 (TDPMCI0)	Input	External counter clock input Cycle measurement control input
	TDPCYI0	Input	External event input
1	TDPCKI1 (TDPMCI1)	Input	External counter clock input Cycle measurement control input
	TDPCYI1	Input	External event input
2	TDPCKI2 (TDPMCI2)	Input	External counter clock input Cycle measurement control input
	TDPCYI2	Input	External event input

12.3 Register Descriptions

The TDP has the following registers.

Table 12.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Data Bus Width
Channel 0	TDP timer counter_0	TDPCNT_0	R/W	H'0000	H'FB40	16
	TDP pulse width upper limit register_0	TDPWDMX_0	R/W	H'FFFF	H'FB42	16
	TDP pulse width lower limit register_0	TDPWDMN_0	R/W	H'0000	H'FB44	16
	TDP cycle upper limit register_0	TDPPDMX_0	R/W	H'FFFF	H'FB46	16
	TDP cycle lower limit register_0	TDPPDMN_0	R/W	H'0000	H'FB50	16
	TDP input capture register_0	TDPICR_0	R	H'0000	H'FB48	16
	TDP input capture buffer register_0	TDPICRF_0	R	H'0000	H'FB4A	16
	TDP status register_0	TDPCSR_0	R/W	H'00	H'FB4C	8
	TDP control register1_0	TDPCR1_0	R/W	H'00	H'FB4D	8
	TDP control register2_0	TDPCR2_0	R/W	H'00	H'FB4F	8
	TDP interrupt enable register_0	TDPIER_0	R/W	H'00	H'FB4E	8

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Data Bus Width
Channel 1	TDP timer counter_1	TDPCNT_1	R/W	H'0000	H'FB60	16
	TDP pulse width upper limit register_1	TDPWDMX_1	R/W	H'FFFF	H'FB62	16
	TDP pulse width lower limit register_1	TDPWDMN_1	R/W	H'0000	H'FB64	16
	TDP cycle upper limit register_1	TDPPDMX_1	R/W	H'FFFF	H'FB66	16
	TDP cycle lower limit register_1	TDPPDMN_1	R/W	H'0000	H'FB70	16
	TDP input capture register_1	TDPICR_1	R	H'0000	H'FB68	16
	TDP input capture buffer register_1	TDPICRF_1	R	H'0000	H'FB6A	16
	TDP status register_1	TDPCSR_1	R/W	H'00	H'FB6C	8
	TDP control register1_1	TDPCR1_1	R/W	H'00	H'FB6D	8
	TDP control register2_1	TDPCR2_1	R/W	H'00	H'FB6F	8
	TDP interrupt enable register_1	TDPIER_1	R/W	H'00	H'FB6E	8
Channel 2	TDP timer counter_2	TDPCNT_2	R/W	H'0000	H'FB80	16
	TDP pulse width upper limit register_2	TDPWDMX_2	R/W	H'FFFF	H'FB82	16
	TDP pulse width lower limit register_2	TDPWDMN_2	R/W	H'0000	H'FB84	16
	TDP cycle upper limit register_2	TDPPDMX_2	R/W	H'FFFF	H'FB86	16
	TDP cycle lower limit register_2	TDPPDMN_2	R/W	H'0000	H'FB90	16
	TDP input capture register_2	TDPICR_2	R	H'0000	H'FB88	16
	TDP input capture buffer register_2	TDPICRF_2	R	H'0000	H'FB8A	16
	TDP status register_2	TDPCSR_2	R/W	H'00	H'FB8C	8
	TDP control register 1_2	TDPCR1_2	R/W	H'00	H'FB8D	8
	TDP control register 2_2	TDPCR2_2	R/W	H'00	H'FB8F	8
	TDP interrupt enable register_2	TDPIER_2	R/W	H'00	H'FB8E	8

12.3.1 TDP Timer Counter (TDPCNT)

TDPCNT is a 16-bit readable/writable up-counter. The input clock is selected by bits CKS2 to CKS0 in TDPCR1. When CKS2 to CKS0 are set to B'111, the external clock is selected. Rising or falling edge is selected by CKSEG in TDPCSR.

When TDPCNT overflows (H'FFFF changes to H'0000), the OVF flag in TDPCSR is set to 1. In timer mode, TDPCNT is initialized to H'0000 when the CST bit in TDPCR1 is cleared. In cycle measurement mode, TDPCNT is cleared when the first edge (the edge selected by the IEDG bit in TDPCR1) of the measurement period (equal to one input waveform period) is detected.

In timer mode, TDPCNT is always writable. In cycle measurement mode, TDPCNT cannot be modified. TDPCNT must always be accessed in 16-bit units and cannot be accessed in 8-bit units. TDPCNT is initialized to H'0000.

12.3.2 TDP Pulse Width Upper Limit Register (TDPWDMX)

TDPWDMX is a 16-bit readable/writable register. When the TDPMDS bit in TDPCR1 is cleared (timer mode), TDPWDMX is available as a compare match register. When the TDPMDS bit in TDPCR1 is set to 1 (cycle measurement mode), TDPWDMX is available as a pulse width upper limit register.

In timer mode, the TDPWDMX value is continually compared with the TDPCNT value. If the values match, the CMF flag in TDPCSR is set to 1. Note, however, that comparison is disabled in the second half of a write cycle to TDPWDMX.

In cycle measurement mode, TDPWDMX can be used to set the upper limit value of the measurement pulse width. When the second edge (the second edge of this period) of the measurement period is detected, the TDPCNT value is transferred to TDPICR and the values of TDPICR and TDPWDMX are compared. If the TDPICR value is greater than the TDPWDMX value, the TWDMXOVF flag in TDPCSR is set to 1. TDPWDMX must always be accessed in 16-bit units and cannot be accessed in 8-bit units. TDPWDMX is initialized to H'FFFF.

12.3.3 TDP Pulse Width Lower Limit Register (TDPWDMN)

TDPWDMN is a 16-bit readable/writable register. When the TDPMDS bit in TDPCR1 is set to 1 (cycle measurement mode), TDPWDMN is available as a pulse width lower limit register.

In cycle measurement mode, TDPWDMN can be used to set the lower limit value of measurement pulse width. When the second edge (the second edge of this period) of the measurement period is detected, the TDPCNT value is transferred to TDPICR and the values of TDPICR and TDPWDMN are compared. If the TDPICR value is less than the TDPWDMN value, the TWDMNUDF flag in TDPCSR is set to 1. TDPWDMN must always be accessed in 16-bit units and cannot be accessed in 8-bit units. TDPWDMN is initialized to H'0000.

12.3.4 TDP Cycle Upper Limit Register (TDPPDMX)

TDPPDMX is a 16-bit readable/writable register. When the TDPMDS bit in TDPCR1 is set to 1 (cycle measurement mode), TDPPDMX is available as a cycle upper limit register.

In cycle measurement mode, TDPPDMX can be used to set the upper limit value of measurement period. When the third edge (the first edge of the next period) of the measurement period is detected, the TDPCNT value is transferred to TDPICR and the values of TDPICR and TDPPDMX are compared. If the TDPICR value is greater than the TDPPDMX value, the TPDMXOVF flag in TDPCSR is set to 1. TDPPDMX must always be accessed in 16-bit units and cannot be accessed in 8-bit units. TDPPDMX is initialized to H'FFFF.

12.3.5 TDP Cycle Lower Limit Register (TDPPDMN)

TDPPDMN is a 16-bit readable/writable register. When the TDPMDS bit in TDPCR1 is set to 1 (cycle measurement mode), TDPPDMN is available as a cycle lower limit register.

In cycle measurement mode, TDPPDMN can be used to set the lower limit value of measurement period. When the third edge (the first edge of the next period) of the measurement period is detected, the TDPCNT value is transferred to TDPICR and the values of TDPICR and TDPPDMN are compared. If the TDPICR value is less than the TDPPDMN value, the TPDMNUDF flag in TDPCSR is set to 1. TDPPDMN must always be accessed in 16-bit units and cannot be accessed in 8-bit units. TDPPDMN is initialized to H'0000.

12.3.6 TDP Input Capture Register (TDPICR)

TDPICR is a 16-bit read-only register. In timer mode, the TDPCNT value is transferred to TDPICR on the edge selected by the IEDG bit in TDPCR1, and the ICPF flag in TDPCSR is set to 1. In cycle measurement mode, the TDPCNT value is transferred to TDPICR when the first edge of the measurement period is detected. At the same time, the ICPF flag in TDPCSR is set to 1. TDPICR must always be accessed in 16-bit units and cannot be accessed in 8-bit units. TDPICR is initialized to H'0000.

12.3.7 TDP Input Capture Buffer Register (TDPICRF)

TDPICRF is a 16-bit read-only register. TDPICRF can be used as a TDPICR buffer register. When input capture occurs, the TDPICR value is transferred to TDPICRF.

TDPICRF must always be accessed in 16-bit units and cannot be accessed in 8-bit units. TDPICRF is initialized to H'0000.

12.3.8 TDP Status Register (TDPCSR)

TDPCSR indicates the status flags and selects the external clock edge.

Bit	Bit Name	Initial Value	R/W	Description
7	OVF	0	R/(W)*	Timer Overflow This flag indicates a TDPCNT overflow. [Setting condition] • TDPCNT overflow (H'FFFF changes to H'0000) [Clearing condition] • Reading OVF when OVF = 1 and then writing 0 to OVF
6	TWDMXOVF	0	R/(W)*	Pulse Width Upper Limit Overflow This flag indicates that the waveform pulse width measured in cycle measurement mode has exceeded the upper limit specified in TDPWDMX. [Setting condition] • When TDPICR is greater than TDPWDMX [Clearing condition] • Reading TWDMXOVF when TWDMXOVF = 1 and then writing 0 to TWDMXOVF

Bit	Bit Name	Initial Value	R/W	Description
5	TWDMNUDF	0	R/(W)*	Pulse Width Lower Limit Underflow This flag indicates that the waveform pulse width measured in cycle measurement mode is below the lower limit specified in TDPWDMN. [Setting condition] • When TDPICR is less than TDPWDMN [Clearing condition] • Reading TWDMNUDF when TWDMNUDF = 1 and then writing 0 to TWDMNUDF
4	TPDMXOVF	0	R/(W)*	Cycle Upper Limit Overflow This flag indicates that the waveform period measured in cycle measurement mode has exceeded the upper limit specified in TDPPDMX. [Setting condition] • When TDPICR is greater than TDPPDMX [Clearing condition] • Reading TPDMXOVF when TPDMXOVF = 1 and then writing 0 to TPDMXOVF
3	ICPF	0	R/(W)*	Input Capture Generation In timer mode, this flag indicates that the value in TDPCNT was transferred to TDPICR when an input capture signal was generated. This flag is set when the input capture signal selected by the IEDG bit is generated on the TDPCYI input pin. In cycle measurement mode, this flag indicates that the value in TDPCNT was transferred to TDPICR when the rising or falling edge of the PWM waveform was detected. [Setting condition] • When an input capture signal is generated [Clearing condition] • Reading ICPF when ICPF = 1 and then writing 0 to ICPF

Bit	Bit Name	Initial Value	R/W	Description
2	CMF	0	R/(W)*	Compare Match Flag (valid only in timer mode) [Setting condition] • When the TDPCNT value matches the TDPWDMX value in timer mode [Clearing condition] • Reading CMF when CMF = 1 and then writing 0 to CMF Note: In cycle measurement mode, even though the TDPCNT value matches the TDPWDMX value, CMF is not set to 1.
1	CKSEG	0	R/(W)*	External Clock Edge Select When CKS2 to CKS0 in TDPCR1 are set to B'111 (external clock), this bit selects the edge for counting of the external count clock edges. 0: Falling edges of the external clock are counted 1: Rising edges of the external clock are counted
0	TPDMNUDF	0	R/(W)*	Cycle Lower Limit Underflow This flag indicates that the waveform period measured in cycle measurement mode is below the lower limit specified in TDPPDMN. [Setting condition] • When TDPICR is less than TDPPDMN [Clearing condition] • Reading TPDMNUDF when TPDMNUDF = 1 and then writing 0 to TPDMNUDF

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

12.3.9 TDP Control Register 1 (TDPCR1)

TDPCR1 selects the input capture input edge, starts the TDPCNT counter, selects the counter clock, and controls the operating mode.

Bit	Bit Name	Initial Value	R/W	Description
7	CST	0	R/W	Counter Start In timer mode, setting this bit to 1 starts counting by TDPCNT, and clearing it stops counting. After the bit is cleared, the counter is initialized to H'0000, and the input-capture operation stops. Clear this bit to initialize TDPCNT to H'0000 before setting to cycle measurement mode.
6	POCTL	0	R/W	TDPCYI Input Polarity Inversion 0: TDPCYI input is used directly 1: TDPCYI input is inverted for use Note: Change this bit when CST = 0 and TDPMDS = 0.
5	CPSPE	0	R/W	Input Capture Stop Enable Controls whether counting up by TDPCNT and input-capture operation stop or continue when any of the TPDMXOVF, TPDMNUDF, TWDMXOVF, and TWDMNUDF flags is set to 1 in cycle measurement mode. This bit does not affect operation in timer mode. 0: Counting up and input-capture operation continue when any of the flags is set to 1. 1: Counting up and input-capture operation stop when any of the flags is set to 1.

Bit	Bit Name	Initial Value	R/W	Description
4	IEDG	0	R/W	Input Edge Select In timer mode, in combination with the value of the POCTL bit, selects the falling or rising edge of the TDPCYI input for capturing input. In cycle measurement mode, this bit does not affect operation. When POCTL = 0 0: The falling edge of TDPCYI input is selected 1: The rising edge of TDPCYI input is selected When POCTL = 1 0: The rising edge of TDPCYI input is selected 1: The falling edge of TDPCYI input is selected
3	TDPMDS	0	R/W	TDP Mode Select Selects the TDP operating mode. 0: Timer mode In timer mode, the operating mode is input capture and compare match. 1: Cycle measurement mode Setting this bit to 1 starts counting by TDPCNT. Clear the CST bit in TDPCR1 to initialize TDPCNT to H'0000 before setting cycle measurement mode.
2	CKS2	0	R/W	Clock Select 2, 1, 0 These bits select the clock signal for input to TDPCNT. Do not select the external clock in level control measurement mode. 000: Counts the ϕ internal clock 001: Counts the $\phi/2$ internal clock 010: Counts the $\phi/4$ internal clock 011: Counts the $\phi/8$ internal clock 100: Counts the $\phi/16$ internal clock 101: Counts the $\phi/32$ internal clock 110: Counts the $\phi/64$ internal clock 111: Counts the external clock (Select the external clock edge with CKSEG in TDPCSR.) Note: Change this bit when CST = 0 and TDPMDS = 0.
1	CKS1	0	R/W	
0	CKS0	0	R/W	

12.3.10 TDP Control Register 2 (TDPCR2)

TDPCR2 selects cycle measurement mode and controls the TDPMCI input polarity.

Bit	Bit Name	Initial Value	R/W	Description
7	PMMS	0	R/W	Cycle Measurement Mode Select Selects whether to use the TDPMCI signal in cycle measurement mode. 0: The TDPMCI signal is not used (cycle measurement is always performed). 1: The TDPMCI signal is used (cycle measurement is performed only while the TDPMCI signal is high). Note: Change this bit when CST = 0 and TDPMDS = 0.
6	MCICTL	0	R/W	TDPMCI Input Polarity Inversion 0: TDPMCI input is used directly 1: TDPMCI input is inverted for use Note: Change this bit when CST = 0 and TDPMDS = 0.
5 to 1	—	0	R/W	Reserved The initial value should not be changed.
0	—	0	R	Reserved This bit is always read as 0 and cannot be modified.

12.3.11 TDP Interrupt Enable Register (TDPIER)

TDPIER enables or disables interrupt requests and controls whether to enable or disable external event input.

Bit	Bit Name	Initial Value	R/W	Description
7	OVIE	0	R/W	Counter Overflow Interrupt Enable Enables or disables the issuing of OVF interrupt requests when the OVF flag in TDPCSR is set to 1. 0: OVF interrupt requests are disabled 1: OVF interrupt requests are enabled
6	TWDMXIE	0	R/W	Pulse Width Upper Limit Overflow Interrupt Enable Enables or disables the issuing of TWDMXOVF interrupt requests when the TWDMXOVF flag in TDPCSR is set to 1. 0: TWDMXOVF interrupt requests are disabled 1: TWDMXOVF interrupt requests are enabled
5	TWDMNIE	0	R/W	Pulse Width Lower Limit Underflow Interrupt Enable Enables or disables the issuing of TWDMNUDF interrupt requests when the TWDMNUDF flag in TDPCSR is set to 1. 0: TWDMNUDF interrupt requests are disabled 1: TWDMNUDF interrupt requests are enabled
4	TPDMXIE	0	R/W	Cycle Upper Limit Overflow Interrupt Enable Enables or disables the issuing of TPDMXOVF interrupt requests when the TPDMXOVF flag in TDPCSR is set to 1. 0: TPDMXOVF interrupt requests are disabled 1: TPDMXOVF interrupt requests are enabled
3	ICPIE	0	R/W	Input Capture Interrupt Enable Enables or disables the issuing of ICPF interrupt requests when the ICPF flag in TDPCSR is set to 1. 0: ICPF interrupt requests are disabled 1: ICPF interrupt requests are enabled
2	CMIE	0	R/W	Compare Match Interrupt Enable Enables or disables the issuing of CMF interrupt requests when the CMF flag in TDPCSR is set to 1. 0: CMF interrupt requests are disabled 1: CMF interrupt requests are enabled

1	TDPIPE	0	R/W	Input Capture Input Enable Enables or disables TDPCYI pin input. To use input capture and cycle measurement mode, set this bit to 1. 0: Disabled 1: Enabled Note: Change this bit when CST = 0 and TDPMDS = 0.
0	TPDMNIE	0	R/W	Cycle Lower Limit Underflow Interrupt Enable Enables or disables the issuing of TPDMNUDF interrupt requests when the TPDMNUDF flag in TDPCSR is set to 1. 0: TPDMNUDF interrupt requests are disabled 1: TPDMNUDF interrupt requests are enabled

12.4 Operation

The TDP operates in timer mode or cycle measurement mode. After a reset, the TDP is in timer mode.

12.4.1 Timer Mode

When the TDPMDS bit in TDPCR1 is cleared to 0, the TDP operates in timer mode.

(1) Counter Operation

The TDP operates as a free-running counter in timer mode. The TDP starts counting up when the CST bit in TDPCR1 is set to 1. When TDPCNT overflows (H'FFFF changes to H'0000), the OVF bit in TDPCSR is set to 1 and an interrupt request is generated if the OVIE bit in TDPIER is 1. Figure 12.2 shows an example of free-running counter operation. In addition, figure 12.3 shows TDPCNT count timing for external clock operation. Note that the external clock requires a pulse width of at least 1.5 cycles. The counter will not operate correctly if the pulses are narrower than this.

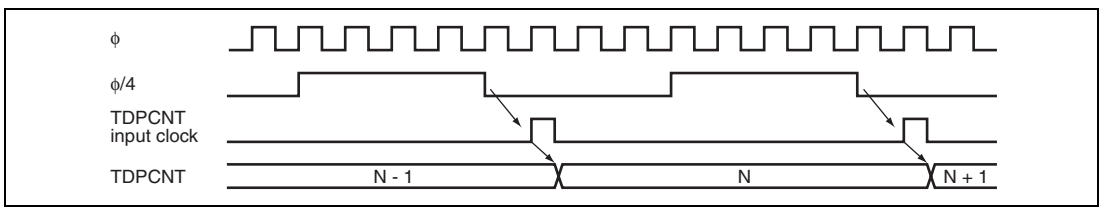


Figure 12.2 Example of Free-Running Counter Operation

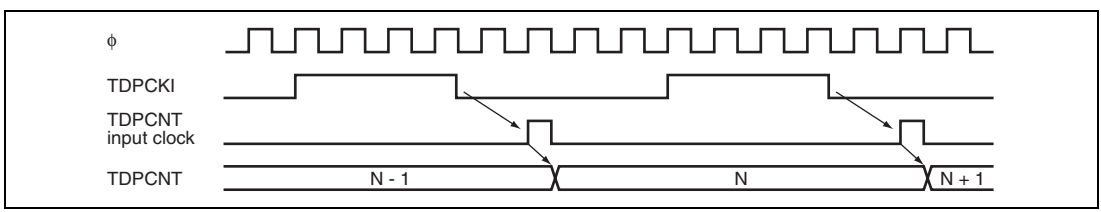


Figure 12.3 Count Timing of External Clock Operation (Falling Edges)

(2) Input Capture

The value in TDPCNT is transferred to TDPICR by detecting the input edge of the TDPCYI pin in timer mode. At the same time, the ICPF flag in TDPCSR is set. Detection of rising or falling edges is selectable by setting the IEDG bit in TDPCR1. Figure 12.4 shows an example of the timing of input capture operations, and figure 12.5 shows an example of buffer operation for input capture.

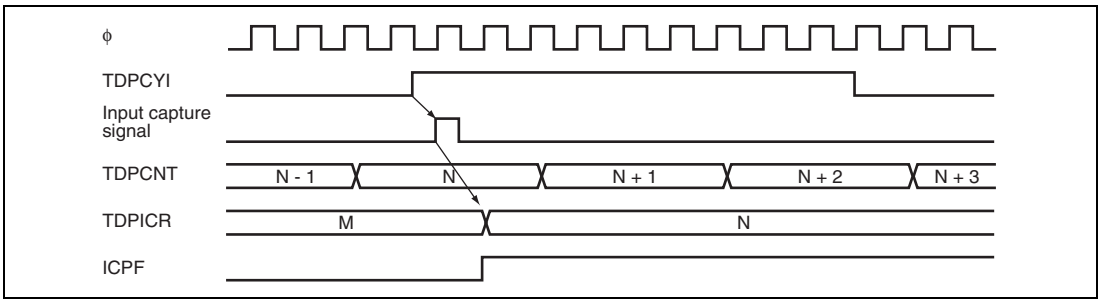


Figure 12.4 Example of Input Capture Operation Timing (Selection of Rising Edges)

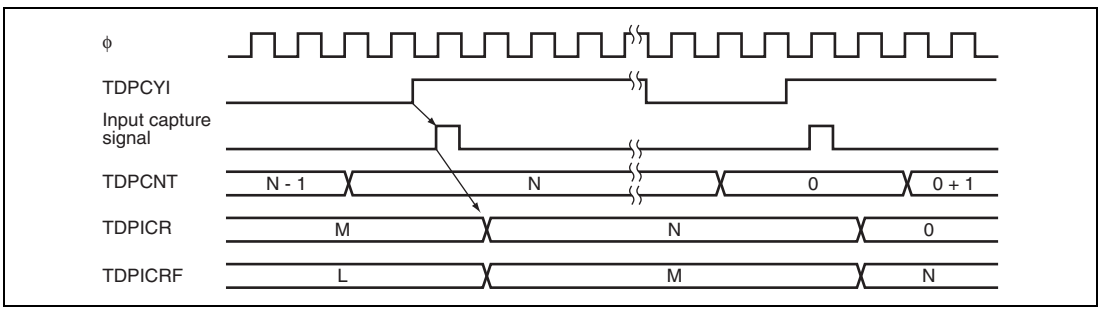


Figure 12.5 Example of Buffer Operation for Input Capture

(3) CMF Setting Timing when a Compare Match Occurs

The CMF flag in TDPCSR is set in the last state in which the values in TDPCNT and TDPWDMX match (timing when TDPCNT updates the matched count value) in timer mode. Accordingly, a compare match signal is not generated until an additional cycle of the TDPCNT input clock is generated after a match between the values in TDPCNT and TDPWDMX. For details, see section 12.6.2, Conflict between TDPPDMX Write and Compare Match. Figure 12.6 shows the timing on which the CMF flag is set.

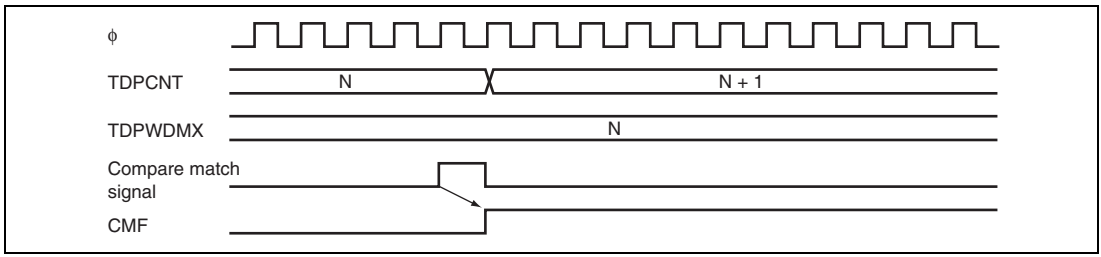


Figure 12.6 Timing of CMF Flag Setting on Compare Match

12.4.2 Cycle Measurement Mode

The TDP operates in cycle measurement mode when the TDPMDS bit in TDPCR1 is set to 1.

(1) Counter Operation

TDPCNT counts up in cycle measurement mode regardless of the setting of the CST bit in TDPCR1. TDPCNT is cleared to H'0000 when the first edge in the measurement period is detected, from which state it counts up. Figure 12.7 shows an example of counter operation in cycle measurement mode.

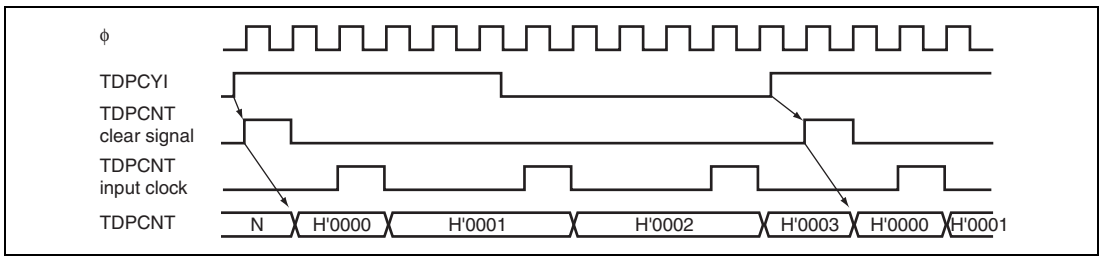


Figure 12.7 Example of Counter Operation in Cycle Measurement Mode

(2) Measuring a Cycle

In cycle measurement mode, one cycle of the TDP input waveform forms one measurement cycle. Start by setting TDPMDS = 0 and CST = 0, which clears TDPCNT to H'0000. Next, set the upper limit and lower limit values of the measurement pulse width in TDPWDMX and TDPWDMN, and set the upper limit and lower limit values of the measurement cycle in the TDPPDMX and TDPPDMN. Finally, place the TDP in cycle measurement mode by setting the TDPMDS bit in TDPCR1 to 1. TDPCNT will count up cycles of the selected clock. When the first edge (either rising or falling, as selected by the POCTL bit in TDPCR1) of the measurement cycle is detected, TDPCNT is automatically cleared to H'0000. When the second edge is detected, the value in TDPCNT is transferred to TDPICR. At this time, the value in TDPICR is compared with the values in TDPWDMX and TDPWDMN. If TDPIR is greater than TDPWDMX or less than TDPWDMN, the TWDMXOVF or TWDMNUDF flag, respectively, in TDPCSR is set to 1. When the third edge is detected, the value in TDPCNT is transferred to TDPICR. At this time, the value in TDPICR is compared with the values in TDPPDMX and TDPPDMN. If TDPICR is greater than TDPPDMX or less than TDPPDMN, the TPDMXOVF or TPDMNUDF flag, respectively, in TDPCSR is set to 1. Generation of the corresponding interrupt request is enabled by the setting in TDPIER. Also, when the third edge is detected, TDPCNT is cleared to H'0000, and the next round of measurement starts.

When the CPSPE bit in TDPCR1 is cleared to 0, the next round of cycle measurement will start regardless of whether any of these flags is set to 1.

If any of these flags is set to 1 while the CPSPE bit in TDPCR1 is set to 1, counting up by TDPCNT stops and cycle measurement also stops. Subsequently clearing the corresponding flag to 0 automatically clears TDPCNT to H'0000, and counting up for cycle measurement is restarted.

Figure 12.8 shows an example of timing in cycle measurement.

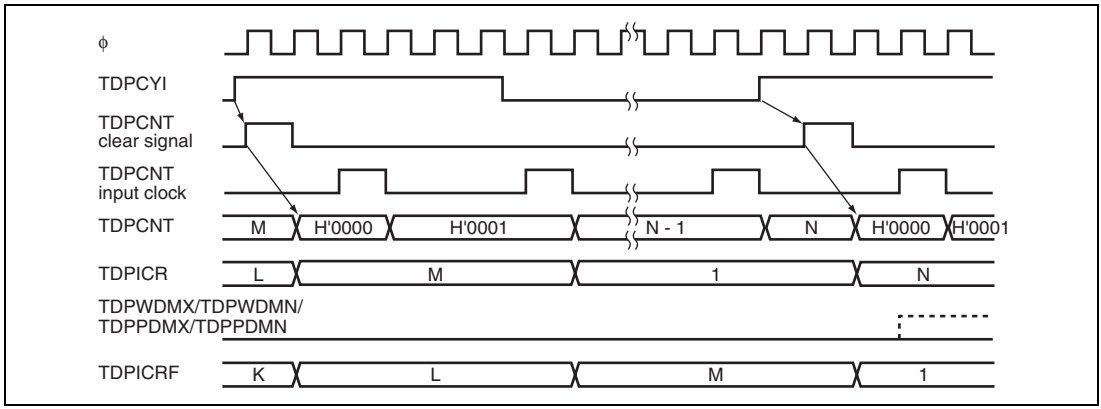


Figure 12.8 Example of Timing in Cycle Measurement

When the PMMS bit in TDPCR2 is set to 1, cycle measurement is performed only while the TDPMCI signal is high. Figure 12.9 shows an example of timing in cycle measurement when the PMMS bit is set to 1.

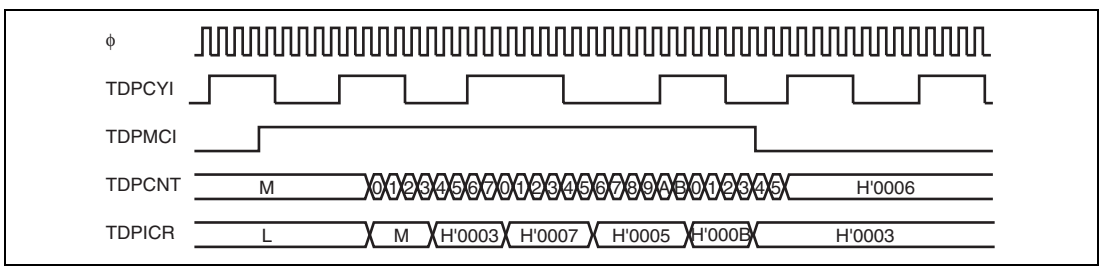


Figure 12.9 Example of Timing in Cycle Measurement (PMMS Bit = 1).

(3) Determination of External Event (TDPCYI) Stoppage

Stoppage for an external event (TDPCYI) can be determined from the timer overflow flag. There are two types of such stoppage.

Stoppage for an external event can be considered to have occurred when the timer overflows within the period from the start of cycle measurement mode to the detection of the first edge (rising or falling, as selected by the POCTL bit in TDPCR1).

Figure 12.10 shows an example of the timing of this type of stoppage for an external event.

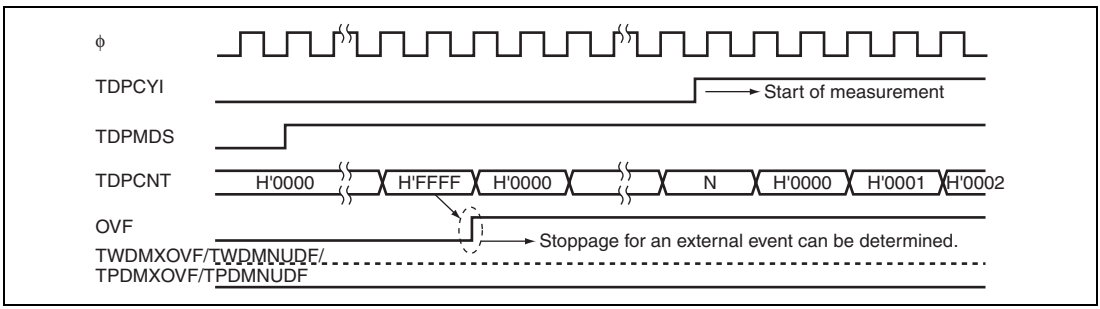


Figure 12.10 Example of Timing for Stoppage for an External Event (1)

When any of the TWDMXOVF, TWDMNUDF, TPDMXOVF, and TPDMNUDF flags is set to 1 while the CPSPE bit in TDPCR1 is 1, cycle measurement stops. Thereafter, when the corresponding flag is cleared, cycle measurement restarts. When the timer overflows before the first edge is detected after the restart of cycle measurement, it is possible to determine stoppage for an external event.

Figure 12.11 shows an example of the timing for this type of stoppage for an external event.

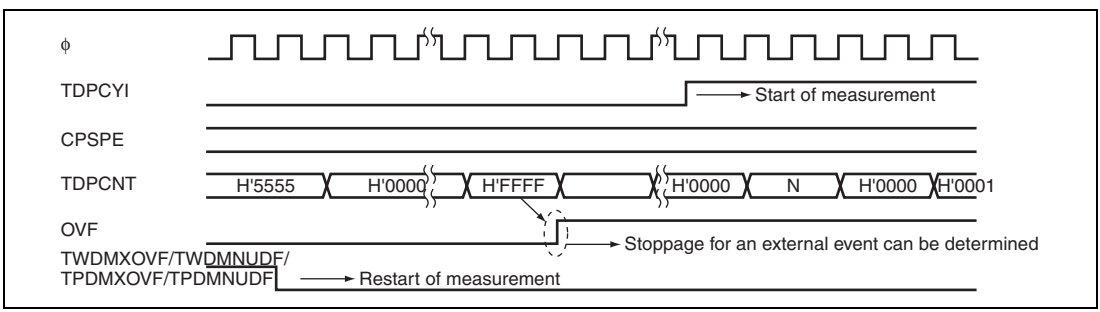


Figure 12.11 Example of Timing for Stoppage for an External Event (2)

(4) Setting Example of Setting Cycle Measurement Mode

Figure 12.12 shows an example of a flowchart for using cycle measurement mode.

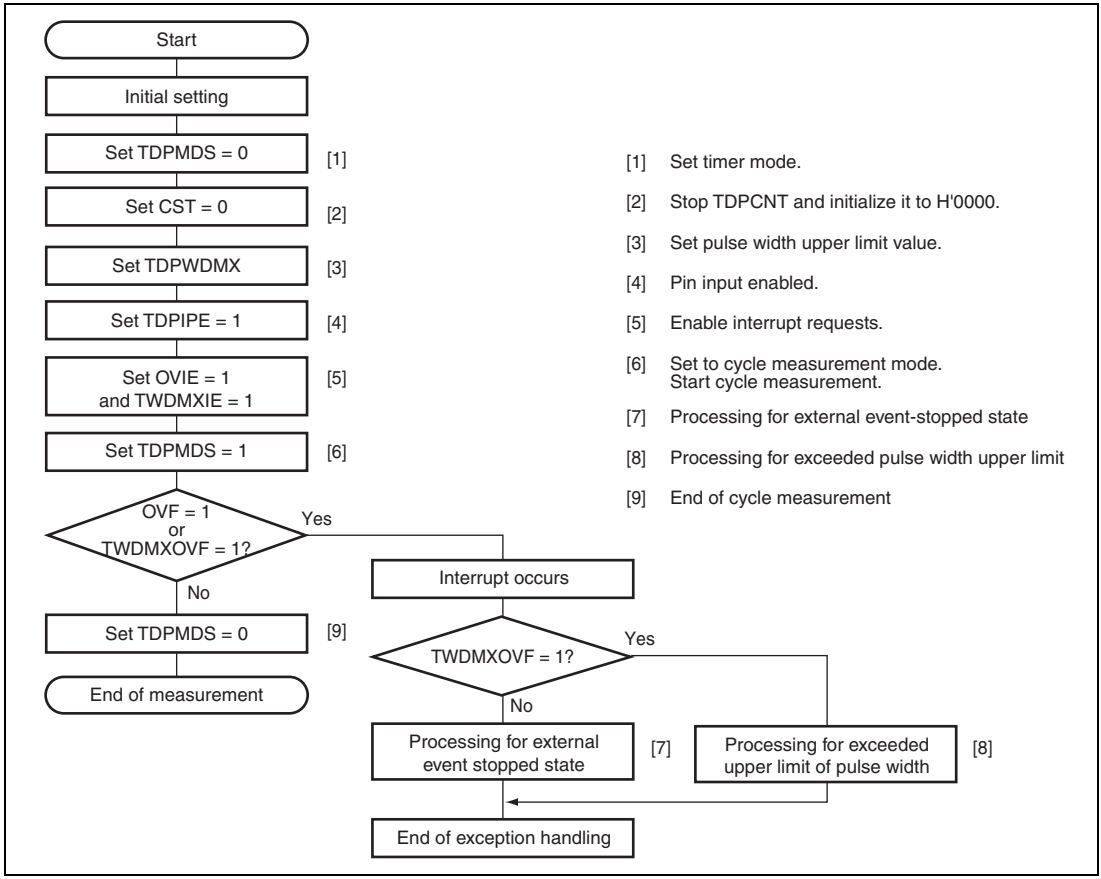


Figure 12.12 Example of Cycle Measurement Mode Settings (for Pulse Width Upper-Limit Value)

12.5 Interrupt Sources

The TDP has seven interrupt sources; TICI, TCMI, TWDMXI, TWDMNI, TPDMXI, TPDNMI, and TOVI. Each interrupt source is enabled or disabled by the corresponding interrupt enable bit in TDPIER and is independently transferred to the interrupt controller. Table 12.3 lists the interrupt sources in order of priority.

Table 12.3 TDP Interrupt Sources

Channel	Name	Interrupt Source	Interrupt Flag	Priority
TDP_0	TICI0	TDPICR_0 input capture	ICPF_0	High
	TCMI0	TDPWDMX_0 compare match	CMF_0	
	TWDMXI0	TDPWDMX_0 overflow	TWDMXOVF_0	
	TWDMNI0	TDPWDMN_0 underflow	TWDMNUDF_0	
	TPDMXI0	TDPPDMX_0 overflow	TPDMXOVF_0	
	TPDMNI0	TDPPDMN_0 underflow	TPDMNUDF_0	
	TOVI0	TDPCNT_0 overflow	OVF_0	
TDP_1	TICI1	TDPICR_1 input capture	ICPF_1	
	TCMI1	TDPWDMX_1 compare match	CMF_1	
	TWDMXI1	TDPWDMX_1 overflow	TWDMXOVF_1	
	TWDMNI1	TDPWDMN_1 underflow	TWDMNUDF_1	
	TPDMXI1	TDPPDMX_1 overflow	TPDMXOVF_1	
	TPDMNI1	TDPPDMN_1 underflow	TPDMNUDF_1	
	TOVI1	TDPCNT_1 overflow	OVF_1	
TDP_2	TICI2	TDPICR_1 input capture	ICPF_2	
	TCMI2	TDPWDMX_2 compare match	CMF_2	
	TWDMXI2	TDPWDMX_2 overflow	TWDMXOVF_2	
	TWDMNI2	TDPWDMN_2 underflow	TWDMNUDF_2	
	TPDMXI2	TDPPDMX_2 overflow	TPDMXOVF_2	
	TPDMNI2	TDPPDMN_2 underflow	TPDMNUDF_2	
	TOVI2	TDPCNT_2 overflow	OVF_2	Low

12.6 Usage Notes

12.6.1 Conflict between TDPCNT Write and Count-Up Operation

If a conflict between a TDPCNT write and counting up operation occurs in the second half of the TDPCNT write cycle, writing to TDPCNT takes precedence and TDPCNT is not incremented. Figure 12.13 shows the timing of this conflict.

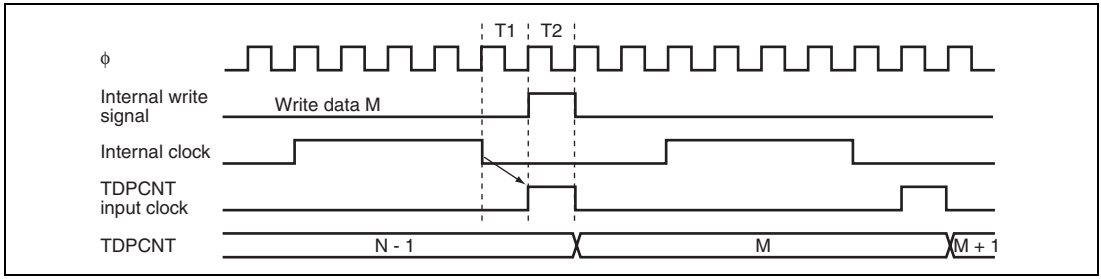


Figure 12.13 Conflict between TDPCNT Write and Counting Up

12.6.2 Conflict between TDPPDMX Write and Compare Match

If a conflict between a TDPPDMX write and a compare match occurs in the second half of the TDPPDMX write cycle, writing to TDPPDMX takes precedence and the compare match signal is inhibited. Figure 12.14 shows the timing of this conflict.

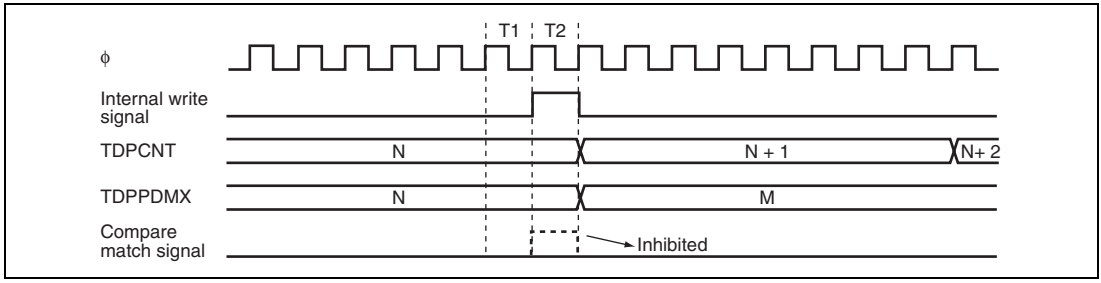


Figure 12.14 Conflict between TDPPDMX Write and Compare Match

12.6.3 Conflict between Input Capture and TDPICR Read

When the corresponding input capture signal is detected during reading of TDPICR in timer mode, the input capture signal is delayed by one cycle of the system clock (ϕ). Figure 12.15 shows the timing of this conflict.

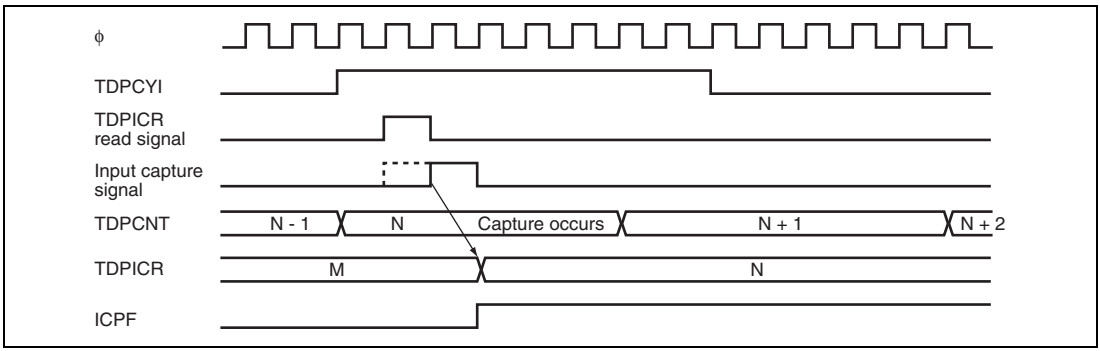


Figure 12.15 Conflict between Input Capture and TDPICR Read

12.6.4 Conflict between Edge Detection in Cycle Measurement Mode and Writing to the Upper Limit or Lower Limit Register

If the edge of TDPCYI is detected in the second half of a cycle of writing to any of the upper limit/lower limit registers (TDPPDMX, TDPPDMN, TDPWDMX, and TDPWDMN) in cycle measurement mode, the detected edge signal is delayed by one cycle of the system clock (ϕ).

Figure 12.16 shows the timing of this conflict.

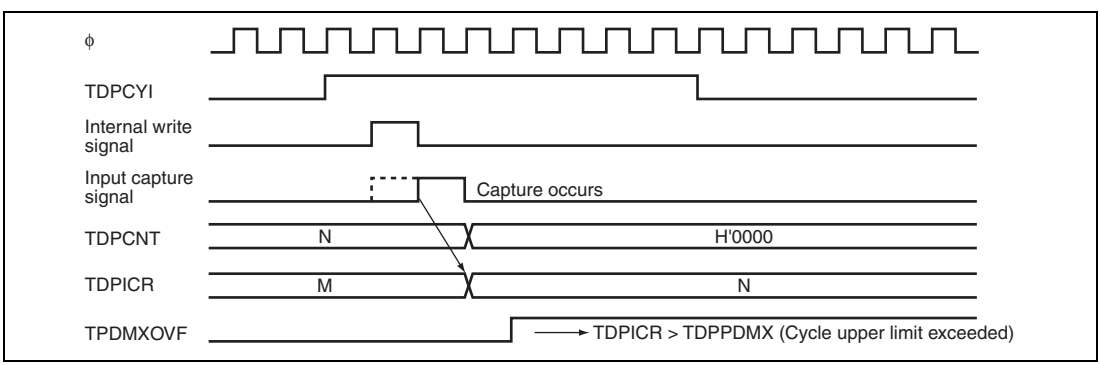


Figure 12.16 Conflict between Edge Detection and Register Write (Cycle Measurement Mode)

12.6.5 Conflict between Edge Detection in Cycle Measurement Mode and TDPMDS Bit Clearing

When the TDPMDS bit in TDPCR1 is cleared in cycle measurement mode while the CST bit in TDPCR1 is 1 and the edge of TDPCYI is detected at the same time, the detected edge signal will cause the timer to continue to operate in cycle measurement mode. The timer enters timer mode when the next edge is detected. Therefore, ensure that the CST bit is cleared to 0 in cycle measurement mode.

Figure 12.17 shows the timing of this conflict.

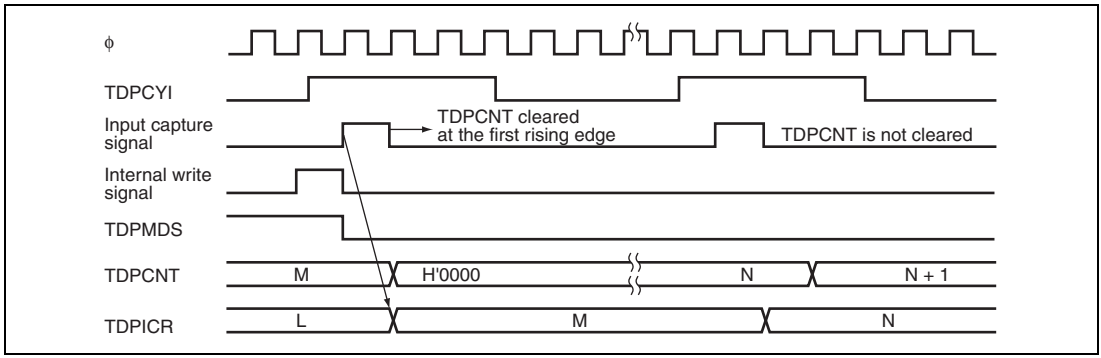


Figure 12.17 Conflict between Edge Detection and TDPMDS Bit Clearing (In Switching from Cycle Measurement Mode to Timer Mode)

12.6.6 Settings for TDPCKI and TDPMCI

TDPCKI and TDPMCI are multiplexed on the same pin of this LSI. Therefore, the selected external clock and the TDPMCI signal cannot be used at the same time. Do not make the settings CKS2 to CKS0 = B'111 and PMMS = B'1.

12.6.7 Setting for Module Stop Mode

The module-stop control register can be used to specify whether to continue or stop TDP operation. The default setting is for the TDP operation to stop. The TDP registers become accessible on release from module stop mode. For details, see section 26, Power-Down Modes.

Section 13 8-Bit Timer (TMR)

This LSI has an on-chip 8-bit timer module (TMR_0, TMR_1, TMR_Y, and TMR_X) with four channels operating on the basis of an 8-bit counter. The 8-bit timer module can be used as a multifunction timer in a variety of applications, such as generation of counter reset, interrupt requests, and pulse output with an arbitrary duty cycle using a compare-match signal with two registers.

13.1 Features

Selection of clock sources

The counter input clock can be selected from six internal clocks and an external clock

- Selection of three ways to clear the counters

  The counters can be cleared on compare-match A, compare-match B, or by an external reset signal.

  Timer output controlled by two compare-match signals

  The timer output signal in each channel is controlled by two independent compare-match signals, enabling the timer to be used for various applications, such as the generation of pulse output or PWM output with an arbitrary duty cycle.
- Cascading of two channels
  - — Cascading of TMR_0 and TMR_1

    Operation as a 16-bit timer can be performed using TMR_0 as the upper half and TMR_1 as the lower half (16-bit count mode).

    TMR_1 can be used to count TMR_0 compare-match occurrences (compare-match count mode).
  - — Cascading of TMR_Y and TMR_X

    Operation as a 16-bit timer can be performed using TMR_Y as the upper half and TMR_X as the lower half (16-bit count mode).

    TMR_X can be used to count TMR_Y compare-match occurrences (compare-match count mode).
- Multiple interrupt sources for each channel

  TMR_0, TMR_1, and TMR_Y: Three types of interrupts: Compare-match A, compare-match B, and overflow

  TMR_X: Four types of interrupts: Compare-match A, compare match B, overflow, and input capture

Figures 13.1 and 13.2 show block diagrams of 8-bit timers.

An input capture function is added to TMR_X.

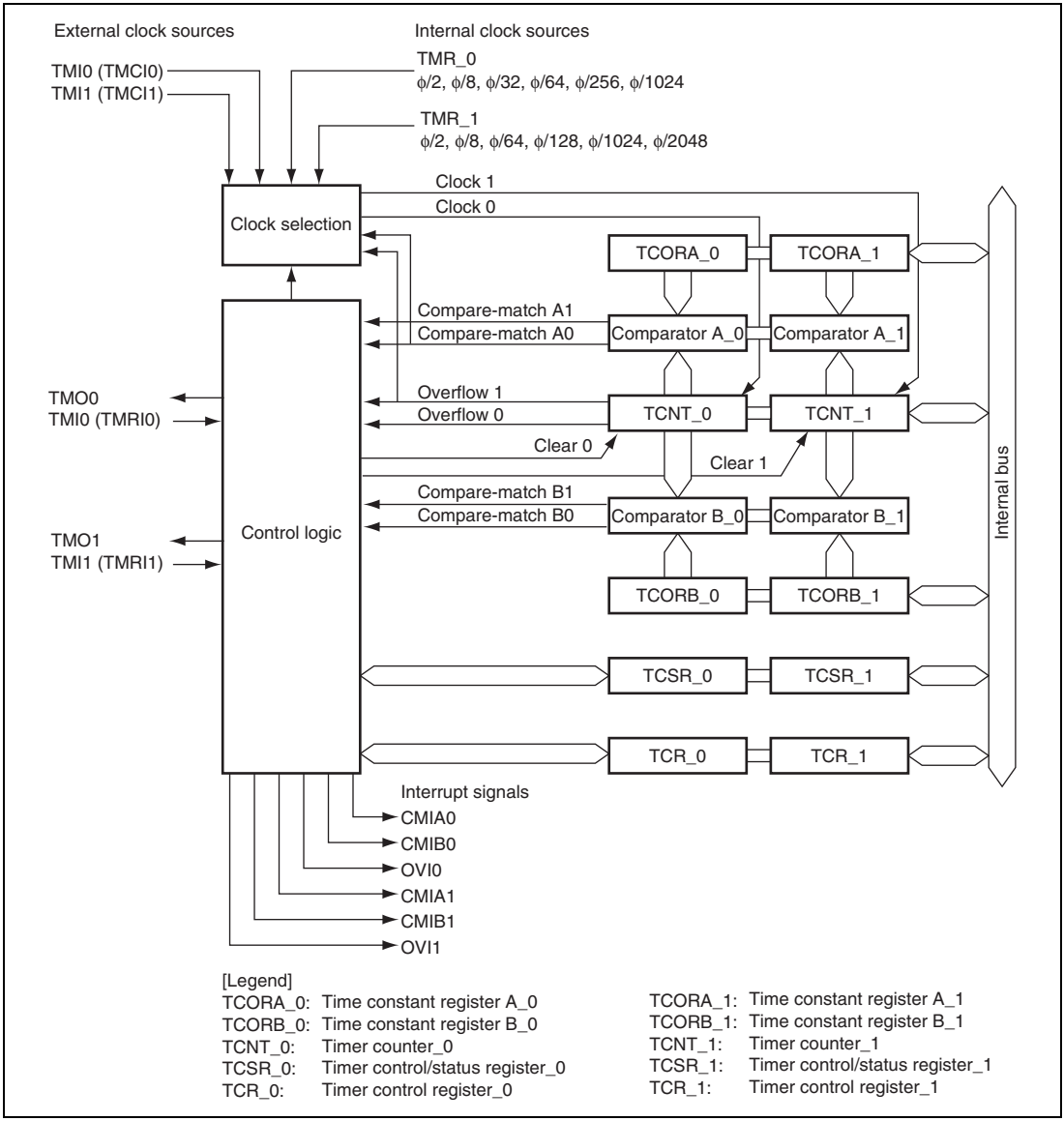
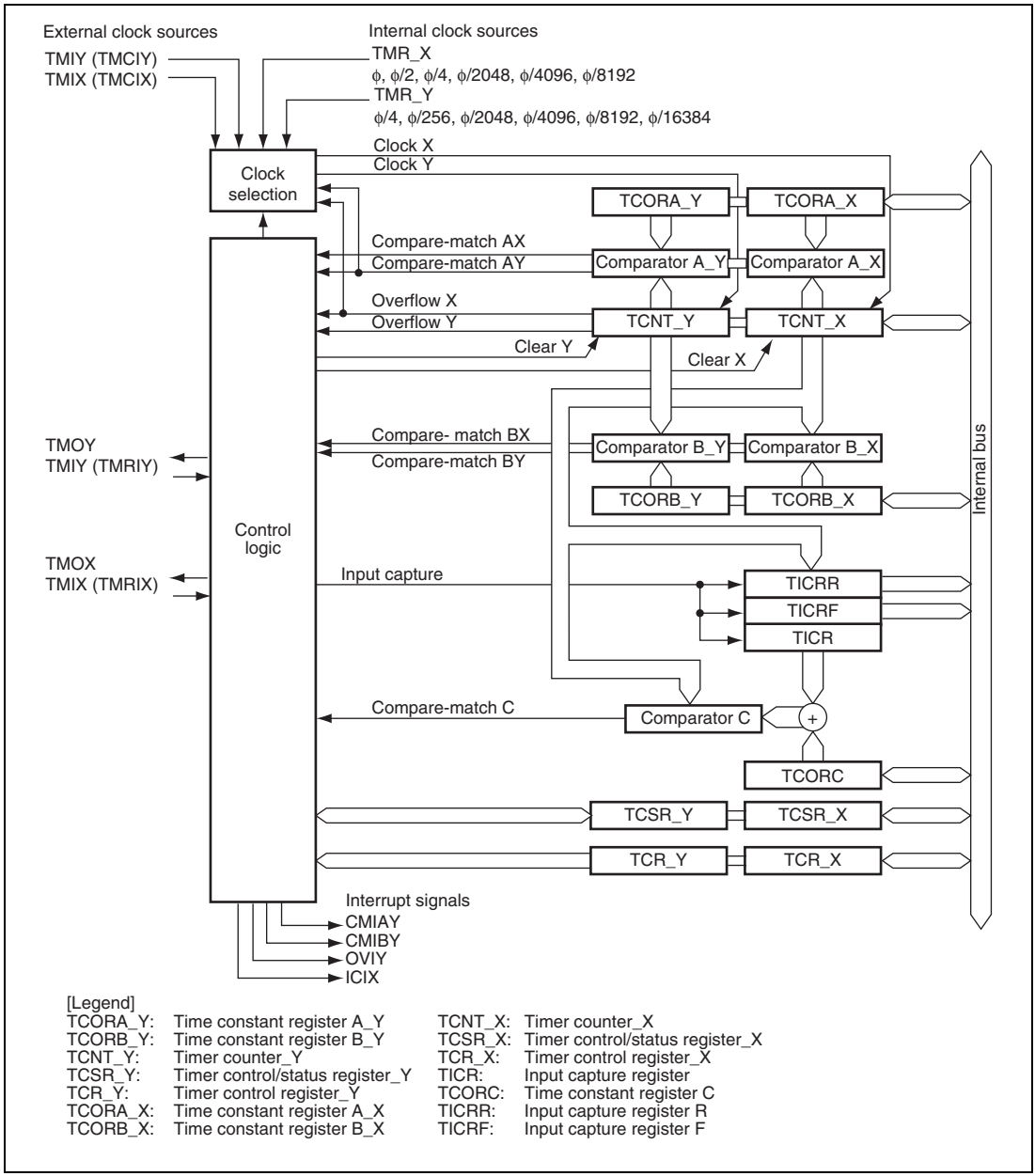


Figure 13.1 Block Diagram of 8-Bit Timer (TMR_0 and TMR_1)



[Legend]

TCORA_Y:	Time constant register A_Y	TCNT_X:	Timer counter_X
TCORB_Y:	Time constant register B_Y	TCSR_X:	Timer control/status register_X
TCNT_Y:	Timer counter_Y	TCR_X:	Timer control register_X
TCSR_Y:	Timer control/status register_Y	TICR:	Input capture register
TCR_Y:	Timer control register_Y	TCORC:	Time constant register C
TCORA_X:	Time constant register A_X	TICRR:	Input capture register R
TCORB_X:	Time constant register B_X	TICRF:	Input capture register F

Figure 13.2 Block Diagram of 8-Bit Timer (TMR_Y and TMR_X)

13.2 Input/Output Pins

Table 13.1 summarizes the input and output pins of the TMR.

Table 13.1 Pin Configuration

Channel	Pin Name	I/O	Function
TMR_0	TMO0	Output	Output controlled by compare-match
	TMI0 (TMCI0/TMRI0)	Input	External clock input/external reset input for the counter
TMR_1	TMO1	Output	Output controlled by compare-match
	TMI1 (TMCI1/TMRI1)	Input	External clock input/external reset input for the counter
TMR_Y	TMIY (TMCIY/TMRIY)	Input	External clock input/external reset input for the counter
	TMOY	Output	Output controlled by compare-match
TMR_X	TMOX	Output	Output controlled by compare-match
	TMIX (TMCIX/TMRIX)	Input	External clock input/external reset input for the counter

13.3 Register Descriptions

The TMR has the following registers. For details on the serial timer control register, see section 3.2.3, Serial Timer Control Register (STCR).

Table 13.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Data Bus Width
Channel 0	Timer counter_0	TCNT_0	R/W	H'00	H'FFD0	16
	Time constant register A_0	TCORA_0	R/W	H'FF	H'FFCC	16
	Time constant register B_0	TCORB_0	R/W	H'FF	H'FFCE	16
	Timer control register_0	TCR_0	R/W	H'00	H'FFC8	8
	Timer control/status register_0	TCSR_0	R/W	H'00	H'FFCA	8
Channel 1	Timer counter_1	TCNT_1	R/W	H'00	H'FFD1	16
	Time constant register A_1	TCORA_1	R/W	H'FF	H'FFCD	16
	Time constant register B_1	TCORB_1	R/W	H'FF	H'FFCF	16
	Timer control register_1	TCR_1	R/W	H'00	H'FFC9	8
	Timer control/status register_1	TCSR_1	R/W	H'10	H'FFCB	8
Channel Y	Timer counter_Y	TCNT_Y	R/W	H'00	H'FFF4 H'FECC*	8
	Time constant register A_Y	TCORA_Y	R/W	H'FF	H'FFF2 H'FECA*	8
	Time constant register B_Y	TCORB_Y	R/W	H'FF	H'FFF3 H'FECB*	8
	Timer control register_Y	TCR_Y	R/W	H'00	H'FFF0 H'FEC8*	8
	Timer control/status register_Y	TCSR_Y	R/W	H'00	H'FFF1 H'FEC9*	8
	Timer connection register S	TCONRS	R/W	H'00	H'FFFE	8

Note: * Upper address: when RELOCATE = 0
Lower address: when RELOCATE = 1

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Data Bus Width
Channel X	Timer counter_X	TCNT_X	R/W	H'00	H'FFF4	8
	Time constant register A_X	TCORA_X	R/W	H'FF	H'FFF6	8
	Time constant register B_X	TCORB_X	R/W	H'FF	H'FFF7	8
	Timer control register_X	TCR_X	R/W	H'00	H'FFF0	8
	Timer control/status register_X	TCSR_X	R/W	H'00	H'FFF1	8
	Time constant register	TCORC	R/W	H'FF	H'FFF5	8
	Input capture register R	TICRR	R	H'00	H'FFF2	8
	Input capture register F	TICRF	R	H'00	H'FFF3	8
	Timer connection register I	TCONRI	R/W	H'00	H'FFFC	8
Common	Timer XY control register	TCRXY	R/W	H'00	H'FEC6	8

Note: Some of the registers of TMR_X and TMR_Y use the same address. The registers can be switched by the TMRX/Y bit in TCONRS.
TCNT_Y, TCORA_Y, TCORB_Y, and TCR_Y can be accessed when the RELOCATE bit in SYSCR3 and the KINWUE bit in SYSCR are cleared to 0 and the TMRX/Y bit in TCONRS is set to 1, or when the RELOCATE bit in SYSCR3 is set to 1. TCNT_X, TCORA_X, TCORB_X, and TCR_X can be accessed when the RELOCATE bit in SYSCR3, the KINWUE bit in SYSCR, and the TMRX/Y bit in TCONRS are cleared to 0, or when the RELOCATE bit in SYSCR3 is set to 1.

13.3.1 Timer Counter (TCNT)

Each TCNT is an 8-bit readable/writable up-counter. TCNT_0 and TCNT_1 (or TCNT_X and TCNT_Y) comprise a single 16-bit register, so they can be accessed together by word access. The clock source is selected by the CKS2 to CKS0 bits in TCR. TCNT can be cleared by an external reset input signal, compare-match A signal or compare-match B signal. The method of clearing can be selected by the CCLR1 and CCLR0 bits in TCR. When TCNT overflows (changes from H'FF to H'00), the OVF bit in TCSR is set to 1. TCNT is initialized to H'00.

13.3.2 Time Constant Register A (TCORA)

TCORA is an 8-bit readable/writable register. TCORA_0 and TCORA_1 (or TCORA_X and TCORA_Y) comprise a single 16-bit register, so they can be accessed together by word access. TCORA is continually compared with the value in TCNT. When a match is detected, the corresponding compare-match flag A (CMFA) in TCSR is set to 1. Note however that comparison is disabled during the T_2 state of a TCORA write cycle. The timer output from the TMO pin can be freely controlled by these compare-match A signals and the settings of output select bits OS1 and OS0 in TCSR. TCORA is initialized to H'FF.

13.3.3 Time Constant Register B (TCORB)

TCORB is an 8-bit readable/writable register. TCORB_0 and TCORB_1 (or TCORB_X and TCORB_Y) comprise a single 16-bit register, so they can be accessed together by word access. TCORB is continually compared with the value in TCNT. When a match is detected, the corresponding compare-match flag B (CMFB) in TCSR is set to 1. Note however that comparison is disabled during the T_2 state of a TCORB write cycle. The timer output from the TMO pin can be freely controlled by these compare-match B signals and the settings of output select bits OS3 and OS2 in TCSR. TCORB is initialized to H'FF.

13.3.4 Timer Control Register (TCR)

TCR selects the TCNT clock source and the condition by which TCNT is cleared, and enables/disables interrupt requests.

Bit	Bit Name	Initial Value	R/W	Description
7	CMIEB	0	R/W	Compare-Match Interrupt Enable B Selects whether the CMFB interrupt request (CMIB) is enabled or disabled when the CMFB flag in TCSR is set to 1. 0: CMFB interrupt request (CMIB) is disabled 1: CMFB interrupt request (CMIB) is enabled
6	CMIEA	0	R/W	Compare-Match Interrupt Enable A Selects whether the CMFA interrupt request (CMIA) is enabled or disabled when the CMFA flag in TCSR is set to 1. 0: CMFA interrupt request (CMIA) is disabled 1: CMFA interrupt request (CMIA) is enabled
5	OVIE	0	R/W	Timer Overflow Interrupt Enable Selects whether the OVF interrupt request (OVI) is enabled 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	These bits select the method by which the timer counter is cleared. 00: Clearing is disabled 01: Cleared on compare-match A 10: Cleared on compare-match B 11: Cleared on rising edge of external reset input
2	CKS2	0	R/W	Clock Select 2 to 0
1	CKS1	0	R/W	These bits select the clock input to TCNT and count condition, together with the ICKS1 and ICKS0 bits in STCR. For details, see table 13.3.
0	CKS0	0	R/W	

Table 13.3 Clock Input to TCNT and Count Condition (1)

	TCR			STCR		
Channel	**CKS2**	**CKS1**	**CKS0**	**ICKS1**	**ICKS0**	**Description**
TMR_0	0	0	0	—	—	Disables clock input
	0	0	1	—	0	Increments at falling edge of internal clock ϕ/8
	0	0	1	—	1	Increments at falling edge of internal clock ϕ/2
	0	1	0	—	0	Increments at falling edge of internal clock ϕ/64
	0	1	0	—	1	Increments at falling edge of internal clock ϕ/32
	0	1	1	—	0	Increments at falling edge of internal clock ϕ/1024
	0	1	1	—	1	Increments at falling edge of internal clock ϕ/256
	1	0	0	—	—	Increments at overflow signal from TCNT_1*
TMR_1	0	0	0	—	—	Disables clock input
	0	0	1	0	—	Increments at falling edge of internal clock ϕ/8
	0	0	1	1	—	Increments at falling edge of internal clock ϕ/2
	0	1	0	0	—	Increments at falling edge of internal clock ϕ/64
	0	1	0	1	—	Increments at falling edge of internal clock ϕ/128
	0	1	1	0	—	Increments at falling edge of internal clock ϕ/1024
	0	1	1	1	—	Increments at falling edge of internal clock ϕ/2048
	1	0	0	—	—	Increments at compare-match A from TCNT_0*

	TCR			STCR		
Channel	**CKS2**	**CKS1**	**CKS0**	**ICKS1**	**ICKS0**	**Description**
Common	1	0	1	—	—	Increments at rising edge of external clock
	1	1	0	—	—	Increments at falling edge of external clock
	1	1	1	—	—	Increments at both rising and falling edges of external clock

Note: * If the TMR_0 clock input is set as the TCNT_1 overflow signal and the TMR_1 clock input is set as the TCNT_0 compare-match signal simultaneously, a count-up clock cannot be generated. These settings should not be made.

Table 13.3

Clock Input to TCNT and Count Condition (2)

	TCR			TCRXY		
Channel	**CKS2**	**CKS1**	**CKS0**	**CKSX**	**CKSY**	**Description**
TMR_Y	0	0	0	—	0	Disables clock input
	0	0	1	—	0	Increments at ϕ/4
	0	1	0	—	0	Increments at ϕ/256
	0	1	1	—	0	Increments at ϕ/2048
	1	0	0	—	0	Disables clock input
	0	0	0	—	1	Disables clock input
	0	0	1	—	1	Increments at ϕ/4096
	0	1	0	—	1	Increments at ϕ/8192
	0	1	1	—	1	Increments at ϕ/16384
	1	0	0	—	1	Increments at overflow signal from TCNT_X*
	1	0	1	—	x	Increments at rising edge of external clock
	1	1	0	—	x	Increments at falling edge of external clock
	1	1	1	—	x	Increments at both rising and falling edges of external clock

	TCR			TCRXY		
Channel	**CKS2**	**CKS1**	**CKS0**	**CKSX**	**CKSY**	**Description**
TMR_X	0	0	0	0	—	Disables clock input
	0	0	1	0	—	Increments at ϕ
	0	1	0	0	—	Increments at $\phi/2$
	0	1	1	0	—	Increments at $\phi/4$
	1	0	0	0	—	Disables clock input
	0	0	0	1	—	Disables clock input
	0	0	1	1	—	Increments at $\phi/2048$
	0	1	0	1	—	Increments at $\phi/4096$
	0	1	1	1	—	Increments at $\phi/8192$
	1	0	0	1	—	Increments at compare-match A from TCNT_Y*
	1	0	1	x	—	Increments at rising edge of external clock
	1	1	0	x	—	Increments at falling edge of external clock
	1	1	1	x	—	Increments at both rising and falling edges of external clock

Note: * If the TMR_Y clock input is set as the TCNT_X overflow signal and the TMR_X clock input is set as the TCNT_Y compare-match signal simultaneously, a count-up clock cannot be generated. These settings should not be made.

[Legend]

x: Don't care

—: Invalid

13.3.5 Timer Control/Status Register (TCSR)

TCSR indicates the status flags and controls compare-match output.

- TCSR_0

Bit	Bit Name	Initial Value	R/W	Description
7	CMFB	0	R/(W)*	Compare-Match Flag B [Setting condition] When the values of TCNT_0 and TCORB_0 match [Clearing condition] Read CMFB when CMFB = 1, then write 0 in CMFB
6	CMFA	0	R/(W)*	Compare-Match Flag A [Setting condition] When the values of TCNT_0 and TCORA_0 match [Clearing condition] Read CMFA when CMFA = 1, then write 0 in CMFA
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 Enables or disables A/D converter start requests by compare-match A. 0: A/D converter start requests by compare-match A are disabled 1: A/D converter start requests by compare-match A are enabled
3	OS3	0	R/W	Output Select 3 and 2
2	OS2	0	R/W	These bits specify how the TMO0 pin output level is to be changed by compare-match B of TCORB_0 and TCNT_0. 00: No change 01: 0 is output 10: 1 is output 11: Output is inverted (toggle output)

Bit	Bit Name	Initial Value	R/W	Description
1	OS1	0	R/W	Output Select 1 and 0
0	OS0	0	R/W	These bits specify how the TMO0 pin output level is to be changed by compare-match A of TCORA_0 and TCNT_0. 00: No change 01: 0 is output 10: 1 is output 11: Output is inverted (toggle output)

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

- TCSR_1

Bit	Bit Name	Initial Value	R/W	Description
7	CMFB	0	R/(W)*	Compare-Match Flag B [Setting condition] When the values of TCNT_1 and TCORB_1 match [Clearing condition] Read CMFB when CMFB = 1, then write 0 in CMFB
6	CMFA	0	R/(W)*	Compare-Match Flag A [Setting condition] When the values of TCNT_1 and TCORA_1 match [Clearing condition] Read CMFA when CMFA = 1, then write 0 in CMFA
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	—	1	R	Reserved This bit is always read as 1 and cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
3	OS3	0	R/W	Output Select 3 and 2
2	OS2	0	R/W	These bits specify how the TMO1 pin output level is to be changed by compare-match B of TCORB_1 and TCNT_1. 00: No change 01: 0 is output 10: 1 is output 11: Output is inverted (toggle output)
1	OS1	0	R/W	Output Select 1 and 0
0	OS0	0	R/W	These bits specify how the TMO1 pin output level is to be changed by compare-match A of TCORA_1 and TCNT_1. 00: No change 01: 0 is output 10: 1 is output 11: Output is inverted (toggle output)

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

- TCSR_X

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

Bit	Bit Name	Initial Value	R/W	Description
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	ICF	0	R/(W)*	Input Capture Flag [Setting condition] When a rising edge and falling edge is detected in the external reset signal in that order. [Clearing condition] Read ICF when ICF = 1, then write 0 in ICF
3	OS3	0	R/W	Output Select 3 and 2 These bits specify how the TMOX pin output level is to be changed by compare-match B of TCORB_X and TCNT_X. 00: No change 01: 0 is output 10: 1 is output 11: Output is inverted (toggle output)
2	OS2	0	R/W	
1	OS1	0	R/W	Output Select 1 and 0 These bits specify how the TMOX pin output level is to be changed by compare-match A of TCORA_X and TCNT_X. 00: No change 01: 0 is output 10: 1 is output 11: Output is inverted (toggle output)
0	OS0	0	R/W	

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

- TCSR_Y

Bit	Bit Name	Initial Value	R/W	Description
7	CMFB	0	R/(W)*	Compare-Match Flag B [Setting condition] When the values of TCNT_Y and TCORB_Y match [Clearing condition] Read CMFB when CMFB = 1, then write 0 in CMFB
6	CMFA	0	R/(W)*	Compare-Match Flag A [Setting condition] When the values of TCNT_Y and TCORA_Y match [Clearing condition] Read CMFA when CMFA = 1, then write 0 in CMFA
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	ICIE	0	R/W	Input Capture Interrupt Enable Enables or disables the ICF interrupt request (ICIX) when the ICF bit in TCSR_X is set to 1. 0: ICF interrupt request (ICIX) is disabled 1: ICF interrupt request (ICIX) is enabled
3	OS3	0	R/W	Output Select 3 and 2
2	OS2	0	R/W	These bits specify how the TMOY pin output level is to be changed by compare-match B of TCORB_Y and TCNT_Y. 00: No change 01: 0 is output 10: 1 is output 11: Output is inverted (toggle output)

Bit	Bit Name	Initial Value	R/W	Description
1	OS1	0	R/W	Output Select 1 and 0
0	OS0	0	R/W	These bits specify how the TMOY pin output level is to be changed by compare-match A of TCORA_Y and TCNT_Y. 00: No change 01: 0 is output 10: 1 is output 11: Output is inverted (toggle output)

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

13.3.6 Time Constant Register C (TCORC)

TCORC is an 8-bit readable/writable register. The sum of contents of TCORC and TICR is always compared with TCNT. When a match is detected, a compare-match C signal is generated. However, comparison at the T_2 state in the write cycle to TCORC and at the input capture cycle of TICR is disabled. TCORC is initialized to H'FF.

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

TICRR and TICRF are 8-bit read-only registers. While the ICST bit in TCONRI is set to 1, the contents of TCNT are transferred at the rising edge and falling edge of the external reset input (TMRIX) in that order. The ICST bit is cleared to 0 when one capture operation ends. TICRR and TICRF are initialized to H'00.

13.3.8 Timer Connection Register I (TCONRI)

TCONRI controls the input capture function.

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

13.3.9 Timer Connection Register S (TCONRS)

TCONRS selects whether to access TMR_X or TMR_Y registers.

Bit	Bit Name	Initial Value	R/W	Description
7	TMRX/Y	0	R/W	TMR_X/TMR_Y Access Select For details, see table 13.4. 0: The TMR_X registers are accessed at addresses H'(FF)FFF0 to H'(FF)FFF5 1: The TMR_Y registers are accessed at addresses H'(FF)FFF0 to H'(FF)FFF5
6 to 0	—	All 0	R/W	Reserved The initial values should not be modified.

Table 13.4 Registers Accessible by TMR_X/TMR_Y

TMRX/Y	H'FFF0	H'FFF1	H'FFF2	H'FFF3	H'FFF4	H'FFF5	H'FFF6	H'FFF7
0	TMR_X	TMR_X	TMR_X	TMR_X	TMR_X	TMR_X	TMR_X	TMR_X
	TCR_X	TCSR_X	TICRR	TICRF	TCNT	TCORC	TCORA_X	TCORB_X
1	TMR_Y	TMR_Y	TMR_Y	TMR_Y	TMR_Y	TMR_Y		
	TCR_Y	TCSR_Y	TCORA_Y	TCORB_Y	TCNT_Y			

13.3.10 Timer XY Control Register (TCRXY)

TCRXY selects the TMR_X and TMR_Y output pins and internal clock.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R/W	Reserved The initial value should not be changed.
5	CKSX	0	R/W	TMR_X Clock Select For details about selection, see table 13.3.
4	CKSY	0	R/W	TMR_Y Clock Select For details about selection, see table 13.3.
3 to 0	—	All 0	R/W	Reserved The initial value should not be changed.

13.4 Operation

13.4.1 Pulse Output

Figure 13.3 shows an example for outputting an arbitrary duty pulse.

1. Clear the CCLR1 bit in TCR to 0, and set the CCLR0 bit in TCR to 1 so that TCNT is cleared according to the compare match of TCORA.
2. Set the OS3 to OS0 bits in TCSR to B'0110 so that 1 is output according to the compare match of TCORA and 0 is output according to the compare match of TCORB.

According to the above settings, the waveforms with the TCORA cycle and TCORB pulse width can be output without the intervention of software.

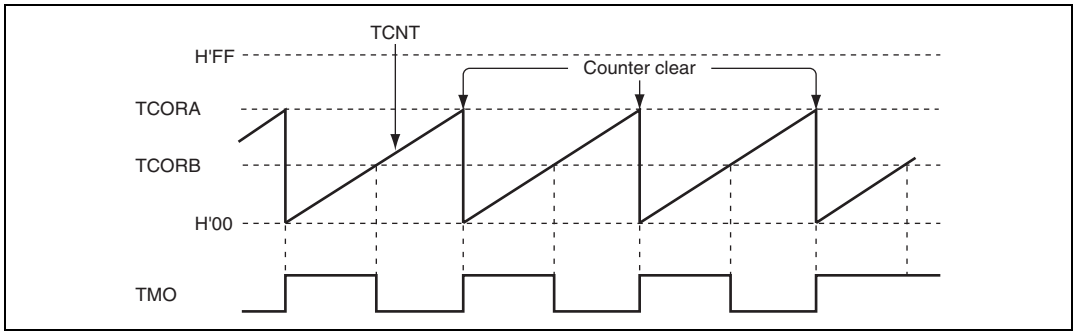


Figure 13.3 Pulse Output Example

13.5 Operation Timing

13.5.1 TCNT Count Timing

Figure 13.4 shows the TCNT count timing with an internal clock source. Figure 13.5 shows the TCNT count timing with an external clock source. The pulse width of the external clock signal must be at least 1.5 system clocks (ϕ) for a single edge and at least 2.5 system clocks (ϕ) for both edges. The counter will not increment correctly if the pulse width is less than these values.

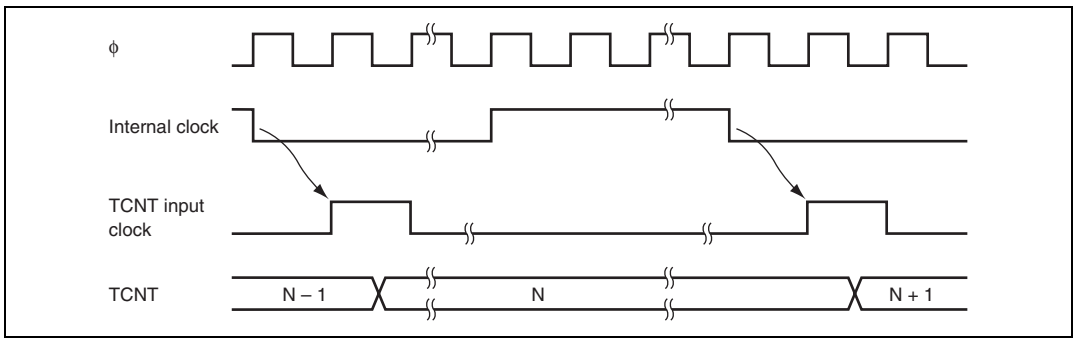


Figure 13.4 Count Timing for Internal Clock Input

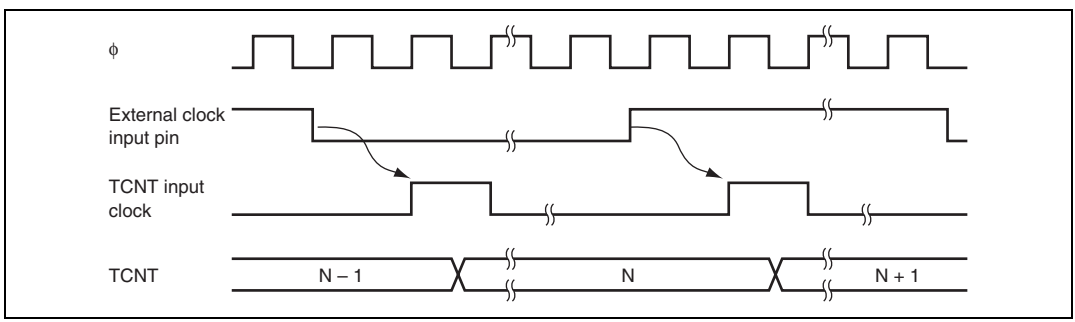


Figure 13.5 Count Timing for External Clock Input (Both Edges)

13.5.2 Timing of CMFA and CMFB Setting at Compare-Match

The CMFA and CMFB flags in TCSR are set to 1 by a compare-match signal generated when the TCNT and TCOR values match. The compare-match signal is generated at the last state in which the match is true, just when the timer counter is updated. Therefore, when TCNT and TCOR match, the compare-match signal is not generated until the next TCNT input clock. Figure 13.6 shows the timing of CMF flag setting.

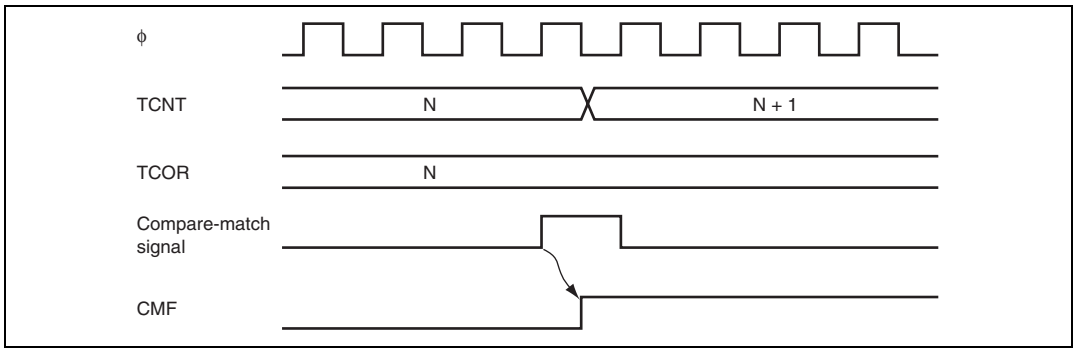


Figure 13.6 Timing of CMF Setting at Compare-Match

13.5.3 Timing of Timer Output at Compare-Match

When a compare-match signal occurs, the timer output changes as specified by the OS3 to OS0 bits in TCSR. Figure 13.7 shows the timing of timer output when the output is set to toggle by a compare-match A signal.

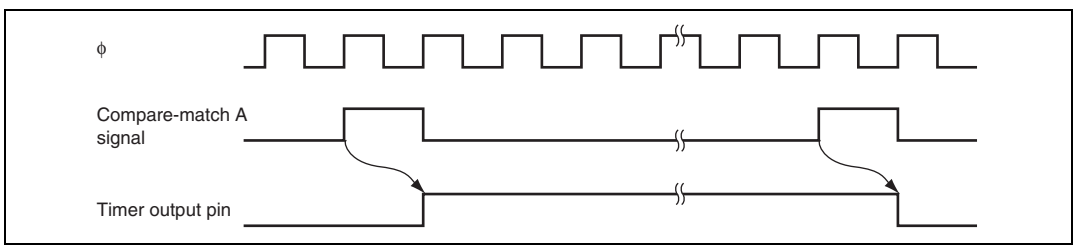


Figure 13.7 Timing of Toggled Timer Output by Compare-Match A Signal

13.5.4 Timing of Counter Clear at Compare-Match

TCNT is cleared when compare-match A or compare-match B occurs, depending on the setting of the CCLR1 and CCLR0 bits in TCR. Figure 13.8 shows the timing of clearing the counter by a compare-match.

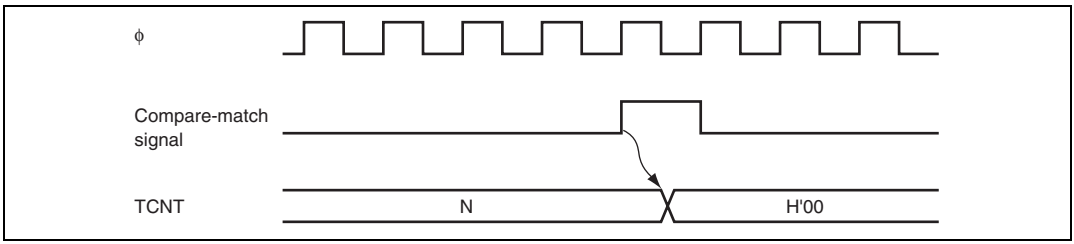


Figure 13.8 Timing of Counter Clear by Compare-Match

13.5.5 TCNT External Reset Timing

TCNT is cleared at the rising edge of an external reset input, depending on the settings of the CCLR1 and CCLR0 bits in TCR. The width of the clearing pulse must be at least 1.5 states. Figure 13.9 shows the timing of clearing the counter by an external reset input.

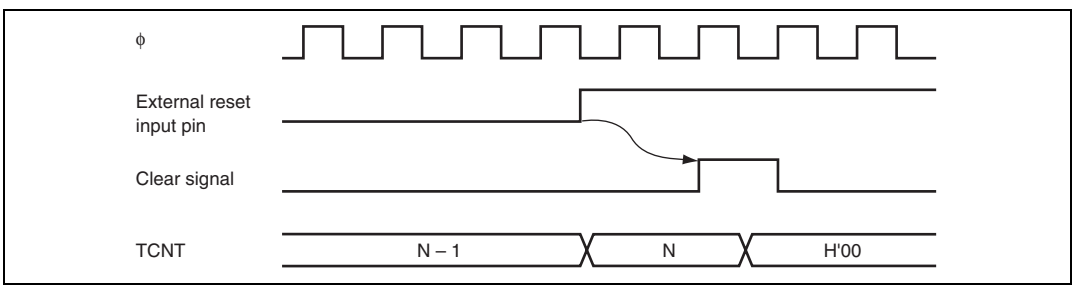


Figure 13.9 Timing of Counter Clear by External Reset Input

13.5.6 Timing of Overflow Flag (OVF) Setting

The OVF bit in TCSR is set to 1 when the TCNT overflows (changes from H'FF to H'00). Figure 13.10 shows the timing of OVF flag setting.

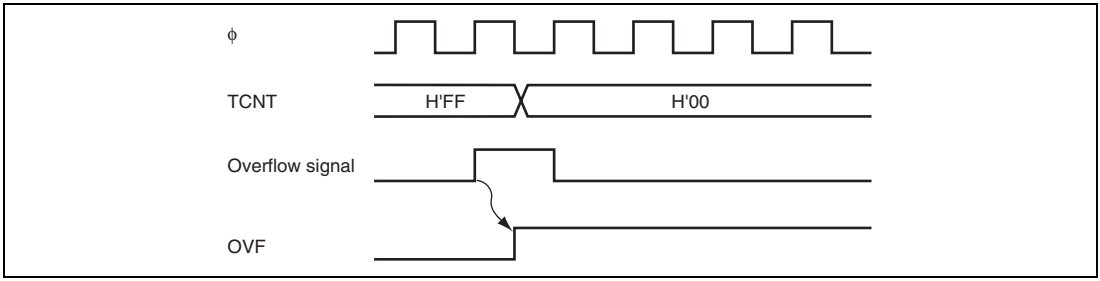


Figure 13.10 Timing of OVF Flag Setting

13.6 TMR_0 and TMR_1 Cascaded Connection

If bits CKS2 to CKS0 in either TCR_0 or TCR_1 are set to B'100, the 8-bit timers of the two channels are cascaded. With this configuration, the 16-bit count mode or compare-match count mode is available.

13.6.1 16-Bit Count Mode

When bits CKS2 to CKS0 in TCR_0 are set to B'100, the timer functions as a single 16-bit timer with TMR_0 occupying the upper 8 bits and TMR_1 occupying the lower 8 bits.

- 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 compare-match, the 16-bit counter (TCNT_0 and TCNT_1 together) is cleared when a 16-bit compare-match occurs. The 16-bit counter (TCNT_0 and TCNT_1 together) is also cleared when 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 bits cannot be cleared independently.
- Pin output
  - Control of output from the TMO0 pin by bits OS3 to OS0 in TCSR_0 is in accordance with the 16-bit compare-match conditions.
  - Control of output from the TMO1 pin by bits OS3 to OS0 in TCSR_1 is in accordance with the lower 8-bit compare-match conditions.

13.6.2 Compare-Match Count Mode

When bits CKS2 to CKS0 in TCR_1 are B'100, TCNT_1 counts the occurrence of compare-match A for TMR_0. TMR_0 and TMR_1 are controlled independently. Conditions such as setting of the CMF flag, generation of interrupts, output from the TMO pin, and counter clearing are in accordance with the settings for each or TMR_0 and TMR_1.

13.7 TMR_Y and TMR_X Cascaded Connection

If bits CKS2 to CKS0 in either TCR_Y or TCR_X are set to B'100, the 8-bit timers of the two channels are cascaded. With this configuration, 16-bit count mode or compare-match count mode can be selected by the settings of the CKSX and CKSY bits in TCRXY.

13.7.1 16-Bit Count Mode

When bits CKS2 to CKS0 in TCR_Y are set to B'100 and the CKSY bit in TCRXY is set to 1, the timer functions as a single 16-bit timer with TMR_Y occupying the upper eight bits and TMR_X occupying the lower 8 bits.

- Setting of compare-match flags
  - The CMF flag in TCSR_Y is set to 1 when an upper 8-bit compare-match occurs.
  - The CMF flag in TCSR_X is set to 1 when a lower 8-bit compare-match occurs.
- Counter clear specification
  - If the CCLR1 and CCLR0 bits in TCR_Y have been set for counter clear at compare-match, only the upper eight bits of TCNT_Y are cleared. The upper eight bits of TCNT_Y are also cleared when counter clear by the TMRIY pin has been set.
  - The settings of the CCLR1 and CCLR0 bits in TCR_X are enabled, and the lower 8 bits of TCNT_X can be cleared by the counter.
- Pin output
  - Control of output from the TMOY pin by bits OS3 to OS0 in TCSR_Y is in accordance with the upper 8-bit compare-match conditions.
  - Control of output from the TMOX pin by bits OS3 to OS0 in TCSR_X is in accordance with the lower 8-bit compare-match conditions.

13.7.2 Compare-Match Count Mode

When bits CKS2 to CKS0 in TCR_X are set to B'100 and the CKSX bit in TCRXY is set to 1, TCNT_X counts the occurrence of compare-match A for TMR_Y. TMR_X and TMR_Y are controlled independently. Conditions such as setting of the CMF flag, generation of interrupts, output from the TMO pin, and counter clearing are in accordance with the settings for each channel.

13.7.3 Input Capture Operation

TMR_X has input capture registers (TICRR and TICRF). A narrow pulse width can be measured with TICRR and TICRF, using a single capture. If the falling edge of TMRIX (TMR_X input capture input signal) is detected after its rising edge has been detected, the value of TCNT_X at that time is transferred to both TICRR and TICRF.

(1) Input Capture Signal Input Timing

Figure 13.11 shows the timing of the input capture operation.

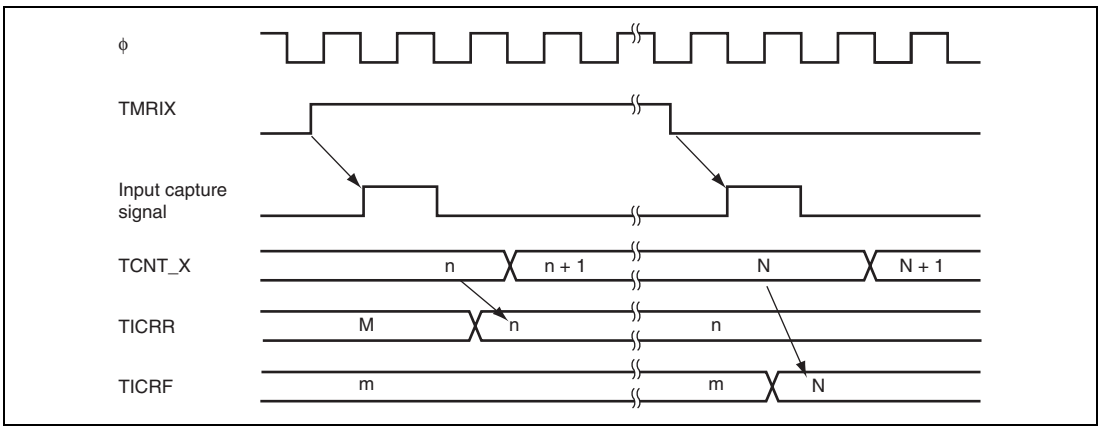


Figure 13.11 Timing of Input Capture Operation

If the input capture signal is input while TICRR and TICRF are being read, the input capture signal is delayed by one system clock (ϕ) cycle. Figure 13.12 shows the timing of this operation.

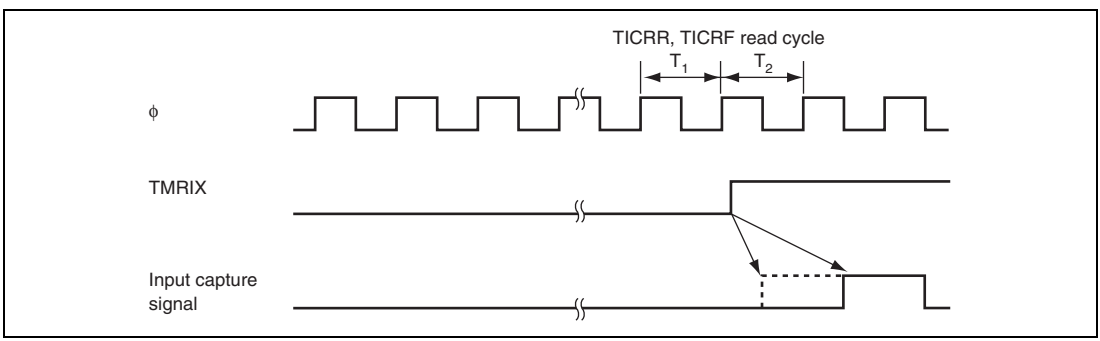


Figure 13.12 Timing of Input Capture Signal (Input capture signal is input during TICRR and TICRF read)

(2) Selection of Input Capture Signal Input

TMRIX (input capture input signal of TMR_X) is selected according to the setting of the ICST bit in TCONRI. The input capture signal selection is shown in table 13.5.

Table 13.5 Input Capture Signal Selection

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

13.8 Interrupt Sources

TMR_0, TMR_1, and TMR_Y can generate three types of interrupts: CMIA, CMIB, and OVI. TMR_X can generate four types of interrupts: CMIA, CMIB, OVI, and ICIX. Table 13.6 shows the interrupt sources and priorities. Each interrupt source can be enabled or disabled independently by interrupt enable bits in TCR or TCSR. Independent signals are sent to the interrupt controller for each interrupt.

Table 13.6 Interrupt Sources of 8-Bit Timers TMR_0, TMR_1, TMR_Y, and TMR_X

Channel	Name	Interrupt Source	Interrupt Flag	Interrupt Priority
TMR_0	CMIA0	TCORA_0 compare-match	CMFA	High
	CMIB0	TCORB_0 compare-match	CMFB	
	OVI0	TCNT_0 overflow	OVF	
TMR_1	CMIA1	TCORA_1 compare-match	CMFA	
	CMIB1	TCORB_1 compare-match	CMFB	
	OVI1	TCNT_1 overflow	OVF	
TMR_Y	CMIAY	TCORA_Y compare-match	CMFA	
	CMIBY	TCORB_Y compare-match	CMFB	
	OVIY	TCNT_Y overflow	OVF	
TMR_X	ICIX	Input capture	ICF	
	CMIAX	TCORA_X compare-match	CMFA	
	CMIBX	TCORB_X compare-match	CMFB	
	OVIX	TCNT_X overflow	OVF	Low

13.9 Usage Notes

13.9.1 Conflict between TCNT Write and Counter Clear

If a counter clear signal is generated during the T_2 state of a TCNT write cycle as shown in figure 13.13, clearing takes priority and the counter write is not performed.

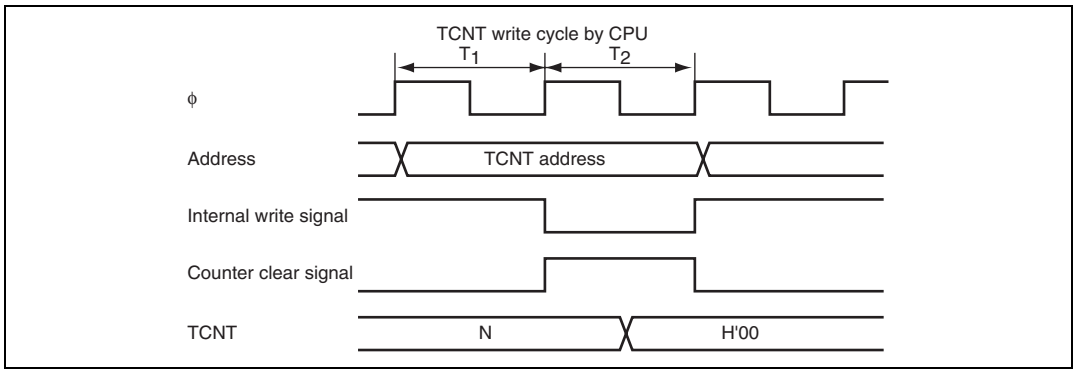


Figure 13.13 Conflict between TCNT Write and Clear

13.9.2 Conflict between TCNT Write and Count-Up

If a count-up occurs during the T_2 state of a TCNT write cycle as shown in figure 13.14, the counter write takes priority and the counter is not incremented.

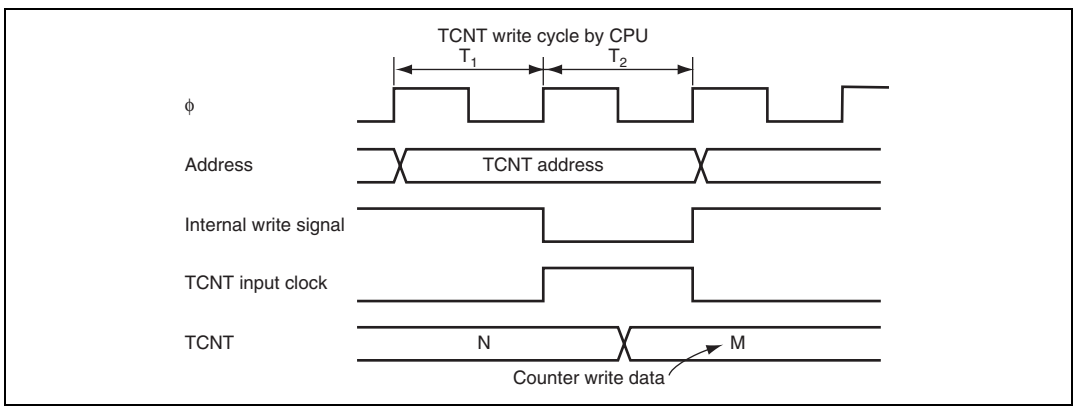


Figure 13.14 Conflict between TCNT Write and Count-Up

13.9.3 Conflict between TCOR Write and Compare-Match

If a compare-match occurs during the T_2 state of a TCOR write cycle as shown in figure 13.15, the TCOR write takes priority and the compare-match signal is disabled. With TMR_X, a TICR input capture conflicts with a compare-match in the same way as with a write to TCORC. In this case also, the input capture takes priority and the compare-match signal is disabled.

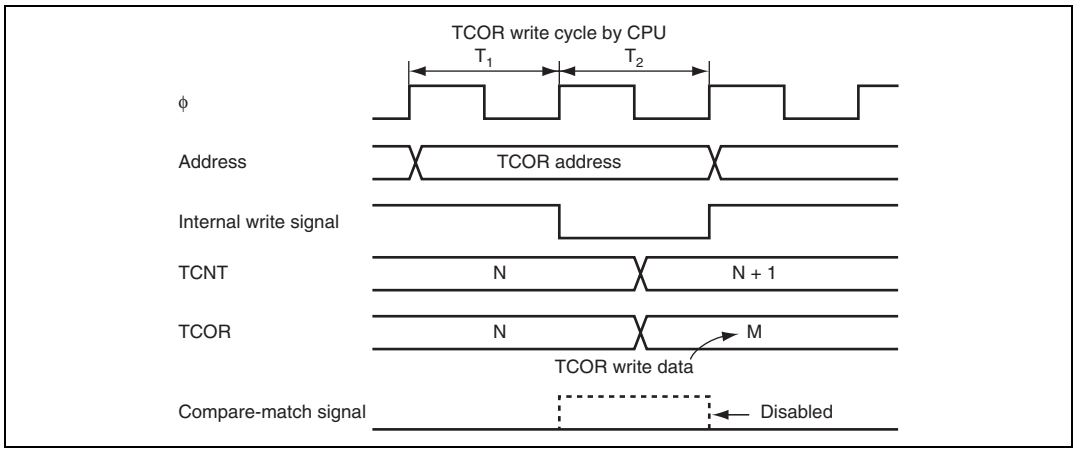


Figure 13.15 Conflict between TCOR Write and Compare-Match

13.9.4 Conflict between Compare-Matches A and B

If compare-matches A and B occur at the same time, the operation follows the output status that is defined for compare-match A or B, according to the priority of the timer output shown in table 13.7.

Table 13.7 Timer Output Priorities

Output Setting	Priority
Toggle output	High
1 output	↑
0 output	↑
No change	Low

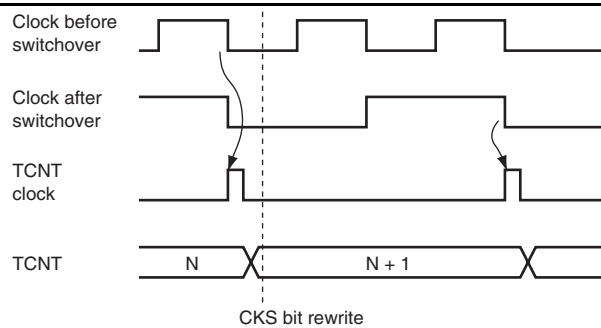
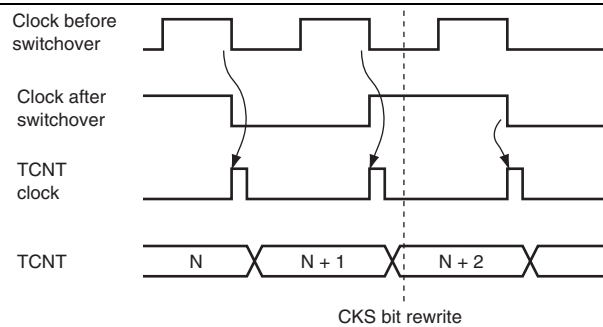
13.9.5 Switching of Internal Clocks and TCNT Operation

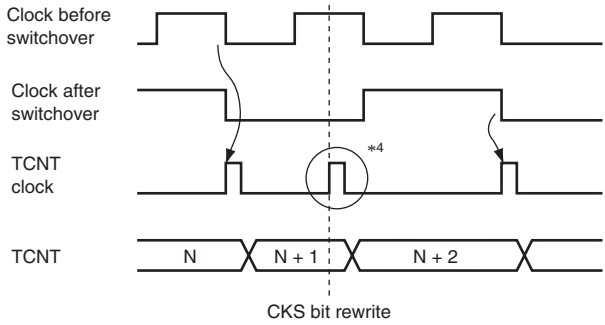
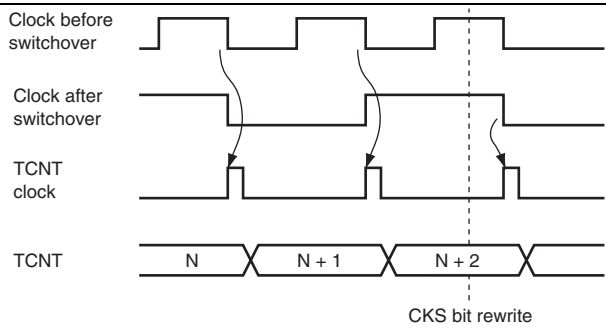
TCNT may increment erroneously when the internal clock is switched over. Table 13.8 shows the relationship between the timing at which the internal clock is switched (by writing to the CKS1 and CKS0 bits) and the TCNT operation.

When the TCNT clock is generated from an internal clock, the falling edge of the internal clock pulse is detected. If clock switching causes a change from high to low level, as shown in no. 3 in table 13.8, a TCNT clock pulse is generated on the assumption that the switchover is a falling edge, and TCNT is incremented.

Erroneous incrementation can also happen when switching between internal and external clocks.

Table 13.8 Switching of Internal Clocks and TCNT Operation

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

No.	Timing of Switchover by Means of CKS1 and CKS0 Bits	TCNT Clock Operation
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.

13.9.6 Mode Setting with Cascaded Connection

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

13.9.7 Module Stop Mode Setting

TMR operation can be enabled or disabled using the module stop control register. The initial setting is for TMR operation to be halted. Register access is enabled by canceling the module stop mode. For details, see section 26, Power-Down Modes.

Section 14 Watchdog Timer (WDT)

This LSI incorporates two watchdog timer channels (WDT_0 and WDT_1). The watchdog timer can generate an internal reset signal or an internal NMI interrupt signal if a system crash prevents the CPU from writing to the timer counter, thus allowing it to overflow.

When this watchdog function is not needed, the WDT can be used as an interval timer. In interval timer operation, an interval timer interrupt is generated each time the counter overflows. A block diagram of the WDT_0 and WDT_1 are shown in figure 14.1.

14.1 Features

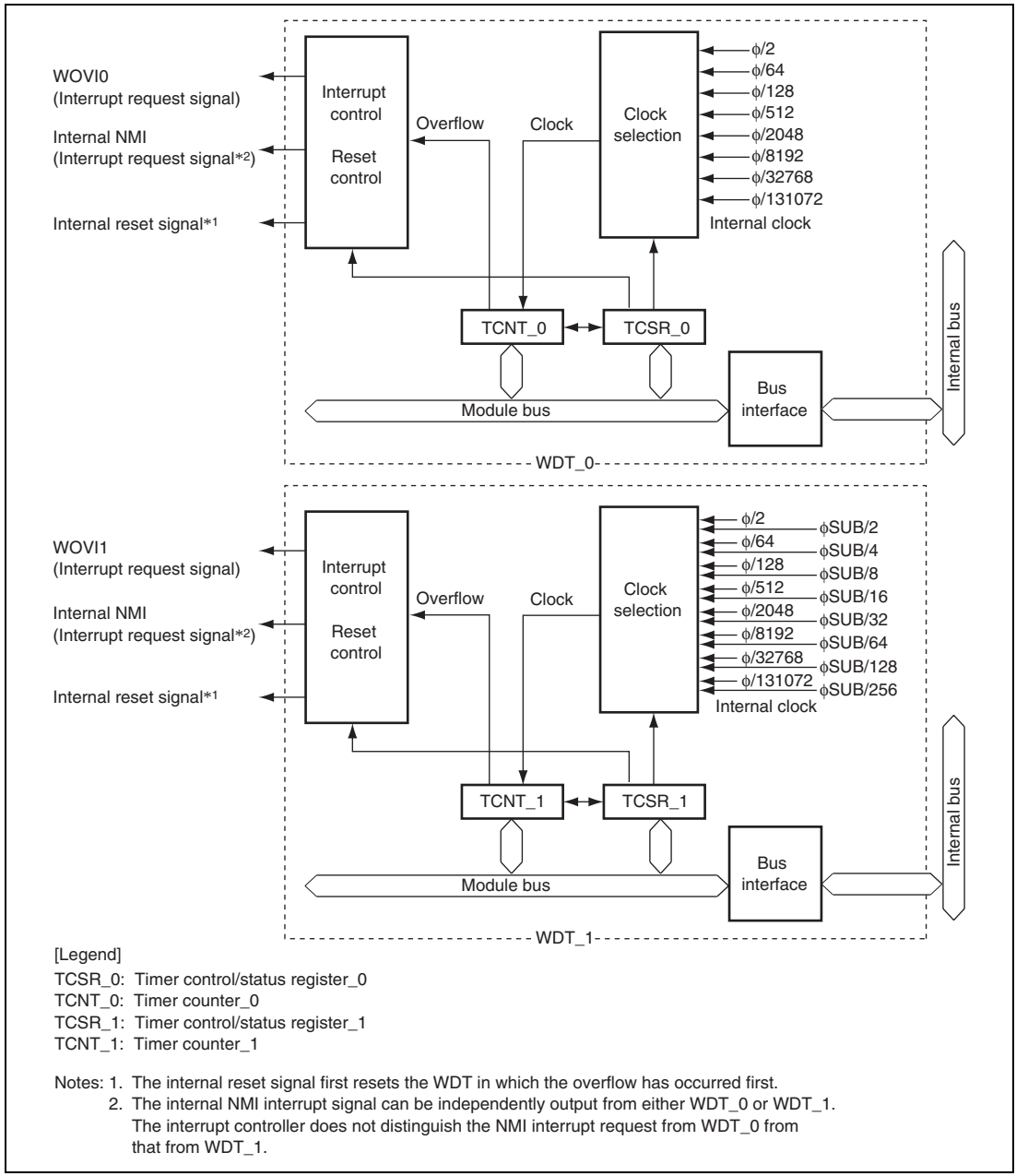
- Selectable from eight (WDT_0) or 16 (WDT_1) counter input clocks.
- Switchable between watchdog timer mode and interval timer mode

Watchdog Timer Mode:

- If the counter overflows, whether an internal reset or an internal NMI interrupt is generated can be selected.

Interval Timer Mode:

- If the counter overflows, an interval 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 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_1. The interrupt controller does not distinguish the NMI interrupt request from WDT_0 from that from WDT_1.

Figure 14.1 Block Diagram of WDT

14.2 Input/Output Pins

The WDT has the pins listed in table 14.1.

Table 14.1 Pin Configuration

Name	Pin Name	I/O	Function
External sub-clock input pin	EXCL	Input	Inputs the clock pulses to the WDT_1 prescaler counter

14.3 Register Descriptions

The WDT has the following registers. To prevent accidental overwriting, TCSR and TCNT have to be written to in a method different from normal registers. For details, see section 14.6.1, Notes on Register Access. For details on the system control register, see section 3.2.2, System Control Register (SYSCR).

Table 14.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Data Bus Width
Channel 0	Timer counter_0	TCNT_0	R/W	H'00	H'FFA8	16
					H'FFA9*	8
	Timer control/status register_0	TCSR_0	R/W	H'00	H'FFA8	16
					H'FFA8*	8
Channel 1	Timer counter_1	TCNT_1	R/W	H'00	H'FFEA	16
					H'FFEB*	8
	Timer control/status register_1	TCSR_1	R/W	H'00	H'FFEA	16
					H'FFEA*	8

Note: * Address in the upper cell: when writing.
Address in the lower cell: when reading

14.3.1 Timer Counter (TCNT)

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

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

14.3.2 Timer Control/Status Register (TCSR)

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

- TCSR_0

Bit	Bit Name	Initial Value	R/W	Description
7	OVF	0	R/(W)*	Overflow Flag Indicates that TCNT has overflowed (changes from H'FF to H'00). [Setting condition] When TCNT overflows (changes from H'FF to H'00) When internal reset request generation is selected in watchdog timer mode, OVF is cleared automatically by the internal reset. [Clearing conditions] • When TCSR is read when OVF = 1, then 0 is written to OVF • When 0 is written to TME
6	WT/$\overline{IT}$	0	R/W	Timer Mode Select Selects whether the WDT is used as a watchdog timer or interval timer. 0: Interval timer mode 1: Watchdog timer mode
5	TME	0	R/W	Timer Enable When this bit is set to 1, TCNT starts counting. When this bit is cleared, TCNT stops counting and is initialized to H’00.
4	—	0	R/(W)	Reserved The initial value should not be changed.

Bit	Bit Name	Initial Value	R/W	Description
3	RST/$\overline{\text{NMI}}$	0	R/W	Reset or NMI Selects to request an internal reset or an NMI interrupt when TCNT has overflowed. 0: An NMI interrupt is requested 1: An internal reset is requested
2	CKS2	0	R/W	Clock Select 2 to 0 Selects the clock source to be input to TCNT. The overflow frequency for ϕ = 20 MHz is enclosed in parentheses. 000: ϕ/2 (frequency: 25.6 μs) 001: ϕ/64 (frequency: 819.2 μs) 010: ϕ/128 (frequency: 1.6 ms) 011: ϕ/512 (frequency: 6.6 ms) 100: ϕ/2048 (frequency: 26.2 ms) 101: ϕ/8192 (frequency: 104.9 ms) 110: ϕ/32768 (frequency: 419.4 ms) 111: ϕ/131072 (frequency: 1.68 s)
1	CKS1	0	R/W	
0	CKS0	0	R/W	

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

- TCSR_1

Bit	Bit Name	Initial Value	R/W	Description
7	OVF	0	R/(W)*[1]	Overflow Flag Indicates that TCNT has overflowed (changes from H'FF to H'00). [Setting condition] When TCNT overflows (changes from H'FF to H'00) When internal reset request generation is selected in watchdog timer mode, OVF is cleared automatically by the internal reset. [Clearing conditions] When TCSR is read when OVF = 1*[2], then 0 is written to OVF When 0 is written to TME
6	WT/$\overline{IT}$	0	R/W	Timer Mode Select Selects whether the WDT is used as a watchdog timer or interval timer. 0: Interval timer mode 1: Watchdog timer mode
5	TME	0	R/W	Timer Enable When this bit is set to 1, TCNT starts counting. When this bit is cleared, TCNT stops counting and is initialized to H'00.
4	PSS	0	R/W	Prescaler Select Selects the clock source to be input to TCNT. 0: Counts the divided cycle of ϕ–based prescaler (PSM) 1: Counts the divided cycle of ϕSUB–based prescaler (PSS)
3	RST/$\overline{NMI}$	0	R/W	Reset or NMI Selects to request an internal reset or an NMI interrupt when TCNT has overflowed. 0: An NMI interrupt is requested 1: An internal reset is requested

Bit	Bit Name	Initial Value	R/W	Description
2	CKS2	0	R/W	Clock Select 2 to 0
1	CKS1	0	R/W	Selects the clock source to be input to TCNT. The overflow cycle for ϕ = 20 MHz and ϕSUB = 32.768 kHz is enclosed in parentheses.
0	CKS0	0	R/W	
				When PSS = 0:
				000: ϕ/2 (frequency: 25.6 μs)
				001: ϕ/64 (frequency: 819.2 μs)
				010: ϕ/128 (frequency: 1.6 ms)
				011: ϕ/512 (frequency: 6.6 ms)
				100: ϕ/2048 (frequency: 26.2 ms)
				101: ϕ/8192 (frequency: 104.9 ms)
				110: ϕ/32768 (frequency: 419.4 ms)
				111: ϕ/131072 (frequency: 1.68 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 be read at least twice.

14.4 Operation

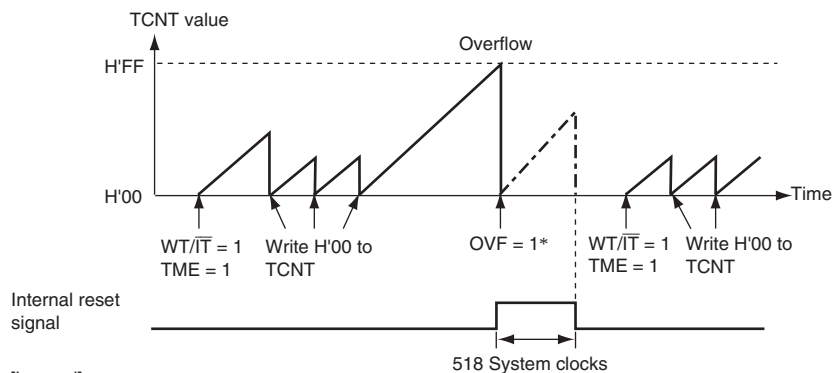
14.4.1 Watchdog Timer Mode

To use the WDT as a watchdog timer, set the WT/$\overline{IT}$ bit and the TME bit in TCSR to 1. While the WDT is used as a watchdog timer, if TCNT overflows without being rewritten because of a system malfunction or another error, an internal reset or NMI interrupt request is generated. TCNT does not overflow while the system is operating normally. Software must prevent TCNT overflows by rewriting the TCNT value (normally be writing H'00) before overflows occurs.

If the RST/$\overline{NMI}$ bit of TCSR is set to 1, when the TCNT overflows, an internal reset signal for this LSI is issued for 518 system clocks as shown in figure 14.2. If the RST/$\overline{NMI}$ bit is cleared to 0, when the TCNT overflows, an NMI interrupt request is generated.

An internal reset request from the watchdog timer and a reset input from the $\overline{RES}$ pin are processed in the same vector. Reset source can be identified by the XRST bit status in SYSCR. If a reset caused by a signal input to the $\overline{RES}$ pin occurs at the same time as a reset caused by a WDT overflow, the $\overline{RES}$ 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 NMI pin are processed in the same vector. Do not handle an NMI interrupt request from the watchdog timer and an interrupt request from the NMI pin at the same time.



[Legned]
WT/$\overline{IT}$: Timer mode select bit
TME: Timer enable bit
OVF: Overflow flag
Note * After the OVF bit becomes 1, it is cleared to 0 by an internal reset. The XRST bit is also cleared to 0.

Figure 14.2 Watchdog Timer Mode (RST/$\overline{NMI}$ = 1) Operation

14.4.2 Interval Timer Mode

When the WDT is used as an interval timer, an interval timer interrupt (WOVI) is generated each time the TCNT overflows, as shown in figure 14.3. Therefore, an interrupt can be generated at intervals. When the TCNT overflows in interval timer mode, an interval timer interrupt (WOVI) is requested at the same time the OVF flag of TCSR is set to 1. The timing is shown figure 14.4.

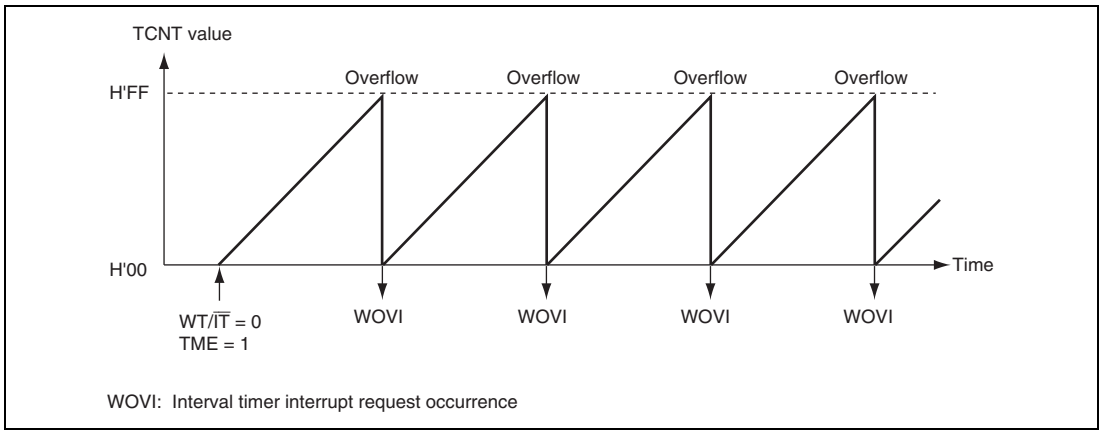


Figure 14.3 Interval Timer Mode Operation

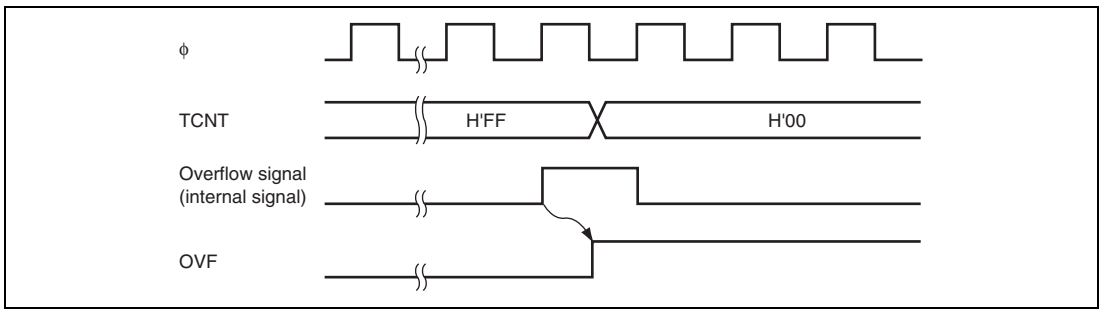


Figure 14.4 OVF Flag Set Timing

14.5 Interrupt Sources

During interval timer mode operation, an overflow generates an interval timer interrupt (WOVI). The interval timer interrupt is requested whenever the OVF flag is set to 1 in TCSR. OVF must be cleared to 0 in the interrupt handling routine.

When the NMI interrupt request is selected in watchdog timer mode, an NMI interrupt request is generated by an overflow

Table 14.3 WDT Interrupt Source

Name	Interrupt Source	Interrupt Flag
WOVI	TCNT overflow	OVF

14.6 Usage Notes

14.6.1 Notes on Register Access

The watchdog timer's registers, TCNT and TCSR differ from other registers in being more difficult to write to. The procedures for writing to and reading from these registers are given below.

(1) Writing to TCNT and TCSR (Example of WDT_0)

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

TCNT and TCSR both have the same write address. Therefore, satisfy the relative condition shown in figure 14.5 to write to TCNT or TCSR. To write to TCNT, the higher bytes must contain the value H'5A and the lower bytes must contain the write data before the transfer instruction execution. To write to TCSR, the higher bytes must contain the value H'A5 and the lower bytes must contain the write data.

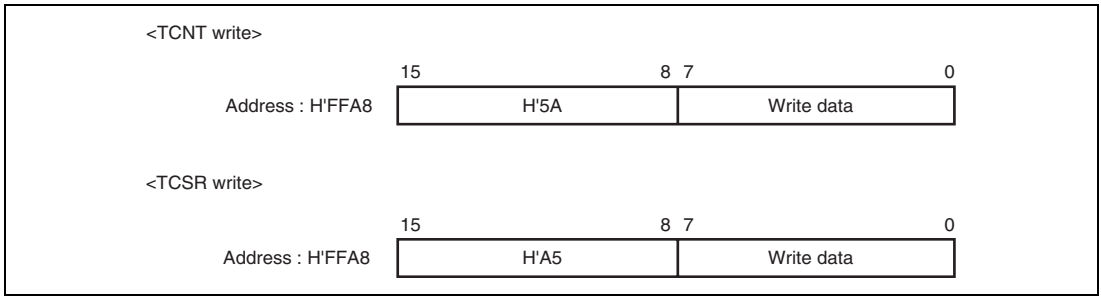


Figure 14.5 Writing to TCNT and TCSR (WDT_0)

(2) 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 for TCSR and H'FFA9 for TCNT.

14.6.2 Conflict between Timer Counter (TCNT) Write and Increment

If a timer counter clock pulse is generated during the T_2 state of a TCNT write cycle, the write takes priority and the timer counter is not incremented. Figure 14.6 shows this operation.

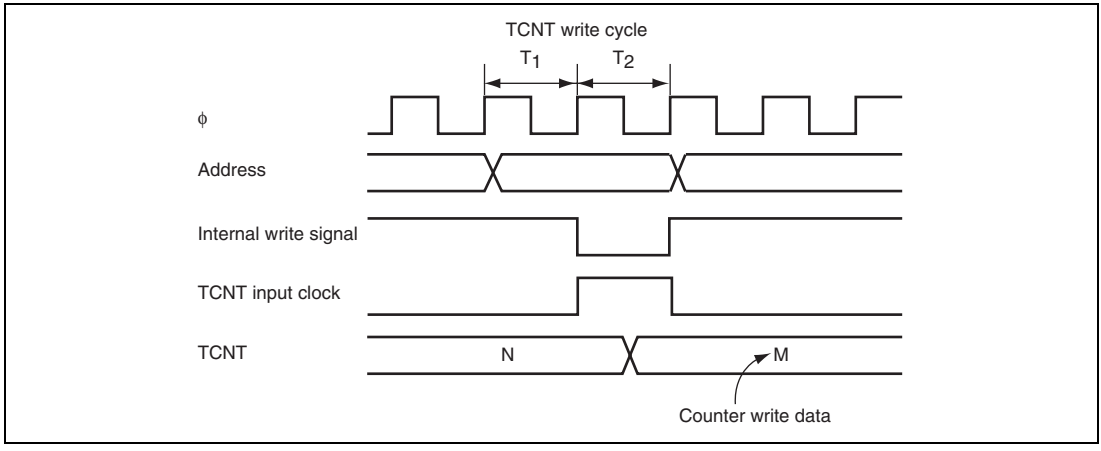


Figure 14.6 Conflict between TCNT Write and Increment

14.6.3 Changing Values of CKS2 to CKS0 Bits

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

14.6.4 Changing Value of PSS Bit

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

14.6.5 Switching between Watchdog Timer Mode and Interval Timer Mode

If the mode is switched from/to watchdog timer to/from interval timer, while the WDT is operating, errors could occur in the operation. Software must stop the watchdog timer (by clearing the TME bit to 0) before switching the mode.

Section 15 Serial Communication Interface (SCI)

This LSI has two independent serial communication interface (SCI) channels. The SCI can handle both asynchronous and clocked synchronous serial communication. Asynchronous serial data communication can be carried out with standard asynchronous communication chips such as a Universal Asynchronous Receiver/Transmitter (UART) or Asynchronous Communication Interface Adapter (ACIA). A function is also provided for serial communication between processors (multiprocessor communication function). The SCI also supports the smart card (IC card) interface based on ISO/IEC 7816-3 (Identification Card) as an enhanced asynchronous communication function.

15.1 Features

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

  The transmitter and receiver are mutually independent, enabling transmission and reception to be executed simultaneously. Double-buffering is used in both the transmitter and the receiver, 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 card interface).
- Choice of LSB-first or MSB-first transfer (except in the case of asynchronous mode 7-bit data)
- Four interrupt sources

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

Asynchronous Mode:

- Data length: 7 or 8 bits
- Stop bit length: 1 or 2 bits
- Parity: Even, odd, or none
- Receive error detection: Parity, overrun, and framing errors
- Break detection: Break can be detected by reading the RxD pin level directly in case of a framing error
- Multiprocessor communication capability

Clocked Synchronous Mode:

- 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 reception.
- Data can be automatically re-transmitted on detection of an error signal during transmission.
- Both direct convention and inverse convention are supported.

Figure 15.1 shows a block diagram of SCI.

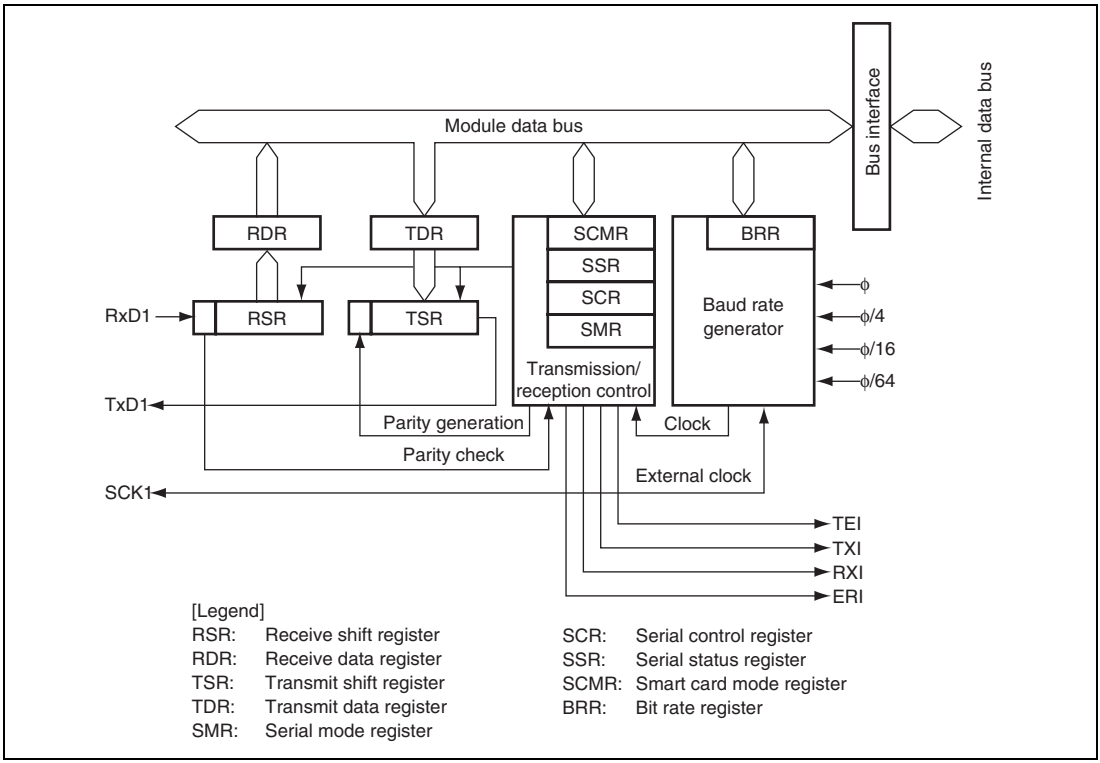


Figure 15.1 Block Diagram of SCI

15.2 Input/Output Pins

Table 15.1 shows the input/output pins for each SCI channel.

Table 15.1 Pin Configuration

Channel	Pin Name*	Input/Output	Function
1	SCK1	Input/Output	Channel 1 clock input/output
	RxD1	Input	Channel 1 receive data input
	TxD1	Output	Channel 1 transmit data output
2	SCK2	Input/Output	Channel 2 clock input/output
	RxD2	Input	Channel 2 receive data input
	TxD2	Output	Channel 2 transmit data output

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

15.3 Register Descriptions

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

Table 15.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Data Bus Width
Channel 0	Serial mode register_1	SMR_1	R/W	H'00	H'FF88	8
	Bit rate register_1	BRR_1	R/W	H'FF	H'FF89	8
	Serial control register_1	SCR_1	R/W	H'00	H'FF8A	8
	Transmit data register_1	TDR_1	R/W	H'FF	H'FF8B	8
	Serial status register_1	SSR_1	R/W	H'84	H'FF8C	8
	Receive data register_1	RDR_1	R	H'00	H'FF8D	8
	Smart card mode register_1	SCMR_1	R/W	H'F2	H'FF8E	8
Channel 1	Serial mode register_2	SMR_2	R/W	H'00	H'FFA0	8
	Bit rate register_2	BRR_2	R/W	H'FF	H'FFA1	8
	Serial control register_2	SCR_2	R/W	H'00	H'FFA2	8
	Transmit data register_2	TDR_2	R/W	H'FF	H'FFA3	8
	Serial status register_2	SSR_2	R/W	H'84	H'FFA4	8
	Receive data register_2	RDR_2	R	H'00	H'FFA5	8
	Smart card mode register_2	SCMR_2	R/W	H'F2	H'FFA6	8

15.3.1 Receive Shift Register (RSR)

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

15.3.2 Receive Data Register (RDR)

RDR is an 8-bit register that stores receive data. When the SCI has received one frame of serial data, it transfers the received serial data from RSR to RDR where it is stored. After this, RSR can receive the next data. Since RSR and RDR function as a double buffer in this way, continuous receive operations be performed. After confirming that the RDRF bit in SSR is set to 1, read RDR for only once. RDR cannot be written to by the CPU. The initial value of RDR is H'00.

15.3.3 Transmit Data Register (TDR)

TDR is an 8-bit register that stores transmit data. When the SCI detects that TSR is empty, it transfers the transmit data written in TDR to TSR and starts transmission. The double-buffered structures of TDR and TSR enable continuous serial transmission. If the next transmit data has already been written to TDR when one frame of data is transmitted, the SCI transfers the written data to TSR to continue transmission. Although TDR can be read from or written to by the CPU at all times, to achieve reliable serial transmission, write transmit data to TDR for only once after confirming that the TDRE bit in SSR is set to 1. The initial value of TDR is H'FF.

15.3.4 Transmit Shift Register (TSR)

TSR is a shift register that transmits serial data. To perform serial data transmission, the SCI first transfers transmit data from TDR to TSR, and then sends the data to the TxD pin. TSR cannot be directly accessed by the CPU.

15.3.5 Serial Mode Register (SMR)

SMR is used to set the SCI's serial transfer format and select the baud rate generator clock source. Some bits in SMR have different functions in normal mode and smart card interface mode. The CPU can always read SMR. The CPU can write to SMR only at the initial settings; do not have the CPU write to SMR in transmission, reception, and simultaneous data transmission and reception.

- Bit Functions in Normal Serial Communication Interface Mode (when SMIF in SCMR = 0)

Bit	Bit Name	Initial Value	R/W	Description
7	C/$\overline{A}$	0	R/W	Communication Mode 0: Asynchronous mode 1: Clocked synchronous mode
6	CHR	0	R/W	Character Length (enabled only in asynchronous mode) 0: Selects 8 bits as the data length. 1: Selects 7 bits as the data length. LSB-first is fixed and the MSB of TDR is not transmitted in transmission. In clocked synchronous mode, a fixed data length of 8 bits is used.
5	PE	0	R/W	Parity Enable (enabled only in asynchronous mode) When this bit is set to 1, the parity bit is added to transmit data before transmission, and the parity bit is checked in reception. For a multiprocessor format, parity bit addition and checking are not performed regardless of the PE bit 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 mode) 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 the second stop bit is 0, it is treated as the start bit of the next transmit frame.

Bit	Bit Name	Initial Value	R/W	Description
2	MP	0	R/W	Multiprocessor Mode (enabled only in asynchronous mode) When this bit is set to 1, the multiprocessor communication function is enabled. The PE bit and O/$\overline{E}$ bit settings are invalid in multiprocessor mode.
1	CKS1	0	R/W	Clock Select 1 and 0 These bits select the clock source for the baud rate 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 setting and the baud rate, see section 15.3.9, Bit Rate Register (BRR). n is the decimal display of the value of n in BRR (see section 15.3.9, Bit Rate Register (BRR)).
0	CKS0	0	R/W	

- 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. In GSM mode, the TEND set timing is put forward to 11.0 etu* from the start and the clock output control function is appended. For details, see section 15.7.8, Clock Output Control.
6	BLK	0	R/W	Setting this bit to 1 allows block transfer mode operation. For details, see section 15.7.3, Block Transfer Mode.
5	PE	0	R/W	Parity Enable (valid only in asynchronous mode) When this bit is set to 1, the parity bit is added to transmit data before transmission, and the parity bit is checked in reception. Set this bit to 1 in smart card interface mode.

Bit	Bit Name	Initial Value	R/W	Description
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 interface mode, see section 15.7.2, Data Format (Except in Block Transfer Mode).
3	BCP1	0	R/W	Basic Clock Pulse 1 and 0
2	BCP0	0	R/W	These bits select the number of basic clock cycles in a 1-bit data transfer time in smart card interface mode. 00: 32 clock cycles (S = 32) 01: 64 clock cycles (S = 64) 10: 372 clock cycles (S = 372) 11: 256 clock cycles (S = 256) For details, see section 15.7.4, Receive Data Sampling Timing and Reception Margin. S is described in section 15.3.9, 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 rate 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 setting and the baud rate, see section 15.3.9, Bit Rate Register (BRR). n is the decimal display of the value of n in BRR (see section 15.3.9, Bit Rate Register (BRR)).

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

15.3.6 Serial Control Register (SCR)

SCR is a register that performs enabling or disabling of SCI transfer operations and interrupt requests, and selection of the transfer clock source. For details on interrupt requests, see section 15.8, Interrupt Sources. Some bits in SCR have different functions in normal mode and smart card interface mode. The CPU can always read SCR. The CPU can write to SCR only at the initial settings; do not have the CPU write to SCR in transmission, reception, and simultaneous data transmission and reception.

- Bit Functions in Normal Serial Communication Interface Mode (when SMIF in SCMR = 0)

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable When this bit is set to 1, a TXI interrupt request is enabled.
6	RIE	0	R/W	Receive Interrupt Enable When this bit is set to 1, RXI and ERI interrupt requests are enabled.
5	TE	0	R/W	Transmit Enable When this bit is set to 1, transmission is enabled.
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 when the MP bit in SMR is 1 in asynchronous mode) When this bit is set to 1, receive data in which the multiprocessor bit is 0 is skipped, and setting of the RDRF, FER, and ORER status flags in SSR is disabled. On receiving data in which the multiprocessor bit is 1, this bit is automatically cleared and normal reception is resumed. For details, see section 15.5, Multiprocessor Communication Function.
2	TEIE	0	R/W	Transmit End Interrupt Enable When this bit is set to 1, a TEI interrupt request is enabled.

Bit	Bit Name	Initial Value	R/W	Description
1	CKE1	0	R/W	Clock Enable 1 and 0
0	CKE0	0	R/W	These bits select the clock source and SCK pin function. • Asynchronous mode 00: Internal clock (SCK pin functions as I/O port.) 01: Internal clock (Outputs a clock of the same frequency as the bit rate from the SCK pin.) 1x: External clock (Inputs a clock with a frequency 16 times the bit rate from the SCK pin.) • Clocked synchronous mode 0x: Internal clock (SCK pin functions as clock output.) 1x External clock (SCK pin functions as clock input.)

[Legend]

x: Don't care

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

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable When this bit is set to 1,a TXI interrupt request is enabled.
6	RIE	0	R/W	Receive Interrupt Enable When this bit is set to 1, RXI and ERI interrupt requests are enabled.
5	TE	0	R/W	Transmit Enable When this bit is set to 1, transmission is enabled.
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 when the MP 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	Controls the clock output from the SCK pin. In GSM mode, clock output can be dynamically switched. For details, see section 15.7.8, Clock Output Control. • When GM in SMR = 0 00: Output disabled (SCK pin functions as I/O port.) 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

15.3.7 Serial Status Register (SSR)

SSR is a register containing status flags of the SCI and multiprocessor bits for transfer. TDRE, RDRF, ORER, PER, and FER can only be cleared. Some bits in SSR have different functions in normal mode and smart card interface mode.

- Bit Functions in Normal Serial Communication Interface Mode (when SMIF in SCMR = 0)

Bit	Bit Name	Initial Value	R/W	Description
7	TDRE	1	R/(W)*	Transmit Data Register Empty 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 TDR is ready for data write [Clearing condition] When 0 is written to TDRE after reading TDRE = 1
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 receive data is transferred from RSR to RDR [Clearing condition] When 0 is written to RDRF after reading RDRF = 1 The RDRF flag is not affected and retains its previous value when the RE bit in SCR is cleared to 0.
5	ORER	0	R/(W)*	Overrun Error [Setting condition] When the next serial reception is completed while RDRF = 1 [Clearing condition] When 0 is written to ORER after reading ORER = 1

Bit	Bit Name	Initial Value	R/W	Description
4	FER	0	R/(W)*	Framing Error [Setting condition] 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 checked.
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 bit of a 1-byte serial transmit character [Clearing condition] When 0 is written to TDRE after reading TDRE = 1
1	MPB	0	R	Multiprocessor Bit MPB stores the multiprocessor bit in the receive frame. When the RE bit in SCR is cleared to 0 its previous state is retained.
0	MPBT	0	R/W	Multiprocessor Bit Transfer MPBT stores the multiprocessor bit to be added to the transmit frame.

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

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

Bit	Bit Name	Initial Value	R/W	Description
7	TDRE	1	R/(W)*	Transmit Data Register Empty 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 TDR can be written to. [Clearing condition] When 0 is written to TDRE after reading TDRE = 1
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 receive data is transferred from RSR to RDR [Clearing condition] When 0 is written to RDRF after reading RDRF = 1 The RDRF flag is not affected and retains its previous 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 RDRF = 1 [Clearing condition] When 0 is written to ORER after reading ORER = 1
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

Bit	Bit Name	Initial Value	R/W	Description
3	PER	0	R/(W)*1	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 TEND is set to 1 when the receiving end acknowledges no error signal and the next transmit data is ready to be transferred to TDR. [Setting conditions] • When both TE and EPS in SCR are 0 • When ERS = 0 and TDRE = 1 after a specified time passed after the start of 1-byte data transfer. The set timing depends on the register setting as follows. • 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 condition] When 0 is written to TDRE after reading TDRE = 1
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)

15.3.8 Smart Card Mode Register (SCMR)

SCMR selects smart card interface mode and its format.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	R	Reserved These bits are always read as 1 and cannot be modified.
3	SDIR	0	R/W	Smart Card Data Transfer Direction Selects the serial/parallel conversion format. 0: TDR contents are transmitted with LSB-first. Receive data is stored as LSB first in RDR. 1: TDR contents are transmitted with MSB-first. Receive data is stored as MSB first in RDR. The SDIR bit is valid only when the 8-bit data format is used for transmission/reception; when the 7-bit data format is used, data is always transmitted/received with LSB-first.
2	SINV	0	R/W	Smart Card Data Invert Specifies inversion of the data logic level. The SINV bit does not affect the logic level of the parity bit. When the parity bit is inverted, invert the O/$\overline{E}$ bit in SMR. 0: TDR contents are transmitted as they are. Receive data is stored as it is in RDR. 1: TDR contents are inverted before being transmitted. Receive data is stored in inverted form in RDR.
1	—	1	R	Reserved This bit is always read as 1 and cannot be modified.
0	SMIF	0	R/W	Smart Card Interface Mode Select When this bit is set to 1, smart card interface mode is selected. 0: Normal asynchronous or clocked synchronous mode 1: Smart card interface mode

15.3.9 Bit Rate Register (BRR)

BRR is an 8-bit register that adjusts the bit rate. As the SCI performs baud rate generator control independently for each channel, different bit rates can be set for each channel. Table 15.3 shows the relationships between the N setting in BRR and bit rate B for normal asynchronous mode and clocked synchronous mode, and smart card interface mode. The initial value of BRR is H'FF. The CPU can always read BRR. The CPU can write to BRR only at the initial settings; do not have the CPU write to BRR in transmission, reception, and simultaneous data transmission and reception.

Table 15.3 Relationships between N Setting in BRR and Bit Rate B

Mode	Bit Rate	Error
Asynchronous mode	$B = \frac{\phi \times 10^6}{64 \times 2^{2n-1} \times (N+1)}$	$\text{Error (\%)} = \{\frac{\phi \times 10^6}{B \times 64 \times 2^{2n-1} \times (N+1)} - 1\} \times 100$
Clocked synchronous mode	$B = \frac{\phi \times 10^6}{8 \times 2^{2n-1} \times (N+1)}$	—
Smart card interface mode	$B = \frac{\phi \times 10^6}{S \times 2^{2n+1} \times (N+1)}$	$\text{Error (\%)} = \{\frac{\phi \times 10^6}{B \times S \times 2^{2n+1} \times (N+1)} - 1\} \times 100$

[Legend]
B: Bit rate (bit/s)
N: BRR setting for baud rate generator ($0 \le N \le 255$)
ϕ: Operating frequency (MHz)
n and S: Determined by the SMR settings shown in the following table

SMR Setting

CKS1	CKS0	n
0	0	0
0	1	1
1	0	2
1	1	3

SMR Setting

BCP1	BCP0	S
0	0	32
0	1	64
1	0	372
1	1	256

Table 15.4 shows sample N settings in BRR in normal asynchronous mode. Table 15.5 shows the maximum bit rate settable for each frequency. Table 15.7 and 15.9 show sample N settings in BRR in clocked synchronous mode and smart card interface mode, respectively. In smart card interface mode, the number of basic clock cycles S in a 1-bit data transfer time can be selected. For details, see section 15.7.4, Receive Data Sampling Timing and Reception Margin. Tables 15.6 and 15.8 show the maximum bit rates with external clock input.

Table 15.4 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (1)

	Operating Frequency ϕ (MHz)											
	8			9.8304			10			12		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	141	0.03	2	174	–0.26	2	177	–0.25	2	212	0.03
150	2	103	0.16	2	127	0.00	2	129	0.16	2	155	0.16
300	1	207	0.16	1	255	0.00	2	64	0.16	2	77	0.16
600	1	103	0.16	1	127	0.00	1	129	0.16	1	155	0.16
1200	0	207	0.16	0	255	0.00	1	64	0.16	1	77	0.16
2400	0	103	0.16	0	127	0.00	0	129	0.16	0	155	0.16
4800	0	51	0.16	0	63	0.00	0	64	0.16	0	77	0.16
9600	0	25	0.16	0	31	0.00	0	32	–1.36	0	38	0.16
19200	0	12	0.16	0	15	0.00	0	15	1.73	0	19	–2.34
31250	0	7	0.00	0	9	–1.70	0	9	0.00	0	11	0.00
38400	—	—	—	0	7	0.00	0	7	1.73	0	9	–2.34

	Operating Frequency ϕ (MHz)											
	12.288			14			14.7456			16		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	217	0.08	2	248	–0.17	3	64	0.70	3	70	0.03
150	2	159	0.00	2	181	0.16	2	191	0.00	2	207	0.16
300	2	79	0.00	2	90	0.16	2	95	0.00	2	103	0.16
600	1	159	0.00	1	181	0.16	1	191	0.00	1	207	0.16
1200	1	79	0.00	1	90	0.16	1	95	0.00	1	103	0.16
2400	0	159	0.00	0	181	0.16	0	191	0.00	0	207	0.16
4800	0	79	0.00	0	90	0.16	0	95	0.00	0	103	0.16
9600	0	39	0.00	0	45	–0.93	0	47	0.00	0	51	0.16
19200	0	19	0.00	0	22	–0.93	0	23	0.00	0	25	0.16
31250	0	11	2.40	0	13	0.00	0	14	–1.70	0	15	0.00
38400	0	9	0.00	—	—	—	0	11	0.00	0	12	0.16

[Legend]

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

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

Table 15.4 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (2)

	Operating Frequency ϕ (MHz)											
	17.2032			18			19.6608			20		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	75	0.48	3	79	–0.12	3	86	0.31	3	88	–0.25
150	2	223	0.00	2	233	0.16	2	255	0.00	3	64	0.16
300	2	111	0.00	2	116	0.16	2	127	0.00	2	129	0.16
600	1	223	0.00	1	233	0.16	1	255	0.00	2	64	0.16
1200	1	111	0.00	1	116	0.16	1	127	0.00	1	129	0.16
2400	0	223	0.00	0	233	0.16	0	255	0.00	1	64	0.16
4800	0	111	0.00	0	116	0.16	0	127	0.00	0	129	0.16
9600	0	55	0.00	0	58	–0.69	0	63	0.00	0	64	0.16
19200	0	27	0.00	0	28	1.02	0	31	0.00	0	32	–1.36
31250	0	16	1.20	0	17	0.00	0	19	–1.70	0	19	0.00
38400	0	16	0.00	0	14	–2.34	0	15	0.00	0	15	1.73

[Legend]

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

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

Table 15.5 Maximum Bit Rate for Each Frequency (Asynchronous Mode)

ϕ (MHz)	Maximum Bit Rate (bit/s)	n	N	ϕ (MHz)	Maximum Bit Rate (bit/s)	n	N
8	250000	0	0	14.7456	460800	0	0
9.8304	307200	0	0	16	500000	0	0
10	312500	0	0	17.2032	537600	0	0
12	375000	0	0	18	562500	0	0
12.288	384000	0	0	19.6608	614400	0	0
14	437500	0	0	20	625000	0	0

Table 15.6 Maximum Bit Rate with External Clock Input (Asynchronous Mode)

ϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bit/s)	ϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bit/s)
8	2.0000	125000	14.7456	3.6864	230400
9.8304	2.4576	153600	16	4.0000	250000
10	2.5000	156250	17.2032	4.3008	268800
12	3.0000	187500	18	4.5000	281250
12.288	3.0720	192000	19.6608	4.9152	307200
14	3.5000	218750	20	5.0000	312500

Table 15.7 BRR Settings for Various Bit Rates (Clocked Synchronous Mode)

	Operating Frequency ϕ (MHz)							
	8		10		16		20	
Bit Rate (bit/s)	n	N	n	N	n	N	n	N
110								
250	3	124	—	—	3	249		
500	2	249	—	—	3	124	—	—
1k	2	124	—	—	2	249	—	—
2.5k	1	199	1	249	2	99	2	124
5k	1	99	1	124	1	199	1	249
10k	0	199	0	249	1	99	1	124
25k	0	79	0	99	0	159	0	199
50k	0	39	0	49	0	79	0	99
100k	0	19	0	24	0	39	0	49
250k	0	7	0	9	0	15	0	19
500k	0	3	0	4	0	7	0	9
1M	0	1			0	3	0	4
2.5M			0	0*			0	1
5M							0	0*

[Legend]

Blank: Setting prohibited.

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

*: Continuous transfer or reception is not possible.

Table 15.8 Maximum Bit Rate with External Clock Input (Clocked Synchronous Mode)

φ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bit/s)	φ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bit/s)
8	1.3333	1333333.3	16	2.6667	2666666.7
10	1.6667	1666666.7	18	3.0000	3000000.0
12	2.0000	2000000.0	20	3.3333	3333333.3
14	2.3333	2333333.3			

Table 15.9 BRR Settings for Various Bit Rates (Smart Card Interface Mode, n = 0, s = 372)

Bit Rate (bit/s)	Operating Frequency φ (MHz)											
	10.00			13.00			14.2848			16.00		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	1	30	0	1	-8.99	0	1	0.00	0	1	12.01

Bit Rate (bit/s)	Operating Frequency φ (MHz)					
	18.00			20.00		
	n	N	Error (%)	n	N	Error (%)
9600	0	2	-15.99	0	2	-6.65

Table 15.10 Maximum Bit Rate for Each Frequency (Smart Card Interface Mode, S = 372)

φ (MHz)	Maximum Bit Rate (bit/s)	n	N	φ (MHz)	Maximum Bit Rate (bit/s)	n	N
10.00	13441	0	0	16.00	21505	0	0
13.00	17473	0	0	18.00	24194	0	0
14.2848	19200	0	0	20.00	26882	0	0

15.4 Operation in Asynchronous Mode

Figure 15.2 shows the general format for asynchronous serial communication. One frame consists of a start bit (low level), followed by transmit/receive data, a parity bit, and finally stop bits (high level). In asynchronous serial communication, the transmission line is usually held in the mark state (high level). The SCI monitors the transmission line, and when it goes to the space state (low level), recognizes a start bit and starts serial communication. Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communication. Both the transmitter and the receiver also have a double-buffered structure, so that data can be read or written during transmission or reception, enabling continuous data transfer and reception.

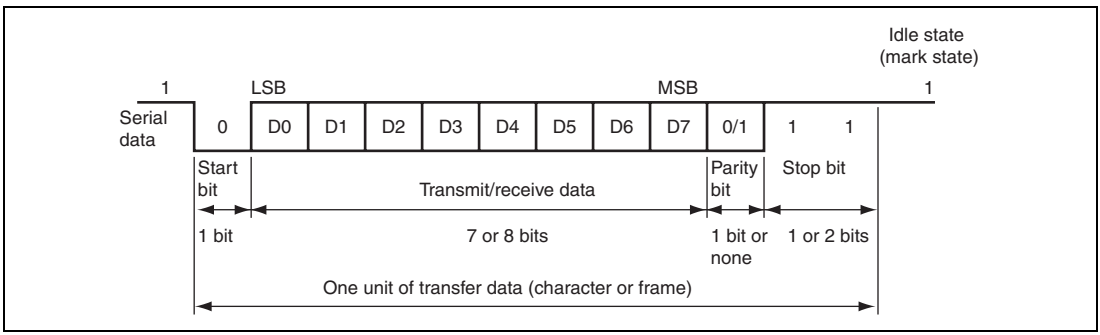


Figure 15.2 Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, Two Stop Bits)

15.4.1 Data Transfer Format

Table 15.11 shows the data transfer formats that can be used in asynchronous mode. Any of 12 transfer formats can be selected according to the SMR setting. For details on the multiprocessor bit, see section 15.5, Multiprocessor Communication Function.

Table 15.11 Serial Transfer Formats (Asynchronous Mode)

SMR Settings				Serial Transmit/Receive Format and Frame Length
CHR	PE	MP	STOP	1 2 3 4 5 6 7 8 9 10 11 12
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

15.4.2 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode

In asynchronous mode, the SCI operates on a basic clock with a frequency of 16 times the bit rate. In reception, the SCI samples the falling edge of the start bit using the basic clock, and performs internal synchronization. Since receive data is latched internally at the rising edge of the 8th pulse of the basic clock, data is latched at the middle of each bit, as shown in figure 15.3. Thus the reception margin in asynchronous mode is determined by formula (1) below.

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

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

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

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

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

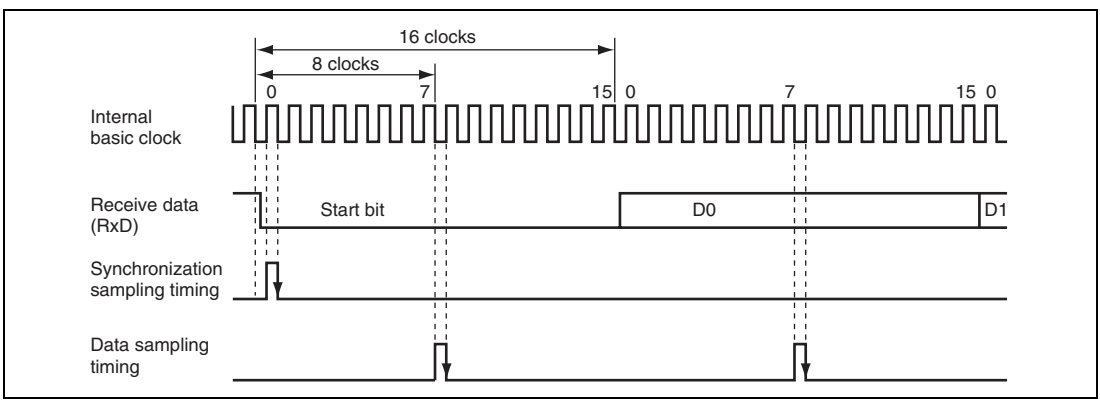


Figure 15.3 Receive Data Sampling Timing in Asynchronous Mode

15.4.3 Clock

Either an internal clock generated by the on-chip baud rate generator or an external clock input at the SCK pin can be selected as the SCI's transfer clock, according to the setting of the C/$\overline{A}$ bit in SMR and the CKE1 and CKE0 bits in SCR. When an external clock is input at the SCK pin, the clock frequency should be 16 times the bit rate used.

When the SCI is operated on an internal clock, the clock can be output from the SCK pin. The frequency of the clock output in this case is equal to the bit rate, and the phase is such that the rising edge of the clock is in the middle of the transmit data, as shown in figure 15.4.

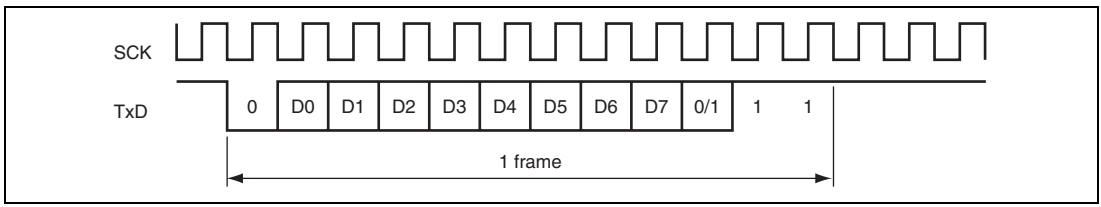
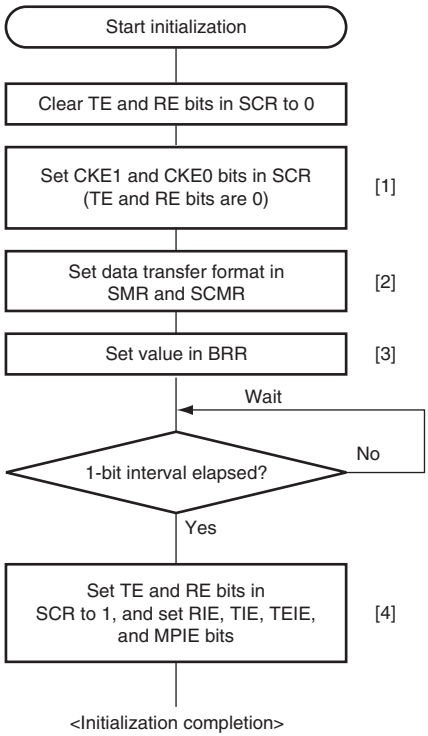


Figure 15.4 Relation between Output Clock and Transmit Data Phase (Asynchronous Mode)

15.4.4 SCI Initialization (Asynchronous Mode)

Before transmitting and receiving data, you should first clear the TE and RE bits in SCR to 0, then initialize the SCI as shown in figure 15.5. When the operating mode, transfer format, etc., is changed, the TE and RE bits must be cleared to 0 before making the change using the following procedure. When the TE bit is cleared to 0, the TDRE flag in SSR is set to 1. Note that clearing the RE bit to 0 does not initialize the contents of the RDRF, PER, FER, and ORER flags in SSR, or the contents of RDR. When the external clock is used in asynchronous mode, the clock must be supplied even during initialization.



[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 15.5 Sample SCI Initialization Flowchart

15.4.5 Serial Data Transmission (Asynchronous Mode)

Figure 15.6 shows an example of the operation for transmission in asynchronous mode. In transmission, the SCI operates as described below.

1. The SCI monitors the TDRE flag in SSR, and if it is cleared to 0, recognizes that data has been written to TDR, and transfers the data from TDR to TSR.
2. After transferring data from TDR to TSR, the SCI sets the TDRE flag to 1 and starts transmission. If the TIE bit in SCR is set to 1 at this time, a transmit data empty interrupt request (TXI) is generated. Because the TXI interrupt routine writes the next transmit data to TDR before transmission of the current transmit data has finished, continuous transmission can be enabled.
3. Data is sent from the TxD pin in the following order: start bit, transmit data, parity bit or 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, and then 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 the “mark state” is entered in which 1 is output. If the TEIE bit in SCR is set to 1 at this time, a TEI interrupt request is generated.

Figure 15.7 shows a sample flowchart for transmission in asynchronous mode.

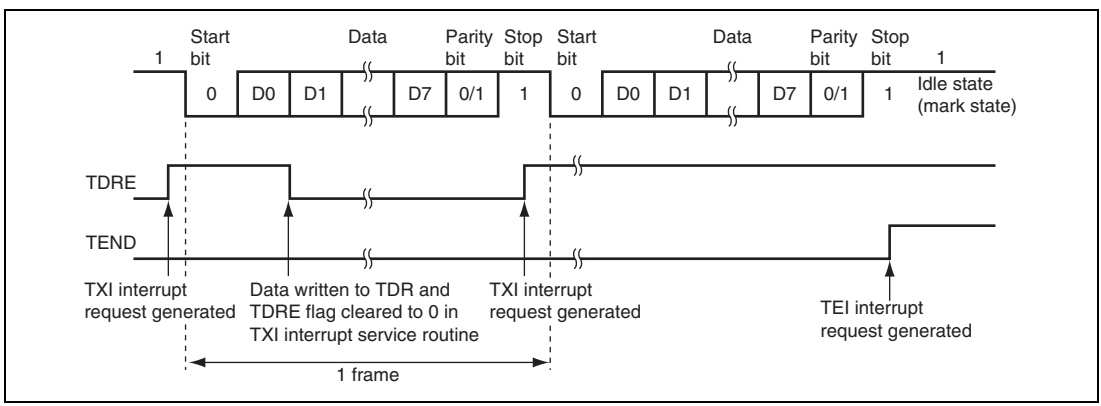
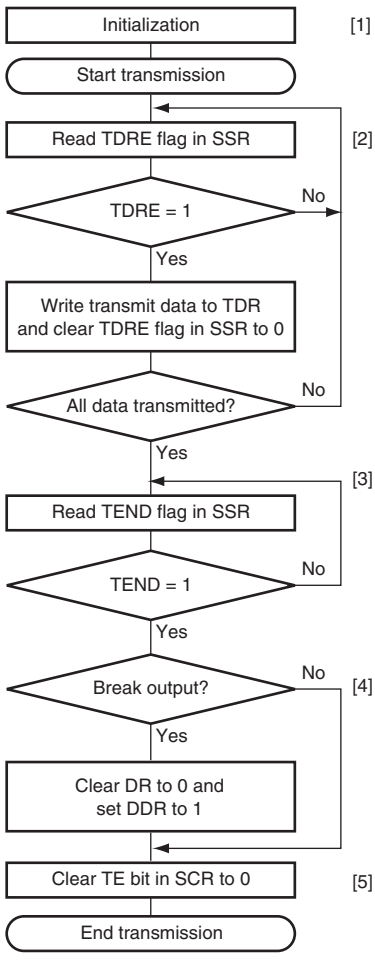


Figure 15.6 Example of Operation in Transmission in Asynchronous Mode (Example with 8-Bit Data, Parity, One Stop Bit)



[1] SCI initialization:
The TxD pin is automatically designated as the transmit data output pin.
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 and clear the TDRE flag to 0.

[3] Serial transmission continuation procedure:

To continue serial transmission, read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and clear the TDRE flag to 0.

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

Note: Do not write to SMR, SCR, BRR, and SDCR from the start to the end of transmission except the process of [5].

Figure 15.7 Sample Serial Transmission Flowchart

15.4.6 Serial Data Reception (Asynchronous Mode)

Figure 15.8 shows an example of the operation for reception in asynchronous mode. In serial reception, the SCI operates as described below.

1. The SCI monitors the communication line, and if a start bit is detected, performs internal synchronization, receives receive data in RSR, and checks the parity bit and stop bit.
2. If an overrun error (when reception of the next data is completed while the RDRF flag in SSR is still set to 1) occurs, the ORER bit in SSR is set to 1. If the RIE bit in SCR is set to 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to RDR. The RDRF 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 transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an ERI interrupt request is generated.
4. If a framing error (when the stop bit is 0) is detected, the FER 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 ERI interrupt 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 request is generated. Because the RXI interrupt routine reads the receive data transferred to RDR before reception of the next receive data has finished, continuous reception can be enabled.

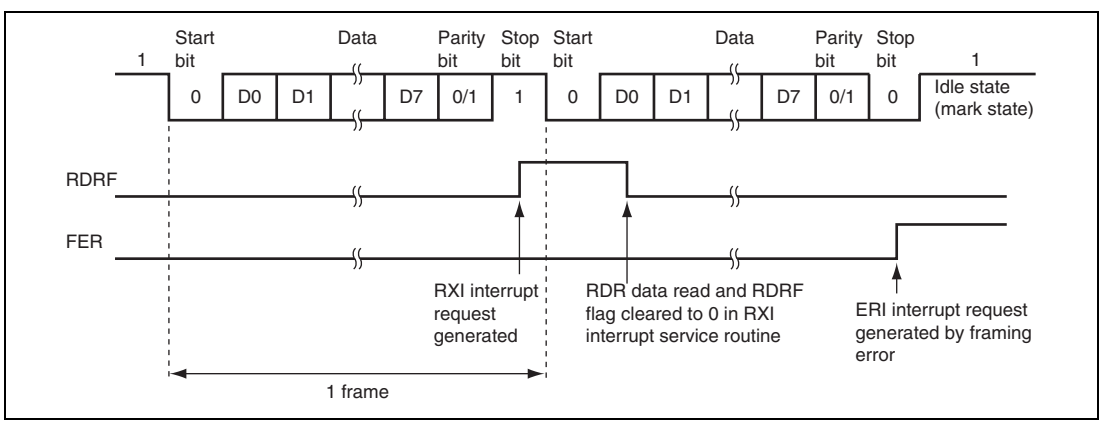


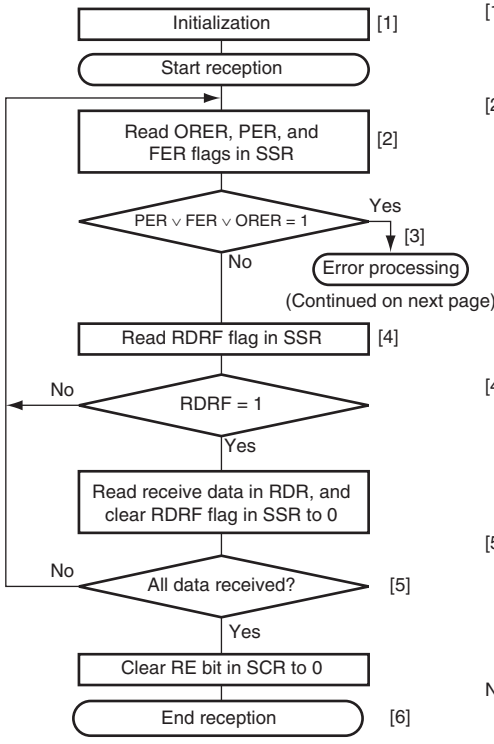
Figure 15.8 Example of SCI Operation in Reception (Example with 8-Bit Data, Parity, One Stop Bit)

Table 15.12 shows the states of the SSR status flags and receive data handling when a receive error is detected. If a receive error is detected, the RDRF flag retains its state before receiving data. Reception cannot be resumed while a receive error flag is set to 1. Accordingly, clear the ORER, FER, PER, and RDRF bits to 0 before resuming reception. Figure 15.9 shows a sample flowchart for serial data reception.

Table 15.12 SSR Status Flags and Receive Data Handling

SSR Status Flag					
RDRF*	**ORER**	**FER**	**PER**	**Receive Data**	**Receive Error Type**
1	1	0	0	Lost	Overrun error
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 + framing error
1	1	0	1	Lost	Overrun error + parity error
0	0	1	1	Transferred to RDR	Framing error + parity error
1	1	1	1	Lost	Overrun error + framing error + parity error

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



[1] SCI initialization:
The RxD pin is automatically designated as the receive data input pin.

[2] [3] Receive error processing and break detection:
If a receive error occurs, read the ORER, PER, and FER flags in SSR to identify the error. After performing the appropriate error processing, ensure that the ORER, PER, and FER flags are all cleared to 0. Reception cannot be resumed if any of these flags are set to 1. In the case of a framing error, a break can be detected by reading the value of the input port corresponding to the RxD pin.

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

[5] Serial reception continuation procedure:
To continue serial reception, before the stop bit for the current frame is received, read the RDRF flag, read RDR, and clear the RDRF flag to 0.

Note: Do not write to SMR, SCR, BRR, and SDCR from the start to the end of transmission except the process of [6].

[Legend]
∨: Logical add (OR)

Figure 15.9 Sample Serial Reception Flowchart (1)

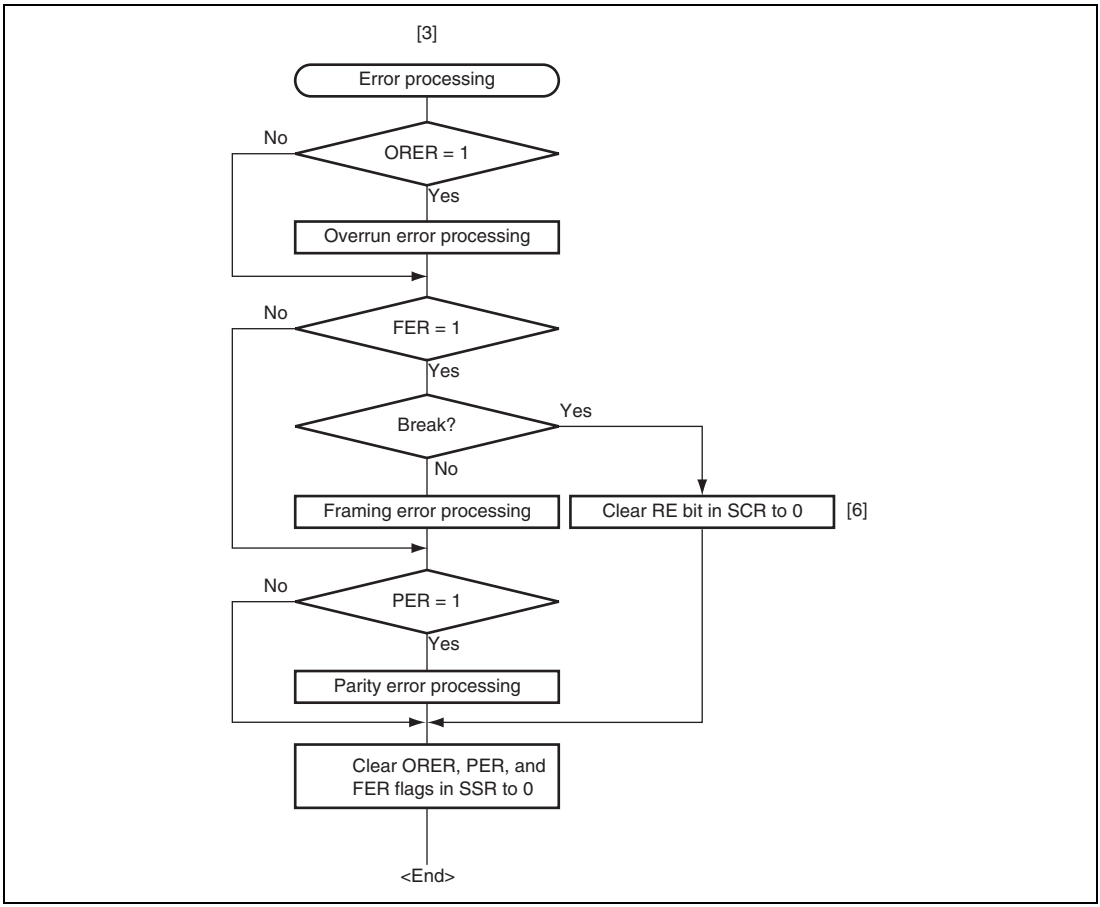


Figure 15.9 Sample Serial Reception Flowchart (2)

15.5 Multiprocessor Communication Function

Use of the multiprocessor communication function enables data transfer to be performed among a number of processors sharing communication lines by means of asynchronous serial communication using the multiprocessor format, in which a multiprocessor bit is added to the transfer data. When multiprocessor communication is carried out, each receiving station is addressed by a unique ID code. The serial communication cycle consists of two component cycles: an ID transmission cycle which specifies the receiving station, and a data transmission cycle for the specified receiving station. The multiprocessor bit is used to differentiate between the ID transmission cycle and the data transmission cycle. If the multiprocessor bit is 1, the cycle is an ID transmission cycle, and if the multiprocessor bit is 0, the cycle is a data transmission cycle. Figure 15.10 shows an example of inter-processor communication using the multiprocessor format. The transmitting station first sends the ID code of the receiving station with which it wants to perform serial communication as data with a 1 multiprocessor bit added. It then sends transmit data as data with a 0 multiprocessor bit added. When data with a 1 multiprocessor bit is received, the receiving station compares that data with its own ID. The station whose ID matches then receives the data 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 to 1, transfer of receive data from RSR to RDR, error flag detection, and setting the RDRF, FER, and ORER status flags in SSR to 1 are prohibited until data with a 1 multiprocessor bit is received. On reception of a receive character with a 1 multiprocessor bit, the MPB bit in SSR is set to 1 and the MPIE bit is automatically cleared, thus normal reception is resumed. If the RIE bit in SCR is set to 1 at this time, an RXI interrupt is generated.

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

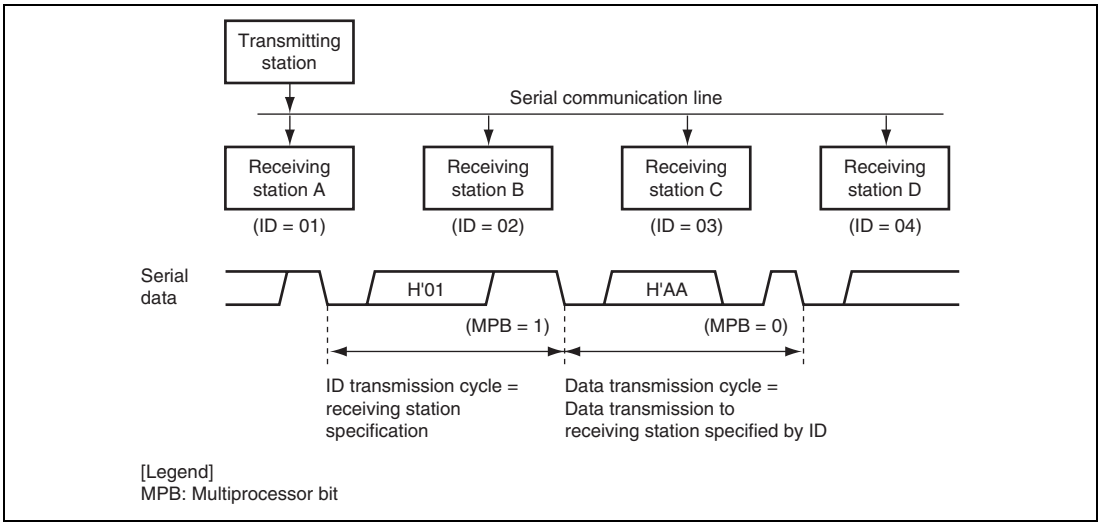
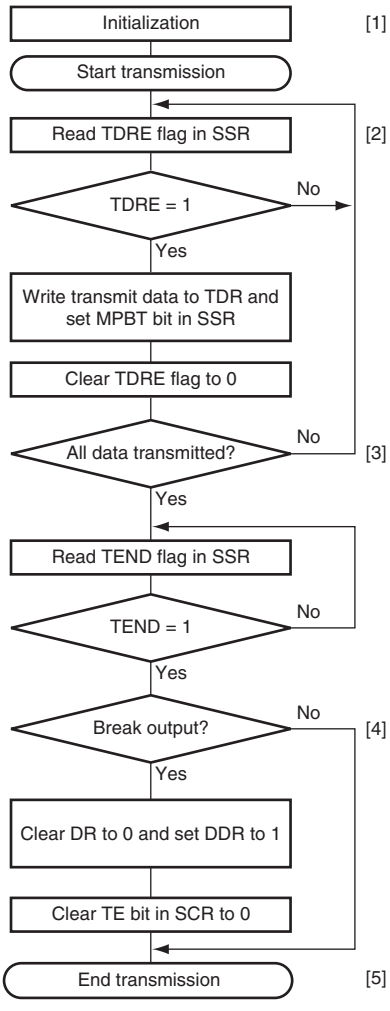


Figure 15.10 Example of Communication Using Multiprocessor Format (Transmission of Data H'AA to Receiving Station A)

15.5.1 Multiprocessor Serial Data Transmission

Figure 15.11 shows a sample flowchart for multiprocessor serial data transmission. For an ID transmission cycle, set the MPBT bit in SSR to 1 before transmission. For a data transmission cycle, clear the MPBT bit in SSR to 0 before transmission. All other SCI operations are the same as those in asynchronous mode.



[1] SCI initialization:
The TxD pin is automatically designated as the transmit data output pin.
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.

[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.

Note: Do not write to SMR, SCR, BRR, and SDCR from the start to the end of transmission except the process of [5].

Figure 15.11 Sample Multiprocessor Serial Transmission Flowchart

15.5.2 Multiprocessor Serial Data Reception

Figure 15.13 shows a sample flowchart for multiprocessor serial data reception. If the MPIE bit in SCR is set to 1, data is skipped until data with a 1 multiprocessor bit is sent. On receiving data with a 1 multiprocessor bit, the receive data is transferred to RDR. An RXI interrupt request is generated at this time. All other SCI operations are the same as in asynchronous mode. Figure 15.12 shows an example of SCI operation for multiprocessor format reception.

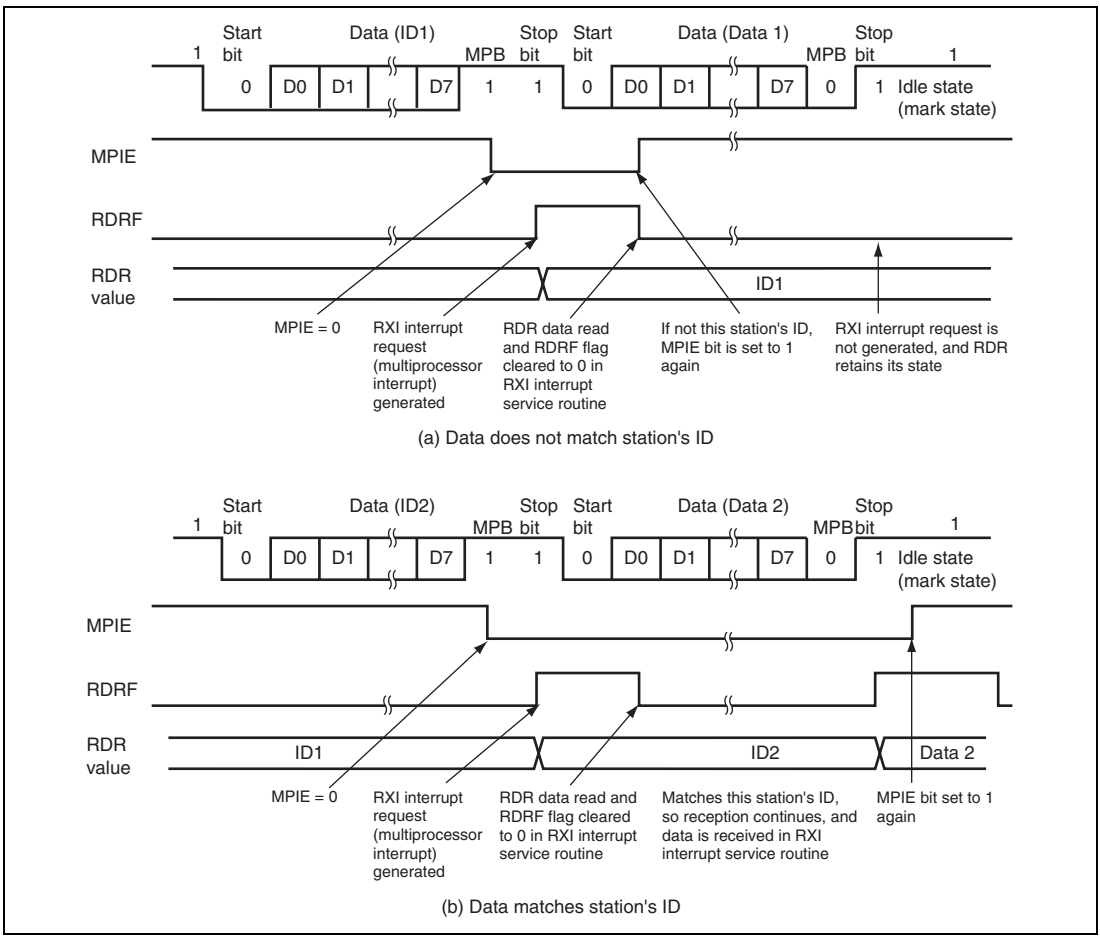


Figure 15.12 Example of SCI Operation in Reception (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)

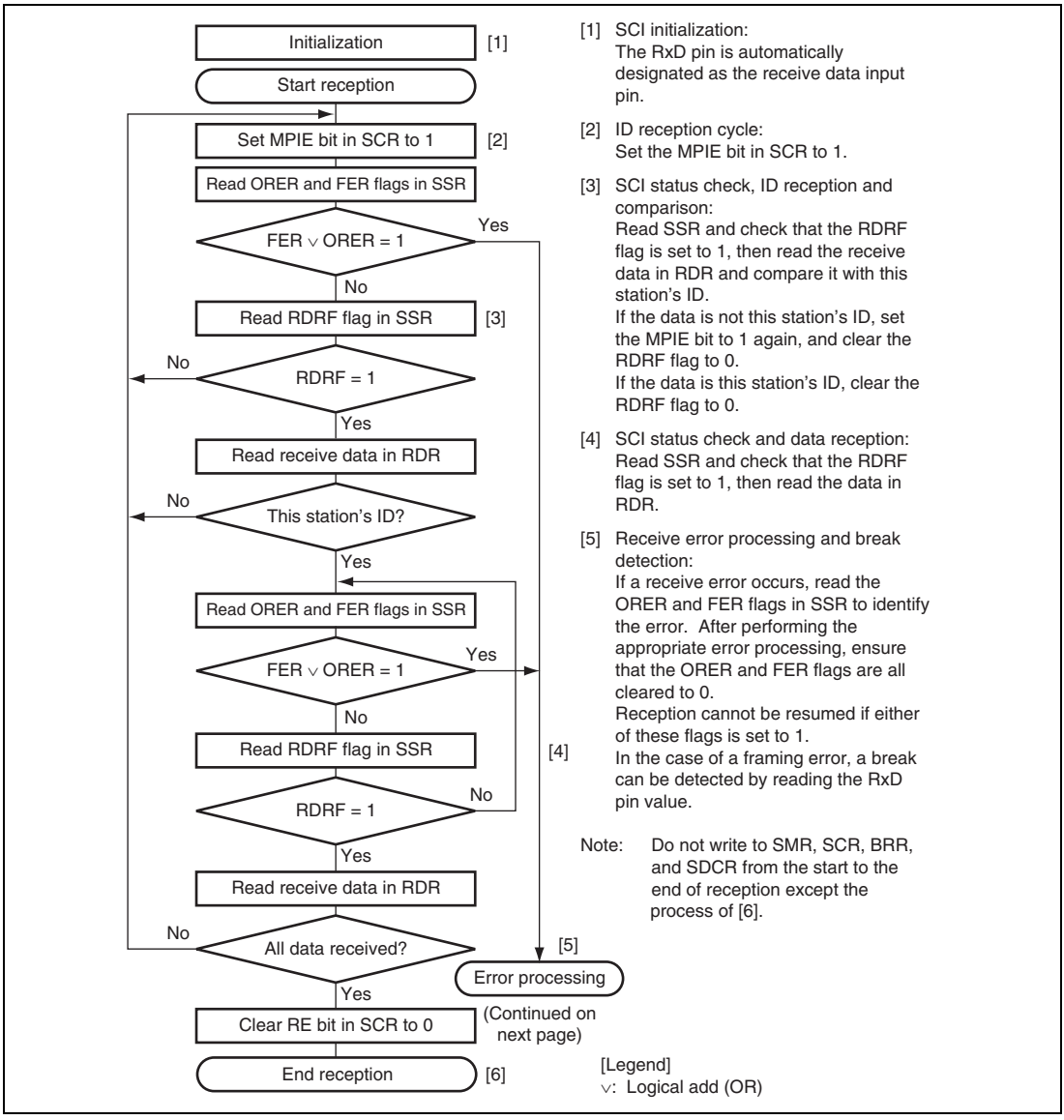


Figure 15.13 Sample Multiprocessor Serial Reception Flowchart (1)

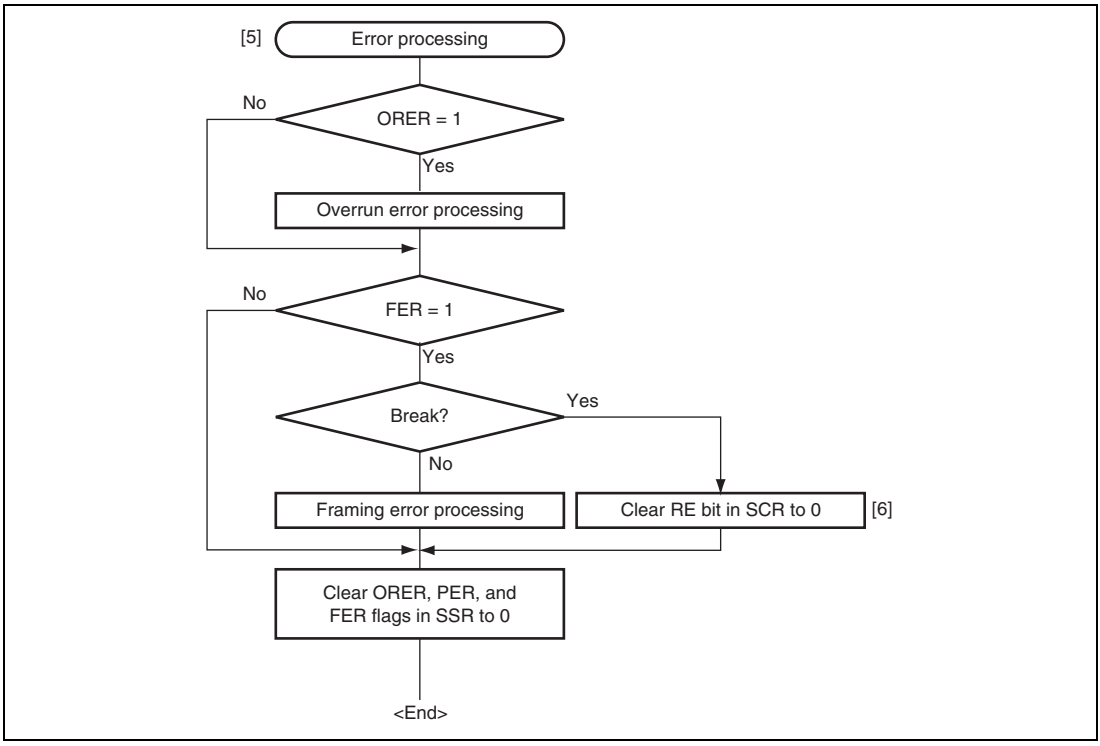


Figure 15.13 Sample Multiprocessor Serial Reception Flowchart (2)

15.6 Operation in Clocked Synchronous Mode

Figure 15.14 shows the general format for clocked synchronous communication. In clocked synchronous mode, data is transmitted or received in synchronization with clock pulses. One character in transfer data consists of 8-bit data. In data transmission, the SCI outputs data from one falling edge of the synchronization clock to the next. In data reception, the SCI receives data in synchronization with the rising edge of the synchronization clock. After 8-bit data is output, the transmission line holds the MSB state. In clocked synchronous mode, no parity or multiprocessor bit is added. Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communication by use of a common clock. Both the transmitter and the receiver also have a double-buffered structure, so that the next transmit data can be written during transmission or the previous receive data can be read during reception, enabling continuous data transfer.

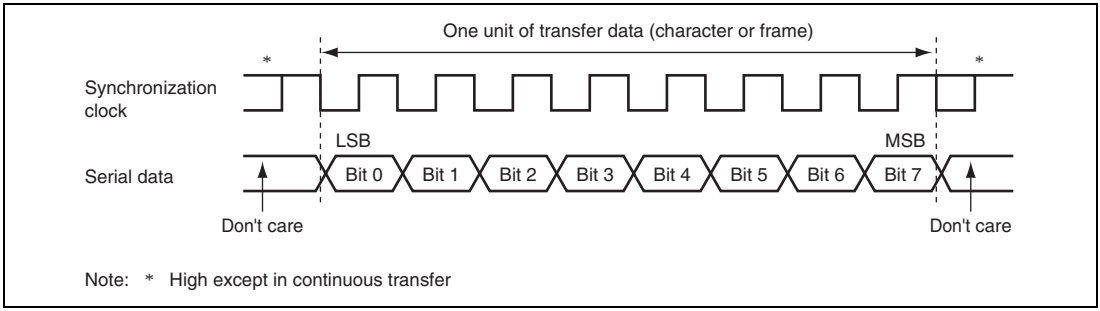


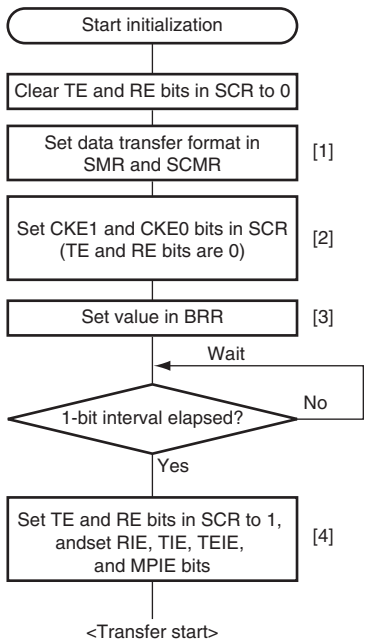
Figure 15.14 Data Format in Synchronous Communication (LSB-First)

15.6.1 Clock

Either an internal clock generated by the on-chip baud rate generator or an external synchronization clock input at the SCK pin can be selected, according to the setting of the CKE1 and CKE0 bits in SCR. When the SCI is operated on an internal clock, the synchronization clock is output from the SCK pin. Eight synchronization clock pulses are output in the transfer of one character, and when no transfer is performed the clock is fixed high.

15.6.2 SCI Initialization (Clocked Synchronous Mode)

Before transmitting and receiving data, you should first clear the TE and RE bits in SCR to 0, then initialize the SCI as described in a sample flowchart in figure 15.15. When the operating mode, transfer format, etc., is changed, the TE and RE bits must be cleared to 0 before making the change using the following procedure. When the TE bit is cleared to 0, the TDRE flag in SSR is set to 1. However, clearing the RE bit to 0 does not initialize the RDRF, PER, FER, and ORER flags in SSR, or RDR.



[1] Set the data transfer format in SMR and SCMR.

[2] Set the clock selection in SCR. Be sure to clear bits RIE, TIE, TEIE, MPIE, TE, and RE to 0.

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

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

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

Figure 15.15 Sample SCI Initialization Flowchart

15.6.3 Serial Data Transmission (Clocked Synchronous Mode)

Figure 15.16 shows an example of SCI operation for transmission in clocked synchronous mode. In serial transmission, the SCI operates as described below.

1. The SCI monitors the TDRE flag in SSR, and if it is 0, recognizes that data has been written to TDR, and transfers the data from TDR to TSR.
2. After transferring data from TDR to TSR, the SCI sets the TDRE flag to 1 and starts transmission. If the TIE bit in SCR is set to 1 at this time, a TXI interrupt request is generated. Because the TXI interrupt routine writes the next transmit data to TDR before transmission of the current transmit data has finished, continuous transmission can be enabled.
3. 8-bit data is sent from the TxD pin synchronized with the output clock when output clock mode has been specified and synchronized with the input clock when use of an external clock 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 transmission 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 maintains the output state of the last bit. If the TEIE bit in SCR is set to 1 at this time, a TEI interrupt request is generated. The SCK pin is fixed high.

Figure 15.17 shows a sample flowchart for serial data transmission. Even if the TDRE flag is cleared to 0, transmission will not start while a receive error flag (ORER, FER, or PER) is set to 1. Make sure to clear the receive error flags to 0 before starting transmission. Note that clearing the RE bit to 0 does not clear the receive error flags.

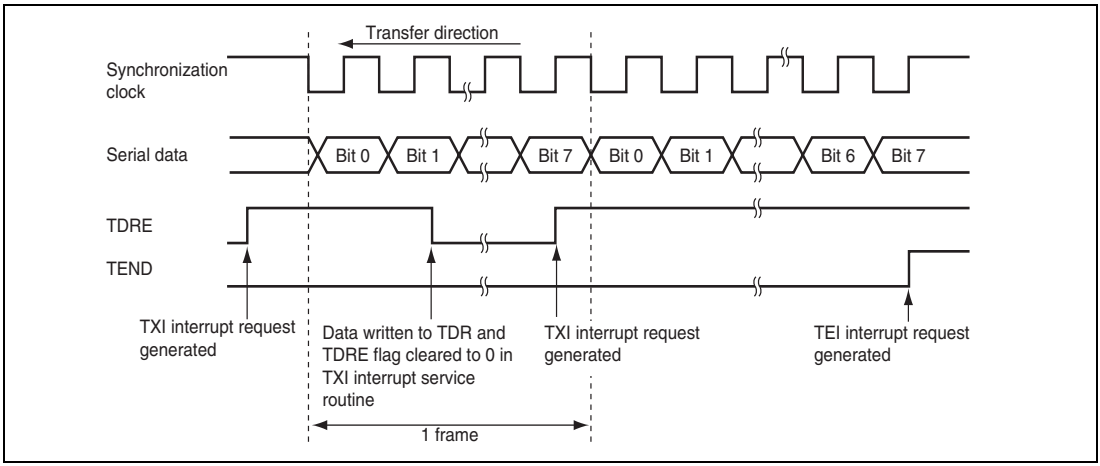
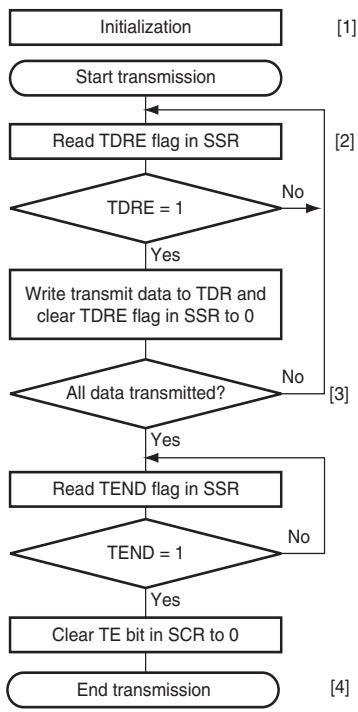


Figure 15.16 Sample SCI Transmission Operation in Clocked Synchronous Mode



[1] SCI initialization:
The TxD pin is automatically designated as the transmit data output pin.

[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 and 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.

Note: Do not write to SMR, SCR, BRR, and SDCR from the start to the end of transmission except the process of [4].

Figure 15.17 Sample Serial Transmission Flowchart

15.6.4 Serial Data Reception (Clocked Synchronous Mode)

Figure 15.18 shows an example of SCI operation for reception in clocked synchronous mode. In serial reception, the SCI operates as described below.

1. The SCI performs internal initialization in synchronization with a synchronization clock input or output, starts receiving data, and stores the receive data in RSR.
2. If an overrun error (when reception of the next data is completed while the RDRF flag is still set to 1) occurs, the ORER bit in SSR is set to 1. If the RIE bit in SCR is set to 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to RDR. The RDRF flag remains to be set to 1.
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 request is generated. Because the RXI interrupt routine reads the receive data transferred to RDR before reception of the next receive data has finished, continuous reception can be enabled.

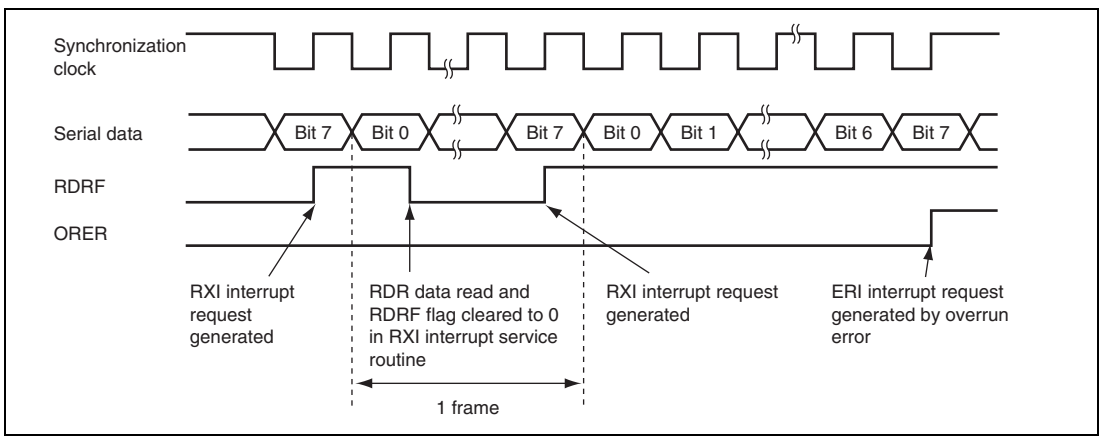
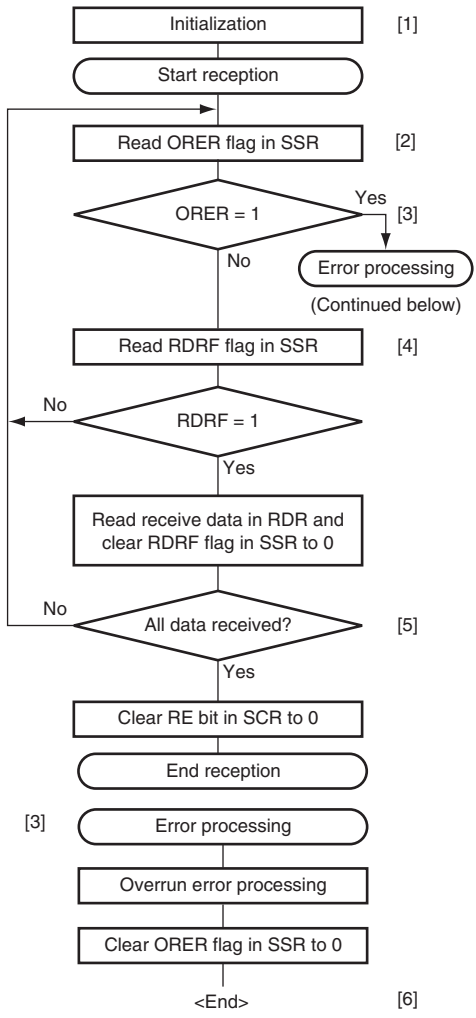


Figure 15.18 Example of SCI Receive Operation in Clocked Synchronous Mode

Reception cannot be resumed while a receive error flag is set to 1. Accordingly, clear the ORER, FER, PER, and RDRF bits to 0 before resuming reception. Figure 15.19 shows a sample flowchart for serial data reception.



[1] SCI initialization:
The RxD pin is automatically designated as the receive data input pin.

[2] [3] Receive error processing:
If a receive error occurs, read the ORER flag in SSR, and after performing the appropriate error processing, clear the ORER flag to 0. Transfer cannot be resumed if the ORER flag is set to 1.

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

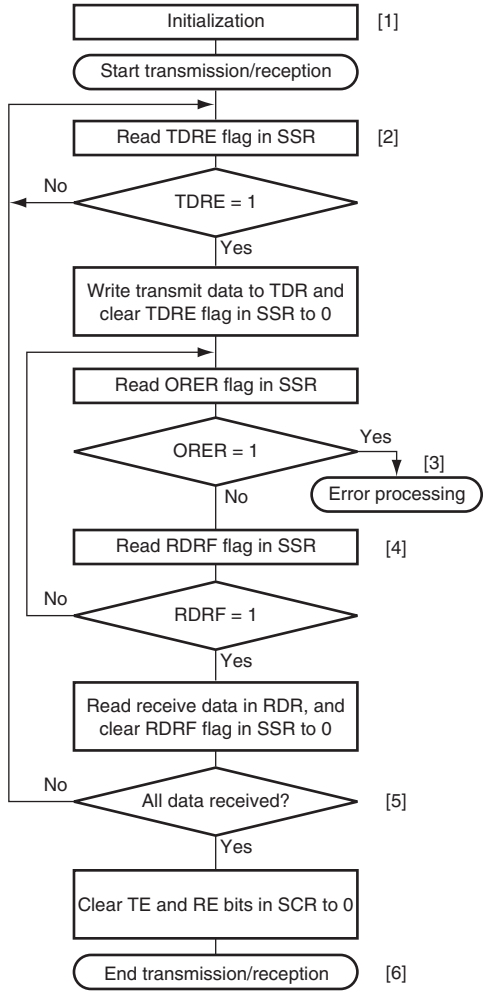
[5] Serial reception continuation procedure:
To continue serial reception, before the MSB (bit 7) of the current frame is received, reading the RDRF flag, reading RDR, and clearing the RDRF flag to 0 should be finished.

Note: Do not write to SMR, SCR, BRR, and SDCR from the start to the end of reception except the process of [6].

Figure 15.19 Sample Serial Reception Flowchart

15.6.5 Simultaneous Serial Data Transmission and Reception (Clocked Synchronous Mode)

Figure 15.20 shows a sample flowchart for simultaneous serial transmit and receive operations. After initializing the SCI, the following procedure should be used for simultaneous serial data transmit and receive operations. To switch from transmit mode to simultaneous transmit and receive mode, after checking that the SCI has finished transmission and the TDRE and TEND flags in SSR are set to 1, clear the TE bit in SCR to 0. Then simultaneously set the TE and RE bits to 1 with a single instruction. To switch from receive mode to simultaneous transmit and receive mode, after checking that the SCI has finished reception, clear the RE bit to 0. Then after checking that the RDRF bit in SSR and receive error flags (ORER, FER, and PER) are cleared to 0, simultaneously set the TE and RE bits to 1 with a single instruction.



[1] SCI initialization:
The TxD pin is designated as the transmit data output pin, and the RxD pin is designated as the receive data input pin, enabling simultaneous transmit and receive operations.

[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 and clear the TDRE flag to 0. Transition of the TDRE flag from 0 to 1 can also be identified by a TXI interrupt.

[3] Receive error processing:
If a receive error occurs, read the ORER flag in SSR, and after performing the appropriate error processing, clear the ORER flag to 0. Transmission/reception cannot be resumed if the ORER flag is set to 1.

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

[5] Serial transmission/reception continuation procedure:
To continue serial transmission/reception, before the MSB (bit 7) of the current frame is received, finish reading the RDRF flag, reading RDR, and clearing the RDRF flag to 0. Also, before the MSB (bit 7) of the current frame is transmitted, read 1 from the TDRE flag to confirm that writing is possible. Then write data to TDR and clear the TDRE flag to 0.

Notes: 1. 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.
2. Do not write to SMR, SCR, BRR, and SDCR from the start to the end of transmission/reception except the process of [6].

Figure 15.20 Sample Flowchart of Simultaneous Serial Transmission and Reception

15.7 Smart Card Interface Description

The SCI supports the IC card (smart card) interface based on the ISO/IEC 7816-3 (Identification Card) standard as an enhanced serial communication interface function. Smart card interface mode can be selected using the appropriate register.

15.7.1 Sample Connection

Figure 15.21 shows a sample connection between the smart card and this LSI. As in the figure, since this LSI communicates with the IC card using a single transmission line, interconnect the TxD and RxD pins and pull up the data transmission line to VCC using a resistor. Setting the RE and TE bits in SCR to 1 with the IC card not connected enables closed transmission/reception allowing self diagnosis. To supply the IC card with the clock pulses generated by the SCI, input the SCK pin output to the CLK pin of the IC card. A reset signal can be supplied via the output port of this LSI.

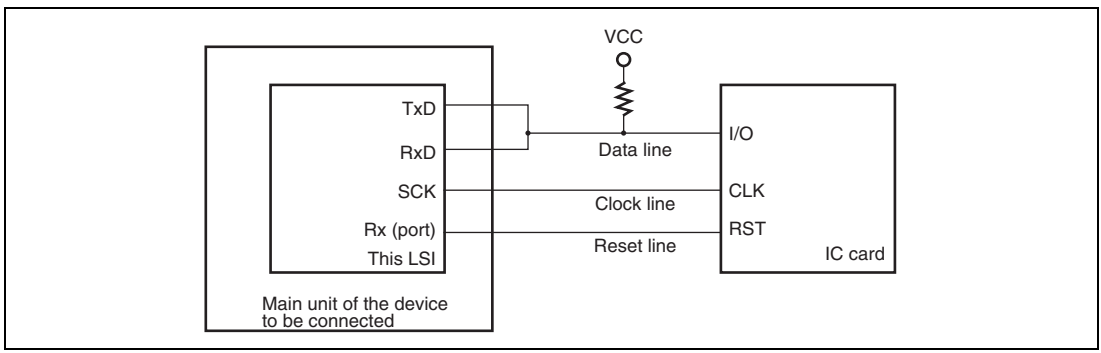


Figure 15.21 Pin Connection for Smart Card Interface

15.7.2 Data Format (Except in Block Transfer Mode)

Figure 15.22 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 transferring one bit) is secured as a guard time after the end of the parity bit before the start of the next frame.
- If a parity error is detected during reception, a low error signal is output for 1 etu after 10.5 etu has passed from the start bit.
- If an error signal is sampled during transmission, the same data is automatically re-transmitted after two or more etu.

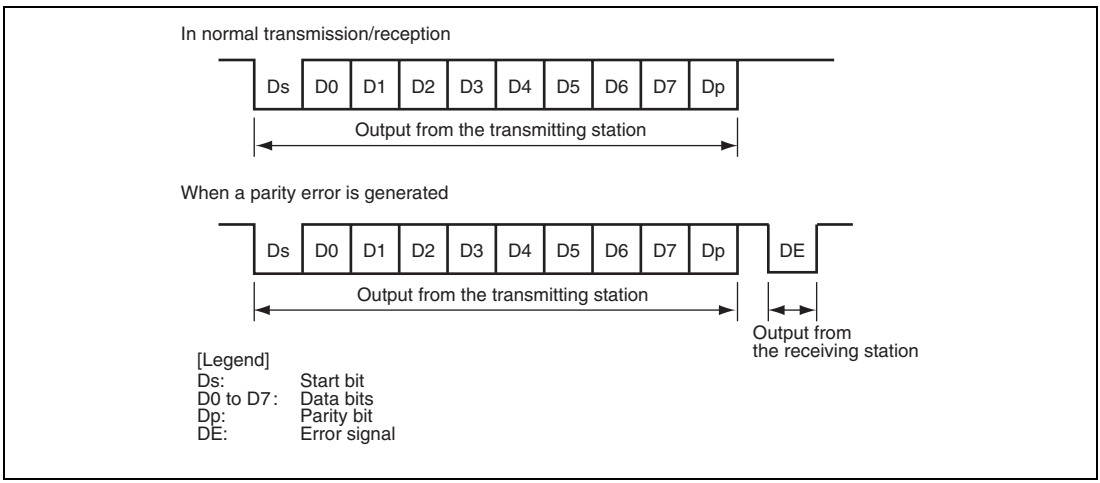


Figure 15.22 Data Formats in Normal Smart Card Interface Mode

For communication with the IC cards of the direct convention and inverse convention types, follow the procedure below.

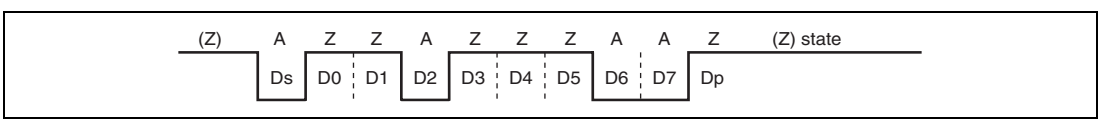


Figure 15.23 Direct Convention (SDIR = SINV = O/$\overline{E}$ = 0)

For the direct convention type, logic levels 1 and 0 correspond to states Z and A, respectively, and data is transferred with LSB-first as the start character, as shown in figure 15.23. Therefore, data in the start character in the figure is H'3B. When using the direct convention type, write 0 to both the SDIR and SINV bits in SCMR. Write 0 to the O/$\overline{E}$ bit in SMR in order to use even parity, which is prescribed by the smart card standard.

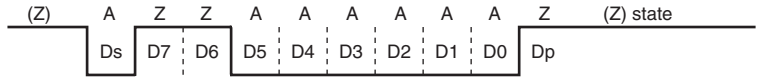


Figure 15.24 Inverse Convention (SDIR = SINV = O/$\overline{E}$ = 1)

For the inverse convention type, logic levels 1 and 0 correspond to states A and Z, respectively and data is transferred with MSB-first as the start character, as shown in figure 15.24. Therefore, data in the start character in the figure is H'3F. When using the inverse convention type, write 1 to both the SDIR and SINV bits in SCMR. The parity bit is logic level 0 to produce even parity, which is prescribed by the smart card standard, and corresponds to state Z. Since the SINV bit of this LSI only inverts data bits D7 to D0, write 1 to the O/$\overline{E}$ bit in SMR to invert the parity bit in both transmission and reception.

15.7.3 Block Transfer Mode

Block transfer mode is different from normal smart card interface mode in the following respects.

- If a parity error is detected during reception, no error signal is output. Since the PER bit in SSR is set by error detection, clear the bit before receiving the parity bit of the next frame.
- During transmission, at least 1 etu is secured as a guard time after the end of the parity bit before the start of the next frame.
- Since the same data is not re-transmitted during transmission, the TEND flag in SSR is set 11.5 etu after transmission start.
- Although the ERS flag in block transfer mode displays the error signal status as in normal smart card interface mode, the flag is always read as 0 because no error signal is transferred.

15.7.4 Receive Data Sampling Timing and Reception Margin

Only the internal clock generated by the internal baud rate generator can be used as a communication clock in smart card interface mode. In this mode, the SCI can operate using a basic clock with a frequency of 32, 64, 372, or 256 times the bit rate according to the BCP1 and BCP0 settings (the frequency is always 16 times the bit rate in normal asynchronous mode). At reception, the falling edge of the start bit is sampled using the internal basic clock in order to perform internal synchronization. Receive data is sampled at the 16th, 32nd, 186th and 128th rising edges of the basic clock pulses so that it can be latched at the center of each bit as shown in figure 15.25. The reception margin here is determined by the following formula.

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

M: Reception margin (%)
N: Ratio of bit rate to clock (N = 32, 64, 372, 256)
D: Clock duty (D = 0 to 1.0)
L: Frame length (L = 10)
F: Absolute value of clock rate deviation

Assuming values of F = 0, D = 0.5, and N = 372 in formula (1), the reception margin is determined by the formula below.

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

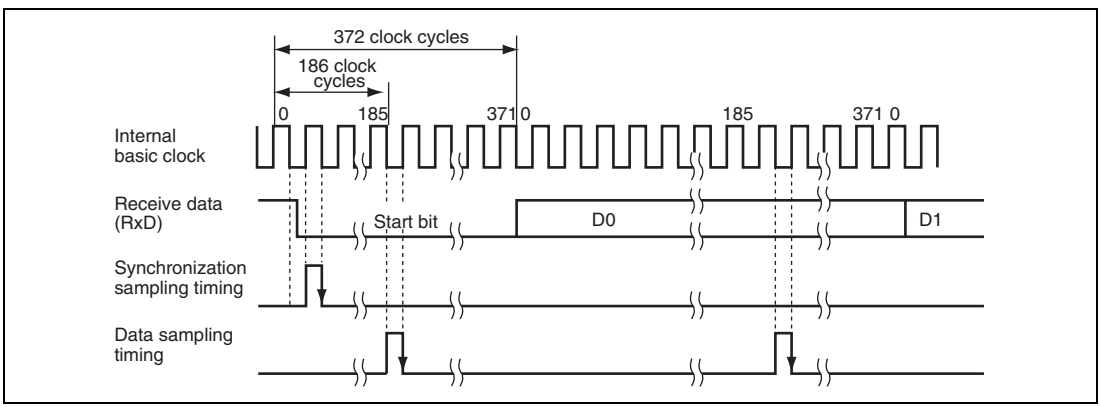


Figure 15.25 Receive Data Sampling Timing in Smart Card Interface Mode (When Clock Frequency is 372 Times the Bit Rate)

15.7.5 Initialization

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

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. Also set the PE bit to 1.
4. Set the SMIF, SDIR, and SINV bits in SCMR appropriately. When the SMIF bit is set to 1, the 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, MPIE, and TEIE bits to 0 simultaneously. When the CKE0 bit is set to 1, the SCK pin is allowed to output clock pulses.
7. Set the TIE, RIE, TE, and RE bits in SCR appropriately after waiting for at least 1 bit interval. 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 initialize the SCI. At the end of initialization, RE and TE should be set to 0 and 1, respectively. Reception completion can be verified by reading the RDRF flag or PER and ORER flags. To switch from transmission to reception, first verify that transmission has completed, and initialize the SCI. At the end of initialization, TE and RE should be set to 0 and 1, respectively. Transmission completion can be verified by reading the TEND flag.

15.7.6 Serial Data Transmission (Except in Block Transfer Mode)

Data transmission in smart card interface mode (except in block transfer mode) is different from that in normal serial communication interface mode in that an error signal is sampled and data is re-transmitted. Figure 15.26 shows the data re-transfer operation during transmission.

1. If an error signal from the receiving end is sampled after one frame of data has been transmitted, the ERS bit in SSR is set to 1. Here, an ERI interrupt request is generated if the RIE bit in SCR is set to 1. Clear the ERS bit to 0 before the next parity bit is sampled.
2. For the frame in which an error signal is received, the TEND bit in SSR is not set to 1. Data is re-transferred from TDR to TSR allowing automatic data retransmission.

3. If no error signal is returned from the receiving end, the ERS bit in SSR is not set to 1. In this case, one frame of data is determined to have been transmitted including re-transfer, and the TEND bit in SSR is set to 1. Here, a TXI interrupt request is generated if the TIE bit in SCR is set to 1. Writing transmit data to TDR starts transmission of the next data.

Figure 15.28 shows a sample flowchart for transmission. In transmission, the TEND and TDRE flags in SSR are simultaneously set to 1, thus generating a TXI interrupt request when TIE in SCR is set. If an error occurs, the SCI automatically re-transmits the same data. During re-transmission, TEND remains 0. Therefore, the SCI automatically transmit the specified number of bytes, including re-transmission in the case of error. However, the ERS flag is not automatically cleared; the ERS flag must be cleared by previously setting the RIE bit to 1 to enable an ERI interrupt request to be generated at error occurrence.

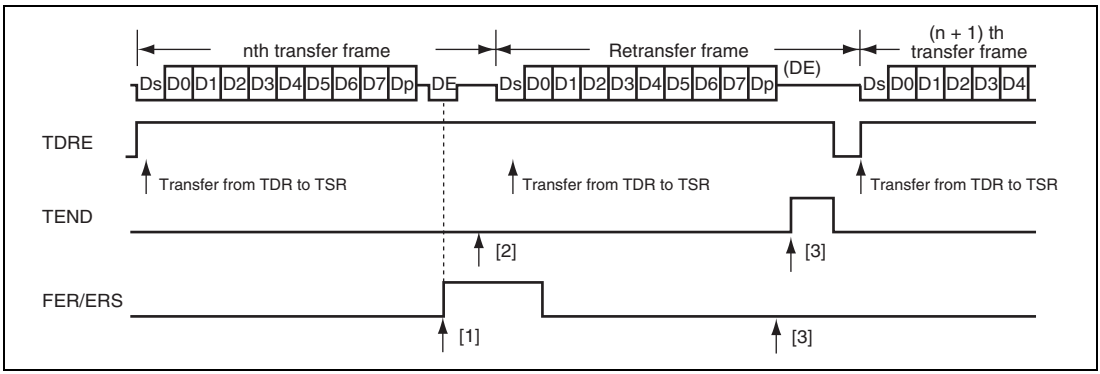


Figure 15.26 Data Re-transfer Operation in SCI Transmission Mode

Note that the TEND flag is set in different timings depending on the GM bit setting in SMR, which is shown in figure 15.27.

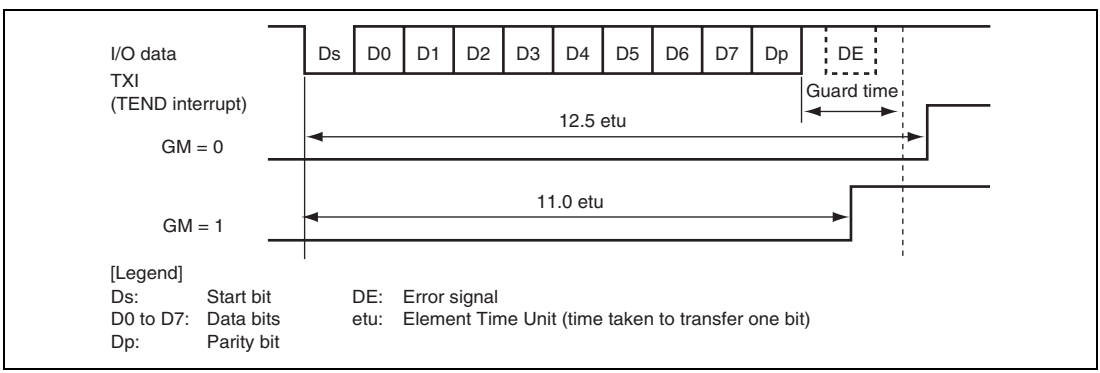


Figure 15.27 TEND Flag Set Timings during Transmission

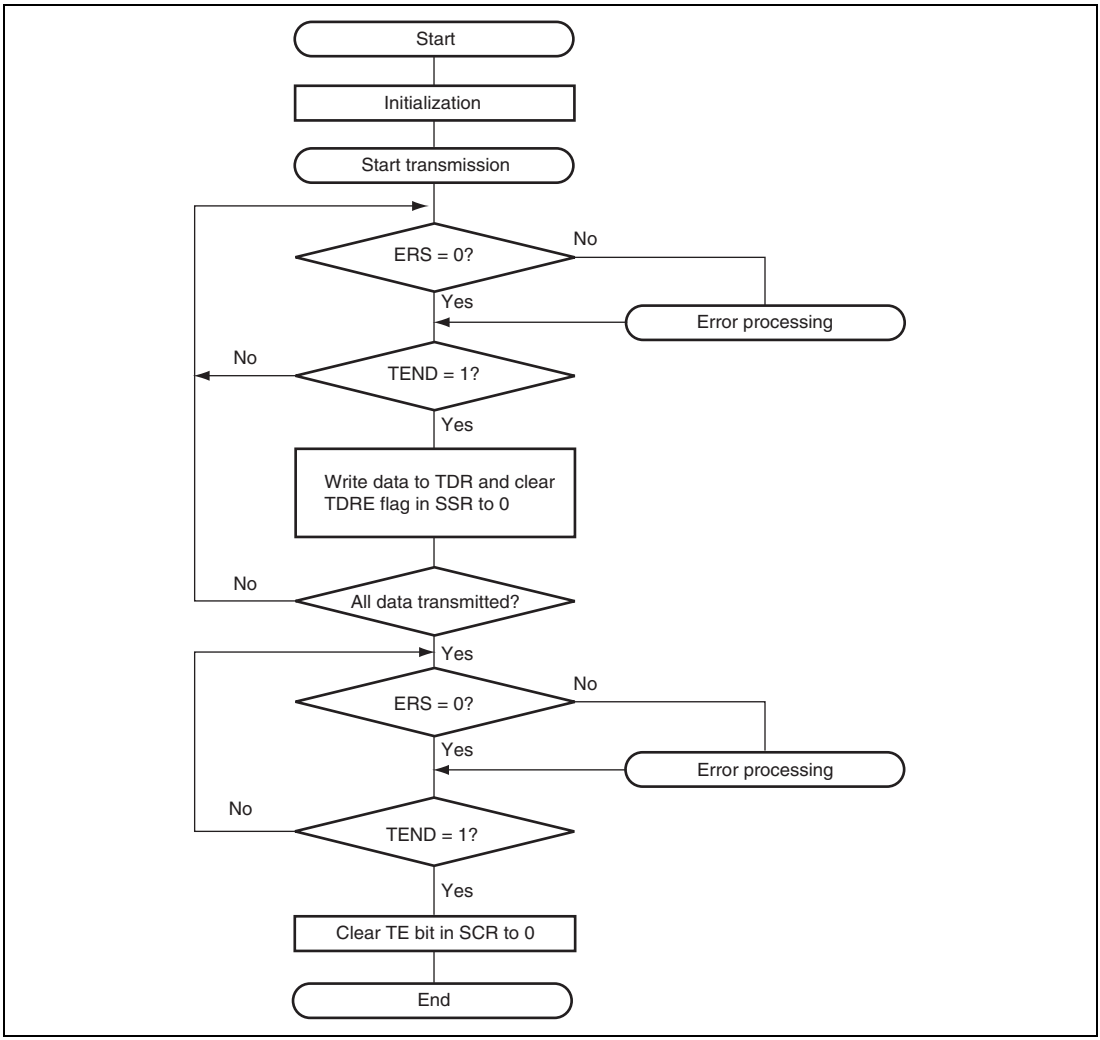


Figure 15.28 Sample Transmission Flowchart

15.7.7 Serial Data Reception (Except in Block Transfer Mode)

Data reception in smart card interface mode is identical to that in normal serial communication interface mode. Figure 15.29 shows the data re-transfer operation during reception.

1. If a parity error is detected in receive data, the PER bit in SSR is set to 1. Here, an ERI interrupt request is generated if the RIE bit in SCR is set to 1. Clear the PER bit to 0 before the next parity bit is sampled.
2. For the frame in which a parity error is detected, the RDRF bit in SSR is not set to 1.
3. If no parity error is detected, the PER bit in SSR is not set to 1. In this case, data is determined to have been received successfully, and the RDRF bit in SSR is set to 1. Here, an RXI interrupt request is generated if the RIE bit in SCR is set.

Figure 15.30 shows a sample flowchart for reception. In reception, setting the RIE bit to 1 allows an RXI interrupt request to be generated when the RDRF flag is set to 1. If an error occurs during reception, i.e., either the ORER or PER flag is set to 1, a transmit/receive error interrupt (ERI) request is generated and the error flag must be cleared. Even if a parity error occurs and PER is 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 15.4, Operation in Asynchronous Mode.

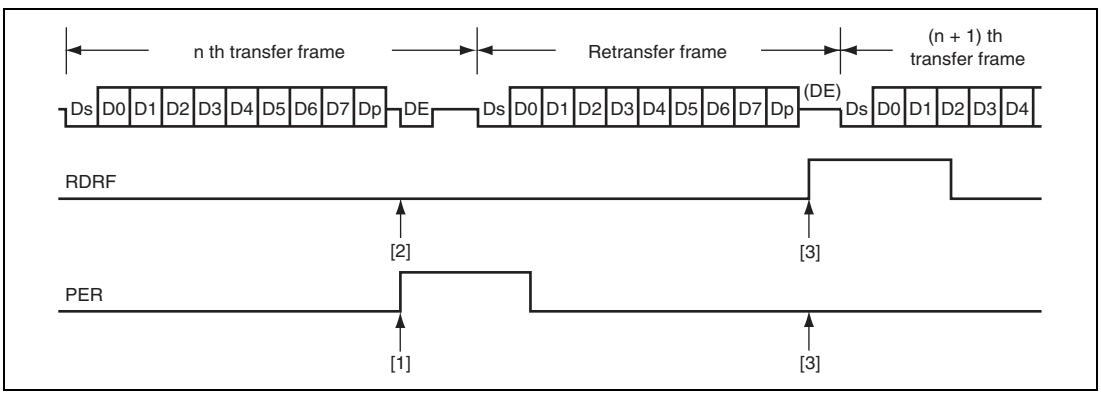


Figure 15.29 Data Re-transfer Operation in SCI Reception Mode

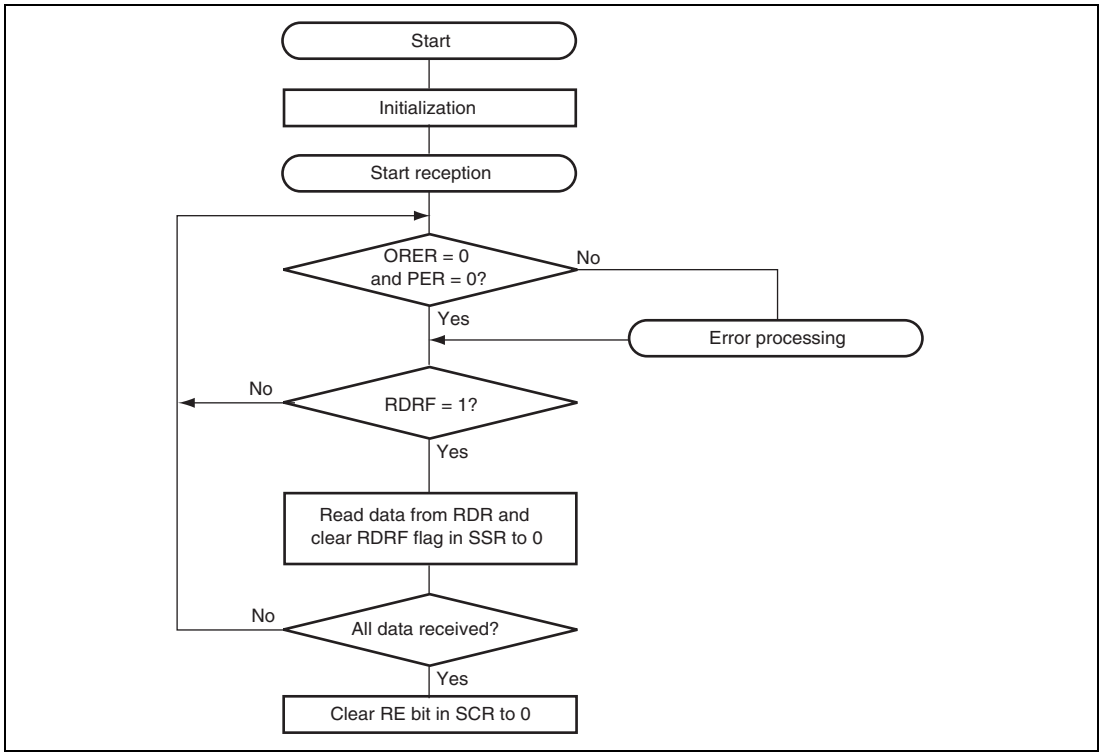


Figure 15.30 Sample Reception Flowchart

15.7.8 Clock Output Control

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

Figure 15.31 shows an example of clock output fixing timing when the CKE0 bit is controlled with GM = 1 and CKE1 = 0.

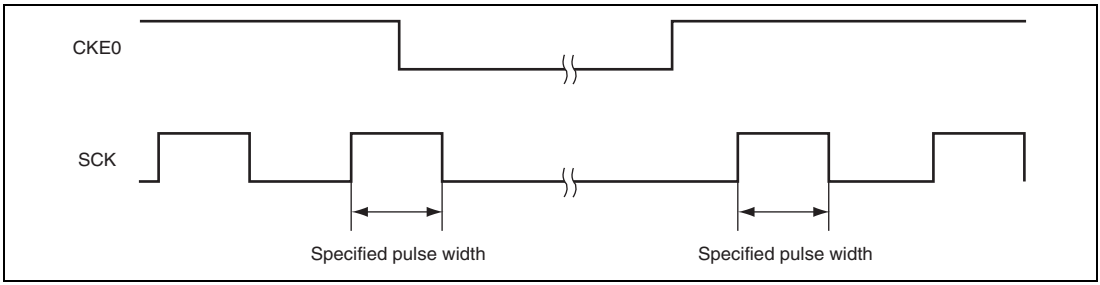


Figure 15.31 Clock Output Fixing Timing

At power-on and transitions to/from software standby mode, use the following procedure to secure the appropriate clock duty ratio.

(1) At Power-On

To secure the appropriate clock duty ratio simultaneously with power-on, use the following procedure.

1. Initially, port input is enabled in the high-impedance state. To fix the potential level, use a 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.

(2) 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 SCK 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, set the 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 the specified level with the duty ratio retained.
5. Make the transition to software standby mode.

(3) 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 appropriate duty ratio is then generated.

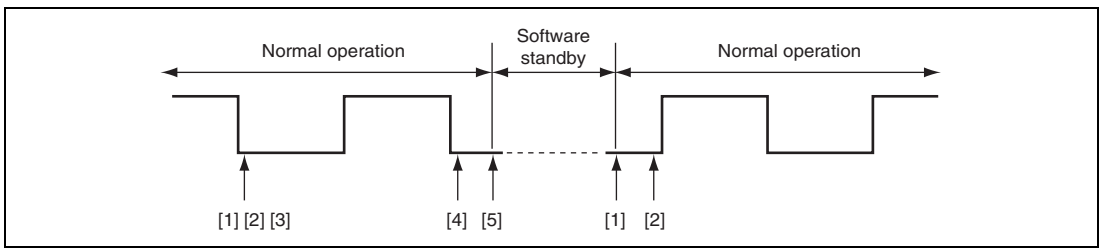


Figure 15.32 Clock Stop and Restart Procedure

15.8 Interrupt Sources

15.8.1 Interrupts in Normal Serial Communication Interface Mode

Table 15.13 shows the interrupt sources in normal serial communication interface mode. A different interrupt vector is assigned to each interrupt source, and individual interrupt sources can 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 TEND flag in SSR is set to 1, a TEI interrupt request is generated.

When the RDRF flag in SSR is set to 1, an RXI interrupt request is generated. When the ORER, PER, or FER flag in SSR is set to 1, an ERI interrupt request is generated.

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

Table 15.13 SCI Interrupt Sources

Channel	Name	Interrupt Source	Interrupt Flag	Priority
1	ERI1	Receive error	ORER, FER, PER	High
	RXI1	Receive data full	RDRF	
	TXI1	Transmit data empty	TDRE	
	TEI1	Transmit end	TEND	
2	ERI2	Receive error	ORER, FER, PER	
	RXI2	Receive data full	RDRF	
	TXI2	Transmit data empty	TDRE	
	TEI2	Transmit end	TEND	Low

15.8.2 Interrupts in Smart Card Interface Mode

Table 15.14 shows the interrupt sources in smart card interface mode. A TEI interrupt request cannot be used in this mode.

Table 15.14 SCI Interrupt Sources

Channel	Name	Interrupt Source	Interrupt Flag	Priority
1	ERI1	Receive error, error signal detection	ORER, PER, ERS	High
	RXI1	Receive data full	RDRF	
	TXI1	Transmit data empty	TEND	
2	ERI2	Receive error, error signal detection	ORER, PER, ERS	
	RXI2	Receive data full	RDRF	
	TXI2	Transmit data empty	TEND	Low

In transmission, the TEND and TDRE flags in SSR are simultaneously set to 1, thus generating a TXI interrupt request. If an error occurs, the SCI automatically re-transmits the same data. During re-transmission, the TEND flag remains 0. Therefore, the SCI automatically transmits the specified number of bytes, including re-transmission in the case of error. However, the ERS flag in SSR, which is set at error occurrence, is not automatically cleared; the ERS flag must be cleared by previously setting the RIE bit in SCR to 1 to enable an ERI interrupt request to be generated at error occurrence.

In reception, an RXI interrupt request is generated when the RDRF flag in SSR is set to 1. If an error occurs, the RDRF flag is not set but the error flag is set. Therefore, an ERI interrupt request is issued to the CPU instead; the error flag must be cleared.

15.9 Usage Notes

15.9.1 Module Stop Mode Setting

SCI operation can be disabled or enabled using the module stop control register. The initial setting is for SCI operation to be halted. Register access is enabled by clearing module stop mode. For details, see section 26, Power-Down Modes.

15.9.2 Break Detection and Processing

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

15.9.3 Mark State and Break Sending

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

15.9.4 Receive Error Flags and Transmit Operations (Clocked Synchronous Mode Only)

Transmission cannot be started when a receive error flag (ORER, FER, or RER) is SSR is set to 1, even if the TDRE flag in SSR is cleared to 0. Be sure to clear the receive error flags to 0 before starting transmission. Note also that the receive error flags cannot be cleared to 0 even if the RE bit in SCR is cleared to 0.

15.9.5 Relation between Writing to TDR and TDRE Flag

Data can be written to TDR irrespective of the TDRE flag status in SSR. However, if the new data is written to TDR when the TDRE flag is 0, that is, when the previous data has not been transferred to TSR yet, the previous data in TDR is lost. Be sure to write transmit data to TDR after verifying that the TDRE flag is set to 1.

15.9.6 SCI Operations during Mode Transitions

(1) Transmission

Before making the transition to module stop or software standby, stop all transmit operations (TE = TIE = TEIE = 0). TSR, TDR, and SSR are reset. The states of the output pins during each mode depend on the port settings, and the pins output a high-level signal after mode is cancelled and then the TE is set to 1 again. If the transition is made during data transmission, the data being transmitted will be undefined.

To transmit data in the same transmission mode after mode cancellation, set TE to 1, read SSR, write to TDR, clear TDRE in this order, and then start transmission. To transmit data in a different transmission mode, initialize the SCI first.

Figure 15.33 shows a sample flowchart for mode transition during transmission. Figures 15.34 and 15.35 show the pin states during transmission.

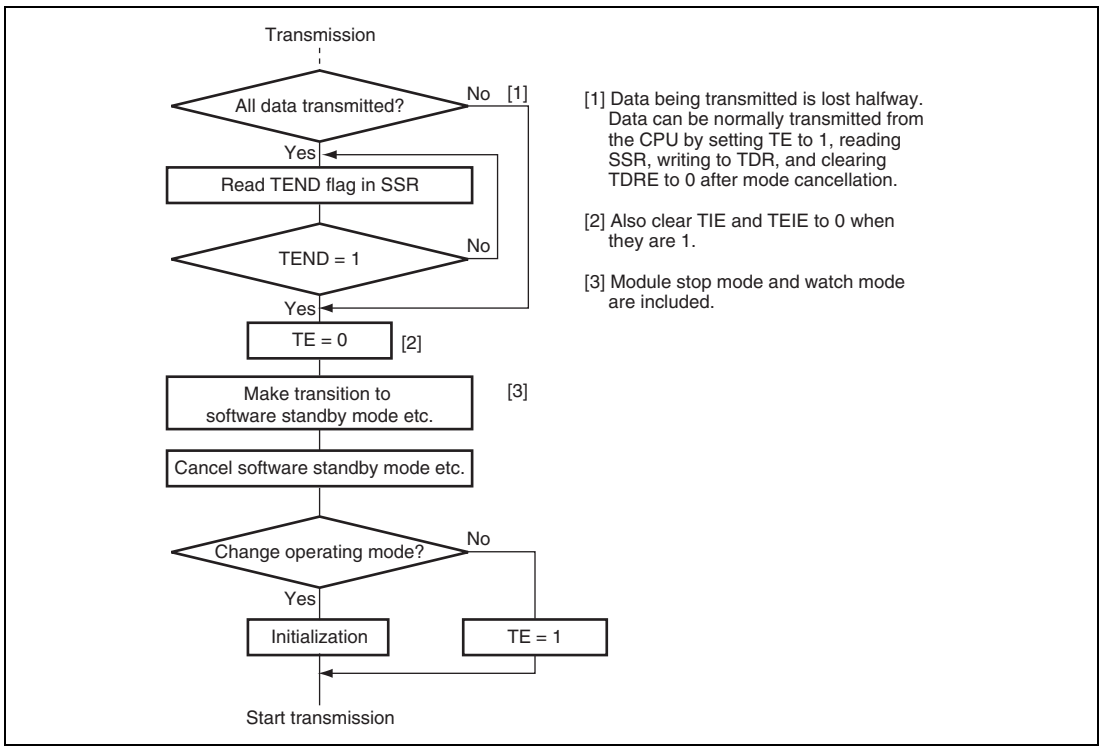


Figure 15.33 Sample Flowchart for Mode Transition during Transmission

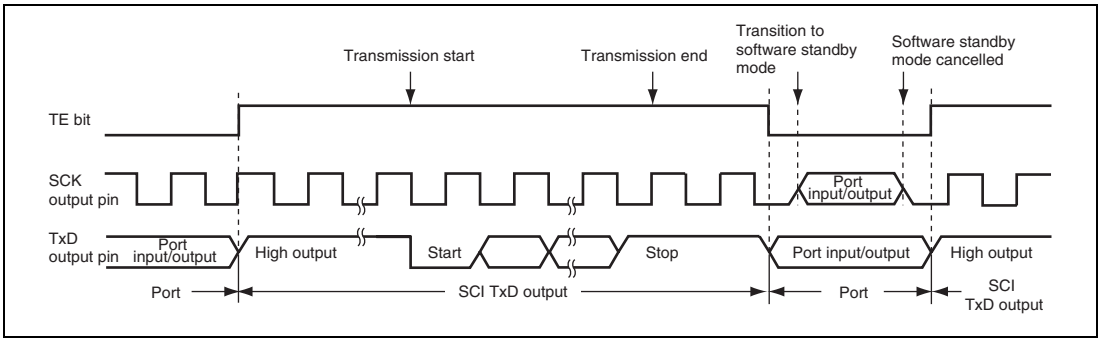


Figure 15.34 Pin States during Transmission in Asynchronous Mode (Internal Clock)

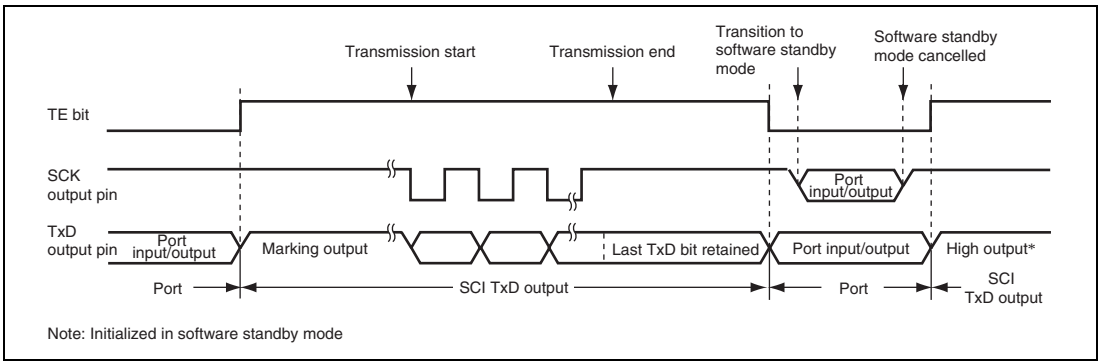


Figure 15.35 Pin States during Transmission in Clocked Synchronous Mode (Internal Clock)

(2) Reception

Before making the transition to module stop, software standby or watch mode, stop reception (RE = 0). RSR, RDR, and SSR are reset. If transition is made during data reception, the data being received will be invalid.

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

Figure 15.36 shows a sample flowchart for mode transition during reception.

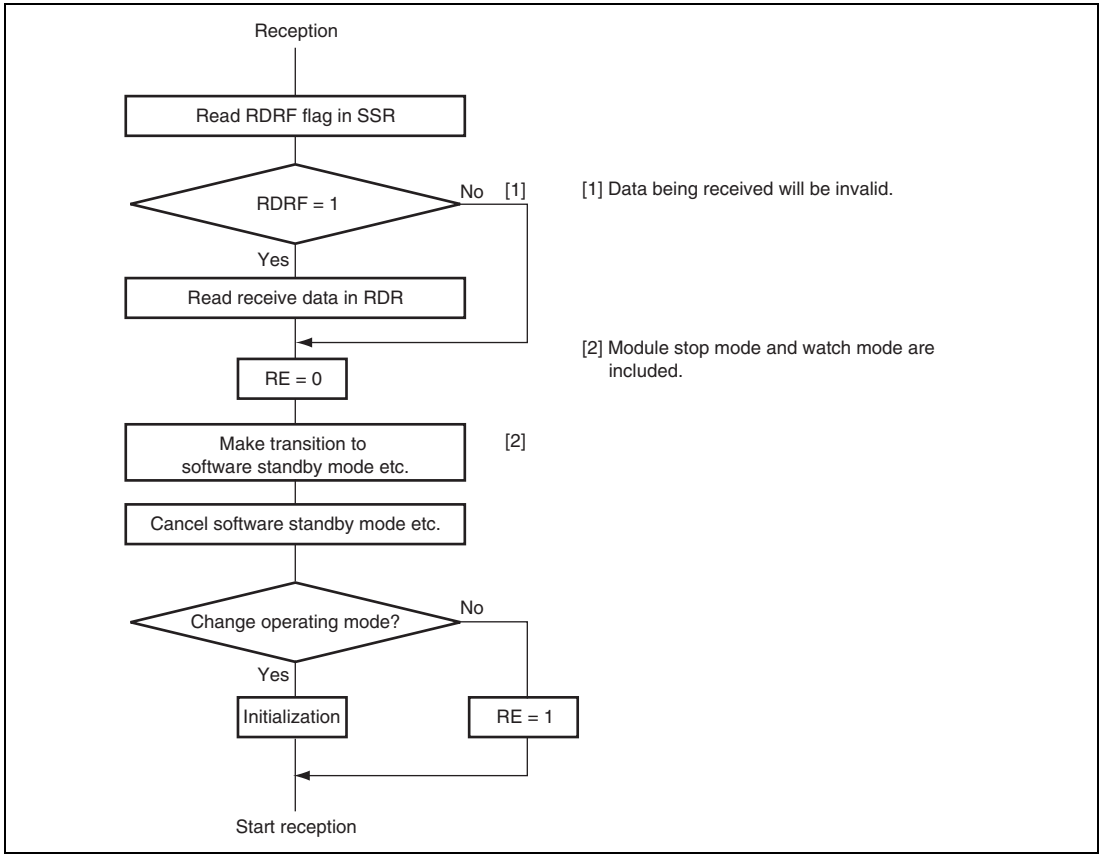


Figure 15.36 Sample Flowchart for Mode Transition during Reception

15.9.7 Notes on Switching from SCK Pins to Port Pins

When SCK pins are switched to port pins after transmission has completed, pins are enabled for port output after outputting a low pulse of half a cycle as shown in figure 15.40.

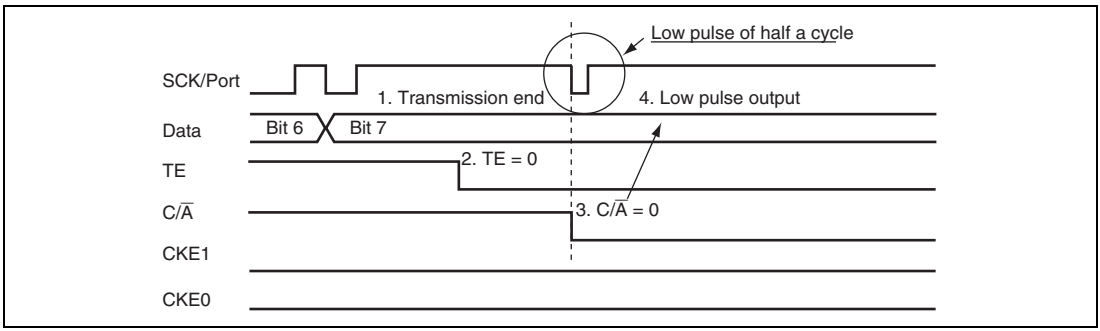


Figure 15.37 Switching from SCK Pins to Port Pins

To prevent the low pulse output that is generated when switching the SCK pins to the port pins, specify the SCK pins for input (pull up the SCK/port pins externally), and follow the procedure below with DDR = 1, DR = 1, C/$\overline{A}$ = 1, CKE1 = 0, CKE0 = 0, and TE = 1.

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

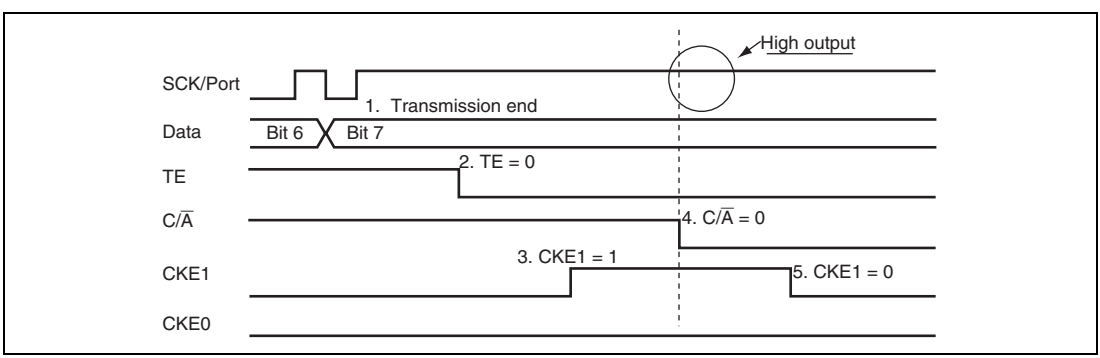


Figure 15.38 Prevention of Low Pulse Output at Switching from SCK Pins to Port Pins

15.9.8 Note on Writing to Registers in Transmission, Reception, and Simultaneous Transmission and Reception

After 1 is set to the TE and RE bits in SCR to start transmission, reception, and simultaneous transmission and reception, do not write to SMR, SCR, BRR, and SDCR. Also, do not overwrite the same value as the register value. However, this does not apply when a register is written to clear the TE and RE bits in SCR to 0 after transmission, reception, or simultaneous transmission and reception is completed. Reading is always allowed.

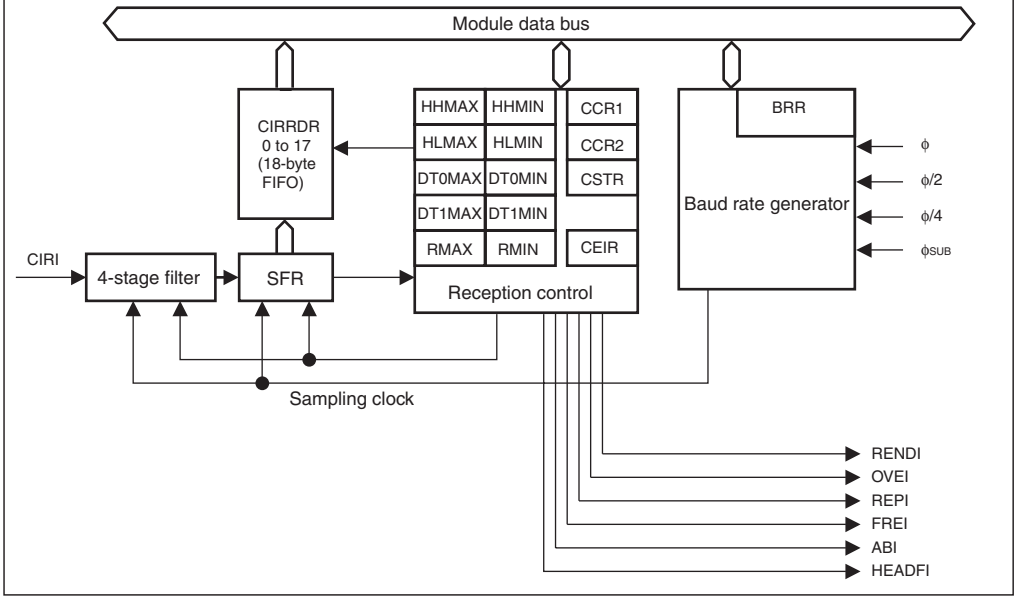
Section 16 CIR Interface

This LSI incorporates a custom infra-red interface (CIR). The CIR has various functions for receiving the IR signal in NEC format.

16.1 Features

- Supports reception of the IR signal in NEC format
- Sampling clock selectable
  Selectable from internal clocks ϕ, $\phi/2$, and $\phi/4$, and subclock (ϕ_{SUB})
- Noise canceling function
  Input noise can be filtered out by using a maximum of four stages of filters.
- Polarity inversion of the input signal supported
- 18-byte FIFO incorporated
- Six interrupt sources: receive end, framing error, overrun error, repeat detection, abort generation, and header detection
  Interrupt sources can be specified by checking each flag.

Figure 16.1 is a block diagram of the CIR.



[Legend]
SFR: Receive shift register
CCR1: Receive control register 1
CCR2: Receive control register 2
CSTR: Receive status register
CEIR: Interrupt enable register
BRR: Bit rate register
CIRRDR0 to 17: Receive data register 0 to 17
HHMIN: Header minimum high-level period register
HHMAX: Header maximum high-level period register
HLMIN: Header minimum low-level period register
HLMAX: Header maximum low-level period register
DT1MIN: Data level 1 minimum period register
DT1MAX: Data level 1 maximum period register
DT0MIN: Data level 0 minimum period register
DT0MAX: Data level 0 maximum period register
RMIN: Repeat header minimum low-level period register
RMAX: Repeat header maximum low-level period register

Figure 16.1 CIR Block Diagram

16.2 Input Pins

Table 16.1 shows the input pin of the CIR.

Table 16.1 Pin Configuration

Pin Name	Symbol	I/O	Function
CIR input pin	CIRI	Input	CIR receive data input pin

16.3 Register Description

Table 16.2 shows the CIR register configuration.

Table 16.2 List of Register Addresses

Register Name	Abbreviation	R/W	Initial Value	Address
Receive control register 1	CCR1	R/W	H'00	H'FA40
Receive control register 2	CCR2	R/W	H'00	H'FA41
Receive status register	CSTR	R/W	H'00	H'FA42
Interrupt enable register	CEIR	R/W	H'00	H'FA43
Bit rate register	BRR	R/W	H'FF	H'FA44
Receive data register 0 to 17	CIRRDR0 to CIRRDR17	R	H'00	H'FA45
Header minimum high-level period register	HHMIN	R/W	H'0000	H'FA46
Header maximum high-level period register	HHMAX	R/W	H'0000	H'FA48
Header minimum low-level period register	HLMIN	R/W	H'00	H'FA4A
Header maximum low-level period register	HLMAX	R/W	H'00	H'FA4B
Data level 0 minimum period register	DT0MIN	R/W	H'00	H'FA4C
Data level 0 maximum period register	DT0MAX	R/W	H'00	H'FA4D
Data level 1 minimum period register	DT1MIN	R/W	H'00	H'FA4E
Data level 1 maximum period register	DT1MAX	R/W	H'00	H'FA4F

Register Name	Abbreviation	R/W	Initial Value	Address
Repeat header minimum low-level period register	RMIN	R/W	H'00	H'FA50
Repeat header maximum low-level period register	RMAX	R/W	H'00	H'FA51

Notes: 1. Before accessing these registers, clear the MSTPA3 bit (bit 3) in MSTPCRA to 0.
2. See the description of each register for details on R/W.

16.3.1 Receive Control Register 1 (CCR1)

CCR1 enable/disable the CIR reception, controls a software reset of the CIR, select the polarity of the CIR input signals, and select the reference clock for CIR reception.

Bit	Bit Name	Initial Value	R/W	Description
7	CIRE	0	R/W	CIR Receive Enable 0: The CIR reception is disabled. 1: The CIR reception is enabled (Port is CIRI input pin).
6	SRES	0	R/W	CIR Software Reset Controls initialization of the internal sequencer of the CIR. 0: Normal operation 1: The internal sequencer is cleared. Writing 1 to this bit generates a clear signal for the internal sequencer in the corresponding module, resulting in the initialization of the CIR's internal state.
5	CPHS	0	R/W	Input Signal Polarity Select 0: CIR input signal is used as is. 1: CIR input signal is inverted before use.
4	MLS	0	R/W	Receive Data Format Select 0: LSB-first data is received. 1: MSB-first data is received.

Bit	Bit Name	Initial Value	R/W	Description
3	REPRCVE	0	R/W	Receive Enable after Repeat Detection Enables/disables the CIR reception after a repeat detection. 0: The CIR reception is disabled by a repeat detection. 1: The CIR reception is enabled by a repeat detection
2	—	0	R/W	Reserved The initial value should not be changed.
1	CLK1	0	R/W	Reference Clock
0	CLK0	0	R/W	00: Internal clock ϕ 01: Internal clock ϕ/2 10: Internal clock ϕ/4 11: subclock ϕsub

16.3.2 Receive Control Register 2 (CCR2)

CCR2 consists of the bits that select the CIR communication format.

Bit	Bit Name	Initial Value	R/W	Description
7	TFM1	0	R/W	Reception Signal Format Select
6	TFM0	0	R/W	00: NEC format (4 bytes are used) (Address, $\overline{\text{address}}$, command, and $\overline{\text{command}}$ are stored in CIRRDR.) 01: NEC format (2 bytes are used) (Address and command are stored in CIRRDR.) 10: Setting prohibited 11: Setting prohibited
5 to 0	—	All 0	R/W	Reserved The initial value should not be changed.

16.3.3 Receive Status Register (CSTR)

CSTR indicates the data reception state of the CIR.

Bit	Bit Name	Initial Value	R/W	Description
7	CIRBUSY	0	R	CIR Busy Flag Indicates the data receive state of the CIR. [Setting condition] When the CIR starts data reception. [Clearing condition] When the CIR has finished data reception.
6	CIRRDRF	0	R	Receive Data Register Full Indicates whether CIRRDR contains a receive data or not. This bit cannot be modified. [Setting condition] When a receive data is stored into CIRRDR. [Clearing condition] When a receive data has been read from CIRRDR.
5	REPF	0	R/W*	Repeat Detection Flag Indicates a repeat is generated. [Setting condition] When a repeat is detected. [Clearing condition] When writing 0 after reading REPF = 1.
4	OVRF	0	R/W*	Overrun Error Flag Indicates CIRRDR overflows. [Setting condition] When the next data is stored in CIRRDR while CIRRDR is full. [Clearing condition] When writing 0 after reading OVRF = 1.

Bit	Bit Name	Initial Value	R/W	Description
3	REND	0	R/W*	Reception End Flag [Setting condition] When the CIR has finished data reception. (When a stop is detected.) [Clearing condition] When writing 0 after reading REND = 1.
2	ABF	0	R/W*	Abort Flag An internal reset is generated when an abort (transfer format) is detected. [Setting condition] When data other than logic 0 or 1 is detected. [Clearing condition] When writing 0 after reading ABF = 1.
1	FRF	0	R/W*	Framing Error Flag [Setting condition] • When a stop is detected during data reception. • When the time period of a stop is too short. [Clearing condition] When writing 0 after reading FRF = 1.
0	HEADF	0	R/W*	Header Detection Flag [Setting condition] When a header is detected. [Clearing condition] When writing 0 after reading HEADF = 1.

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

16.3.4 Interrupt Enable Register (CEIR)

CEIR consists of the bits that enable/disable various interrupts.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R/W	Reserved The initial value should not be changed.
5	REPIE	0	R/W	Repeat Detection Interrupt Enable 0: REPI interrupt request is disabled. 1: REPI interrupt request is enabled.
4	OVEIE	0	R/W	Overrun Error Interrupt Enable 0: OVEI interrupt request is disabled. 1: OVEI interrupt request is enabled.
3	RENDIE	0	R/W	Receive End Interrupt Enable 0: RENDI interrupt request is disabled. 1: RENDI interrupt request is enabled.
2	ABIE	0	R/W	Abort Interrupt Enable 0: ABI interrupt request is disabled. 1: ABI interrupt request is enabled.
1	FREIE	0	R/W	Framing Error Interrupt Enable 0: FREI interrupt request is disabled. 1: FREI interrupt request is enabled.
0	HEADFIE	0	R/W	Header Detection Interrupt Enable 0: HEADFI interrupt request is disabled. 1: HEADFI interrupt request is enabled.

16.3.5 Bit Rate Register (BRR)

BRR is an 8-bit register that adjusts the sampling clock signal used for CIR reception. The bit rate for the CIR reception is determined by a combination of the setting value in BRR and the CLK1 and CLK0 bits in CCR1.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	BRR7 to BRR1	All 1	R/W	Sets the value of the sampling clock.

The following formula is used for calculating the bit rate, and the following table shows BRR setting examples to obtain a target bit rate.

$$B = T / (N + 1)$$

B: Bit rate (bits/s)

T: Frequency of the reference clock (Hz) set by the CLK1 and CLK0 bits in CCR1 (ϕ, ϕ/2, ϕ/4, or ϕsub)

N: Set value in BRR ($0 \leq N \leq 255$)

Table 16.3 Setting Example of BRR

Carrier Frequency	ϕ	CLK1 and CLK0 Setting	BRR Setting Value	Bit Rate (Kbit/s)	Deviation from Target Carrier Frequency
38kHz	20 MHz	ϕ	H'FF	78.1	51.36%
		ϕ/2	H'FF	39.1	2.72%
		ϕ/4	H'83	37.9	–0.32%
	10 MHz	ϕ	H'FF	39.1	2.72%
		ϕ/2	H'83	37.9	–0.32%
		ϕ/4	H'41	37.9	–0.32%
	8 MHz	ϕ	H'D2	37.9	–0.23%
		ϕ/2	H'69	37.7	–0.70%
		ϕ/4	H'34	37.7	–0.70%
	—	ϕsub	H'00	32.8	2.34%

16.3.6 Receive Data Register 0 to 17 (CIRRDR0 to CIRRDR17)

CIRRDR0 to CIRRDR17 are an 18-byte register that stores receive data, totaling to 18 bytes. CIRRDR0 to CIRRDR17 share one byte of the register address. A receive data in CIRRDR should be read after the CIR has finished data reception (CIRBUSY = 0). If CIRRDR is read during the CIR reception (CIRBUSY = 1), an undefined value is read.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	CIRRDR7 to CIRRDR0	H'00	R	Stores the CIR receive data.

16.3.7 Header Minimum/Maximum High-Level Period Register (HHMIN and HHMAX)

HHMIN and HHMAX control the noise canceler circuit, and specify the minimum and maximum high-level period for a header or repeat header, and low-level period for a stop.

- HHMIN

Bit	Bit Name	Initial Value	R/W	Description
15 to 11	RFMBN4 to RFMBN0	All 0	R	Receive Byte Counter The RFMBN value is incremented by 1 (+1) each time a byte is received. However, when RFMBN reaches B'10011, an overrun error occurs. At this time, a receive data is not stored in CIRRDR. When CIRRDR is read after the CIR has finished receiving (CIRBUSY = 0), RFMBN is decremented by 1 (−1). When CIRRDR is read while RFMBN = B'00000, an undefined value is read. When CIRRDR is read during the CIR reception, an undefined value is read and RFMBN is not decremented.
10	—	0	R/W	Reserved The initial value should not be changed.
9 to 0	HHMIN9 to HHMIN0	All 0	R/W	Specifies the minimum high-level period for a header or repeat header and the minimum low-level period for a stop.

- HHMAX

Bit	Bit Name	Initial Value	R/W	Description
15	FLT1	0	R/W	Number of Stages of Noise Canceler Circuit Select
14	FLT0	0	R/W	00: The noise canceler circuit consists of one stage 01: The noise canceler circuit consists of two stages 10: The noise canceler circuit consists of three stages 11: The noise canceler circuit consists of four stages
13	FLTE	0	R/W	Noise Canceler Circuit Enable 0: Disables the noise canceler circuit 1: Enables the noise canceler circuit
12	FLTCK1	0	R/W	Division Ratio Select for Noise Canceler Circuit Clock Divides the frequency of the sampling clock for CIR reception selected by BRR. 00: Not divided 01: Divided by 2 10: Divided by 4 11: Divided by 8
11	FLTCK0	0	R/W	
10	—	0	R/W	Reserved The initial value should not be changed.
9 to 0	HHMAX9 to HHMAX0	All 0	R/W	Specifies the maximum high-level period for a header or repeat header and the maximum low-level period for a stop.

16.3.8 Header Minimum/Maximum Low-Level Period Register (HLMIN/HLMAX)

HLMIN and HLMAX specify the minimum and maximum low-level period for a header.

- HLMIN

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	HLMIN7 to HLMIN0	H'00	R/W	Specifies the minimum low-level period for a header.

- HLMAX

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	HLMAX7 to HLMAX0	H'00	R/W	Specifies the maximum low-level period for a header.

16.3.9 Data Level 1 Minimum/Maximum Period Register (DT1MIN/DT1MAX)

DT1MIN and DT1MAX specify the minimum and maximum low-level period for logic 1.

- DT1MIN

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	DT1MIN7 to DT1MIN0	H'00	R/W	Specifies the minimum low-level period for logic 1.

- DT1MAX

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	DT1MAX7 to DT1MAX0	H'00	R/W	Specifies the maximum low-level period for logic 1.

16.3.10 Data Level 0 Minimum/Maximum Period Register (DT0MIN/DT0MAX)

DT0MIN and DT0MAX specify the minimum and maximum low/high-level period for logic 0, high-level period for logic 1, and high-level period for a stop/repeat.

- DT0MIN

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	DT0MIN7 to DT0MIN0	H'00	R/W	Specifies the minimum low/high-level period for logic 0, high-level period for logic 1, and high-level period for a stop/repeat.

- DT0MAX

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	DT0MAX7 to DT0MAX0	H'00	R/W	Specifies the maximum low/high-level period for logic 0, high-level period for logic 1, and high-level period for a stop/repeat.

16.3.11 Repeat Header Minimum/Maximum Low-Level Period Register (RMIN/RMAX)

RMIN and RMAX specify the minimum and maximum low-level period for a repeat header.

- RMIN

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	RMIN7 to RMIN0	H'00	R/W	Specifies the minimum low-level period for a repeat header.

- RMAX

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	RMAX7 to RMAX0	H'00	R/W	Specifies the maximum low-level period for a repeat header.

16.4 Operation

The communication protocol of the NEC format is shown in figure 16.2. In the NEC format, data consists of a header part, an address part, a command part, and a stop part. The TFM bits in CCR2 can select which data bytes to be stored in CIRRDR: four bytes of the address, address, command, and command, or two bytes of the address and command. The carrier frequency is 38kHz.

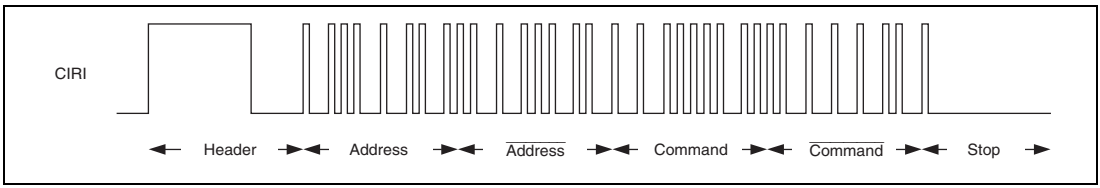


Figure 16.2 NEC Format

(1) Header, Address, and Command

When a 9-ms high level and the following 4.5-ms low level are detected, they are recognized as a header. For addresses and commands, when both of a high-level period and the following low-level period are 0.56 ms, they are recognized as logic 0. When a high-level period is 0.56 ms and the following low-level period is 1.78 ms, they are recognized as logic 1.

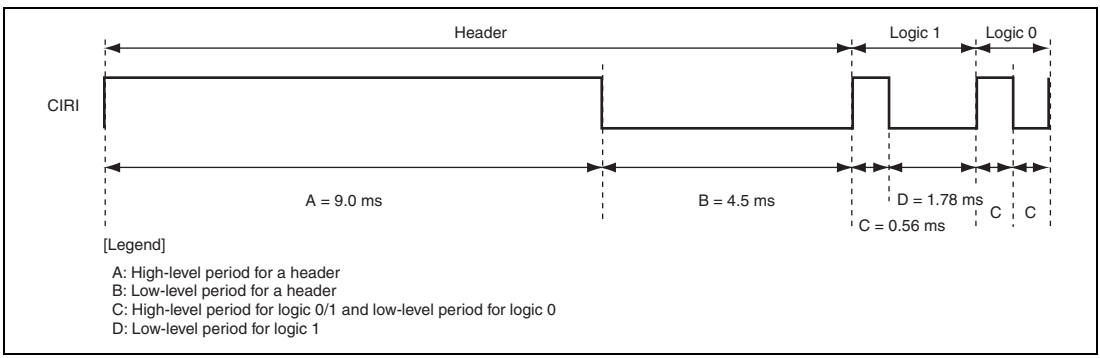


Figure 16.3 Header, Address, and Command

(2) Stop

When a low-level period of 9 ms or more is detected after the reception of a command, the CIR stops data reception. This is not defined in the NEC format.

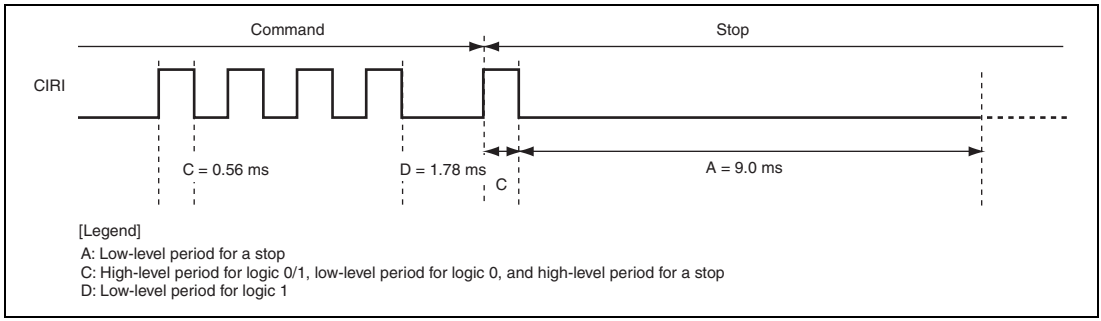


Figure 16.4 Stop

(3) Repeat

When a key of the remote controller remains pressed, the command is sent only once, followed by a repeat signal. When a 9-ms high level and the following 2.25-ms low level are detected, they are recognized as a repeat header.

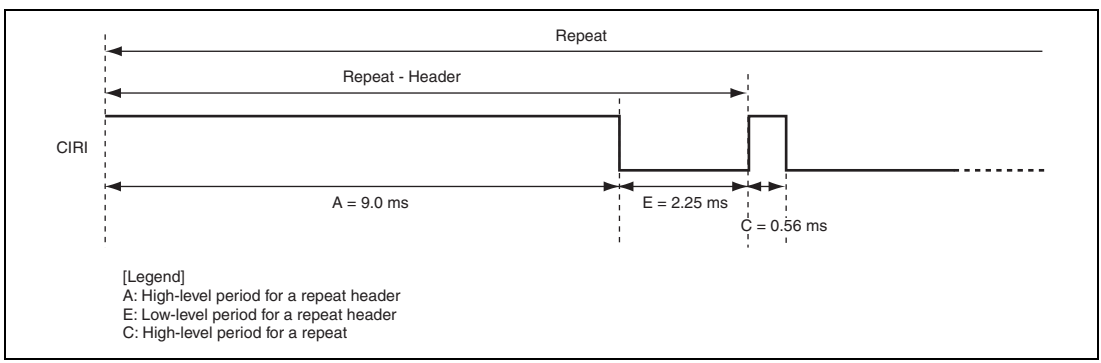


Figure 16.5 Repeat

16.4.1 Determination of Signal Type by Low/High-Level Period

The signal type is determined by low/high-level period that is specified in the HHMIN, HHMAX, HLMIN, HLMAX, DT1MIN, DT1MAX, DT0MIN, DT0MAX, RMIN, and RMAX registers. Calculating formula for specified time, setting examples of each maximum/minimum value register during the specified time, and use for each register are described as follows. The symbols in table 16.4 correspond to the ones used in the figure 16.3 to figure 16.5.

$S.E = M (N + 1)/T$

S: Specified time of the NEC format

E: Error from the NEC format

T: Frequency of the reference clock (Hz) set by the CLK1 and CLK0 bits in CCR1 (ϕ, $\phi/2$, $\phi/4$, or ϕsub)

N: Setting value in BRR ($0 \leq N \leq 255$)

M: Value in the maximum/minimum value setting register

Table 16.4 An Example of Signal Type Determination Register Setting

Description	Register Name	Symbol	Setting Value	Setting Time	Prescribed Time (Error: 30%)	Notes
Minimum high-level period for a header or repeat header and minimum low-level period for a stop	HHMIN	A	H'079	6.34 ms	6.3 ms	HHMIN9 to HHMIN0
Maximum high-level period for a header or repeat header and maximum low-level period for a stop	HHMAX	A	H'0DF	11.7 ms	11.7 ms	HHMAX9 to HHMAX0
Minimum low-level period for a header	HLMIN	B	H'3D	3.20 ms	3.15 ms	
Maximum low-level period for a header	HLMAX	B	H'6F	5.82 ms	5.85 ms	
Minimum value of low/high-level period for logic 0, high-level period for logic 1, and high-level period for a burst	DT0MIN	C	H'07	0.37 ms	0.39 ms	
Maximum value of low/high-level period for logic 0, high-level period for logic 1, and high-level period for a burst	DT0MAX	C	H'0D	0.68 ms	0.73 ms	
Minimum low-level period for logic 1	DT1MIN	D	H'0F	0.78 ms	0.78 ms	
Maximum low-level period for logic 1	DT1MAX	D	H'1B	1.42 ms	1.46 ms	
Minimum low-level period for a repeat header	RMIN	E	H'1F	1.62 ms	1.58 ms	
Maximum low-level period for a repeat header	RMAX	E	H'37	2.88 ms	2.92 ms	

Note: The above table shows the values when the system clock is 10MHz, CLK1, CLK0 = B'10, and BRR = H'82 (when the error is 30%).

16.4.2 Operation of FIFO Register

A FIFO structure provides first-in first-out operation.

Operation of the FIFO when it receives data three times (byte 0, byte 1, and byte 2 in order) and is then read three times is as shown below.

Operation for first reception of data

Number of bytes	FIFO contents
1	Byte 0
2	H'00
3	H'00
4	H'00
. . .	. . .
18	H'00

Operation for data reception three times

Number of bytes	FIFO contents
1	Byte 0
2	Byte 1
3	Byte 2
4	H'00
. . .	. . .
18	H'00

Figure 16.6 Operation when FIFO Data is Received

First read

Number of bytes	FIFO Contents
1	Byte 1
2	Byte 2
3	H'00
4	H'00
. . .	. . .
18	H'00

Second read

Number of bytes	FIFO Contents
1	Byte 2
2	H'00
3	H'00
4	H'00
. . .	. . .
18	H'00

Third read

Number of bytes	FIFO contents
1	H'00
2	H'00
3	H'00
4	H'00
. . .	. . .
18	H'00

Figure 16.7 Operation when FIFO Data is Read

In case of reading more bytes than the number that has been received, (number of received bytes + 1) of data are always read out from the FIFO.

Reception of more than 18 bytes by the FIFO structure for this CIR module leads to an overrun. When an overrun occurs, only values up to the 18th byte to have been received are read out in response to the reading of more than 18 bytes.

16.4.3 Operation in Watch Mode

Initiate the transition to watch mode after making the below settings for the mode transition.

- Select the subclock (ϕsub) as the operating clock for the CIR module.
- Enable the CIR header-detected interrupt.

For a transition from watch mode to high-speed mode, the CIR module generates an interrupt on detection of a received header, in accord with the settings before the transition.

The module is released from watch mode when the interrupt is generated, and makes the transition to the high- or medium-speed mode.

16.4.4 Switching between System Clock and Sub Clock

If the operating clock is switched from the system clock to the subclock (ϕsub) while the CIR module is operating, operation may not proceed correctly. To switch the operating clock, be sure to stop the CIR module (by clearing the CIRE bit) beforehand.

16.5 Noise Canceler Circuit

The CIR incorporates a 4-stage noise canceler. The FLTE, FLT, and FLTCK1 and FLTCK0 bits in HHMAX enable/disable the noise canceler circuit, select the number of stages of the noise canceler circuit, and select the division ratio for generating the noise canceler circuit clock, respectively. Figure 16.6 shows a block diagram of the noise canceler circuit.

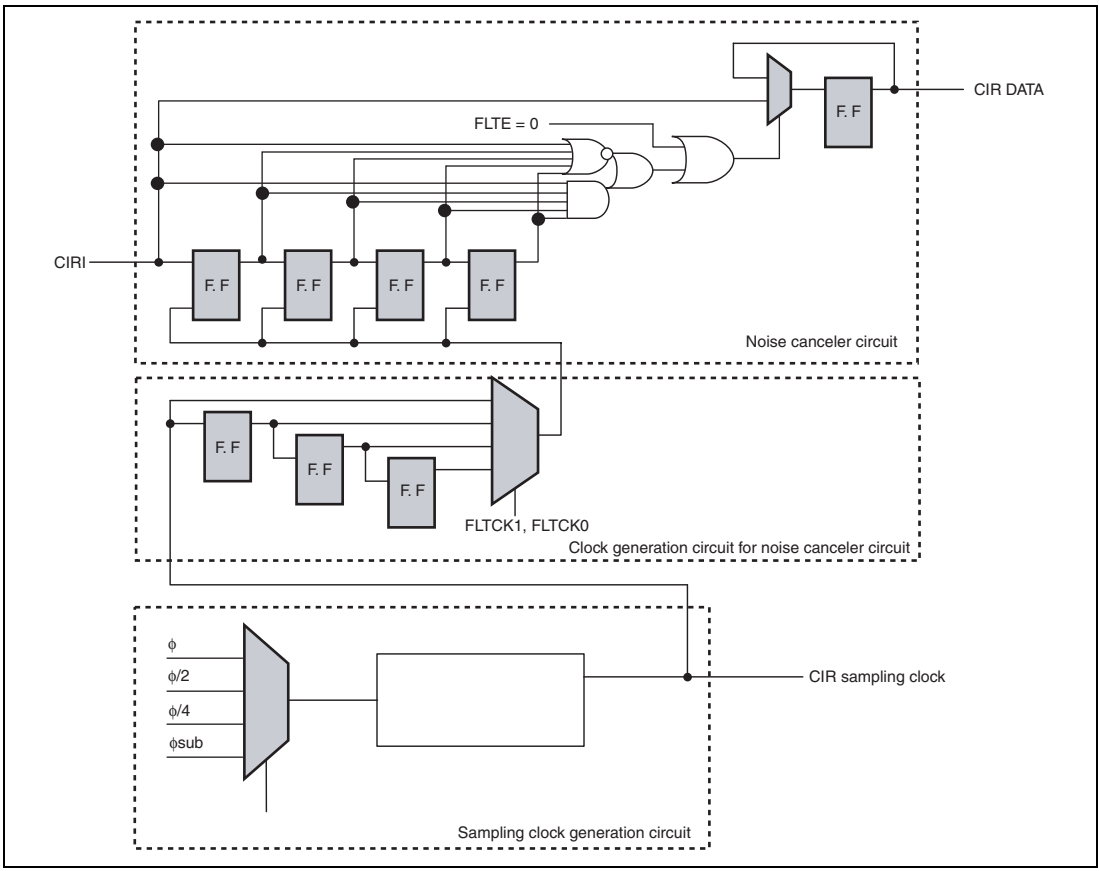


Figure 16.8 Noise Canceler Circuit

Table 16.5 shows sample settings for the noise canceler circuit.

Table 16.5 Sample Settings for Noise Canceler Circuit

ϕ	CLK1 and CLK0 Setting	BRR Setting	FLTCK1 and FLTCK0 Setting	CIR Sampling Clock	Number of Stages of Noise Canceler Circuit	Width of Noise Cancellation
10 MHz	ϕ	H'80	Not divided	12.9 μs	0	12.9 μs
					1	25.8 μs
					2	38.7 μs
					3	51.6 μs
					4	64.5 μs
			Divided by 2	25.8 μs	0	25.8 μs
					2	77.4 μs
					4	129 μs
			Divided by 4	51.6 μs	0	51.6 μs
					2	154.8 μs
					4	258 μs
			Divided by 8	103.2 μs	0	103.2 μs
					2	309.6 μs
					4	516 μs
—	ϕsub	H'00	Not divided	31.3 μs	0	31.3 μs
					1	62.5 μs
					2	93.8 μs
					3	125 μs
					4	156 μs
			Divided by 2	62.5 μs	0	62.5 μs
					2	187.5 μs
					4	312.5 μs
			Divided by 4	125 μs	0	125 μs
					2	375 μs
					4	625 μs
			Divided by 8	250 μs	0	250 μs
					2	750 μs
					4	1.25 ms

16.6 Reset Conditions

The range of initialization caused by a system reset, a software reset controlled by the SRES bit in CCR1, or an abort is shown in table 16.6.

Table 16.6 Range of Initialization of CIR

	HHMIN, HHMAX, HLMIN, HLMAX, DT0MIN, DT0MAX, DT1MIN, DT1MAX, CCR1, CCR2, CEIR	RFMBN bit in HHMIN	CIRRDR	CSTR	Sequence Block	BRR
System reset	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized
SRES software reset	Retained	Initialized	Initialized	Initialized	Initialized	Initialized
Abort	Retained	Retained	Retained	Retained * (CIRBUSY is initialized.)	Initialized	Retained

16.7 Interrupt Sources

The CIR has six interrupt source flags for this LSI. Setting the corresponding enable bit to 1 enables the relevant interrupt request to be issued. Since the six interrupt requests are allocated to one vector address, it is necessary for the CPU to check the interrupt request flags in order to determine which interrupt source has caused the interrupt to be requested.

Table 16.7 Interrupt Sources

Interrupt Name	Interrupt Source Flags		Interrupt Enable Bit
RENDI	REND	Receive end	RENDIE
OVEI	OVRF	Overrun error	OVEIE
REPI	REPF	Repeat detection	REPIE
FREI	FRF	Framing error	FREIE
ABI	ABF	Abort	ABIE
HEADFI	HEADF	Header detection	HEADFIE

16.8 Usage Note

(1) CIR Register Setting

Before starting the CIR reception, set the CIR by following the flow shown in figure 16.7.

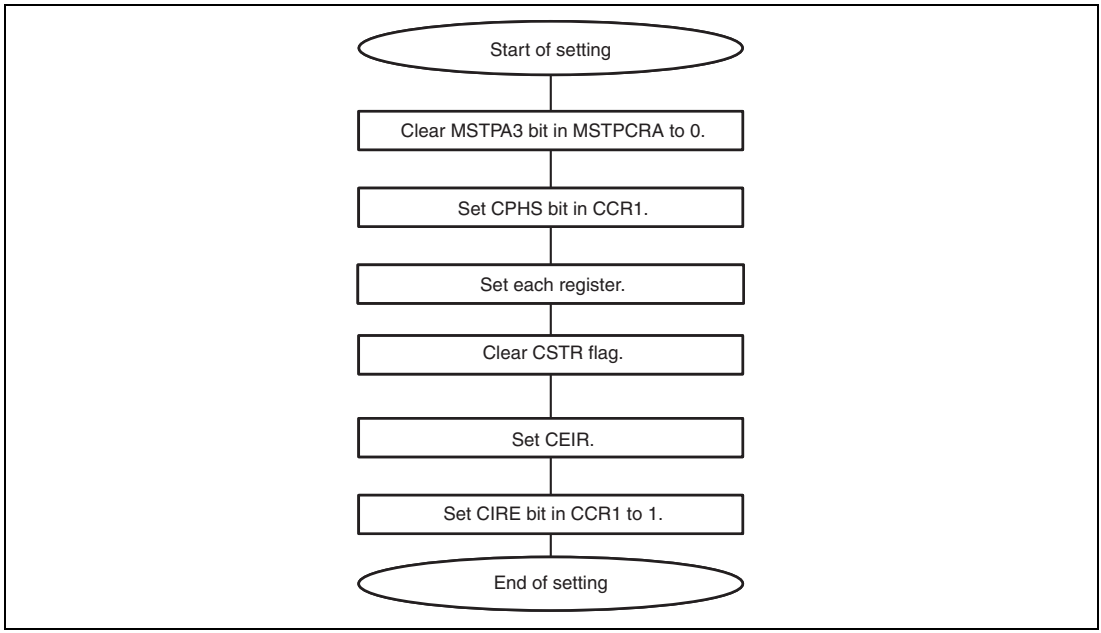


Figure 16.9 CIR Setting Flow

The CPHS bit in CCR1 should be set before starting reception. When the CIRI pin is high in the idle state, set the CPHS bit to 1. When it is low in the idle state, clear the bit to 0. The BRR register is initialized to H'FF by setting the SRES bit in CCR1 to 1. After setting each register in the CIR, set the CIRE bit in CCR1 to 1 to enable the CIR reception.

(2) Switching between System Clock and Sub Clock

The CIR is capable of remote-control reception by using the sub clock in watch mode. Before switching between the system clock and the sub clock, the CIR must be stopped by clearing the CIRE bit to 0.

(3) Overrun Operation with the NEC format (2 Bytes are Used)

When the reception signal format select bits (bits TFM1 and TFM0 in CCR2) are set to the NEC format (2 bytes are used), the OVRF bit in CSTR is set to indicate the overrun on the reception of the 18th byte by the receive data register. However, this does not affect the contents of the 18th–byte data.

Section 17 Serial Communication Interface with FIFO (SCIF)

This LSI has single-channel serial communication interface with FIFO buffers (SCIF) that supports asynchronous serial communication.

The SCIF enables asynchronous serial communication with standard asynchronous communication LSIs such as a Universal Asynchronous Receiver/Transmitter (UART). The SCIF also has independent 16-stage FIFO buffers for transmission and reception to provide efficient high-speed continuous communication.

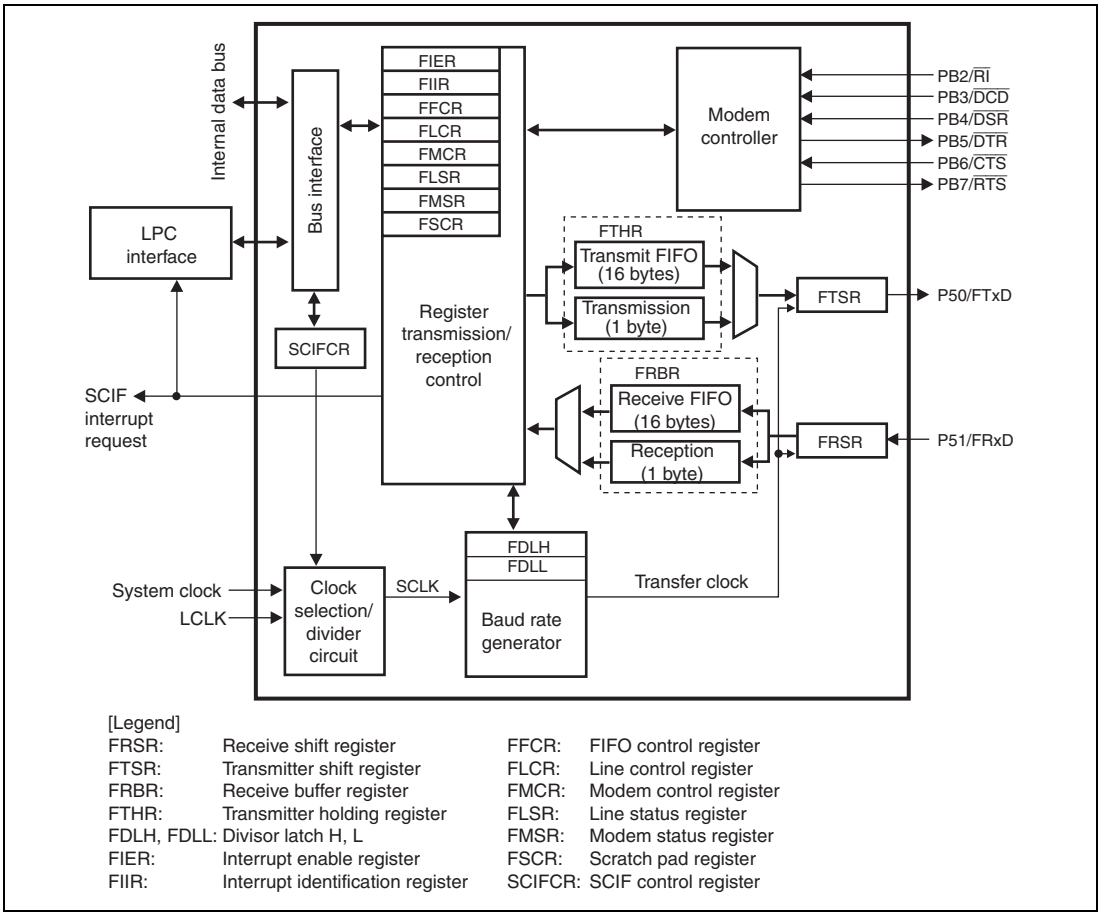
In addition, the SCIF can be connected to the LPC interface for direct control from the LPC host.

17.1 Features

- Full-duplex communication:

  The transmitter and receiver are independent, enabling transmission and reception to be executed simultaneously. Both the transmitter and receiver use 16-stage FIFO buffering, enabling continuous transmission and continuous reception of serial data.
- On-chip baud rate generator allows any bit rate to be selected
- Modem control function
- Data length: Selectable from 5, 6, 7, and 8 bits
- Parity: Selectable from even parity, odd parity, and no parity
- Stop bit length: Selectable from 1, 1.5, and 2 bits
- Receive error detection: Parity, overrun, and framing errors
- Break detection

Figure 17.1 shows a block diagram of the SCIF.



[Legend]

FRSR:	Receive shift register	FFCR:	FIFO control register
FTSR:	Transmitter shift register	FLCR:	Line control register
FRBR:	Receive buffer register	FMCR:	Modem control register
FTHR:	Transmitter holding register	FLSR:	Line status register
FDLH, FDLL:	Divisor latch H, L	FMSR:	Modem status register
FIER:	Interrupt enable register	FSCR:	Scratch pad register
FIIR:	Interrupt identification register	SCIFCR:	SCIF control register

Figure 17.1 Block Diagram of SCIF

17.2 Input/Output Pins

Table 17.1 lists the SCIF input/output pins.

Table 17.1 Pin Configuration

Pin Name	Port	Input/Output	Function
FTxD	P50	Output	Transmit data output
FRxD	P51	Input	Receive data input
$\overline{\text{RI}}$	PB2	Input	Ring indicator input
$\overline{\text{DCD}}$	PB3	Input	Data carrier detect input
$\overline{\text{DSR}}$	PB4	Input	Data set ready input
$\overline{\text{DTR}}$	PB5	Output	Data terminal ready output
$\overline{\text{CTS}}$	PB6	Input	Transmission permission input
$\overline{\text{RTS}}$	PB7	Output	Transmission request output

17.3 Register Descriptions

The SCIF has the following registers. The register configuration of the SCIF is shown below. Access to the registers is switched by the SCIFE bit in HICR5 and bit 3 in MSTPCRB. For details, see table 17.3. For the SCIF address registers H and L (SCIFADRH, SCIFADRL) and serial IRQ control register 4 (SIRQCR4), see section 20, LPC Interface (LPC).

Table 17.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Data Bus Width
Host interface control register 5	HICR5	R/W	H'00	H'FFFE33	8
Module stop control register B	MSTPCRB	R/W	H'00	H'FFFE7F	8
Receive buffer register	FRBR	R	H'00	H'FFFC20	8
Transmitter holding register	FTHR	W	—		
Divisor latch L	FDLL	R/W	H'00		
Interrupt enable register	FIER	R/W	H'00	H'FFFC21	8
Divisor latch H	FDLH	R/W	H'00		
Interrupt identification register	FIIR	R	H'01	H'FFFC22	8
FIFO control register	FFCR	W	H'00		
Line control register	FLCR	R/W	H'00	H'FFFC23	8
Modem control register	FMCR	R/W	H'00	H'FFFC24	8
Line status register	FLSR	R	H'60	H'FFFC25	8
Modem status register	FMSR	R	—	H'FFFC26	8
Scratch pad register	FSCR	R/W	H'00	H'FFFC27	8
SCIF control register	SCIFCR	R/W	H'00	H'FFFC28	8
SCIF address register H	SCIFADRH	R/W	H'03	H'FFFDC4	8
SCIF address register L	SCIFADRL	R/W	H'F8	H'FFFDC5	8
Serial IRQ control register 4	SIRQCR4	R/W	H'00	H'FFFE3B	8

Table 17.3 Register Access

SCIFE Bit in HICR5		0		1
Bit 3 in MSTPCRB	**0**	**1**	**0**	**1**
SCIFCR	H8S CPU access*[2]	Access disabled	H8S CPU access*[2]	Access disabled
Other than SCIFCR	H8S CPU access*[2]	Access disabled	LPC access*[1]	LPC access*[1]

Notes: 1. When LPC access is set, writing from the H8S CPU is disabled. The read value is H'FF.
2. When H8S CPU access is set, writing from the LPC is disabled. The read value is H'00.

17.3.1 Receive Shift Register (FRSR)

FRSR is a register that receives data and converts serial data input from the FRxD pin to parallel data. It stores the data in the order received from the LSB (bit 0). When one frame of serial data has been received, the data is transferred to FRBR.

FRSR cannot be read from the CPU/LPC interface.

17.3.2 Receive Buffer Register (FRBR)

FRBR is an 8-bit read-only register that stores received serial data. It can read data correctly when the DR bit in FLSR is set.

When the FIFO is disabled, the data in FRBR must be read before the next data is received. If new data is received before the remaining data is read, the data is overwritten, resulting in an overrun error.

When this register is read with the FIFO enabled, the first buffer of the receive FIFO is read. When the receive FIFO becomes full, the subsequent receive data is lost, resulting in an overrun error.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	Bit 7 to bit 0	All 0	R	Stores received serial data. The data is 16 bytes when the FIFO is enabled.

17.3.3 Transmitter Shift Register (FTSR)

FTSR is a register that converts parallel data from the FTxD pin to serial data and then transmits the serial data. When one frame transmission of serial data is completed, the next data is transferred from FTHR. The serial data is transmitted from the LSB (bit 0).

FTSR cannot be written from the H8S CPU/LPC interface.

17.3.4 Transmitter Holding Register (FTHR)

FTHR is an 8-bit write-only register that stores serial transmit data. It is accessible when the DLAB bit in FLCR is 0. Write transmit data while the THRE bit in FLCR is set to 1.

Data can be written to FTHR when the THRE bit is set with the FIFO disabled. If data is written to FTHR when the THRE bit is not set, the data is overwritten.

While the THRE bit is set with the FIFO enabled, up to 16 bytes of data can be written. If data is written with the FIFO full, the written data is lost.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	Bit 7 to bit 0	—	W	Stores serial data to be transmitted. The data is 16 bytes when the FIFO is enabled.

17.3.5 Divisor Latch H, L (FDLH, FDLL)

The FDLH and FDLL are registers used to set the baud rate. They are accessible when the DLAB bit in FLCR is 1. Frequency division ranging from 1 to $(2^{16} - 1)$ can be set with these registers. The frequency divider circuit stops when both of FDLH and FDLL are 0 (initial value).

- FDLH

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	Bit 7 to bit 0	All 0	R/W	Upper 8 bits of divisor latch

- FDLL

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	Bit 7 to bit 0	All 0	R/W	Lower 8 bits of divisor latch

Baud rate = (Clock frequency input to baud rate generator) / (16 × divisor value)

17.3.6 Interrupt Enable Register (FIER)

FIER is a register that enables or disables interrupts. It is accessible when the DLAB bit in FLCR is 0.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved This bit is always read as 0 and cannot be modified.
3	EDSSI	0	R/W	Modem Status Interrupt Enable 0: Modem status interrupt disabled 1: Modem status interrupt enabled
2	ELSI	0	R/W	Receive Line Status Interrupt Enable 0: Receive line status interrupt disabled 1: Receive line status interrupt enabled
1	ETBEI	0	R/W	FTHR Empty Interrupt Enable 0: FTHR empty interrupt disabled 1: FTHR empty interrupt enabled
0	ERBFI	0	R/W	Receive Data Ready Interrupt Enable A character timeout interrupt is included when the FIFO is enabled. 0: Receive data ready interrupt disabled 1: Receive data ready interrupt enabled

17.3.7 Interrupt Identification Register (FIIR)

FIIR consists of bits that identify interrupt sources. For details, see table 17.4.

Bit	Bit Name	Initial Value	R/W	Description
7	FIFOE1	0	R	FIFO Enable 1, 0
6	FIFOE0	0	R	These bits indicate the transmit/receive FIFO setting. 00: Transmit/receive FIFOs disabled 11: Transmit/receive FIFOs enabled
5, 4	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
3	INTID2	0	R	Interrupt ID2, ID1, ID0
2	INTID1	0	R	These bits Indicate the interrupt of the highest priority among the pending interrupts.
1	INTID0	0	R	000: Modem status 001: FTHR empty 010: Receive data ready 011: Receive line status 110: Character timeout (when the FIFO is enabled)
0	INTPEND	1	R	Interrupt Pending Indicates whether one or more interrupts are pending. 0: Interrupt pending 1: No interrupt pending

Table 17.4 Interrupt Control Function

FIIR			Setting/Clearing of Interrupt				
INTID							
2	1	0	INTPEND	Priority	Type of Interrupt	Interrupt Source	Clearing of Interrupt
0	0	0	1	—	No interrupt	None	-
0	1	1	0	1 (high)	Receive line status	Overrun error, parity error, framing error, break interrupt	FLSR read
0	1	0	0	2	Receive data ready	Receive data remaining, FIFO trigger level	FRBR read or receive FIFO is below trigger level.
1	1	0	0	2	Character timeout (with FIFO enabled)	No data is input to or output from the receive FIFO for the 4-character time period while one or more characters remain in the receive FIFO.	FRBR read
0	0	1	0	3	FTHR empty	FTHR empty	FIIR read or FTHR write
0	0	0	0	4 (low)	Modem status	CTS, DSR, RI, DCD	FMSR read

17.3.8 FIFO Control Register (FFCR)

FFCR is a write-only register that controls transmit/receive FIFOs.

Bit	Bit Name	Initial Value	R/W	Description
7	RCVRTRIG1	0	W	Receive FIFO Interrupt Trigger Level 1, 0
6	RCVRTRIG0	0	W	These bits set the trigger level of the receive FIFO interrupt. 00: 1 byte 01: 4 bytes 10: 8 bytes 11: 14 bytes
5, 4	–	—	—	Reserved These bits cannot be modified.
3	DMAMODE	0	—	DMA Mode This bit is not supported and cannot be modified.
2	XMITFRST	0	W	Transmit FIFO Reset The transmit FIFO data is cleared when 1 is written. However, FRSR data is not cleared. This bit is automatically cleared.
1	RCVRFRST	0	W	Receive FIFO Reset The receive FIFO data is cleared when 1 is written. However, FTSR data is not cleared. This bit is automatically cleared.
0	FIFOE	0	W	FIFO Enable 0: Transmit/receive FIFOs disabled All bytes of these FIFOs are cleared. 1: Transmit/receive FIFOs enabled

17.3.9 Line Control Register (FLCR)

FLCR sets formats of the transmit/receive data.

Bit	Bit Name	Initial Value	R/W	Description
7	DLAB	0	R/W	Divisor Latch Address FDLL and FDLH are placed at the same addresses as the FRBR/FTHR and FIER addresses. This bit selects which register is to be accessed. 0: FRBR/FTHR and FIER access enabled 1: FDLL and FDLH access enabled
6	BREAK	0	R/W	Break Control Generates a break by driving the serial output signal FTxD low. The break state is released by clearing this bit. 0: Break released 1: Break generated
5	STICK PARITY	0	R	Stick Parity These bits are not supported in this LSI. These bits are always read as 0 and cannot be modified.
4	EPS	0	R/W	Parity Select Selects even or odd parity when the PEN bit is 1. 0: Odd parity 1: Even parity
3	PEN	0	R/W	Parity Enable Selects whether to add a parity bit for data transmission and whether to perform a parity check for data reception. 0: No parity bit added/parity check disabled 1: Parity bit added/parity check enabled

Bit	Bit Name	Initial Value	R/W	Description
2	STOP	0	R/W	Stop Bit Specifies the stop bit length for data transmission. For data reception, only the first stop bit is checked regardless of the setting. 0: 1 stop bit 1: 1.5 stop bits (data length: 5 bits) or 2 stop bits (data length: 6 to 8 bits)
1	CLS1	0	R/W	Character Length Select 1, 0
0	CLS0	0	R/W	These bits specify transmit/receive character data length. 00: Data length is 5 bits 01: Data length is 6 bits 10: Data length is 7 bits 11: Data length is 8 bits

17.3.10 Modem Control Register (FMCR)

FMCR controls output signals.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 1 and cannot be modified.
4	LOOP BACK	0	R/W	Loopback Test The transmit data output is internally connected to the receive data input, and the transmit data output pin (FRxD) becomes 1. The receive data input pin is disconnected from external sources. The four modem control input pins ($\overline{DSR}$, $\overline{CTS}$, $\overline{RI}$, and $\overline{DCD}$) are disconnected from external sources, and the pins are internally connected to the four modem control output signals ($\overline{DTR}$, $\overline{RTS}$, $\overline{OUT1}$, and $\overline{OUT2}$), respectively. The transmit data is received immediately in loopback mode. Enabling/disabling of interrupts is set by the OUT2LOOP bit in SCIFCR and FIER. 0: Loopback function disabled 1: Loopback function enabled

Bit	Bit Name	Initial Value	R/W	Description
3	OUT2	0	R/W	$\overline{\text{OUT2}}$ • Normal operation Enables or disables the SCIF interrupt. 0: Interrupt disabled 1: Interrupt enabled • Loopback test Internally connected to the $\overline{\text{DCD}}$ input pin.
2	OUT1	0	R/W	$\overline{\text{OUT1}}$ • Normal operation No effect on operation • Loopback test Internally connected to the $\overline{\text{RI}}$ input pin.
1	RTS	0	R/W	Request to Send Controls the $\overline{\text{RTS}}$ output. 0: $\overline{\text{RTS}}$ output is high level 1: $\overline{\text{RTS}}$ output is low level
0	DTR	0	R/W	Data Terminal Ready Controls the $\overline{\text{DTR}}$ output. 0: $\overline{\text{DTR}}$ output is high level 1: $\overline{\text{DTR}}$ output is low level

17.3.11 Line Status Register (FLSR)

FLSR is a read-only register that indicates the status information of data transmission.

Bit	Bit Name	Initial Value	R/W	Description
7	RXFIFOERR	0	R	Receive FIFO Error Indicates that at least one data error (parity error, framing error, or break interrupt) has occurred when the FIFO is enabled. 0: No receive FIFO error [Clearing condition] When FRBR is read or FLSR is read while there is no remaining data that could cause an error after an FIFO clear. 1: A receive FIFO error [Setting condition] When at least one data error (parity error, framing error, or break interrupt) has occurred in the FIFO
6	TEMT	1	R	Transmitter Empty Indicates whether transmit data remains. • When the FIFO is disabled 0: Transmit data remains in FTHR or FTSR. [Clearing condition] Transmit data is written to FTHR. 1: No transmit data remains in FTHR and FTSR. [Setting condition] When no transmit data remains in FTHR and FTSR. • When the FIFO is enabled 0: Transmit data remains in the transmit FIFO or FTSR. [Clearing condition] Transmit data is written to FTHR. 1: No transmit data remains in the transmit FIFO and FTSR. [Setting condition] When no transmit data remains in the transmit FIFO and FTSR

Bit	Bit Name	Initial Value	R/W	Description
5	THRE	1	R	FTHR Empty Indicates that FTHR is ready to accept new data for transmission. • When the FIFO is enabled 0: Transmit data of one or more bytes remains in the transmit FIFO. [Clearing condition] Transmit data is written to FTHR. 1: No transmit data remains in the transmit FIFO. [Setting condition] When the transmit FIFO becomes empty • When the FIFO is disabled 0: Transmit data remains in FTHR. [Clearing condition] Transmit data is written to FTHR 1: No transmit data in FTHR [Setting condition] When data transfer from FTHR to FTSR is completed
4	BI	0	R	Break Interrupt Indicates detection of the receive data break signal. When the FIFO is enabled, a break interrupt occurs in any receive data in the FIFO, and this bit is set when the receive data is in the first FIFO buffer. Reception of the next data starts after the input receive data becomes mark and a valid start bit is received. 0: Break signal not detected [Clearing condition] FLSR read 1: Break signal detected [Setting condition] When input receive data stays at space (low level) for a reception time exceeding the length of one frame

Bit	Bit Name	Initial Value	R/W	Description
3	FE	0	R	Framing Error Indicates that the stop bit of the receive data is invalid. When the FIFO is enabled, this error occurs in any receive data in the FIFO, and this bit is set when the receive data is in the first FIFO buffer. The UART attempts resynchronization after a framing error occurs. The UART, which assumes that the framing error is due to the next start bit, samples the start bit and treats it as a start bit. 0: No framing error [Clearing condition] FLSR read 1: A framing error [Setting condition] Invalid stop bit in the receive data
2	PE	0	R	Parity Error This bit indicates a parity error in the receive data when the PEN bit in FLCR is 1. When the FIFO is enabled, this error occurs in any receive data in the FIFO, and this bit is set when the receive data is in the first FIFO buffer. 0: No parity error [Clearing condition] FLSR read If this bit is set during an overrun error, read FLSR twice. 1: A parity error [Setting condition] Detection of parity error in receive data

Bit	Bit Name	Initial Value	R/W	Description
1	OE	0	R	Overrun Error Indicates occurrence of an overrun error. • When the FIFO is disabled When reception of the next data has been completed without the receive data in FRBR having been read, an overrun error occurs and the previous data is lost. • When the FIFO is enabled When the FIFO is full and reception of the next data has been completed, an overrun error occurs. The FIFO data is retained, but the last received data is lost. 0: No overrun error [Clearing condition] FLSR read 1: An overrun error [Setting condition] Occurrence of an overrun error
0	DR	0	R	Data Ready Indicates that receive data is stored in FRBR or the FIFO. 0: No receive data [Clearing condition] FRBR is read or all of the FIFO data is read. 1: Receive data remains. [Setting condition] Reception of data

17.3.12 Modem Status Register (FMSR)

FMSR is a read-only register that indicates the status of or a change in the modem control pins.

Bit	Bit Name	Initial Value	R/W	Description
7	DCD	Undefined	R	Data Carrier Detect Indicates the inverted state of the $\overline{\text{DCD}}$ input pin.
6	RI	Undefined	R	Ring Indicator Indicates the inverted state of the $\overline{\text{RI}}$ input pin.
5	DSR	Undefined	R	Data Set Ready Indicates the inverted state of the $\overline{\text{DSR}}$ input pin.
4	CTS	Undefined	R	Clear to Send Indicates the inverted state of the $\overline{\text{CTS}}$ input pin.
3	DDCD	0	R	Delta Data Carrier Indicator Indicates a change in the $\overline{\text{DCD}}$ input signal after the DDCD bit is read. 0: No change in the $\overline{\text{DCD}}$ input signal after FMSR read [Clearing condition] FMSR read 1: A change in the $\overline{\text{DCD}}$ input signal after FMSR read [Setting condition] A change in the $\overline{\text{DCD}}$ input signal
2	TERI	0	R	Trailing Edge Ring Indicator Indicates a rise in the $\overline{\text{RI}}$ input signal after the TERI bit is read. 0: No change in the $\overline{\text{RI}}$ input signal after FMSR read [Clearing condition] FMSR read 1: A rise in the $\overline{\text{RI}}$ input signal after FMSR read [Setting condition] A rise in the $\overline{\text{RI}}$ input pin

Bit	Bit Name	Initial Value	R/W	Description
1	DDSR	0	R	Delta Data Set Ready Indicator Indicates a change in the $\overline{\text{DSR}}$ input signal after the DDSR bit is read. 0: No change in the $\overline{\text{DSR}}$ input signal after FMSR read [Clearing condition] FMSR read 1: A change in the $\overline{\text{DSR}}$ input signal after FMSR read [Setting condition] A change in the $\overline{\text{DSR}}$ input signal
0	DCTS	0	R	Delta Clear to Send Indicator Indicates a change in the $\overline{\text{CTS}}$ input signal after the DCTS bit is read. 0: No change in the $\overline{\text{CTS}}$ input signal after FMSR read [Clearing condition] FMSR read 1: A change in the $\overline{\text{CTS}}$ input signal after FMSR read [Setting condition] A change in the $\overline{\text{CTS}}$ input signal

17.3.13 Scratch Pad Register (FSCR)

FSCR is not used for SCIF control, but is used to temporarily store program data.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	Bit 7 to bit 0	All 0	R/W	Temporarily stores program data.

17.3.14 SCIF Control Register (SCIFCR)

SCIFCR controls SCIF operations, and is accessible only from the CPU.

Bit	Bit Name	Initial Value	R/W	Description
7	SCIFOE1	0	R/W	These bits enable or disable PORT output of the SCIF. For details, see table 17.5.
6	SCIFOE0	0	R/W	
5	—	0	R/W	Reserved Do not change the initial value.
4	OUT2LOOP	0	R/W	Enables or disables interrupts during a loopback test. 0: Interrupt enabled 1: Interrupt disabled
3	CKSEL1	0	R/W	These bits select the clock (SCLK) to be input to the baud rate generator. 00: LCLK divided by 18 01: System clock divided by 11 10: Reserved for LCLK (not selectable) 11: Reserved for system clock (not selectable)
2	CKSEL0	0	R/W	
1	SCIFRST	0	R/W	Resets the baud rate generator, FRSR, and FTSR. 0: Normal operation 1: Reset
0	REGRST	0	R/W	Resets registers (except SCIFCR) accessible from the H8S CPU or LPC interface. 0: Normal operation 1: Reset

Table 17.5 SCIF Output Setting

Bit 3 in HICR5	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**1**
Bit 7 in SCIFCR	**0**	**0**	**1**	**1**	**0**	**0**	**1**	**1**
Bit 6 in SCIFCR	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**
PB7 and PB5 pins	PORT	PORT	SCIF	PORT	SCIF	PORT	SCIF	PORT
P50 pin	PORT	PORT	SCIF	SCIF	SCIF	SCIF	SCIF	SCIF

Note: P51, PB2 to PB4, and PB6 are input to the SCIF even when the outputs on the PB7, PB5, and P50 pins are set to PORT.

17.4 Operation

17.4.1 Baud Rate

The SCIF includes a baud rate generator and can set the desired baud rate using registers FDLH, FDLL, and the CKSEL bit in SCIFCR. Table 17.6 shows an example of baud rate settings.

Table 17.6 Example of Baud Rate Settings

	00		01		01	
CKSEL1, CKSEL0	LCLK (33 MHz) divided by 18		System Clock (20 MHz) divided by 11		System Clock (10 MHz) divided by 11	
Baud rate	FDLH, FDLL (Hex)	Error (%)	FDLH, FDLL (Hex)	Error (%)	FDLH, FDLL (Hex)	Error (%)
50	0900	0.54 %	0900	1.36 %	0480	1.36 %
75	0600	0.54 %	0600	1.36 %	0300	1.36 %
110	0417	0.54 %	0417	1.36 %	—	—
300	0180	0.54 %	0180	1.36 %	00C0	1.36 %
600	00C0	0.54 %	00C0	1.36 %	0060	1.36 %
1200	0060	0.54 %	0060	1.36 %	0030	1.36 %
1800	0040	0.54 %	0040	1.36 %	0020	1.36 %
2400	0030	0.54 %	0030	1.36 %	0018	1.36 %
4800	0018	0.54 %	0018	1.36 %	000C	1.36 %
9600	000C	0.54 %	000C	1.36 %	0006	1.36 %
14400	0008	0.54 %	0008	1.36 %	0004	1.36 %
19200	0006	0.54 %	0006	1.36 %	0003	1.36 %
38400	0003	0.54 %	0003	1.36 %	—	—
57600	0002	0.54 %	0002	1.36 %	0001	1.36 %
115200	0001	0.54 %	0001	1.36 %	—	—

17.4.2 Operation in Asynchronous Communication

Figure 17.2 illustrates the typical format for asynchronous serial communication. One frame consists of a start bit (low level), followed by transmit/receive data (LSB-first: from the least significant bit), a parity bit, and a stop bit (high level). In asynchronous serial communication, the transmission line is usually held high in the mark state (high level). The SCIF monitors the transmission line, and when it detects the space state (low level), recognizes a start bit and starts serial communication. Inside the SCIF, the transmitter and receiver are independent units, enabling full-duplex communication. Both of the transmitter and receiver also have a 16-stage FIFO buffered structure so that data can be read or written during transmission or reception, enabling continuous data transmission and reception.

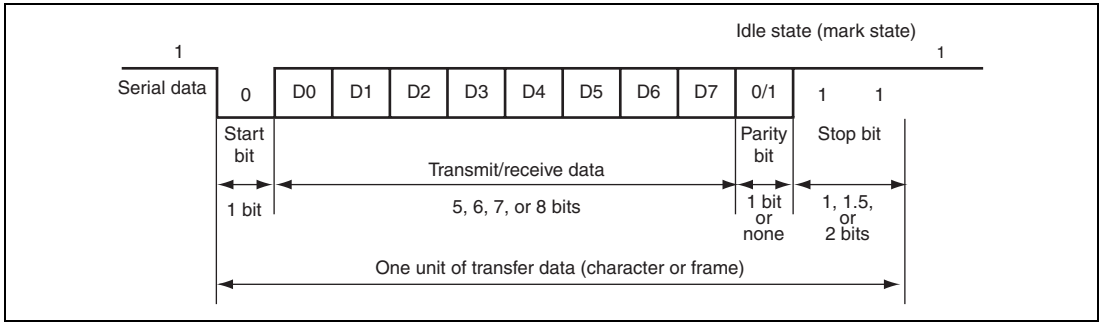


Figure 17.2 Data Format in Serial Transmission/Reception (Example with 8-Bit Data, Parity and 2 Stop Bits)

17.4.3 Initialization of the SCIF

(1) Initialization of the SCIF

Use an example of the flowchart in figure 17.3 to initialize the SCIF before transmitting or receiving data.

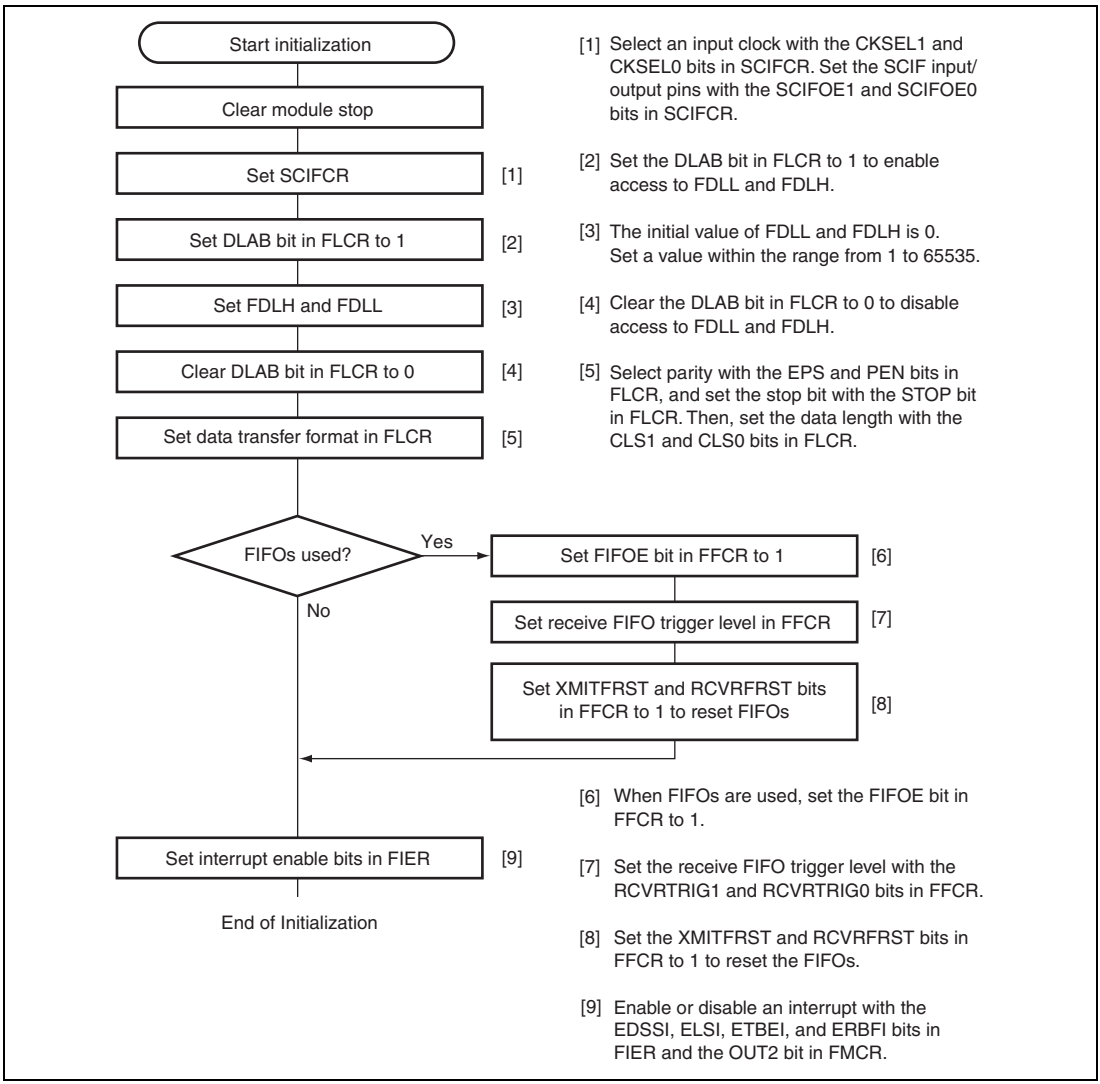
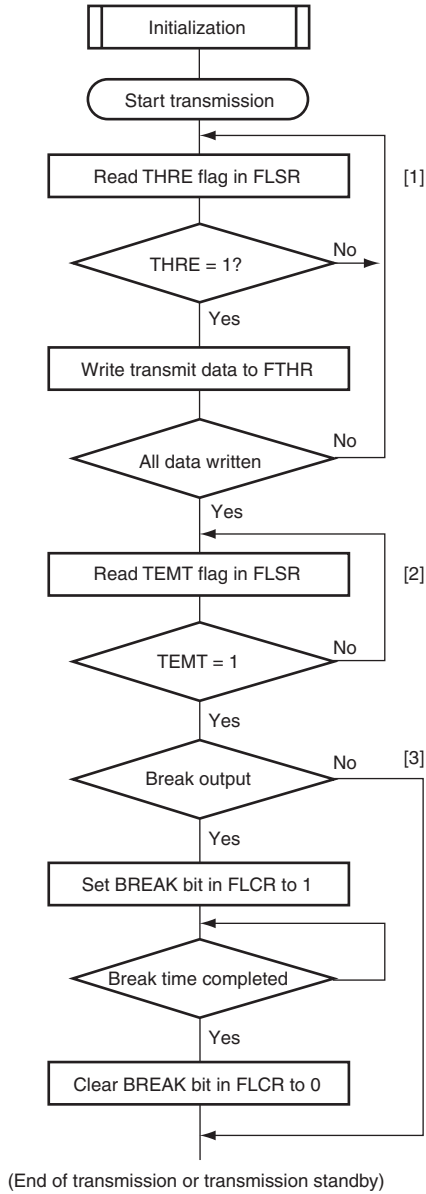


Figure 17.3 Example of Initialization Flowchart

(2) Serial Data Transmission

Figure 17.4 shows an example of the data transmission flowchart.



[1] Confirm that the THRE flag in FLSR is 1, and write transmit data to FTHR. When FIFOs are used, write 1-byte to 16-byte transmit data. When the OUT2 bit in FMCR and the ETBEI bit in FIER are set to 1, an FTHR empty interrupt occurs. When data is written to FTHR, it is transferred automatically to FTSR. The data is then transmitted from the FTxD pin in the order of the start bit, transmit data, parity bit, and stop bit.

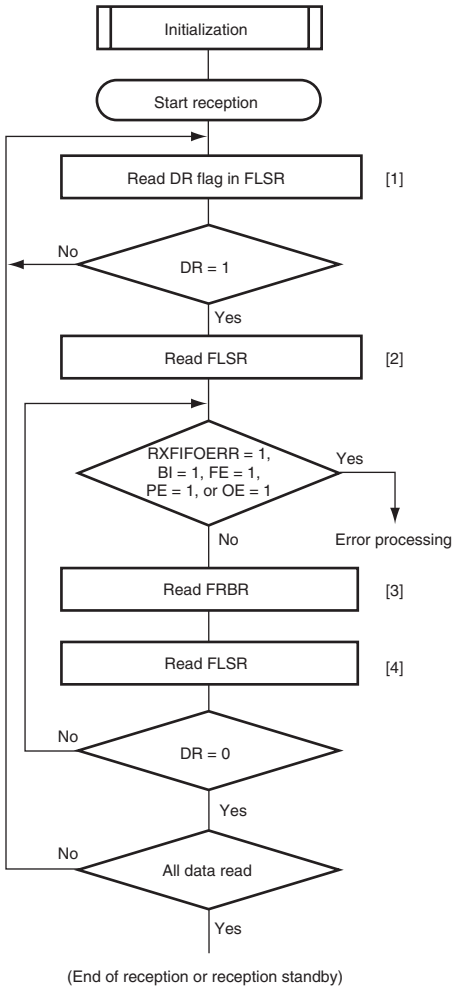
[2] Read the TEMT flag in FLSR, and confirm that TEMT is set to 1 to ensure that all transmit data has been transmitted.

[3] To output a break at the end of serial transmission, set the BREAK bit in FLCR to 1. After completion of the break time, clear the BREAK bit in FLCR to 0 to clear the break.

Figure 17.4 Example of Data Transmission Flowchart

(3) Serial Data Reception

Figure 17.5 shows an example of the data reception flowchart.



[1] Confirm that the DR flag in FLSR is 1 to ensure that receive data is in the buffer. When the OUT2 bit in FMCR and the ERBFI bit in FIER are set to 1, a receive data ready interrupt occurs.

[2] Read the RXFIFOERR, BI, FE, PE, and OE flags in FLSR to ensure that no error has occurred. If an error has occurred, perform error processing. When the OUT2 bit in FMCR and the ELSI bit in FIER are set to 1, a receive line status interrupt occurs.

[3] Read the receive data in FRBR.

[4] Check the DR flag in FLSR. When the DR flag is cleared to 0 and all data has been read, data reception is complete.

Figure 17.5 Example of Data Reception Flowchart

17.4.4 Data Transmission/Reception with Flow Control

The following shows examples of data transmission/reception for flow control using CTS and RTS.

(1) Initialization

Figure 17.6 shows an example of the initialization flowchart.

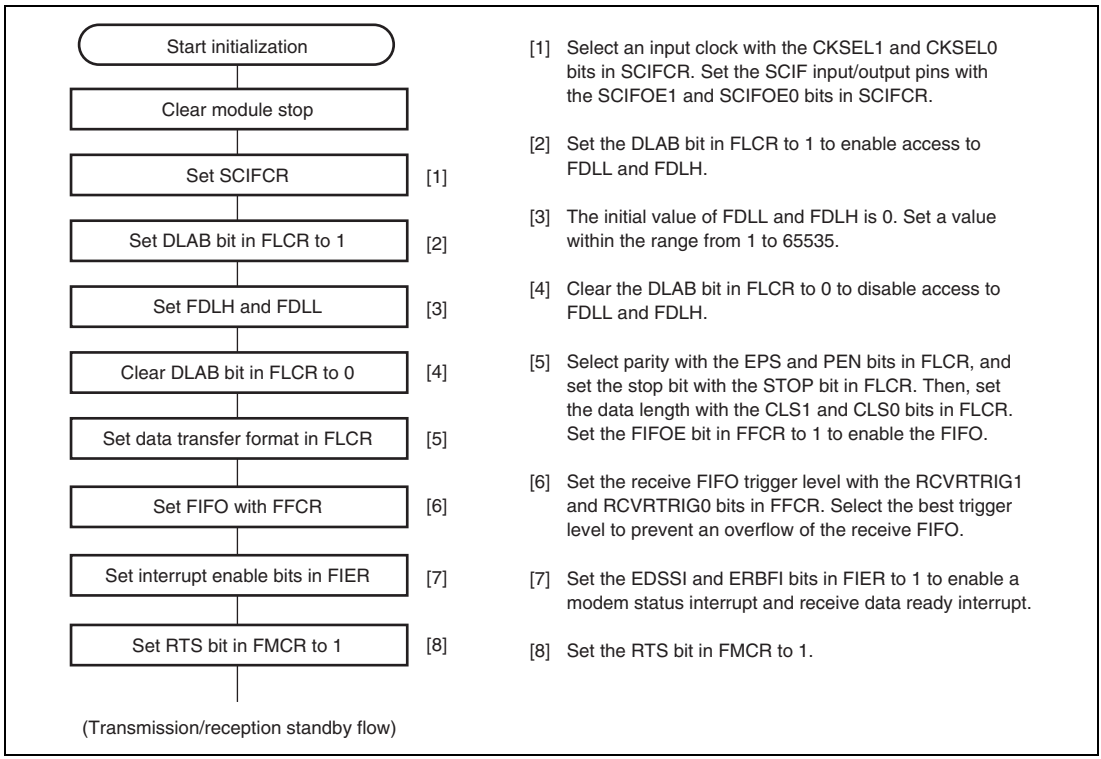


Figure 17.6 Example of Initialization Flowchart

(2) Data Transmission/Reception Standby

Figure 17.7 shows an example of the data transmission/reception standby flowchart.

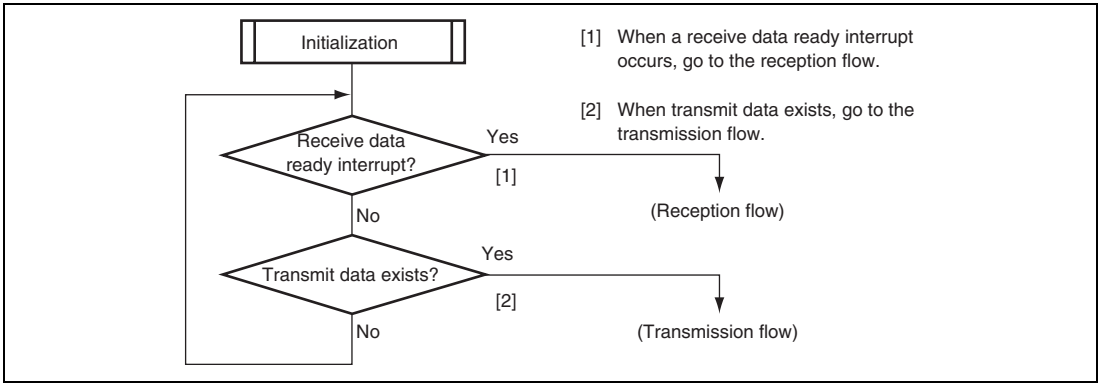


Figure 17.7 Example of Data Transmission/Reception Standby Flowchart

(3) Data Transmission

Figure 17.8 shows an example of the data transmission flowchart.

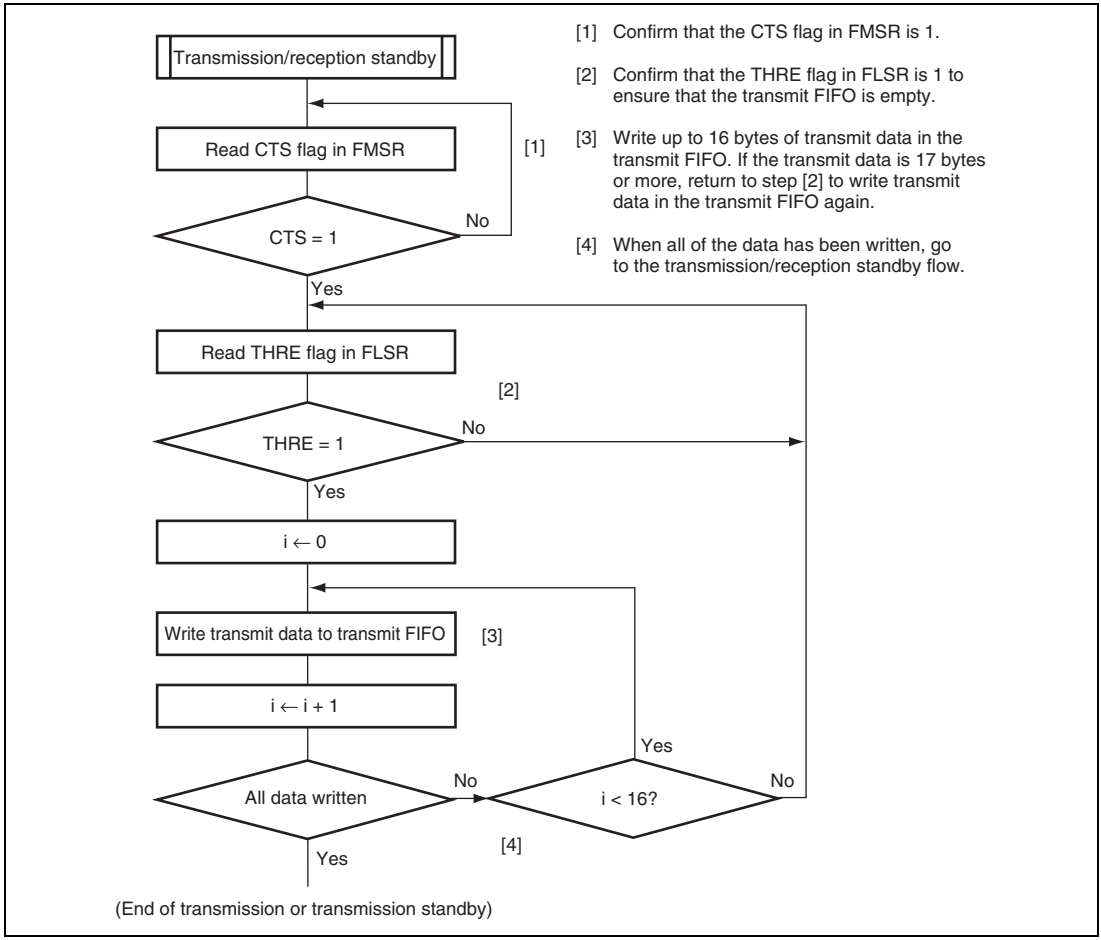


Figure 17.8 Example of Data Transmission Flowchart

(4) Suspension of Data Transmission

Figure 17.9 shows an example of the data transmission suspension flowchart.

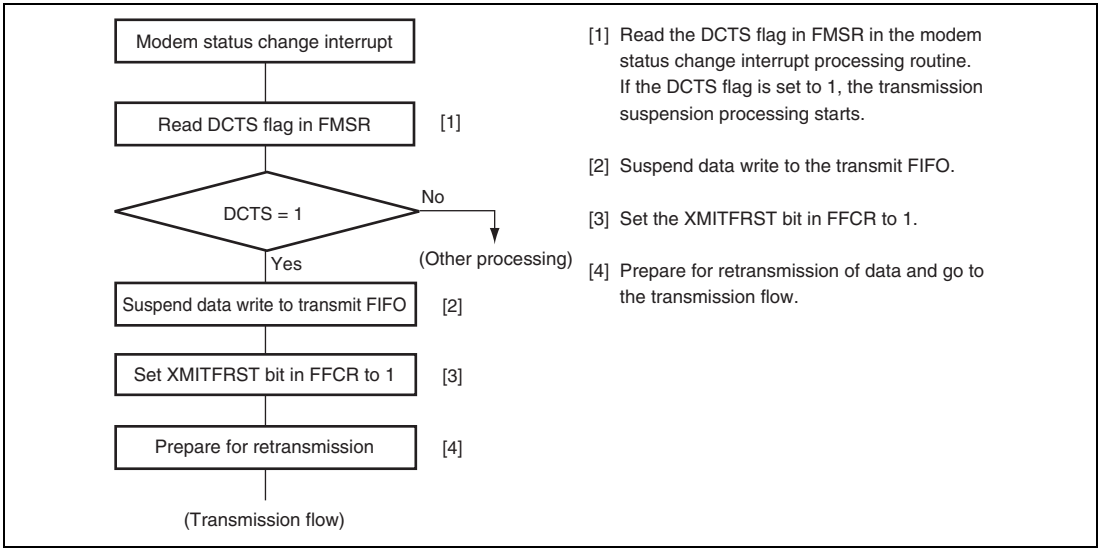
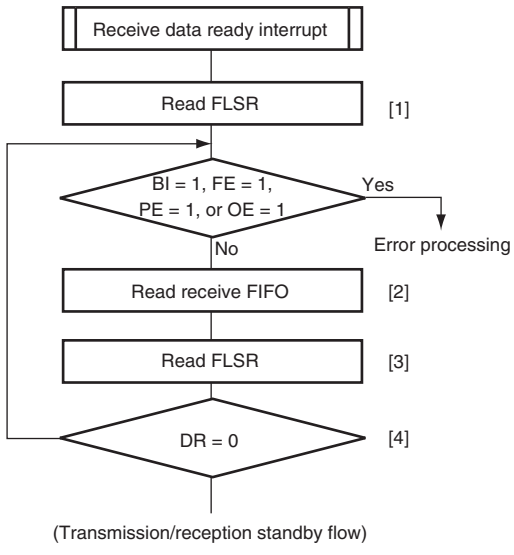


Figure 17.9 Example of Data Transmission Suspension Flowchart

(5) Data Reception

Figure 17.10 shows an example of the data reception flowchart.



[1] When data is received, a receive data ready interrupt occurs. Go to the data reception flow by using this interrupt trigger.

[2] Confirm that the BI, FE, PE, and OE flags in FLSR are all cleared. If any one of these flags is set to 1, perform error processing.

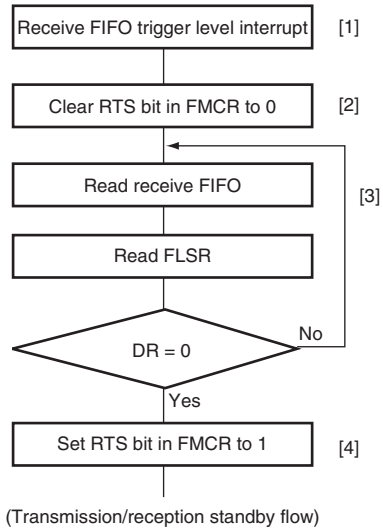
[3] Read the receive FIFO.

[4] Check the DR flag in FLSR. When the DR flag is cleared and all of the data has been read, data reception is complete.

Figure 17.10 Example of Data Reception Flowchart

(6) Suspension of Data Reception

Figure 17.11 shows an example of the data reception suspension flowchart.



[1] When data is received at a trigger level higher than the receive FIFO trigger level specified in the initialization flow, a receive FIFO trigger level interrupt occurs.

[2] Clear the RTS bit in FMCR to 0.

[3] Read the receive FIFO until the DR flag is cleared to 0.

[4] Set the RTS bit in FMCR to 1, and then go to the transmission/reception standby flow.

Figure 17.11 Example of Data Reception Suspension Flowchart

17.4.5 Data Transmission/Reception Through the LPC Interface

As shown in table 17.3, setting the SCIFE bit in HICR5 to 1 allows registers (except SCIFCR) to be accessed from the LPC interface. The initial setting of SCIFCR by the CPU and setting of the SCIFE bit in HICR5 to 1 enable the flow settings for initialization and data transmission/reception shown in figures 17.3 to 17.5 to be made from the LPC interface. Table 17.7 shows the correspondence between LPC interface I/O address and access to the SCIF registers. For details of the LPC interface settings, see section 20, LPC interface (LPC).

Table 17.7 Correspondence Between LPC Interface I/O Address and the SCIF Registers

LPC Interface I/O Address						
Bits 15 to 3	**Bit 2**	**Bit 1**	**Bit 0**	**R/W**	**Condition**	**SCIF Register**
SCIFADR (bits 15 to 3)	0	0	0	R	FLCR[7] = 0	FRBR
				W	FLCR[7] = 0	FTHR
				R/W	FLCR[7] = 1	FDLL
SCIFADR (bits 15 to 3)	0	0	1	R/W	FLCR[7] = 0	FIER
				R/W	FLCR[7] = 1	FDLH
SCIFADR (bits 15 to 3)	0	1	0	R	—	FIIR
				W	—	FFCR
SCIFADR (bits 15 to 3)	0	1	1	R/W	—	FLCR
SCIFADR (bits 15 to 3)	1	0	0	R/W	—	FMCR
SCIFADR (bits 15 to 3)	1	0	1	R	—	FLSR
SCIFADR (bits 15 to 3)	1	1	0	R	—	FMSR
SCIFADR (bits 15 to 3)	1	1	1	R/W	—	FSCR

Table 17.8 shows the range of initialization of the registers related to data transmission/reception through the LPC interface, making a classification by each mode.

Table 17.8 Register States

Register		System Reset	SCIFRST	REGRST	LPC Reset	LPC Shutdown	LPC Abort
SCIFADRH	Bits 15 to 8	Initialized	Retained	Retained	Retained	Retained	Retained
SCIFADRL	Bits 7 to 0	Initialized	Retained	Retained	Retained	Retained	Retained
HICR5	SCIFE	Initialized	Retained	Retained	Retained	Retained	Retained
SIRQCR4	Bits 7 to 4, SCSIRQ3 to 0	Initialized	Retained	Retained	Retained	Retained	Retained
SCIFCR	SCIFOE1, SCIFOE0, OUT2LOOP, CKSEL1, CKSEL0, SCIFRST, REGRST	Initialized	Retained	Retained	Retained	Retained	Retained
FRBR	Bits 7 to 0	Initialized	Retained	Initialized	Initialized	Retained	Retained
FTHR	Bits 7 to 0	Initialized	Retained	Initialized	Initialized	Retained	Retained
FDLL	Bits 7 to 0	Initialized	Retained	Initialized	Initialized	Retained	Retained
FDLH	Bits 7 to 0	Initialized	Retained	Initialized	Initialized	Retained	Retained
FIIR	FIFOE1, FIFOE0, INTID2 to INTID0, INTPEND	Initialized	Retained	Initialized	Initialized	Retained	Retained
FFCR	RCVRTRIG1, RCVRTRIG0, XMITFRST, RCVRFRST, FIFOE	Initialized	Retained	Initialized	Initialized	Retained	Retained
FLCR	DLAB, TREAK, EPS, PEN, STOP, CLS1, CLS0	Initialized	Retained	Initialized	Initialized	Retained	Retained

Register		System Reset	SCIFRST	REGRST	LPC Reset	LPC Shutdown	LPC Abort
FMCR	LOOP BACK, OUT2, OUT1, RTS, DTR	Initialized	Retained	Initialized	Initialized	Retained	Retained
FLSR	RXFEFOERR, TEMT, THRE, BI, FE, PE, OE, DR	Initialized	Retained	Initialized	Initialized	Retained	Retained
FMSR	DDCD, TERI, DDSR, DCTS	Initialized	Retained	Initialized	Initialized	Retained	Retained
FSCR	Bits 7 to 0	Initialized	Retained	Initialized	Initialized	Retained	Retained
SCIF transmission sequencer (inner state)	—	Initialized	Initialized	Retained	Initialized	Retained	Retained

17.5 Interrupt Sources

Table 17.9 lists the interrupt sources. A common interrupt vector is assigned to each interrupt source.

When the LPC uses the SCIF, the LPC does not request any interrupts to be sent to the H8S CPU. The SERIRQ signal of the LPC interface transmits an interrupt request to the host.

Table 17.9 Interrupt Sources

Interrupt Name	Interrupt Source	Priority
Receive line status	Overrun error, parity error, framing error, break interrupt	High
Receive data ready	Acceptance of receive data, FIFO trigger level	
Character timeout (when FIFO is enabled)	No data is input to or output from the receive FIFO for the 4-character time period while one or more characters remain in the receive FIFO.	
FTHR empty	FTHR empty	
Modem status	CTS, DSR, RI, DCD	Low

Table 17.10 shows the interrupt source, vector address, and interrupt priority.

Table 17.10 Interrupt Source, Vector Address, and Interrupt Priority

Interrupt Origin of Interrupt Source	Interrupt Name	Vector Number	Vector Address	ICR
SCIF	SCIF (SCIF interrupt)	82	H'000148	ICRC7

17.6 Usage Note

17.6.1 Power-Down Mode When LCLK is Selected for SCLK

To switch to watch mode or software standby mode when LCLK divided by 18 has been selected for SCLK, use the shutdown function of the LPC interface to stop LCLK.

Section 18 I²C Bus Interface (IIC)

This LSI has a three-channel I²C bus interface. The I²C bus interface conforms to and provides a subset of the Philips I²C bus (inter-IC bus) interface functions. The register configuration that controls the I²C bus differs partly from the Philips configuration, however.

18.1 Features

- Selection of addressing format or non-addressing format
  - I²C bus format: addressing format with an acknowledge bit, for master/slave operation
  - Clocked synchronous serial format: non-addressing format without an acknowledge bit, for master operation only
- Conforms to Philips I²C bus interface (I²C bus format)
- Two ways of setting slave address (I²C bus format)
- Start and stop conditions generated automatically in master mode (I²C bus format)
- Selection of the acknowledge output level in reception (I²C bus format)
- Automatic loading of an acknowledge bit in transmission (I²C bus format)
- Wait function in master mode (I²C 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 (I²C 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 I²C bus format occurs, when ICDR data is transferred from ICDRT to ICDRS or from ICDRS to ICDRR, or during a wait state)
  - Address match: When any slave address matches or the general call address is received in slave receive mode with I²C bus format (including address reception after loss of master arbitration)
  - Arbitration lost
  - Start condition detection (in master mode)
  - Stop condition detection (in slave mode)

- Selection of 16 internal clocks (in master mode)
- Direct bus drive (SCL/SDA pin)
  - Ten pins—P52/SCL0, P97/SDA0, P86/SCL1, P42/SDA1, PG2/SDA2, PG3/SCL2, PG4/ExSDAA, PG5/ExSCLA, PG6/ExSDAB, and PG7/ExSCLB —(normally NMOS push-pull outputs) function as NMOS open-drain outputs when the bus drive function is selected.

Note: When using this IIC module, make sure to set bits HNDS, FNC1, and FNC0 in ICXR to 1 in the initial settings. If other settings are made, restrictions on operation that are not covered in this manual will apply.

Figure 18.1 shows a block diagram of the I²C bus interface. Figure 18.2 shows an example of I/O pin connections to external circuits. Since I²C bus interface I/O pins are different in structure from normal port pins, they have different specifications for permissible applied voltages. For details, see section 28, Electrical Characteristics.

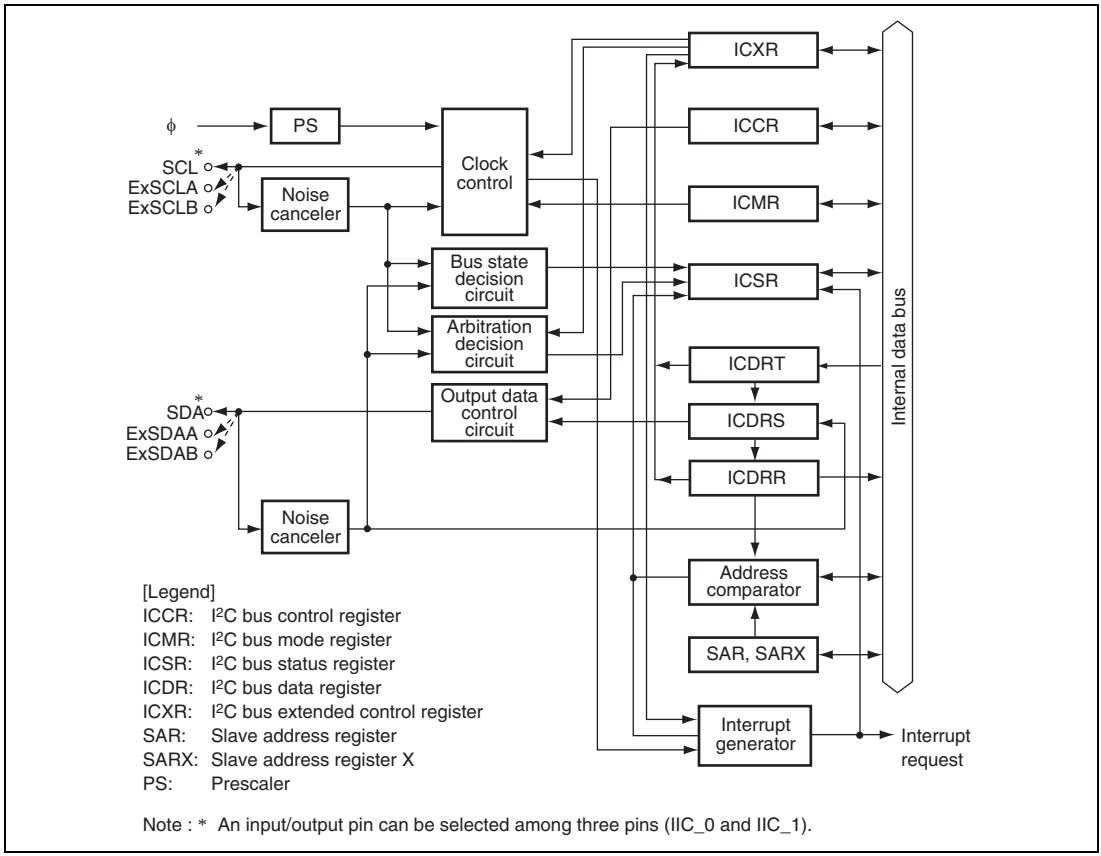


Figure 18.1 Block Diagram of I²C Bus Interface

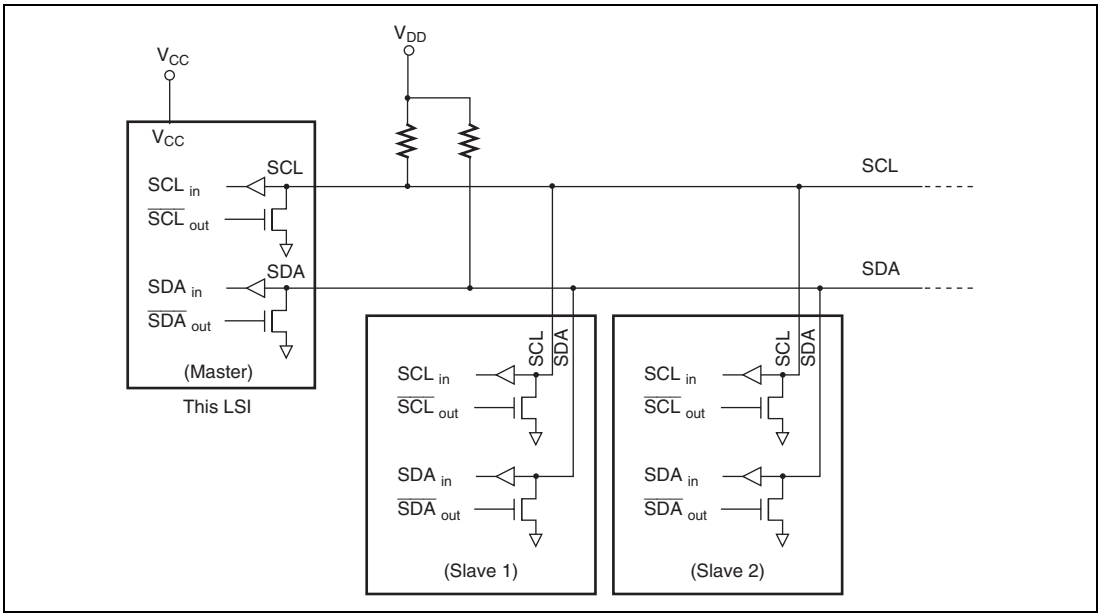


Figure 18.2 I²C Bus Interface Connections (Example: This LSI as Master)

18.2 Input/Output Pins

Table 18.1 summarizes the input/output pins used by the I²C bus interface.

One of three pins can be specified as SCL and SDA input/output pin for IIC_0 and IIC_1. Two or more input/output pins should not be specified for one channel.

For the method of setting pins, see section 7.3.2, Port Control Register 1 (PTCNT1).

Table 18.1 Pin Configuration

Channel	Symbol*	Input/Output	Function
0	SCL0	Input/Output	Serial clock input/output pin of IIC_0
	SDA0	Input/Output	Serial data input/output pin of IIC_0
1	SCL1	Input/Output	Serial clock input/output pin of IIC_1
	SDA1	Input/Output	Serial data input/output pin of IIC_1
2	SCL2	Input/Output	Serial clock input/output pin of IIC_2
	SDA2	Input/Output	Serial data input/output pin of IIC_2
—	ExSCLA	Input/Output	Serial clock input/output pin of IIC_0 or IIC_1
	ExSDAA	Input/Output	Serial data input/output pin of IIC_0 or IIC_1
	ExSCLB	Input/Output	Serial clock input/output pin of IIC_0 or IIC_1
	ExSDAB	Input/Output	Serial data input/output pin of IIC_0 or IIC_1

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

18.3 Register Descriptions

The I²C bus interface has the following registers. Registers ICDR and SARX and registers ICMR and SAR are allocated to the same addresses. Accessible registers differ depending on the ICE bit in ICCR. When the ICE bit is cleared to 0, SAR and SARX can be accessed, and when the ICE bit is set to 1, ICMR and ICDR can be accessed. For details on the serial timer control register, see section 3.2.3, Serial Timer Control Register (STCR).

Table 18.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Data Bus Width
Channel 0	I²C bus extended control register_0	ICXR_0	R/W	H'00	H'FED4	8
	I²C bus control register_0	ICCR_0	R/W	H'01	H'FFD8	8
	I²C bus status register_0	ICSR_0	R/W	H'00	H'FFD9	8
	I²C bus data register_0	ICDR_0	R/W	—	H'FFDE	8
	Second slave address register_0	SARX_0	R/W	H'01	H'FFDE	8
	I²C bus mode register_0	ICMR_0	R/W	H'00	H'FFDF	8
	Slave address register_0	SAR_0	R/W	H'00	H'FFDF	8
	I²C bus control initialization register_0	ICRES_0	R/W	H'0F	H'FEE6	8
Channel 1	I²C bus extended control register_1	ICXR_1	R/W	H'00	H'FED5	8
	I²C bus control register_1	ICCR_1	R/W	H'01	H'FF88 H'FED0*	8
	I²C bus status register_1	ICSR_1	R/W	H'00	H'FF89 H'FED1*	8
	I²C bus data register_1	ICDR_1	R/W	—	H'FF8E H'FECE*	8
	Second slave address register_1	SARX_1	R/W	H'01	H'FF8E H'FECE*	8
	I²C bus mode register_1	ICMR_1	R/W	H'00	H'FF8F H'FECF*	8
	Slave address register_1	SAR_1	R/W	H'00	H'FF8F H'FECF*	8

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Data Bus Width
Channel 2	I²C bus extended control register_2	ICXR_2	R/W	H'00	H'FE8C	8
	I²C bus control register_2	ICCR_2	R/W	H'01	H'FE88	8
	I²C bus status register_2	ICSR_2	R/W	H'00	H'FE89	8
	I²C bus data register_2	ICDR_2	R/W	—	H'FE8E	8
	Second slave address register_2	SARX_2	R/W	H'01	H'FE8E	8
	I²C bus mode register_2	ICMR_2	R/W	H'00	H'FE8F	8
	Slave address register_2	SAR_2	R/W	H'00	H'FE8F	8
	I²C bus control initialization register_2	ICRES_2	R/W	H'0F	H'FE8A	8

Note: * Upper address: when RELOCATE = 0
Lower address: when RELOCATE = 1

18.3.1 I²C Bus Data Register (ICDR)

ICDR is an 8-bit readable/writable register that is used as a transmit data register when transmitting and a receive data register when receiving. ICDR is internally divided into a shift register (ICDRS), receive buffer (ICDRR), and transmit buffer (ICDRT). Data transfers among these three registers are performed automatically in accordance with changes in the bus state, and they 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 be performed after start condition detection. When the start condition is detected, previous write data is ignored. In slave transmit mode, writing should be performed after the slave addresses match and the TRS bit is automatically changed to 1.

In transmit mode (TRS = 1), transmit data can be written to ICDRT when the ICDRE flag is 1. After the transmit data has been written to ICDRT, the ICDRE flag is cleared to 0. Then, when ICDRS becomes empty on completion of the previous transmission, the data are automatically transferred from ICDRT to ICDRS and the ICDRE flag is set to 1. As long as ICDRS contains data to be transmitted or data being transmitted, data written to ICDRT are retained there.

In receive mode (TRS = 0), data is not transferred from ICDRT to ICDRS. Thus, do not write to ICDRT when in this mode.

In receive mode (TRS = 0), data received in ICDRR can be read when the ICDRF flag is 1. After the data has been read from ICDRR, the ICDRF flag is cleared to 0. Each time ICDRS contains data on completion of one round of reception, the data is automatically transferred from ICDRS to ICDRR and the ICDRF flag is set to 1. If ICDRR contains receive data that hasn't been read out, any further receive data is retained in ICDRS.

Since data are not transferred from ICDRS to ICDRR in transmit mode (TRS = 1), do not read ICDRR in transmit mode (excluding the case where final receive data is read out in the recommended operation flow of master receive mode).

If the number of bits in a frame, excluding the acknowledge bit, is less than eight, transmit data and receive data are stored differently. Transmit data should be written justified toward the MSB side when MLS = 0 in ICMR, and toward the LSB side when MLS = 1. Receive data bits should be read from the LSB side when MLS = 0, and from the MSB side when MLS = 1.

ICDR can be written to and read from only when the ICE bit is set to 1 in ICCR. The initial value of ICDR is undefined.

18.3.2 Slave Address Register (SAR)

SAR sets the slave address and selects the communication format. If the LSI is in slave mode with the I²C bus format selected, when the FS bit is set to 0 and the upper 7 bits of SAR match the upper 7 bits of the first frame received after a start condition, the LSI operates as the slave device specified by the master device. SAR can be accessed only when the ICE bit in ICCR is cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
7	SVA6	0	R/W	Slave Address 6 to 0
6	SVA5	0	R/W	Set a slave address.
5	SVA4	0	R/W	
4	SVA3	0	R/W	
3	SVA2	0	R/W	
2	SVA1	0	R/W	
1	SVA0	0	R/W	
0	FS	0	R/W	Format Select Selects the communication format together with the FSX bit in SARX. See table 18.3. This bit should be set to 0 when general call address recognition is performed.

18.3.3 Second Slave Address Register (SARX)

SARX sets the second slave address and selects the communication format. If the LSI is in slave mode with the I²C bus format selected, when the FSX bit is set to 0 and the upper 7 bits of SARX match the upper 7 bits of the first frame received after a start condition, the LSI operates as the slave device specified by the master device. SARX can be accessed only when the ICE bit in ICCR is cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
7	SVAX6	0	R/W	Second Slave Address 6 to 0
6	SVAX5	0	R/W	Set the second slave address.
5	SVAX4	0	R/W	
4	SVAX3	0	R/W	
3	SVAX2	0	R/W	
2	SVAX1	0	R/W	
1	SVAX0	0	R/W	
0	FSX	1	R/W	Format Select X Selects the communication format together with the FS bit in SAR. See table 18.3.

Table 18.3 Communication Format

SAR	SARX	
FS	**FSX**	**Operating Mode**
0	0	I²C bus format • SAR and SARX slave addresses recognized • General call address recognized
	1	I²C bus format • SAR slave address recognized • SARX slave address ignored • General call address recognized
1	0	I²C bus format • SAR slave address ignored • SARX slave address recognized • General call address ignored
	1	Clocked synchronous serial format • SAR and SARX slave addresses ignored • General call address ignored

- I²C bus format: addressing format with an acknowledge bit
- Clocked synchronous serial format: non-addressing format without an acknowledge bit, for master mode only

18.3.4 I²C Bus Mode Register (ICMR)

ICMR sets the communication format and transfer rate. It can only be accessed when the ICE bit in ICCR is set to 1.

Bit	Bit Name	Initial Value	R/W	Description
7	MLS	0	R/W	MSB-First/LSB-First Select 0: MSB-first 1: LSB-first Set this bit to 0 when the I²C bus format is used.
6	WAIT	0	R/W	Wait Insertion Bit This bit is valid only in master mode with the I²C bus 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 (8th clock), the IRIC flag is set to 1 in ICCR, and a wait state begins (with SCL at the low level). When the IRIC flag is cleared to 0 in ICCR, the wait ends and the acknowledge bit is transferred. For details, see section 18.4.7, IRIC Setting Timing and SCL Control.
5	CKS2	0	R/W	Transfer Clock Select 2 to 0 These bits are used only in master mode. These bits select the required transfer rate, together with the IICX2 (IIC_2), IICX1 (IIC_1), and IICX0 (IIC_0) bits in STCR. See table 18.4.
4	CKS1	0	R/W	
3	CKS0	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
2	BC2	0	R/W	Bit Counter 2 to 0
1	BC1	0	R/W	These bits specify the number of bits to be transferred next. Bit BC2 to BC0 settings should be made during an interval between transfer frames. If bits BC2 to BC0 are set to a value other than 000, the setting should be made while the SCL line is low.
0	BC0	0	R/W	The bit counter is initialized to B'000 when a start condition is detected. The value returns to B'000 at the end of a data transfer.
				I²C Bus Format / Clocked Synchronous Serial Mode
				000: 9 bits / 000: 8 bits
				001: 2 bits / 001: 1 bits
				010: 3 bits / 010: 2 bits
				011: 4 bits / 011: 3 bits
				100: 5 bits / 100: 4 bits
				101: 6 bits / 101: 5 bits
				110: 7 bits / 110: 6 bits
				111: 8 bits / 111: 7 bits

Table 18.4 I²C Transfer Rate

STCR	ICMR							
Bits 5, 6, and 7	Bit 5	Bit 4	Bit 3		Transfer Rate			
IICX	CKS2	CKS1	CKS0	Clock	ϕ = 8 MHz	ϕ = 10 MHz	ϕ = 16 MHz	ϕ = 20 MHz
0	0	0	0	ϕ/28	286 kHz	357 kHz	571 kHz*	714 kHz*
0	0	0	1	ϕ/40	200 kHz	250 kHz	400 kHz	500 kHz*
0	0	1	0	ϕ/48	167 kHz	208 kHz	333 kHz	417 kHz*
0	0	1	1	ϕ/64	125 kHz	156 kHz	250 kHz	313 kHz
0	1	0	0	ϕ/80	100 kHz	125 kHz	200 kHz	250 kHz
0	1	0	1	ϕ/100	80.0 kHz	100 kHz	160 kHz	200 kHz
0	1	1	0	ϕ/112	71.4 kHz	89.3 kHz	143 kHz	179 kHz
0	1	1	1	ϕ/128	62.5 kHz	78.1 kHz	125 kHz	156 kHz
1	0	0	0	ϕ/56	143 kHz	179 kHz	286 kHz	357 kHz
1	0	0	1	ϕ/80	100 kHz	125 kHz	200 kHz	250 kHz
1	0	1	0	ϕ/96	83.3 kHz	104 kHz	167 kHz	208 kHz
1	0	1	1	ϕ/128	62.5 kHz	78.1 kHz	125 kHz	156 kHz
1	1	0	0	ϕ/160	50.0 kHz	62.5 kHz	100 kHz	125 kHz
1	1	0	1	ϕ/200	40.0 kHz	50.0 kHz	80.0 kHz	100 kHz
1	1	1	0	ϕ/224	35.7 kHz	44.6 kHz	71.4 kHz	89.3 kHz
1	1	1	1	ϕ/256	31.3 kHz	39.1 kHz	62.5 kHz	78.1 kHz

Notes: n = 0, 1, or 2

* Correct operation cannot be guaranteed since the transfer rate is beyond the I²C bus interface specification (normal mode: maximum 100 kHz, high-speed mode: maximum 400 kHz).

18.3.5 I²C Bus Control Register (ICCR)

ICCR controls the I²C bus interface and performs interrupt flag confirmation.

Bit	Bit Name	Initial Value	R/W	Description
7	ICE	0	R/W	I²C Bus Interface Enable 0: I²C bus interface modules are stopped and I²C bus interface module internal state is initialized. SAR and SARX can be accessed. 1: I²C bus interface modules can perform transfer operation, and the ports function as the SCL and SDA input/output pins. ICMR and ICDR can be accessed.
6	IEIC	0	R/W	I²C Bus Interface Interrupt Enable 0: Disables interrupts from the I²C bus interface to the CPU 1: Enables interrupts from the I²C bus interface to the CPU.
5	MST	0	R/W	Master/Slave Select
4	TRS	0	R/W	Transmit/Receive Select MST TRS 0 0: Slave receive mode 0 1: Slave transmit mode 1 0: Master receive mode 1 1: Master transmit mode Both these bits will be cleared by hardware when they lose in a bus contention in master mode with the I²C bus format. In slave receive mode with I²C bus format, the R/$\overline{W}$ bit in the first frame immediately after the start condition sets these bits in receive mode or transmit mode automatically by hardware. Modification of the TRS bit during transfer is deferred until transfer is completed, and the changeover is made after completion of the transfer (at the rising edge of the 9th clock).

Bit	Bit Name	Initial Value	R/W	Description
5	MST	0	R/W	[MST clearing conditions]
4	TRS	0	R/W	1. When 0 is written by software 2. When lost in bus contention in I²C bus format master mode [MST setting conditions] 1. When 1 is written by software (for MST clearing condition 1) 2. When 1 is written in MST after reading MST = 0 (for MST clearing condition 2) [TRS clearing conditions] 1. When 0 is written by software (except for TRS setting condition 3) 2. When 0 is written in TRS after reading TRS = 1 (for TRS setting condition 3) 3. When lost in bus contention in I²C bus format master mode [TRS setting conditions] 1. When 1 is written by software (except for TRS clearing condition 3) 2. When 1 is written in TRS after reading TRS = 0 (for TRS clearing condition 3) 3. When 1 is received as the R/$\overline{W}$ bit after the first frame address matching in I²C bus format slave mode
3	ACKE	0	R/W	Acknowledge Bit Decision and Selection 0: The value of the acknowledge bit is ignored, and continuous transfer is performed. The value of the received acknowledge bit is not indicated by the ACKB bit in ICSR, which is always 0. 1: If the received acknowledge bit is 1, continuous transfer is halted. Depending on the receiving device, the acknowledge bit may be significant, in indicating completion of processing of the received data, for instance, or may be fixed at 1 and have no significance.

Bit	Bit Name	Initial Value	R/W	Description
2	BBSY	0	R/W*	Bus Busy
0	SCP	1	W	Start Condition/Stop Condition Prohibit In master mode: • Writing 0 in BBSY and 0 in SCP: A stop condition is issued • Writing 1 in BBSY and 0 in SCP: A start condition and a restart condition are issued In slave mode: • Writing to the BBSY flag is disabled. [BBSY setting condition] When the SDA level changes from high to low under the condition of SCL = high, assuming that the start condition has been issued. [BBSY clearing condition] When the SDA level changes from low to high under the condition of SCL = high, assuming that the stop condition has been issued. To issue a start/stop condition, use the MOV instruction. The I²C bus interface must be set in master transmit mode before the issue of a start condition. Set MST to 1 and TRS to 1 before writing 1 in BBSY and 0 in SCP. The BBSY flag can be read to check whether the I²C bus (SCL, SDA) is busy or free. The SCP bit is always read as 1. If 0 is written, the data is not stored.

Note: * The value in BBSY flag does not change even if written.

Bit	Bit Name	Initial Value	R/W	Description
1	IRIC	0	R/(W)*	I²C Bus Interface Interrupt Request Flag Indicates that the I²C bus interface has issued an interrupt request to the CPU. IRIC is set at different times depending on the FS bit in SAR, the FSX bit in SARX, and the WAIT bit in ICMR. See section 18.4.7, IRIC Setting Timing and SCL Control. The conditions under which IRIC is set also differ depending on the setting of the ACKE bit in ICCR. [Setting conditions] All operating modes: 1. When a start condition is detected in transmit mode and the ICDRE flag is set to 1 2. When data is transferred from ICDRT to ICDRS in transmit mode and the ICDRE flag is set to 1 3. When data is transferred from ICDRS to ICDRR in receive mode and the ICDRF flag is set to 1 4. If 1 is received as the acknowledge bit (when the ACKE bit is 1 in transmit mode) at the completion of data transmission I²C bus format master mode: 1. When a wait is inserted between the data and acknowledge bit when the WAIT bit is 1 2. When the AL flag is set to 1 after bus arbitration is lost while the ALIE bit is 1 I²C bus format slave mode: 1. When the slave address (SVA or SVAX) matches after the reception of the first frame following the start condition and the AAS flag or AASX flag is set to 1 2. When the general call address is detected after the reception of the first frame following the start condition and the ADZ flag is set to 1 (the FS bit in SAR is 0) 3. When a stop condition is detected (when the STOP or ESTP flag is set to 1) while the STOPIM bit is 0

Bit	Bit Name	Initial Value	R/W	Description
1	IRIC	0	R/(W)*	Note: When the slave address does not match and the general call address is not detected (with all flags of AAS, AASX, and ADZ cleared to 0), transmission and reception do not proceed. Thus, the ICDRE and ICDRF flags will not be set. Nor will the IRIC flag. However, even in this case, if STOPIM is 0, the IRIC flag is set by condition 3 above. If detection of a stop condition is not necessary, set STOPIM to 1 to disable setting of the IRIC flag. [Clearing condition] • When 0 is written in IRIC after reading IRIC = 1

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

When, with the I²C bus format selected, IRIC is set to 1 and an interrupt is generated, other flags must be checked in order to identify the source that set IRIC to 1. Although each source has a corresponding flag, caution is needed at the end of a transfer.

When the ICDRE or ICDRF flag is set, the IRTR flag may or may not be set. The IRTR flag is not set at the end of a data transfer up to detection of a retransmission start condition or stop condition after a slave address (SVA) or general call address match in I²C bus format slave mode.

Tables 18.5 and 18.6 show the relationship between the flags and the transfer states.

Table 18.5 Flags and Transfer States (Master Mode)

MST	TRS	BBSY	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB	ICDRF	ICDRE	State
1	1	0	0	0	0	0↓	0	0↓	0↓	0	—	0	Idle state (flag clearing required)
1	1	1↑	0	0	1↑	0	0	0	0	0	—	1↑	Start condition detected
1	—	1	0	0	—	0	0	0	0	—	—	—	Wait state
1	1	1	0	0	—	0	0	0	0	1↑	—	—	Transmission end (ACKE=1 and ACKB=1)
1	1	1	0	0	1↑	0	0	0	0	0	—	1↑	Transmission end with ICDRE=0
1	1	1	0	0	—	0	0	0	0	0	—	0↓	ICDR write with the above state
1	1	1	0	0	—	0	0	0	0	0	—	1	Transmission end with ICDRE=1
1	1	1	0	0	—	0	0	0	0	0	—	0↓	ICDR write with the above state or after start condition detected
1	1	1	0	0	1↑	0	0	0	0	0	—	1↑	Automatic data transfer from ICDRT to ICDRS with the above state
1	0	1	0	0	1↑	0	0	0	0	—	1↑	—	Reception end with ICDRF=0
1	0	1	0	0	—	0	0	0	0	—	0↓	—	ICDR read with the above state
1	0	1	0	0	—	0	0	0	0	—	1	—	Reception end with ICDRF=1
1	0	1	0	0	—	0	0	0	0	—	0↓	—	ICDR read with the above state
1	0	1	0	0	1↑	0	0	0	0	—	1↑	—	Automatic data transfer from ICDRS to ICDRR with the above state

MST	TRS	BBSY	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB	ICDRF	ICDRE	State
0↓	0↓	1	0	0	—	0	1↑	0	0	—	—	—	Arbitration lost
1	—	0↓	0	0	—	0	0	0	0	—	—	0↓	Stop condition detected

[Legend]

0: 0-state retained

1: 1-state retained

—: Previous state retained

0↓: Cleared to 0

1↑: Set to 1

Table 18.6 Flags and Transfer States (Slave Mode)

MST	TRS	BBSY	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB	ICDRF	ICDRE	State
0	0	0	0	0	0	0	0	0	0	0	—	0	Idle state (flag clearing required)
0	0	1↑	0	0	0	0↓	0	0	0	0	—	1↑	Start condition detected
0	1↑/0 *1	1	0	0	0	0	—	1↑	0	0	1↑	1	SAR match in first frame (SARX≠SAR)
0	0	1	0	0	0	0	—	1↑	1↑	0	1↑	1	General call address match in first frame (SARX≠H'00)
0	1↑/0 *1	1	0	0	1↑	1↑	—	0	0	0	1↑	1	SAR match in first frame (SAR≠SARX)
0	1	1	0	0	—	—	—	—	0	1↑	—	—	Transmission end (ACKE=1 and ACKB=1)
0	1	1	0	0	1↑/0 *2	—	—	—	0	0	—	1↑	Transmission end with ICDRE=0
0	1	1	0	0	—	—	0↓	0↓	0	0	—	0↓	ICDR write with the above state
0	1	1	0	0	—	—	—	—	1	0		1	Transmission end with ICDRE=1
0	1	1	0	0	—	—	0↓	0↓	0	0		0↓	ICDR write with the above state
0	1	1	0	0	1↑/0 *2	—	0	0	0	0		1↑	Automatic data transfer from ICDRT to ICDRS with the above state
0	0	1	0	0	1↑/0 *2	—	—	—	—	—	1↑	—	Reception end with ICDRF=0
0	0	1	0	0	—	—	0↓	0↓	0↓	—	0↓	—	ICDR read with the above state
0	0	1	0	0	—	—	—	—	—	—	1	—	Reception end with ICDRF=1
0	0	1	0	0	—	—	0↓	0↓	0↓	—	0↓	—	ICDR read with the above state
0	0	1	0	0	1↑/0 *2	—	0	0	0	—	1↑	—	Automatic data transfer from ICDRS to ICDRR with the above state
0	—	0↓	1↑/0 *3	0/1↑ *3	—	—	—	—	—	—	—	0↓	Stop condition detected

[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.

18.3.6 I²C Bus Status Register (ICSR)

ICSR consists of status flags. Also see tables 18.5 and 18.6.

Bit	Bit Name	Initial Value	R/W	Description
7	ESTP	0	R/(W)*	Error Stop Condition Detection Flag This bit is valid in I²C bus format slave mode. [Setting condition] When a stop condition is detected during frame transfer. [Clearing conditions] • When 0 is written in ESTP after reading ESTP = 1 • When the IRIC flag in ICCR is cleared to 0
6	STOP	0	R/(W)*	Normal Stop Condition Detection Flag This bit is valid in I²C bus format slave mode. [Setting condition] When a stop condition is detected after frame transfer completion. [Clearing conditions] • When 0 is written in STOP after reading STOP = 1 • When the IRIC flag is cleared to 0
5	IRTR	0	R/(W)*	I²C Bus Interface Continuous Transfer Interrupt Request Flag Indicates that the I²C bus interface has issued an interrupt request to the CPU, and the source is completion of reception/transmission of one frame. When the IRTR flag is set to 1, the IRIC flag is also set to 1 at the same time. [Setting conditions] I²C bus format slave mode: • When the ICDRE or ICDRF flag in ICDR is set to 1 when AASX = 1 Master mode or clocked synchronous serial format mode with I²C bus format: • 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 is 1

Bit	Bit Name	Initial Value	R/W	Description
4	AASX	0	R/(W)*	Second Slave Address Recognition Flag In I²C bus format slave receive mode, this flag is set to 1 if the first frame following a start condition matches bits SVAX6 to SVAX0 in SARX. [Setting condition] When the second slave address is detected in slave receive mode and FSX = 0 in SARX [Clearing conditions] • When 0 is written in AASX after reading AASX = 1 • When a start condition is detected • In master mode
3	AL	0	R/(W)*	Arbitration Lost Flag Indicates that arbitration was lost in master mode. [Setting conditions] When ALSL=0 • If the internal SDA and SDA pin disagree at the rise of SCL in master transmit mode • If the internal SCL line is high at the fall of SCL in master mode When ALSL=1 • If the internal SDA and SDA pin disagree at the rise of SCL in master transmit mode • If the SDA pin is driven low by another device before the I²C bus interface drives the SDA pin low, after the start condition instruction was executed in master transmit mode [Clearing conditions] • When ICDR is written to (transmit mode) or read from (receive mode) • When 0 is written in AL after reading AL = 1

Bit	Bit Name	Initial Value	R/W	Description
2	AAS	0	R/(W)*	Slave Address Recognition Flag In I²C bus format slave receive mode, this flag is set to 1 if the first frame following a start condition matches bits SVA6 to SVA0 in SAR, or if the general call address (H'00) is detected. [Setting condition] When the slave address or general call address (one frame including a R/$\overline{W}$ bit is H'00) is detected in slave receive mode and FS = 0 in SAR [Clearing conditions] • When ICDR is written to (transmit mode) or read from (receive mode) • When 0 is written in AAS after reading AAS = 1 • In master mode
1	ADZ	0	R/(W)*	General Call Address Recognition Flag In I²C bus format slave receive mode, this flag is set to 1 if the first frame following a start condition is the general call address (H'00). [Setting condition] When the general call address (one frame including a R/$\overline{W}$ bit is H'00) is detected in slave receive mode and FS = 0 or FSX = 0 [Clearing conditions] • When ICDR is written to (transmit mode) or read from (receive mode) • When 0 is written in ADZ after reading ADZ = 1 • In master mode If a general call address is detected while FS=1 and FSX=0, the ADZ flag is set to 1; however, the general call address is not recognized (AAS flag is not set to 1).

Bit	Bit Name	Initial Value	R/W	Description
0	ACKB	0	R/W	Acknowledge Bit Stores acknowledge data. The bit function varies depending on transmit mode and receive mode. Transmit mode: Holds the acknowledge data returned by the receiving device. [Setting condition] When 1 is received as the acknowledge bit when ACKE = 1 in transmit mode [Clearing conditions] • When 0 is received as the acknowledge bit when ACKE = 1 in transmit mode • When 0 is written to the ACKE bit Receive mode: Sets the acknowledge data to be returned to the transmitting device. 0: Returns 0 as acknowledge data after data reception 1: Returns 1 as acknowledge data after data reception When this bit is read, the value loaded from the bus line (returned by the receiving device) is read in transmission (when TRS = 1). In reception (when TRS = 0), the value set by internal software is read. When this bit is written, acknowledge data that is returned after receiving is rewritten regardless of the TRS value. Note: When, in transmit mode, this bit has been overwritten by a bit manipulation instruction with a value other than that of the ACKB flag in ICSR, the value of the ACKB bit as the acknowledge data setting for receive mode is overwritten by this value. Thus, always reset the acknowledge data when switching to receive mode. Write 0 to the ACKE bit to clear the ACKB flag to 0 in the following cases: in master mode—before transmission is ended and a stop condition is generated; and in slave mode—before transmission is ended and SDA is released to allow a master device to issue a stop condition.

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

18.3.7 I²C Bus Control Initialization Register (ICRES)

ICRES controls IIC internal latch clearance.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R/W	Reserved The initial value should not be changed.
4	—	0	R	Reserved
3	CLR3	1	W*	IIC Clear 3 to 0
2	CLR2	1	W*	Controls initialization of the internal state of IIC_0 and IIC_1. (ICRES_0)
1	CLR1	1	W*	
0	CLR0	1	W*	00--: Setting prohibited 0100: Setting prohibited 0101: IIC_0 internal latch cleared 0110: IIC_1 internal latch cleared 0111: IIC_0 and IIC_1 internal latches cleared 1---: Invalid setting Controls initialization of the internal state of IIC_2. (ICRES_2) 00--: Setting prohibited 0100: Setting prohibited 0101: IIC_2 internal latch cleared 0110: Setting prohibited 0111: IIC_2 internal latch cleared 1---: Invalid setting When a write operation is performed on these bits, a clear signal is generated for the internal latch circuit of the corresponding module, and the internal state of the IIC module is initialized. These bits can only be written to; they are always read as 1. Write data to this bit is not retained. To perform IIC clearance, bits CLR3 to CLR0 must be written to simultaneously using an MOV instruction. Do not use a bit manipulation instruction such as BCLR. When clearing is required again, all the bits must be written to in accordance with the setting.

Note: * This bit is always read as 1.

18.3.8 I²C Bus Extended Control Register (ICXR)

ICXR enables or disables the I²C bus interface interrupt generation and handshake control, and indicates the status of receive/transmit operations.

Bit	Bit Name	Initial Value	R/W	Description
7	STOPIM	0	R/W	Stop Condition Interrupt Source Mask Enables or disables the interrupt generation when the stop condition is detected in slave mode. 0: Enables IRIC flag setting and interrupt generation when the stop condition is detected (STOP = 1 or ESTP = 1) in slave mode. 1: Disables IRIC flag setting and interrupt generation when the stop condition is detected.
6	HNDS	0	R/W	Enables or disables handshake control in receive mode for the selection of reception with handshaking. 0: Disables handshake control 1: Enables handshake control Note: When the IIC module is in use, be sure to set this bit to 1. When the HNDS bit is cleared to 0 and a round of reception is completed with ICDRR empty (the ICDRF flag is 0), successive reception will proceed with the next round of reception. At the same time, a clock is continuously supplied over the SCL line. In this case, the sequence of operations should be such that unnecessary clock cycles are not output to the bus after reception of the last of the data. When the HNDS bit is set to 1, SCL is fixed low and clock output stops on completion of reception. SCL is released and reception of the next frame is enabled by reading the receive data from ICDR.

Bit	Bit Name	Initial Value	R/W	Description
5	ICDRF	0	R	Receive Data Read Request Flag Indicates the ICDR (ICDRR) status in receive mode. 0: Indicates that the data has been already read from ICDR (ICDRR) or ICDR is initialized. 1: Indicates that data has been received successfully and transferred from ICDRS to ICDRR, and the data is ready to be read out. [Setting conditions] • When data is received successfully and transferred from ICDRS to ICDRR. (1) When data is received successfully while ICDRF = 0 (at the rise of the 9th clock pulse). (2) When ICDR is read successfully in receive mode after data was received while ICDRF = 1. [Clearing conditions] • When ICDR (ICDRR) is read. • When 0 is written to the ICE bit. • When the IIC is internally initialized using the CLR3 to CLR0 bits in DDCSWR. When ICDRF is set due to the condition (2) above, ICDRF is temporarily cleared to 0 when ICDR (ICDRR) is read; however, since data is transferred from ICDRS to ICDRR immediately, ICDRF is set to 1 again. Note that ICDR cannot be read successfully in transmit mode (TRS = 1) because data is not transferred from ICDRS to ICDRR. Be sure to read data from ICDR in receive mode (TRS = 0).

Bit	Bit Name	Initial Value	R/W	Description
4	ICDRE	0	R	Transmit Data Write Request Flag Indicates the ICDR (ICDRT) status in transmit mode. 0: Indicates that the data has been already written to ICDR (ICDRT) or ICDR is initialized. 1: Indicates that data has been transferred from ICDRT to ICDRS and is being transmitted, or the start condition has been detected or transmission has been complete, thus allowing the next data to be written to. [Setting conditions] • When the start condition is detected from the bus line state with I²C bus format or serial format. • When data is transferred from ICDRT to ICDRS. 1. When data transmission completed while ICDRE = 0 (at the rise of the 9th clock pulse). 2. When data is written to ICDR in transmit mode after data transmission was completed while ICDRE = 1. [Clearing conditions] • When data is written to ICDR (ICDRT). • When the stop condition is detected with I²C bus format or serial format. • When 0 is written to the ICE bit. • When the IIC is internally initialized using the CLR3 to CLR0 bits in DDCSWR. Note that if the ACKE bit is set to 1 with I²C bus format thus enabling acknowledge bit decision, ICDRE is not set when data transmission is completed while the acknowledge bit is 1. When ICDRE is set due to the condition (2) above, ICDRE is temporarily cleared to 0 when data is written to ICDR (ICDRT); however, since data is transferred from ICDRT to ICDRS immediately, ICDRE is set to 1 again. Do not write data to ICDR when TRS = 0 because the ICDRE flag value is invalid during the time.

Bit	Bit Name	Initial Value	R/W	Description
3	ALIE	0	R/W	Arbitration Lost Interrupt Enable Enables or disables IRIC flag setting and interrupt generation when arbitration is lost. 0: Disables interrupt request when arbitration is lost. 1: Enables interrupt request when arbitration is lost.
2	ALSL	0	R/W	Arbitration Lost Condition Select Selects the condition under which arbitration is lost. 0: When the SDA pin state disagrees with the data that IIC bus interface outputs at the rise of SCL, or when the SCL pin is driven low by another device. 1: When the SDA pin state disagrees with the data that IIC bus interface outputs at the rise of SCL, or when the SDA line is driven low by another device in idle state or after the start condition instruction was executed.
1	FNC1	0	R/W	Function 1, 0
0	FNC0	0	R/W	These bits cancel some restrictions on usage. FNC0 FNC1 0 0: Restrictions on operation canceled 0 1: Setting prohibited 1 0: Setting prohibited 1 1: Restrictions on operation remaining in effect Note: When the IIC module is used, make sure to set both of the bits to 1.

18.4 Operation

The I²C bus interface has an I²C bus format and a serial format.

18.4.1 I²C Bus Data Format

The I²C bus format is an addressing format with an acknowledge bit. This is shown in figure 18.3. The first frame following a start condition always consists of 9 bits.

The serial format is a non-addressing format with no acknowledge bit. This is shown in figure 18.4.

Figure 18.5 shows the I²C bus timing.

The symbols used in figures 18.3 to 18.5 are explained in table 18.7.

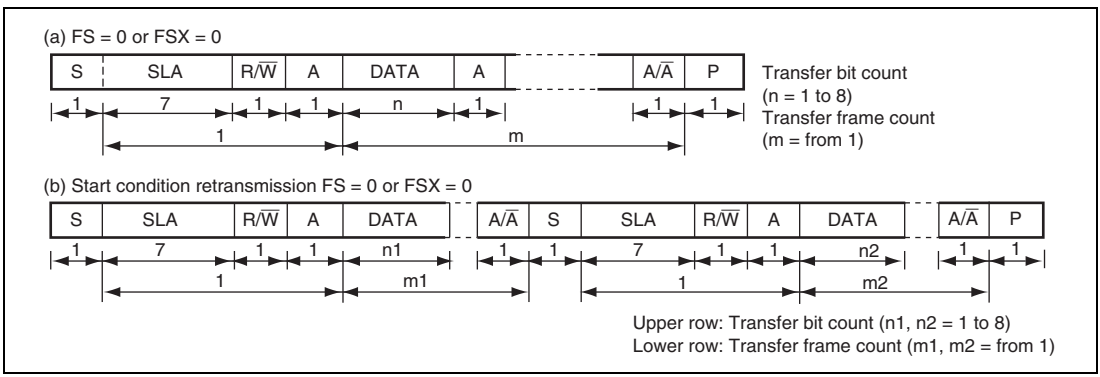


Figure 18.3 I²C Bus Data Format (I²C Bus Format)

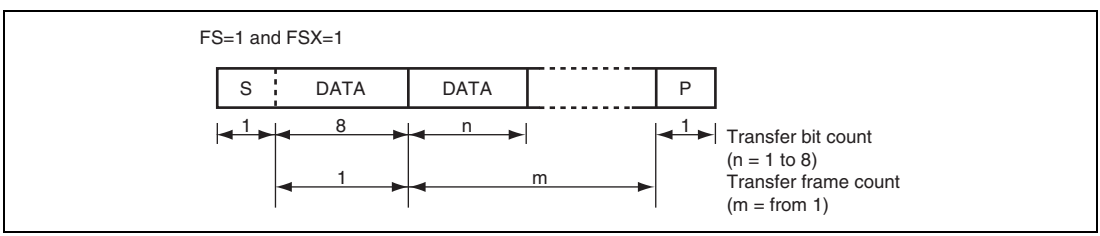


Figure 18.4 I²C Bus Data Format (Serial Format)

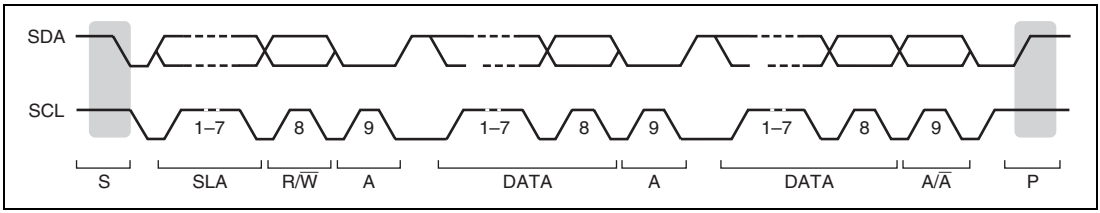


Figure 18.5 I²C Bus Timing

Table 18.7 I²C Bus Data Format Symbols

Legend	
S	Start condition. The master device drives SDA from high to low while SCL is high
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 device when R/$\overline{W}$ is 1, or from the master device to the slave device when R/$\overline{W}$ is 0
A	Acknowledge. The receiving device drives SDA low to acknowledge a transfer. (The slave device returns acknowledge in master transmit mode, and the master device returns acknowledge in master receive mode.)
DATA	Transferred data. The bit length of transferred data is set with the BC2 to BC0 bits in 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 high

18.4.2 Initialization

Initialize the IIC by the procedure shown in figure 18.6 before starting transmission/reception of data.

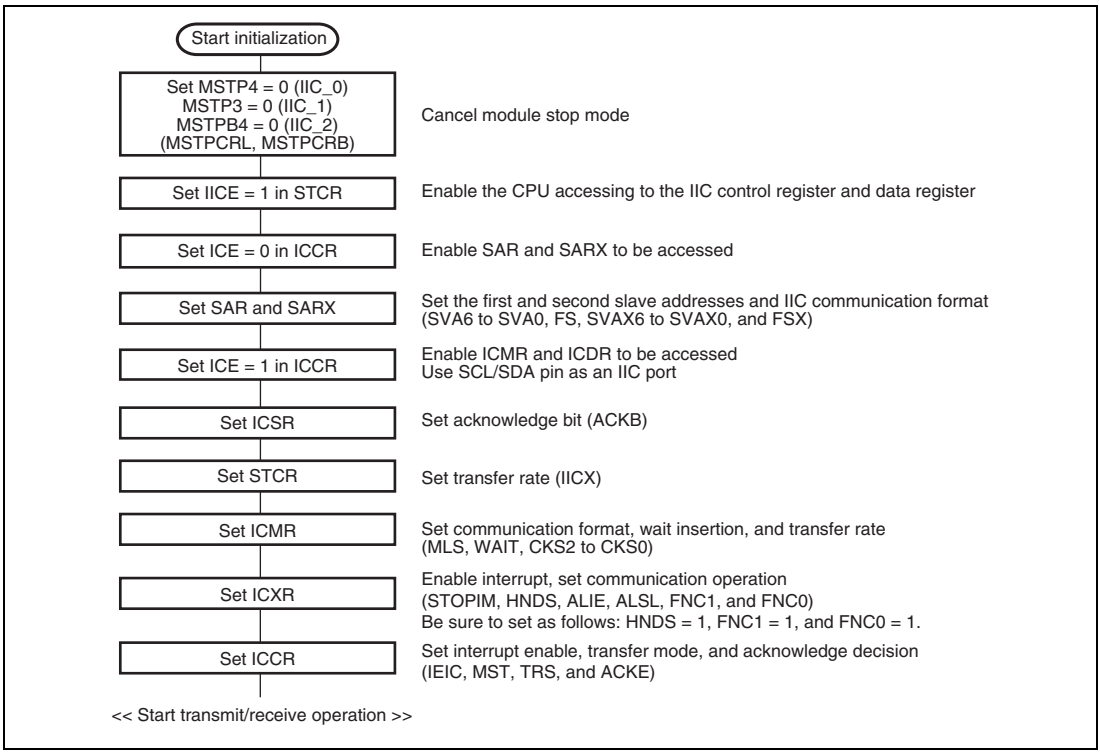


Figure 18.6 Sample Flowchart for IIC Initialization

Note: Be sure to modify the ICMR register after transmit/receive operation has been completed. If the ICMR register is modified during transmit/receive operation, bit counter BC2 to BC0 will be modified erroneously, thus causing incorrect operation.

18.4.3 Master Transmit Operation

In I²C bus format master transmit mode, the master device outputs the transmit clock and transmit data, and the slave device returns an acknowledge signal.

Figure 18.7 shows the sample flowchart for the operations in master transmit mode.

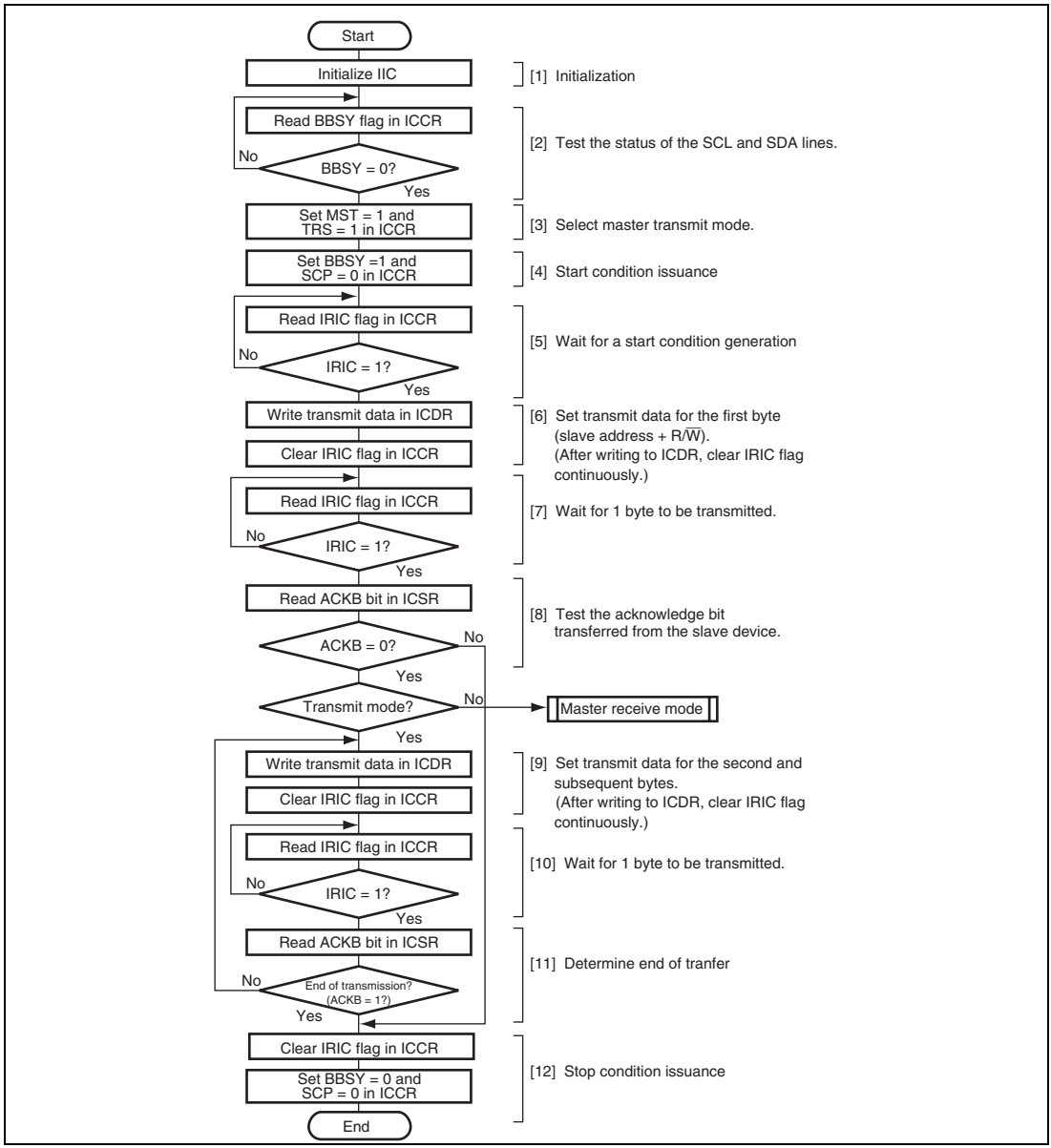


Figure 18.7 Sample Flowchart for Operations in Master Transmit Mode

The master mode transmission procedure and operations are described below.

1. Initialize the IIC as described in section 18.4.2, Initialization.
2. Read the BBSY flag in ICCR to confirm that the bus is free.
3. Set bits MST and TRS to 1 in ICCR to select master transmit mode.
4. Write 1 to BBSY and 0 to SCP in ICCR. This changes SDA from high to low when SCL is high, and generates the start condition.
5. Then the IRIC and IRTR flags are set to 1. If the IEIC bit in ICCR has been set to 1, an interrupt request is sent to the CPU.
6. Write the data (slave address + R/$\overline{W}$) to ICDR.

   With the I²C bus format (when the FS bit in SAR or the FSX bit in SARX is 0), the first frame data following the start condition indicates the 7-bit slave address and transmit/receive direction (R/$\overline{W}$).

   To determine the end of the transfer, the IRIC flag is cleared to 0. After writing to ICDR, clear IRIC continuously so no other interrupt handling routine is executed. If the time for transmission of one frame of data has passed before the IRIC clearing, the end of transmission cannot be determined. The master device sequentially sends the transmission clock and the data written to ICDR. The selected slave device (i.e. the slave device with the matching slave address) drives SDA low at the 9th transmit clock pulse and returns an acknowledge signal.
7. When one frame of data has been transmitted, the IRIC flag is set to 1 at the rise of the 9th transmit clock pulse. After one frame has been transmitted, SCL is automatically fixed low in 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 device has not acknowledged (ACKB bit is 1), operate step [12] to end transmission, and retry the transmit operation.
9. Write the transmit data to ICDR.

   As indicating the end of the transfer, the IRIC flag is cleared to 0. Perform the ICDR write and the IRIC flag clearing sequentially, just as in step [6]. Transmission of the next frame is 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 the 9th transmit clock pulse. After one frame has been transmitted, SCL is automatically fixed low in synchronization with the internal clock until the next transmit data is written.
11. Read the ACKB bit in ICSR.

    Confirm that the slave device has been acknowledged (ACKB bit is 0). When there is still data to be transmitted, go to step [9] to continue the next transmission operation. When the slave device has not acknowledged (ACKB bit is set to 1), operate step [12] to end transmission.

12. Clear the IRIC flag to 0.

    Write 0 to ACKE in ICCR, to clear received ACKB contents to 0.

    Write 0 to BBSY and SCP in ICCR. This changes SDA from low to high when SCL is high, and generates the stop condition.

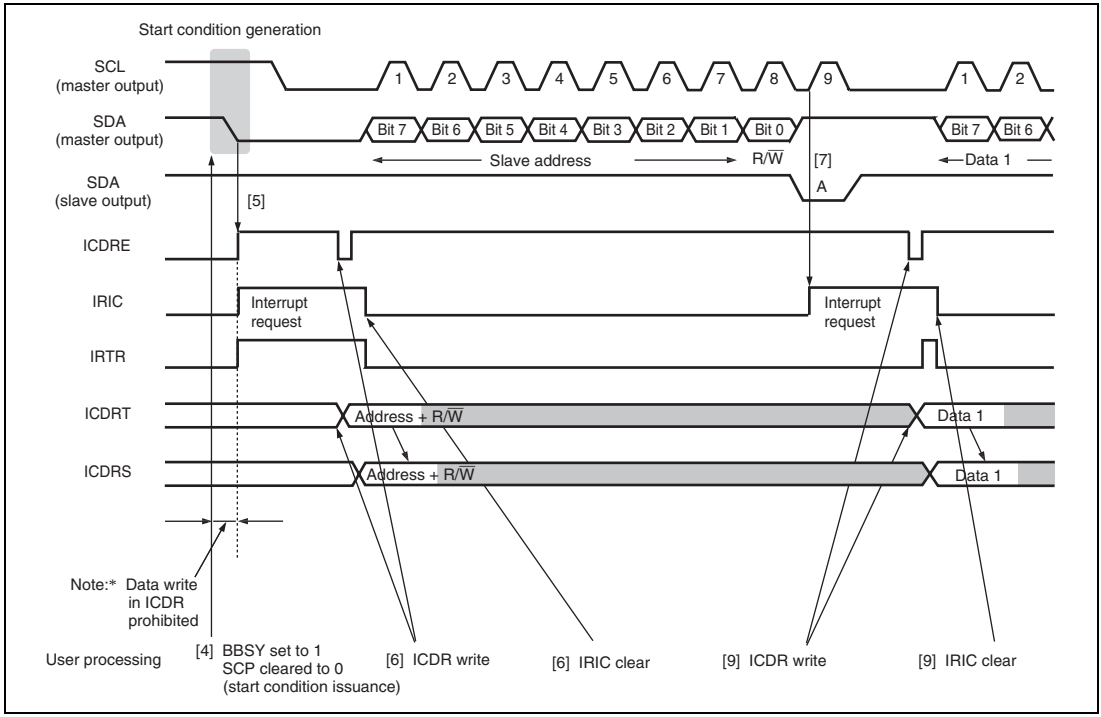


Figure 18.8 Example of Operation Timing in Master Transmit Mode (MLS = WAIT = 0)

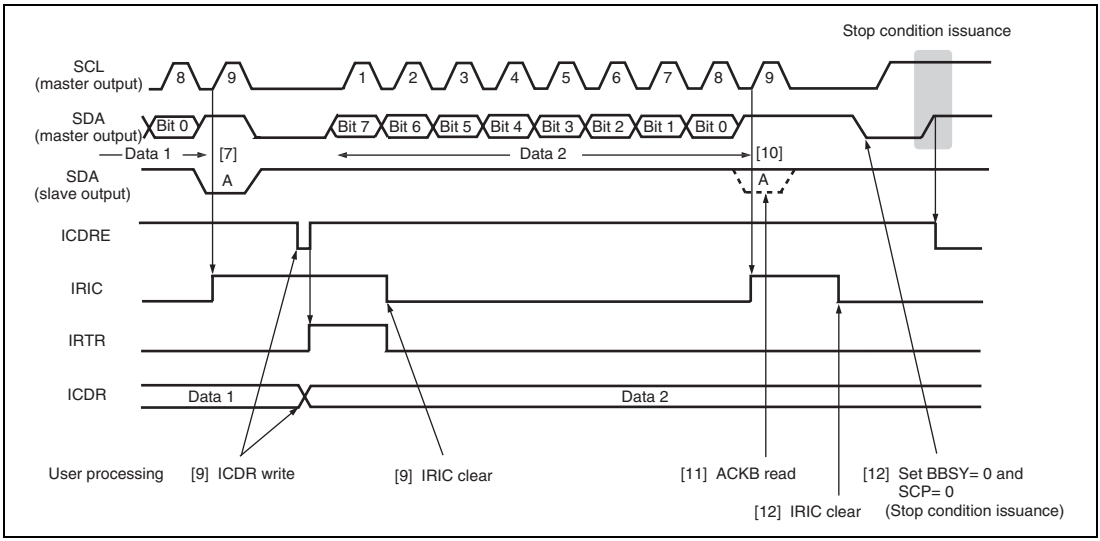


Figure 18.9 Example of Stop Condition Issuance Operation Timing in Master Transmit Mode (MLS = WAIT = 0)

18.4.4 Master Receive Operation

In I²C bus format master receive mode, the master device outputs the receive clock, receives data, 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 first frame following the start condition issuance in master transmit mode, selects the slave device, and then switches the mode for receive operation.

Figure 18.10 shows the sample flowchart for the operations in master receive mode.

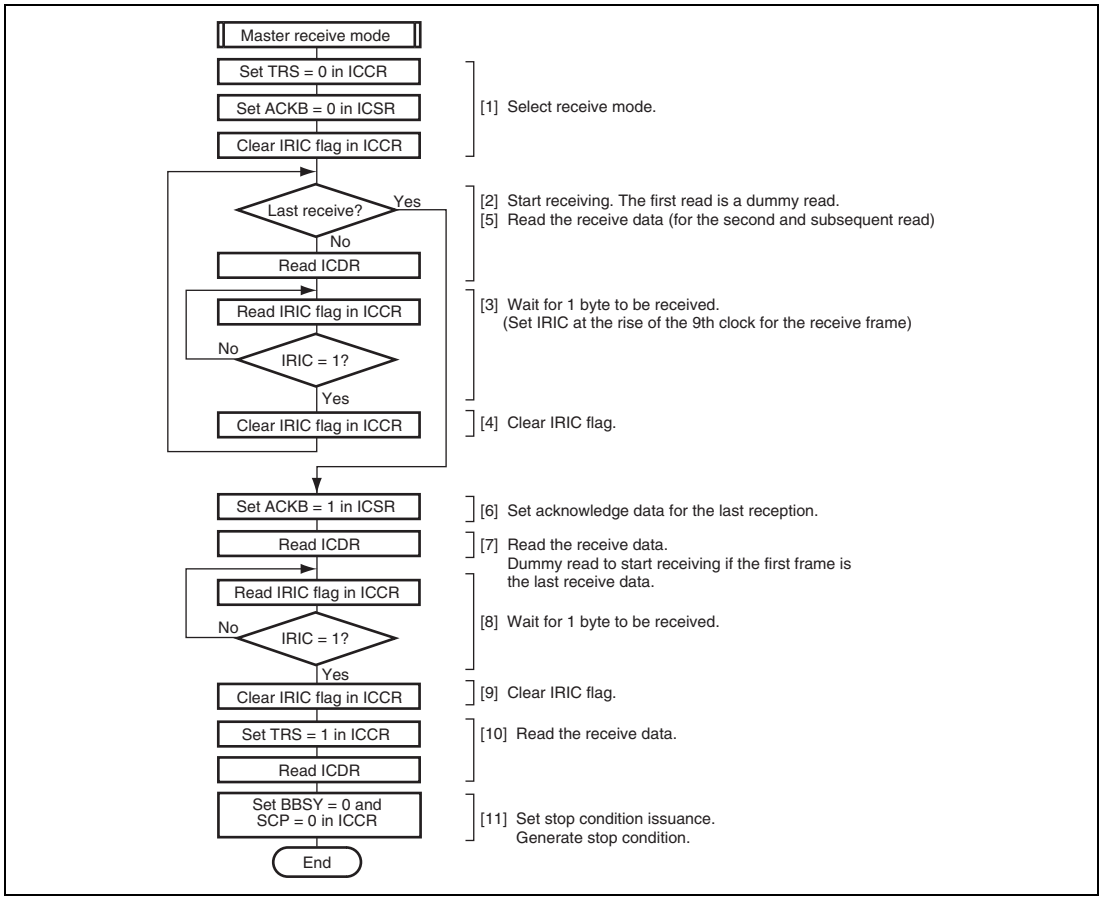


Figure 18.10 Sample Flowchart for Operations in Master Receive Mode

The master mode reception procedure and operations are described below.

1. Clear the TRS bit in ICCR to 0 to switch from transmit mode to receive mode.
   Clear the ACKB bit in ICSR to 0 (acknowledge data setting).
   Clear the IRIC flag to 0 to determine the end of reception.
   Go to step [6] to halt reception operation if the first frame is the last receive data.
2. When ICDR is read (dummy data read), reception is started, the receive clock is output in synchronization with the internal clock, and data is received. (Data from the SDA pin is sequentially transferred to ICDRS in synchronization with the rise of the receive clock pulses.)
3. The master device drives SDA low to return the acknowledge data at the 9th receive clock pulse. The receive data is transferred from ICDRS to ICDRR at the rise of the 9th clock pulse, setting the ICDRF, IRIC, and IRTR flags to 1. If the IEIC bit has been set to 1, an interrupt request is sent to the CPU.
   The master device drives SCL low from the fall of the 9th receive clock pulse to the ICDR 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 outputs the receive clock continuously to receive the next data.

Data can be received continuously by repeating steps [3] to [5].

6. Set the ACKB bit to 1 so as to return the acknowledge data for the last reception.
7. Read ICDR receive data. This clears the ICDRF flag to 0. The master device outputs the receive clock to receive data.
8. When one frame of data has been received, the ICDRF, IRIC, and IRTR flags are set to 1 at the rise of the 9th receive clock pulse.
9. Clear the IRIC flag to 0.
10. Read ICDR receive data after setting the TRS bit. This clears the ICDRF flag to 0.
11. Clear the BBSY bit and SCP bit to 0 in ICCR. This changes SDA from low to high when SCL is high, and generates the stop condition.

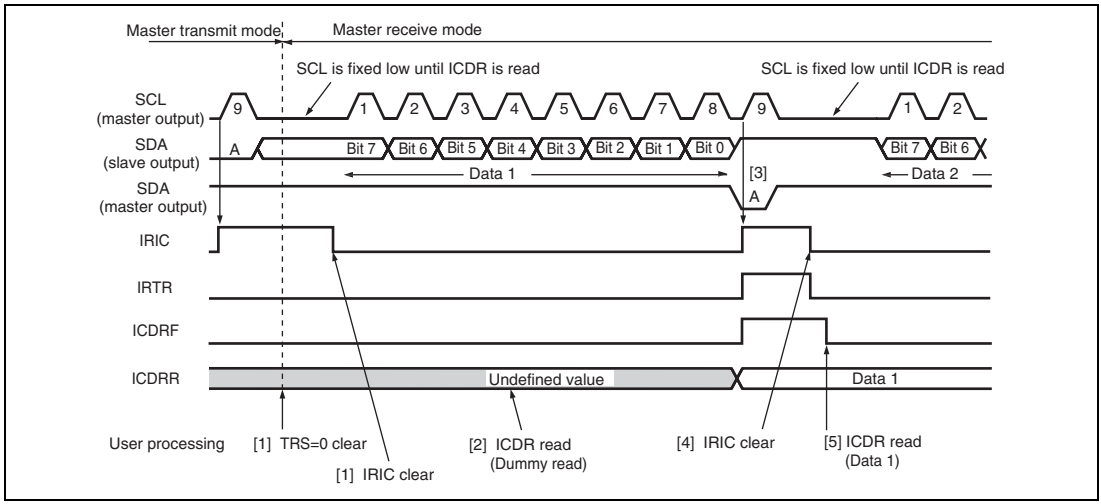


Figure 18.11 Example of Operation Timing in Master Receive Mode (MLS = WAIT = 0)

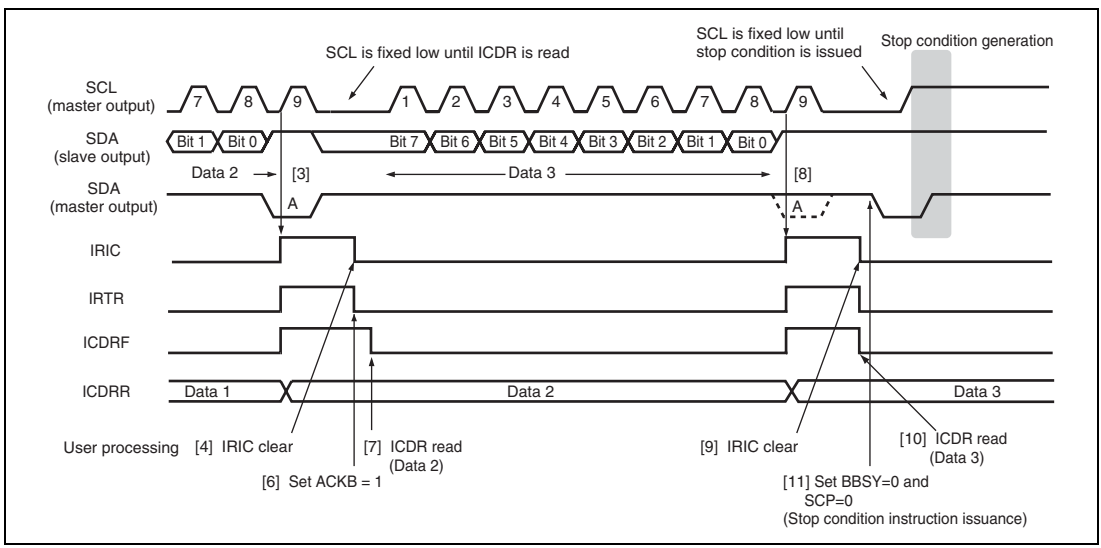


Figure 18.12 Example of Stop Condition Issuance Operation Timing in Master Receive Mode (MLS = WAIT = 0)

18.4.5 Slave Receive Operation

In I²C bus format slave receive mode, the master device outputs the transmit clock and transmit data, and the slave device returns an acknowledge signal.

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

Figure 18.13 shows the sample flowchart for the operations in slave receive mode.

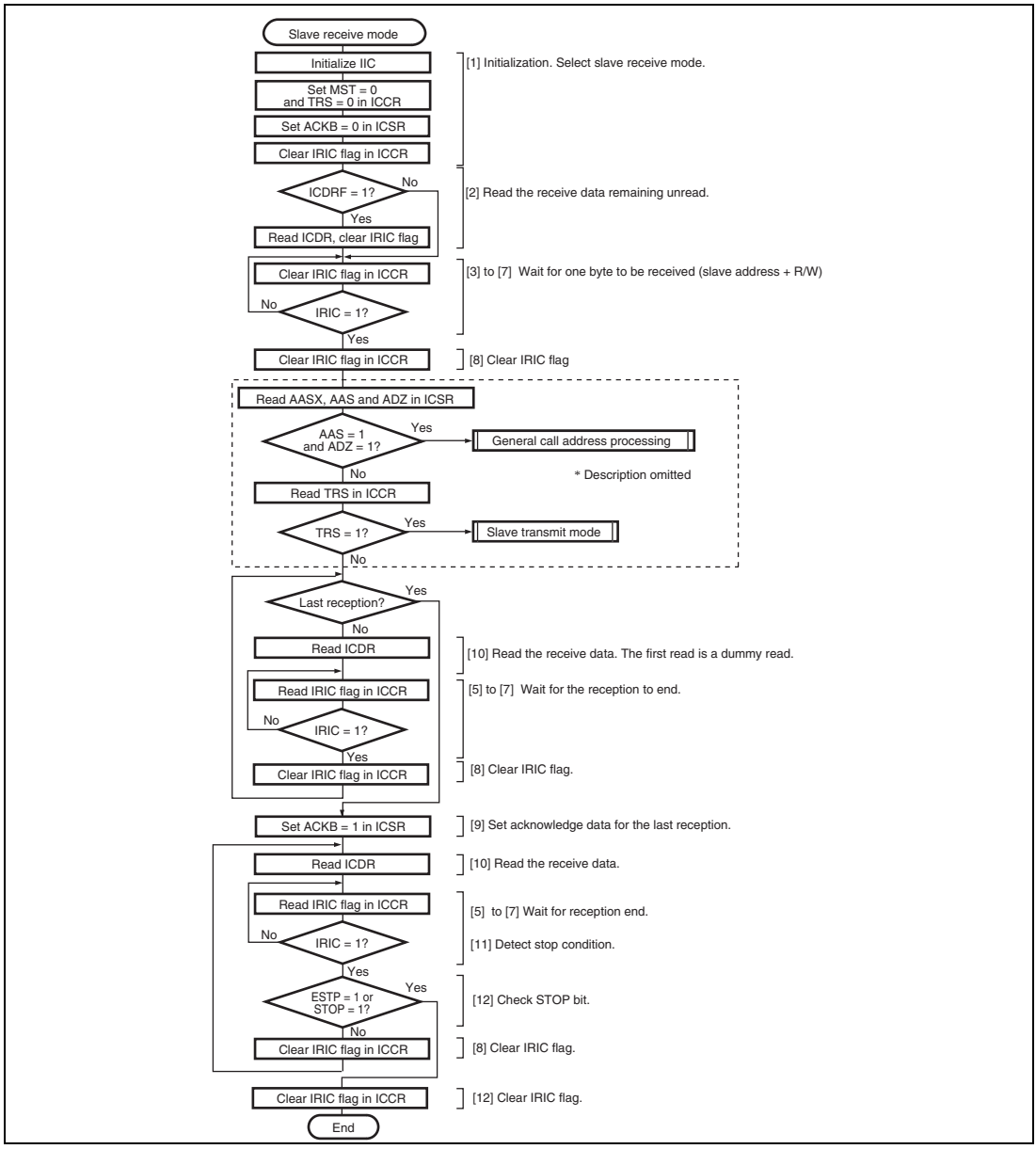


Figure 18.13 Sample Flowchart for Operations in Slave Receive Mode

The slave mode reception procedure and operations are described below.

1. Initialize the IIC as described in section 18.4.2, Initialization.

   Clear the MST and TRS bits to 0 to set slave receive mode, and set the ACKB bit to 0. Clear the IRIC flag in ICCR to 0 to see the end of reception.
2. Confirm that the ICDRF flag is 0. If the ICDRF flag is set to 1, read the ICDR and then clear the IRIC flag to 0.
3. When the start condition output by the master device is detected, the BBSY flag in ICCR is set to 1. The master device then outputs the 7-bit slave address and transmit/receive direction (R/W), in synchronization with the transmit clock pulses.
4. When the slave address matches in the first frame following the start condition, the device operates as the slave device specified by the master device. If the 8th data bit (R/$\overline{W}$) is 0, the TRS bit remains cleared to 0, and slave receive operation is performed. If the 8th data bit (R/$\overline{W}$) is 1, the TRS bit is set to 1, and slave transmit operation is performed. When the slave address does not match, receive operation is halted until the next start condition is detected.
5. At the 9th clock pulse of the receive frame, the slave device returns the data in the ACKB bit as an acknowledge signal.
6. At the rise of the 9th clock pulse, the IRIC flag is set to 1. If the IEIC bit has been set to 1, an 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 ICDRR, setting the ICDRF flag to 1. The slave device drives SCL low from the fall of the 9th receive 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 enables the master device to transfer the next data.

Receive operations can be performed continuously by repeating steps [5] to [10].

11. When the stop condition is detected (SDA is changed from low to high when SCL is high), the BBSY flag is cleared to 0 and the STOP bit is set to 1. If the STOPIM bit has been cleared to 0, the IRIC flag is set to 1.
12. Confirm that the STOP bit is set to 1, and clear the IRIC flag to 0.

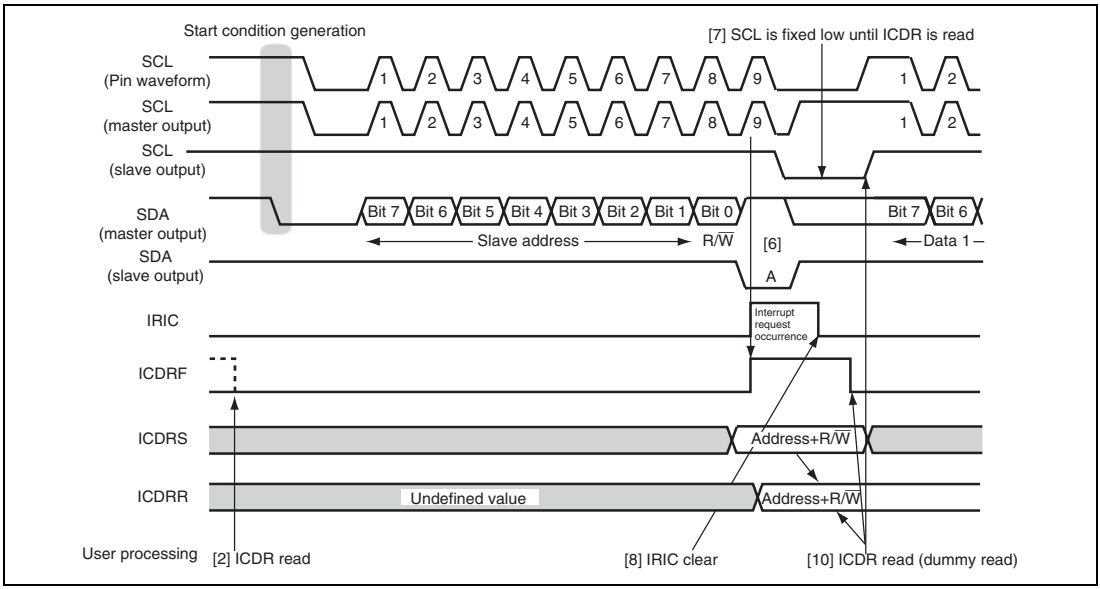


Figure 18.14 Example of Slave Receive Mode Operation Timing (1) (MLS = 0)

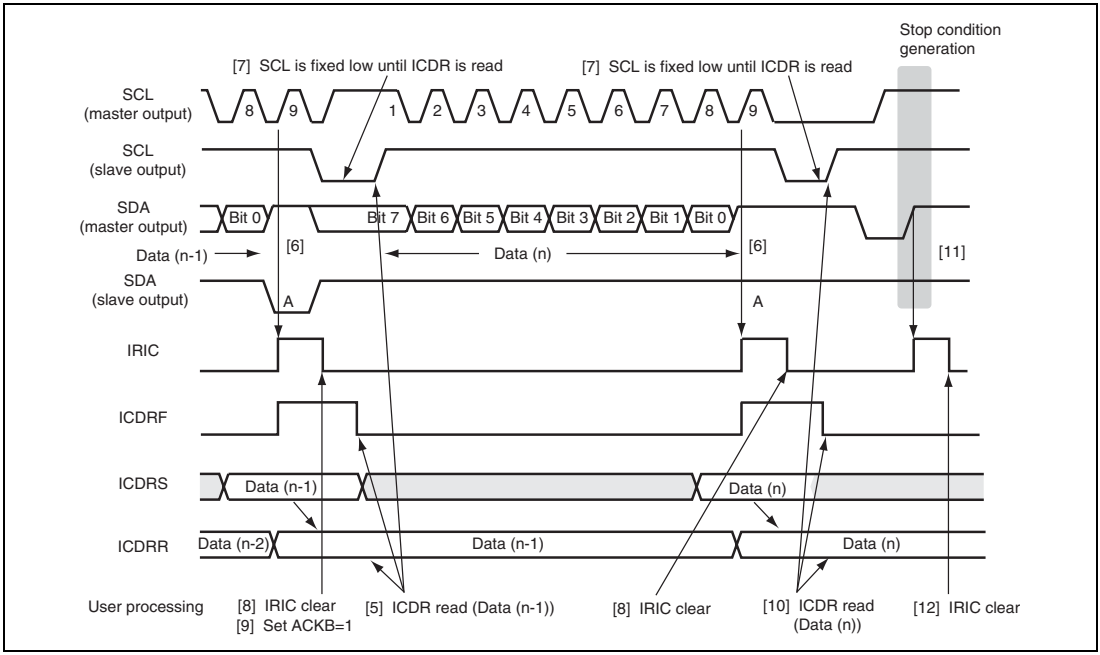


Figure 18.15 Example of Slave Receive Mode Operation Timing (2) (MLS = 0)

18.4.6 Slave Transmit Operation

If the slave address matches to the address in the first frame (address reception frame) following the start condition detection when the 8th bit data (R/$\overline{W}$) is 1 (read), the TRS bit in ICCR is automatically set to 1 and the mode changes to slave transmit mode.

Figure 18.16 shows the sample flowchart for the operations in slave transmit mode.

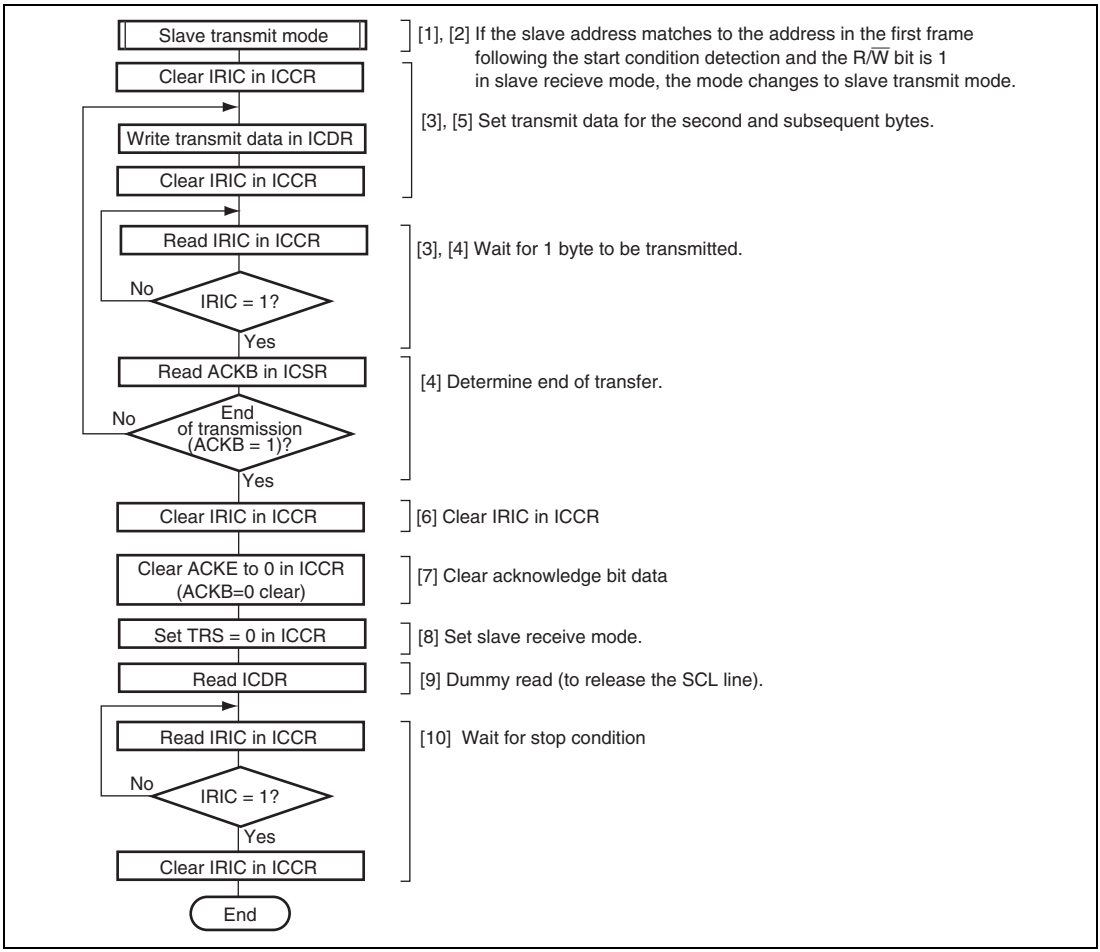


Figure 18.16 Sample Flowchart for Slave Transmit Mode

In slave transmit mode, the slave device outputs the transmit data, while the master device outputs the receive clock and returns an acknowledge signal. The transmission procedure and operations in slave transmit mode are described below.

1. Initialize slave receive mode and wait for slave address reception.
2. When the slave address matches in the first frame following detection of the start condition, the slave device drives SDA low at the 9th clock pulse and returns an acknowledge signal. If the 8th data bit (R/W) is 1, the TRS bit in ICCR is set to 1, and the mode changes to slave transmit mode automatically. The IRIC flag is set to 1 at the rise of the 9th clock. If the IEIC bit in ICCR has been set to 1, an interrupt request is sent to the CPU. At the same time, the ICDRE flag is set to 1. The slave device drives SCL low from the fall of the transmit 9th clock until ICDR data is written, to disable the master device to output the next transfer clock.
3. After clearing the IRIC flag to 0, write data to ICDR. At this time, the ICDRE flag is cleared to 0. The written data is transferred to ICDRS, and the ICDRE and IRIC flags are set to 1 again. The slave device sequentially sends the data written into ICDRS in accordance with the clock output by the master device.

   The IRIC flag is cleared to 0 to detect the end of transmission. Processing from the ICDR register writing to the IRIC flag clearing should be performed continuously. Prevent any other interrupt processing from being inserted.
4. The master device drives SDA low at the 9th clock pulse, and returns an acknowledge signal. As this acknowledge signal is stored in the ACKB bit in ICSR, this bit can be used to determine whether the transfer operation was performed successfully. When one frame of data has been transmitted, the IRIC flag in ICCR is set to 1 at the rise of the 9th transmit clock pulse. When the ICDRE flag is 0, the data written into ICDR is transferred to ICDRS, transmission starts, and the ICDRE and IRIC flags are set to 1 again. If the ICDRE flag has been set to 1, this slave device drives SCL low from the fall of the 9th transmit clock until data is written to ICDR.
5. To continue transmission, write the next data to be transmitted into ICDR. The ICDRE flag is cleared to 0. The IRIC flag is cleared to 0 to detect the end of transmission. Processing from the ICDR writing to the IRIC flag clearing should be performed continuously. 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 ICCR to 0, to clear the acknowledge bit stored in the ACKB bit to 0.
8. Clear the TRS bit to 0 for the next address reception, to set slave receive mode.
9. Dummy-read ICDR to release SCL on the slave side.
10. When the stop condition is detected, that is, when SDA is changed from low to high when SCL is high, the BBSY flag in ICCR is cleared to 0 and the STOP flag in ICSR is set to 1. When the STOPIM bit in ICXR is 0, the IRIC flag is set to 1. If the IRIC flag has been set, it is cleared to 0.

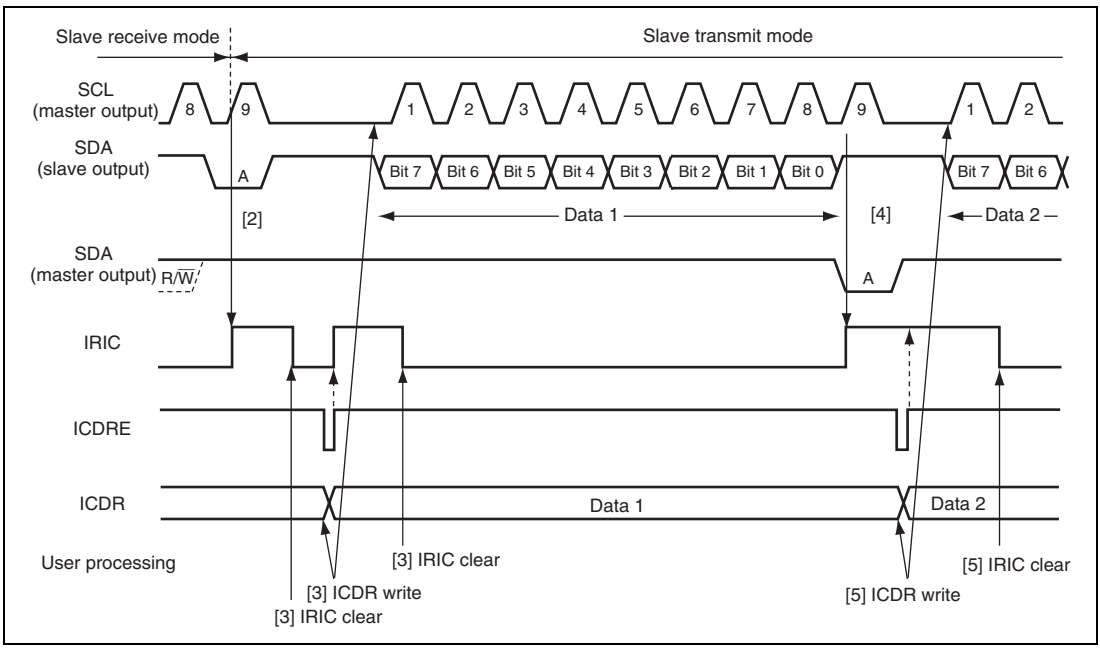


Figure 18.17 Example of Slave Transmit Mode Operation Timing (MLS = 0)

18.4.7 IRIC Setting Timing and SCL Control

The interrupt request flag (IRIC) is set at different times depending on the WAIT bit in ICMR, the FS bit in SAR, and the FSX bit in SARX. If the ICDRE or ICDRF flag is set to 1, SCL is automatically held low after one frame has been transferred in synchronization with the internal clock. Figures 18.18 to 18.20 show the IRIC set timing and SCL control.

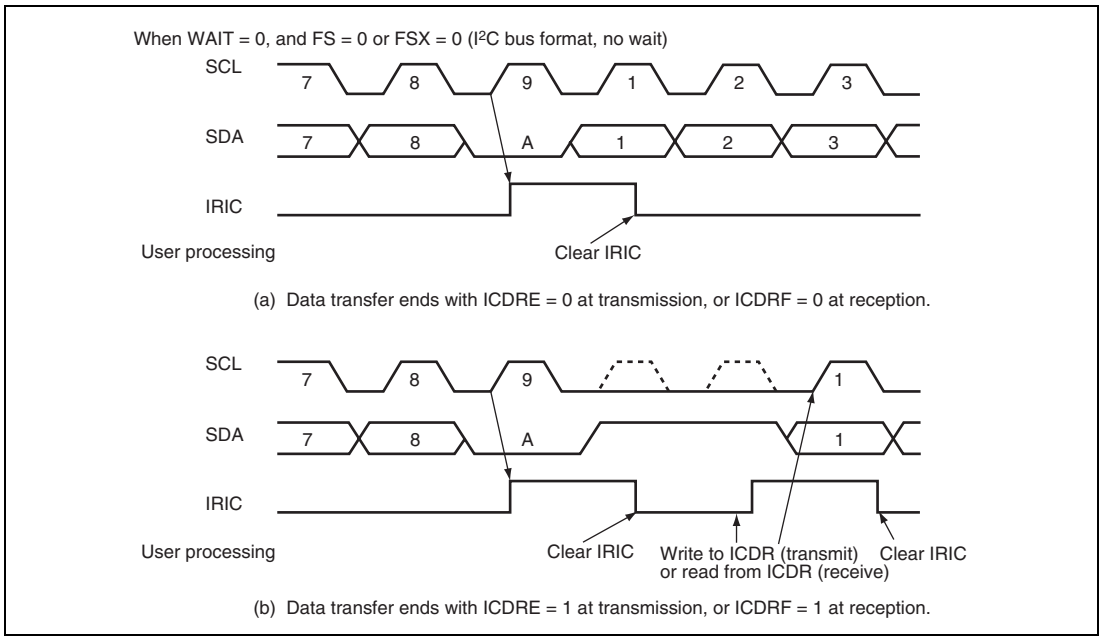


Figure 18.18 IRIC Setting Timing and SCL Control (1)

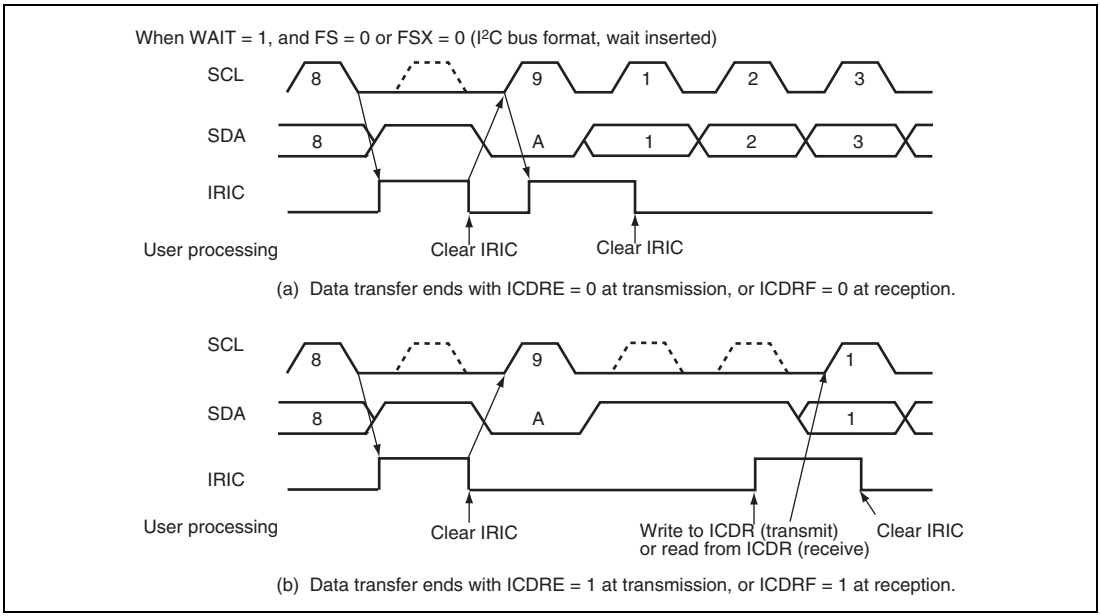


Figure 18.19 IRIC Setting Timing and SCL Control (2)

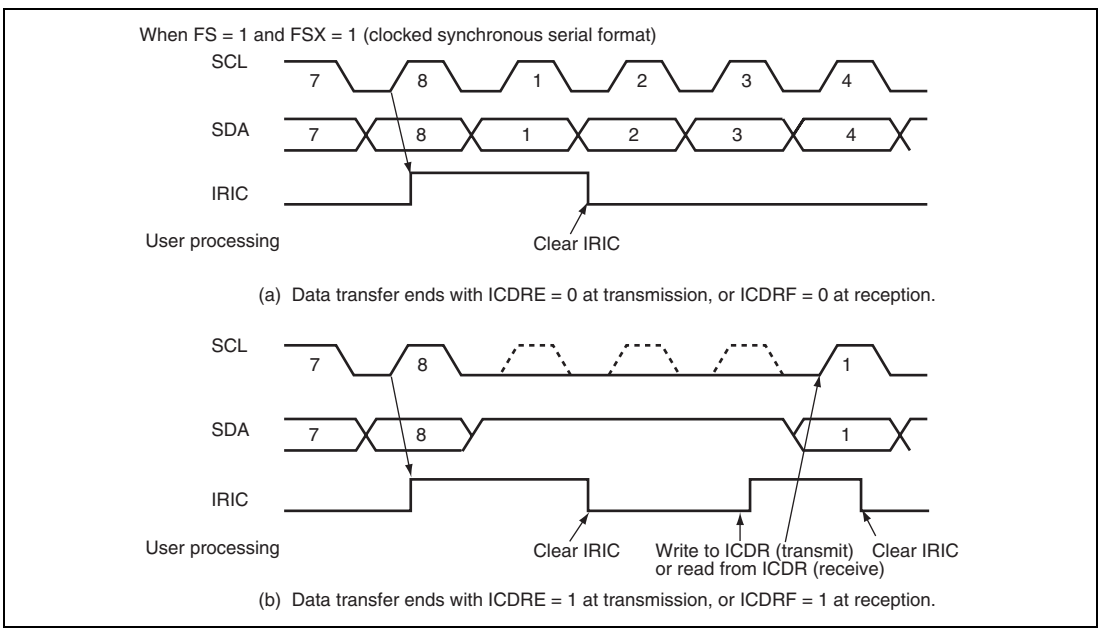


Figure 18.20 IRIC Setting Timing and SCL Control (3)

18.4.8 Noise Canceler

The logic levels at the SCL and SDA pins are routed through noise cancelers before being latched internally. Figure 18.21 shows a block diagram of the noise canceler.

The noise canceler consists of two cascaded latches and a match detector. The SCL (or SDA) pin input signal is sampled on the system clock, but is not passed forward to the next circuit unless the outputs of both latches agree. If they do not agree, the previous value is held.

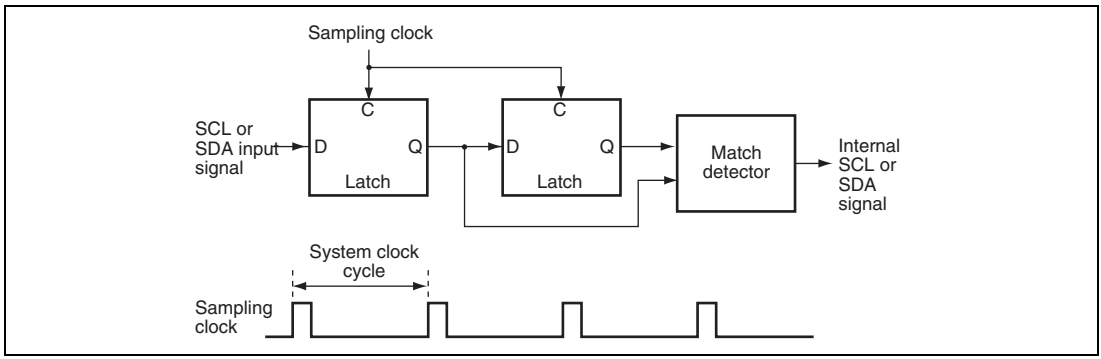


Figure 18.21 Block Diagram of Noise Canceler

18.4.9 Initialization of Internal State

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

Initialization is executed in accordance with the setting of bits CLR3 to CLR0 in ICRES or clearing ICE bit. For details on the setting of bits CLR3 to CLR0, see section 18.3.7, I²C Bus Control Initialization Register (ICRES).

(1) Scope of Initialization

The initialization executed by this function covers the following items:

- 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, data output, etc.)

The following items are not initialized:

- Actual register values (ICDR, SAR, SARX, ICMR, ICCR, ICSR, ICXR (except for the ICDRE and ICDRF flags)
- Internal latches used to retain register read information for setting/clearing flags in ICMR, ICCR, and ICSR
- The value of the ICMR bit counter (BC2 to BC0)
- Generated interrupt sources (interrupt sources transferred to the interrupt controller)

(2) Notes on Initialization

- Interrupt flags and interrupt sources are not cleared, and so flag clearing measures must be taken as necessary.
- Basically, other register flags are not cleared either, and so flag clearing measures must be taken as necessary.
- When initialization is executed by ICRES, the write data for bits CLR3 to CLR0 is not retained. To perform IIC clearance, bits CLR3 to CLR0 must be written to simultaneously using an MOV instruction. Do not use a bit manipulation instruction such as BCLR.
- Similarly, when clearing is required again, all the bits must be written to simultaneously in accordance with the setting.
- If a flag clearing setting is made during transmission/reception, the IIC module will stop 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 as necessary to enable correct communication as a system.

The value of the BBSY bit cannot be modified directly by this module clear function, but since the stop condition pin waveform is generated according to the state and release timing of the SCL and SDA pins, the BBSY bit may be cleared as a result. Similarly, state switching of other bits and flags may also have an effect.

To prevent problems caused by these factors, the following procedure should be used when initializing the IIC state.

1. Execute initialization of the internal state according to the setting of bits CLR3 to CLR0 or ICE bit clearing.
2. Execute a stop condition issuance instruction (write 0 to BBSY and SCP) to clear the BBSY bit to 0, and wait for two transfer rate clock cycles.
3. Re-execute initialization of the internal state according to the setting of bits CLR3 to CLR0 or ICE bit clearing.
4. Initialize (re-set) the IIC registers.

18.5 Interrupt Sources

The IIC has interrupt source IICI. Table 18.8 shows the interrupt sources and priority. Individual interrupt sources can be enabled or disabled using the enable bits in ICCR, and are sent to the interrupt controller independently.

The IIC interrupts are used as on-chip DTC activation sources.

Table 18.8 IIC Interrupt Sources

Channel	Name	Enable Bit	Interrupt Source	Interrupt Flag	Priority
0	IICI0	IEIC	I²C bus interface interrupt request	IRIC	High
1	IICI1	IEIC	I²C bus interface interrupt request	IRIC	
2	IICI2	IEIC	I²C bus interface interrupt request	IRIC	Low

18.6 Usage Notes

1. In master mode, if an instruction to generate a start condition is issued and then an instruction to generate a stop condition is issued before the start condition is output to the I²C bus, neither condition will be output correctly.
2. Either of the following two conditions will start the next transfer. Pay attention to these conditions when accessing ICDR.
   — Write to ICDR when ICE = 1 and TRS = 1 (including automatic transfer from ICDRT to ICDRS)
   — Read from ICDR when ICE = 1 and TRS = 0 (including automatic transfer from ICDRS to ICDRR)
3. Table 18.9 shows the timing of SCL and SDA outputs in synchronization with the internal clock. Timings on the bus are determined by the rise and fall times of signals affected by the bus load capacitance, series resistance, and parallel resistance.

Table 18.9 I²C Bus Timing (SCL and SDA Outputs)

Item	Symbol	Output Timing	Unit	Notes
SCL output cycle time	t_{SCLO}	$28t_{cyc}$ to $256t_{cyc}$	ns	See figure 28.24 (for reference)
SCL output high pulse width	t_{SCLHO}	$0.5t_{SCLO}$	ns	
SCL output low pulse width	t_{SCLLO}	$0.5t_{SCLO}$	ns	
SDA output bus free time	t_{BUFO}	$0.5t_{SCLO} – 1t_{cyc}$	ns	
Start condition output hold time	t_{STAHO}	$0.5t_{SCLO} – 1t_{cyc}$	ns	
Retransmission start condition output setup time	t_{STASO}	$1t_{SCLO}$	ns	
Stop condition output setup time	t_{STOSO}	$0.5t_{SCLO} + 2t_{cyc}$	ns	
Data output setup time (master)	t_{SDASO}	$1t_{SCLLO} – 3t_{cyc}$	ns	
Data output setup time (slave)		$1t_{SCLL} – (6t_{cyc}$ or $12t_{cyc}$ *)		
Data output hold time	t_{SDAHO}	$3t_{cyc}$	ns	

Note: * $6t_{cyc}$ when IICX is 0, $12t_{cyc}$ when 1.

4. The I²C bus interface specification for the SCL rise time t_{sr} is 1000 ns or less (300 ns for high-speed mode). In master mode, the I²C bus interface monitors the SCL line and synchronizes one bit at a time during communication. If t_{sr} (the time for SCL to go from low to V_{IH}) exceeds the time determined by the input clock of the I²C bus interface, the high period of SCL is extended. The SCL rise time is determined by the pull-up resistance and load capacitance of the SCL line. To insure proper operation at the set transfer rate, adjust the pull-up resistance and load capacitance so that the SCL rise time does not exceed the values given in table 18.10.

Table 18.10 Permissible SCL Rise Time (t_{sr}) Values

			Time Indication [ns]				
IICX	t_{cyc} Indication		I²C Bus Specification (Max.)	φ = 8 MHz	φ = 10 MHz	φ = 16 MHz	φ = 20 MHz
0	7.5 t_{cyc}	Standard mode	1000	937	750	468	375
		High-speed mode	300	300	300	300	300
1	17.5 t_{cyc}	Standard mode	1000	1000	1000	1000	875
		High-speed mode	300	300	300	300	300

5. The I²C bus interface specifications for the SCL and SDA rise and fall times are under 1000 ns and 300 ns. The I²C bus interface SCL and SDA output timing is prescribed by t_{cyc}, as shown in table 18.11. However, because of the rise and fall times, the I²C bus interface specifications may not be satisfied at the maximum transfer rate. Table 18.11 shows output timing calculations for different operating frequencies, including the worst-case influence of rise and fall times.

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

   t_{SCLLO} in high-speed mode and t_{STASO} in standard mode fail to satisfy the I²C bus interface specifications for worst-case calculations of t_{Sr}/t_{Sf}. Possible solutions that should be investigated include (a) adjusting the rise and fall times by means of a pull-up resistor and capacitive load, (b) reducing the transfer rate to meet the specifications, or (c) selecting devices whose input timing permits this output timing for use as slave devices connected to the I²C bus.

Table 18.11 I²C Bus Timing (with Maximum Influence of t_{Sr}/t_{Sf})

Item	t_{cyc} Indication		Time Indication (at Maximum Transfer Rate) [ns]: t_{Sr}/t_{Sf} Influence (Max.)	I²C Bus Specification (Min.)	ϕ = 8 MHz	ϕ = 10 MHz	ϕ = 16 MHz	ϕ = 20 MHz
t_{SCLHO}	0.5 t_{SCLO} ($-t_{Sr}$)	Standard mode	–1000	4000	4000	4000	4000	4000
		High-speed mode	–300	600	950	950	950	950
t_{SCLLO}	0.5 t_{SCLO} ($-t_{Sf}$)	Standard mode	–250	4700	4750	4750	4750	4750
		High-speed mode	–250	1300	1000*[1]	1000*[1]	1000*[1]	1000*[1]
t_{BUFO}	0.5 t_{SCLO}–1 t_{cyc} ($-t_{Sr}$)	Standard mode	–1000	4700	3875*[1]	3900*[1]	3939*[1]	3950*[1]
		High-speed mode	–300	1300	825*[1]	850*[1]	888*[1]	900*[1]
t_{STAHO}	0.5 t_{SCLO}–1 t_{cyc} ($-t_{Sf}$)	Standard mode	–250	4000	4625	4650	4688	4700
		High-speed mode	–250	600	875	900	938	900
t_{STASO}	1 t_{SCLO} ($-t_{Sr}$)	Standard mode	–1000	4700	9000	9000	9000	9000
		High-speed mode	–300	600	2200	2200	2200	2200
t_{STOSO}	0.5 t_{SCLO} + 2 t_{cyc} ($-t_{Sr}$)	Standard mode	–1000	4000	4250	4200	4125	4100
		High-speed mode	–300	600	1200	1150	1075	1050
t_{SDASO} (master)	1 t_{SCLLO}*[3] –3 t_{cyc} ($-t_{Sr}$)	Standard mode	–1000	250	3325	3400	3513	3550
		High-speed mode	–300	100	625	700	813	850
t_{SDASO} (slave)	1 t_{SCLL}*[3] –12 t_{cyc}*[2] ($-t_{Sr}$)	Standard mode	–1000	250	2200	2500	2950	3100
		High-speed mode	–300	100	–500*[1]	–200*[1]	250	400
t_{SDAHO}	3 t_{cyc}	Standard mode	0	0	375	300	188	150
		High-speed mode	0	0	375	300	188	150

Notes: 1. Does not meet the I²C bus interface specification. Remedial action such as the following is necessary: (a) secure a start/stop condition issuance interval; (b) adjust the rise and fall times by means of a pull-up resistor and capacitive load; (c) reduce the transfer rate; (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 IICX bit and bits CKS0 to CKS2. Depending on the frequency it may not be possible to achieve the maximum transfer rate; therefore, whether or not the I²C bus interface specifications are met must be determined in accordance with the actual setting conditions.

2. Value when the IICX bit is set to 1. When the IICX bit is cleared to 0, the value is (t_{SCLL} – 6 t_{cyc}).
3. Calculated using the I²C bus specification values (standard mode: 4700 ns min.; high-speed mode: 1300 ns min.).

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

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

7. Note on ACKE and TRS bits in slave mode

   In the I²C bus interface, if 1 is received as the acknowledge bit value (ACKB = 1) in transmit mode (TRS = 1) and then the address is received in slave mode without performing appropriate processing, interrupt handling may start at the rising edge of the 9th clock pulse even when the address does not match. Similarly, if the start condition or address is transmitted from the master device in slave transmit mode (TRS = 1), the IRIC flag may be set after the ICDRE flag is set and 1 received as the acknowledge bit value (ACKB = 1), thus causing an interrupt source even when the address does not match.

   To use the I²C bus interface module in slave mode, be sure to follow the procedures below.

   A. When having received 1 as the acknowledge bit value for the last transmit data at the end of a series of transmit operation, clear the ACKE bit in ICCR once to initialize the ACKB bit to 0.

   B. Set receive mode (TRS = 0) before the next start condition is input in slave mode. Complete transmit operation by the procedure shown in figure 18.16, in order to switch from slave transmit mode to slave receive mode.

18.6.1 Module Stop Mode Setting

The IIC operation can be enabled or disabled using the module stop control register. The initial setting is for the IIC operation to be halted. Register access is enabled by canceling module stop mode. For details, see section 26, Power-Down Modes.

Section 19 Keyboard Buffer Control Unit (PS2)

This LSI has four on-chip keyboard buffer control unit (PS2) channels. The PS2 is provided with functions conforming to the PS/2 interface specifications.

Data transfer using the PS2 employs a data line (KD) and a clock line (KCLK), providing economical use of connectors, board surface area, etc. Figure 19.1 shows a block diagram of the PS2.

19.1 Features

- Conforms to PS/2 interface specifications
- Direct bus drive (via the KCLK and KD pins)
- Interrupt sources: on completion of data reception/transmission, on detection of clock falling edge, and on detection of the first falling edge of a clock
- Error detection: parity error, stop bit monitoring, and receive notify monitoring

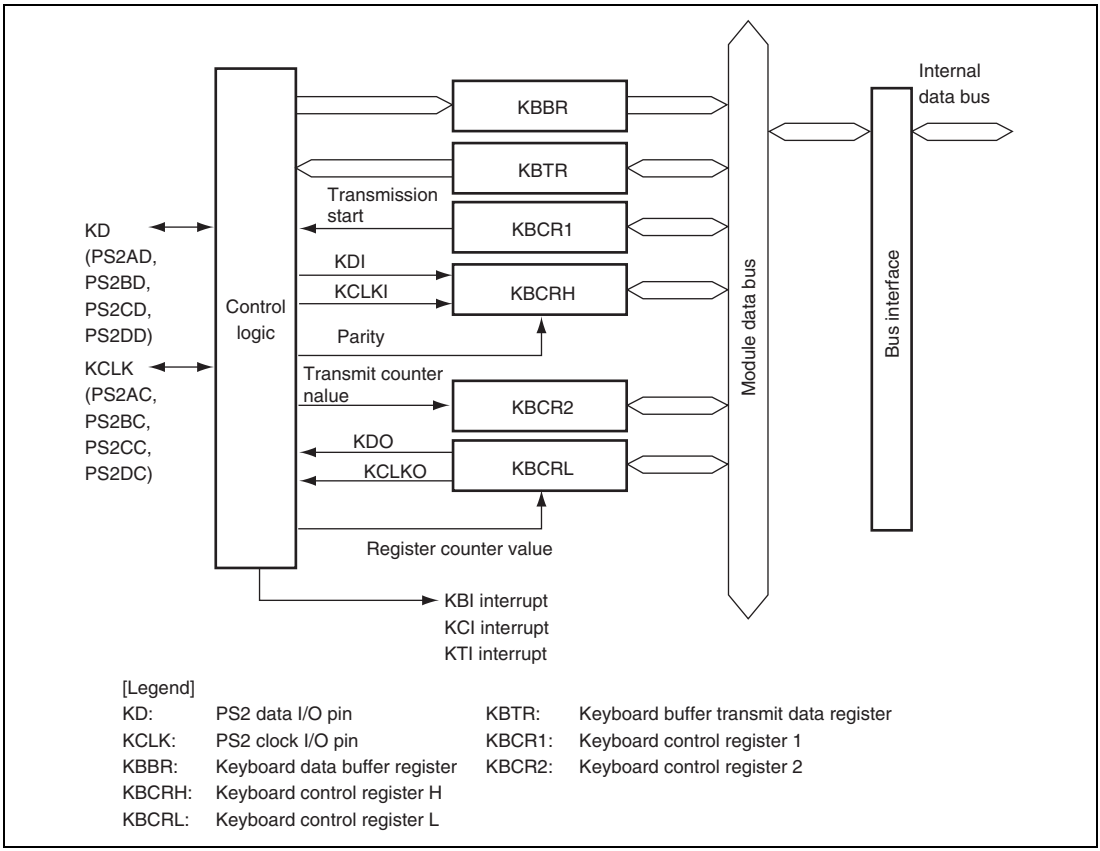


Figure 19.1 Block Diagram of PS2

Figure 19.2 shows how the PS2 is connected.

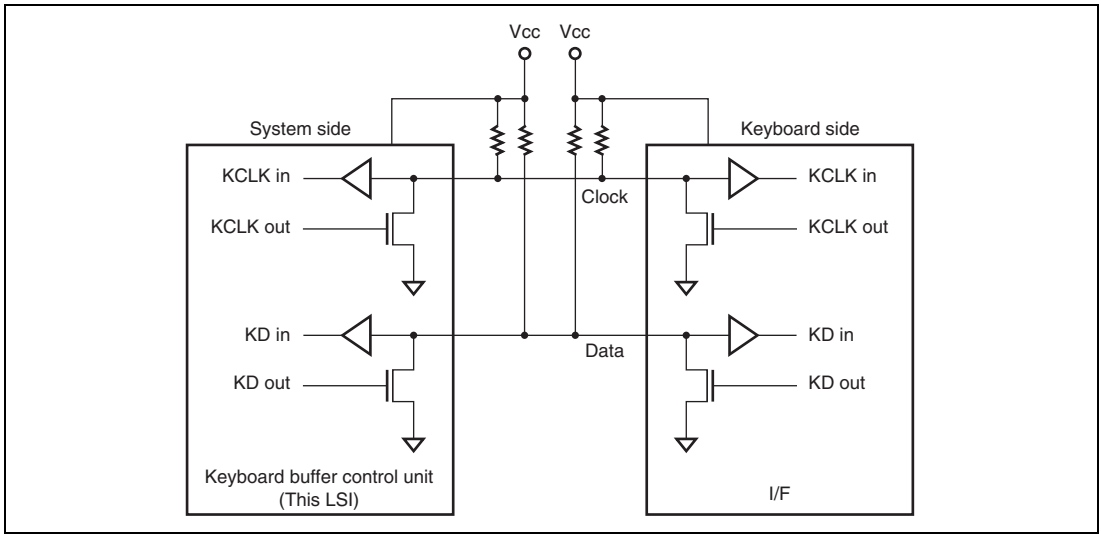


Figure 19.2 PS2 Connection

19.2 Input/Output Pins

Table 19.1 lists the input/output pins used by the keyboard buffer control unit.

Table 19.1 Pin Configuration

Channel	Name	Abbreviation*	I/O	Function
0	PS2 clock I/O pin (KCLK0)	PS2AC	I/O	PS2 clock input/output
	PS2 data I/O pin (KD0)	PS2AD	I/O	PS2 data input/output
1	PS2 clock I/O pin (KCLK1)	PS2BC	I/O	PS2 clock input/output
	PS2 data I/O pin (KD1)	PS2BD	I/O	PS2 data input/output
2	PS2 clock I/O pin (KCLK2)	PS2CC	I/O	PS2 clock input/output
	PS2 data I/O pin (KD2)	PS2CD	I/O	PS2 data input/output
3	PS2 clock I/O pin (KCLK3)	PS2DC	I/O	PS2 clock input/output
	PS2 data I/O pin (KD3)	PS2DD	I/O	PS2 data input/output

Note: * These are the external I/O pin names. In the text, clock I/O pins are referred to as KCLK and data I/O pins as KD, omitting the channel designations.

19.3 Register Descriptions

The PS2 has the following registers for each channel.

Table 19.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Data Bus Width
Channel 0	Keyboard control register 1_0	KBCR1_0	R/W	H'00	H'FEC0	8
	Keyboard control register 2_0	KBCR2_0	R/W	H'F0	H'FEDB	8
	Keyboard buffer transmit data register_0	KBTR_0	R/W	H'FF	H'FEC1	8
	Keyboard control register H_0	KBCRH_0	R/W	H'70	H'FED8	8
	Keyboard control register L_0	KBCRL_0	R/W	H'70	H'FED9	8
	Keyboard data buffer register_0	KBBR_0	R	H'00	H'FEDA	8
Channel 1	Keyboard control register 1_1	KBCR1_1	R/W	H'00	H'FEC2	8
	Keyboard control register 2_1	KBCR2_1	R/W	H'F0	H'FEDF	8
	Keyboard buffer transmit data register_1	KBTR_1	R/W	H'FF	H'FEC3	8
	Keyboard control register H_1	KBCRH_1	R/W	H'70	H'FEDC	8
	Keyboard control register L_1	KBCRL_1	R/W	H'70	H'FEDD	8
	Keyboard data buffer register_1	KBBR_1	R	H'00	H'FEDE	8
Channel 2	Keyboard control register 1_2	KBCR1_2	R/W	H'00	H'FEC4	8
	Keyboard control register 2_2	KBCR2_2	R/W	H'F0	H'FEE3	8
	Keyboard buffer transmit data register_2	KBTR_2	R/W	H'FF	H'FEC5	8
	Keyboard control register H_2	KBCRH_2	R/W	H'70	H'FEE0	8
	Keyboard control register L_2	KBCRL_2	R/W	H'70	H'FEE1	8
	Keyboard data buffer register_2	KBBR_2	R	H'00	H'FEE2	8
Channel 3	Keyboard control register 1_3	KBCR1_3	R/W	H'00	H'FED2	8
	Keyboard control register 2_3	KBCR2_3	R/W	H'F0	H'FFE3	8
	Keyboard buffer transmit data register_3	KBTR_3	R/W	H'FF	H'FED3	8
	Keyboard control register H_3	KBCRH_3	R/W	H'70	H'FFE0	8
	Keyboard control register L_3	KBCRL_3	R/W	H'70	H'FFE1	8
	Keyboard data buffer register_3	KBBR_3	R	H'00	H'FFE2	8

19.3.1 Keyboard Control Register 1 (KBCR1)

KBCR1 controls data transmission and interrupt, selects parity, and detects transmit error.

Bit	Bit Name	Initial Value	R/W	Description
7	KBTS	0	R/W	Transmit Start Selects start of data transmission or disables transmission. 0: Data transmission is disabled [Clearing conditions] • When 0 is written • When the KBTE is set to 1 • When the KBIOE is cleared to 0 1: Starts data transmission [Setting condition] When 1 is written after reading the KBTS = 0
6	PS	0	R/W	Transmit Parity Selection Selects even or odd parity. 0: Selects odd parity 1: Selects even parity
5	KCIE	0	R/W	First KCLK Falling Interrupt Enable Selects whether an interrupt at the first falling edge of KCLK is enabled or disabled. 0: Disables first KCLK falling interrupt 1: Enables first KCLK falling interrupt
4	KTIE	0	R/W	Transmit Completion Interrupt Enable Selects whether a transmit completion interrupt is enabled or disabled. 0: Disables transmit completion interrupt 1: Enables transmit completion interrupt
3	—	0	—	Reserved The initial value should not be changed.

Bit	Bit Name	Initial Value	R/W	Description
2	KCIF	0	R/(W)*	First KCLK Falling Interrupt Flag Indicates that the first falling edge of KCLK is detected. When KCIE and KCIF are set to 1, requests the CPU an interrupt. 0: [Clearing condition] After reading KCIF = 1, 0 is written 1: [Setting condition] When the first falling edge of KCLK is detected Note that this flag cannot be set when software standby mode or watch mode is cancelled. (However, internal flag is set.)
1	KBTE	0	R/(W)*	Transmit Completion Flag Indicates that data transmission is completed. When KTIE and KBTE are set to 1, requests the CPU an interrupt. 0: [Clearing condition] After reading KBTE = 1, 0 is written 1: [Setting Condition] When all KBTR data has been transmitted (Set at the eleventh rising edge of the KCLK signal)
0	KTER	0	R	Transmit Error Stores a notification of receive completion. Valid only when KBTE = 1. 0: 0 received as a notification of receive completion. 1: 1 received as a notification of receive completion.

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

19.3.2 Keyboard Buffer Control Register 2 (KBCR2)

KBCR2 is a 4-bit counter which performs counting synchronized with the falling edge of KCLK. Transmit data is synchronized with the transmit counter, and data in the KBTR is sent to the KD (LSB-first).

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	R/W	Reserved These bits are always read as 0. The initial value should not be changed.
3	TXCR3	0	R	Transmit Counter Indicates bit of transmit data. Counter is incremented at the falling edge of KCLK. The transmit counter is initialized by a reset, when the KBTS is cleared to 0, the KBIOE is cleared to 0, or the KBTE is set to 1. 0000: Clear 0001: KBT0 0010: KBT1 0011: KBT2 0100: KBT3 0101: KBT4 0110: KBT5 0111: KBT6 1000: KBT7 1001: Parity bit 1010: Stop bit 1011: Transmit completion notification
2	TXCR2	0	R	
1	TXCR1	0	R	
0	TXCR0	0	R	

19.3.3 Keyboard Control Register H (KBCRH)

KBCRH indicates the operating status of the keyboard buffer control unit.

Bit	Bit Name	Initial Value	R/W	Description
7	KBIOE	0	R/W	Keyboard In/Out Enable Selects whether or not the keyboard buffer control unit is used. 0: The keyboard buffer control unit is non-operational (KCLK and KD signal pins have port functions) 1: The keyboard buffer control unit is enabled for transmission and reception (KCLK and KD signal pins are in the bus drive state)
6	KCLKI	1	R	Keyboard Clock In Monitors the KCLK I/O pin. This bit cannot be modified. 0: KCLK I/O pin is low 1: KCLK I/O pin is high
5	KDI	1	R	Keyboard Data In Monitors the KDI I/O pin. This bit cannot be modified. 0: KD I/O pin is low 1: KD I/O pin is high
4	KBFSEL	1	R/W	Keyboard Buffer Register Full Select Selects whether the KBF bit is used as the keyboard buffer register full flag or as the KCLK fall interrupt flag. When KBF bit is used as the KCLK fall interrupt flag, the KBE bit in KBCRL should be cleared to 0 to disable reception. 0: KBF bit is used as KCLK fall interrupt flag 1: KBF bit is used as keyboard buffer register full flag
3	KBIE	0	R/W	Keyboard Interrupt Enable Enables or disables interrupts from the keyboard buffer control unit to the CPU. 0: Interrupt requests are disabled 1: Interrupt requests are enabled

Bit	Bit Name	Initial Value	R/W	Description
2	KBF	0	R/(W)*	Keyboard Buffer Register Full Indicates that data reception has been completed and the received data is in KBBR. When both KBIE and KBF are set to1, an interrupt request is sent to the CPU. 0: [Clearing condition] Read KBF when KBF =1, then write 0 in KBF 1: [Setting conditions] • When data has been received normally and has been transferred to KBBR while KBFSEL = 1 (keyboard buffer register full flag) • When a KCLK falling edge is detected while KBFSEL = 0 (KCLK interrupt flag)
1	PER	0	R/(W)*	Parity Error Indicates that an odd parity error has occurred. 0: [Clearing condition] Read PER when PER =1, then write 0 in PER 1: [Setting condition] When an odd parity error occurs
0	KBS	0	R	Keyboard Stop Indicates the receive data stop bit. Valid only when KBF = 1. 0: 0 stop bit received 1: 1 stop bit received

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

19.3.4 Keyboard Control Register L (KBCRL)

KBCRL enables the receive counter count and controls the keyboard buffer control unit pin output.

Bit	Bit Name	Initial Value	R/W	Description
7	KBE	0	R/W	Keyboard Enable Enables or disables loading of receive data into KBBR. 0: Loading of receive data into KBBR is disabled 1: Loading of receive data into KBBR is enabled
6	KCLKO	1	R/W	Keyboard Clock Out Controls PS2 clock I/O pin output. 0: PS2 clock I/O pin is low 1: PS2 clock I/O pin is high
5	KDO	1	R/W	Keyboard Data Out Controls PS2 data I/O pin output. 0: PS2 data I/O pin is low 1: PS2 data I/O pin is high When the start bit (KDO) is automatically cleared (KDO = 1) by means of automatic transmission, 0 is written after reading 1.
4	—	1	—	Reserved This bit is always read as 1 and cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
3	RXCR3	0	R	Receive Counter
2	RXCR2	0	R	These bits indicate the received data bit. Their value is incremented on the fall of KCLK. These bits cannot be modified.
1	RXCR1	0	R	
0	RXCR0	0	R	The receive counter is initialized by a reset and when 0 is written in KBE. Its value returns to B'0000 after a stop bit is received.
				0000: —
				0001: Start bit
				0010: KB0
				0011: KB1
				0100: KB2
				0101: KB3
				0110: KB4
				0111: KB5
				1000: KB6
				1001: KB7
				1010: Parity bit
				1011: —
				11- -: —

19.3.5 Keyboard Data Buffer Register (KBBR)

KBBR stores receive data. Its value is valid only when KBF = 1.

Bit	Bit Name	Initial Value	R/W	Description
7	KB7	0	R	Keyboard Data 7 to 0
6	KB6	0	R	8-bit read only data.
5	KB5	0	R	Initialized to H'00 by a reset or when KBIOE is cleared to 0.
4	KB4	0	R	
3	KB3	0	R	
2	KB2	0	R	
1	KB1	0	R	
0	KB0	0	R	

19.3.6 Keyboard Buffer Transmit Data Register (KBTR)

KBTR stores transmit data.

Bit	Bit Name	Initial Value	R/W	Description
7	KBT7	1	R/W	Keyboard Buffer Transmit Data Register 7 to 0
6	KBT6	1	R/W	Initialized to H'00 at reset.
5	KBT5	1	R/W	
4	KBT4	1	R/W	
3	KBT3	1	R/W	
2	KBT2	1	R/W	
1	KBT1	1	R/W	
0	KBT0	1	R/W	

19.4 Operation

19.4.1 Receive Operation

In a receive operation, both KCLK (clock) and KD (data) are outputs on the keyboard side and inputs on this LSI chip (system) side. KD receives a start bit, 8 data bits (LSB-first), an odd parity bit, and a stop bit, in that order. The KD value is valid when KCLK is low. Value of KD is valid when the KCLK is low. A sample receive processing flowchart is shown in figure 19.3, and the receive timing in figure 19.4.

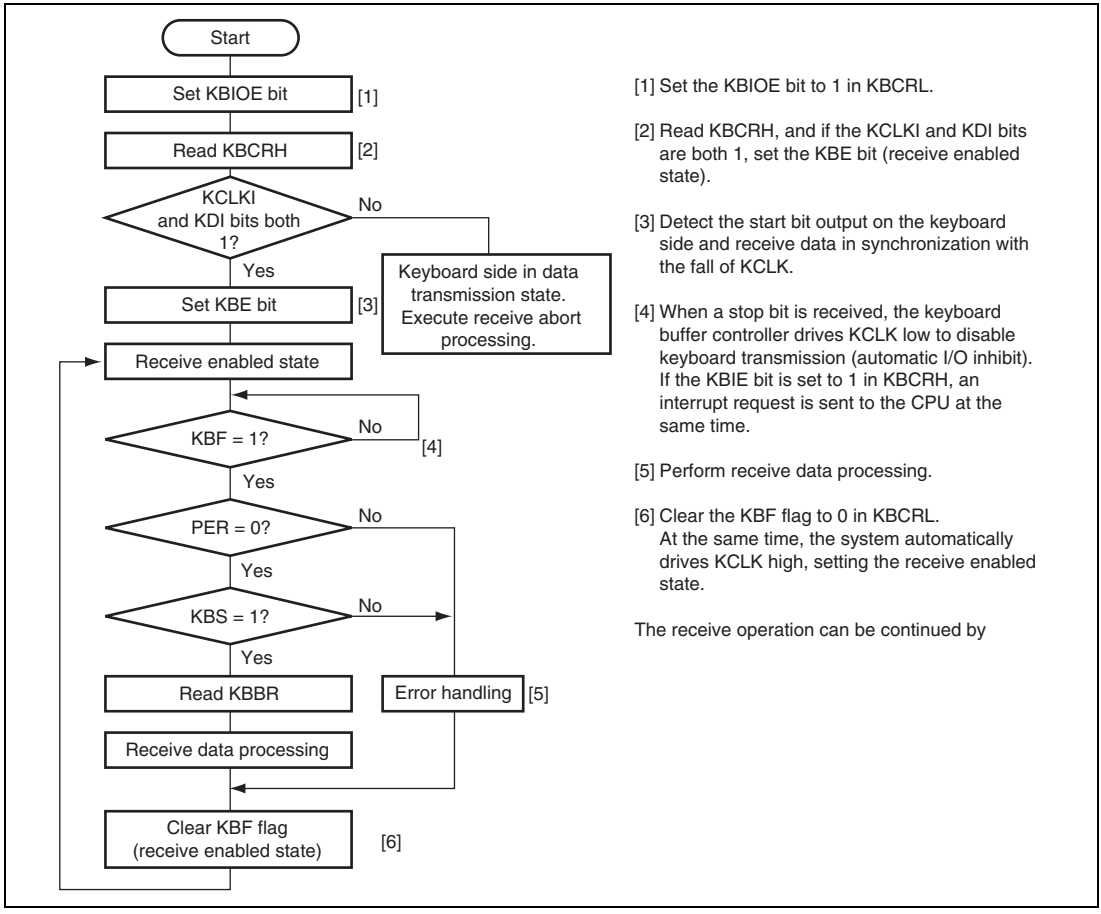


Figure 19.3 Sample Receive Processing Flowchart

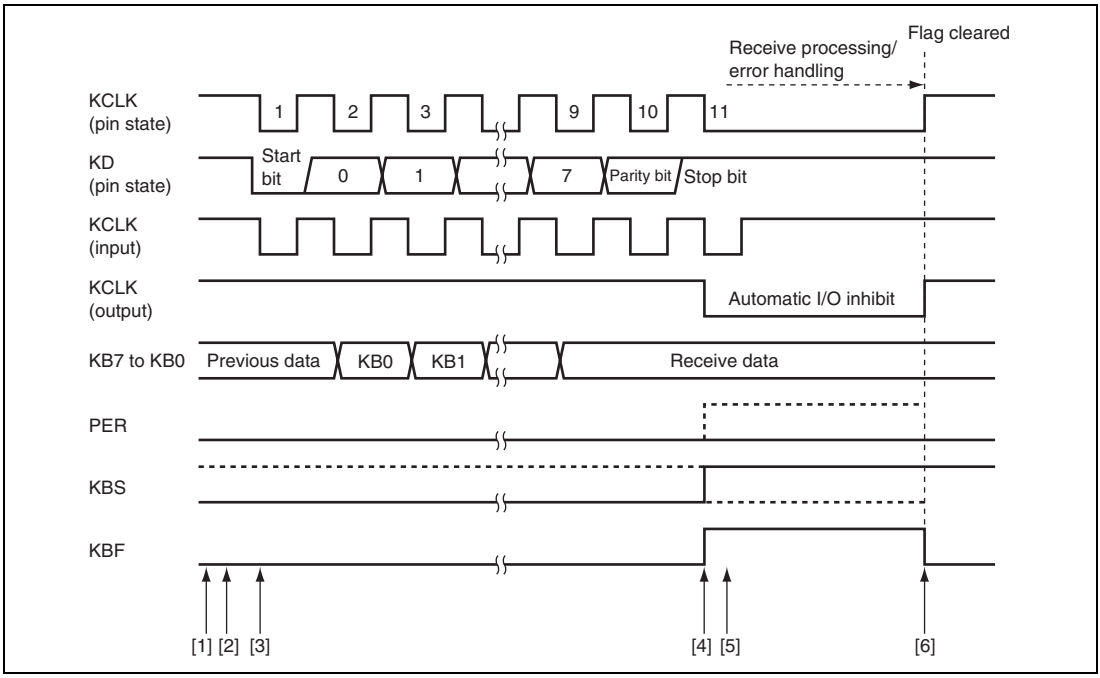


Figure 19.4 Receive Timing

19.4.2 Transmit Operation

In a transmit operation, KCLK (clock) is an output on the keyboard side, and KD (data) is an output on the chip (system) side. KD outputs a start bit, 8 data bits (LSB-first), an odd parity bit, and a stop bit, in that order. The KD value is valid when KCLK is high. A sample transmit processing flowchart is shown in figure 19.5, and the transmit timing in figure 19.6.

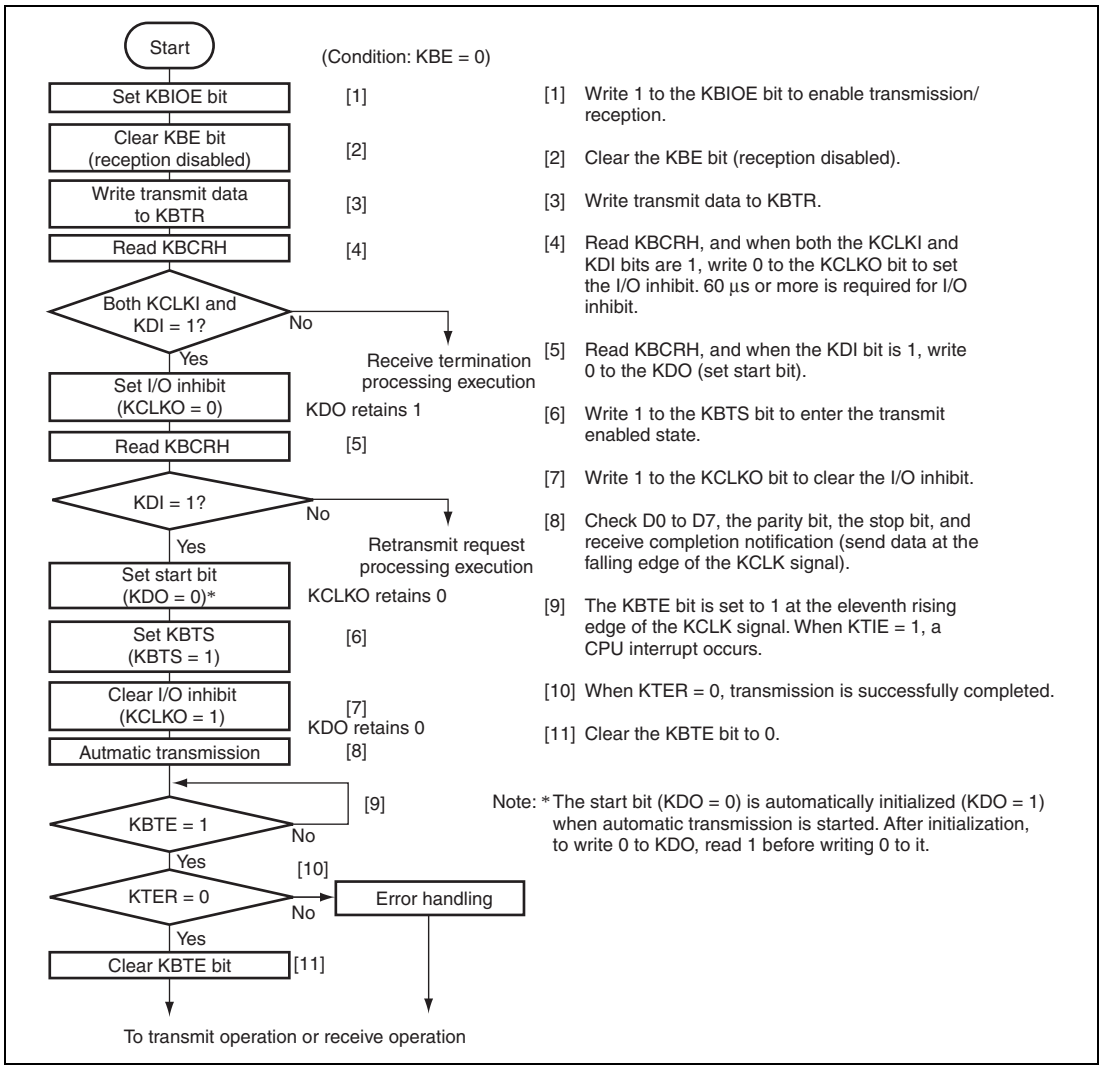


Figure 19.5 Sample Transmit Processing Flowchart

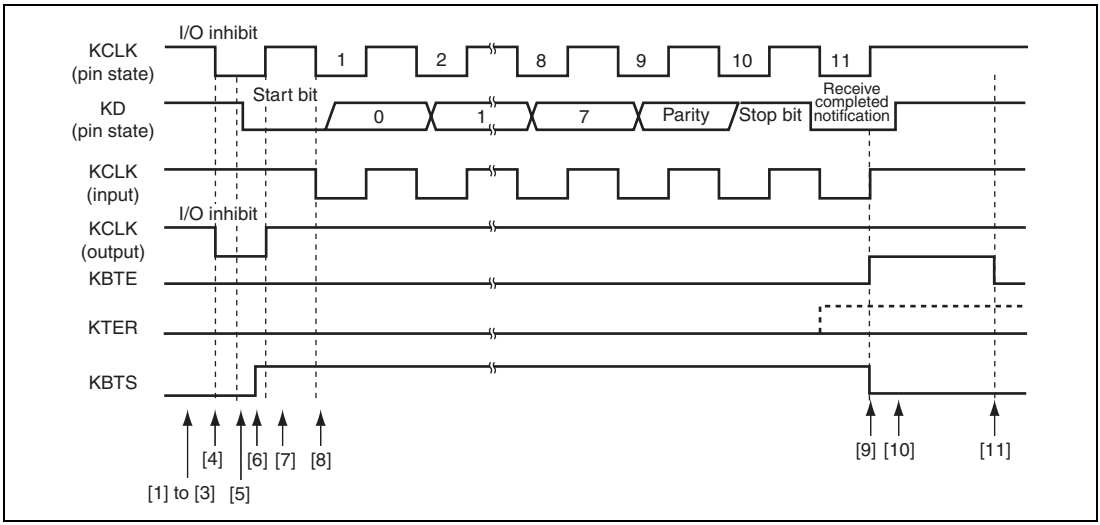


Figure 19.6 Transmit Timing

19.4.3 Receive Abort

This LSI (system side) can forcibly abort transmission from the device connected to it (keyboard side) in the event of a protocol error, etc. In this case, the system holds the clock low. During reception, the keyboard also outputs a clock for synchronization, and the clock is monitored when the keyboard output clock is high. If the clock is low at this time, the keyboard judges that there is an abort request from the system, and data transmission from the keyboard is aborted. Thus the system can abort reception by holding the clock low for a certain period. A sample receive abort processing flowchart is shown in figure 19.7, and the receive abort timing in figure 19.8.

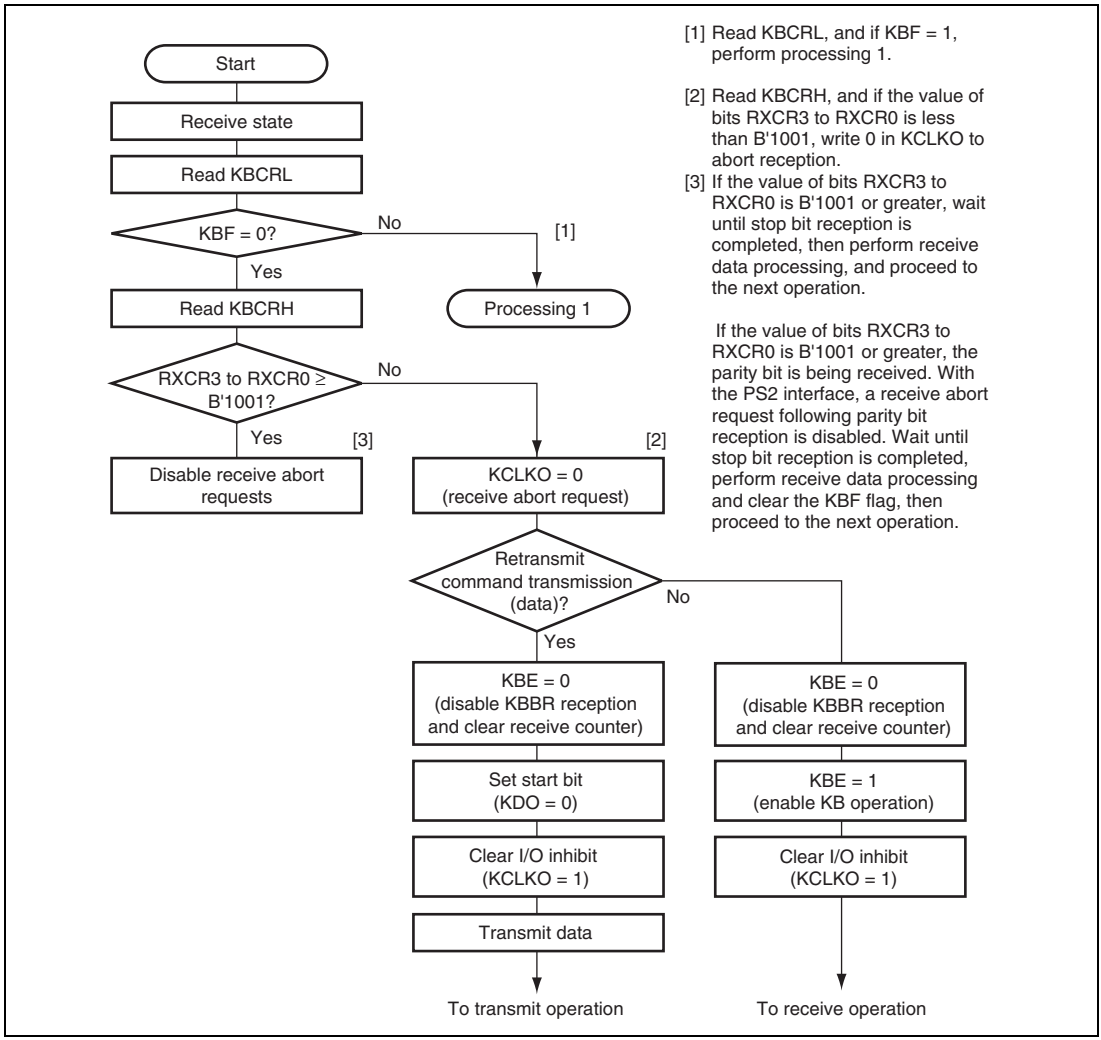


Figure 19.7 Sample Receive Abort Processing Flowchart (1)

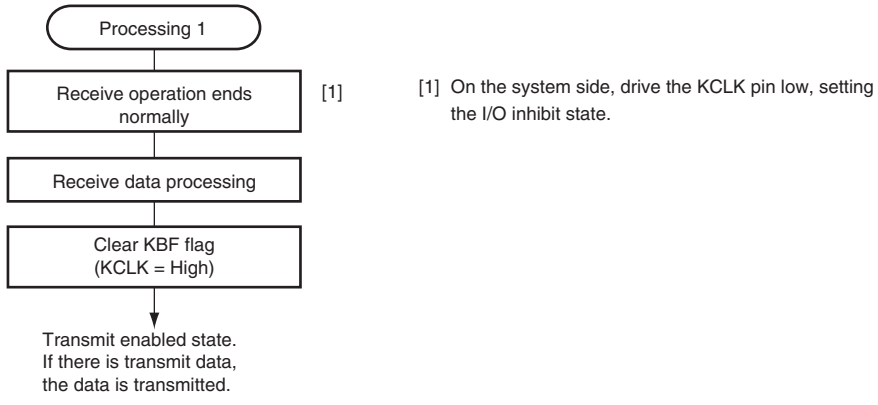


Figure 19.7 Sample Receive Abort Processing Flowchart (2)

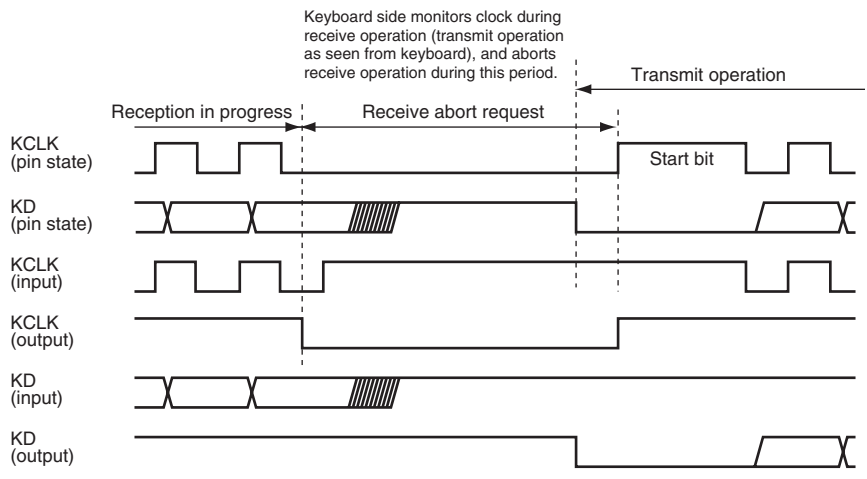


Figure 19.8 Receive Abort and Transmit Start (Transmission/Reception Switchover) Timing

19.4.4 KCLKI and KDI Read Timing

Figure 19.9 shows the KCLKI and KDI read timing.

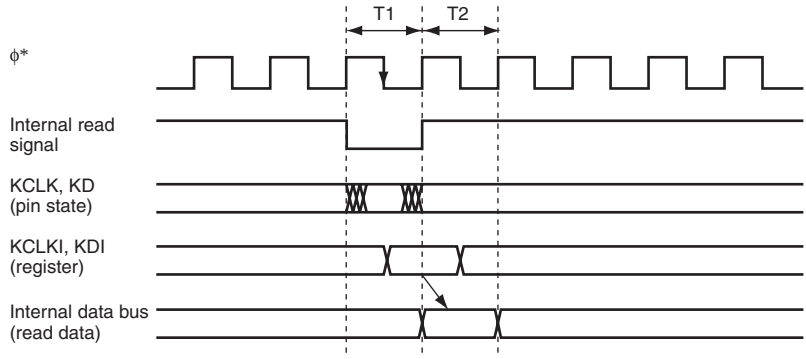


Figure 19.9 KCLKI and KDI Read Timing

19.4.5 KCLKO and KDO Write Timing

Figure 19.10 shows the KLCKO and KDO write timing and the KCLK and KD pin states.

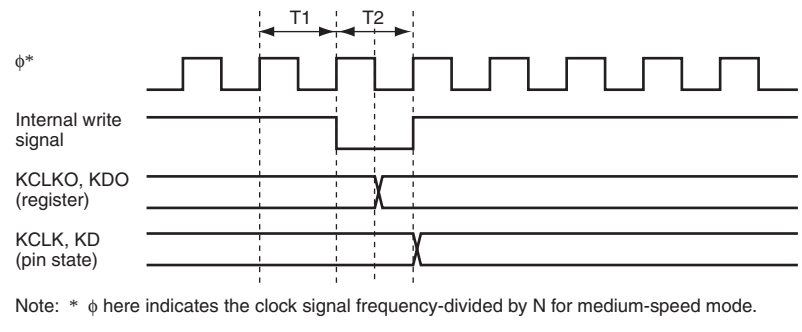


Figure 19.10 KCLKO and KDO Write Timing

19.4.6 KBF Setting Timing and KCLK Control

Figure 19.11 shows the KBF setting timing and the KCLK pin states.

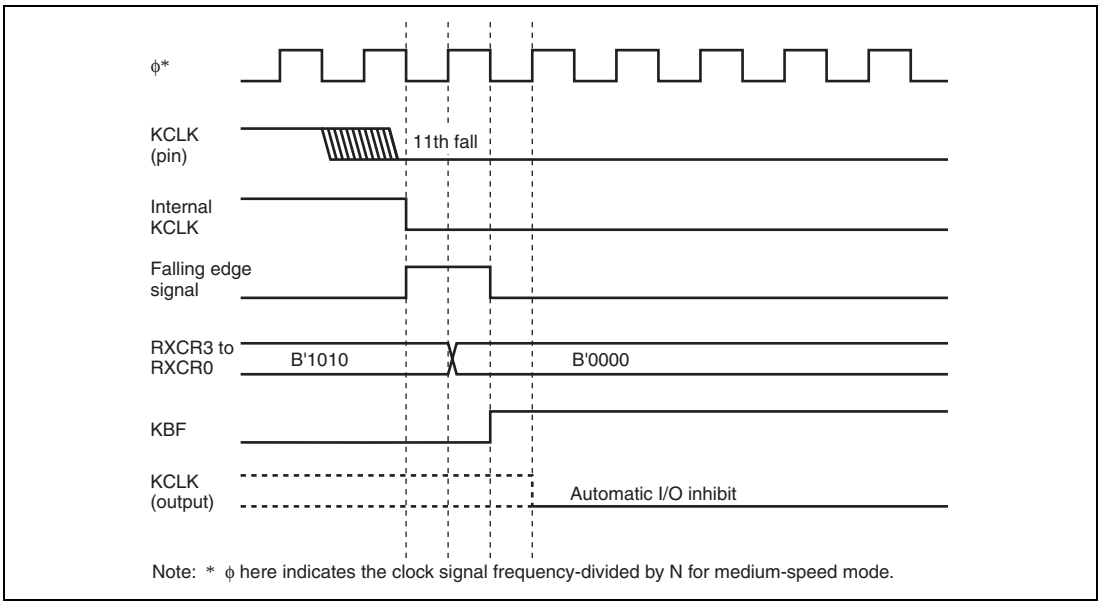


Figure 19.11 KBF Setting and KCLK Automatic I/O Inhibit Generation Timing

19.4.7 Receive Timing

Figure 19.12 shows the receive timing.

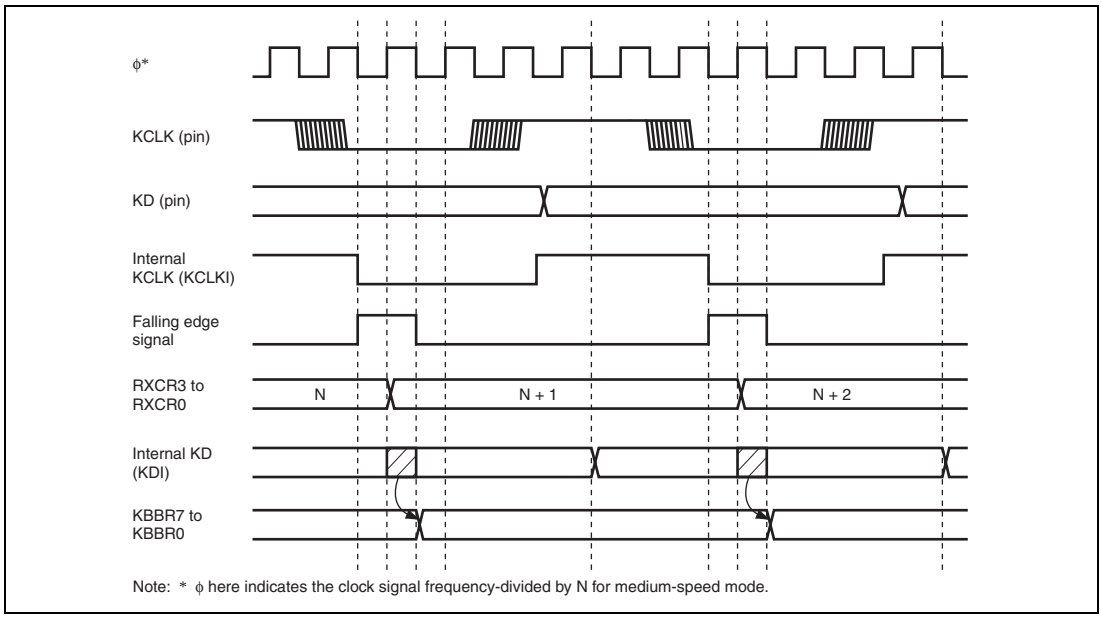


Figure 19.12 Receive Counter and KBBR Data Load Timing

19.4.8 Operation during Data Reception

If the KBS bit in KBCRH is set to 1 with other keyboard buffer control units in reception*, the KCLK is automatically pulled down. Figure 19.13 shows receive timing and the KCLK.

Note: * Period from the first falling edge of KCLK to completion of reception (KBF = 1).

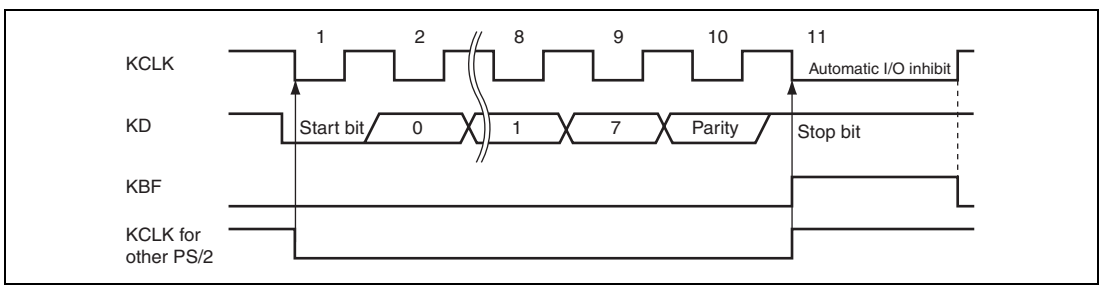
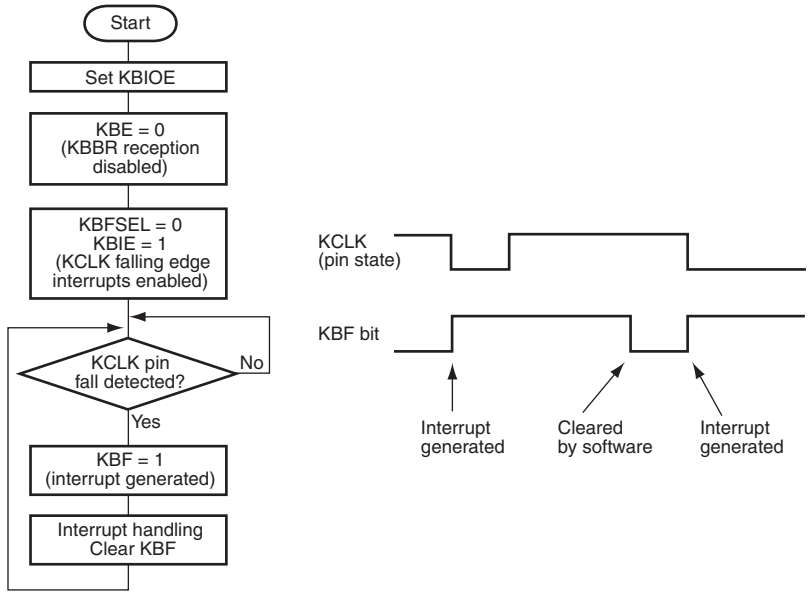


Figure 19.13 Receive Timing and KCLK

19.4.9 KCLK Fall Interrupt Operation

In this device, clearing the KBFSEL bit to 0 in KBCRH enables the KBF bit in KBCRH to be used as a flag for the interrupt generated by the fall of KCLK input.

Figure 19.14 shows the setting method and an example of operation.



Note: * The KBF setting timing is the same as the timing of KBF setting and KCLK automatic I/O inhibit bit generation in figure 17.11. When the KBF bit is used as the KCLK input fall interrupt flag, the automatic I/O inhibit function does not operate.

Figure 19.14 Example of KCLK Input Fall Interrupt Operation

19.4.10 First KCLK Falling Interrupt

An interrupt can be generated by detecting the first falling edge of KCLK on reception and transmission. Software standby mode and watch mode can be cancelled by a first KCLK falling interrupt.

- Reception

  When both KBIOE and KBE are set to 1, KCIF is set after the first falling edge of KCLK has been detected.

  At this time, if KCIE is set to 1, the CPU is requested an interrupt.

  KCIF is set at the same time when the RXCR3 to RXCR0 bits in KBCRL are incremented from B'0000 to B'0001.

- Transmission

  When both KBIOE and KBTS are set to 1, the KCIF is set after the first falling edge of KCLK has been detected.

  At this time, if KCIE is set to 1, the CPU is requested an interrupt.

  KCIF is set at the same time when the TXCR3 to TXCR0 bits in KBCR2 are incremented from B'0000 to B'0001.

- Determining interrupt generation

  By checking the KBE, KBTS, and KBTE bits, it can be determined whether the first KCLK falling interrupt is occurred during reception or transmission.

  During reception: KBE = 1

  During transmission: KBTS = 1 or KBTE = 1 (Check KBTE = 1 because the KBTS is automatically cleared after transfer has been completed.)

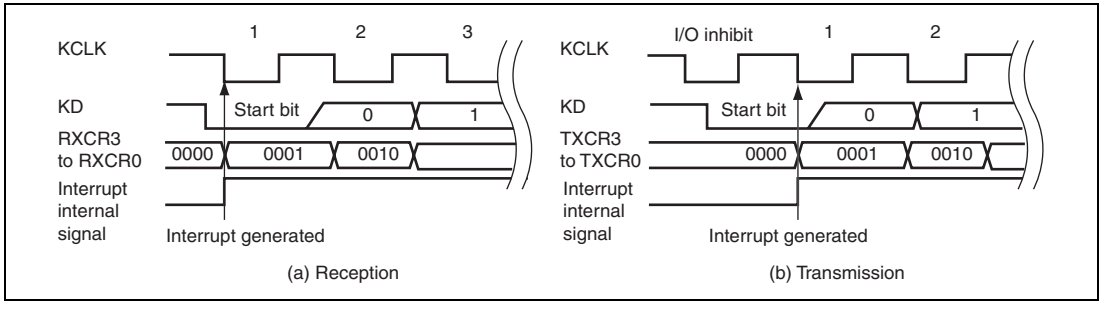


Figure 19.15 Timing of First KCLK Interrupt

- Canceling software standby mode and watch mode

  Software standby mode and watch mode are cancelled by a first KCLK falling interrupt. In this case, an interrupt is generated at the first KCLK since software standby mode or watch mode has been shifted (figure 19.17).

  Notes on canceling operation are explained below.

  — When a transition to software standby mode or watch mode is performed while both KBIOE and KCIE are set to 1, canceling the current mode is enabled by a first KCLK falling interrupt (the KBE and KBTS are not affected).

  — When software standby mode and watch mode are cancelled by a first KCLK falling interrupt, the KCIF flag is not set (only the internal flag is set).
  In the first KCLK interrupt handling routine, the KCIF bit is checked. If the KCIF is 0, it indicates that the interrupt is generated after software standby mode and watch mode have been cancelled.

  — When software standby mode or watch mode is cancelled by receiving a receive clock, the reception is ignored. Execute reception terminating processing by an interrupt handing routine, and then request retransfer.

  — When transition to software standby mode or watch mode is made and the mode is canceled by a first KCLK falling interrupt during data transmission, state before performing mode transition is held immediately after canceling the mode. Therefore, initialization by an interrupt handling routine is required. Precautions as (b) and (c) which are shown in figure 19.17 should be applied on interrupt generation.

  — Priority of canceling software standby mode and watch mode is decided by the setting of ICR.

  — The interrupt signal path and flag setting of the first KCLK interrupt in normal operation differ from those in software standby mode and watch mode. Figure 19.6 shows the interrupt signal paths of the first KCLK interrupt.

  Signal A: Interrupt signal in normal operation

  Signal B: Interrupt signal in software standby mode and watch mode

  — KCLK is input directly to the interrupt control block, not through the PS2, in software standby mode and watch mode, and then an interrupt is generated by detection of a falling edge. Therefore, the KCIF flag is not set. In this case, a flag that is in the interrupt control block is set. The internal flag is automatically cleared after an interrupt request is sent to the CPU. Figure 19.18 shows setting and clearing timing.

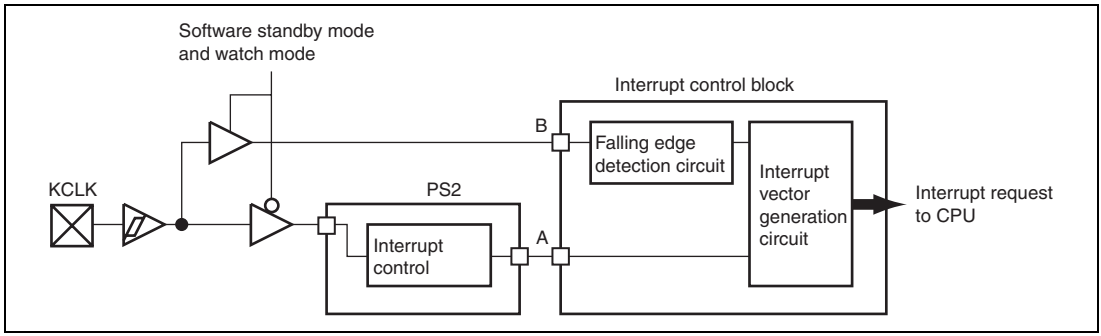


Figure 19.16 First KCLK Interrupt Path

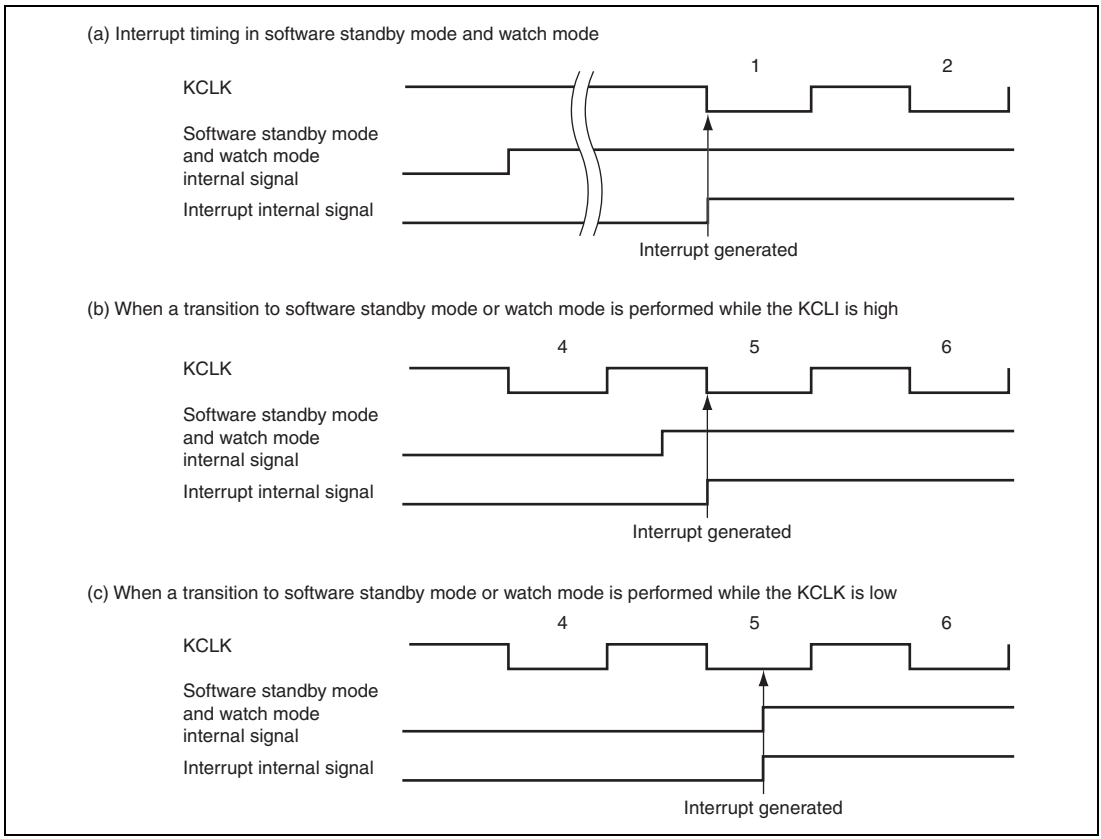


Figure 19.17 Interrupt Timing in Software Standby Mode and Watch Mode

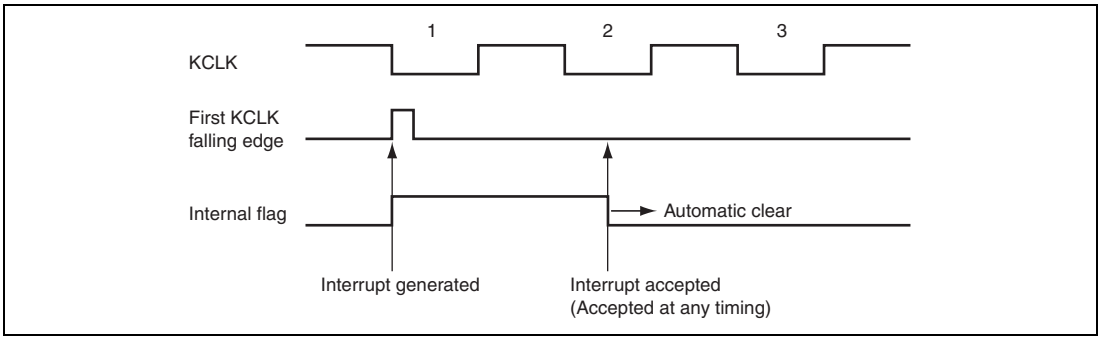


Figure 19.18 Internal Flag of First KCLK Falling Interrupt in Software Standby Mode and Watch Mode

19.5 Usage Notes

19.5.1 KBIOE Setting and KCLK Falling Edge Detection

When KBIOE is 0, the internal KCLK and internal KD settings are fixed at 1. Therefore, if the KCLK pin is low when the KBIOE bit is set to 1, the edge detection circuit operates and the KCLK falling edge is detected.

If the KBFSEL bit and KBE bit are both 0 at this time, the KBF bit is set. Figure 19.19 shows the timing of KBIOE setting and KCLK falling edge detection.

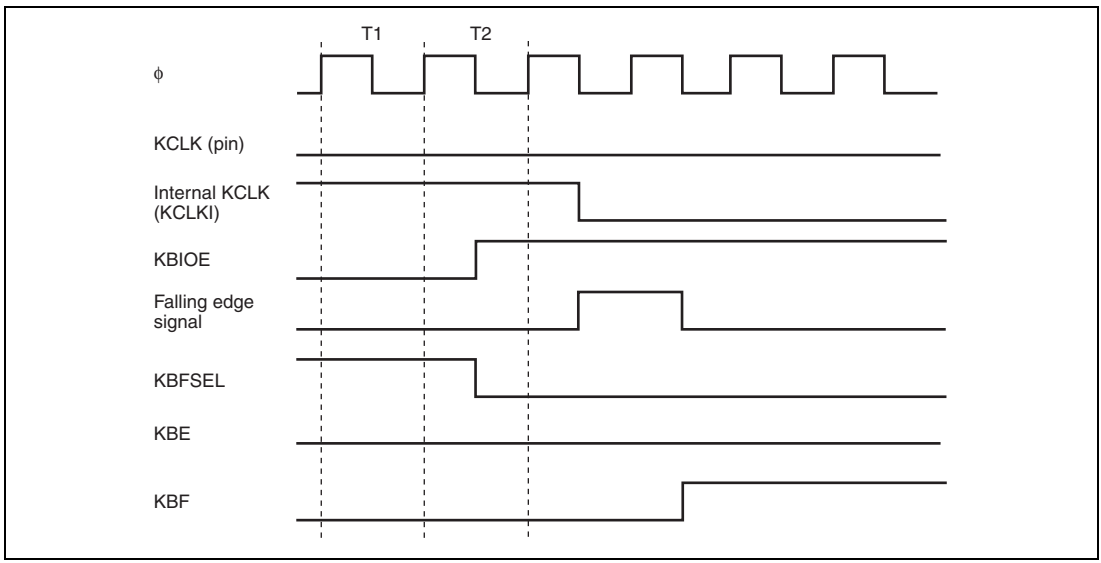


Figure 19.19 KBIOE Setting and KCLK Falling Edge Detection Timing

19.5.2 KD Output by KDO bit (KBCRL) and by Automatic Transmission

Figure 19.20 shows the relationship between the KD output by the KDO bit (KBCRL) and by the automatic transmission. Switch to the KD output by the automatic transmission is performed when KBTS is set to 1 and TXCR is not cleared to 0. In this case, the KD output by the KDO bit (KBCRL) is masked.

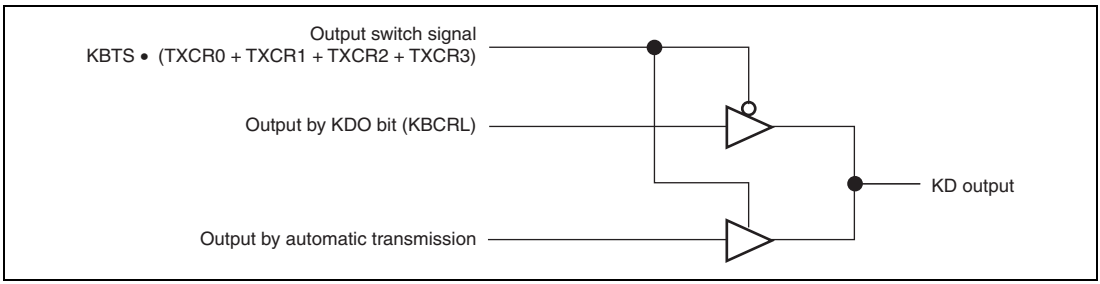


Figure 19.20 KDO Output

19.5.3 Module Stop Mode Setting

Keyboard buffer control unit operation can be enabled or disabled using the module stop control register. The initial setting is for keyboard buffer control unit operation to be halted. Register access is enabled by canceling module stop mode. For details, see section 26, Power-Down Modes.

19.5.4 Medium-Speed Mode

In medium-speed mode, the KBU operates with the medium-speed clock. For normal operation of the KBU, set the medium-speed clock to a frequency of 300 kHz or higher.

19.5.5 Transmit Completion Flag (KBTE)

When TXCR3 to TXCR0 are 1011 (transmit completion notification) and then the TXCR3 to TXCR0 are initialized by clearing KBIOE or KBTS to 0, the transmit completion flag (KBTE) is set. In this case, KTER is invalid.

Section 20 LPC Interface (LPC)

This LSI has an on-chip LPC interface.

The LPC includes four register sets, each of which comprises data and status registers, control register, the fast Gate A20 logic circuit, and the host interrupt request circuit.

The LPC performs serial transfer of cycle type, address, and data, synchronized with the 33 MHz PCI clock. It uses four signal lines for address/data and one for host interrupt requests. This LPC module supports I/O read and I/O write cycle transfers. It is also provided with power-down functions that can control the PCI clock and shut down the LPC interface.

20.1 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 data.
  - Uses three control signals: clock (LCLK), reset ($\overline{\text{LRESET}}$), and frame ($\overline{\text{LFRAME}}$).
- Four register sets comprising data and status registers
  - The basic register set comprises three bytes: an input register (IDR), output register (ODR), and status register (STR).
  - I/O addresses from H'0000 to H'FFFF are selected for channels 1 to 4.
  - A fast Gate A20 function is provided for channel 1.
  - For channel 3, sixteen bidirectional data register bytes can be manipulated in addition to the basic register set.
- Supports SCIF
  - The LPC interface is connected to the SCIF, allowing direct control of the SCIF by the LPC host.
- 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, 3 and 4, SMI, HIRQ6, and HIRQ9 to HIRQ11 can be generated.
  - In the SCIF, HIRQ1, SMI, and HIRQ3 to HIRQ15 can be generated.
  - Operation can be switched between quiet mode and continuous mode.
  - The $\overline{\text{CLKRUN}}$ signal can be manipulated to restart the PCI clock (LCLK).
- Power-down modes and interrupts
  - The LPC module can be shut down by inputting the $\overline{\text{LPCPD}}$ signal.
  - Three pins, $\overline{\text{PME}}$, $\overline{\text{LSMI}}$, and LSCI, are provided for general input/output.

Figure 20.1 shows a block diagram of the LPC.

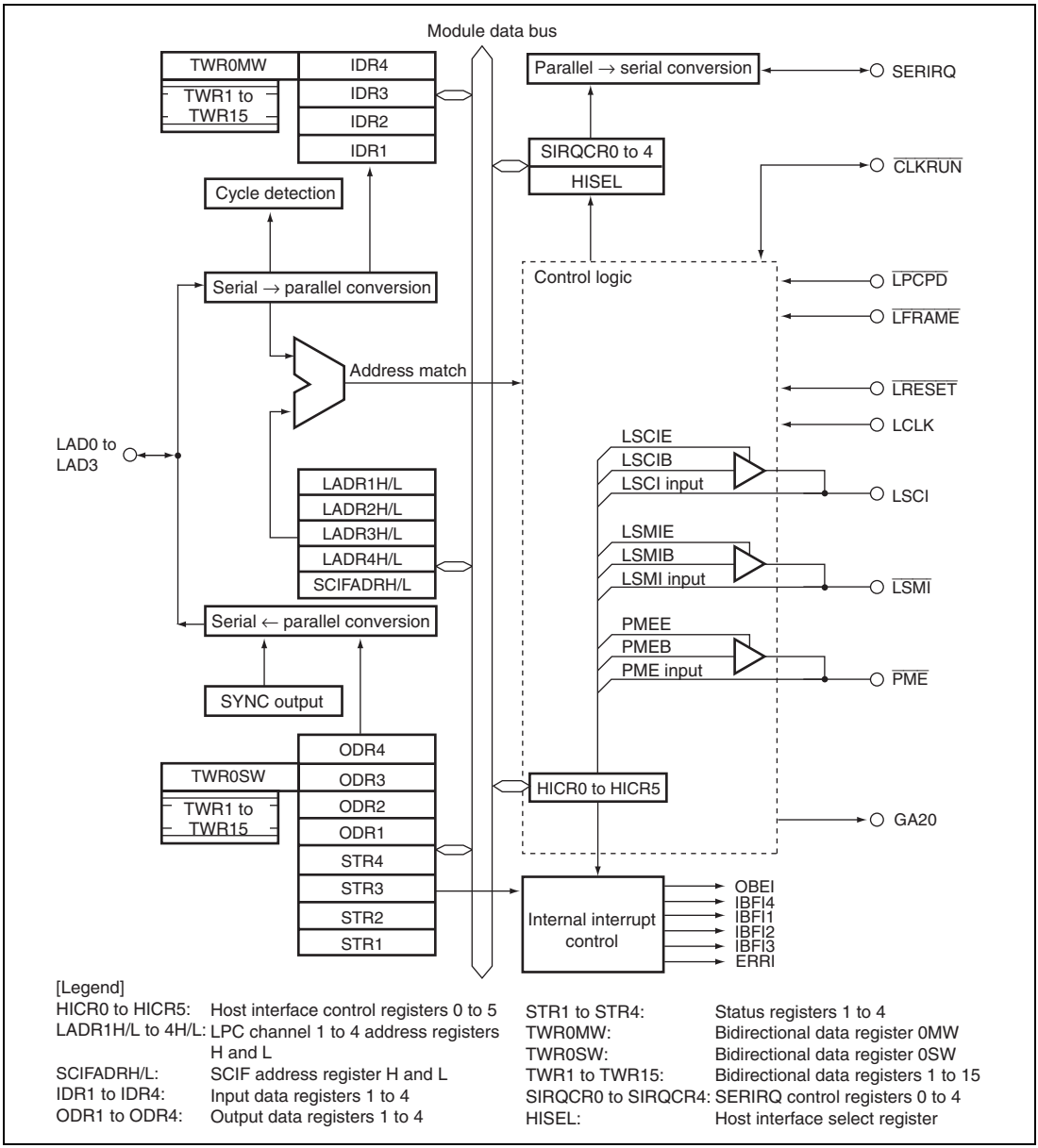


Figure 20.1 Block Diagram of LPC

20.2 Input/Output Pins

Table 20.1 lists the LPC pin configuration.

Table 20.1 Pin Configuration

Name	Abbreviation	Port	I/O	Function
LPC address/ data 3 to 0	LAD3 to LAD0	P33 to P30	I/O	Cycle type/address/data signals serially (4-signal-line) transferred in synchronization with LCLK
LPC frame	$\overline{\text{LFRAME}}$	P34	Input*[1]	Transfer cycle start and forced termination signal
LPC reset	$\overline{\text{LRESET}}$	P35	Input*[1]	LPC interface reset signal
LPC clock	LCLK	P36	Input	33-MHz PCI clock signal
Serialized interrupt request	SERIRQ	P37	I/O*[1]	Serialized host interrupt request signal in synchronization with LCLK
LSCI general output	LSCI	PB1	Output*[1, 2]	General output
LSMI general output	$\overline{\text{LSMI}}$	PB0	Output*[1, 2]	General output
PME general output	$\overline{\text{PME}}$	P80	Output*[1, 2]	General output
GATE A20	GA20	P81	Output*[1, 2]	Gate A20 control signal output
LPC clock run	$\overline{\text{CLKRUN}}$	P82	I/O*[1, 2]	LCLK restart request signal when serial host interrupt is requested
LPC power-down	$\overline{\text{LPCPD}}$	P83	Input*[1]	LPC module shutdown signal

Notes: 1. Pin state monitoring input is possible in addition to the LPC interface control input/output function.

2. Only 0 can be output. If 1 is output, the pin is in the high-impedance state, so an external resistor is necessary to pull the signal up to VCC.

20.3 Register Descriptions

The LPC has the following registers.

Table 20.2 Register Configuration

Register Name	Abbreviation	R/W		Initial Value	Address	Data Bus Width
		Slave	Host			
Host interface control register 0	HICR0	R/W	—	H'00	H'FE40	8
Host interface control register 1	HICR1	R/W	—	H'00	H'FE41	8
Host interface control register 2	HICR2	R/W	—	—	H'FE42	8
Host interface control register 3	HICR3	R	—	—	H'FE43	8
Host interface control register 4	HICR4	R/W	—	H'00	H'FDD9	8
Host interface control register 5	HICR5	R/W	—	H'00	H'FE33	8
LPC channel 1 address register H	LADR1H	R/W	—	H'00	H'FDC0	8
LPC channel 1 address register L	LADR1L	R/W	—	H'60	H'FDC1	8
LPC channel 2 address register H	LADR2H	R/W	—	H'00	H'FDC2	8
LPC channel 2 address register L	LADR2L	R/W	—	H'62	H'FDC3	8
LPC channel 3 address register H	LADR3H	R/W	—	H'00	H'FE34	8
LPC channel 3 address register L	LADR3L	R/W	—	H'00	H'FE35	8
LPC channel 4 address register H	LADR4H	R/W	—	H'00	H'FDD4	8
LPC channel 4 address register L	LADR4L	R/W	—	H'00	H'FDD5	8
Input data register 1	IDR1	R	W	H'00	H'FE38	8
Input data register 2	IDR2	R	W	H'00	H'FE3C	8
Input data register 3	IDR3	R	W	H'00	H'FE30	8
Input data register 4	IDR4	R	W	H'00	H'FDD6	8
Output data register 1	ODR1	R/W	R	H'00	H'FE39	8
Output data register 2	ODR2	R/W	R	H'00	H'FE3D	8
Output data register 3	ODR3	R/W	R	H'00	H'FE31	8
Output data register 4	ODR4	R/W	R	H'00	H'FDD7	8
Status register 1	STR1	R/W	R	H'00	H'FE3A	8
Status register 2	STR2	R/W	R	H'00	H'FE3E	8
Status register 3	STR3	R/W	R	H'00	H'FE32	8
Status register 4	STR4	R/W	R	H'00	H'FDD8	8

Register Name	Abbreviation	R/W		Initial Value	Address	Data Bus Width
		Slave	Host			
Bidirectional data register 0MW	TWR0MW	R/W	R/W	H'00	H'FE20	8
Bidirectional data register 0SW	TWR0SW	R/W	R/W	H'00	H'FE20	8
Bidirectional data register 1	TWR1	R/W	R/W	H'00	H'FE21	8
Bidirectional data register 2	TWR2	R/W	R/W	H'00	H'FE22	8
Bidirectional data register 3	TWR3	R/W	R/W	H'00	H'FE23	8
Bidirectional data register 4	TWR4	R/W	R/W	H'00	H'FE24	8
Bidirectional data register 5	TWR5	R/W	R/W	H'00	H'FE25	8
Bidirectional data register 6	TWR6	R/W	R/W	H'00	H'FE26	8
Bidirectional data register 7	TWR7	R/W	R/W	H'00	H'FE27	8
Bidirectional data register 8	TWR8	R/W	R/W	H'00	H'FE28	8
Bidirectional data register 9	TWR9	R/W	R/W	H'00	H'FE29	8
Bidirectional data register 10	TWR10	R/W	R/W	H'00	H'FE2A	8
Bidirectional data register 11	TWR11	R/W	R/W	H'00	H'FE2B	8
Bidirectional data register 12	TWR12	R/W	R/W	H'00	H'FE2C	8
Bidirectional data register 13	TWR13	R/W	R/W	H'00	H'FE2D	8
Bidirectional data register 14	TWR14	R/W	R/W	H'00	H'FE2E	8
Bidirectional data register 15	TWR15	R/W	R/W	H'00	H'FE2F	8
SERIRQ control register 0	SIRQCR0	R/W	—	H'00	H'FE36	8
SERIRQ control register 1	SIRQCR1	R/W	—	H'00	H'FE37	8
SERIRQ control register 2	SIRQCR2	R/W	—	H'00	H'FDDA	8
SERIRQ control register 3	SIRQCR3	R/W	—	H'00	H'FDDB	8
SERIRQ control register 4	SIRQCR4	R/W	—	H'00	H'FE3B	8
Host interface select register	HISEL	R/W	—	H'03	H'FE3F	8
SCIF address register H	SCIFADRH	R/W	—	H'03	H'FDC4	8
SCIF address register L	SCIFADRL	R/W	—	H'F8	H'FDC5	8

Notes: R/W in the register description means as follows:

1. R/W slave indicates access from the slave (this LSI).
2. R/W host indicates access from the host.

20.3.1 Host Interface Control Registers 0 and 1 (HICR0 and HICR1)

HICR0 and HICR1 contain control bits that enable or disable LPC interface functions, control bits that determine pin output and the internal state of the LPC interface, and status flags that monitor the internal state of the LPC interface.

- HICR0

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
7	LPC3E	0	R/W	—	LPC Enables 3 to 1
6	LPC2E	0	R/W	—	Enable or disable the LPC interface function. When the LPC interface is enabled (one of the three bits is set to 1), processing for data transfer between the slave (this LSI) and the host is performed using pins LAD3 to LAD0, $\overline{\text{LFRAME}}$, $\overline{\text{LRESET}}$, LCLK, SERIRQ, $\overline{\text{CLKRUN}}$, and $\overline{\text{LPCPD}}$. • LPC3E 0: LPC channel 3 operation is disabled No address (LADR3) matches for IDR3, ODR3, STR3, or TWR0 to TWR15 1: LPC channel 3 operation is enabled • LPC2E 0: LPC channel 2 operation is disabled No address (LADR2) matches for IDR2, ODR2, or STR2 1: LPC channel 2 operation is enabled • LPC1E 0: LPC channel 1 operation is disabled No address (LADR1) matches for IDR1, ODR1, or STR1 1: LPC channel 1 operation is enabled
5	LPC1E	0	R/W	—	

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
4	FGA20E	0	R/W	—	Fast Gate A20 Function Enable Enables or disables the fast Gate A20 function. When the fast Gate A20 is disabled, the normal Gate A20 can be implemented by firmware controlling P81 output. 0: Fast Gate A20 function disabled Other function (input/output) of pin P81 is enabled The internal state of GA20 output is initialized to 1 1: Fast Gate A20 function enabled GA20 pin output is open-drain (external pull-up resistor (Vcc) required)
3	SDWNE	0	R/W	—	LPC Software Shutdown Enable Controls LPC interface shutdown. For details of the LPC shutdown function, and the scope of initialization by an LPC reset and an LPC shutdown, see section 20.4.4, LPC Interface Shutdown Function (LPCPD). 0: Normal state, LPC software shutdown setting enabled [Clearing conditions] • Writing 0 • LPC hardware reset or LPC software reset • LPC hardware shutdown release (rising edge of $\overline{\text{LPCPD}}$ signal) 1: LPC hardware shutdown state setting enabled Hardware shutdown state when $\overline{\text{LPCPD}}$ signal is low level [Setting condition] Writing 1 after reading SDWNE = 0

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
2	PMEE	0	R/W	—	PME Output Enable Controls PME output in combination with the PMEB bit in HICR1. $\overline{\text{PME}}$ pin output is open-drain, and an external pull-up resistor (Vcc) is needed. PMEE PMEB 0 X : PME output disabled, other function of pin is enabled 1 0 : PME output enabled, $\overline{\text{PME}}$ pin output goes to 0 level 1 1 : PME output enabled, $\overline{\text{PME}}$ pin output is high-impedance
1	LSMIE	0	R/W	—	LSMI output Enable Controls LSMI output in combination with the LSMIB bit in HICR1. $\overline{\text{LSMI}}$ pin output is open-drain, and an external pull-up resistor (Vcc) is needed. LSMIE LSMIB 0 X : LSMI output disabled, other function of pin is enabled 1 0 : LSMI output enabled, LSMI pin output goes to 0 level 1 1 : LSMI output enabled, LSMI pin output is Hi-Z
0	LSCIE	0	R/W	—	LSCI output Enable Controls LSCI output in combination with the LSCIB bit in HICR1. LSCI pin output is open-drain, and an external pull-up resistor (Vcc) is needed. LSCIE LSCIB 0 X : LSCI output disabled, other function of pin is enabled 1 0 : LSCI output enabled, LSCI pin output goes to 0 level 1 1 : LSCI output enabled, LSCI pin output is high-impedance

[Legend]

X: Don't care

- HICR1

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
7	LPCBSY	0	R	—	LPC Busy Indicates that the LPC interface is processing a transfer cycle. 0: LPC interface is in transfer cycle wait state • Bus idle, or transfer cycle not subject to processing is in progress • Cycle type or address indeterminate during transfer cycle [Clearing conditions] • LPC hardware reset or LPC software reset • LPC hardware shutdown or LPC software shutdown • Forced termination (abort) of transfer cycle subject to processing • Normal termination of transfer cycle subject to processing 1: LPC interface is performing transfer cycle processing [Setting condition] Match of cycle type and address
6	CLKREQ	0	R	—	LCLK Request Indicates that the LPC interface's SERIRQ output is requesting a restart of LCLK. 0: No LCLK restart request [Clearing conditions] • LPC hardware reset or LPC software reset • LPC hardware shutdown or LPC software shutdown • There are no further interrupts for transfer to the host in quiet mode in which SERIRQ is set to continuous mode 1: LCLK restart request issued [Setting condition] In quiet mode, SERIRQ interrupt output becomes necessary while LCLK is stopped

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
5	IRQBSY	0	R	—	SERIRQ Busy Indicates that the LPC interface's SERIRQ is engaged in transfer processing. 0: SERIRQ transfer frame wait state [Clearing conditions] • LPC hardware reset or LPC software reset • LPC hardware shutdown or LPC software shutdown • End of SERIRQ transfer frame 1: SERIRQ transfer processing in progress [Setting condition] Start of SERIRQ transfer frame
4	LRSTB	0	R/W	—	LPC Software Reset Bit Resets the LPC interface. For the scope of initialization by an LPC reset, see section 20.4.4, LPC Interface Shutdown Function (LPCPD). 0: Normal state [Clearing conditions] • Writing 0 • LPC hardware reset 1: LPC software reset state [Setting condition] Writing 1 after reading LRSTB = 0

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
3	SDWNB	0	R/W	—	LPC Software Shutdown Bit Controls LPC interface shutdown. For details of the LPC shutdown function, and the scope of initialization by an LPC reset and an LPC shutdown, see section 20.4.4, LPC Interface Shutdown Function (LPCPD). 0: Normal state [Clearing conditions] • Writing 0 • LPC hardware reset or LPC software reset • LPC hardware shutdown (falling edge of $\overline{\text{LPCPD}}$ signal when SDWNE = 1) • LPC hardware shutdown release (rising edge of $\overline{\text{LPCPD}}$ signal when SDWNE = 0) 1: LPC software shutdown state [Setting condition] Writing 1 after reading SDWNB = 0
2	PMEB	0	R/W	—	PME Output Bit Controls PME output in combination with the PMEE bit. For details, refer to description on the PMEE bit in HICR0.
1	LSMIB	0	R/W	—	LSMI Output Bit Controls LSMI output in combination with the LSMIE bit. For details, refer to description on the LSMIE bit in HICR0.
0	LSCIB	0	R/W	—	LSCI output Bit Controls LSCI output in combination with the LSCIE bit IN HICR0. For details, refer to description on the LSCIE bit in HICR0.

20.3.2 Host Interface Control Registers 2 and 3 (HICR2 and HICR3)

HICR2 controls interrupts to an LPC interface slave (this LSI). HICR3 and the bit 7 in HICR2 monitor the states of the LPC interface pins. Bits 6 to 0 in HICR2 are initialized to H'00 by a reset. The states of other bits are decided by the pin states. The pin states can be monitored by the pin monitoring bits regardless of the LPC interface operating state or the operating state of the functions that use pin multiplexing.

- HICR2

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
7	GA20	Undefined	R	—	GA20 Pin Monitor
6	LRST	0	R/(W)*	—	LPC Reset Interrupt Flag This bit is a flag that generates an ERRI interrupt when an LPC hardware reset occurs. 0: [Clearing condition] Writing 0 after reading LRST = 1 1: [Setting condition] $\overline{\text{LRESET}}$ pin falling edge detection
5	SDWN	0	R/(W)*	—	LPC Shutdown Interrupt Flag This bit is a flag that generates an ERRI interrupt when an LPC hardware shutdown request is generated. 0: [Clearing conditions] • Writing 0 after reading SDWN = 1 • LPC hardware reset ($\overline{\text{LRESET}}$ pin falling edge detection) • LPC software reset (LRSTB = 1) 1: [Setting condition] $\overline{\text{LPCPD}}$ pin falling edge detection

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
4	ABRT	0	R/(W)*	—	LPC Abort Interrupt Flag This bit is a flag that generates an ERRI interrupt when a forced termination (abort) of an LPC transfer 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 hardware shutdown (SDWNE = 1 and $\overline{\text{LPCPD}}$ pin falling edge detection) • LPC software shutdown (SDWNB = 1) 1: [Setting condition] $\overline{\text{LFRAME}}$ pin falling edge detection during LPC transfer cycle
3	IBFIE3	0	R/W	—	IDR3 and TWR Receive Complete interrupt Enable Enables or disables IBFI3 interrupt to the slave (this LSI). 0: Input data register IDR3 and TWR receive complete interrupt requests disabled 1: [When TWRIE = 0 in LADR3] Input data register (IDR3) receive complete interrupt requests enabled [When TWRIE = 1 in LADR3] Input data register (IDR3) and TWR receive complete interrupt requests enabled
2	IBFIE2	0	R/W	—	IDR2 Receive Complete interrupt Enable Enables or disables IBFI2 interrupt to the slave (this LSI). 0: Input data register (IDR2) receive complete interrupt requests disabled 1: Input data register (IDR2) receive complete interrupt requests enabled

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
1	IBFIE1	0	R/W	—	IDR1 Receive Complete interrupt Enable Enables or disables IBFI1 interrupt to the slave (this LSI). 0: Input data register (IDR1) receive complete interrupt requests disabled 1: Input data register (IDR1) receive complete interrupt requests enabled
0	ERRIE	0	R/W	—	Error Interrupt Enable Enables or disables ERRI interrupt to the slave (this LSI). 0: Error interrupt requests disabled 1: Error interrupt requests enabled

Note: * Only 0 can be written to bits 6 to 4, to clear the flag.

- HICR3

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
7	LFRAME	Undefined	R	—	$\overline{\text{LFRAME}}$ Pin Monitor
6	CLKRUN	Undefined	R	—	$\overline{\text{CLKRUN}}$ Pin Monitor
5	SERIRQ	Undefined	R	—	SERIRQ Pin Monitor
4	LRESET	Undefined	R	—	$\overline{\text{LRESET}}$ Pin Monitor
3	LPCPD	Undefined	R	—	$\overline{\text{LPCPD}}$ Pin Monitor
2	PME	Undefined	R	—	$\overline{\text{PME}}$ Pin Monitor
1	LSMI	Undefined	R	—	$\overline{\text{LSMI}}$ Pin Monitor
0	LSCI	Undefined	R	—	LSCI Pin Monitor

20.3.3 Host Interface Control Register 4 (HICR4)

HICR4 enables/disables channel 4 and controls interrupts to the channel 4 of an LPC interface slave (this LSI).

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
7	—	0	R/W	—	Reserved The initial value bit should not be changed.
6	LPC4E	0	R/W	—	LPC Enable 4 0: LPC channel 4 is disabled For IDR4, ODR4, and STR4, address (LADR4) match is not occurred. 1: LPC channel 4 enabled
5	IBFIE4	0	R/W	—	IDR4 Receive Completion Interrupt Enable Enables or disables IBFI4 interrupt to the slave (this LSI). 0: Input data register (IDR4) receive complete interrupt requests disabled 1: Input data register (IDR4) receive complete interrupt requests enabled
4 to 0	—	All 0	R/W	—	Reserved The initial value should not be changed.

20.3.4 Host Interface Control Register 5 (HICR5)

HICR5 enables or disables the operation of the SCIF interface, and controls OBEI interrupts.

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
7	OBEIE	0	R/W	—	Output Buffer Empty Interrupt Enable Enables or disables OBEI interrupts (for this LSI). 0: Output buffer empty interrupt request is disabled 1: Output buffer empty interrupt request is enabled
6	OBEI	0	R/W	—	Output Buffer Empty Interrupt Flag 0: [Clearing conditions] • Writing 0 after reading OBEI = 1 • LPC hardware reset or LPC software reset 1: [Setting condition] When one of OBF1, OBF2, OBF3A, OBF3B, and OBF4 is cleared
5 to 4	—	All 0	R/W	—	Reserved The initial value bit should not be changed.
3	SCIFE	0	R/W	—	SCIF Enable Enables or disables access from the LPC host of the SCIF. 0: Disables access from the LPC host of the SCIF 1: Enables access from the LPC host of the SCIF
2 to 0	—	All 0	R/W	—	Reserved The initial value should not be changed.

20.3.5 LPC Channel 1 Address Registers H and L (LADR1H and LADR1L)

LADR1 sets the LPC channel 1 host address. The LADR1 contents must not be changed while channel 1 is operating (while LPC1E is set to 1).

- LADR1H

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
7	Bit 15	0	R/W	—	Channel 1 Address Bits 15 to 8
6	Bit 14	0	R/W	—	Set the LPC channel 1 host address.
5	Bit 13	0	R/W	—	
4	Bit 12	0	R/W	—	
3	Bit 11	0	R/W	—	
2	Bit 10	0	R/W	—	
1	Bit 9	0	R/W	—	
0	Bit 8	0	R/W	—	

- LADR1L

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
7	Bit 7	0	R/W	—	Channel 1 Address Bits 7 to 3
6	Bit 6	1	R/W	—	Set the LPC channel 1 host address.
5	Bit 5	1	R/W	—	
4	Bit 4	0	R/W	—	
3	Bit 3	0	R/W	—	
2	Bit 2	0	R/W	—	Reserved This bit is ignored when an address match is decided.
1	Bit 1	0	R/W	—	Channel 1 Address Bits 1 and 0
0	Bit 0	0	R/W	—	Set the LPC channel 1 host address.

- Host select register

I/O Address			Transfer	
Bits 5 to 3	**Bit 2**	**Bits 1 and 0**	**Cycle**	**Host Select Register**
Bits 15 to 3 in LADR1	0	Bits 1 and 0 in LADR1	I/O write	IDR1 write (data)
Bits 15 to 3 in LADR1	1	Bits 1 and 0 in LADR1	I/O write	IDR1 write (command)
Bits 15 to 3 in LADR1	0	Bits 1 and 0 in LADR1	I/O read	ODR1 read
Bits 15 to 3 in LADR1	1	Bits 1 and 0 in LADR1	I/O read	STR1 read

Note: * When channel 1 is used, the content of LADR1 must be set so that the addresses for channels 2, 3, 4, and SCIF are different.

20.3.6 LPC Channel 2 Address Registers H and L (LADR2H and LADR2L)

LADR2 sets the LPC channel 2 host address. The LADR2 contents must not be changed while channel 2 is operating (while LPC2E is set to 1).

- LADR2H

		Initial	R/W		
Bit	**Bit Name**	**Value**	**Slave**	**Host**	**Description**
7	Bit 15	0	R/W	—	Channel 2 Address Bits 15 to 8
6	Bit 14	0	R/W	—	Set the LPC channel 2 host address.
5	Bit 13	0	R/W	—	
4	Bit 12	0	R/W	—	
3	Bit 11	0	R/W	—	
2	Bit 10	0	R/W	—	
1	Bit 9	0	R/W	—	
0	Bit 8	0	R/W	—	

- LADR2L

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
7	Bit 7	0	R/W	—	Channel 2 Address Bits 7 to 3
6	Bit 6	1	R/W	—	Set the LPC channel 2 host address.
5	Bit 5	1	R/W	—	
4	Bit 4	0	R/W	—	
3	Bit 3	0	R/W	—	
2	Bit 2	0	R/W	—	Reserved This bit is ignored when an address match is decided.
1	Bit 1	1	R/W	—	Channel 2 Address Bits 1 and 0
0	Bit 0	0	R/W	—	Set the LPC channel 2 host address.

- Host select register

I/O Address Bits 5 to 3	I/O Address Bit 2	I/O Address Bits 1 and 0	Transfer Cycle	Host Select Register
Bits 15 to 3 in LADR2	0	Bits 1 and 0 in LADR2	I/O write	IDR2 write (data)
Bits 15 to 3 in LADR2	1	Bits 1 and 0 in LADR2	I/O write	IDR2 write (command)
Bits 15 to 3 in LADR2	0	Bits 1 and 0 in LADR2	I/O read	ODR2 read
Bits 15 to 3 in LADR2	1	Bits 1 and 0 in LADR2	I/O read	STR2 read

Note: * When channel 2 is used, the content of LADR2 must be set so that the addresses for channels 1, 3, 4, and SCIF are different.

20.3.7 LPC Channel 3 Address Registers H and L (LADR3H and LADR3L)

LADR3 sets the LPC channel 3 host address and controls the operation of the bidirectional data registers. The contents of the address fields in LADR3 must not be changed while channel 3 is operating (while LPC3E is set to 1).

- LADR3H

			R/W		
Bit	**Bit Name**	**Initial Value**	**Slave**	**Host**	**Description**
7	Bit 15	0	R/W	—	Channel 3 Address Bits 15 to 8
6	Bit 14	0	R/W	—	Set the LPC channel 3 host address.
5	Bit 13	0	R/W	—	
4	Bit 12	0	R/W	—	
3	Bit 11	0	R/W	—	
2	Bit 10	0	R/W	—	
1	Bit 9	0	R/W	—	
0	Bit 8	0	R/W	—	

- LADR3L

			R/W		
Bit	**Bit Name**	**Initial Value**	**Slave**	**Host**	**Description**
7	Bit 7	0	R/W	—	Channel 3 Address Bits 7 to 3
6	Bit 6	0	R/W	—	Set the LPC channel 3 host address.
5	Bit 5	0	R/W	—	
4	Bit 4	0	R/W	—	
3	Bit 3	0	R/W	—	
2	—	0	R/W	—	Reserved The initial value should not be changed.
1	Bit 1	0	R/W	—	Channel 3 Address Bit 1 Sets the LPC channel 3 host address.
0	TWRE	0	R/W	—	Bidirectional Data Register Enable Enables or disables bidirectional data register operation. 0: TWR operation is disabled TWR-related I/O address match determination is halted 1: TWR operation is enabled

When LPC3E = 1, an I/O address received in an LPC I/O cycle is compared with the contents of LADR3. When determining an IDR3, ODR3, or STR3 address match, bit 0 in LADR3 is regarded as 0, and the value of bit 2 is ignored. When determining a TWR0 to TWR15 address match, bit 4 in LADR3 is inverted, and the values of bits 3 to 0 are ignored.

- Host select register

I/O Address					Transfer	
Bit 4	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**	**Cycle**	**Host Select Register**
Bit 4	Bit 3	0	Bit 1	0	I/O write	IDR3 write, C/$\overline{D}$3 ← 0
Bit 4	Bit 3	1	Bit 1	0	I/O write	IDR3 write, C/$\overline{D}$3 ← 1
Bit 4	Bit 3	0	Bit 1	0	I/O read	ODR3 read
Bit 4	Bit 3	1	Bit 1	0	I/O read	STR3 read
$\overline{\text{Bit 4}}$	0	0	0	0	I/O write	TWR0MW write
$\overline{\text{Bit 4}}$	0	0	0	1	I/O write	TWR1 to TWR15 write
	:	:	:	:		
	1	1	1	1		
$\overline{\text{Bit 4}}$	0	0	0	0	I/O read	TWR0SW read
$\overline{\text{Bit 4}}$	0	0	0	1	I/O read	TWR1 to TWR15 read
	:	:	:	:		
	1	1	1	1		

Note: * When channel 3 is used, the content of LADR3 must be set so that the addresses for channels 1, 2, 4, and SCIF are different.

20.3.8 LPC Channel 4 Address Registers H and L (LADR4H and LADR4L)

LADR4 sets the LPC channel 4 host address. The LADR4 contents must not be changed while channel 4 is operating (while LPC4E is set to 1).

- LADR4H

			R/W		
Bit	**Bit Name**	**Initial Value**	**Slave**	**Host**	**Description**
7	Bit 15	0	R/W	—	Channel 4 Address Bits 15 to 8
6	Bit 14	0	R/W	—	Set the LPC channel 4 host address.
5	Bit 13	0	R/W	—	
4	Bit 12	0	R/W	—	
3	Bit 11	0	R/W	—	
2	Bit 10	0	R/W	—	
1	Bit 9	0	R/W	—	
0	Bit 8	0	R/W	—	

- LADR4L

			R/W		
Bit	**Bit Name**	**Initial Value**	**Slave**	**Host**	**Description**
7	Bit 7	0	R/W	—	Channel 4 Address Bits 7 to 3
6	Bit 6	0	R/W	—	Set the LPC channel 4 host address.
5	Bit 5	0	R/W	—	
4	Bit 4	0	R/W	—	
3	Bit 3	0	R/W	—	
2	Bit2	0	R/W	—	Reserved This bit is ignored when an address match is decided.
1	Bit 1	0	R/W	—	Channel 4 Address Bits 1 and 0
0	Bit 0	0	R/W	—	Set the LPC channel 4 host address.

- Host select register

I/O Address			Transfer	
Bits 5 to 3	**Bit 2**	**Bits 1 and 0**	**Cycle**	**Host Select Register**
Bits 15 to 3 in LADR4	0	Bits 1 and 0 in LADR4	I/O write	IDR4 write (data)
Bits 15 to 3 in LADR4	1	Bits 1 and 0 in LADR4	I/O write	IDR4 write (command)
Bits 15 to 3 in LADR4	0	Bits 1 and 0 in LADR4	I/O read	ODR4 read
Bits 15 to 3 in LADR4	1	Bits 1 and 0 in LADR4	I/O read	STR4 read

Note: * When channel 4 is used, the content of LADR4 must be set so that the addresses for channels 1, 2, 3 and SCIF are different.

20.3.9 Input Data Registers 1 to 4 (IDR1 to IDR4)

IDR1 to IDR4 are 8-bit read-only registers for the slave (this LSI), and 8-bit write-only registers for the host. The registers selected from the host according to the I/O address are shown in the following table. Data transferred in an LPC I/O write cycle is written to the selected register. The value of bit 2 of the I/O address is latched into the C/$\overline{D}$ bit in STR, to indicate whether the written information is a command or data. The initial values of IDR1 to IDR4 are H'00.

I/O Address					Transfer	
Bits 15 to 4	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**	**Cycle**	**Host Register Selection**
Bits 15 to 4	Bit 3	0	Bit 1	Bit 0	I/O write	IDRn write, C/$\overline{D}$n ← 0
Bits 15 to 4	Bit 3	1	Bit 1	Bit 0	I/O write	IDRn write, C/$\overline{D}$n ← 1

n = 1 to 4

20.3.10 Output Data Registers 1 to 4 (ODR1 to ODR4)

ODR1 to ODR4 are 8-bit readable/writable registers for the slave (this LSI), and 8-bit read-only registers for the host. The registers selected from the host according to the I/O address are shown in the following table. In an LPC I/O read cycle, the data in the selected register is transferred to the host. The initial values of ODR1 to ODR4 are H'00.

I/O Address					Transfer	
Bits 15 to 4	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**	**Cycle**	**Host Register Selection**
Bits 15 to 4	Bit 3	0	Bit1	Bit 0	I/O read	ODRn read

n = 1 to 4

20.3.11 Bidirectional Data Registers 0 to 15 (TWR0 to TWR15)

TWR0 to TWR15 are sixteen 8-bit readable/writable registers to both the slave (this LSI) and host. In TWR0, however, two registers (TWR0MW and TWR0SW) are allocated to the same address for both the host and the slave addresses. TWR0MW is a write-only register for the host, and a read-only register for the slave, while TWR0SW is a write-only register for the slave and a read-only register for the host. When the host and slave begin a write, after the respective registers of TWR0 have been written to, arbitration for simultaneous access is performed by checking the status flags whether or not those writes were valid. For the registers selected from the host according to the I/O address, see section 20.3.7, LPC Channel 3 Address Registers H and L (LADR3H and LADR3L).

Data transferred in an LPC I/O write cycle is written to the selected register; in an LPC I/O read cycle, the data in the selected register is transferred to the host. The initial values of TWR0 to TWR15 are H'00.

20.3.12 Status Registers 1 to 4 (STR1 to STR4)

STR1 to STR4 are 8-bit registers that indicate status information during LPC interface processing. The registers selected from the host according to the I/O address are shown in the following table. In an LPC I/O read cycle, the data in the selected register is transferred to the host.

I/O Address					Transfer	
Bits 15 to 4	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**	**Cycle**	**Host Register Selection**
Bits 15 to 4	Bit 3	1	Bit1	Bit 0	I/O read	STRn read

n = 1 to 4

- STR1

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
7	DBU17	0	R/W	R	Defined by User
6	DBU16	0	R/W	R	The user can use these bits as necessary.
5	DBU15	0	R/W	R	
4	DBU14	0	R/W	R	
3	C/$\overline{D}$1	0	R	R	Command/Data When the host writes to IDR1, bit 2 of the I/O address is written into this bit to indicate whether IDR1 contains data or a command. 0: Content of input data register (IDR1) is a data 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.
1	IBF1	0	R	R	Input Buffer Full This bit is an internal interrupt source to the slave (this LSI). The IBF1 flag setting and clearing conditions are different when the fast Gate A20 is used. For details, see table 20.5. 0: [Clearing condition] When the slave reads IDR1 1: [Setting condition] When the host writes to IDR1 in I/O write cycle
0	OBF1	0	R/(W)*	R	Output Buffer Full 0: [Clearing conditions] • When the host reads ODR1 in I/O read cycle • When the slave writes 0 to the OBF1 bit 1: [Setting condition] When the slave writes to ODR1

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

- STR2

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
7	DBU27	0	R/W	R	Defined by User
6	DBU26	0	R/W	R	The user can use these bits as necessary.
5	DBU25	0	R/W	R	
4	DBU24	0	R/W	R	
3	C/$\overline{D}$2	0	R	R	Command/Data When the host writes to IDR2, bit 2 of the I/O address is written into this bit to indicate whether IDR2 contains data or a command. 0: Content of input data register (IDR2) is a data 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 Buffer Full This bit is an internal interrupt source to the slave (this LSI). 0: [Clearing condition] When the slave reads IDR2 1: [Setting condition] When the host writes to IDR2 in I/O write cycle
0	OBF2	0	R/(W)*	R	Output Buffer Full 0: [Clearing conditions] • When the host reads ODR2 in I/O read cycle • When the slave writes 0 to the OBF2 bit 1: [Setting condition] When the slave writes to ODR2

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

- STR3 (TWRE = 1 or SELSTR3 = 0)

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
7	IBF3B	0	R	R	Bidirectional Data Register Input Buffer Full Flag This is an internal interrupt source to the slave (this LSI). 0: [Clearing condition] When the slave reads TWR15 1: [Setting condition] When the host writes to TWR15 in I/O write cycle
6	OBF3B	0	R/(W)*	R	Bidirectional Data Register Output Buffer Full Flag 0: [Clearing conditions] • When the host reads TWR15 in I/O read cycle • When the slave writes 0 to the OBF3B bit 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 cycle while SWMF = 0
4	SWMF	0	R/(W)*	R	Slave Write Mode Flag In the event of simultaneous writes by the master and the slave, the master write has priority. 0: [Clearing conditions] • When the host reads TWR15 in I/O read cycle • When the slave writes 0 to the SWMF bit 1: [Setting condition] When the slave writes to TWR0 while MWMF = 0

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
3	C/$\overline{D}$3	0	R	R	Command/Data Flag When the host writes to IDR3, bit 2 of the I/O address is written into this bit to indicate whether IDR3 contains data or a command. 0: Content of input data register (IDR3) is a data 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 Buffer Full This bit is an internal interrupt source to the slave (this LSI). 0: [Clearing condition] When the slave reads IDR3 1: [Setting condition] When the host writes to IDR3 in I/O write cycle
0	OBF3A	0	R/(W)*	R	Output Buffer Full 0: [Clearing conditions] • When the host reads ODR3 in I/O read cycle • When the slave writes 0 to the OBF3 bit 1: [Setting condition] When the slave writes to ODR3

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

- STR3 (TWRE = 0 and SELSTR3 = 1)

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
7	DBU37	0	R/W	R	Defined by User
6	DBU36	0	R/W	R	The user can use these bits as necessary.
5	DBU35	0	R/W	R	
4	DBU34	0	R/W	R	
3	C/$\overline{D}$3	0	R	R	Command/Data Flag When the host writes to IDR3, bit 2 of the I/O address is written into this bit to indicate whether IDR3 contains data or a command. 0: Content of input data register (IDR3) is a data 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	IBF3	0	R	R	Input Buffer Full This bit is an internal interrupt source to the slave (this LSI). 0: [Clearing condition] When the slave reads IDR3 1: [Setting condition] When the host writes to IDR3 in I/O write cycle
0	OBF3	0	R/(W)*	R	Output Buffer Full 0: [Clearing conditions] • When the host reads ODR3 in I/O read cycle • When the slave writes 0 to the OBF3 bit 1: [Setting condition] When the slave writes to ODR3

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

- STR4

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
7	DBU47	0	R/W	R	Defined by User
6	DBU46	0	R/W	R	The user can use these bits as necessary.
5	DBU45	0	R/W	R	
4	DBU44	0	R/W	R	
3	C/$\overline{D}$4	0	R	R	Command/Data Flag When the host writes to IDR4, bit 2 of the I/O address is written into this bit to indicate whether IDR4 contains data or a command. 0: Content of input data register (IDR4) is a data 1: Content of input data register (IDR4) is a command
2	DBU42	0	R/W	R	Defined by User The user can use this bit as necessary.
1	IBF4	0	R	R	Input Buffer Full This bit is an internal interrupt source to the slave (this LSI). 0: [Clearing condition] When the slave reads IDR4 1: [Setting condition] When the host writes to IDR4 in I/O write cycle
0	OBF4	0	R/(W)*	R	Output Buffer Full 0: [Clearing conditions] • When the host reads ODR4 in I/O read cycle • When the slave writes 0 to the OBF3 bit 1: [Setting condition] When the slave writes to ODR4

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

20.3.13 SERIRQ Control Register 0 (SIRQCR0)

SIRQCR0 contains status bits that indicate the SERIRQ operating mode and bits that specify SERIRQ interrupt sources.

			R/W		
Bit	**Bit Name**	**Initial Value**	**Slave**	**Host**	**Description**
7	Q/$\overline{C}$	0	R	—	Quiet/Continuous Mode Flag Indicates the mode specified by the host at the end of an SERIRQ transfer cycle (stop frame). 0: Continuous mode [Clearing conditions] • LPC hardware reset, LPC software reset • Specification by SERIRQ transfer cycle stop frame 1: Quiet mode [Setting condition] Specification by SERIRQ transfer cycle stop frame.
6	SELREQ	0	R/W	—	Start Frame Initiation Request Select Selects the condition of a start frame initiation request when a host interrupt request is cleared in quiet mode. 0: Start frame initiation is requested when all interrupt requests are cleared 1: Start frame initiation is requested when one or more interrupt requests are cleared
5	IEDIR2	0	R/W	—	Interrupt Enable Direct Mode 2 Selects whether an SERIRQ interrupt generation of LPC channel 2 is affected only by a host interrupt enable bit or by an OBF flag in addition to the enable bit. 0: A host interrupt is generated when both the enable bit and the corresponding OBF flag are set 1: A host interrupt is generated when the enable bit is set

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
4	SMIE3B	0	R/W	—	Host SMI Interrupt Enable 3B Enables or disables an SMI interrupt request when OBF3B is set by a TWR15 write. 0: Host SMI interrupt request by OBF3B and SMIE3B is disabled [Clearing conditions] • Writing 0 to SMIE3B • LPC hardware reset, LPC software reset • Clearing OBF3B to 0 (when IEDIR3 = 0) 1: [When IEDIR3 = 0] Host SMI interrupt request by setting OBF3B to 1 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 request when OBF3A is set by an ODR3 write. 0: Host SMI interrupt request by OBF3A and SMIE3A is disabled [Clearing conditions] • Writing 0 to SMIE3A • LPC hardware reset, LPC software reset • Clearing OBF3A to 0 (when IEDIR3 = 0) 1: [When IEDIR3 = 0] Host SMI interrupt request by setting is enabled [When IEDIR3 = 1] Host SMI interrupt is requested [Setting condition] Writing 1 after reading SMIE3A = 0

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
2	SMIE2	0	R/W	—	Host SMI Interrupt Enable 2 Enables or disables an SMI interrupt request when OBF2 is set by an ODR2 write. 0: Host SMI interrupt request by OBF2 and SMIE2 is disabled [Clearing conditions] • Writing 0 to SMIE2 • LPC hardware reset, LPC software reset • Clearing OBF2 to 0 (when IEDIR2 = 0) 1: [When IEDIR2 = 0] Host SMI interrupt request by setting OBF2 to 1 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 request when OBF1 is set by an ODR1 write. 0: HIRQ12 interrupt request by OBF1 and IRQ12E1 is disabled [Clearing conditions] • Writing 0 to IRQ12E1 • LPC hardware reset, LPC software reset • Clearing OBF1 to 0 1: HIRQ12 interrupt request by setting OBF1 to 1 is enabled [Setting condition] Writing 1 after reading IRQ12E1 = 0

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
0	IRQ1E1	0	R/W	—	Host IRQ1 Interrupt Enable 1 Enables or disables a host HIRQ1 interrupt request when OBF1 is set by an ODR1 write. 0: HIRQ1 interrupt request by OBF1 and IRQ1E1 is disabled [Clearing conditions] • Writing 0 to IRQ1E1 • LPC hardware reset, LPC software reset • Clearing OBF1 to 0 1: HIRQ1 interrupt request by setting OBF1 to 1 is enabled [Setting condition] Writing 1 after reading IRQ1E1 = 0

20.3.14 SERIRQ Control Register 1 (SIRQCR1)

SIRQCR1 contains status bits that indicate the SERIRQ operating mode and bits that specify SERIRQ interrupt sources.

			R/W		
Bit	**Bit Name**	**Initial Value**	**Slave**	**Host**	**Description**
7	IRQ11E3	0	R/W	—	Host IRQ11 Interrupt Enable 3 Enables or disables an HIRQ11 interrupt request when OBF3A is set by an ODR3 write. 0: HIRQ11 interrupt request by OBF3A and IRQE11E3 is disabled [Clearing conditions] • Writing 0 to IRQ11E3 • LPC hardware reset, LPC software reset • Clearing OBF3A to 0 (when IEDIR3 = 0) 1: [When IEDIR3 = 0] HIRQ11 interrupt request by setting OBF3A to 1 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 request when OBF3A is set by an ODR3 write. 0: HIRQ10 interrupt request by OBF3A and IRQE10E3 is disabled [Clearing conditions] • Writing 0 to IRQ10E3 • LPC hardware reset, LPC software reset • Clearing OBF3A to 0 (when IEDIR3 = 0) 1: [When IEDIR3 = 0] HIRQ10 interrupt request by setting OBF3A to 1 is enabled [When IEDIR3 = 1] HIRQ10 interrupt is requested [Setting condition] Writing 1 after reading IRQ10E3 = 0

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
5	IRQ9E3	0	R/W	—	Host IRQ9 Interrupt Enable 3 Enables or disables an HIRQ9 interrupt request when OBF3A is set by an ODR3 write. 0: HIRQ9 interrupt request by OBF3A and IRQE9E3 is disabled [Clearing conditions] • Writing 0 to IRQ9E3 • LPC hardware reset, LPC software reset • Clearing OBF3A to 0 (when IEDIR3 = 0) 1: [When IEDIR3 = 0] HIRQ9 interrupt request by setting OBF3A to 1 is 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 request when OBF3A is set by an ODR3 write. 0: HIRQ6 interrupt request by OBF3A and IRQE6E3 is disabled [Clearing conditions] • Writing 0 to IRQ6E3 • LPC hardware reset, LPC software reset • Clearing OBF3A to 0 (when IEDIR3 = 0) 1: [When IEDIR3 = 0] HIRQ6 interrupt request by setting OBF3A to 1 is enabled [When IEDIR3 = 1] HIRQ6 interrupt is requested [Setting condition] Writing 1 after reading IRQ6E3 = 0

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
3	IRQ11E2	0	R/W	—	Host IRQ11 Interrupt Enable 2 Enables or disables an HIRQ11 interrupt request when OBF2 is set by an ODR2 write. 0: HIRQ11 interrupt request by OBF2 and IRQE11E2 is disabled [Clearing conditions] • Writing 0 to IRQ11E2 • LPC hardware reset, LPC software reset • Clearing OBF2 to 0 (when IEDIR2 = 0) 1: [When IEDIR2 = 0] HIRQ11 interrupt request by setting OBF2 to 1 is 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 request 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 OBF2 to 1 is enabled [When IEDIR2 = 1] HIRQ10 interrupt is requested [Setting condition] Writing 1 after reading IRQ10E2 = 0

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
1	IRQ9E2	0	R/W	—	Host IRQ9 Interrupt Enable 2 Enables or disables an HIRQ9 interrupt request when OBF2 is set by an ODR2 write. 0: HIRQ9 interrupt request by OBF2 and IRQE9E2 is disabled [Clearing conditions] • Writing 0 to IRQ9E2 • LPC hardware reset, LPC software reset • Clearing OBF2 to 0 (when IEDIR2 = 0) 1: [When IEDIR2 = 0] HIRQ9 interrupt request by setting OBF2 to 1 is 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 request when OBF2 is set by an ODR2 write. 0: HIRQ6 interrupt request by OBF2 and IRQE6E2 is disabled [Clearing conditions] • Writing 0 to IRQ6E2 • LPC hardware reset, LPC software reset • Clearing OBF2 to 0 (when IEDIR2 = 0) 1: [When IEDIR2 = 0] HIRQ6 interrupt request by setting OBF2 to 1 is enabled [When IEDIR2 = 1] HIRQ6 interrupt is requested [Setting condition] Writing 1 after reading IRQ6E2 = 0

20.3.15 SERIRQ Control Register 2 (SIRQCR2)

SIRQCR2 contains bits that enable or disable SERIRQ interrupt requests and select the host interrupt request outputs.

			R/W		
Bit	**Bit Name**	**Initial Value**	**Slave**	**Host**	**Description**
7	IEDIR3	0	R/W	—	Interrupt Enable Direct Mode 3 Selects whether an SERIRQ interrupt generation of LPC channel 3 is affected only by a host interrupt enable bit or by an OBF flag in addition to the enable bit. 0: A host interrupt is generated when both the enable bit and the corresponding OBF flag are set 1: A host interrupt is generated when the enable bit is set
6	IEDIR4	0	R/W	—	Interrupt Enable Direct Mode 4 Selects whether an SERIRQ interrupt generation of LPC channel 4 is affected only by a host interrupt enable bit or by an OBF flag in addition to the enable bit. 0: A host interrupt is generated when both the enable bit and the corresponding OBF flag are set 1: A host interrupt is generated when the enable bit is set

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
5	IRQ11E4	0	R/W	—	Host IRQ11 Interrupt Enable 4 Enables or disables an HIRQ11 interrupt request when OBF4 is set by an ODR4 write. 0: HIRQ11 interrupt request by OBF4 and IRQE11E4 is disabled [Clearing conditions] • Writing 0 to IRQ11E4 • LPC hardware reset, LPC software reset • Clearing OBF4 to 0 (when IEDIR4 = 0) 1: [When IEDIR4 = 0] HIRQ11 interrupt request by setting OBF4 to 1 is enabled [When IEDIR4 = 1] HIRQ11 interrupt is requested [Setting condition] Writing 1 after reading IRQ11E4 = 0
4	IRQ10E4	0	R/W	—	Host IRQ10 Interrupt Enable 4 Enables or disables an HIRQ10 interrupt request when OBF4 is set by an ODR4 write. 0: HIRQ10 interrupt request by OBF4 and IRQE10E4 is disabled [Clearing conditions] • Writing 0 to IRQ10E4 • LPC hardware reset, LPC software reset • Clearing OBF4 to 0 (when IEDIR4 = 0) 1: [When IEDIR4 = 0] HIRQ10 interrupt request by setting OBF4 to 1 is enabled [When IEDIR4 = 1] HIRQ10 interrupt is requested [Setting condition] Writing 1 after reading IRQ10E4 = 0

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
3	IRQ9E4	0	R/W	—	Host IRQ9 Interrupt Enable 4 Enables or disables an HIRQ9 interrupt request when OBF4 is set by an ODR4 write. 0: HIRQ9 interrupt request by OBF4 and IRQE9E4 is disabled [Clearing conditions] • Writing 0 to IRQ9E4 • LPC hardware reset, LPC software reset • Clearing OBF4 to 0 (when IEDIR4 = 0) 1: [When IEDIR4 = 0] HIRQ9 interrupt request by setting OBF4 to 1 is enabled [When IEDIR4 = 1] HIRQ9 interrupt is requested [Setting condition] Writing 1 after reading IRQ9E4 = 0
2	IRQ6E4	0	R/W	—	Host IRQ6 Interrupt Enable 4 Enables or disables an HIRQ6 interrupt request when OBF4 is set by an ODR4 write. 0: HIRQ6 interrupt request by OBF4 and IRQE6E4 is disabled [Clearing conditions] • Writing 0 to IRQ6E4 • LPC hardware reset, LPC software reset • Clearing OBF4 to 0 (when IEDIR4 = 0) 1: [When IEDIR4 = 0] HIRQ6 interrupt request by setting OBF4 to 1 is enabled [When IEDIR4 = 1] HIRQ6 interrupt is requested [Setting condition] Writing 1 after reading IRQ6E4 = 0

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
1	SMIE4	0	R/W	—	Host SMI Interrupt Enable 4 Enables or disables an SMI interrupt request when OBF4 is set by an ODR4 write. 0: Host SMI interrupt request by OBF4 and SMIE4 is disabled [Clearing conditions] • Writing 0 to SMIE4 • LPC hardware reset, LPC software reset • Clearing OBF4 to 0 (when IEDIR4 = 0) 1: [When IEDIR4 = 0] Host SMI interrupt request by setting OBF4 to 1 is enabled [When IEDIR4 = 1] Host SMI interrupt is requested [Setting condition] Writing 1 after reading SMIE4 = 0
0	—	0	R/W	—	Reserved The initial value should not be changed.

20.3.16 SERIRQ Control Register 3 (SIRQCR3)

SIRQCR3 contains bits that select the host interrupt request outputs.

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
7	SELIRQ15	0	R/W	—	Host IRQ Interrupt Select These bits select the state of the output on the SERIRQ pins. 0: SERIRQ pin output is in the Hi-Z state 1: SERIRQ pin output is low
6	SELIRQ14	0	R/W	—	
5	SELIRQ13	0	R/W	—	
4	SELIRQ8	0	R/W	—	
3	SELIRQ7	0	R/W	—	
2	SELIRQ5	0	R/W	—	
1	SELIRQ4	0	R/W	—	
0	SELIRQ3	0	R/W	—	

20.3.17 SERIRQ Control Register 4 (SIRQCR4)

SIRQCR4 is used to select the SERIRQ interrupt requests of the SCIF.

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
7 to 4	—	All 0	R/W	—	Reserved The initial value should not be changed.
3	SCSIRQ3	0	R/W	—	SCIF SERIRQ Request
2	SCSIRQ2	0	R/W	—	These bits select host interrupt requests of the SCIF.
1	SCSIRQ1	0	R/W	—	0000: No host interrupt request
0	SCSIRQ0	0	R/W	—	0001: HIRQ1
					0010: SMI
					0011: HIRQ3
					0100: HIRQ4
					0101: HIRQ5
					0110: HIRQ6
					0111: HIRQ7
					1000: HIRQ8
					1001: HIRQ9
					1010: HIRQ10
					1011: HIRQ11
					1100: HIRQ12
					1101: HIRQ13
					1110: HIRQ14
					1111: HIRQ15

20.3.18 SCIF Address Register (SCIFADRH, SCIFADRL)

SCIFADR sets the host addresses of the SCIF. Do not change the contents of SCIFADR during operation of the SCIF (i.e. while SCIFE is set to 1).

- SCIFADRH

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
7	—	0	R/W	—	SCIF Addresses 15 to 8
6	—	0	R/W	—	These bits set the host addresses of the SCIF.
5	—	0	R/W	—	
4	—	0	R/W	—	
3	—	0	R/W	—	
2	—	0	R/W	—	
1	—	1	R/W	—	
0	—	1	R/W	—	

- SCIFADRL

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
7	—	1	R/W	—	SCIF Addresses 7 to 0
6	—	1	R/W	—	These bits set the host addresses of the SCIF.
5	—	1	R/W	—	
4	—	1	R/W	—	
3	—	1	R/W	—	
2	—	0	R/W	—	
1	—	0	R/W	—	
0	—	0	R/W	—	

Note: When the SCIF is in use, set different addresses in the SCIFADR for channels 1, 2, 3, and 4.

20.3.19 Host Interface Select Register (HISEL)

HISEL selects the function of bits 7 to 4 in STR3 and selects the output of the host interrupt request signal of each frame.

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
7	SELSTR3	0	R/W	—	Status Register 3 Selection Selects the function of bits 7 to 4 in STR3 in combination with the TWRE bit in LADR3L. For details of STR3, see section 20.3.12, Status Registers 1 to 4 (STR1 to STR4). 0: Bits 7 to 4 in STR3 indicate processing status of the LPC interface. 1: [When TWRE = 1] Bits 7 to 4 in STR3 indicate processing status of the LPC interface. [When TWRE = 0] Bits 7 to 4 in STR3 are readable/writable bits which user can use as necessary
6	SELIRQ11	0	R/W	—	Host IRQ Interrupt Select These bits select the state of the output on the SERIRQ pins. 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	—	

20.4 Operation

20.4.1 LPC interface Activation

The LPC interface is activated by setting one of the following bits to 1: LPC3E to LPC1E in HICR0 and LPC4E in HICR4. When the LPC interface is activated, the related I/O ports (P37 to P30, P83 and P82) function as dedicated LPC interface input/output pins. In addition, setting the FGA20E, PMEE, LSMIE, and LSCIE bits to 1 adds the related I/O ports (P81, P80, PB0, and PB1) to the LPC interface's input/output pins.

Use the following procedure to activate the LPC interface after a reset release.

1. Read the signal line status and confirm that the LPC module can be connected. Also check that the LPC module is initialized internally.
2. When using channels 1, 2 and 4, set LADR1, LADR2, and LADR4 to determine the I/O address.
3. When using channel 3, set LADR3 to determine the I/O address and whether bidirectional data registers are to be used.
4. Set the enable bit (LPC4E to LPC1E) for the channel to be used.
5. Set the enable bits (FGA20E, PMEE, LSMIE, and LSCIE) for the additional functions to be used.
6. Set the selection bits for other functions (SDWNE, IEDIR).
7. As a precaution, clear the interrupt flags (LRST, SDWN, ABRT, OBF, and OBEI). Read IDR or TWR15 to clear IBF.
8. Set receive complete interrupt enable bits (IBFIE4 to IBFIE1, ERRIE, and OBEI) as necessary.

20.4.2 LPC I/O Cycles

There are 12 types of LPC transfer cycle: LPC memory read, LPC memory write, I/O read, I/O write, DMA read, DMA write, bus master memory read, bus master memory write, bus master I/O read, bus master I/O write, FW memory read, and FW memory write. Of these, the LPC of this LSI supports I/O read and I/O write.

An LPC transfer cycle is started when the $\overline{\text{LFRAME}}$ signal goes low in the bus idle state. If the $\overline{\text{LFRAME}}$ signal goes low when the bus is not idle, this means that a forced termination (abort) of the LPC transfer cycle has been requested.

In an I/O read cycle or I/O write cycle, transfer is carried out using LAD3 to LAD0 in the following order, in synchronization with LCLK. The host can be made to wait by sending back a value other than B'0000 in the slave's synchronization return cycle, but with the LPC of this LSI a value of B'0000 always returns.

If the received address matches the host address in an LPC register (IDR, ODR, STR, and TWR), the LPC interface enters the busy state; it returns to the idle state by output of a state count 12 turnaround. Register and flag changes are made at this timing, so in the event of a transfer cycle forced termination (abort), registers and flags are not changed.

The timing of the $\overline{\text{LFRAME}}$, LCLK, and LAD signals is shown in figures 20.2 and 20.3.

Table 20.3 LPC I/O Cycle

	I/O Read Cycle			I/O Write Cycle		
State Count	**Contents**	**Drive Source**	**Value (3 to 0)**	**Contents**	**Drive Source**	**Value (3 to 0)**
1	Start	Host	0000	Start	Host	0000
2	Cycle type/direction	Host	0000	Cycle type/direction	Host	0010
3	Address 1	Host	Bits 15 to 12	Address 1	Host	Bits 15 to 12
4	Address 2	Host	Bits 11 to 8	Address 2	Host	Bits 11 to 8
5	Address 3	Host	Bits 7 to 4	Address 3	Host	Bits 7 to 4
6	Address 4	Host	Bits 3 to 0	Address 4	Host	Bits 3 to 0
7	Turnaround (recovery)	Host	1111	Data 1	Host	Bits 3 to 0
8	Turnaround	None	ZZZZ	Data 2	Host	Bits 7 to 4
9	Synchronization	Slave	0000	Turnaround (recovery)	Host	1111
10	Data 1	Slave	Bits 3 to 0	Turnaround	None	ZZZZ
11	Data 2	Slave	Bits 7 to 4	Synchronization	Slave	0000
12	Turnaround (recovery)	Slave	1111	Turnaround (recovery)	Slave	1111
13	Turnaround	None	ZZZZ	Turnaround	None	ZZZZ

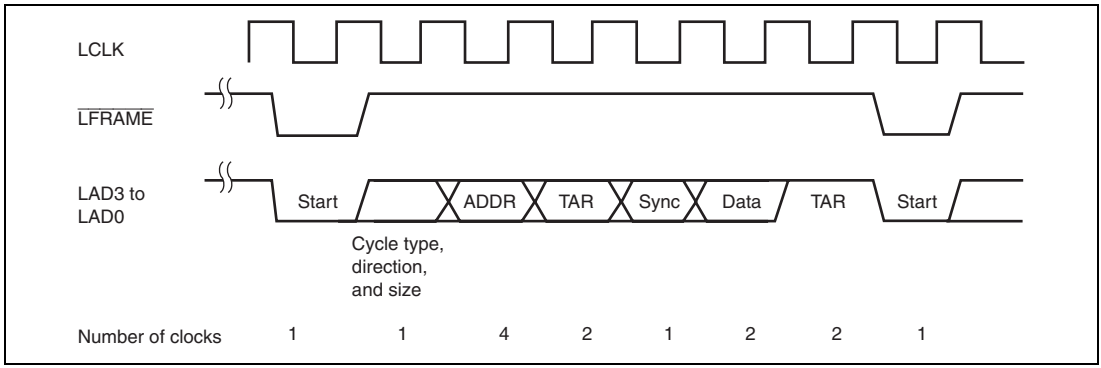


Figure 20.2 Typical LFRAME Timing

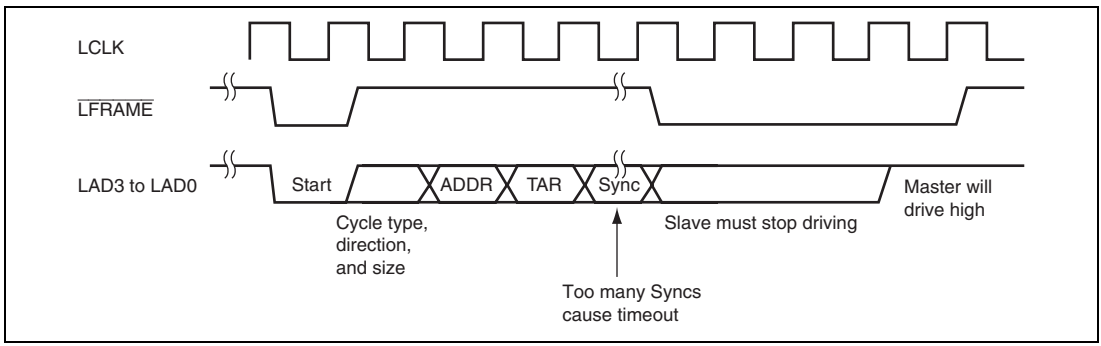


Figure 20.3 Abort Mechanism

20.4.3 Gate A20

The Gate A20 signal can mask address A20 to emulate the address mode of the 8086* architecture CPU used in personal computers. Normally, the Gate A20 signal can be controlled by a firmware. The fast Gate A20 function that realizes high-seed performance by hardware is enabled by setting the FGA20E bit to 1 in HICR0.

Note: An Intel microprocessor

(1) Regular Gate A20 Operation

Output of the Gate A20 signal can be controlled by an H'D1 command and data. When the slave (this LSI) receives data, it normally reads IDR1 in the interrupt handling routine activated by the IBFI1 interrupt. At this time, firmware copies bit 1 of data following an H'D1 command and outputs it on pin GA20.

(2) Fast Gate A20 Operation

The internal state of pin GA20 is initialized to 1 since the initial value of the FGA20E bit is 0. When the FGA20E bit is set to 1, pin P81/GA20 functions as the output of the fast GA20 signal. The state of pin GA20 can be monitored by reading bit GA20 in HICR2.

The initial output from this pin is 1, which is the initial value. Afterward, the host can manipulate the output from this pin by sending commands and data. This function is only available via the IDR1. The LPC decodes commands input from the host. When an H'D1 host command is detected, bit 1 of the data following the host command is output from pin GA20. This operation does not depend on firmware or interrupts, and is faster than the regular processing using interrupts. Table 20.4 shows the conditions that set and clear pin GA20. Figure 20.4 shows the GA20 output flow. Table 20.5 indicates the GA20 output signal values.

Table 20.4 GA20 Setting/Clearing Timing

Pin Name	Setting Condition	Clearing Condition
GA20	When bit 1 of the data that follows an H'D1 host command is 1	When bit 1 of the data that follows an H'D1 host command is 0

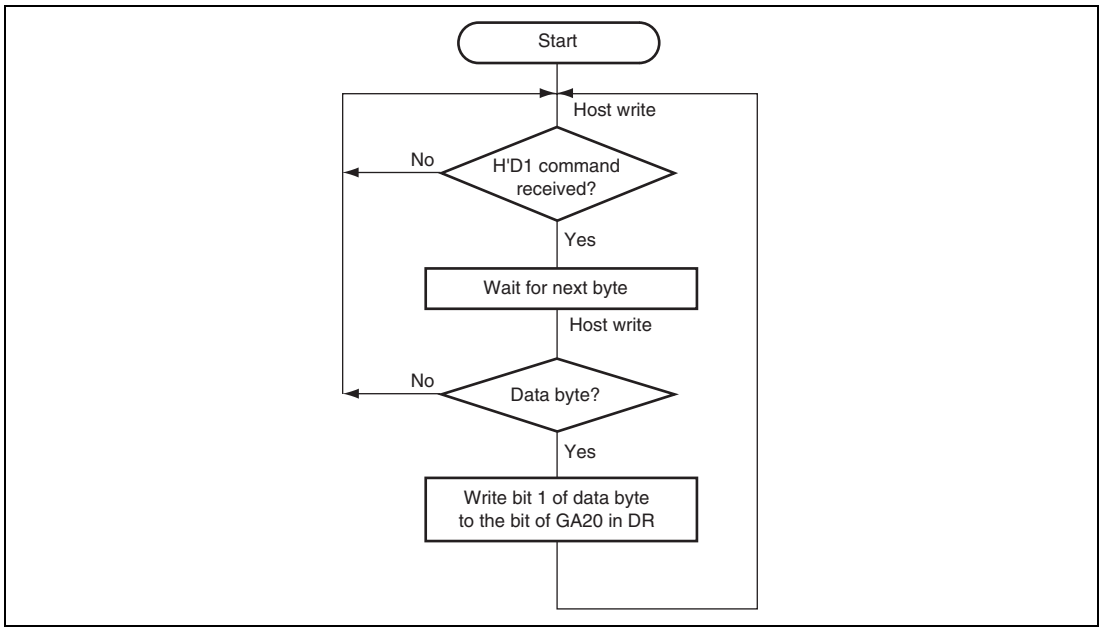


Figure 20.4 GA20 Output

Table 20.5 Fast Gate A20 Output Signals

C/$\overline{D}$1	Data/Command	Internal CPU Interrupt Flag (IBF)	GA20 (P81)	Remarks
1	H'D1 command	0	Q	Turn-on sequence
0	1 data*[1]	0	1	
1	H'FF command	0	Q (1)	
1	H'D1 command	0	Q	Turn-off sequence
0	0 data*[2]	0	0	
1	H'FF command	0	Q (0)	
1	H'D1 command	0	Q	Turn-on sequence (abbreviated form)
0	1 data*[1]	0	1	
1/0	Command other than H'FF and H'D1	1	Q (1)	
1	H'D1 command	0	Q	Turn-off sequence (abbreviated form)
0	0 data*[2]	0	0	
1/0	Command other than H'FF and H'D1	1	Q (0)	
1	H'D1 command	0	Q	Cancelled sequence
1	Command other than H'D1	1	Q	
1	H'D1 command	0	Q	Retriggered sequence
1	H'D1 command	0	Q	
1	H'D1 command	0	Q	Consecutively executed sequences
0	Any data	0	1/0	
1	H'D1 command	0	Q (1/0)	

Notes: 1. Any data with bit 1 set to 1.
2. Any data with bit 1 cleared to 0.

20.4.4 LPC Interface Shutdown Function (LPCPD)

The LPC interface can be placed in the shutdown state according to the state of the $\overline{\text{LPCPD}}$ pin. There are two kinds of LPC interface shutdown state: LPC hardware shutdown and LPC software shutdown. The LPC hardware shutdown state is controlled by the $\overline{\text{LPCPD}}$ pin, while the LPC software shutdown state is controlled by the SDWNB bit. In both states, the LPC interface enters the reset state by itself, and is no longer affected by external signals other than the $\overline{\text{LRESET}}$ and $\overline{\text{LPCPD}}$ signals.

Placing the slave in sleep mode or software standby mode is effective in reducing current dissipation in the shutdown state. If software standby mode is set, some means must be provided for exiting software standby mode before clearing the shutdown state with the $\overline{\text{LPCPD}}$ signal.

If the SDWNE bit has been set to 1 beforehand, the LPC hardware shutdown state is entered at the same time as the $\overline{\text{LPCPD}}$ signal falls, and prior preparation is not possible. If the LPC software shutdown state is set by means of the SDWNB bit, on the other hand, the LPC software shutdown state cannot be cleared at the same time as the rising edge of the $\overline{\text{LPCPD}}$ signal. Taking these points into consideration, the following operating procedure uses a combination of LPC software shutdown and LPC hardware shutdown.

1. Clear the SDWNE bit to 0.
2. Set the ERRIE bit to 1 and wait for an interrupt by the SDWN flag.
3. When an ERRI interrupt is generated by the SDWN flag, check the LPC interface internal status flags and perform any necessary processing.
4. Set the SDWNB bit to 1 to set LPC software standby mode.
5. Set the SDWNE bit to 1 and make a transition to LPC hardware standby mode. The SDWNB bit is cleared automatically.
6. Check the state of the $\overline{\text{LPCPD}}$ signal to make sure that the $\overline{\text{LPCPD}}$ signal has not risen during steps 3 to 5. If the signal has risen, clear SDWNE to 0 to return to the state in step 1.
7. If software standby mode has been set, exit software standby mode by some means independent of the LPC.
8. When a rising edge is detected in the $\overline{\text{LPCPD}}$ signal, the SDWNE bit is automatically cleared to 0. If the slave has been placed in sleep mode, the mode is exited by means of $\overline{\text{LRESET}}$ signal input, on completion of the LPC transfer cycle, or by some other means.

Table 20.6 shows the scope of the LPC interface pin shutdown.

Table 20.6 Scope of LPC Interface Pin Shutdown

Abbreviation	Port	Scope of Shutdown	I/O	Notes
LAD3 to LAD0	P33 to P30	O	I/O	Hi-Z
$\overline{\text{LFRAME}}$	P34	O	Input	Hi-Z
$\overline{\text{LRESET}}$	P35	X	Input	LPC hardware reset function is active
LCLK	P36	O	Input	Hi-Z
SERIRQ	P37	O	I/O	Hi-Z
LSCI	PB1	Δ	I/O	Hi-Z, only when LSCIE = 1
$\overline{\text{LSMI}}$	PB0	Δ	I/O	Hi-Z, only when LSMIE = 1
$\overline{\text{PME}}$	P80	Δ	I/O	Hi-Z, only when PMEE = 1
GA20	P81	Δ	I/O	Hi-Z, only when FGA20E = 1
$\overline{\text{CLKRUN}}$	P82	O	Input	Hi-Z
$\overline{\text{LPCPD}}$	P83	X	Input	Needed to clear shutdown state

[Legend]

O: Pin that is shutdown by the shutdown function

Δ: Pin that is shutdown only when the LPC function is selected by register setting

X: Pin that is not shutdown

In the LPC shutdown state, the LPC's internal state and some register bits are initialized. The order of priority of LPC shutdown and reset states is as follows.

1. System reset (reset by $\overline{\text{RES}}$ pin input, or WDT overflow)
   All register bits, including bits LPC4E to LPC1E, are initialized.
2. LPC hardware reset (reset by $\overline{\text{LRESET}}$ pin input)
   LRSTB, SDWNE, and SDWNB bits are cleared to 0.
3. LPC software reset (reset by LRSTB)
   SDWNE and SDWNB bits are cleared to 0.
4. LPC hardware shutdown
   SDWNB bit is cleared to 0.
5. LPC software shutdown

The scope of the initialization in each mode is shown in table 20.7.

Table 20.7 Scope of Initialization in Each LPC interface Mode

Items Initialized	System Reset	LPC Reset	LPC Shutdown
LPC transfer cycle sequencer (internal state), LPCBSY and ABRT flags	Initialized	Initialized	Initialized
SERIRQ transfer cycle sequencer (internal state), CLKREQ and IRQBSY flags	Initialized	Initialized	Initialized
LPC interface flags (IBF1, IBF2, IBF3A, IBF3B, IBF4, MWMF, C/$\overline{D}$1, C/$\overline{D}$2, C/$\overline{D}$3, C/$\overline{D}$4, OBF1, OBF2, OBF3A, OBF3B, OBF4, SWMF, DBU), GA20 (internal state)	Initialized	Initialized	Retained
Host interrupt enable bits (IRQ1E1, IRQ12E1, SMIE2, IRQ6E2, IRQ9E2 to IRQ11E2, SMIE3B, SMIE3A, IRQ6E3, IRQ9E3 to IRQ11E3, SELREQ, SMIE4, IRQ6E4, IRQ9E4 to IRQ11E4, IEDIR2 to IEDIR4), Q/$\overline{C}$ flag	Initialized	Initialized	Retained
LRST flag	Initialized (0)	Can be set/cleared	Can be set/cleared
SDWN flag	Initialized (0)	Initialized (0)	Can be set/cleared
LRSTB bit	Initialized (0)	HR: 0 SR: 1	0 (can be set)
SDWNB bit	Initialized (0)	Initialized (0)	HS: 0 SS: 1
SDWNE bit	Initialized (0)	Initialized (0)	HS: 1 SS: 0 or 1
LPC interface operation control bits (LPC4E to LPC1E, FGA20E, LADR1 to LADR4, IBFIE1 to IBFIE4, PMEE, PMEB, LSMIE, LSMIB, LSCIE, LSCIB, TWRE, SELSTR3, SELIRQ1, SELSMI, SELIRQ3 to SELIRQ15, OBEIE, SCIFE, IDR1 to IDR4, ODR1 to ODR4, TWR0 to TWR15, SCSIRQ0 to SCSIRQ3, and SCIFADRH/L)	Initialized	Retained	Retained
$\overline{\text{LRESET}}$ signal	Input (port function	Input	Input
$\overline{\text{LPCPD}}$ signal		Input	Input
LAD3 to LAD0, $\overline{\text{LFRAME}}$, LCLK, SERIRQ, $\overline{\text{CLKRUN}}$ signals		Input	Hi-Z
$\overline{\text{PME}}$, $\overline{\text{LSMI}}$, LSCI, GA20 signals (when function is selected)		Output	Hi-Z
$\overline{\text{PME}}$, $\overline{\text{LSMI}}$, LSCI, GA20 signals (when function is not selected)		Port function	Port function

Note: System reset: Reset by $\overline{\text{RES}}$ pin input, or WDT overflow
LPC reset: Reset by LPC hardware reset (HR) or LPC software reset (SR)
LPC shutdown: Reset by LPC hardware shutdown (HS) or LPC software shutdown (SS)

Figure 20.5 shows the timing of the $\overline{\text{LPCPD}}$ and $\overline{\text{LRESET}}$ signals.

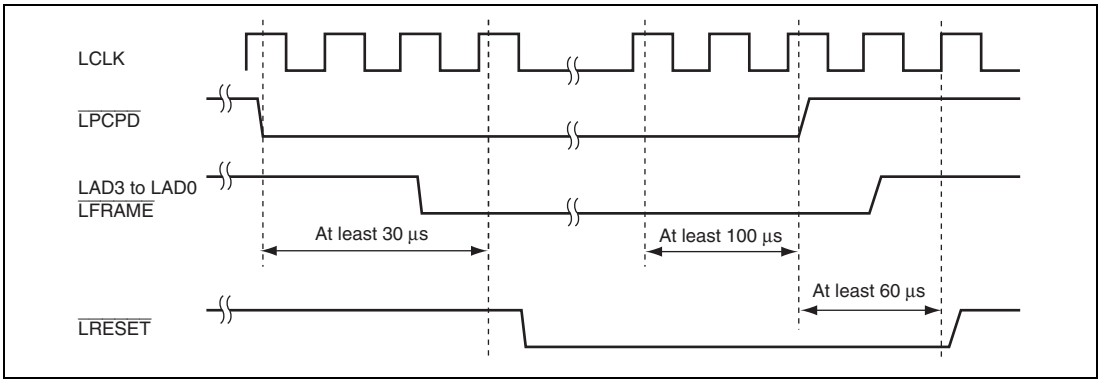


Figure 20.5 Power-Down State Termination Timing

20.4.5 LPC Interface Serialized Interrupt Operation (SERIRQ)

A host interrupt request can be issued from the LPC interface by means of the SERIRQ pin. In a host interrupt request via the SERIRQ pin, LCLK cycles are counted from the start frame of the serialized interrupt transfer cycle generated by the host or a peripheral function, and a request signal is generated by the frame corresponding to that interrupt. The timing is shown in figure 20.6.

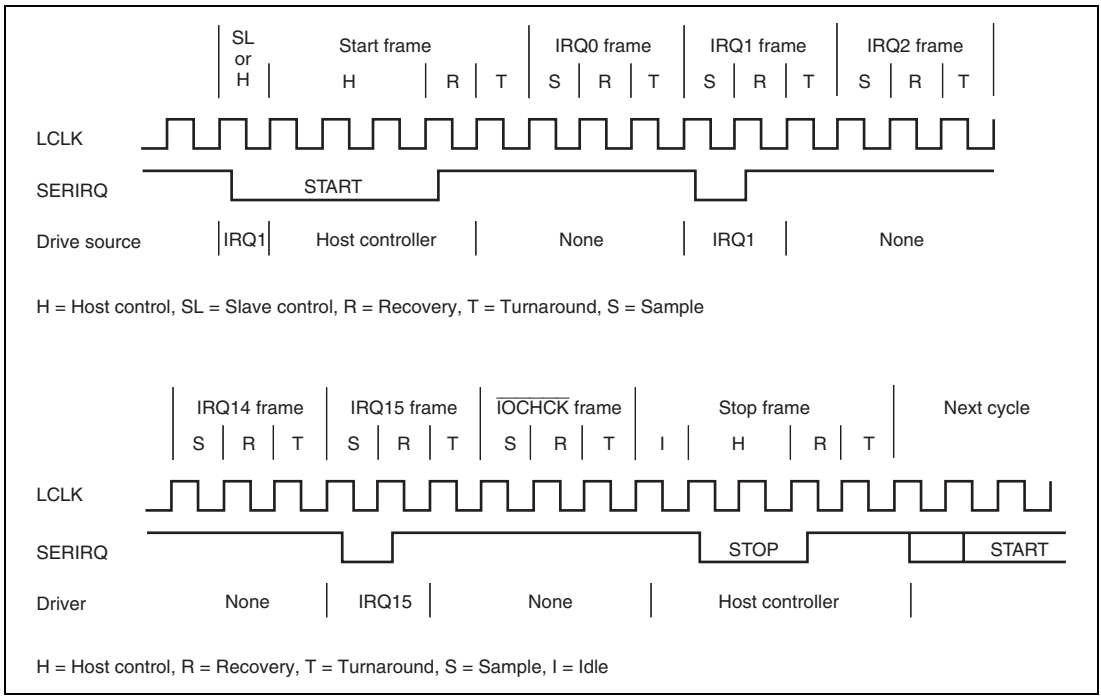


Figure 20.6 SERIRQ Timing

The serialized interrupt transfer cycle frame configuration is as follows. Two of the states comprising each frame are the recover state in which the SERIRQ signal is returned to the 1-level at the end of the frame, and the turnaround state in which the SERIRQ signal is not driven. The recover state must be driven by the host or slave that was driving the preceding state.

Table 20.8 Serialized Interrupt Transfer Cycle Frame Configuration

	Serial Interrupt Transfer Cycle			
Frame Count	**Contents**	**Drive Source**	**Number of States**	**Notes**
0	Start	Slave Host	6	In quiet mode only, slave drive possible in first state, then next 3 states 0-driven by host
1	IRQ0	Slave	3	
2	IRQ1	Slave	3	Drive possible in LPC channel 1 and SCIF
3	SMI	Slave	3	Drive possible in LPC channels 2, 3, 4, and SCIF
4	IRQ3	Slave	3	Drive possible in SCIF
5	IRQ4	Slave	3	Drive possible in SCIF
6	IRQ5	Slave	3	Drive possible in SCIF
7	IRQ6	Slave	3	Drive possible in LPC channels 2, 3, 4, and SCIF
8	IRQ7	Slave	3	Drive possible in SCIF
9	IRQ8	Slave	3	Drive possible in SCIF
10	IRQ9	Slave	3	Drive possible in LPC channels 2, 3, 4, and SCIF
11	IRQ10	Slave	3	Drive possible in LPC channels 2, 3, 4, and SCIF
12	IRQ11	Slave	3	Drive possible in LPC channels 2, 3, 4, and SCIF
13	IRQ12	Slave	3	Drive possible in LPC channel 1 and SCIF
14	IRQ13	Slave	3	Drive possible in SCIF
15	IRQ14	Slave	3	Drive possible in SCIF
16	IRQ15	Slave	3	Drive possible in SCIF
17	IOCHCK	Slave	3	
18	Stop	Host	Undefined	First, 1 or more idle states, then 2 or 3 states 0-driven by host 2 states: Quiet mode next 3 states: Continuous mode next

There are two modes—continuous mode and quiet mode—for serialized interrupts. The mode initiated in the next transfer cycle is selected by the stop frame of the serialized interrupt transfer cycle that ended before that cycle.

In continuous mode, the host initiates host interrupt transfer cycles at regular intervals. In quiet mode, the slave with interrupt sources requiring a request can also initiate an interrupt transfer cycle, in addition to the host. In quiet mode, since the host does not necessarily initiate interrupt transfer cycles, it is possible to suspend the clock (LCLK) supply and enter the power-down state. In order for a slave to transfer an interrupt request in this case, a request to restart the clock must first be issued to the host. For details see section 20.4.6, LPC Interface Clock Start Request.

20.4.6 LPC Interface Clock Start Request

A request to restart the clock (LCLK) can be sent to the host by means of the $\overline{\text{CLKRUN}}$ pin. With LPC data transfer and SERIRQ in continuous mode, a clock restart is never requested since the transfer cycles are initiated by the host. With SERIRQ in quiet mode, when a host interrupt request is generated the $\overline{\text{CLKRUN}}$ signal is driven and a clock (LCLK) restart request is sent to the host. The timing for this operation is shown in figure 20.7.

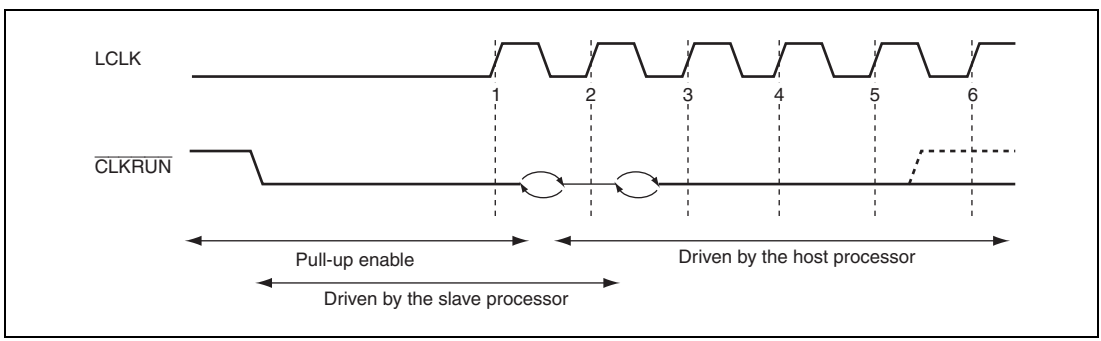


Figure 20.7 Clock Start Request Timing

Cases other than SERIRQ in quiet mode when clock restart is required must be handled with a different protocol, using the $\overline{\text{PME}}$ signal, etc.

20.4.7 SCIF Control from LPC Interface

Setting the SCIFE bit in HICR5 to 1 allows the LPC host to communicate with the SCIF. Then, the LPC interface can access the registers of the module SCIF other than SCIFCR. For details on transmission and reception, see section 17, Serial Communication Interface with FIFO (SCIF).

20.5 Interrupt Sources

20.5.1 IBFI1, IBFI2, IBFI3, IBFI4, OBEI, and ERRI

The host has six interrupt requests for the slave (this LSI): IBF1, IBF2, IBF3, IBF4, OBEI, and ERRI. IBFI1, IBFI2, IBFI3, and IBFI4 are IDR receive complete interrupts for IDR1, IDR2, and IDR3 and TWR, respectively. The ERRI interrupt indicates the occurrence of a special state such as an LPC reset, LPC shutdown, or transfer cycle abort. The LMCI and LMCUI interrupts are command receive complete interrupts. OBEI is an output buffer empty interrupt. An interrupt request is enabled by setting the corresponding enable bit.

Table 20.9 Receive Complete Interrupts and Error Interrupt

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 TWRE and IBFIE3 are set to 1 and reception is completed up to TWR15
IBFI4	When IBFIE4 is set to 1 and IDR4 reception is completed
OBEI	When OBEIE is set to 1 with OBEI set to 1.
ERRI	When ERRIE is set to 1 and one of LRST, SDWN and ABRT is set to 1

20.5.2 SMI, HIRQ1, HIRQ3, HIRQ4, HIRQ5, HIRQ6, HIRQ7, HIRQ8, HIRQ9, HIRQ10, HIRQ11, HIRQ12, HIRQ13, HIRQ14, and HIRQ15

The LPC interface can request 15 kinds of host interrupt by means of SERIRQ. HIRQ1 and HIRQ12 are used on LPC channel 1 and the SCIF, while SMI, HIRQ6, HIRQ9, HIRQ10, and HIRQ11 can be requested from LPC channel 2, 3, 4 or SCIF. HIRQ3, HIRQ4, HIRQ5, HIRQ7, HIRQ8, HIRQ13, HIRQ14, and HIRQ15 are only for the SCIF.

There are two ways of clearing a host interrupt request when the LPC channels are used.

When the IEDIR bit in SIRQCR is cleared to 0, host interrupt sources and LPC channels are all linked to the host interrupt request enable bits. When the OBF flag is cleared to 0 by a read of ODR or TWR15 by the host in the corresponding LPC channel, the corresponding host interrupt 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 requested by the only upon the host interrupt enable bits. The host interrupt enable bit is not cleared when OBF is cleared. Therefore, SMIE1, SMIE2, SMIE3A and SMIE3B, SMIE4, IRQ6En, IRQ9En, IRQ10En, and IRQ11En lose their respective functional differences. In order to clear a host interrupt request, it is necessary to clear the host interrupt enable bit. (n = 2 to 4.)

When the SCIF channels are used, clearing the DDCD bit in FMSR of the SCIF clears a host interrupt request.

Table 20.10 summarizes the methods of setting and clearing these bits when the LPC channels are used, and table 20.11 summarizes the methods of setting and clearing these bits when the SCIF channels are used. Figure 20.8 shows the processing flowchart.

Table 20.10 HIRQ Setting and Clearing Conditions when LPC Channels are Used

Host Interrupt	Setting Condition	Clearing Condition
HIRQ1	Internal CPU writes to ODR1, then reads 0 from bit IRQ1E1 and writes 1	Internal CPU writes 0 to bit IRQ1E1, or host reads ODR1
HIRQ12	Internal CPU writes to ODR1, then reads 0 from bit IRQ12E1 and writes 1	Internal CPU writes 0 to bit IRQ12E1, or host reads ODR1
SMI (IEDIR2 = 0, IEDIR3 = 0, or IEDIR4 = 0)	Internal CPU • writes to ODR2, then reads 0 from bit SMIE2 and writes 1 • writes to ODR3, then reads 0 from bit SMIE3A and writes 1 • writes to TWR15, then reads 0 from bit SMIE3B and writes 1 • writes to ODR4, then reads 0 from bit SMIE4 and writes 1	Internal CPU • writes 0 to bit SMIE2, or host reads ODR2 • writes 0 to bit SMIE3A, or host reads ODR3 • writes 0 to bit SMIE3B, or host reads TWR15 • writes 0 to bit SMIE4, or host reads ODR4
SMI (IEDIR2 = 1, IEDIR3 = 1, or IEDIR4 = 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 • reads 0 from bit SMIE4, then writes 1	Internal CPU • writes 0 to bit SMIE2 • writes 0 to bit SMIE3A • writes 0 to bit SMIE3B • writes 0 to bit SMIE4
HIRQi (i = 6, 9, 10, 11) (IEDIR2 = 0, IEDIR3 = 0, or IEDIR4 = 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 • writes to ODR4, then reads 0 from bit IRQiE4 and writes 1	Internal CPU • writes 0 to bit IRQiE2, or host reads ODR2 • CPU writes 0 to bit IRQiE3, or host reads ODR3 • CPU writes 0 to bit IRQiE4, or host reads ODR4
HIRQi (i = 6, 9, 10, 11) (IEDIR2 = 1, IEDIR3 = 1, or IEDIR4 = 1)	Internal CPU • reads 0 from bit IRQiE2, then writes 1 • reads 0 from bit IRQiE3, then writes 1 • reads 0 from bit IRQiE4, then writes 1	Internal CPU • writes 0 to bit IRQiE2 • writes 0 to bit IRQiE3 • writes 0 to bit IRQiE4

Table 20.11 HIRQ Setting and Clearing Conditions when SCIF Channels are Used

Host Interrupt	Setting Condition	Clearing Condition
HIRQi (i = 1 to 15)	Internal CPU sets the corresponding SERIRQ host interrupt request for the SCIF in SIRQCR4 (for details, see the description of SIRQCR4). Changes in the SCIF input signal $\overline{\text{DCD}}$ are detected.	Reads FMSR and clears the DDCD bit in FMSR

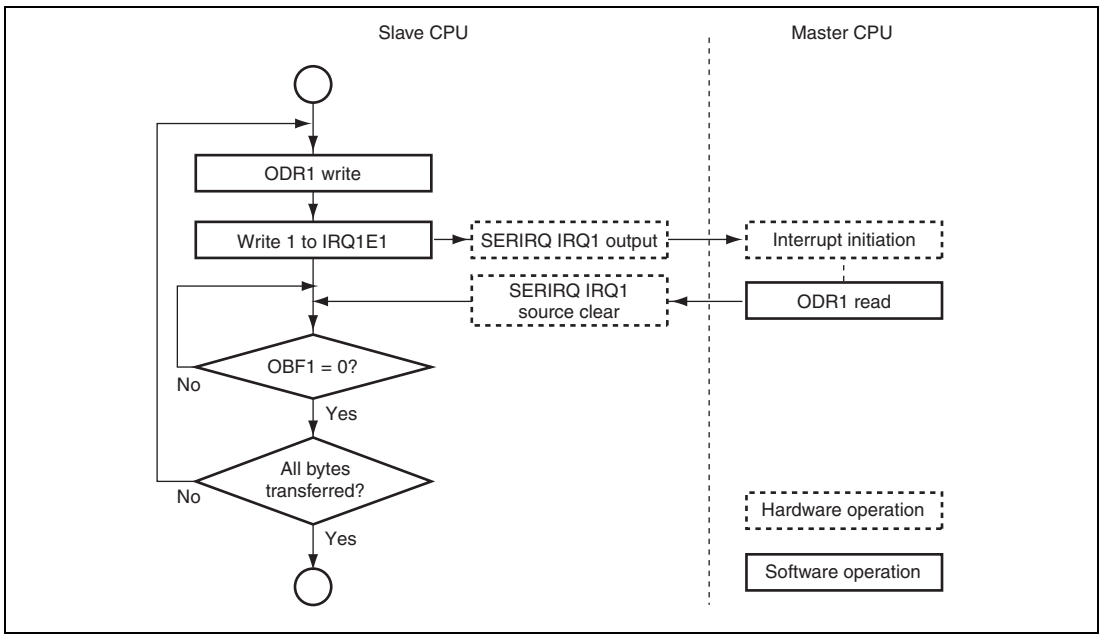


Figure 20.8 HIRQ Flowchart (Example of Channel 1)

20.6 Usage Note

20.6.1 Data Conflict

The LPC interface provides buffering of asynchronous data from the host and slave (this LSI), but an interface protocol that uses the flags in STR must be followed to avoid data conflict. For example, if the host and slave both try to access IDR or ODR at the same time, the data will be corrupted. To prevent simultaneous accesses, IBF and OBF must be used to allow access only to data for which writing has finished.

Unlike the IDR and ODR registers, the transfer direction is not fixed for the bidirectional data registers (TWR). MWMF and SWMF are provided in STR to handle this situation. After writing to TWR0, MWMF and SWMF must be used to confirm that the write authority for TWR1 to TWR15 has been obtained.

Table 20.12 shows host address examples for LADR3 and registers, IDR3, ODR3, STR3, TWR0MW, TWR0SW, and TWR1 to TWR15.

Table 20.12 Host Address Example

Register	Host Address when LADR3 = H'A24F	Host Address when LADR3 = H'3FD0
IDR3	H'A24A and H'A24E	H'3FD0 and H'3FD4
ODR3	H'A24A	H'3FD0
STR3	H'A24E	H'3FD4
TWR0MW	H'A250	H'3FC0
TWR0SW	H'A250	H'3FC0
TWR1	H'A251	H'3FC1
TWR2	H'A252	H'3FC2
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

Section 21 FSI Interface

This LSI incorporates the SPI flash memory serial interface (FSI) that supports the communication between this LSI and SPI flash memory. The FSI performs communications using the LPC or CPU of this LSI as a master.

21.1 Features

Figure 21.1 shows a block diagram of the FSI.

- Supports communications between this LSI and SPI flash memory.
- Can operate as a master.
- Transfer clock selectable from system clock ϕ or LCLK.
- Four interrupt sources: Transmit end, receive data full, and command and write receive interrupts
- Direct transfer between LPC and SPI: Supports Read instruction, and Byte/Page-Program, and AAI-Program instructions.
- LPC-SPI command transfer: Supports instructions other than above.
- Supports LPC/FW memory cycles of the LPC interface.
- Supports byte, word, and longword transfers of FW memory cycles.
- Provides independent LPC communication enable bits
- Supports LPC reset and LPC shut-down.

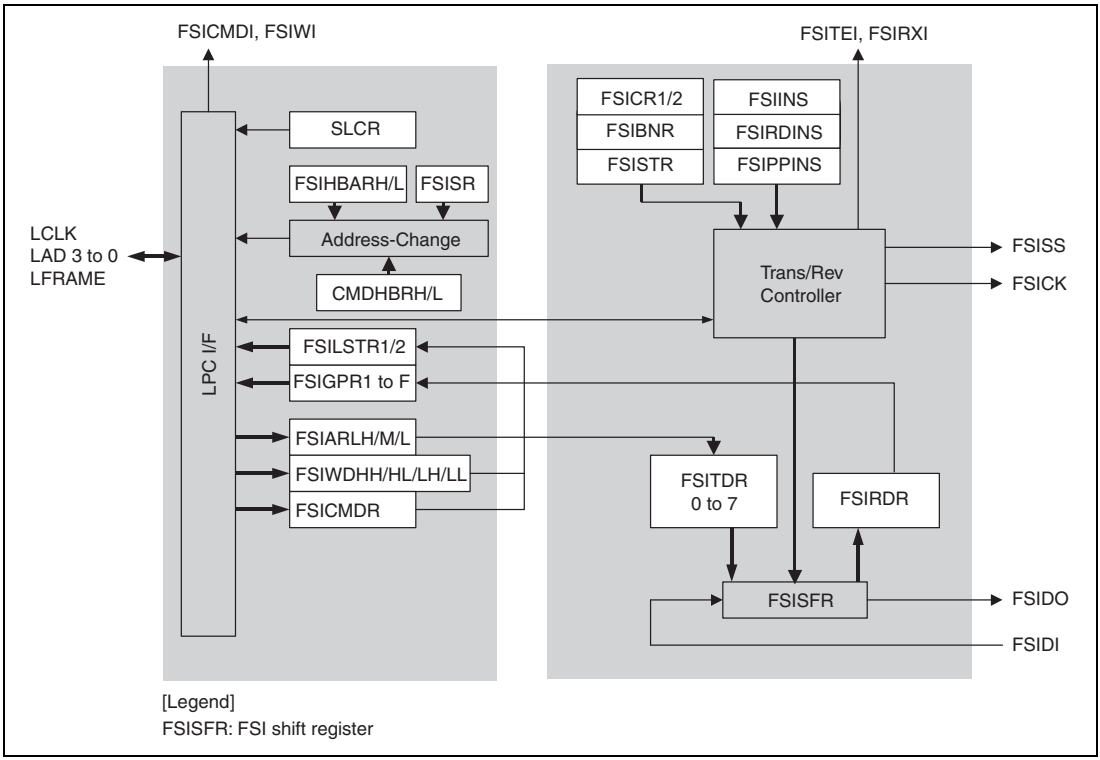


Figure 21.1 FSI Block Diagram

21.2 Input/Output Pins

Table 21.1 shows the input/output pins of the FSI.

Table 21.1 Pin Configuration

Pin Name	Symbol	I/O	Function
FSI slave select	FSISS	Output	FSI slave select signal
FSI clock	FSICK	Output	FSI clock signal
FSI master data input	FSIDI	Input	FSI data input signal
FSI master data output	FSIDO	Output	FSI address/direction/data output signal

For details on the input/output pins of the LPC interface, see section 20.2, Input/Output Pins.

Table 21.2 shows the initial state of the FSI input/output pins when the FSIE bit in the FSICR1 register is set to 1.

Table 21.2 Initial State of FSI Pins (when FSIE = 1)

Pin Name	Symbol	Pin State When FSIE is Set to 1
FSI slave select	FSISS	Outputs high level.
FSI clock	FSICK	Outputs high level or low level depending on CPHS and CPOS.
FSI master data input	FSIDI	Inputs data.
FSI master data output	FSIDO	Outputs high level.

21.3 Register Description

The FSI consists of the following registers.

Table 21.3 List of Register Addresses

Register Name	Abbreviation	R/W		Initial Value	Address
		EC	Host		
FSI control register 1	FSICR1	R/W	—	H'00	H'FC90
FSI control register 2	FSICR2	R/W	—	H'00	H'FC91
FSI byte count register	FSIBNR	R/W	—	H'00	H'FC92
FSI instruction register	FSIINS	R/W	—	H'00	H'FC93
FSI read instruction register	FSIRDINS	R/W	—	H'00	H'FC94
FSI program instruction register	FSIPPINS	R/W	—	H'00	H'FC95
FSI status register	FSISTR	R/W	—	H'00	H'FC96
FSI transmit data register 0	FSITDR0	R/W	—	H'00	H'FC98
FSI transmit data register 1	FSITDR1	R/W	—	H'00	H'FC99
FSI transmit data register 2	FSITDR2	R/W	—	H'00	H'FC9A
FSI transmit data register 3	FSITDR3	R/W	—	H'00	H'FC9B
FSI transmit data register 4	FSITDR4	R/W	—	H'00	H'FC9C
FSI transmit data register 5	FSITDR5	R/W	—	H'00	H'FC9D
FSI transmit data register 6	FSITDR6	R/W	—	H'00	H'FC9E
FSI transmit data register 7	FSITDR7	R/W	—	H'00	H'FC9F
FSI receive data register	FSIRDR	R	—	H'00	H'FCA0
FSI access host base address register H	FSIHBARH	R/W	—	H'00	H'FC50
FSI access host base address register L	FSIHBARL	R/W	—	H'00	H'FC51
FSI flash memory size register	FSISR	R/W	—	H'00	H'FC52
FSI command host base address register H	CMDHBARH	R/W	—	H'00	H'FC53
FSI command host base address register L	CMDHBARL	R/W	—	H'00	H'FC54
FSI command register	FSICMDR	R	—	H'00	H'FC55
FSILPC command status register 1	FSILSTR1	R/W	R	H'00	H'FC56

Register Name	Abbreviation	R/W EC	R/W Host	Initial Value	Address
FSI general-purpose register 1	FSIGPR1	R/W	R	H'00	H'FC57
FSI general-purpose register 2	FSIGPR2	R/W	R	H'00	H'FC58
FSI general-purpose register 3	FSIGPR3	R/W	R	H'00	H'FC59
FSI general-purpose register 4	FSIGPR4	R/W	R	H'00	H'FC5A
FSI general-purpose register 5	FSIGPR5	R/W	R	H'00	H'FC5B
FSI general-purpose register 6	FSIGPR6	R/W	R	H'00	H'FC5C
FSI general-purpose register 7	FSIGPR7	R/W	R	H'00	H'FC5D
FSI general-purpose register 8	FSIGPR8	R/W	R	H'00	H'FC5E
FSI general-purpose register 9	FSIGPR9	R/W	R	H'00	H'FC5F
FSI general-purpose register A	FSIGPRA	R/W	R	H'00	H'FC60
FSI general-purpose register B	FSIGPRB	R/W	R	H'00	H'FC61
FSI general-purpose register C	FSIGPRC	R/W	R	H'00	H'FC62
FSI general-purpose register D	FSIGPRD	R/W	R	H'00	H'FC63
FSI general-purpose register E	FSIGPRE	R/W	R	H'00	H'FC64
FSI general-purpose register F	FSIGPRF	R/W	R	H'00	H'FC65
FSILPC control register	SLCR	R/W	—	H'00	H'FC66
FSI address register H	FSIARH	R	—	H'00	H'FC67
FSI address register M	FSIARM	R	—	H'00	H'FC68
FSI address register L	FSIARL	R	—	H'00	H'FC69
FSI write data register HH	FSIWDRHH	R	—	H'00	H'FC6A
FSI write data register HL	FSIWDRHL	R	—	H'00	H'FC6B
FSI write data register LH	FSIWDRLH	R	—	H'00	H'FC6C
FSI write data register LL	FSIWDRLL	R	—	H'00	H'FC6D
FSI LPC command register 2	FSILSTR2	R/W	—	H'01	H'FC6E

Note: 1. Before accessing these registers, clear bit 0 in MSTPCRL (MSTP0) and bit 2 in MSTPCRA (MSTPA2) to 0.

2. "R/W" in table 21.3 has the following meanings.

a) "R/W EC" indicates the access from the EC (Embedded Controller = this LSI).

b) "R/W Host" indicates the access from the host.

21.3.1 FSI Control Register 1 (FSICR1)

The FSICR1 control bits are classified into three functionalities: resetting the FSI internal signals, enabling/disabling FSI functions, and selecting FSI functions.

Bit	Bit Name	Initial Value	R/W		Description
			EC	Host	
7	SRES	0	R/W	—	Software Reset Controls initialization of the FSI internal sequencer. 0: Normal state 1: Clears the internal sequencer. Writing 1 to this bit generates a clear signal for the sequencer in the corresponding module, resulting in the initialization of the FSI's internal state.
6	FSIE	0	R/W	—	FSI Enable 0: Disables FSI operation. 1: Enables FSI operation. The following shows the initial state of the FSI pins when FSIE is set to 1: FSISS: Outputs high level. FSICK: Outputs high level or low level depending on DPHS and CPOS. FSIDO: Outputs high level. FSIDI: Inputs data.
5	FRDE	0	R/W	—	Fast-Read Enable 0: The FSI is in normal read operation mode. 1: The FSI is in fast-read operation mode.
4	AAIE	0	R/W	—	AAI (Auto Address Increment) Program Enable 0: The FSI performs byte-program operation. 1: The FSI performs AAI program operation.

Bit	Bit Name	Initial Value	R/W EC	R/W Host	Description
3	CPHS	0	R/W	—	CPHS: Selects the polarity of the FSICK clock.
2	CPOS	0	R/W	—	CPOS: Selects the phase of the FSICK clock. CPHS CPOS 0 0 Initial value of FSICK: Low level Data changes at the FSICK falling edge. 1 1 Initial value of FSICK: High level Data changes at the FSICK falling edge. 0 1 Setting prohibited 1 0 Setting prohibited
1	—	All 0	R/W	—	Reserved The initial value should not be modified.
0	CKSEL	0	R/W	—	Clock select 0: Selects the system clock for FSICK 1: Selects LCLK for FSICK Note: Before selecting LCLK for FSICK, clear the CPHS and CPOS bits of FSICR1 to 0.

21.3.2 FSI Control Register 2 (FSICR2)

The FSICR2 control bits are classified into two functionalities: enabling/disabling the FSI communications and enabling/disabling the FSI internal interrupts.

Bit	Bit Name	Initial Value	R/W		Description
			EC	Host	
7	TE	0	R/W	—	FSI Transmit Enable Controls FSI transmission and indicates transmission status in combination with the LFBUSY bit. 0: FSI transmission wait state [Clearing condition] When FSI data transmission is completed. 1: When LFBUSY = 0: Starts transmission. When LFBUSY = 1: FSI transmission is in progress (automatically set).
6	RE	0	R/W	—	FSI Reception Enable Controls FSI reception and indicates reception status in combination with the LFBUSY bit. 0: FSI reception wait state [Clearing condition] When FSI data reception is completed. 1: When LFBUSY = 0: Starts reception. When LFBUSY = 1: FSI reception is in progress (automatically set).
5	FSITEIE	0	R/W	—	FSI Transmit End Interrupt Enable 0: Disables the FSITEI interrupt request. 1: Enables the FSITEI interrupt request.
4	FSIRXIE	0	R/W	—	FSI Receive Interrupt Enable 0: Disables the FSIRXI interrupt request. 1: Enables the FSIRXI interrupt request.
3 to 0	—	All 0	R/W	—	Reserved The initial value should not be modified.

21.3.3 FSI Byte Count Register (FSIBNR)

The FSIBNR sets the number of bytes to be transmitted or received by the FSI. This register should not be set in the processing other than FSICMDI and FSIWI interrupt processing.

Bit	Bit Name	Initial Value	R/W EC	R/W Host	Description
7 to 4	TBN3	0	R/W	—	Transmit Byte Count 3-0
	TBN2	0			These bits specify the number of data bytes to be transmitted. The TBN value is decremented each time one byte of FSI data transmission is completed. When the FSI transmission ends, TBN is cleared to B'0000.
	TBN1	0			
	TBN0	0			
					0000: Transmits no data
					0001: Transmits one byte of data
					0010: Transmits two bytes of data
					0011: Transmits three bytes of data
					0100: Transmits four bytes of data
					0101: Transmits five bytes of data
					0110: Transmits six bytes of data
					0111: Transmits seven bytes of data
					1000: Transmits eight bytes of data
					1001 to 1111: Setting prohibited
					If transmission of nine bytes or more is specified, data in FSITDR7 is transmitted.
3	—	0	R/W	—	Reserved
					The initial value should not be modified.

Bit	Bit Name	Initial Value	R/W EC	R/W Host	Description
2 to 0	RBN2	0	R/W	—	Receive Byte Count 2-0
	RBN1	0			These bits specify the number of data bytes to be received. After the FSI reception operation ends (when FSIRXI in FSISTR is 1), the RBN value is decremented (–1) each time FSIRDR is read. When all the data bytes have been received, RBN is cleared to B'000.
	RBN0	0			
					000: Receives no data
					001: Receives one byte of data
					010: Receives two bytes of data
					011: Receives three bytes of data
					100: Receives four bytes of data
					101 to 111: Setting prohibited
					If reception of five bytes or more is specified, FSIRDR is overwritten.

21.3.4 FSI Instruction Register (FSIINS)

FSIINS sets an instruction to be sent to the SPI flash memory during command transfer. When LFBUSY is 1, a write to this register by the EC (this LSI) is invalid. This register should not be set in the processing other than FSICMDI and FSIWI interrupt processing.

Bit	Bit Name	Initial Value	R/W EC	R/W Host	Description
7 to 0	bit 7 to bit 0	All 0	R/W	—	These bits store an instruction to be transmitted to the SPI flash memory.

21.3.5 FSI Instruction Register (FSIRDINS)

FSIRDINS sets a read operation instruction to be sent to FSITDR during read operation. When LFBUSY is set to 1, a write to this register by the EC (this LSI) is invalid. This register should be modified during initialization.

Bit	Bit Name	Initial Value	R/W		Description
			EC	Host	
7 to 0	bit 7 to bit 0	All 0	R/W	—	These bits store a read operation instruction.

21.3.6 FSI Program Instruction Register (FSIPPINS)

FSIPPINS sets a program operation instruction to be sent to FSITDR during program operation. When LFBUSY is set to 1, a write to this register by the EC (this LSI) is invalid. This register should be modified during initialization.

Bit	Bit Name	Initial Value	R/W		Description
			EC	Host	
7 to 0	bit 7 to bit 0	All 0	R/W	—	These bits store a program operation instruction.

21.3.7 FSI Status Register (FSISTR)

FSISTR indicates the processing status of the EC (this LSI) and the SPI flash memory transfer.

Bit	Bit Name	Initial Value	R/W		Description
			EC	Host	
7	FSITEI	0	R/(W)*	—	FSI Transmit End Interrupt Flag [Setting condition] When write data has been transmitted to the SPI flash memory. [Clearing condition] When this bit is read as 1 and then written with 0.

Bit	Bit Name	Initial Value	R/W EC	R/W Host	Description
6	OBF	0	R	—	Transmit Data Register Full Indicates whether or not there is data to be written by the EC (this LSI). 0: There is no write data. [Clearing condition] When write data transmission to the SPI flash memory is completed. 1: There is write data. [Setting condition] When the TE bit is set to 1.
5	FSIRXI	0	R	—	FSI Receive End Interrupt Flag Indicates whether or not there is data to be read by the EC (this LSI). 0: There is no read data. [Clearing condition] • LFBUSY = 0: When all receive data has been read by the EC (when RBN is cleared to 0). • LFBUSY = 1: When all receive data has been I/O-read by the host (automatically cleared). 1: There is read data. [Setting condition] When receive data has been transferred to FSIRDR.
4 to 0	—	All 0	R/W	—	Reserved The initial value should not be modified.

Note: * Only 0 can be written to bit 7 to clear it.

21.3.8 FSI Transmit Data Registers 0 to 7 (FSITDR0 to FSITDR7)

FSITDR stores a total of 8 bytes of transmit data. A total of 8 bytes of addresses, instructions, and data items can be transferred continuously from FSITDR0 through FSITDR7 in this order to the SPI flash memory. When LFBUSY is set to 1, a write to this register by the EC (this LSI) is invalid. This register should not be set in the processing other than FSICMDI and FSIWI interrupt processing.

Bit	Bit Name	Initial Value	R/W		Description
			EC	Host	
7 to 0	bit 7 to bit 0	All 0	R/W	—	These bits store transmit data.

21.3.9 FSI Receive Data Register (FSIRDR)

FSIRDR stores a total of 4 bytes of receive data items continuously sent from the SPI flash memory. This register should not be read in the processing other than FSICMDI interrupt processing. Note that four bytes of receive registers share a single register address. A register to be read will be determined according to the RBN bits in FSIBNR. When RBN = B'000, H'00 is read out.

Bit	Bit Name	Initial Value	R/W		Description
			EC	Host	
7 to 0	bit 7 to bit 0	All 0	R	—	These bits store receive data.

21.3.10 FSI Access Host Base Address Registers H and L (FSIHBARH and FSIHBARL)

FSIHBARH and FSIHBARL store the upper 16 bits of the host start address which is necessary to convert the host address to the SPI flash memory address. The input range of the host address will be determined based on the host start address set in these registers and the memory size set in FSISR. If a host address to be input is out of the determined range, Sync will not be returned. If FW memory cycle is used, bit 31 to bit 28 in FSIHBARH is set as IDSEL. During FSI operation (in the state where FSIE or FSILIE is set), do not change the setting in this register.

- FSIHBARH

Bit	Bit Name	Initial Value	R/W		Description
			EC	Host	
7 to 0	bit 31 to bit 24	All 0	R/W	—	These bits specify bits [31:24] of the host start address.

- FSIHBARL

Bit	Bit Name	Initial Value	R/W		Description
			EC	Host	
7 to 0	bit 23 to bit 16	All 0	R/W	—	These bits specify bits [23:16] of the host start address. The settings by bit 19 to bit 16 do not affect the operation.

21.3.11 FSI Flash Memory Size Register (FSISR)

FSISR sets the size of SPI flash memory. The host input address range will be determined based on the size set in this register. Note that the host input address should not be greater than the SPI flash memory capacity. During FSI operation (in the state where FSIE or FSILIE is set), do not change the setting in this register.

Bit	Bit Name	Initial Value	R/W EC	R/W Host	Description
7 to 2	—	All 0	R/W	—	Reserved The initial value should not be changed.
1	FSIMS1	0	R/W	—	These bits specify the SPI flash memory size.
0	FSIMS0	0	R/W	—	00: 1 MB 01: 2 MB 10: 4 MB 11: 8 MB

21.3.12 FSI Command Host Base Address Registers H and L (CMDHBARH and CMDHBARL)

CMDHBARH and CMDHBARL set the upper 16 bits of the host start address which is necessary to set a command address. The lower 16 bits of the host start address range from H'F000 to H'F00F. If a host address to be input to CMDHBARH and CMDHBARL is out of the determined range, Sync will not be returned. If FW memory cycle is used, bit 31 to bit 28 in CMDHBARH is set as IDSEL. During FSI operation (in the state where FSIE or FSILIE is set), do not change the setting in this register.

- CMDHBARH

Bit	Bit Name	Initial Value	R/W EC	R/W Host	Description
7 to 0	bit 31 to bit 24	All 0	R/W	—	These bits specify bits [31:24] of the host start address.

- CMDHBARL

Bit	Bit Name	Initial Value	R/W EC	R/W Host	Description
7 to 0	bit 23 to bit 16	All 0	R/W	—	These bits specify bits [23:16] of the host start address.

21.3.13 FSI Command Register (FSICMDR)

FSICMDR stores command data during FSI command reception. FSICMDR stores command data when the FSICMDI bit in FSILSTR1 is cleared to 0. It does not store command data when the FSICMDI bit is set to 1.

Bit	Bit Name	Initial Value	R/W EC	R/W Host	Description
7 to 0	bit 7 to bit 0	All 0	R	—	These bits store an FSI command.

21.3.14 FSI LPC Command Status Register 1 (FSILSTR1)

FSILSTR1 indicates the LPC internal status.

Bit	Bit Name	Initial Value	R/W EC	R/W Host	Description
7	CMDBUSY	0	R/W*	R	FSI Command Busy Flag 0: The FSI command execution is completed. [Clearing condition] • When this bit is read as 1 and then written with 0. 1: The FSI command execution is in progress. [Setting condition] • When an FSI command is received.
6	FSICMDI	0	R/W*	R	FSI Command Interrupt Flag 0:The FSI command interrupt processing is completed. [Clearing condition] • When this bit is read as 1 and then written with 0. 1: The FSI command interrupt processing is in progress. [Setting condition] • When an FSI command is received.

Bit	Bit Name	Initial Value	R/W EC	R/W Host	Description
5	FSIDMYE	0	R/W	R	FSI Dummy Enable 0: Disables FSI dummy. 1: Enables FSI dummy.
4	FSIWBUSY	0	R/W*	R	FSI Write Busy Flag 0: FSI write transfer is completed. [Clearing condition] • When this bit is read as 1 and then written with 0. 1: FSI write in transferring [Setting condition] • SPI flash memory write is received when FLDCT=0.
3	FSIWI	0	R/W*	R	FSI Write Interrupt Flag 0: FSI write interrupt is completed. [Clearing condition] • Read FSIWI=1 and then write 0. 1: FSI write interrupt is in progress. [Setting condition] • SPI flash memory write is received when FLDCT=0.
2	FLBUSY	0	R	R	LPC-SPI Direct Transfer Busy Flag Indicates an LPC-SPI direct transfer status. 0: Direct transfer is completed. 1: During direct transfer
1, 0	—	All 0	R/W	R	Reserved The initial value should not be modified.

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

21.3.15 FSI LPC Command Status Register 2 (FSILSTR2)

FSILSTR2 indicates the LPC internal status.

Bit	Bit Name	Initial Value	R/W		Description
			EC	Host	
7 to 5	—	All 0	R/W	—	Reserved The initial value should not be modified.
4	FSIDWBUSY	0	R	—	FSI Direct Write Busy Flag Indicates a FSI write transfer status during LPC-SPI direct transfer. 0: FSI write transfer is completed. 1: During FSI write transfer
3	FSIDRBUSY	0	R	—	FSI Direct Read Busy Flag Indicates a FSI read transfer status during LPC-SPI direct transfer. 0: FSI read transfer is completed. 1: During FSI read transfer
2 to 0	SIZE2	0	R	—	Transfer Byte Count Monitor Indicates the number of transferred bytes when data is received in the LPC/FW memory cycles. When the Byte/Page-Program or AAI-Program instruction is executed from the EC CPU, the number of transferred bytes can be confirmed by these bits. 001: LPC/FW memory cycle (byte transfer) 010: FW memory cycle (word transfer) 100: FW memory cycle transfer (longword transfer) When a transfer is made in units other than byte/word/longword, the previous value is retained. Note: This bit is not set to the value other than above.
	SIZE1	0	R	—	
	SIZE0	1	R	—	

21.3.16 FSI General-Purpose Registers 1 to F (FSIGPR1 to FSIGPRF)

FSIGPR1 to FSIGPRF store data such as the result of FSI command interrupt processing.

- FSIGPR1 to FSIGPRF

Bit	Bit Name	Initial Value	R/W		Description
			EC	Host	
7 to 0	bit 7 to bit 0	All 0	R/W	R	These bits store results of FSI command interrupt processing.

21.3.17 FSI LPC Control Register (SLCR)

SLCR enables or disables the LPC host interface function of the FSI, FSI interrupt enable bit, and FSI operation mode control bit.

Bit	Bit Name	Initial Value	R/W		Description
			EC	Host	
7	FSILIE	0	R/W	—	FSI LPC Interface Enable Enables or disables the LPC host interface function of the FSI. When disabled, address-matching is not performed and Sync is not returned. 0: Disables the LPC host interface function. 1: Enables the LPC host interface function.
6	FSICMDIE	0	R/W	—	FSI Command Interrupt Enable 0: Disables the FSI command interrupt. 1: Enables the FSI command interrupt.
5	FSIWIE	0	R/W	—	FSI Write Interrupt Enable 0: Disables the FSI write interrupt. 1: Enables the FSI write interrupt.

Bit	Bit Name	Initial Value	R/W EC	R/W Host	Description
4	FLDCT	0	R/W	—	FSI LPC Direct Selects access mode in SPI flash memory write. For details, see section 21.4.6, SPI Flash Memory Write Operation Mode. 0: LPC-SPI indirect transfer 1: LPC-SPI direct transfer
3	FLWAIT	0	R/W	—	FSI LPC Wait Selects access mode in SPI flash memory write. For details, see section 21.4.6, SPI Flash Memory Write Operation Mode. 0: No wait cycle is inserted. 1: Wait cycles can be inserted.
2 to 0	—	All 0	R/W	—	Reserved The initial value should not be modified.

21.3.18 FSI Address Registers H, M, and L (FSIARH, FSIARM, and FSIARL)

FSIAR stores an SPI flash memory address. If the host address matches FSIHBAR, the FSIAR value is updated. FSIAR value is not updated by command access.

- FSIARH

Bit	Bit Name	Initial Value	R/W EC	R/W Host	Description
7 to 0	bit 23 to bit 16	All 0	R	—	These bits store bits [23:16] of the SPI flash memory address.

- FSIARM

Bit	Bit Name	Initial Value	R/W EC	R/W Host	Description
7 to 0	bit 15 to bit 8	All 0	R	—	These bits store bits [15:8] of the SPI flash memory address.

- FSIARL

Bit	Bit Name	Initial Value	R/W EC	R/W Host	Description
7 to 0	bit 7 to bit 0	All 0	R	—	These bits store bits [7:0] of the SPI flash memory address.

21.3.19 FSI Write Data Registers HH, HL, LH, and LL (FSIWDRHH, FSIWDRHL, FSIWDRLH, and FSIWDRLL)

FSIWDR stores data to be written to the SPI flash memory. If the host address matches FSIHBAR during LPC/FW memory write cycle, the FSIWDR value will be updated. FSIHBAR value is not updated by command access.

- FSIWDRHH

Bit	Bit Name	Initial Value	R/W EC	R/W Host	Description
7 to 0	bit 31 to bit 24	All 0	R	—	These bits store bits [31:24] of the SPI flash memory write data

- FSIWDRHL

Bit	Bit Name	Initial Value	R/W EC	R/W Host	Description
7 to 0	bit 23 to bit 16	All 0	R	—	These bits store bits [23:16] of the SPI flash memory write data.

- FSIWDRLH

Bit	Bit Name	Initial Value	R/W EC	R/W Host	Description
7 to 0	bit 15 to bit 8	All 0	R	—	These bits store bits [15:8] of the SPI flash memory write data.

- FSIWDRLL

Bit	Bit Name	Initial Value	R/W EC	R/W Host	Description
7 to 0	bit 7 to bit 0	All 0	R	—	These bits store bits [7:0] of the SPI flash memory write data.

21.4 Operation

21.4.1 LPC/FW Memory Cycles

In LPC/FW memory read and write cycles, data is transferred using LAD3 to LAD0 synchronously with LCLK. The order of data transfer is shown in table 21.4. In a cycle returning synchronization signal from the slave, the slave usually returns B'1010 to notify the host of error occurrence; while the FSI in this LSI always returns B'0000 (Ready) or B'0110 (Long wait).

The FSI becomes busy if the received address matches an address in the host accessible range set in the registers (FSIHBARH, FSIHBARL, FSISR, and CMDHBAR), and outputs a turn-around signal to return to the idle state.

Table 21.4 LPC Memory Read/Write Cycles

State Counts	LPC Memory Read Cycles			LPC Memory Write Cycles		
	Content	Driven by	Value (3 to 0)	Content	Driven by	Value (3 to 0)
1	Start	Host	0000	Start	Host	0000
2	Cycle type/ direction	Host	0100	Cycle type/ direction	Host	0110
3	Address 1	Host	bit 31 to bit 28	Address 1	Host	bit 31 to bit 28
4	Address 2	Host	bit 27 to bit 24	Address 2	Host	bit 27 to bit 24
5	Address 3	Host	bit 23 to bit 20	Address 3	Host	bit 23 to bit 20
6	Address 4	Host	bit 19 to bit 16	Address 4	Host	bit 19 to bit 16
7	Address 5	Host	bit 15 to bit 12	Address 5	Host	bit 15 to bit 12
8	Address 6	Host	bit 11 to bit 8	Address 6	Host	bit 11 to bit 8
9	Address 7	Host	bit 7 to bit 4	Address 7	Host	bit 7 to bit 4
10	Address 8	Host	bit 3 to bit 0	Address 8	Host	bit 3 to bit 0
11	Turn-around (recovery)	Host	1111	Data 1	Host	bit 3 to bit 0
12	Turn-around	None	ZZZZ	Data 2	Host	bit 7 to bit 4
13	Wait*	Slave	0110	Turn-around (recovery)	Host	1111

State Counts	LPC Memory Read Cycles			LPC Memory Write Cycles		
	Content	Driven by	Value (3 to 0)	Content	Driven by	Value (3 to 0)
14	Synchronization	Slave	0000	Turn-around	None	ZZZZ
15	Data 1	Slave	bit 3 to bit 0	Wait*	Slave	0110
16	Data 2	Slave	bit 7 to bit 4	Synchronization	Slave	0000
17	Turn-around (recovery)	Slave	1111	Turn-around (recovery)	Slave	1111
18	Turn-around	None	ZZZZ	Turn-around	None	ZZZZ

Note: * The number of wait cycles depends on the system.

Table 21.5 FW Memory Read/Write Cycles (Byte Transfer)

State Counts	FW Memory Read Cycles			FW Memory Write Cycles		
	Content	Driven by	Value (3 to 0)	Content	Driven by	Value (3 to 0)
1	Start	Host	1101	Start	Host	1110
2	Device select	Host	ID3 to ID0	Device select	Host	ID3 to ID0
3	Address 1	Host	bit 27 to bit 24	Address 1	Host	bit 27 to bit 24
4	Address 2	Host	bit 23 to bit 20	Address 2	Host	bit 23 to bit 20
5	Address 3	Host	bit 19 to bit 16	Address 3	Host	bit 19 to bit 16
6	Address 4	Host	bit 15 to bit 12	Address 4	Host	bit 15 to bit 12
7	Address 5	Host	bit 11 to bit 8	Address 5	Host	bit 11 to bit 8
8	Address 6	Host	bit 7 to bit 4	Address 6	Host	bit 7 to bit 4
9	Address 7	Host	bit 3 to bit 0	Address 7	Host	bit 3 to bit 0
10	Size	Host	0000	Size	Host	0000
11	Turn-around (recovery)	Host	1111	Data 1	Host	bit 3 to bit 0
12	Turn-around	None	ZZZZ	Data 2	Host	bit 7 to bit 4
13	Wait*	Slave	0110	Turn-around (recovery)	Host	1111
14	Synchronization	Slave	0000	Turn-around	None	ZZZZ
15	Data 1	Slave	bit 3 to bit 0	Wait*	Slave	0110
16	Data 2	Slave	bit 7 to bit 4	Synchronization	Slave	0000

State Counts	FW Memory Read Cycles			FW Memory Write Cycles		
	Content	Driven by	Value (3 to 0)	Content	Driven by	Value (3 to 0)
17	Turn-around (recovery)	Slave	1111	Turn-around (recovery)	Slave	1111
18	Turn-around	None	ZZZZ	Turn-around	None	ZZZZ

Note: * The number of wait cycles depends on the system clock.

The FSI supports byte, word, and longword transfers of FW memory read and write cycles. In word transfer, the least address bit is fixed to B'0; while in longword transfer, the lower 2 bits are fixed to B'00. When longword transfers of FW memory write cycles are used, the maximum operating frequency of the system clock is 10 MHz.

21.4.2 SPI Flash Memory Transfer

The SPI flash memory transfer is performed using FSIDO and FSIDI synchronously with FSICK. The initial value of FSICK can be either fixed to high or low through programming.

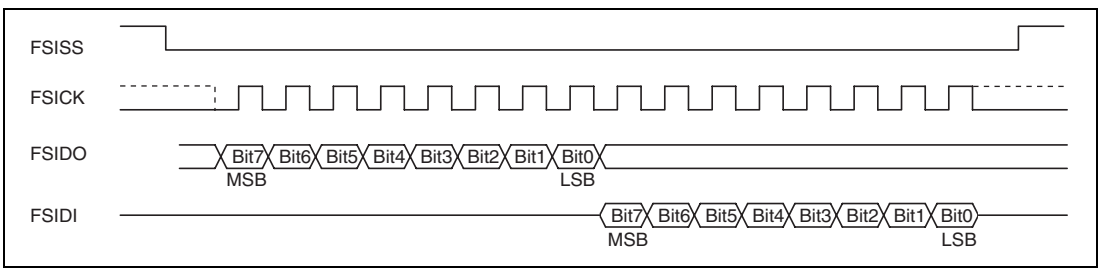


Figure 21.2 Example of SPI Flash Memory Transfer

21.4.3 Flash Memory Instructions

Table 21.6 lists the flash memory instructions (INS).

Table 21.6 List of Instructions (INS)

Instruction Name	Description
WREN	Sets write-enable
WRDI	Resets write-enable
RDSR	Reads status register
WRSR	Writes status register
READ	Reads SPI flash memory
Fast-Read	Fast-reads SPI flash memory
Byte-Program	Byte-programs SPI flash memory
Page-Program	Page-programs SPI flash memory
AAI-Program	Address auto increment program
Sector-Erase	Sector erasure
Block-Erase	Block erasure
Chip/Bulk-Erase	Chip/bulk erasure
RDID	Reads manufacturing ID and product ID
EWSR	Enables status register write
DP (DEEP POWER DOWN)	Deep power-down
RES	Releases deep power-down

21.4.4 FSI Memory Cycle (Direct Transfer between LPC and SPI)

The FSI supports direct transfer between the host and SPI flash memory. If the host address input in LPC/FW memory write cycle matches the host address set in FSIHBARH, FSIHBARL, or FSISR, the FSI memory cycle starts. In LPC/FW memory write cycle, the FSI supports three types of instructions: Byte/Page-Program instructions and AAI-Program instruction. In LPC/FW memory read cycle, the FSI supports two types of instructions: Read instruction and Fast-Read instruction. In the case that LPC-SPI direct transfer is selected in Byte-Program, Page-Program, or AAI-Program instruction execution, set FLDCT of SLCR to 1. The FSI reads the data with LPC-SPI direct transfer regardless of the status of FLDCT in Read and Fast-Read instruction execution.

(1) FSI Address Conversion

The host address can be converted into the SPI flash memory address by setting FSIHBARH, FSIHBARL, and FSISR. The host address space ranges from H'0000_0000 to H'FFFF_FFFF. The SPI flash memory address space ranges from H'00_0000 to H'FF_FFFF. Figure 21.3 shows an example of the FSI memory address conversion.

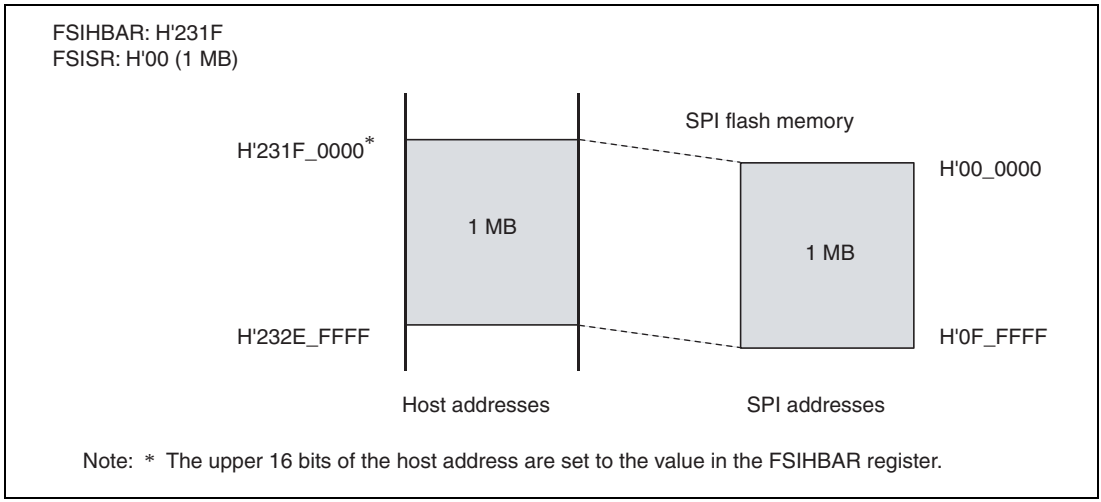


Figure 21.3 FSI Address Conversion Example

As shown in figure 21.3, if an address ranging from H'231F_0000 to H'232E_FFFF is accessed in LPC/FW memory write cycle, the SPI flash memory is accessed. If a host address to be input is out of the determined range, Sync will not be returned. During an SPI flash memory access, a long wait cycle will be inserted to the LPC bus cycle. In an LPC memory cycle, one-byte transfer is enabled. In an FW memory cycle, a byte, word, and a longword transfer are enabled.

(2) Byte/Page-Program Instruction

If an LPC/FW memory write cycle occurs while the AAIE bit in FSICR1 and the FSIDMYE bit in FSILSTR1 are cleared to 0, and the FLDCT bit in SLCR and the FLWAIT bit in SLCR are set to 1, the SPI flash memory address and write data are stored in FSIAR and FSIWDR, respectively. Then, the SPI flash memory address, the write data, and the Byte/Page-Program instruction which is stored in FSIPPINS in advance are transferred to FSITDR. After SYNC (long wait) has been returned, the TE bit in FSICR2 is set, starting the Byte/Page-Program instruction execution. When the transmission has been completed, SYNC (Ready) and TAR are returned to the host. To execute the Byte-Program instruction, byte transfer access in LPC memory write cycle or FW memory write cycle should be performed. Figure 21.4 shows an example of data transfer to FSITDR. Figure 21.5 shows the Page-Program instruction execution timing.

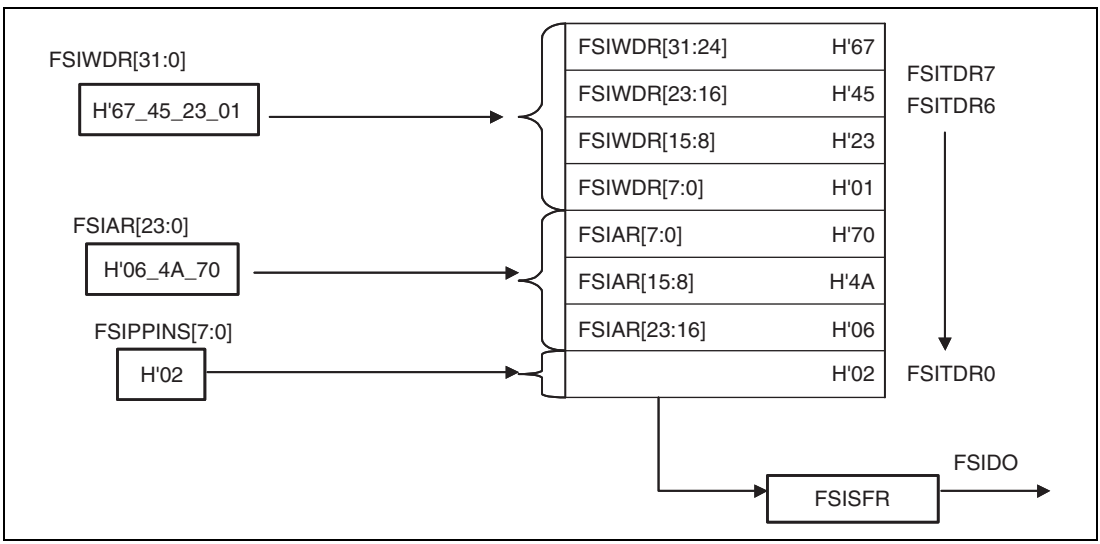


Figure 21.4 Data Transfer to FSITDR (Example)

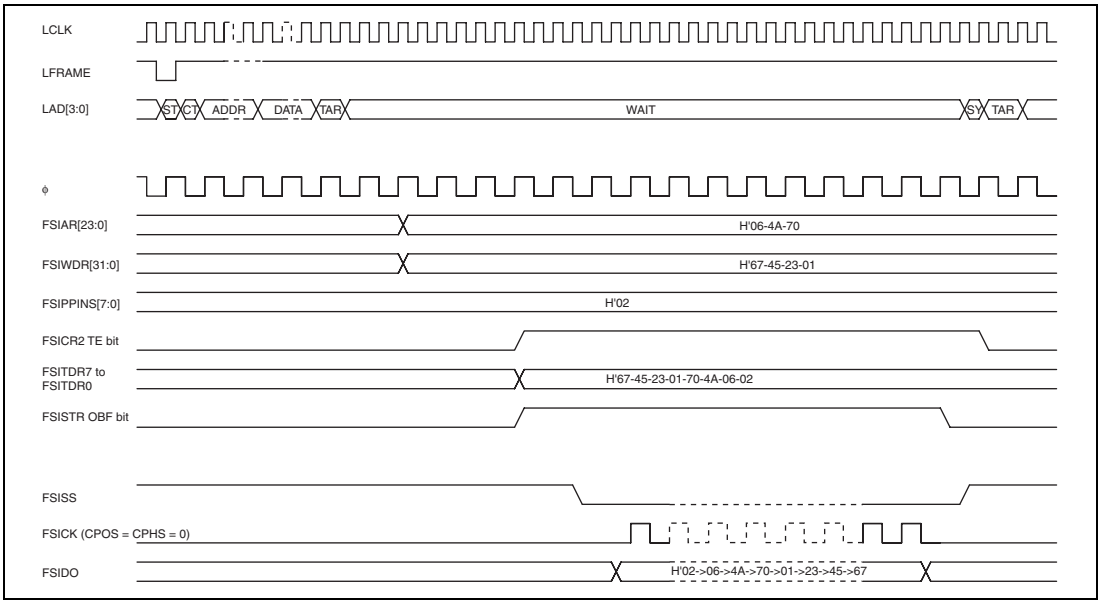


Figure 21.5 Page-Program Instruction Execution Timing

(3) AAI-Program Instruction

If an LPC/FW memory write cycle occurs while the AAIE bit in FSICR1 is set to 1 and the FSIDMYE bit in FSILSTR1 is cleared to 0, and the FLDCT bit in SLCR and the FLWAIT bit in SLCR are set to 1, the flash memory address and write data are stored in FSIAR and FSIWDR, respectively. Then, the flash memory address, write data, and the AAI-Program instruction which is stored in FSI hardware in advance are transferred to FSITDR. After SYNC (long wait) has been returned, the transmit enable signal TE is set, and AAI-Program instruction execution starts. In the first byte, the instruction, address, and data in this order are transmitted to the SPI flash memory. In the second and the following bytes, an instruction and data in this order are transmitted to the SPI flash memory. When the transmission has been completed, SYNC (Ready) and TAR are returned to the host. To execute the AAI-Program instruction, byte transfer access in LPC memory write cycle or FW memory write cycle should be performed. To return to the AAI-Program instruction (first byte), clear the AAIE bit once or perform initialization of the FSI internal sequencer in SRES of FSICR1. After the Read instruction or the LPC-SPI command is transferred during the AAI-Program instruction execution, the FSI internal sequencer is initialized to return to the AAI-Program Instruction (first byte). Figures 21.6 and 21.7 show AAI-Program execution timings.

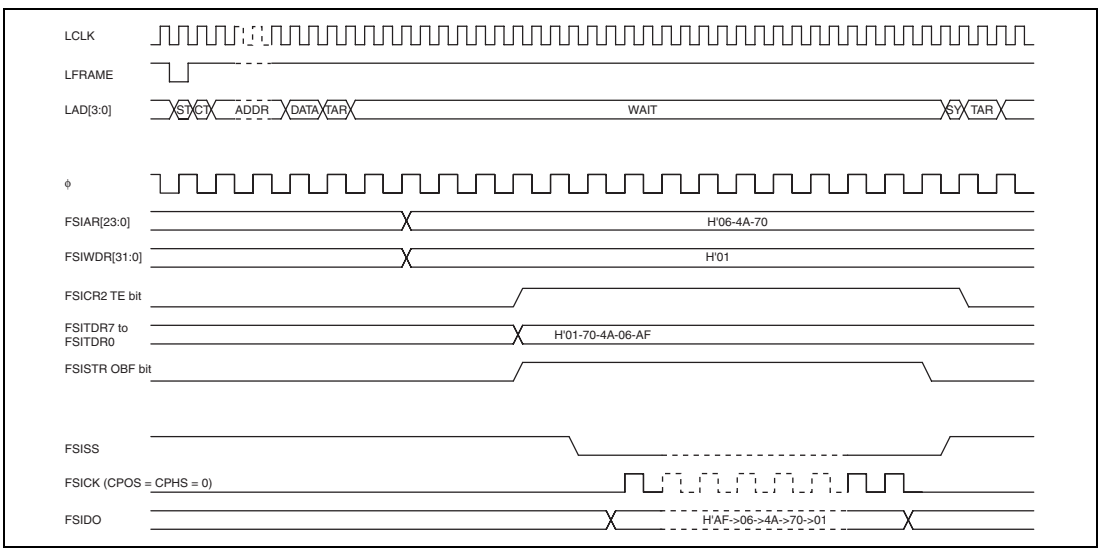


Figure 21.6 AAI-Program Instruction Execution Timing (First Byte)

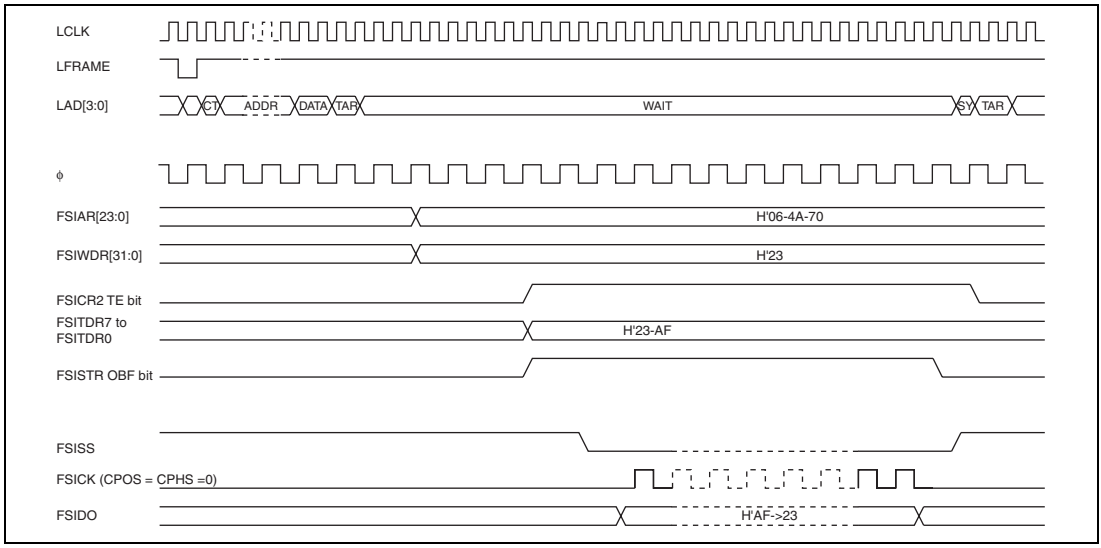


Figure 21.7 AAI-Program Instruction Execution Timing (Second and Following Bytes)

(4) Read Instructions

If an LPC/FW memory read cycle occurs while the FRDE bit in FSICR1 is cleared to 0, the SPI flash memory address is stored in FSIAR. Then, the SPI flash memory address and the instruction which is stored in FSIRDINS in advance are transferred to FSITDR. After SYNC (long wait) has been returned, the RE bit in FSICR2 is set, and Read instruction execution starts. The read data is then received and stored in FSIRDR. When the reception has been completed, SYNC (Ready), read data, and TAR are returned to the host. Figure 21.8 shows an example of data transfer to FSIRDR. Figure 21.9 shows the Read instruction execution timing.

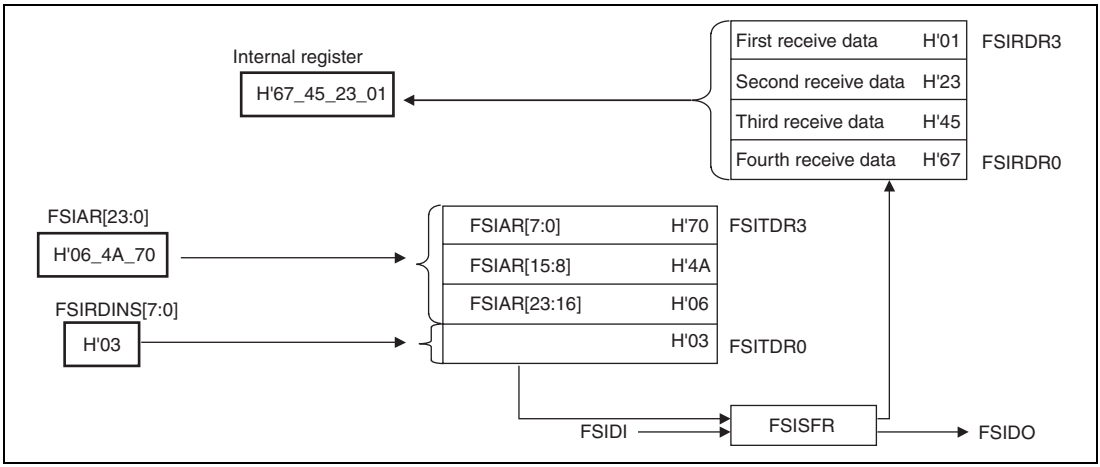


Figure 21.8 Data Transfer to FSIRDR (Example)

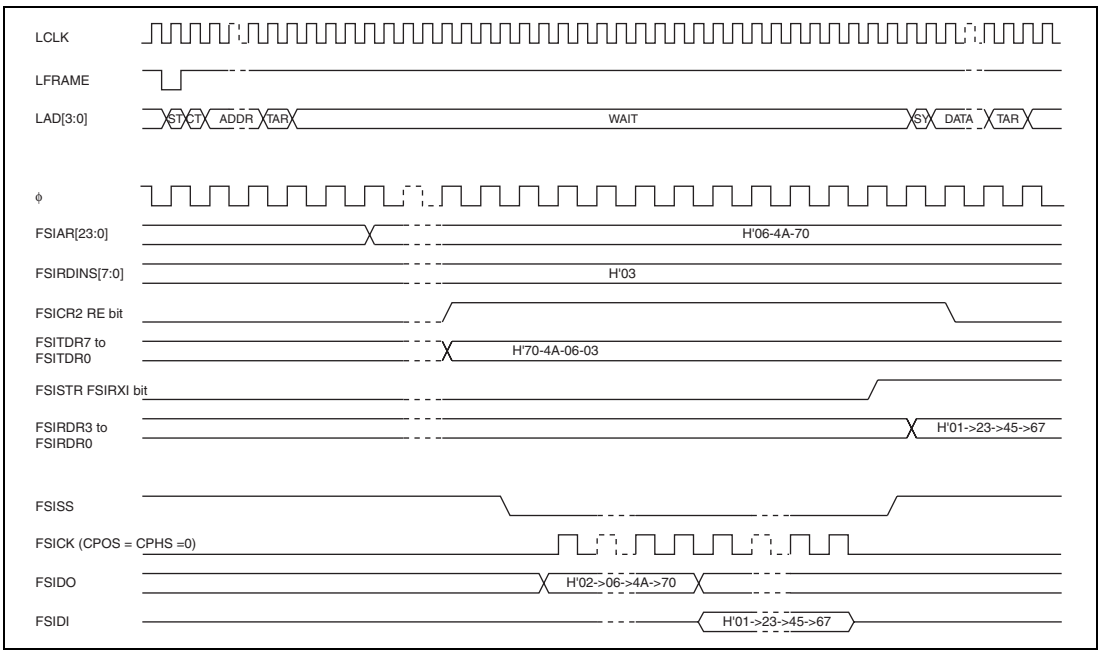


Figure 21.9 Read Instruction Execution Timing

(5) Fast-Read Instruction

If an LPC/FW memory read cycle occurs while the FRDE bit in FSICR1 is set to 1, the host address is stored in FSIAR. Then, the SPI flash memory address and the instruction which is stored in FSIRDINS in advance are transferred to FSITDR. After SYNC (long wait) has been returned, the RE bit in FSICR2 is set, and Fast-Read instruction execution starts. The read data is then received and stored in FSIRDR. When the reception has been completed, SYNC (Ready), read data, and TAR are returned to the host. Figure 21.10 shows the Fast-Read Instruction Execution Timing.

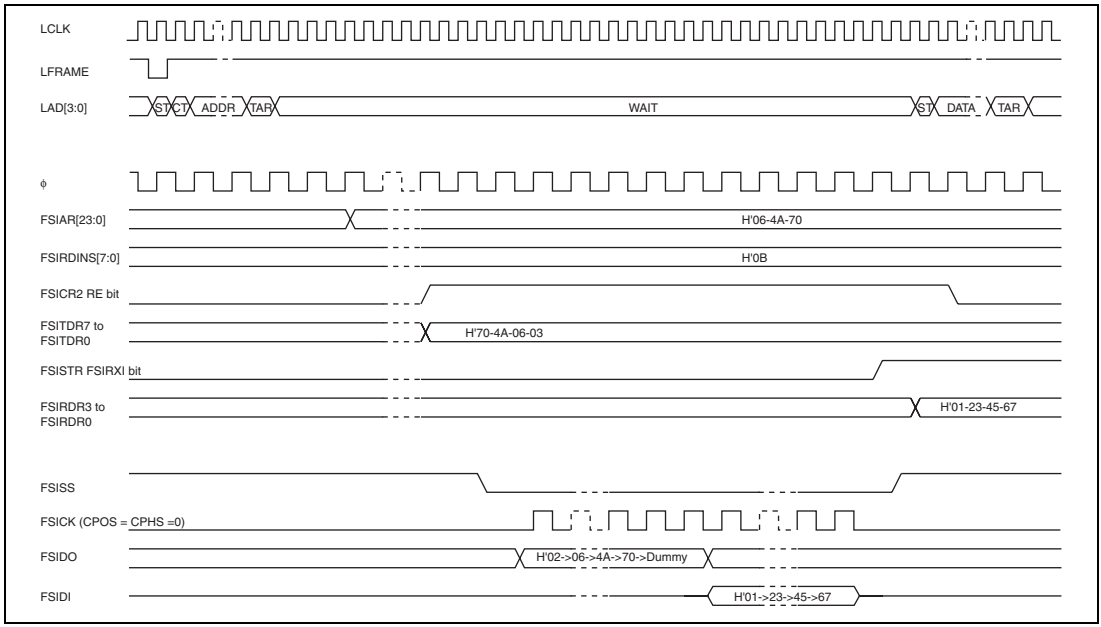


Figure 21.10 Fast-Read Instruction Execution Timing

21.4.5 FSI Memory Cycle (LPC-SPI Command Transfer)

The FSI supports instructions other than Byte/Page-Program instructions, AAI-Program instruction, Read instruction, and Fast-Read instruction by using an LPC-SPI command transfer.

(1) FSI Command Space

Specific host address space can be used as FSI command space according to the CMDHBAR settings. Figure 21.11 shows an example of FSI command space settings.

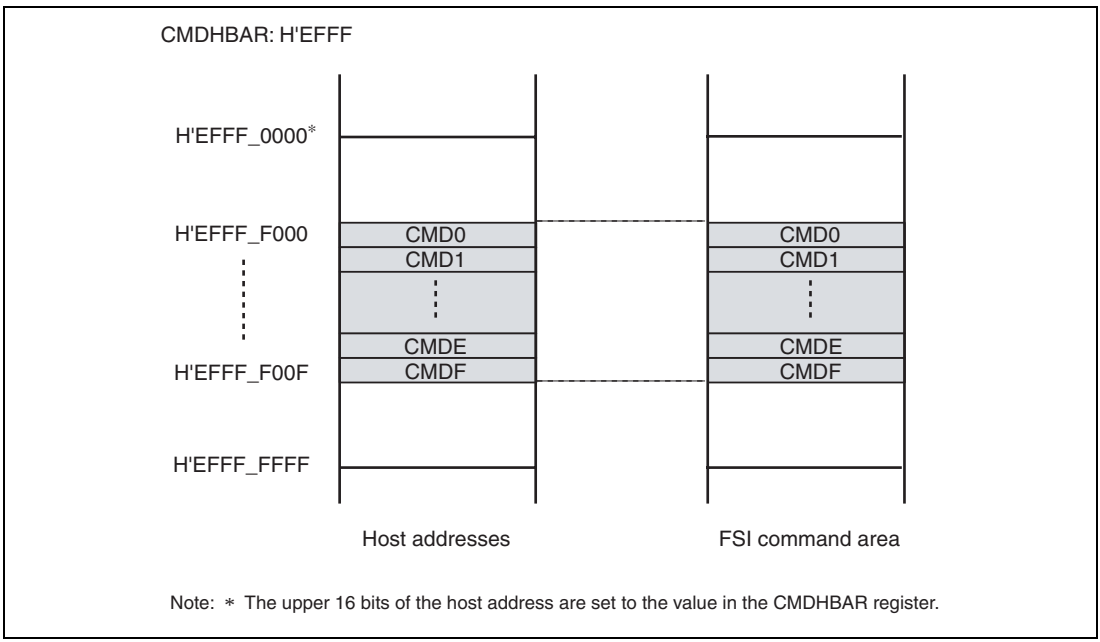


Figure 21.11 FSI Command Space Settings (Example)

As shown in figure 21.11, a host address ranging from H'EFFF_F000 to H'EFFF_F00F is used as the FSI command space while the CMDHBAR register is set to H'EFFF.

(2) FSI Command Write

If an LPC/FW memory write cycle for the FSI command space occurs, the FSI performs the FSI-FLASH command write operation. Figure 21.12 shows an example of FSI Command write operation.

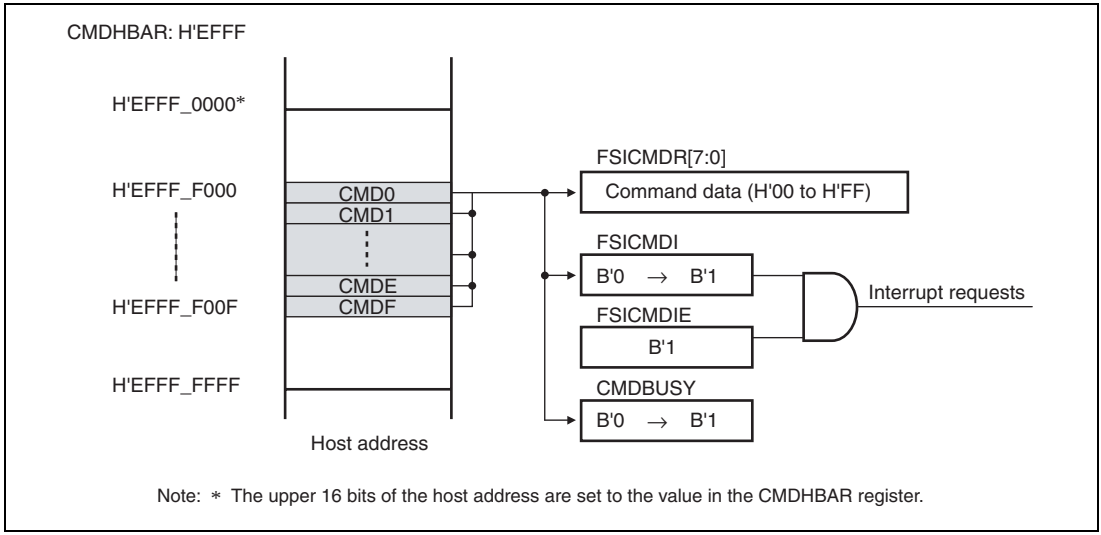
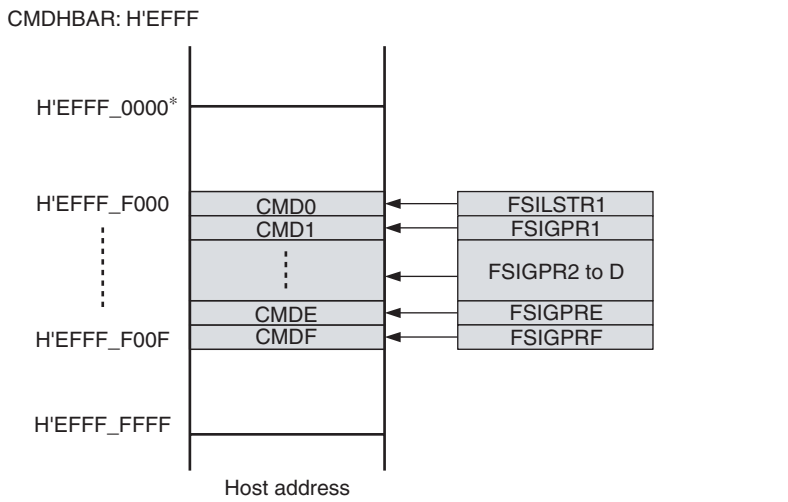


Figure 21.12 FSI Command Write Operation (Example)

As shown in figure 21.12, if a host address ranging from H'EFFF_F000 to H'EFFF_F00F is accessed in LPC/FW memory write cycle while the CMDHBAR register is set to H'EFFF, the write data is stored in FSICMDR, and then the CMDBUSY and FSICMDI flags in FSILSTR1 are set to 1. In this case, an interrupt is requested according to the FSICMDIE state. Sync is not returned if the host address to be input is out of the determined range. In FSI command write, no wait cycle will be inserted to the LPC bus cycle. If the CMDBUSY flag is set to 1, Sync is not returned during the operations other than FSI command read.

(3) FSI Command Read

Figure 21.13 shows an example of FSI command read.



Note: * The upper 16 bits of the host address are set to the value in the CMDHBAR register.

Figure 21.13 FSI Command Read (Example)

As shown in figure 21.13, if a host address ranging from H'EFFF_F000 to H'EFFF_F00F is accessed in LPC/FW memory read cycle while the CMDHBAR register is set to H'EFFF, the FSILSTR1 or data in FSIGPR1 to FSIGPRF is returned. Sync is not returned if the host address to be input is out of the determined range. In FSI command read, no wait cycle will be inserted to the LPC bus cycle. Before reading the FSIGPR, ensure that the CMDBUSY bit in FSILSTR1 has been cleared to 0.

(4) FSI Dummy Write

Figure 21.14 shows an example of FSI dummy write.

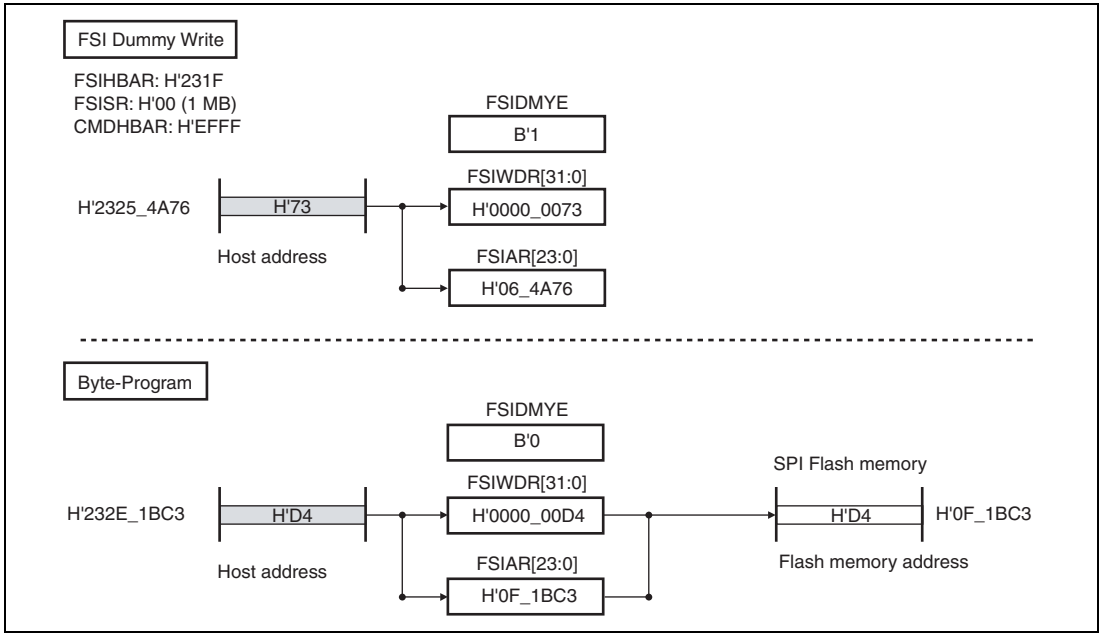


Figure 21.14 FSI Dummy Write (Example)

As shown in figure 21.14, if an LPC/FW memory write cycle occurs while the FSIDMYE bit in FSILSTR1 is 1, the FSI does not access the SPI flash memory but stores the SPI flash memory address and write data in FSIAR and FSIWDR, respectively.

(5) FSI Command Usage Example 1 (SPI Flash Memory Erasure)

The FSI commands enable the execution of several instructions for the SPI flash memory. Figure 21.15 shows an example of executing the SPI flash memory erasure instruction.

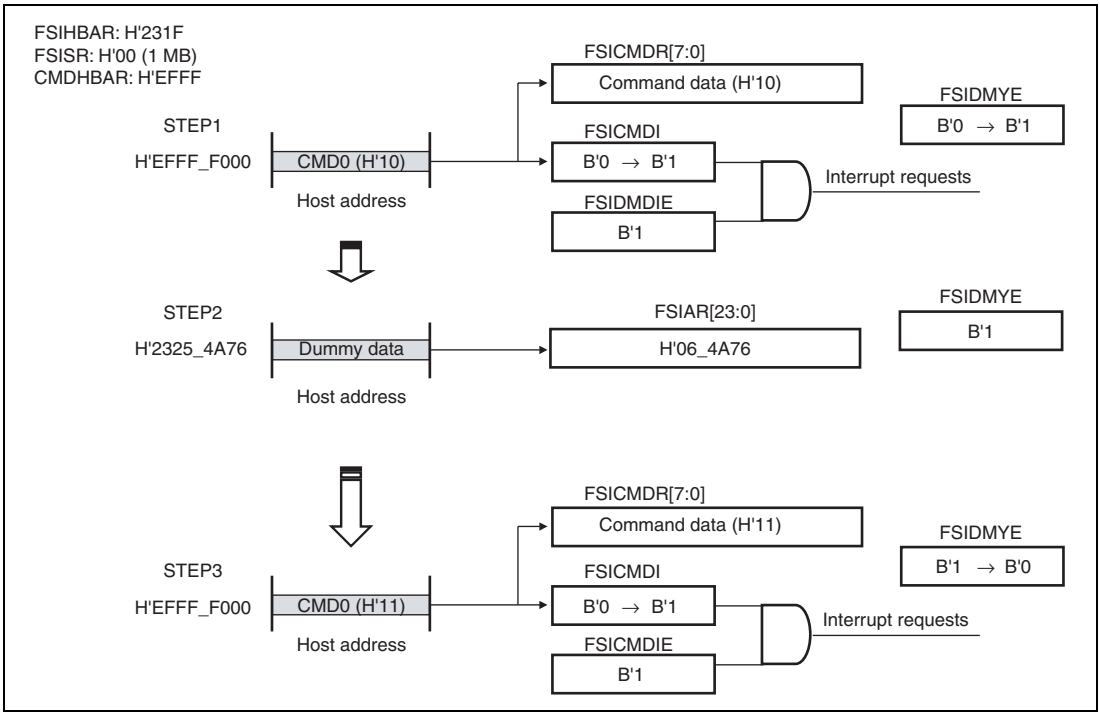


Figure 21.15 SPI Flash Memory Erasure (Example)

In flash memory erasure, the SPI flash memory address is stored in FSIAR and an erasure instruction for the SPI flash memory is executed by an SPI command. The flash memory address storage in FSIAR is performed by writing data to the sector or block address to be erased via the host. To distinguish the SPI flash memory erasure from the SPI flash memory programming, the erasure is performed in the following sequence using the FSIDMYE.

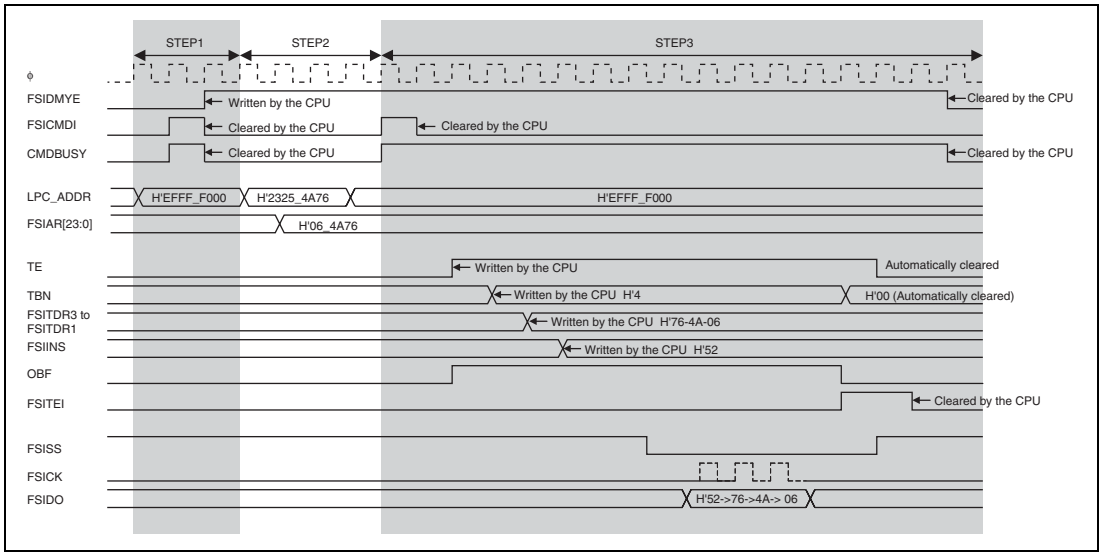


Figure 21.16 Execution Timing of SPI Flash Memory

Step 1:

1. Write an erasure setting command (Host).
2. Generate an FSICMDI interrupt request.
3. Set the FSIDMYE bit in FSILSTR1 to 1 and clear the FSICMDI and CMDBUSY bits in FSILSTR1 to 0.
4. Complete the interrupt processing.
5. Check that the FSIDMYE bit in FSILSTR1 is set to 1 and that the CMDBUSY and FSICMDI bits in FSILSTR1 are cleared to 0 (Host).

Step 2:

1. Perform a dummy write to the sector or block address to be erased (Host).
2. Store the SPI flash memory address and write data in the FSIAR register and FSIWDR register, respectively*.

Note: * Use the data stored in FSIWDR if necessary on the user side.

Step 3:

1. Write an erasure setting command (Host).
2. Generate an FSICMDI interrupt request.
3. Clear the FSICMDI bit in FSILSTR1 to 0.

4. Execute the SPI flash memory erasure instruction.
   — Set the TE bit in FSICR2 to 1.
   — Set the TBN bit in FSIBNR to 4-byte transfer.
   — Write the FSI address stored in FSIAR to FSITDR1 to FSITDR3.
   — Write the erasure instruction to FSIINS (start the SPI flash memory erasure instruction execution).
5. Complete the interrupt processing.
6. Generate an FSITEI interrupt request.
7. Clear the FSIDMYE and CMDBUSY bits in FSILSTR1 to 0.
8. Complete the interrupt processing.
9. Check that the FSIDMYE, CMDBUSY, and FSICMDI bits in FSILSTR1 are cleared to 0 (Host).

(6) FSI Command Usage Example 2 (SPI Flash Memory Status Read)

Figure 21.17 shows an example of the execution timing of the SPI flash memory status read instruction.

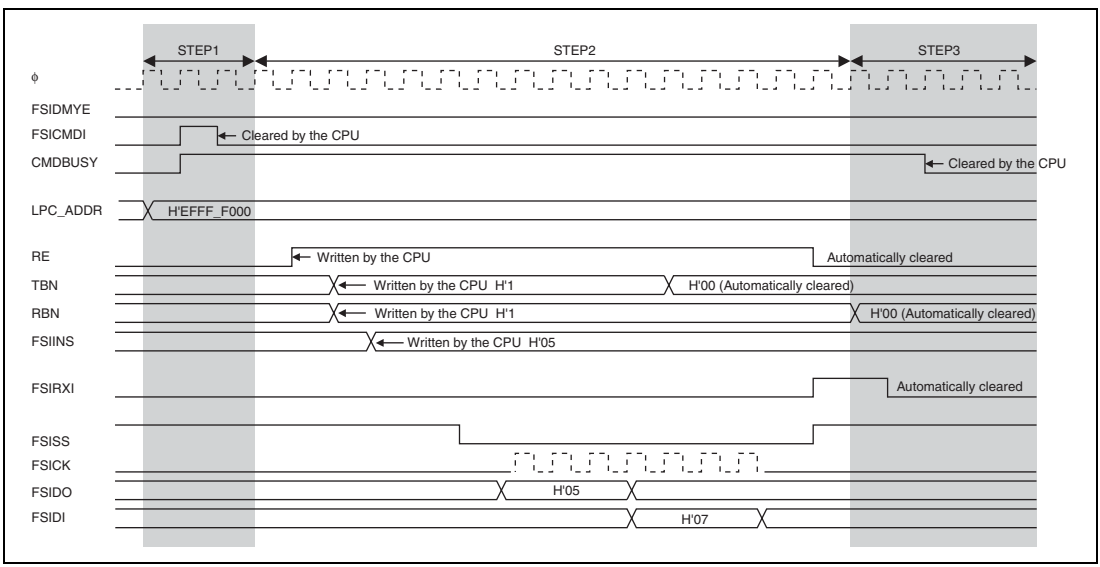


Figure 21.17 Execution Timing of SPI Flash Memory Status Read Instruction

The SPI flash memory status read instruction is executed in the following sequence.

Step 1:

1. Write a status read setting command (Host).
2. Generate an FSICMDI interrupt request.
3. Clear the FSICMDI bit in FSILSTR1 to 0.
4. Check that the CMDBUSY bit in FSILSTR1 is set to 1 and that the FSICMDI bit in FSILSTR1 is cleared to 0 (Host).

Step 2:

1. Perform the SPI flash memory status read instruction.
   - Set the RE bit in FSICR2 to 1.
   - Set the TBN bit in FSIBNR to 1-byte transfer and set the RBN bit in FSIBNR to 1-byte reception.
   - Write the status read instruction to FSIINS (start the SPI flash memory status read instruction execution).
2. Complete the interrupt processing.

Step 3:

1. Generate an FSIRXI interrupt request.
2. Write read data stored in FSIRDR to SPIGPR.
3. Clear the CMDBUSY bit in FSILSTR1 to 0.
4. Complete the interrupt processing
5. Check that the CMDBUSY and FSICMDI bits in FSILSTR1 are cleared to 0 (Host).
6. Read the SPI flash memory status from FSIGPR (Host).

21.4.6 SPI Flash Memory Write Operation Mode

The write operation to the SPI flash memory in the LPC/FW memory write cycles can be classified into the following four operation modes, depending or the state of FLDCT and FLWAIT.

Table 21.7 SPI Flash Memory Write Operation in LPC/FW Memory Write Cycles

Operation Mode	FLDCT	FLWAIT	Selected Register	Operation
Mode 1	0	0	FSIWBUSY ← 1 FSIWI ← 1	Control the write operation to the SPI flash memory by the EC CPU. No wait cycle is inserted to the LPC bus. Confirm by FSIWBUSY whether or not a write transfer has been completed.
Mode 2	0	1	FSIWBUSY ← 1 FSIWI ← 1	Control the write operation to the SPI flash memory by the EC CPU. Wait cycles are inserted to the LPC bus. Provision of wait cycles can be canceled by clearing FSIWBUSY.
Mode 3	1	0	LFBUSY ← 1 (Automatically cleared)	Control the write operation to the SPI flash memory by the FSI. No wait cycle is inserted to the LPC bus. Confirm by LFBUSY whether or not a write transfer has been completed.
Mode 4	1	1	LFBUSY ← 1 (Automatically cleared)	Control the write operation to the SPI flash memory by the FSI. Wait cycles are inserted to the LPC bus. Provision of wait cycles can be canceled by clearing LFBUSY.

21.5 Reset Conditions

The FSI supports the LPC shut-down mode. The range of initialization in each mode is shown in table 21.8.

Table 21.8 Range of Initialization of FSI in Each Mode

Register Name		System Reset	LPC Reset	LPC Shutdown	LPC Abort	FSI Reset
FSIHBARH/ FSIHBARL	Bits 7 to 0	Initialized	Retained	Retained	Retained	Retained
FSISR	Bits 7 to 0	Initialized	Retained	Retained	Retained	Retained
CMDHBARH/ CMDHBARL	Bits 7 to 0	Initialized	Retained	Retained	Retained	Retained
FSICMDR	Bits 7 to 0	Initialized	Retained	Retained	Retained	Retained
FSILSTR1	Bits 7, 6, 4, and 3	Initialized	Initialized	Retained	Retained	Initialized
	Bits 5 and 2 to 0	Initialized	Retained	Retained	Retained	Retained
FSILSTR2	Bits 7 to 5	Initialized	Retained	Retained	Retained	Retained
	Bits 4 and 3	Initialized	Initialized	Retained	Retained	Initialized
	Bits 2 to 0	Initialized	Retained	Retained	Retained	Retained
SPIGPR1 to SPIGPRF	Bits 7 to 0	Initialized	Retained	Retained	Retained	Retained
SLCR	Bits 7 to 0	Initialized	Retained	Retained	Retained	Retained
FSIARH/ FSIARM/ FSIARL	Bits 7 to 0	Initialized	Retained	Retained	Retained	Retained
FSIWDRHH/ FSIWDRHL/ FSIWDRLH/ FSIWDRLL	Bits 7 to 0	Initialized	Retained	Retained	Retained	Retained
LPC internal sequencer		Initialized	Initialized	Initialized	Initialized	Retained

Register Name		System Reset	LPC Reset	LPC Shutdown	LPC Abort	FSI Reset
FSICR1	Bits 7 to 0	Initialized	Retained	Retained	Retained	Retained
FSICR2	Bits 7 and 6	Initialized	Retained	Retained	Retained	Initialized
	Bits 5 to 0	Initialized	Retained	Retained	Retained	Retained
FSIBNR	Bits 7 to 4	Initialized	Retained	Retained	Retained	Initialized
	Bit 3	Initialized	Retained	Retained	Retained	Retained
	Bits 2 to 0	Initialized	Retained	Retained	Retained	Initialized
FSIINS	Bits 7 to 0	Initialized	Retained	Retained	Retained	Retained
FSIRDINS	Bits 7 to 0	Initialized	Retained	Retained	Retained	Retained
FSIPPINS	Bits 7 to 0	Initialized	Retained	Retained	Retained	Retained
FSISTR	Bit 7	Initialized	Retained	Retained	Retained	Initialized
	Bits 6 and 5	Initialized	Initialized	Retained	Retained	Initialized
	Bits 4 to 0	Initialized	Retained	Retained	Retained	Retained
FSITDR7 to FSITDR0	Bits 7 to 0	Initialized	Retained	Retained	Retained	Retained
FSIRDR	Bits 7 to 0	Initialized	Retained	Retained	Retained	Retained
FSI internal sequencer		Initialized	Retained	Retained	Retained	Initialized

21.6 Interrupt Sources

The FSI has four interrupt sources for the slave (this LSI): FSITEI, FSIRXI, FSICMDI, and FSIWI. FSITEI is a transmit end interrupt when the slave executes the SPI flash memory write transfer. FSIRXI is a receive end interrupt when the slave executes the SPI flash memory read transfer. FSICMDI is a command receive interrupt in host FSI command write. FSIWI is a write receive interrupt in the case of write from the host to the SPI flash memory. Setting the corresponding enable bit to 1 enables the relevant interrupt request to be issued.

Table 21.9 FSI Interrupt Sources

Interrupt Name	Interrupt Source		Interrupt Enable Bit
FSII	FSITEI	Transmit end	FSITEIE
	FSIRXI	Receive data full	FSIRXIE
LFSII	FSICMDI	FSI command reception	FSICMDIE
	FSIWI	FSI write reception	FSIWIE

21.7 Usage Note

21.7.1 Longword Transfer in FW Memory Write Cycles

When longword transfers of FW memory write cycles are used, the maximum operating frequency of the system clock is 10 MHz.

Section 22 A/D Converter

This LSI includes one unit (unit 0) of successive-approximation-type 10-bit A/D converter that allows up to sixteen analog input channels to be selected. Figure 22.1 shows a block diagram for unit 0.

22.1 Features

- 10-bit resolution
- Input channels: Sixteen channels
- Conversion cycle: 40 cycles (A/D conversion clock)
- Two kinds of operating modes
  Single mode: Single-channel A/D conversion
  Scan mode: Continuous A/D conversion on one to four channels or continuous A/D conversion on one to eight channels
- A/D conversion clocks specifiable (φ, φ/2, φ/4, or φ/8)
- Eight data registers
  Conversion results are held in a 16-bit data register for each channel
- Sample and hold function
- Three kinds of A/D conversion start
  Software
  Conversion start trigger from 16-bit timer pulse unit (TPU) or 8-bit timer (TMR)
- Interrupt source
  A/D conversion end interrupt (ADI) request can be generated

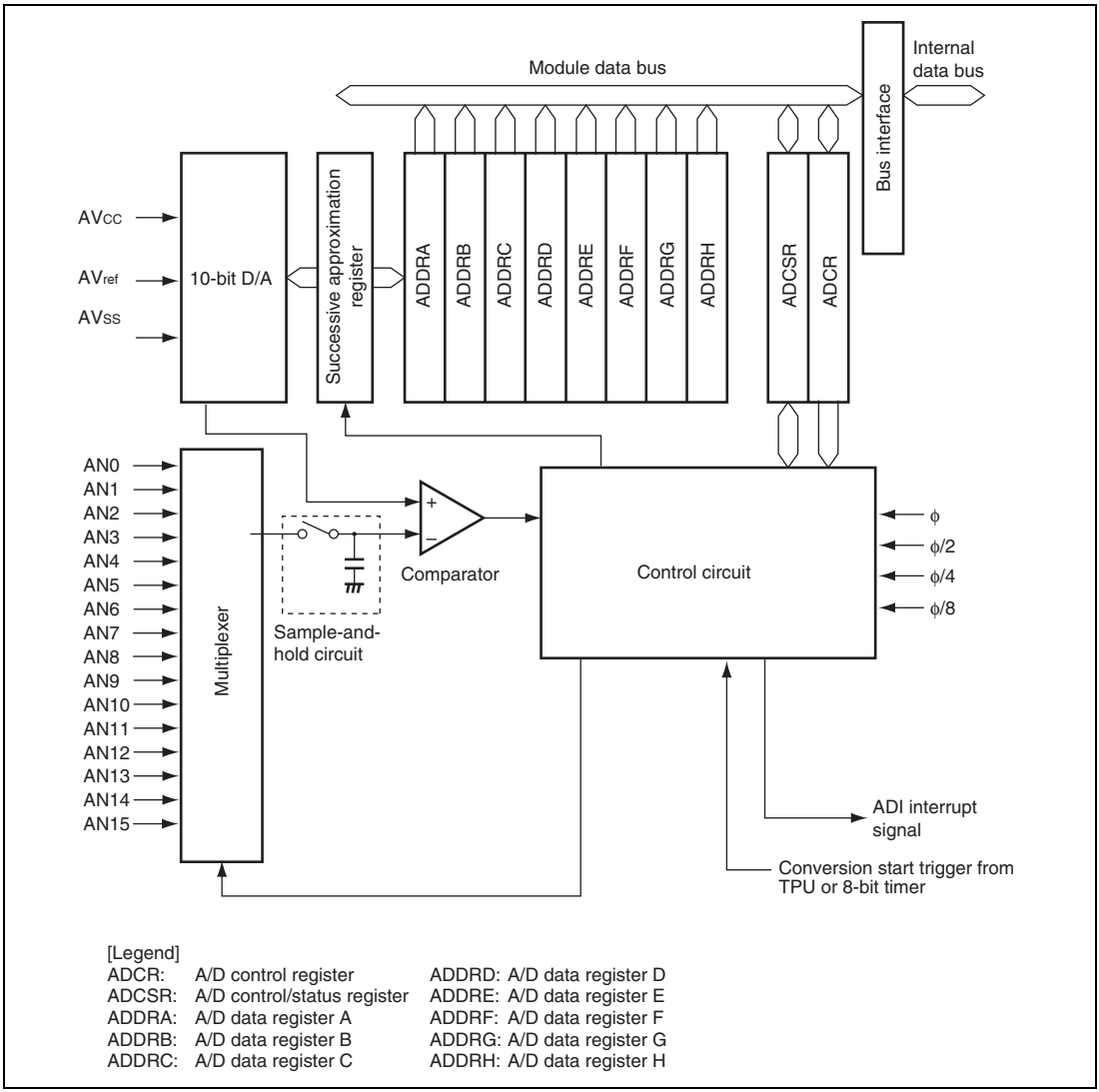


Figure 22.1 Block Diagram of A/D Converter

22.2 Input/Output Pins

Table 22.1 summarizes the pins used by the A/D converter.

The AVCC and AVSS pins are the power supply pins for the analog block in the A/D converter. The AVref pin is a reference voltage pin for the A/D converter. The sixteen analog input pins are divided into two channel sets: analog input pins 0 to 7 (AN0 to AN7) comprising channel set 0 and analog input pins 8 to 15 (AN8 to AN15) comprising channel set 1.

Table 22.1 Pin Configuration

Pin Name	Symbol	I/O	Function
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
Analog input pin 0	AN0	Input	Channel set 0 analog input
Analog input pin 1	AN1	Input	
Analog input pin 2	AN2	Input	
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	
Analog input pin 8	AN8	Input	Channel set 1 analog input
Analog input pin 9	AN9	Input	
Analog input pin 10	AN10	Input	
Analog input pin 11	AN11	Input	
Analog input pin 12	AN12	Input	
Analog input pin 13	AN13	Input	
Analog input pin 14	AN14	Input	
Analog input pin 15	AN15	Input	

22.3 Register Descriptions

The A/D converter has the following registers.

Table 22.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Data Bus Width
A/D data register A	ADDRA	R	H'0000	H'FC00	16
A/D data register B	ADDRB	R	H'0000	H'FC02	16
A/D data register C	ADDRC	R	H'0000	H'FC04	16
A/D data register D	ADDRD	R	H'0000	H'FC06	16
A/D data register E	ADDRE	R	H'0000	H'FC08	16
A/D data register F	ADDRF	R	H'0000	H'FC0A	16
A/D data register G	ADDRG	R	H'0000	H'FC0C	16
A/D data register H	ADDRH	R	H'0000	H'FC0E	16
A/D control/status register	ADCSR	R/W	H'00	H'FC10	8
A/D control register	ADCR	R/W	H'00	H'FC11	8

22.3.1 A/D Data Registers A to H (ADDRA to ADDRH)

There are eight 16-bit read-only ADDR registers, ADDRA to ADDRH, used to store the results of A/D conversion. The ADDR registers which store a conversion result for each channel are shown in table 22.3.

The 10-bit conversion data is stored in bits 15 to 6. The lower six bits are always read as 0.

The data bus between the CPU and the A/D converter is sixteen bits wide. The data can be read directly from the CPU. ADDR must not be accessed in 8-bit units and must be accessed in 16-bit units.

Table 22.3 Analog Input Channels and Corresponding ADDR

Analog Input Channel		A/D Data Register to Store A/D Conversion Results
Channel Set 0 (CH3 = 0)	Channel Set 1 (CH3 = 1)	
AN0	AN8	ADDRA
AN1	AN9	ADDRB
AN2	AN10	ADDRC
AN3	AN11	ADDRD
AN4	AN12	ADDRE
AN5	AN13	ADDRF
AN6	AN14	ADDRG
AN7	AN15	ADDRH

22.3.2 A/D Control/Status Register (ADCSR)

ADCSR controls A/D converter operation.

Bit	Bit Name	Initial Value	R/W	Description
7	ADF	0	R/(W)*	A/D End Flag A status flag that indicates the end of A/D conversion. [Setting conditions] • When A/D conversion ends in single mode • When A/D conversion ends on all channels specified in scan mode [Clearing condition] When 0 is written after reading ADF = 1
6	ADIE	0	R/W	A/D Interrupt Enable Enables ADI interrupt by ADF when this bit is set to 1.
5	ADST	0	R/W	A/D Start When this bit is cleared to 0, A/D conversion stops and enters wait state. When this bit is set to 1 by a conversion start trigger from software, TPU, or TMR, A/D conversion starts. This bit remains set to 1 during A/D conversion. In single mode, this bit is automatically cleared to 0 when conversion on the specified channel ends. In scan mode, conversion continues sequentially on the specified channels until this bit is cleared to 0 by a reset, or software.
4	—	0	—	Reserved This bit is always read as 0 and cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description		
3	CH3	0	R/W	Channel Select 3 to 0		
2	CH2	0	R/W	Select analog input channels with the SCANE and SCANS bits in ADCRS.		
1	CH1	0	R/W			
0	CH0	0	R/W	The input channel setting must be made when conversion is halted (ADST = 0).		
				When SCANE = 0 and SCANS = X	When SCANE = 1 and SCANS = 0	When SCANE = 1 and SCANS = 1
				0000: AN0	0000: AN0	0000: AN0
				0001: AN1	0001: AN0, AN1	0001: AN0, AN1
				0010: AN2	0010: AN0 to AN2	0010: AN0 to AN2
				0011: AN3	0011: AN0 to AN3	0011: AN0 to AN3
				0100: AN4	0100: AN4	0100: AN0 to AN4
				0101: AN5	0101: AN4, AN5	0101: AN0 to AN5
				0110: AN6	0110: AN4 to AN6	0110: AN0 to AN6
				0111: AN7	0111: AN4 to AN7	0111: AN0 to AN7
				1000: AN8	1000: AN8	1000: AN8
				1001: AN9	1001: AN8, AN9	1001: AN8, AN9
				1010: AN10	1010: AN8 to AN10	1010: AN8 to AN10
				1011: AN11	1011: AN8 to AN11	1011: AN8 to AN11
				1100: AN12	1100: AN12	1100: AN8 to AN12
				1101: AN13	1101: AN12, AN13	1101: AN8 to AN13
				1110: AN14	1110: AN12 to AN14	1110: AN8 to AN14
				1111: AN15	1111: AN12 to AN15	1111: AN8 to AN15

[Legend]

X: Don't care

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

22.3.3 A/D Control Register (ADCR)

ADCR enables A/D conversion started by an external trigger signal.

Bit	Bit Name	Initial Value	R/W	Description
7	TRGS1	0	R/W	Timer Trigger Select 1 and 0
6	TRGS0	0	R/W	Enable the start of A/D conversion by a trigger signal. 00: A/D conversion start by external trigger is disabled 01: A/D conversion start by conversion trigger from TPU 10: A/D conversion start by conversion trigger from TMR 11: Setting prohibited
5	SCANE	0	R/W	Scan Mode
4	SCANS	0	R/W	Select the A/D conversion operating mode. 0X: Single mode 10: Scan mode Continuous A/D conversion on 1 to 4 channels 11: Scan mode Continuous A/D conversion on 1 to 8 channels
3	CKS1	0	R/W	Clock Select 1 and 0
2	CKS0	0	R/W	These bits select the clock (ADCLK)* used in A/D conversion. Set these bits while the ADST bit in ADCSR is 0, then set the conversion mode. 00: ϕ 01: $\phi/2$ 10: $\phi/4$ 00: $\phi/8$
1	ADSTCLR	0	R/W	A/D Start Clear Sets the automatic clearing of the ADST bit in scan mode. 0: Disables the automatic clearing of the ADST bit in scan mode 1: Automatically clears the bit when A/D conversion of all of the selected channels are completed
0	—	0	R	Reserved This bit is always read as 0 and cannot be modified.

[Legend]

X: Don't care

Note: * Set the clock so that ADCLK ≤ 10 MHz.

22.4 Operation

The A/D converter operates by successive approximation with 10-bit resolution. It has two operating modes: single mode and scan mode. First, select the clock used in A/D conversion. When changing the operating mode or analog input channel, to prevent incorrect operation, first clear the ADST bit in ADCSR to 0 to halt A/D conversion. The ADST bit can be set at the same time the operating mode or analog input channel is changed.

22.4.1 Single Mode

In single mode, A/D conversion is to be performed only once on the specified single channel. Operations are as follows.

1. A/D conversion on the specified channel is started when the ADST bit in ADCSR is set to 1 by software, the TMR, or the TPU.
2. When A/D conversion is completed, the result is transferred to the A/D data register corresponding to the channel.
3. On completion of A/D conversion, the ADF bit in ADCSR is set to 1. If the ADIE bit is set to 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 ADST bit is automatically cleared to 0 and the A/D converter enters wait state. When the ADST bit is cleared to 0 during A/D conversion, the conversion stops and the A/D converter enters wait state.

22.4.2 Scan Mode

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

1. When the ADST bit in ADCSR is set to 1 by software, the TPU, or the TMR, A/D conversion starts on the first channel in the selected channel set.
2. Continuous A/D conversion on up to four channels (SCANE = 1 and SCANS = 0) or continuous A/D conversion on up to eight channels (SCANE = 1 and SCANS = 1) can be selected. When continuous A/D conversion on four channels is selected, A/D conversion starts from the following channels: AN0 when CH3 = 0 and CH2 = 0, AN4 when CH3 = 0 and CH2 = 1, AN8 when CH3 = 1 and CH2 = 0, and AN12 when CH3 = 1 and CH2 = 1.

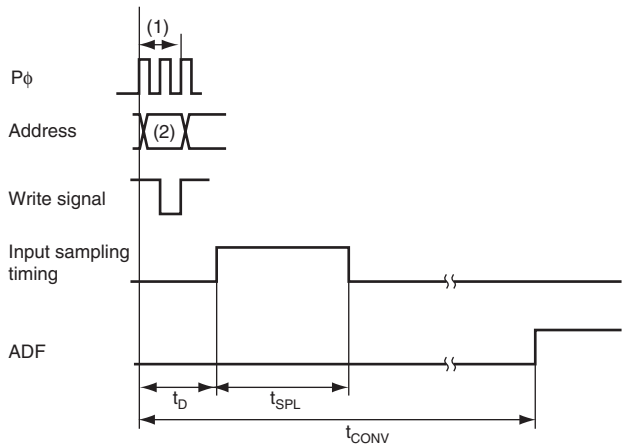
   When continuous A/D conversion on eight channels is selected, A/D conversion starts from the following channels: AN0 when CH3 = 0 and CH2 = 0 and AN8 when CH3 = 1 and CH2 = 0.
3. When A/D conversion for each channel is completed, the result is sequentially transferred to the A/D data register corresponding to each channel.
4. When conversion of all the selected channels is completed, the ADF bit in ADCSR is set to 1. If the ADIE bit is set to 1 at this time, an ADI interrupt is requested. Conversion from the first channel in the channel set starts again.
5. The ADST bit is not automatically cleared to 0 so steps [2] and [3] are repeated as long as the ADST bit remains set to 1. When the ADST bit is cleared to 0, A/D conversion stops and the A/D converter enters wait state. After this, setting the ADST bit to 1 starts A/D conversion from the first channel again.
6. When the ADST bit is automatically cleared on completion of the A/D conversion of all of the selected channels with the ADSTCLR bit in ADCR set to 1, A/D conversion stops and enters the wait state.

22.4.3 Input Sampling and A/D Conversion Time

The A/D converter has a built-in sample-and-hold circuit. The A/D converter samples the analog input when the A/D conversion start delay time (t_D) passes after the ADST bit in ADCSR is set to 1, then starts A/D conversion. Figure 22.2 shows the A/D conversion timing. Table 22.4 indicates the A/D conversion time.

As indicated in figure 22.2, the A/D conversion time (t_{CONV}) includes t_D and the input sampling time (t_{SPL}). The length of t_D varies depending on the timing of write to ADCSR. The total conversion time therefore varies within the ranges indicated in table 22.4.

In scan mode, the values shown in table 22.4 become those for the first conversion time. The second and subsequent conversion times are listed in table 22.5. In either case, bits CKS1 and CKS0 in ADCR should be set so that the conversion time is within the ranges indicated by the A/D conversion characteristics.



[Legend]
(1): ADCSR write cycle
(2): ADCSR address
t_D: A/D conversion start delay time
t_{SPL}: Input sampling time
t_{CONV}: A/D conversion time

Figure 22.2 A/D Conversion Timing

Table 22.4 A/D Conversion Time (Single Mode)

		CKS1 = 0						CKS1 = 1					
		CKS0 = 0			CKS0 = 1			CKS0 = 0			CKS0 = 1		
Item	**Symbol**	**Min.**	**Typ.**	**Max.**	**Min.**	**Typ.**	**Max.**	**Min.**	**Typ.**	**Max.**	**Min.**	**Typ.**	**Max.**
A/D conversion start delay time	t_D	(4)	—	(5)	(6)	—	(9)	(10)	—	(17)	(18)	—	(33)
Input sampling time	t_{SPL}	—	15	—	—	30	—	—	60	—	—	120	—
A/D conversion time	t_{CONV}	44	—	45	8x	—	8x	16x	—	16x	32x	—	32x

Note: Values in the table indicate the number of states.

Table 22.5 A/D Conversion Time (Scan Mode)

CKS1	CKS0	Conversion Time (State)
0	0	40 (fixed)
0	1	80 (fixed)
1	0	160 (fixed)
1	0	320 (fixed)

22.5 Interrupt Source

The A/D converter generates an A/D conversion end interrupt (ADI) at the end of A/D conversion. If the ADF bit in ADCSR has been set to 1 after A/D conversion ends and the ADIE bit is set to 1, an ADI interrupt request is enabled.

Table 22.6 A/D Converter Interrupt Source

Name	Interrupt Source	Interrupt Flag
ADI	A/D conversion end	ADF

22.6 A/D Conversion Accuracy Definitions

This LSI's A/D conversion accuracy definitions are given below.

- Resolution

  The number of A/D converter digital output codes
- Quantization error

  The deviation inherent in the A/D converter, given by 1/2 LSB (see figure 22.3).
- Offset error

  The deviation of the analog input voltage value from the ideal A/D conversion characteristics when the digital output changes from the minimum voltage value B'00 0000 0000 (H'000) to B'00 0000 0001 (H'001) (see figure 22.4).
- Full-scale error

  The deviation of the analog input voltage value from the ideal A/D conversion characteristics when the digital output changes from B'11 1111 1110 (H'3FE) to B'11 1111 1111 (H'3FF) (see figure 22.4).
- Nonlinearity error

  The error with respect to the ideal A/D conversion characteristics between the zero voltage and the full-scale voltage. Does not include the offset error, full-scale error, or quantization error (see figure 22.4).
- Absolute accuracy

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

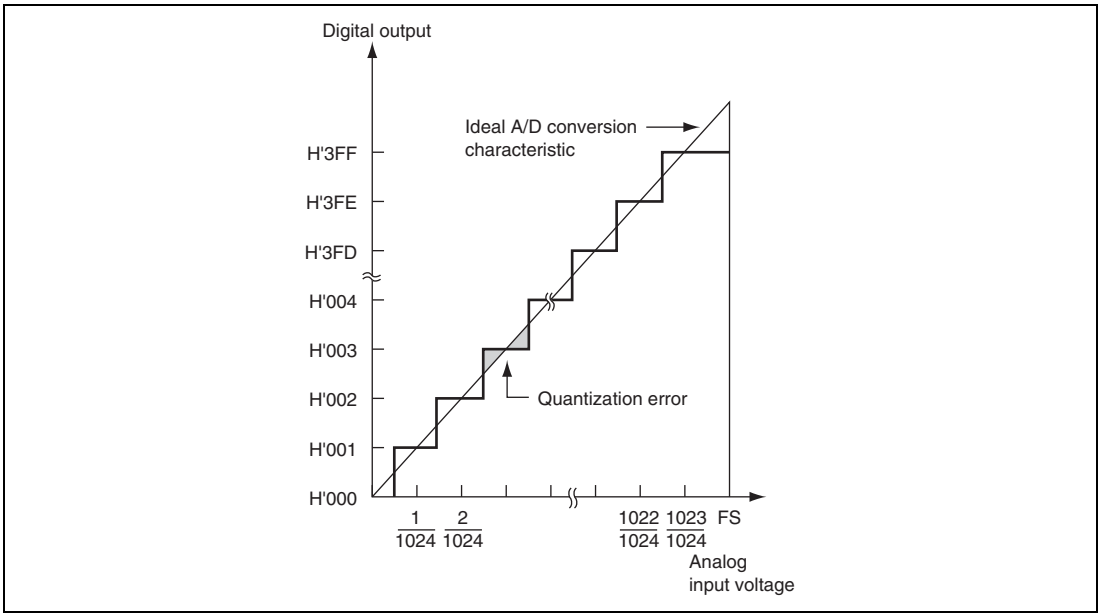


Figure 22.3 A/D Conversion Accuracy Definitions

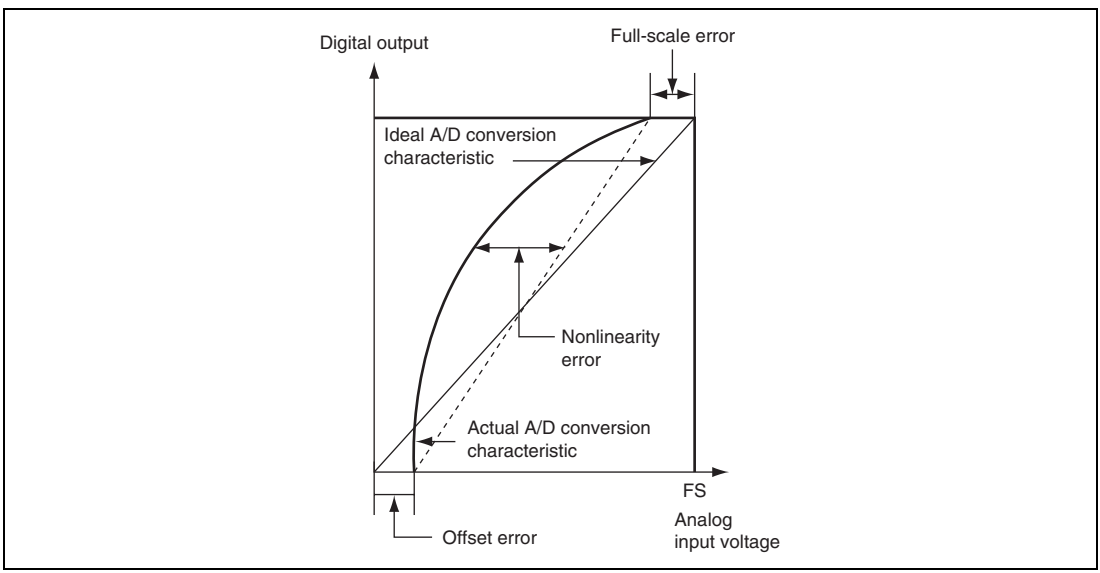


Figure 22.4 A/D Conversion Accuracy Definitions

22.7 Usage Notes

22.7.1 Module Stop Mode Setting

The A/D converter operation can be enabled or disabled using the module stop control register. With the initial setting, the A/D converter is stopped. Register access is enabled by canceling module stop mode. For details, see section 26, Power-Down Modes.

22.7.2 Permissible Signal Source Impedance

This LSI's analog input is designed so that the conversion accuracy is guaranteed for an input signal for which the signal source impedance is 5 kΩ or less. This specification is provided to enable the A/D converter's sample-and-hold circuit input capacitance to be charged within the sampling time; if the sensor output impedance exceeds 5 kΩ, charging may be insufficient and it may not be possible to guarantee the A/D conversion accuracy. However, if a large capacitance is provided externally in single mode, the input load will essentially comprise only the internal input resistance of 10 kΩ, and the signal source impedance is ignored. However, since a low-pass filter effect is obtained in this case, it may not be possible to follow an analog signal with a large differential coefficient (e.g., voltage fluctuation ratio of 5 mV/µs or greater) (see figure 22.5). When converting a high-speed analog signal or converting in scan mode, a low-impedance buffer should be inserted.

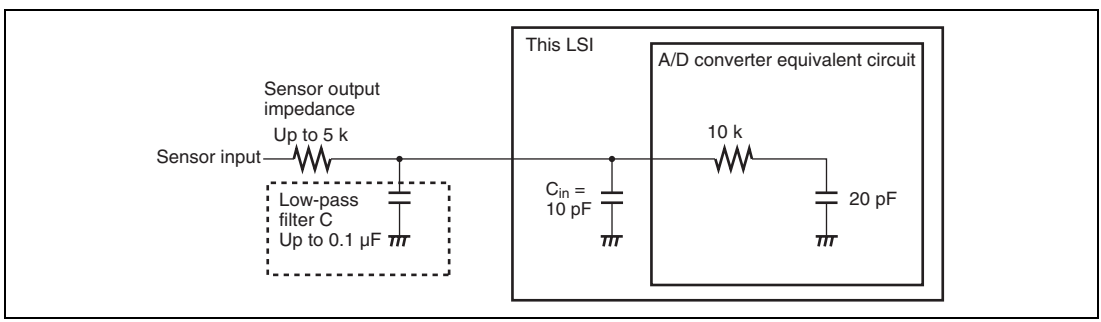


Figure 22.5 Example of Analog Input Circuit

22.7.3 Influences on Absolute Accuracy

Adding capacitance results in coupling with GND, and therefore noise in GND may adversely affect the absolute accuracy. Be sure to make the connection to an electrically stable GND such as AVss.

Care is also required to insure that filter circuits do not interfere with digital signals on the mounting board, so acting as antennas.

22.7.4 Setting Range of Analog Power Supply and Other Pins

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

- Analog input voltage range
  The voltage applied to analog input pins (AN0 to AN15) during A/D conversion should be in the range AVss ≤ ANn ≤ AVref (n = 0 to 15).
- Relation between AVcc, AVss and Vcc, Vss
  As the relationship between AVcc, AVss and Vcc, Vss, set AVcc = Vcc ±0.3 V and AVss = Vss. If the A/D converter is not used, set AVcc = Vcc and AVss = Vss.
- AVref pin range
  The reference voltage of the AVref pin should be in the range AVref ≤ AVcc.

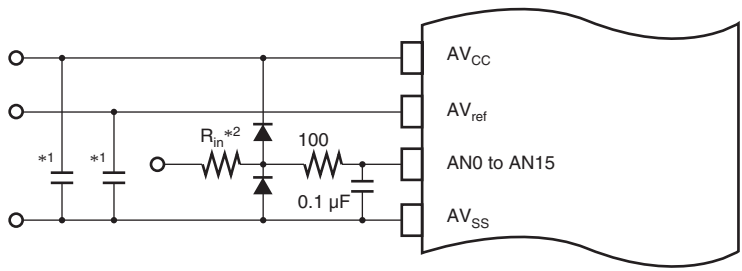
22.7.5 Notes on Board Design

In board design, digital circuitry and analog circuitry should be as mutually isolated as possible, and layout in which digital circuit signal lines and analog circuit signal lines cross or are in close proximity should be avoided as far as possible. Failure to do so may result in incorrect operation of the analog circuitry due to inductance, adversely affecting A/D conversion values. Also, digital circuitry must be isolated from the analog input pins (AN0 to AN15), analog reference voltage (AVref), and analog power supply voltage (AVcc) by the analog ground (AVss). Also, the analog ground (AVss) should be connected at one point to a stable ground (Vss) on the board.

22.7.6 Notes on Noise Countermeasures

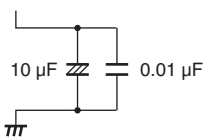
A protection circuit connected to prevent damage of the analog input pins (AN0 to AN15) and analog reference voltage pin (AVref) due to an abnormal voltage such as an excessive surge should be connected between AVcc and AVss, as shown in figure 22.6. Also, the bypass capacitors connected to AVcc and AVref, and the filter capacitors connected to AN0 to AN15 must be connected to AVss.

If a filter capacitor is connected, the input currents at the analog input pins (AN0 to AN15) are averaged, and so an error may arise. Also, when A/D conversion is performed frequently, as in scan mode, if the current charged and discharged by the capacitance of the sample-and-hold circuit in the A/D converter exceeds the current input via the input impedance (R_{in}), an error will arise in the analog input pin voltage. Careful consideration is therefore required when deciding the circuit constants.



Notes: Values are reference values.

1.

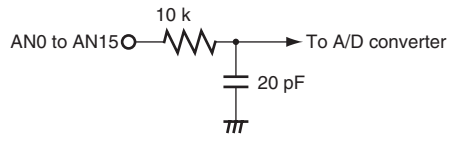


2. R_{in}: Input impedance

Figure 22.6 Example of Analog Input Protection Circuit

Table 22.7 Analog Pin Specifications

Item	Min.	Max.	Unit
Analog input capacitance	—	20	pF
Permissible signal-source impedance	—	5	kΩ



Note: Values are reference values.

Figure 22.7 Analog Input Pin Equivalent Circuit

22.7.7 Module Stop Mode Setting

When this LSI enters software standby mode with A/D conversion enabled, the analog inputs are retained, and the analog power supply current is equal to the current as during A/D conversion. If the analog power supply current needs to be reduced in software standby mode, clear the ADST, TRGS1, and TRGS0 bits all to 0 to disable A/D conversion.

22.7.8 Note on Activation of the A/D Converter by an External Trigger

When starting of the A/D converter by an external trigger* is in use, any of the following actions (1. to 3.) may lead to a situation where stopping of the A/D converter is not possible.

Note: * External trigger: Conversion-start trigger from the peripheral modules (TMU and TPU)

1. Changing the value of the ADST bit in ADCSR from 0 to 1
2. Changing from the activation by external trigger setting to the external-trigger-disabled setting
3. Changing the scan-mode setting (changing the setting of the SCANE and ADSTCLR bits to switch from continuous scan mode to single mode or one-cycle scan mode)

If any of the above points is applicable, please make settings in accord with the instructions below.

If 1. is applicable:

Do not set the ADST bit in ADCSR to 1.

If 2. or 3. is applicable:

Be sure to invalidate the external trigger input before changing the setting from activation by the external trigger to disabling of the external trigger or changing the scan-mode setting (changing the setting of the SCANE and ADSTCLR bits) while activation by the external trigger is in use.

Setting the TRGS1 and TRGS0 bits in ADCR according to the procedure overleaf invalidates the external trigger input.

See figure 22.8 for details of the procedure in cases where 2. or 3. is applicable.

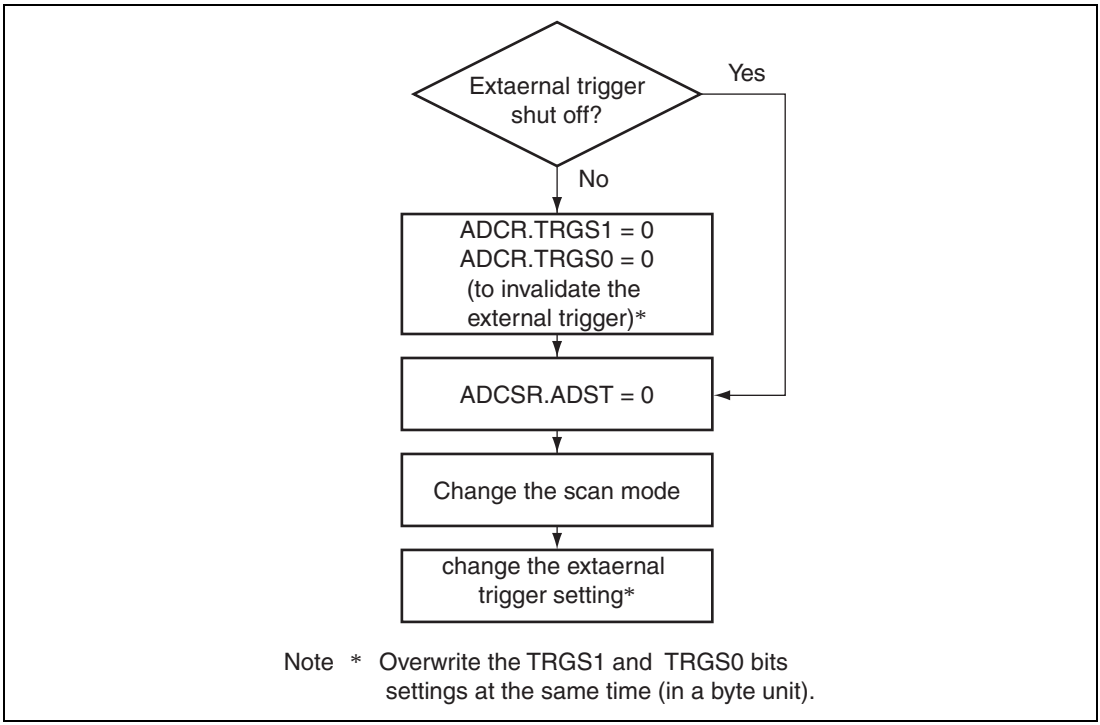


Figure 22.8 Procedure for Changing Modes when Starting of the A/D Converter by an External Trigger has been Selected

Section 23 RAM

This LSI has 8 Kbytes of on-chip high-speed static RAM. The RAM is connected to the CPU by a 16-bit data bus, enabling one-state access by the CPU for both byte data and word data.

The on-chip RAM can be enabled or disabled by means of the RAME bit in the system control register (SYSCR). For details on SYSCR, see section 3.2.2, System Control Register (SYSCR).

Section 24 Flash Memory

The flash memory has the following features. Figure 24.1 is a block diagram of the flash memory.

24.1 Features

- Size

Product Classification		ROM Size	ROM Address
H8S/2117R	R4F2117R	160 kbytes	H'000000 to H'027FFF (mode 2)

- Programming/erasing interface by the download of on-chip program

  This LSI has a programming/erasing program. After downloading this program to the on-chip RAM, programming/erasing can be performed by setting the parameters.
- Programming/erasing time

  Programming time: 1 ms (typ) for 128-byte simultaneous programming, 7.8 μs per byte

  Erasing time: 600 ms (typ) per 1 block (64 kbytes)
- Number of programming

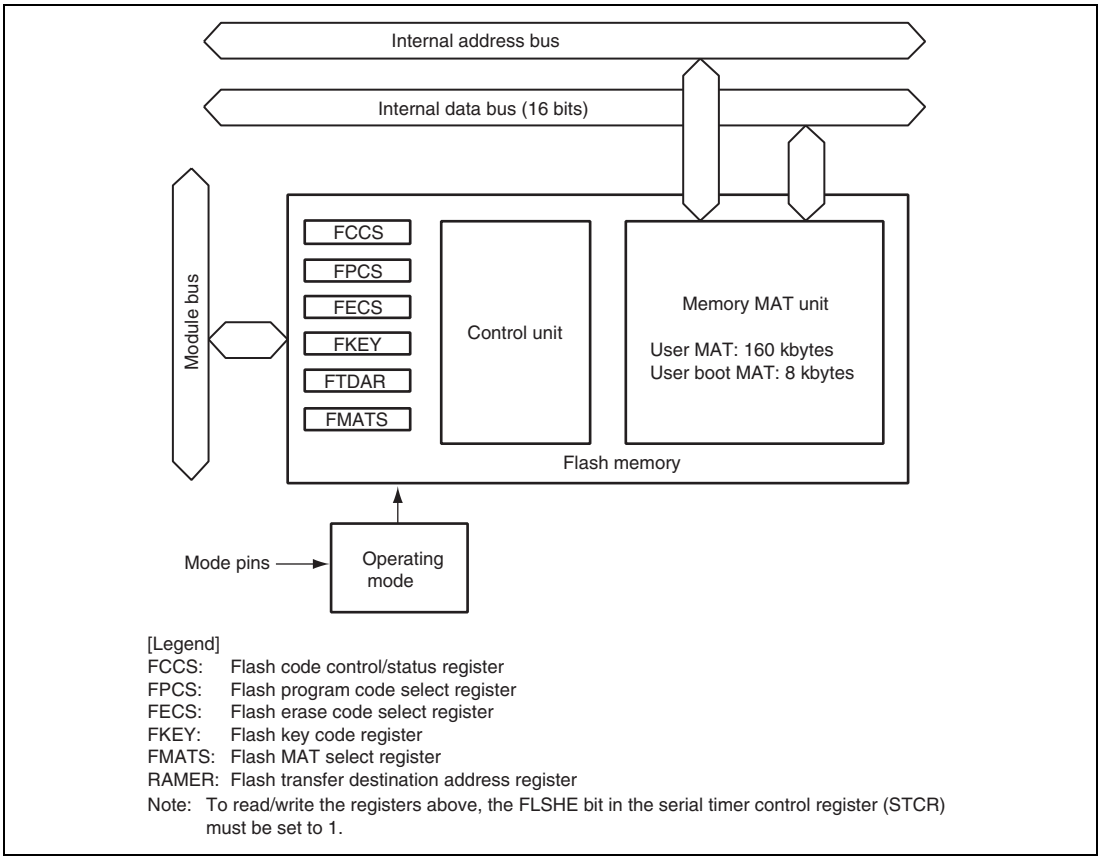
  The number of programming can be up to 100 times at the minimum. (1 to 100 times are guaranteed.)
- Two on-board programming modes

  Boot mode: Using the on-chip SCI-1, the user MAT can be programmed/erased. In boot mode, the bit rate between the host and this LSI can be adjusted automatically.

  User program mode: Using a desired interface, the user MAT can be programmed/erased.
- Off-board programming mode

  Programmer mode: Using a PROM programmer, the user MAT can be programmed/erased.
- Programming/erasing protection

  Protection against programming/erasing of the flash memory can be set by hardware protection, software protection, or error protection.



[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
RAMER: Flash transfer destination address register
Note: To read/write the registers above, the FLSHE bit in the serial timer control register (STCR) must be set to 1.

Figure 24.1 Block Diagram of Flash Memory

24.2 Mode Transition Diagram

When the mode pins are set in the reset state and reset start is performed, this LSI enters each operating mode as shown in figure 24.2. Although the flash memory can be read in user mode, it cannot be programmed or erased. The flash memory can be programmed or erased in boot mode, user program mode, user boot mode, and programmer mode. The differences between boot mode, user program mode, user boot mode, and programmer mode are shown in table 24.1.

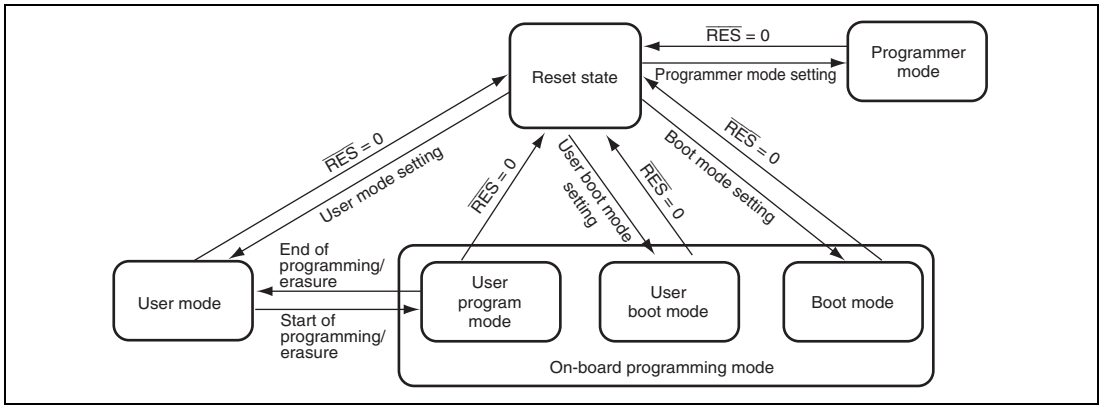


Figure 24.2 Mode Transition of Flash Memory

Table 24.1 Differences between Boot Mode, User Program Mode, and Programmer Mode

Item	Boot Mode	User Program Mode	User Boot Mode	Programmer Mode
Programming/ erasing environment	On-board programming	On-board programming	On-board programming	PROM programmer
Programming/ erasing enable MAT	• User MAT • User boot MAT	• User MAT	• User MAT	• User MAT • User boot MAT
All erasure	O (Automatic)	O	O	O (Automatic)
Block division erasure	O*[1]	O	O	×
Program data transfer	From host via SCI	Via any device	Via any device	Via programmer
Reset initiation MAT	Embedded program storage area	User MAT	User boot MAT*[2]	—
Transition to user mode	Changing mode and reset	Changing FLSHE bit setting	Changing mode and reset	—

Notes: 1. All-erasure is performed. After that, the specified block can be erased.

2. In this LSI, the user programming mode is defined as the period from the timing when a program concerning programming and erasure is started to the timing when the program is completed. For details on a program concerning programming and erasure, see section 24.8.2, User Program Mode.

24.3 Flash Memory MAT Configuration

This LSI's flash memory is configured by the 160-Kbyte user MAT and 8-Kbyte user boot MAT. The start address is allocated to the same address in the user MAT and user boot MAT. Therefore, when program execution or data access is performed between two MATs, the MAT must be switched by using FMATS.

The user MAT or user boot MAT can be read in all modes. However, the user boot MAT can be programmed only in boot mode and programmer mode.

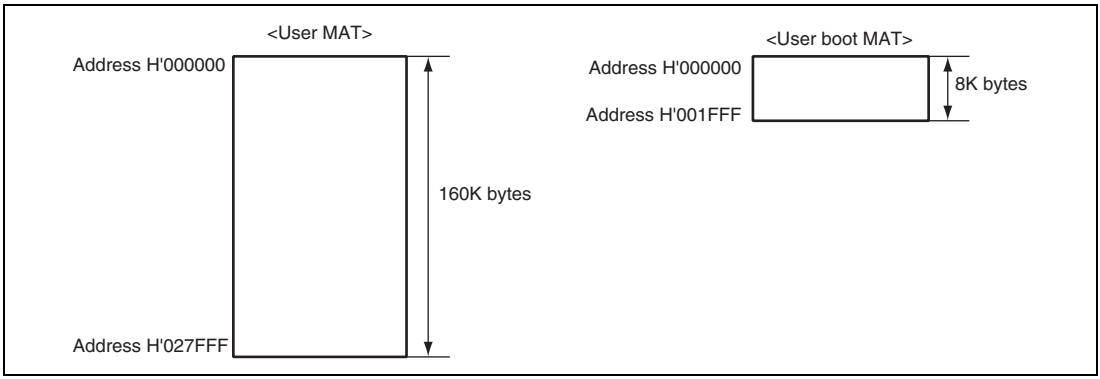


Figure 24.3 Flash Memory Configuration

The size of the user MAT is different from that of the user boot MAT. An address that exceeds the size of the 8-Kbyte user boot MAT should not be accessed. If the attempt is made, data is read as an undefined value.

24.4 Block Structure

Figure 24.4 shows the 160-kbyte block structure. The heavy-line frames indicate the erase blocks. The thin-line frames indicate the programming units and the values inside the frames indicates the addresses. The 160-kbyte user MAT is divided into one 64-kbyte block, two 32-kbyte blocks, and eight 4-kbyte blocks. The user MAT can be erased in these block units. Programming is done in 128-byte units starting from where the lower address is H'00 or H'80.

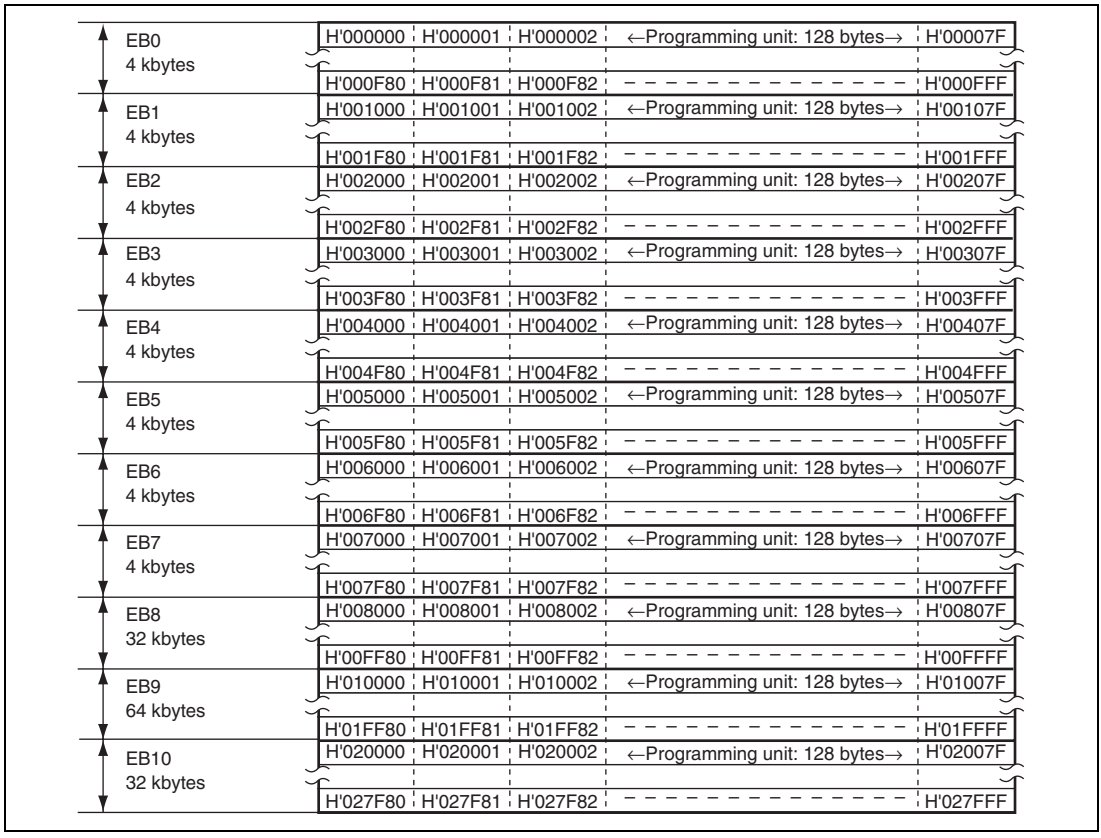


Figure 24.4 Block Structure of the User MAT

24.5 Programming/Erasing Interface

Programming/erasing of the flash memory is done by downloading an on-chip programming/erasing program to the on-chip RAM and specifying the start address of the programming destination, the program data, and the erase block number using the programming/erasing interface registers and programming/erasing interface parameters.

The procedure program for user program mode is made by the user. Figure 24.5 shows the procedure for creating the procedure program. For details, see section 24.8.2, User Program Mode.

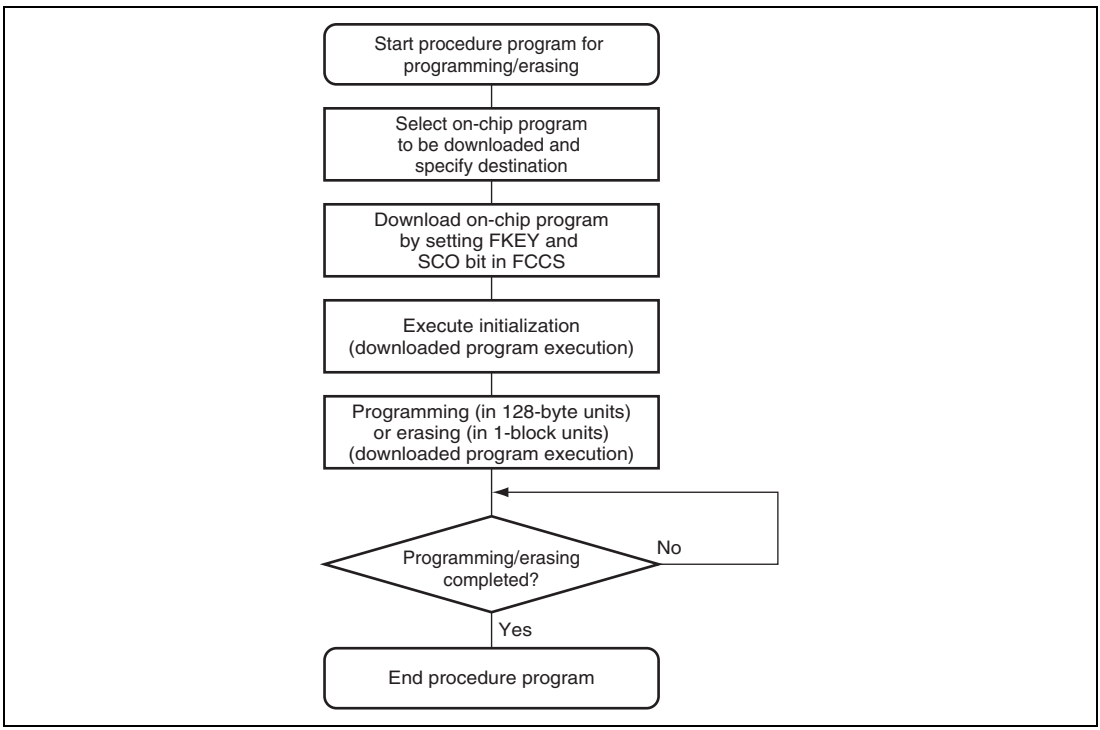


Figure 24.5 Procedure for Creating Procedure Program

(1) Selection of On-Chip Program to be Downloaded

This LSI has programming/erasing programs which can be downloaded to the on-chip RAM. The on-chip program to be downloaded is selected by the programming/erasing interface registers. The start address of the on-chip RAM where an on-chip program is downloaded is specified by the flash transfer destination address register (FTDAR).

(2) Download of On-Chip Program

The on-chip program is automatically downloaded by setting the flash key code register (FKEY) and the SCO bit in the flash code control/status register (FCCS). The memory MAT is replaced with the embedded program storage area during download. Since the memory MAT cannot be read during programming/erasing, the procedure program must be executed in a space other than the flash memory (for example, on-chip RAM). Since the download result is returned to the programming/erasing interface parameter, whether download is normally executed or not can be confirmed.

(3) Initialization of Programming/Erasing

A pulse with the specified period must be applied when programming or erasing. The specified pulse width is made by the method in which wait loop is configured by the CPU instruction. Accordingly, the operating frequency of the CPU needs to be set before programming/erasing. The operating frequency of the CPU is set by the programming/erasing interface parameter.

(4) Execution of Programming/Erasing

The start address of the programming destination and the program data are specified in 128-byte units when programming. The block to be erased is specified with the erase block number in erase-block units when erasing. Specifications of the start address of the programming destination, program data, and erase block number are performed by the programming/erasing interface parameters, and the on-chip program is initiated. The on-chip program is executed by using the JSR or BSR instruction and executing the subroutine call of the specified address in the on-chip RAM. The execution result is returned to the programming/erasing interface parameter.

The area to be programmed must be erased in advance when programming flash memory. All interrupts are disabled during programming/erasing.

(5) When Programming/Erasing is Executed Consecutively

When processing does not end by 128-byte programming or 1-block erasure, consecutive programming/erasing can be realized by updating the start address of the programming destination and program data, or the erase block number. Since the downloaded on-chip program is left in the on-chip RAM even after programming/erasing completes, download and initialization are not required when the same processing is executed consecutively.

24.6 Input/Output Pins

The flash memory is controlled through the input/output pins shown in table 24.2.

Table 24.2 Pin Configuration

Abbreviation	I/O	Function
$\overline{\text{RES}}$	Input	Reset
MD2, MD1	Input	Set operating mode of this LSI
TxD1	Output	Serial transmit data output (used in boot mode)
RxD1	Input	Serial receive data input (used in boot mode)

24.7 Register Descriptions

The flash memory has the following registers and parameters.

Table 24.3 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Data Bus Width
Flash code control status register	FCCS	R/W*	H'80	H'FEA8	8
Flash program code select register	FPCS	R/W	H'00	H'FEA9	8
Flash erase code select register	FECS	R/W	H'00	H'FEAA	8
Flash key code register	FKEY	R/W	H'00	H'FEAC	8
Flash MAT select register	FMATS	R/W	H'00	H'FEAD	8
Flash transfer destination address register	FTDAR	R/W	H'00	H'FEAE	8

Note: * Bits other than the SCO bit are read-only bits. The SCO bit is a write-only bit and is always read as 0.

Table 24.4 Parameter Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Data Bus Width
Download path fail result parameter	DPFR	R/W*	Undefined	On-chip RAM*	8, 16, 32
Flash path/fail parameter	FPFR	R/W	Undefined	R0L of CPU	8, 16, 32
Flash program/erase frequency parameter	FPEFEQ	R/W	Undefined	ER0 of CPU	8, 16, 32
Flash multipurpose address area parameter	FMPAR	R/W	Undefined	ER1 of CPU	8, 16, 32
Flash multipurpose data destination parameter	FMPDR	R/W	Undefined	ER0 of CPU	8, 16, 32
Flash erase block select parameter	FEBS	R/W	Undefined	ER0 of CPU	8, 16, 32

Note: * One byte of the start address on the on-chip RAM specified by FTDAR

There are several operating modes for accessing the flash memory. Respective operating modes, registers, and parameters are assigned to the user MAT. The correspondence between operating modes and registers/parameters for use is shown in table 24.5.

Table 24.5 Registers/Parameters and Target Modes

Register/Parameter		Download	Initialization	Programming	Erasure	Read
Programming/ erasing interface registers	FCCS	O	—	—	—	—
	FPCS	O	—	—	—	—
	FECS	O	—	—	—	—
	FKEY	O	—	O	O	—
	FMATS	—	—	O*[1]	O*[1]	O*[2]
	FTDAR	O	—	—	—	—
Programming/ erasing interface parameters	DPFR	O	—	—	—	—
	FPFR	—	O	O	O	—
	FPEFEQ	—	O	—	—	—
	FMPAR	—	—	O	—	—
	FMPDR	—	—	O	—	—
	FEBS	—	—	—	O	—

Notes: 1. Programming and erasure of the user MAT in user boot mode require settings.
2. A setting may be required depending on the combination of the startup mode and the MAT to be read.

24.7.1 Programming/Erasing Interface Registers

The programming/erasing interface registers are 8-bit registers that can be accessed only in bytes. These registers are initialized by a power-on reset.

(1) Flash Code Control/Status Register (FCCS)

FCCS monitors errors during programming/erasing the flash memory and requests the on-chip program to be downloaded to the on-chip RAM.

Bit	Bit Name	Initial Value	R/W	Description
7	—	1	R	Reserved
6	—	0	R	These are read-only bits and cannot be modified.
5	—	0	R	
4	FLER	0	R	Flash Memory Error Indicates that an error has occurred during programming or erasing the flash memory. When this bit is set to 1, the flash memory enters the error protection state. When this bit is set to 1, high voltage is applied to the internal flash memory. To reduce the damage to the flash memory, the reset must be released after the reset input period (period of $\overline{RES}$ = 0) of at least 100 μs. 0: Flash memory operates normally (Error protection is invalid) [Clearing condition] • At a power-on reset 1: An error occurs during programming/erasing flash memory (Error protection is valid) [Setting conditions] • When an interrupt, such as NMI, occurs during programming/erasing. • When the flash memory is read during programming/erasing (including a vector read and an instruction fetch). • When the SLEEP instruction is executed during programming/erasing (including software standby mode).

Bit	Bit Name	Initial Value	R/W	Description
3 to 1	—	All 0	R	Reserved These are read-only bits and cannot be modified.
0	SCO	0	(R)/W*	Source Program Copy Operation Requests the on-chip programming/erasing program to be downloaded to the on-chip RAM. When this bit is set to 1, the on-chip program which is selected by FPCS or FECS is automatically downloaded in the on-chip RAM area specified by FTDAR. In order to set this bit to 1, H'A5 must be written to FKEY, and this operation must be executed in the on-chip RAM. Dummy read of FCCS must be executed twice immediately after setting this bit to 1. All interrupts must be disabled during download. This bit is cleared to 0 when download is completed. During program download initiated with this bit, particular processing which accompanies bank-switching of the program storage area is executed. 0: Download of the programming/erasing program is not requested. [Clearing condition] • When download is completed 1: Download of the programming/erasing program is requested. [Setting conditions] (When all of the following conditions are satisfied) • H'A5 is written to FKEY • Setting of this bit is executed in the on-chip RAM

Note: * This is a write-only bit. This bit is always read as 0.

(2) Flash Program Code Select Register (FPCS)

FPCS selects the programming program to be downloaded.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These are read-only bits and cannot be modified.
0	PPVS	0	R/W	Program Pulse Verify Selects the programming program to be downloaded. 0: Programming program is not selected. [Clearing condition] When transfer is completed 1: Programming program is selected.

(3) Flash Erase Code Select Register (FECS)

FECS selects the erasing program to be downloaded.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These are read-only bits and cannot be modified.
0	EPVB	0	R/W	Erase Pulse Verify Block Selects the erasing program to be downloaded. 0: Erasing program is not selected. [Clearing condition] When transfer is completed 1: Erasing program is selected.

(4) Flash Key Code Register (FKEY)

FKEY is a register for software protection that enables to download the on-chip program and perform programming/erasing of the flash memory.

Bit	Bit Name	Initial Value	R/W	Description
7	K7	0	R/W	Key Code
6	K6	0	R/W	When H'A5 is written to FKEY, writing to the SCO bit in FCCS is enabled. When a value other than H'A5 is written, the SCO bit cannot be set to 1. Therefore, the on-chip program cannot be downloaded to the on-chip RAM.
5	K5	0	R/W	
4	K4	0	R/W	
3	K3	0	R/W	
2	K2	0	R/W	Only when H'5A is written can programming/erasing of the flash memory be executed. When a value other than H'5A is written, even if the programming/erasing program is executed, programming/erasing cannot be performed.
1	K1	0	R/W	
0	K0	0	R/W	
				H'A5: Writing to the SCO bit is enabled. (The SCO bit cannot be set to 1 when FKEY is a value other than H'A5.)
				H'5A: Programming/erasing of the flash memory is enabled. (When FKEY is a value other than H'A5, the software protection state is entered.)
				H'00: Initial value

(5) Flash MAT Select Register (FMATS)

FMATS specifies whether the user MAT or user boot MAT is selected.

Bit	Bit Name	Initial Value	R/W	Description
7	MS7	0/1*	R/W	MAT Select
6	MS6	0	R/W	The user MAT is selected when a value other than H'AA is written, and the user boot MAT is selected when H'AA is written. The MAT is switched by writing a value in FMATS. To switch the MAT, make sure to follow section 24.10, Switching between User MAT and User Boot MAT. (The user boot MAT cannot be programmed in user program mode even if the user boot MAT is selected by FMATS. The user boot MAT must be programmed in boot mode or programmer mode.)
5	MS5	0/1*	R/W	
4	MS4	0	R/W	
3	MS3	0/1*	R/W	
2	MS2	0	R/W	
1	MS1	0/1*	R/W	
0	MS0	0	R/W	
				H'AA: User boot MAT is selected (user MAT is selected when the value of these bits is other than H'AA). Initial value when initiated in user boot mode.
				H'00: Initial value when initiated in a mode other than user boot mode (user MAT is selected).
				[Programmable condition]
				Execution state in the on-chip RAM

Note: * The initial value is 1 in user boot mode and 0 in a mode other than user boot mode.

(6) Flash Transfer Destination Address Register (FTDAR)

FTDAR specifies the start address of the on-chip RAM at which to download an on-chip program. FTDAR must be set before setting the SCO bit in FCCS to 1.

Bit	Bit Name	Initial Value	R/W	Description
7	TDER	0	R/W	Transfer Destination Address Setting Error This bit is set to 1 when an error has occurred in setting the start address specified by bits TDA6 to TDA0. A start address error is determined by whether the value set in bits TDA6 to TDA0 is within the range of H'00 to H'01 when download is executed by setting the SCO bit in FCCS to 1. Make sure that this bit is cleared to 0 before setting the SCO bit to 1 and the value specified by bits TDA6 to TDA0 should be within the range of H'00 to H'01. 0: The value specified by bits TDA6 to TDA0 is within the range. 1: The value specified by bits TDA6 to TDA0 is between H'02 and H'FF and download has stopped.
6	TDA6	0	R/W	Transfer Destination Address Specifies the on-chip RAM start address of the download destination. A value between H'00 and H'01, and up to 3 kbytes can be specified as the start address of the on-chip RAM. H'00: H'FFD080 is specified as the start address. H'01: H'FFD880 is specified as the start address. H'02 to H'7F: Setting prohibited. (Specifying a value from H'02 to H'7F sets the TDER bit to 1 and stops download of the on-chip program.)
5	TDA5	0	R/W	
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	

24.7.2 Programming/Erasing Interface Parameters

The programming/erasing interface parameters specify the operating frequency, storage place for program data, start address of programming destination, and erase block number, and exchanges the execution result. These parameters use the general registers of the CPU (ER0 and ER1) or the on-chip RAM area. The initial values of programming/erasing interface parameters are undefined at a power-on reset or a transition to software standby mode.

Since registers of the CPU except for R0L are saved in the stack area during download of an on-chip program, initialization, programming, or erasing, allocate the stack area before performing these operations (the maximum stack size is 128 bytes). The return value of the processing result is written in R0L. The programming/erasing interface parameters are used in download control, initialization before programming or erasing, programming, and erasing. Table 24.6 shows the usable parameters and target modes. The meaning of the bits in the flash pass and fail result parameter (FPFR) varies in initialization, programming, and erasure.

Table 24.6 Parameters and Target Modes

Parameter	Download	Initialization	Programming	Erasure	R/W	Initial Value	Allocation
DPFR	O	—	—	—	R/W	Undefined	On-chip RAM*
FPFR	—	O	O	O	R/W	Undefined	R0L of CPU
FPEFEQ	—	O	—	—	R/W	Undefined	ER0 of CPU
FMPAR	—	—	O	—	R/W	Undefined	ER1 of CPU
FMPDR	—	—	O	—	R/W	Undefined	ER0 of CPU
FEBS	—	—	—	O	R/W	Undefined	ER0 of CPU

Note: * A single byte of the start address of the on-chip RAM specified by FTDAR

(a) Download Control

The on-chip program is automatically downloaded by setting the SCO bit in FCCS to 1. The on-chip RAM area to download the on-chip program is the 4-kbyte area starting from the start address specified by FTDAR. Download is set by the programming/erasing interface registers, and the download pass and fail result parameter (DPFR) indicates the return value.

(b) Initialization before Programming/Erasing

The on-chip program includes the initialization program. A pulse with the specified period must be applied when programming or erasing. The specified pulse width is made by the method in which wait loop is configured by the CPU instruction. Accordingly, the operating frequency of the CPU must be set. The initial program is set as a parameter of the programming/erasing program which has been downloaded to perform these settings.

(c) Programming

When the flash memory is programmed, the start address of the programming destination on the user MAT and the program data must be passed to the programming program.

The start address of the programming destination on the user MAT must be stored in general register ER1. This parameter is called the flash multipurpose address area parameter (FMPAR).

The program data is always in 128-byte units. When the program data does not satisfy 128 bytes, 128-byte program data is prepared by filling the dummy code (H'FF). The boundary of the start address of the programming destination on the user MAT is aligned at an address where the lower eight bits (A7 to A0) are H'00 or H'80.

The program data for the user MAT must be prepared in consecutive areas. The program data must be in a consecutive space which can be accessed using the MOV.B instruction of the CPU and is not in the flash memory space.

The start address of the area that stores the data to be written in the user MAT must be set in general register ER0. This parameter is called the flash multipurpose data destination area parameter (FMPDR).

For details on the programming procedure, see section 24.8.2, User Program Mode.

(d) Erasure

When the flash memory is erased, the erase block number on the user MAT must be passed to the erasing program which is downloaded.

The erase block number on the user MAT must be set in general register ER0. This parameter is called the flash erase block select parameter (FEBS).

One block is selected from the block numbers of 0 to 10 as the erase block number.

For details on the erasing procedure, see section 24.8.2, User Program Mode.

(1) Download Pass and Fail Result Parameter (DPFR: Single Byte of Start Address in On-Chip RAM Specified by FTDAR)

DPFR indicates the return value of the download result. The DPFR value is used to determine the download result.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	—	—	Unused These bits return 0.
2	SS	—	R/W	Source Select Error Detect Only one type can be specified for the on-chip program which can be downloaded. When the program to be downloaded is not selected, more than two types of programs are selected, or a program which is not mapped is selected, an error occurs. 0: Download program selection is normal 1: Download program selection is abnormal
1	FK	—	R/W	Flash Key Register Error Detect Checks the FKEY value (H'A5) and returns the result. 0: FKEY setting is normal (H'A5) 1: FKEY setting is abnormal (value other than H'A5)
0	SF	—	R/W	Success/Fail Returns the download result. Reads back the program downloaded to the on-chip RAM and determines whether it has been transferred to the on-chip RAM. 0: Download of the program has ended normally (no error) 1: Download of the program has ended abnormally (error occurs)

(2) Flash Pass and Fail Parameter (FPFR: General Register R0L of CPU)

FPFR indicates the return values of the initialization, programming, and erasure results. The meaning of the bits in FPFR varies depending on the processing.

(a) Initialization before programming/erasing

FPFR indicates the return value of the initialization result.

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	—	—	Unused These bits return 0.
1	FQ	—	R/W	Frequency Error Detect Compares the specified CPU operating frequency with the operating frequencies supported by this LSI, and returns the result. 0: Setting of operating frequency is normal 1: Setting of operating frequency is abnormal
0	SF	—	R/W	Success/Fail Returns the initialization result. 0: Initialization has ended normally (no error) 1: Initialization has ended abnormally (error occurs)

(b) Programming

FPFR indicates the return value of the programming result.

Bit	Bit Name	Initial Value	R/W	Description
7	—	—	—	Unused Returns 0.
6	MD	—	R/W	Programming Mode Related Setting Error Detect Detects the error protection state and returns the result. When the error protection state is entered, this bit is set to 1. Whether the error protection state is entered or not can be confirmed with the FLER bit in FCCS. For conditions to enter the error protection state see section 24.9.3, Error Protection. 0: Normal operation (FLER = 0) 1: Error protection state, and programming cannot be performed (FLER = 1)
5	EE	—	R/W	Programming Execution Error Detect Writes 1 to this bit when the specified data could not be written because the user MAT was not erased. If this bit is set to 1, there is a high possibility that the user MAT has been written to partially. In this case, after removing the error factor, erase the user MAT. Also an attempt to write the user MAT when the FMATS value is H'AA and the user boot MAT is selected leads to a programming execution error. In that case, both the user MAT and user boot MAT are not rewritten. Writing to the user boot MAT must be performed in boot mode or programmer mode. 0: Programming has ended normally 1: Programming has ended abnormally (programming result is not guaranteed)
4	FK	—	R/W	Flash Key Register Error Detect Checks the FKEY value (H'5A) before programming starts, and returns the result. 0: FKEY setting is normal (H'5A) 1: FKEY setting is abnormal (value other than H'5A)
3	—	—	—	Unused Returns 0.

Bit	Bit Name	Initial Value	R/W	Description
2	WD	—	R/W	Write Data Address Detect When an address not in the flash memory area is specified as the start address of the storage destination for the program data, an error occurs. 0: Setting of the start address of the storage destination for the program data is normal 1: Setting of the start address of the storage destination for the program data is abnormal
1	WA	—	R/W	Write Address Error Detect When the following items are specified as the start address of the programming destination, an error occurs. • An area other than flash memory • The specified address is not aligned with the 128-byte boundary (lower eight bits of the address are other than H'00 and H'80) 0: Setting of the start address of the programming destination is normal 1: Setting of the start address of the programming destination is abnormal
0	SF	—	R/W	Success/Fail Returns the programming result. 0: Programming has ended normally (no error) 1: Programming has ended abnormally (error occurs)

(c) Erasure

FPFR indicates the return value of the erasure result.

Bit	Bit Name	Initial Value	R/W	Description
7	—	—	—	Unused Returns 0.
6	MD	—	R/W	Erasure Mode Related Setting Error Detect Detects the error protection state and returns the result. When the error protection state is entered, this bit is set to 1. Whether the error protection state is entered or not can be confirmed with the FLER bit in FCCS. For conditions to enter the error protection state see section 24.9.3, Error Protection. 0: Normal operation (FLER = 0) 1: Error protection state, and programming cannot be erased (FLER = 1)
5	EE	—	R/W	Erasure Execution Error Detect Returns 1 when the user MAT could not be erased or when the flash memory related register settings are partially changed. If this bit is set to 1, there is a high possibility that the user MAT has been erased partially. In this case, after removing the error factor, erase the user MAT. Also an attempt to erase the user MAT when the FMATS value is H'AA and the user boot MAT is selected leads to an erasure execution error. In that case, both the user MAT and user boot MAT are not erased. Erasure of the user boot MAT must be performed in boot mode or programmer mode. 0: Erasure has ended normally 1: Erasure has ended abnormally
4	FK	—	R/W	Flash Key Register Error Detect Checks the FKEY value (H'5A) before erasure starts, and returns the result. 0: FKEY setting is normal (H'5A) 1: FKEY setting is abnormal (value other than H'5A)

Bit	Bit Name	Initial Value	R/W	Description
3	EB	—	R/W	Erase Block Select Error Detect Checks whether the specified erase block number is in the block range of the user MAT, and returns the result. 0: Setting of erase block number is normal 1: Setting of erase block number is abnormal
2, 1	—	—	—	Unused These bits return 0.
0	SF	—	R/W	Success/Fail Indicates the erasure result. 0: Erasure has ended normally (no error) 1: Erasure has ended abnormally (error occurs)

(3) Flash Program/Erase Frequency Parameter (FPEFEQ: General Register ER0 of CPU)

FPEFEQ sets the operating frequency of the CPU. The operating frequency available in this LSI ranges from 8 MHz to 20 MHz.

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	—	—	Unused These bits should be cleared to 0.
15 to 0	F15 to F0	—	R/W	Frequency Set These bits set the operating frequency of the CPU. The setting value must be calculated as follows: 1. Round off the operating frequency expressed in MHz unit at the third decimal place to make it into two decimal places. 2. Multiply the rounded number by 100 and convert the result into binary and write it to FPEFEQ (general register ER0). For example, when the operating frequency of the CPU is 20.000 MHz, the setting value is as follows: 1. Round 20.000 off at the third decimal place as 20.00. 2. Convert 20.00 × 100 = 2000 into a binary number and set B'0000 0111 1101 0000 (H'07D0) in ER0.

(4) Flash Multipurpose Address Area Parameter (FMPAR: General Register ER1 of CPU)

FMPAR stores the start address of the programming destination on the user MAT.

When an address in an area other than the flash memory is set, or the start address of the programming destination is not aligned with the 128-byte boundary, an error occurs. The error occurrence is indicated by the WA bit in FPFR.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	MOA31 to MOA0	—	R/W	These bits store the start address of the programming destination on the user MAT. Consecutive 128-byte programming is executed starting from the specified start address of the user MAT. Therefore, the specified start address of the programming destination becomes a 128-byte boundary, and MOA6 to MOA0 are always cleared to 0.

(5) Flash Multipurpose Data Destination Parameter (FMPDR: General Register ER0 of CPU)

FMPDR stores the start address in the area which stores the data to be programmed in the user MAT.

When the storage destination for the program data is in flash memory, an error occurs. 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	These bits store the start address of the area which stores the program data for the user MAT. Consecutive 128-byte data is programmed to the user MAT starting from the specified start address.

(6) Flash Erase Block Select Parameter (FEBS: General Register ER0 of CPU)

FEBS specifies the erase block number. Settable values range from 0 to 10 (H'00000000 to H'0000000A). A value of 0 corresponds to block EB0 and a value of 10 corresponds to block EB10. Do not set a value outside the range from 0 to 10.

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	Undefined	R/W	Unused These bits should be set to 0.
7 to 0	EBS7 to EBS0	Undefined	R/W	These bits specify the erase block number from 0 to 10. A value of 0 corresponds to block EB0 and 10 corresponds to block EB10. Do not set a value outside the range from 0 to 10 (from H'00 to H'0A).

24.8 On-Board Programming Mode

When the mode pins (MD1 and MD2) are set to on-board programming mode and the reset start is executed, a transition is made to on-board programming mode in which the on-chip flash memory can be programmed/erased. On-board programming mode has three operating modes: boot mode, user boot mode, and user program mode.

Table 24.7 shows the pin setting for each operating mode. For details on the state transition of each operating mode for flash memory, see figure 24.2.

Table 24.7 On-Board Programming Mode Setting

Mode Setting	MD2	MD1	NMI
Boot mode	1	0	1
User program mode	0	1	0/1
User boot mode	1	0	0

24.8.1 Boot Mode

Boot mode executes programming/erasing of the user MAT and the user boot MAT by means of the control command and program data transmitted from the externally connected host via the on-chip SCI_1.

In boot mode, the tool for transmitting the control command and program data, and the program data must be prepared in the host. The serial communication mode is set to asynchronous mode. The system configuration in boot mode is shown in figure 24.6. Interrupts are ignored in boot mode. Configure the user system so that interrupts do not occur.

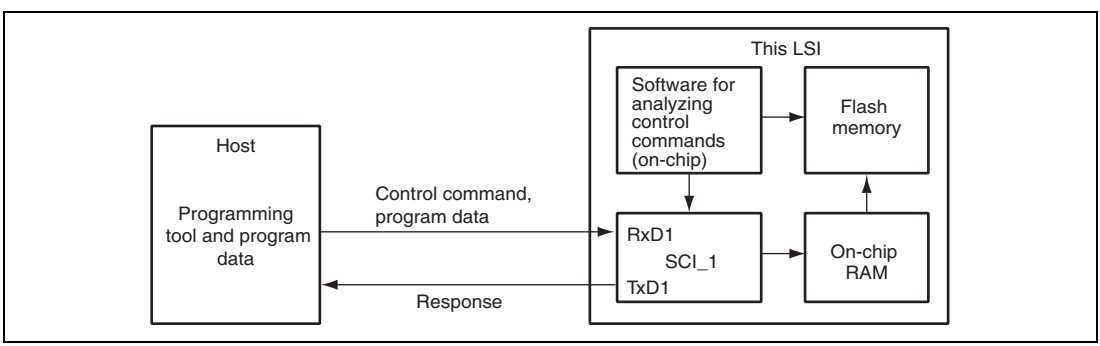


Figure 24.6 System Configuration in Boot Mode

(1) Serial Interface Setting by Host

The SCI_1 is set to asynchronous mode, and the serial transmit/receive format is set to 8-bit data, one stop bit, and no parity.

When a transition to boot mode is made, the boot program embedded in this LSI is initiated.

When the boot program is initiated, this LSI measures the low period of asynchronous serial communication data (H'00) transmitted consecutively by the host, calculates the bit rate, and adjusts the bit rate of the SCI_1 to match that of the host.

When bit rate adjustment is completed, this LSI transmits 1 byte of H'00 to the host as the bit adjustment end sign. When the host receives this bit adjustment end sign normally, it transmits 1 byte of H'55 to this LSI. When reception is not executed normally, initiate boot mode again. The bit rate may not be adjusted within the allowable range depending on the combination of the bit rate of the host and the system clock frequency of this LSI. Therefore, the transfer bit rate of the host and the system clock frequency of this LSI must be as shown in table 24.8.

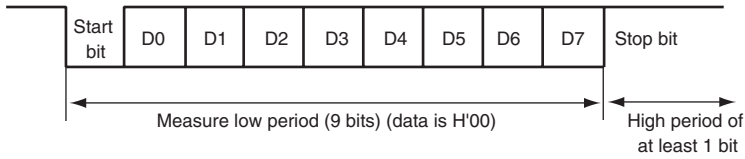


Figure 24.7 Automatic-Bit-Rate Adjustment Operation

Table 24.8 System Clock Frequency for Automatic-Bit-Rate Adjustment

Bit Rate of Host	System Clock Frequency of This LSI
9,600 bps	8 to 20 MHz
19,200 bps	8 to 20 MHz

(2) State Transition Diagram

The state transition after boot mode is initiated is shown in figure 24.8.

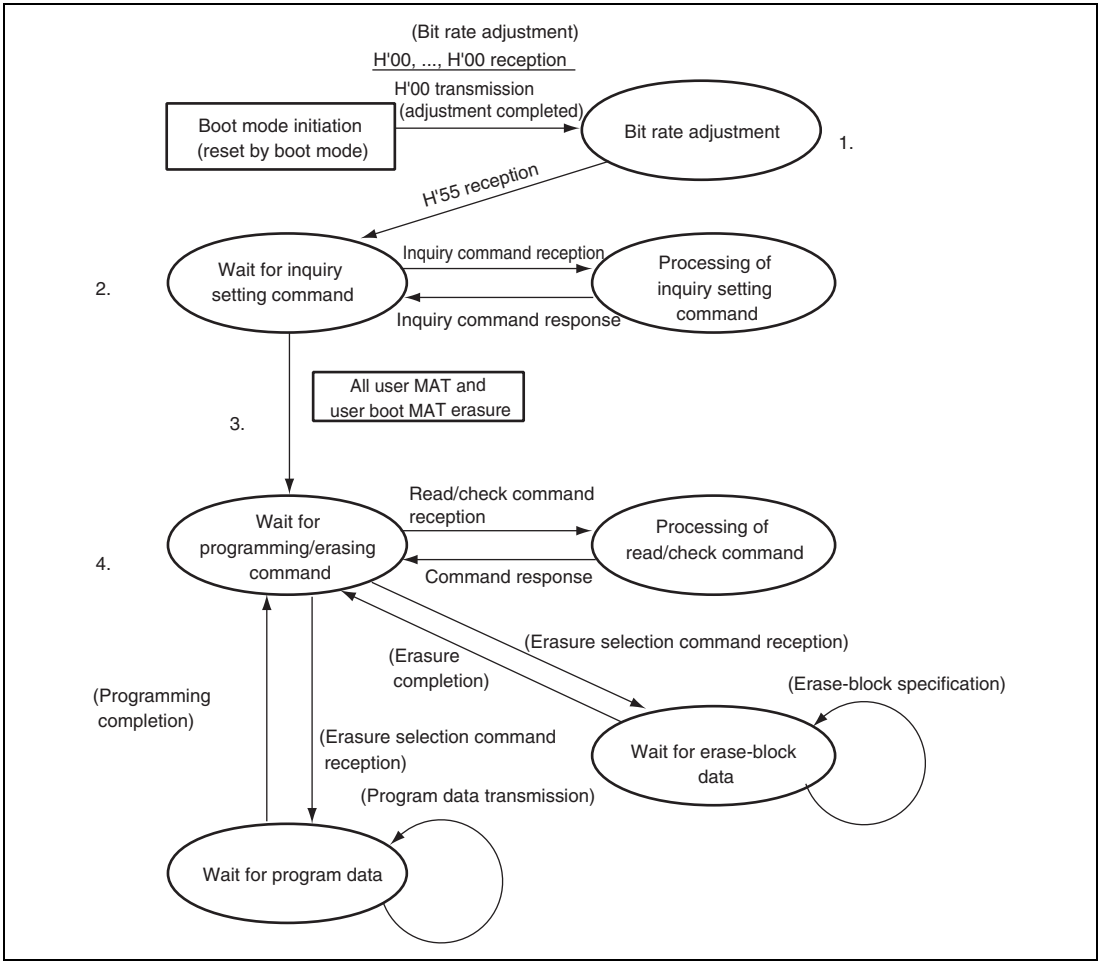


Figure 24.8 Boot Mode State Transition Diagram

1. After boot mode is initiated, the bit rate of the SCI_1 is adjusted with that of the host.
2. Inquiry information about the size, configuration, start address, and support status of the user MAT is transmitted to the host.
3. After inquiries have finished, all user MAT and user boot MAT are automatically erased.
4. When the program preparation notice is received, the state of waiting for program data is entered. The start address of the programming destination and program data must be transmitted after the programming command is transmitted. When programming is finished, the start address of the programming destination must be set to H'FFFFFFFF and transmitted. Then the state of waiting for program data is returned to the state of waiting for programming/erasing command. When reprogramming an erase block including an area on which the programming end command is issued, erase the erase block. An example of the erase block is shown in figure 24.9. When the erasure preparation notice is received, the state of waiting for erase block data is entered. The erase block number must be transmitted after the erasing command is transmitted. When the erasure is finished, the erase block number must be set to H'FF and transmitted. Then the state of waiting for erase block data is returned to the state of waiting for programming/erasing command. Erasure must be executed when the specified block is programmed without a reset start after programming is executed in boot mode. When programming can be executed by only one operation, all blocks are erased before entering the state of waiting for programming/erasing command or another command. Thus, in this case, the erasing operation is not required. The commands other than the programming/erasing command perform sum check, blank check (erasure check), and memory read of the user MAT and user boot MAT, and acquisition of current status information.

Memory read of the user MAT and user boot MAT can only read the data programmed after all user MAT and user boot MAT have automatically been erased. No other data can be read.

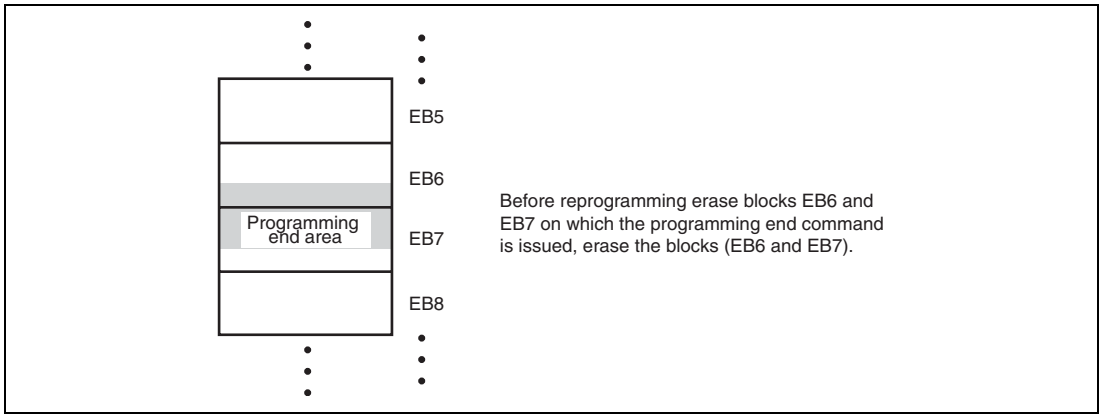
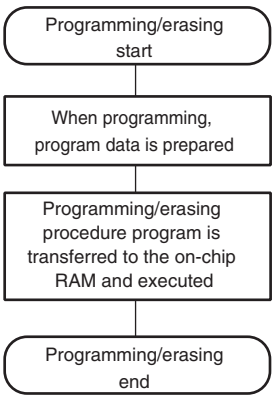


Figure 24.9 Example of Erase Block Including Programmed Area

24.8.2 User Program Mode

Programming/erasing of the user MAT is executed by downloading an on-chip program. The programming/erasing flow is shown in figure 24.10.

Since high voltage is applied to the internal flash memory during programming/erasing, a transition to the reset state must not be made during programming/erasing. A transition to the reset state during programming/erasing may damage the flash memory. If a reset is input, the reset must be released after the reset input period (period of $\overline{RES}$ = 0) of at least 100 μs.



1. Programming/erasing is executed only in the on-chip RAM.
2. After programming/erasing is finished, protect the flash memory by the hardware protection.
3. Make sure that the program data do not overlap with the download destination specified by FTDAR.

Figure 24.10 Programming/Erasing Flow

(1) On-Chip RAM Address Map when Programming/Erasing is Executed

Parts of the procedure program that is made by the user, like download request, programming/erasing procedure, and decision of the result, must be executed in the on-chip RAM. Since the on-chip program to be downloaded is embedded in the on-chip RAM, make sure the on-chip program and procedure program do not overlap. Figure 24.11 shows the area of the on-chip program to be downloaded.

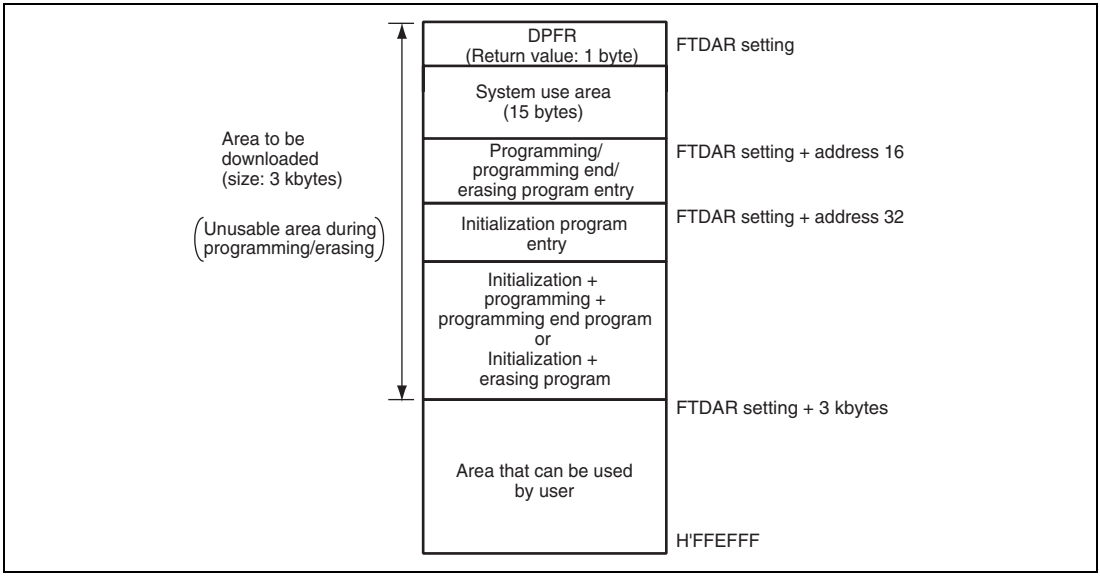


Figure 24.11 RAM Map when Programming/Erasing is Executed

(2) Programming Procedure in User Program Mode

The procedures for download of the on-chip program, initialization, and programming are shown in figure 24.12.

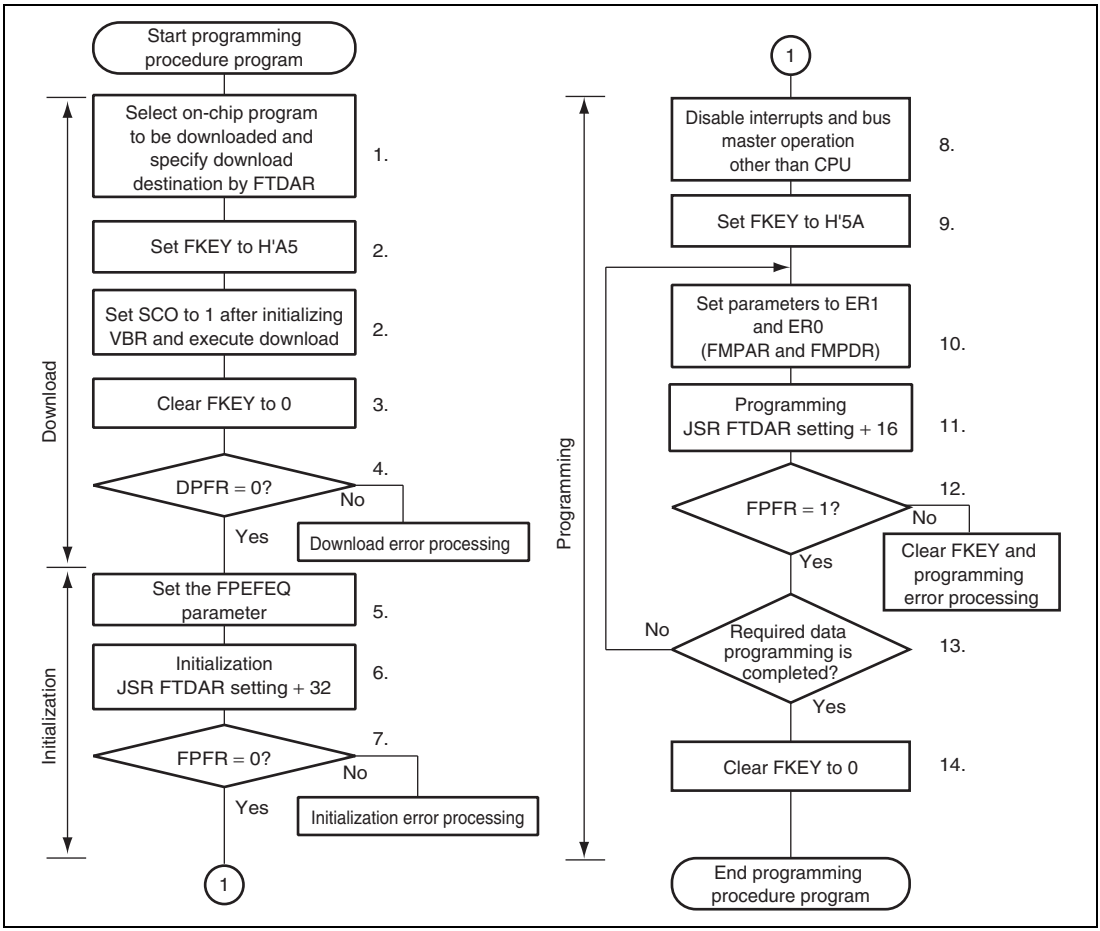


Figure 24.12 Programming Procedure in User Program Mode

The procedure program must be executed in an area other than the flash memory to be programmed. Setting the SCO bit in FCCS to 1 to request download must be executed in the on-chip RAM. The area that can be executed in the steps of the procedure program (on-chip RAM, user MAT, and external space) is shown in section 24.8.4, Storable Areas for On-Chip Program and Program Data. The following description assumes that the area to be programmed on the user MAT is erased and that program data is prepared in the consecutive area.

The program data for one programming operation is always 128 bytes. When the program data exceeds 128 bytes, the start address of the programming destination and program data parameters are updated in 128-byte units and programming is repeated. When the program data is less than 128 bytes, invalid data is filled to prepare 128-byte program data. If the invalid data to be added is H'FF, the program processing time can be shortened.

1. Select the on-chip program to be downloaded and the download destination. When the PPVS bit in FPCS is set to 1, the programming program is selected. Several programming/erasing programs cannot be selected at one time. If several programs are selected, a download error is returned to the SS bit in the DPFR parameter. The on-chip RAM start address of the download destination is specified by FTDAR.
2. Write H'A5 in FKEY. If H'A5 is not written to FKEY, the SCO bit in FCCS cannot be set to 1 to request download of the on-chip program.
   — H'A5 is written to FKEY.
   — Setting the SCO bit is executed in the on-chip RAM.

   When the SCO bit is set to 1, download is started automatically. Since the SCO bit is cleared to 0 when the procedure program is resumed, the SCO bit cannot be confirmed to be 1 in the procedure program. The download result can be confirmed by the return value of the DPFR parameter. To prevent incorrect decision, before setting the SCO bit to 1, set one byte of the on-chip RAM start address specified by FTDAR, which becomes the DPFR parameter, to a value other than the return value (e.g. H'FF). Particular processing that is accompanied by bank switching as described below is performed when download is executed. Dummy read of FCCS must be performed twice immediately after the SCO bit is set to 1.
   — The user-MAT space is switched to the on-chip program storage area.
   — After the program to be downloaded and the on-chip RAM start address specified by FTDAR are checked, they are transferred to the on-chip RAM.
   — FPCS, FECS, and the SCO bit in FCCS are cleared to 0.
   — The return value is set in the DPFR parameter.
   — The values of general registers of the CPU are held.
   — During download, no interrupts can be accepted. However, since the interrupt requests are held, when the procedure program is resumed, the interrupts are requested.

— To hold a level-detection interrupt request, the interrupt must continue to be input until the download is completed.

— Allocate a stack area of 128 bytes at the maximum in the on-chip RAM before setting the SCO bit to 1.

3. FKEY is cleared to H'00 for protection.
4. The download result must be confirmed by the value of the DPFR parameter. Check the value of the DPFR parameter (one byte of start address of the download destination specified by FTDAR). If the value of the DPFR parameter is H'00, download has been performed 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 that before downloading, the setting of the start address of the download destination in FTDAR may be abnormal. In this case, confirm the setting of the TDER bit in FTDAR.

   — If the value of the DPFR parameter is different from that before downloading, check the SS bit or FK bit in the DPFR parameter to confirm the download program selection and FKEY setting, respectively.
5. The operating frequency of the CPU is set in the FPEFEQ parameter for initialization. The settable operating frequency of the FPEFEQ parameter ranges from 8 to 32 MHz. When the frequency is set otherwise, an error is returned to the FPFR parameter of the initialization program and initialization is not performed. For details on setting the frequency, see section 24.7.2 (3), Flash Program/Erase Frequency Parameter (FPEFEQ: General Register ER0 of CPU).
6. Initialization is executed. The initialization program is downloaded together with the programming program to the on-chip RAM. The entry point of the initialization program is at the address which is 32 bytes after #DLTOP (start address of the download destination specified by FTDAR). Call the subroutine to execute initialization by using the following steps.

```
MOV.L #DLTOP+32,ER2      ; Set entry address to ER2
JSR   @ER2               ; Call initialization routine
NOP
```

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

   — R0L is a return value of the FPFR parameter.

   — Since the stack area is used in the initialization program, a stack area of 128 bytes at the maximum must be allocated in RAM.

   — Interrupts can be accepted during execution of the initialization program. Make sure the program storage area and stack area in the on-chip RAM and register values are not overwritten.

7. The return value in the initialization program, the FPFR parameter is determined.
8. All interrupts and the use of a bus master other than the CPU are disabled during programming/erasing. The specified voltage is applied for the specified time when programming or erasing. If interrupts occur or the bus mastership is moved to other than the CPU during programming/erasing, causing a voltage exceeding the specifications to be applied, the flash memory may be damaged. Therefore, interrupts are disabled by setting bit 7 (I bit) in the condition code register (CCR) to B'1 in interrupt control mode 0 and by setting bits 2 to 0 (I2 to I0 bits) in the extend register (EXR) to B'111 in interrupt control mode 2. Accordingly, interrupts other than NMI are held and not executed. Configure the user system so that NMI interrupts do not occur. The interrupts that are held must be executed after all programming completes.
9. FKEY must be set to H'5A and the user MAT must be prepared for programming.
10. The parameters required for programming are set. The start address of the programming destination on the user MAT (FMPAR parameter) is set in general register ER1. The start address of the program data storage area (FMPDR parameter) is set in general register ER0.
    - Example of FMPAR parameter setting: When an address other than one in the user MAT area is specified for the start address of the programming destination, even if the programming program is executed, programming is not executed and an error is returned to the FPFR parameter. Since the program data for one programming operation is 128 bytes, the lower eight bits of the address must be H'00 or H'80 to be aligned with the 128-byte boundary.
    - Example of FMPDR parameter setting: When the storage destination for the program data is flash memory, even if the programming routine is executed, programming is not executed and an error is returned to the FPFR parameter. In this case, the program data must be transferred to the on-chip RAM and then programming must be executed.
11. Programming is executed. The entry point of the programming program is at the address which is 16 bytes after #DLTOP (start address of the download destination specified by FTDAR). Call the subroutine to execute programming 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 bytes at the maximum must be allocated in RAM.

12. The return value in the programming program, the FPFR parameter is determined.
13. Determine whether programming of the necessary data has finished. If more than 128 bytes of data are to be programmed, update the FMPAR and FMPDR parameters in 128-byte units, and repeat steps 11 to 14. Increment the programming destination address by 128 bytes and update the programming data pointer correctly. If an address which has already been programmed is written to again, not only will a programming error occur, but also flash memory will be damaged.
14. After programming finishes, clear FKEY and specify software protection. If this LSI is restarted by a reset immediately after programming has finished, secure the reset input period (period of $\overline{\text{RES}}$ = 0) of at least 100 μs.

(3) Erasing Procedure in User Program Mode

The procedures for download of the on-chip program, initialization, and erasing are shown in figure 24.13.

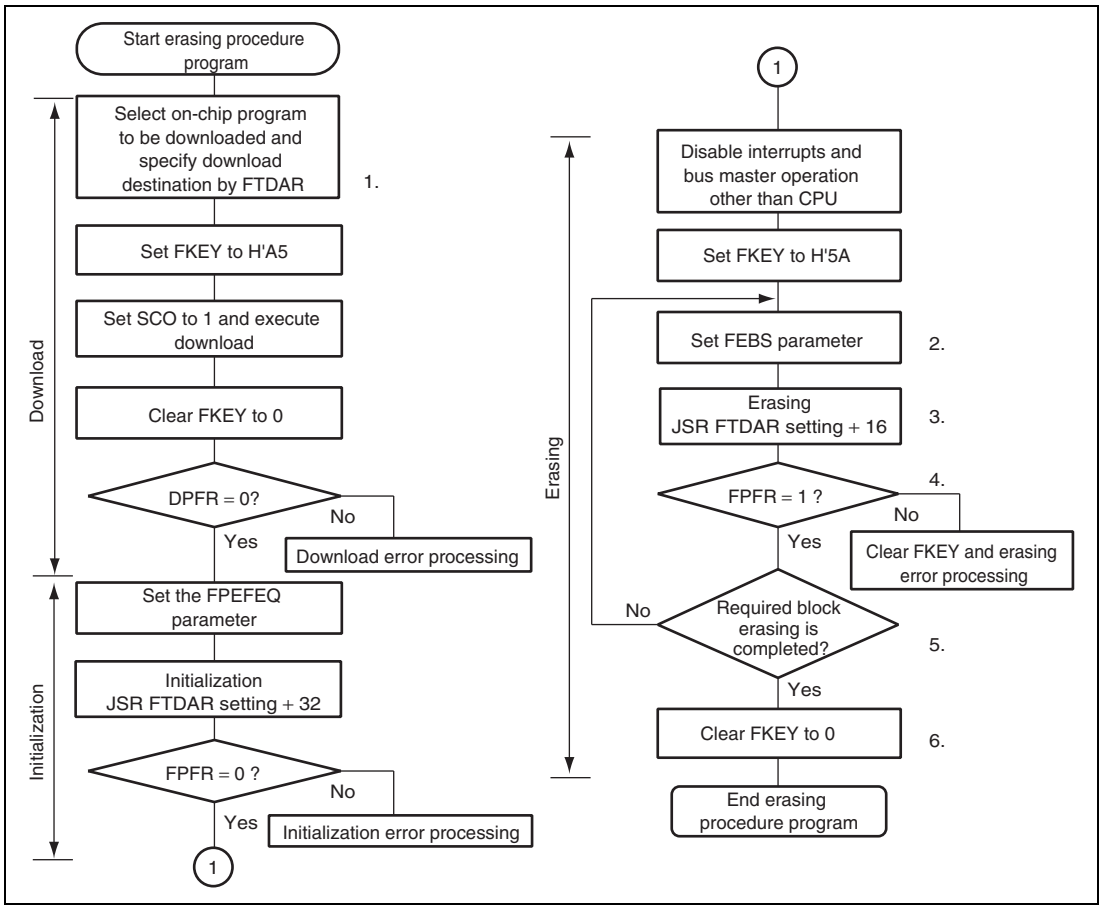


Figure 24.13 Erasing Procedure in User Program Mode

The procedure program must be executed in an area other than the user MAT to be erased. Setting the SCO bit in FCCS to 1 to request download must be executed in the on-chip RAM. The area that can be executed in the steps of the procedure program (on-chip RAM and user MAT) is shown in section 24.8.4, Storable Areas for On-Chip Program and Program Data. For the downloaded on-chip program area, see figure 24.11.

One erasure processing erases one block. For details on block divisions, refer to figure 24.4. To erase two or more blocks, update the erase block number and repeat the erasing processing for each block.

1. Select the on-chip program to be downloaded and the download destination. When the PPVS bit in FPCS is set to 1, the programming program is selected. Several programming/erasing programs cannot be selected at one time. If several programs are selected, a download error is returned to the SS bit in the DPFR parameter. The on-chip RAM start address of the download destination is specified by FTDAR.

   For the procedures to be carried out after setting FKEY, see section 24.8.2 (2), Programming Procedure in User Program Mode.
2. Set the FEBS parameter necessary for erasure. Set the erase block number (FEBS parameter) of the user MAT in general register ER0. If a value other than an erase block number of the user MAT is set, no block is erased even though the erasing program is executed, and an error is returned to the FPFR parameter.
3. Erasure is executed. Similar to as in programming, the entry point of the erasing program is at the address which is 16 bytes after #DLTOP (start address of the download destination specified by FTDAR). Call the subroutine to execute erasure by using the following steps.

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

   - The general registers other than R0L are held in the erasing program.
   - R0L is a return value of the FPFR parameter.
   - Since the stack area is used in the erasing program, a stack area of 128 bytes at the maximum must be allocated in RAM.
4. The return value in the erasing program, the FPFR parameter is determined.
5. Determine whether erasure of the necessary blocks has finished. If more than one block is to be erased, update the FEBS parameter and repeat steps 2 to 5.
6. After erasure completes, clear FKEY and specify software protection. If this LSI is restarted by a reset immediately after erasure has finished, secure the reset input period (period of $\overline{RES}$ = 0) of at least 100 μs.

24.8.3 User Boot Mode

This LSI has user boot mode that is initiated with different mode pin settings than those in boot mode or user program mode. User boot mode is a user-arbitrary boot mode, unlike boot mode that uses the on-chip SCI.

Only the user MAT can be programmed/erased in user boot mode. Programming/erasing of the user boot MAT is only enabled in boot mode or programmer mode.

(1) User Boot Mode Initiation

For the mode pin settings to start up user boot mode, see table 24.7.

When the reset start is executed in user boot mode, the built-in check routine runs. The user MAT 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 boot MAT. At this point, H'AA is set to FMATS because the execution target MAT is the user boot MAT.

(2) User MAT Programming in User Boot Mode

For programming the user MAT in user boot mode, additional processing made by setting FMATS is required: switching from user-boot-MAT selection state to user-MAT selection state, and switching back to user-boot-MAT selection state after programming completes.

Figure 24.14 shows the procedure for programming the user MAT in user boot mode.

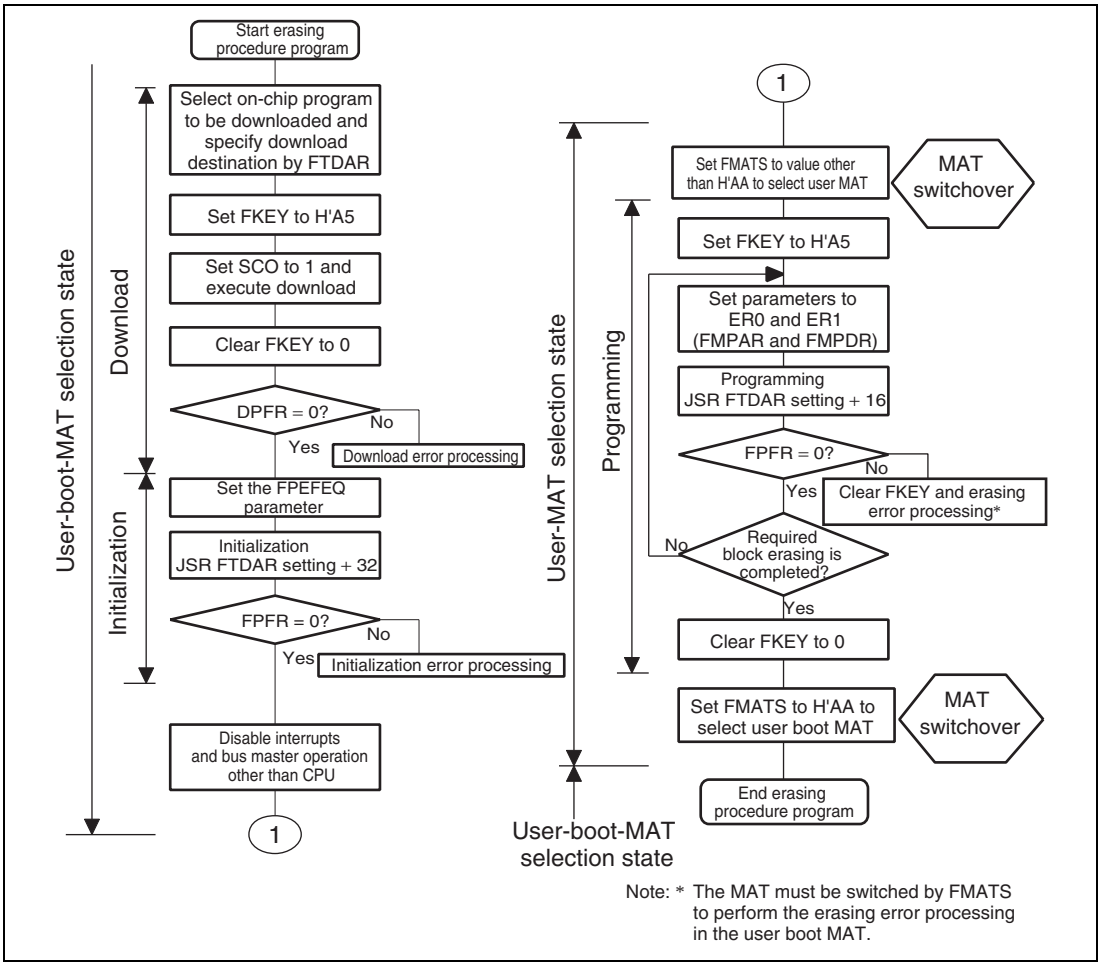


Figure 24.14 Procedure for Programming User MAT in User Boot Mode

The difference between the programming procedures in user program mode and user boot mode is whether the MAT is switched or not as shown in figure 24.14.

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

MAT switching is enabled by writing a specific value to FMATS. Note, however, that while the MATs are being switched, the LSI is in an unstable state, e.g. access to a MAT is not allowed until MAT switching is completed, and if an interrupt occurs, from which MAT the interrupt vector is read is undetermined. Perform MAT switching in accordance with the description in section 24.10, Switching between User MAT and User Boot MAT.

Except for MAT switching, the programming procedure is the same as that in user program mode.

The area that can be executed in the steps of the user procedure program (on-chip RAM and user MAT) is shown in section 24.8.4, Storable Areas for On-Chip Program and Program Data.

(3) User MAT Erasing in User Boot Mode

For erasing the user MAT in user boot mode, additional processing made by setting FMATS are required: switching from user-boot-MAT selection state to user-MAT selection state, and switching back to user-boot-MAT selection state after erasing completes.

Figure 24.15 shows the procedure for erasing the user MAT in user boot mode.

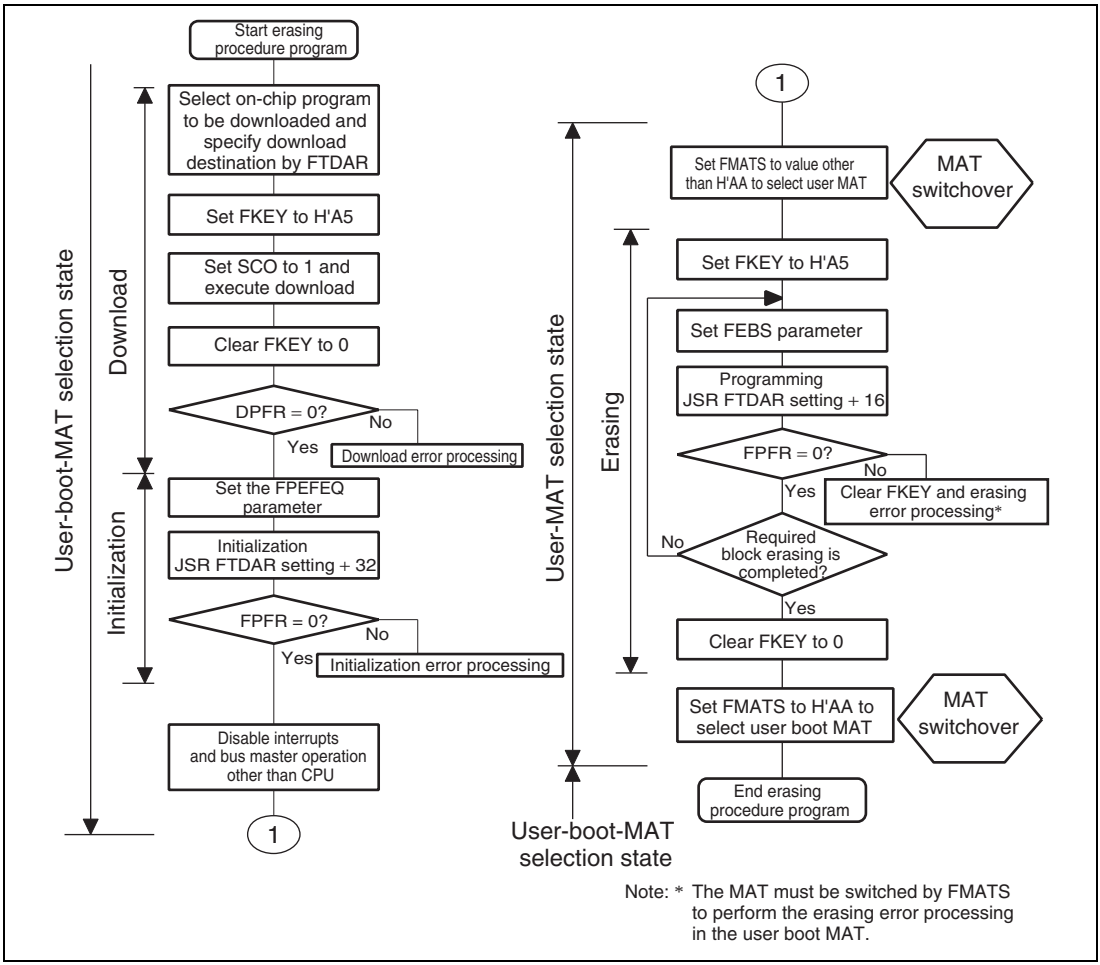


Figure 24.15 Procedure for Erasing User MAT in User Boot Mode

The difference between the erasing procedures in user program mode and user boot mode depends on whether the MAT is switched or not as shown in figure 24.15.

MAT switching is enabled by writing a specific value to FMATS. Note, however, that while the MATs are being switched, the LSI is in an unstable state, e.g. access to a MAT is not allowed until MAT switching is completed, and if an interrupt occurs, from which MAT the interrupt vector is read is undetermined. Perform MAT switching in accordance with the description in section 24.10, Switching between User MAT and User Boot MAT.

Except for MAT switching, the erasing procedure is the same as that in user program mode.

The area that can be executed in the steps of the user procedure program (on-chip RAM and user MAT) is shown in section 24.8.4, Storable Areas for On-Chip Program and Program Data.

24.8.4 Storable Areas for On-Chip Program and Program Data

In the descriptions in this manual, the on-chip programs and program data storage areas are assumed to be in the on-chip RAM. However, they can be executed from part of the flash memory which is not to be programmed or erased as long as the following conditions are satisfied.

- The on-chip program is downloaded to and executed in the on-chip RAM specified by FTDAR. Therefore, this on-chip RAM area is not available for use.
- Since the on-chip program uses a stack area, allocate 128 bytes at the maximum as a stack area.
- Download requested by setting the SCO bit in FCCS to 1 should be executed from the on-chip RAM because it will require switching of the memory MATs.
- In an operating mode in which the external address space is not accessible, such as single-chip mode, the required procedure programs, NMI handling vector table, and NMI handling routine should be transferred to the on-chip RAM before programming/erasing starts (download result is determined).
- The flash memory is not accessible during programming/erasing. Programming/erasing is executed by the program downloaded to the on-chip RAM. Therefore, the procedure program that initiates operation, the NMI handling vector table, and the NMI handling routine should be stored in the on-chip RAM other than the flash memory.
- After programming/erasing starts, access to the flash memory should be inhibited until FKEY is cleared. The reset input state (period of $\overline{\text{RES}}$ = 0) must be set to at least 100 μs when the operating mode is changed and the reset start executed on completion of programming/erasing. Transitions to the reset state are inhibited during programming/erasing. When the reset signal is input, a reset input state (period of $\overline{\text{RES}}$ = 0) of at least 100 μs is needed before the reset signal is released.
- Switching of the MATs by FMATS should be required when programming/erasong of the user MAT is operated in user boot mode. The program that switches the MATs should be executed from the on-chip RAM. (For details, see section 24.10, Switching between User MAT and User Boot MAT.) Make sure you know which MAT is currently selected when witching them.
- When the program data storage area is within the flash memory area, an error will occur even when the data stored is normal program data. Therefore, the data should be transferred to the on-chip RAM to place the address that the FMPDR parameter indicates in an area other than the flash memory.

In consideration of these conditions, the areas in which the program data can be stored and executed are determined by the combination of the processing contents, operating mode, and bank structure of the memory MATs, as shown in tables 24.9 to 24.13.

Table 24.9 Executable Memory MAT

	Operating Mode	
Processing Contents	**User Program Mode**	**User boot Mode***
Programming	See table 24.10.	See table 24.12
Erasing	See table 24.11.	See table 24.13

Note: * Programming/Erasing is possible to the User Mat.

Table 24.10 Usable Area for Programming in User Program Mode

	Storable/Executable Area		Selected MAT	
Item	**On-Chip RAM**	**User MAT**	**User MAT**	**Embedded Program Storage MAT**
Storage area for program data	O	×*	—	—
Operation for selecting on-chip program to be downloaded	O	O	O	
Operation for writing H'A5 to FKEY	O	O	O	
Execution of writing 1 to SCO bit in FCCS (download)	O	×		O
Operation for clearing FKEY	O	O	O	
Decision of download result	O	O	O	
Operation for download error	O	O	O	
Operation for setting initialization parameter	O	O	O	
Execution of initialization	O	×	O	
Decision of initialization result	O	O	O	
Operation for initialization error	O	O	O	
NMI handling routine	O	×	O	
Operation for disabling interrupts	O	O	O	
Operation for writing H'5A to FKEY	O	O	O	
Operation for setting programming parameter	O	×	O	
Execution of programming	O	×	O	
Decision of programming result	O	×	O	
Operation for programming error	O	×	O	
Operation for clearing FKEY	O	×	O	

Note: * Transferring the program data to the on-chip RAM beforehand enables this area to be used.

Table 24.11 Usable Area for Erasure in User Program Mode

	Storable/Executable Area		Selected MAT	
Item	**On-Chip RAM**	**User MAT**	**User MAT**	**Embedded Program Storage MAT**
Operation for selecting on-chip program to be downloaded	O	O	O	
Operation for writing H'A5 to FKEY	O	O	O	
Execution of writing 1 to SCO bit in FCCS (download)	O	×		O
Operation for clearing FKEY	O	O	O	
Decision of download result	O	O	O	
Operation for download error	O	O	O	
Operation for setting initialization parameter	O	O	O	
Execution of initialization	O	×	O	
Decision of initialization result	O	O	O	
Operation for initialization error	O	O	O	
NMI handling routine	O	×	O	
Operation for disabling interrupts	O	O	O	
Operation for writing H'5A to FKEY	O	O	O	
Operation for setting erasure parameter	O	×	O	
Execution of erasure	O	×	O	
Decision of erasure result	O	×	O	
Operation for erasure error	O	×	O	
Operation for clearing FKEY	O	×	O	

Table 24.12 Usable Area for Programming in User Boot Mode

	Storable/Executable Area		Selected MAT		
Item	**On-chip RAM**	**User Boot MAT**	**User MAT**	**User Boot MAT**	**Embedded Program Storage MAT**
Storage area for program data	O	×*[1]	—	—	—
Selecting on-chip program to be downloaded	O	O		O	
Writing H'A5 to FKEY	O	O		O	
Writing 1 to SCO in FCCS (download)	O	×			O
FKEY clearing	O	O		O	
Determination of download result	O	O		O	
Download error processing	O	O		O	
Setting initialization parameter	O	O		O	
Initialization	O	×		O	
Determination of initialization result	O	O		O	
Initialization error processing	O	O		O	
NMI handling routine	O	×		O	
Disabling interrupts	O	O		O	
Switching MATs by FMATS	O	×	O		
Writing H'5A to FKEY	O	×	O		
Setting programming parameter	O	×	O		
Programming	O	×	O		
Determination of programming result	O	×	O		
Programming error processing	O	×*[2]	O		
FKEY clearing	O	×	O		
Switching MATs by FMATS	O	×		O	

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

Table 24.13 Usable Area for Erasure in User Boot Mode

	Storable/Executable Area		Selected MAT		
Item	On-chip RAM	User Boot MAT	User MAT	User Boot MAT	Embedded Program Storage MAT
Selecting on-chip program to be downloaded	O	O		O	
Writing H'A5 to FKEY	O	O		O	
Writing 1 to SCO in FCCS (download)	O	×			O
FKEY clearing	O	O		O	
Determination of download result	O	O		O	
Download error processing	O	O		O	
Setting initialization parameter	O	O		O	
Initialization	O	×		O	
Determination of initialization result	O	O		O	
Initialization error processing	O	O		O	
NMI handling routine	O	×		O	
Disabling interrupts	O	O		O	
Switching MATs by FMATS	O	×		O	
Writing H'5A to FKEY	O	×	O		
Setting erasure parameter	O	×	O		
Erasure	O	×	O		
Determination of erasure result	O	×	O		
Erasing error processing	O	×*	O		
FKEY clearing	O	×	O		
Switching MATs by FMATS	O	×	O		

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

24.9 Protection

There are three types of protection against the flash memory programming/erasing: hardware protection, software protection, and error protection.

24.9.1 Hardware Protection

Programming and erasure of the flash memory is forcibly disabled or suspended by hardware protection. In this state, download of an on-chip program and initialization are possible. However, programming or erasure of the user MAT cannot be performed even if the programming/erasing program is initiated, and the error in programming/erasing is indicated by the FPFR parameter.

Table 24.14 Hardware Protection

		Function to be Protected	
Item	**Description**	**Download**	**Programming/ Erasing**
Reset protection	• The programming/erasing interface registers are initialized in the reset state (including a reset by the WDT) and the programming/erasing protection 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 settled after a power is initially supplied. In the case of a reset during operation, hold the $\overline{\text{RES}}$ pin low for the $\overline{\text{RES}}$ pulse width given in the AC characteristics. If a reset is input during programming or erasure, data in the flash memory is not guaranteed. In this case, execute erasure and then execute programming again.	O	O

24.9.2 Software Protection

The software protection protects the flash memory against programming/erasing by disabling download of the programming/erasing program and using the key code.

Table 24.15 Software Protection

		Function to be Protected	
Item	**Description**	**Download**	**Programming/ Erasing**
Protection by SCO bit	The programming/erasing protection state is entered when the SCO bit in FCCS is cleared to 0 to disable download of the programming/erasing programs.	O	O
Protection by FKEY	The programming/erasing protection state is entered because download and programming/erasing are disabled unless the required key code is written in FKEY.	O	O

24.9.3 Error Protection

Error protection is a mechanism for aborting programming or erasure when a CPU runaway occurs or operations not according to the programming/erasing procedures are detected during programming/erasing of the flash memory. Aborting programming or erasure in such cases prevents damage to the flash memory due to excessive programming or erasing.

If an error occurs during programming/erasing of the flash memory, the FLER bit in FCCS is set to 1 and the error protection state is entered.

- When an interrupt request, such as NMI, occurs during programming/erasing.
- When the flash memory is read from during programming/erasing (including a vector read or an instruction fetch).
- When a SLEEP instruction is executed (including software-standby mode) during programming/erasing.

Error protection is canceled by a reset. Note that the reset should be released after the reset input period of at least 100µs has passed. Since high voltages are applied during programming/erasing of the flash memory, some voltage may remain after the error protection state has been entered. For this reason, it is necessary to reduce the risk of damaging the flash memory by extending the reset input period so that the charge is released.

The state-transition diagram in figure 24.16 shows transitions to and from the error protection state.

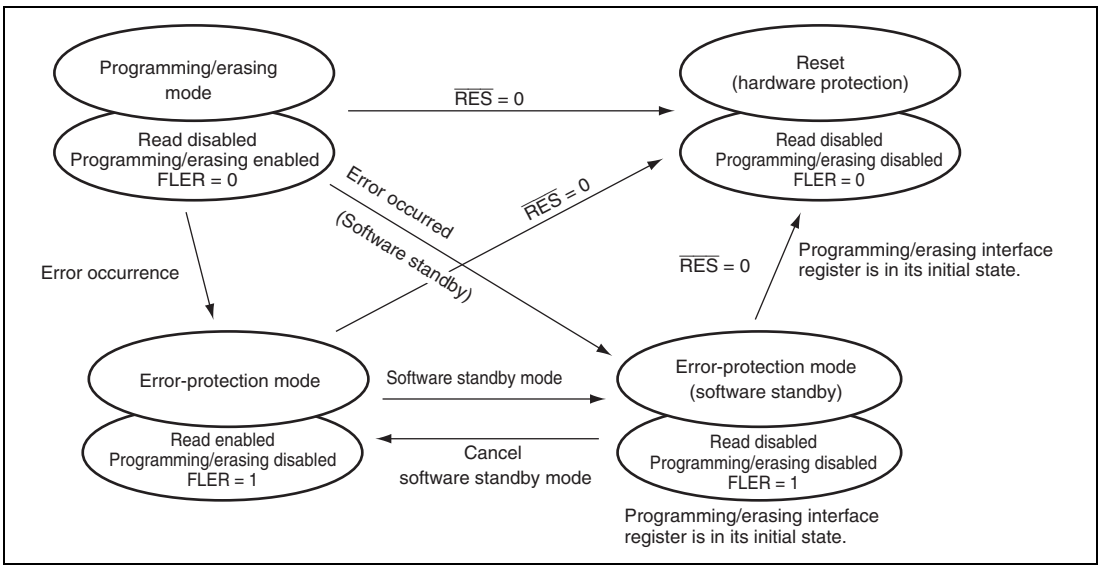


Figure 24.16 Transitions to Error Protection State

24.10 Switching between User MAT and User Boot MAT

It is possible to switch between the user MAT and user boot MAT. However, the following procedure is required because both of these MATs are allocated to address 0.
(Switching to the user boot MAT disables programming and erasing. Programming of the user boot MAT should take place in boot mode or programmer mode.)

1. MAT switching by FMATS should always be executed from the on-chip RAM.
2. To ensure that switching has finished and access is made to the newly switched MAT, execute four NOP instructions in the same on-chip RAM immediately after writing to FMATS (this prevents access to the flash memory during MAT switching).
3. If an interrupt has occurred during switching, there is no guarantee of which memory MAT is being accessed.

   Always mask the maskable interrupts before switching between MATs. In addition, configure the system so that NMI interrupts do not occur during MAT switching.
4. After the MATs have been switched, take care because the interrupt vector table will also have been switched.
5. Memory sizes of the user MAT and user boot MAT are different. Do not access a user boot MAT in a space of 8 kbytes or more. If access goes beyond the 8-kbyte space, the values read are undefined.

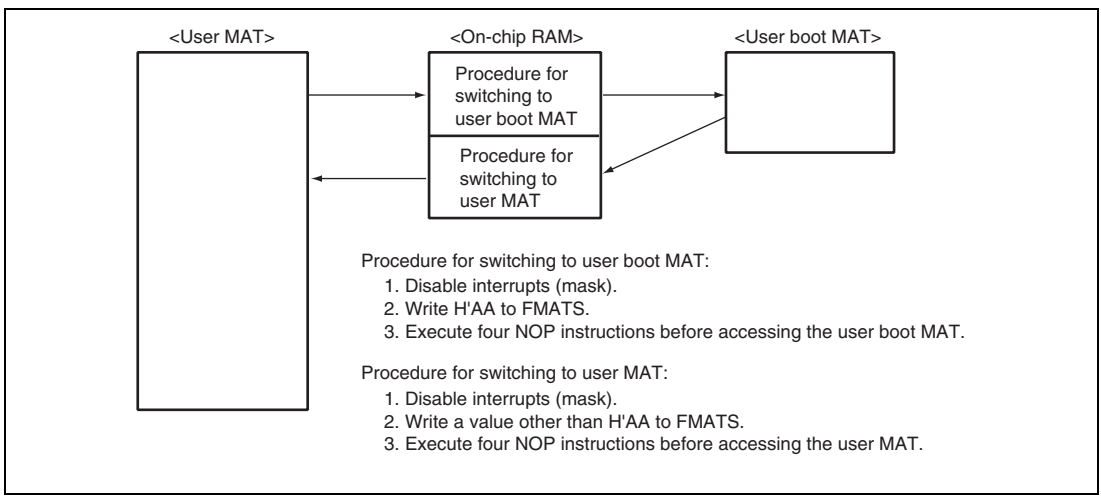


Figure 24.17 Switching between User MAT and User Boot MAT

24.11 Programmer Mode

Along with its on-board programming mode, this LSI also has a programmer mode as a further mode for the writing and erasing of programs and data. In programmer mode, a general-purpose PROM programmer that supports the device types shown in table 24.16 can be used to write programs to the on-chip ROM without any limitation.

Table 24.16 Device Types Supported in Programmer Mode

Target Memory MAT	Size	Device Type
User MAT	256 kbytes*	FZTAT256V3A
User boot MAT	8 kbytes	FZTATUSBTV3A

Note: * For the R4F2117R model, 160 kbytes of ROM space is available when the user MAT is selected. If programming is performed in programmer mode, H'FF data must be written to address H'28000 to H'3FFFF with 256-kbyte capacity setting.

24.12 Standard Serial Communication Interface Specifications for Boot Mode

The boot program initiated in boot mode performs serial communication using the host and on-chip SCI_1. The serial communication interface specifications are shown below.

The boot program has three states.

1. Bit-rate-adjustment state

   In this state, the boot program adjusts the bit rate to achieve serial communication with the host. Initiating boot mode enables starting of the boot program and entry to the bit-rate-adjustment state. The program receives the command from the host to adjust the bit rate. After adjusting the bit rate, 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 device name, clock mode, and bit rate are selected. After selection of these settings, the program is made to enter the programming/erasing state by the command for a transition to the programming/erasing state. The program transfers the libraries required for erasure to the on-chip RAM and erases the user MATs before the transition.

3. Programming/erasing state

   Programming and erasure by the boot program take place in this state. The boot program is made to transfer the programming/erasing programs to the on-chip RAM by commands from the host. Sum checks and blank checks are executed by sending these commands from the host.

These boot program states are shown in figure 24.18.

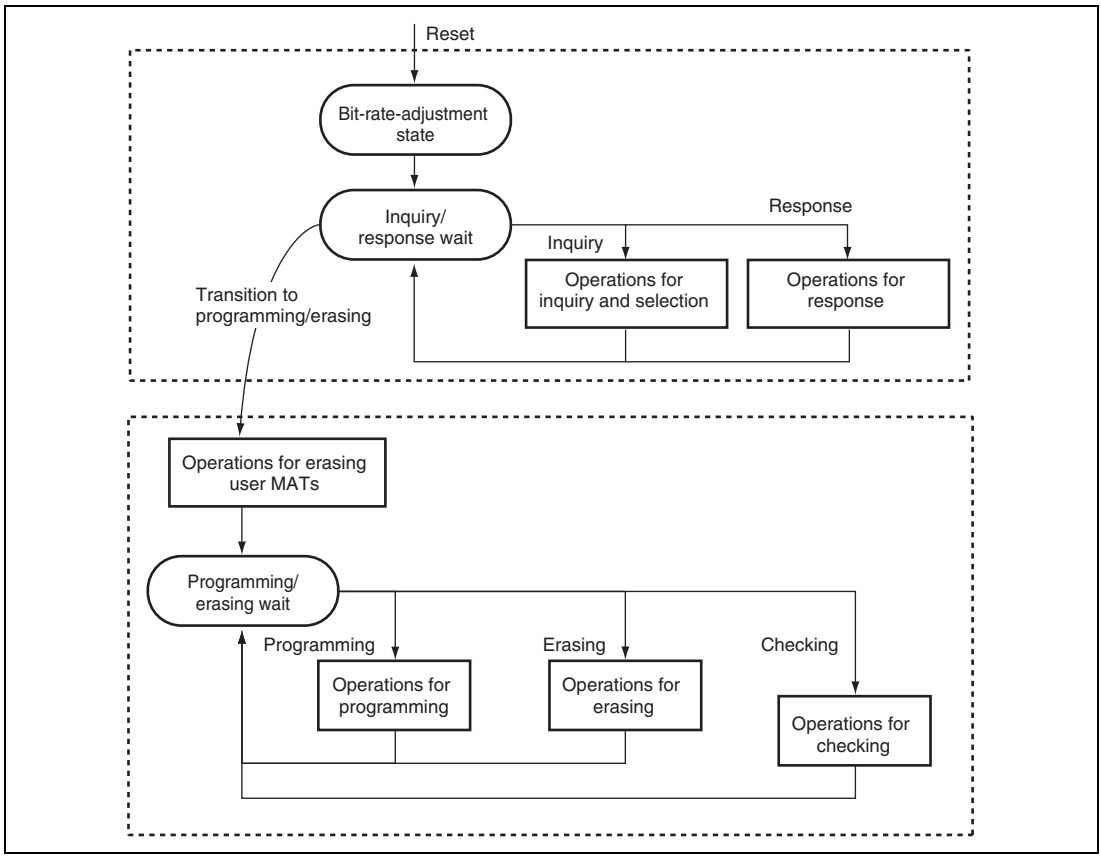


Figure 24.18 Boot Program States

(1) Bit-Rate-Adjustment State

The bit rate is calculated by measuring the period of transfer of a low-level byte (H'00) from the host. The bit rate can be changed by the command for a new bit rate selection. After the bit rate has been adjusted, the boot program enters the inquiry and selection state. The bit-rate-adjustment sequence is shown in figure 24.19.

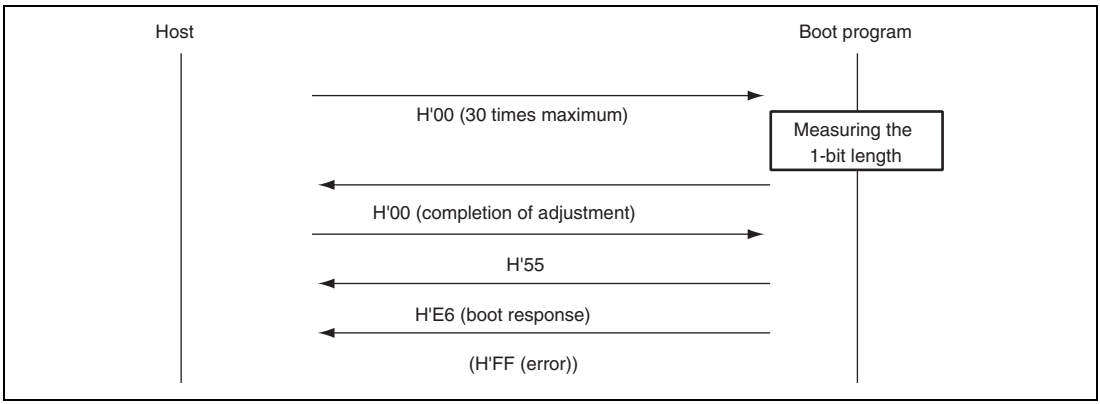


Figure 24.19 Bit-Rate-Adjustment Sequence

(2) Communications Protocol

After adjustment of the bit rate, the protocol for serial communications between the host and the boot program is as shown below.

1. One-byte commands and one-byte responses

   These one-byte commands and one-byte responses consist of the inquiries and the ACK for successful completion.
2. n-byte commands or n-byte responses

   These commands and responses are comprised of n bytes of data. These are selections and responses to inquiries.

   The program data size is not included under this heading because it is determined in another command.
3. Error response

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

4. Programming of 128 bytes

   The size is not specified in commands. The size of n is indicated in response to the programming unit inquiry.

5. Memory read response

   This response consists of four bytes of data.

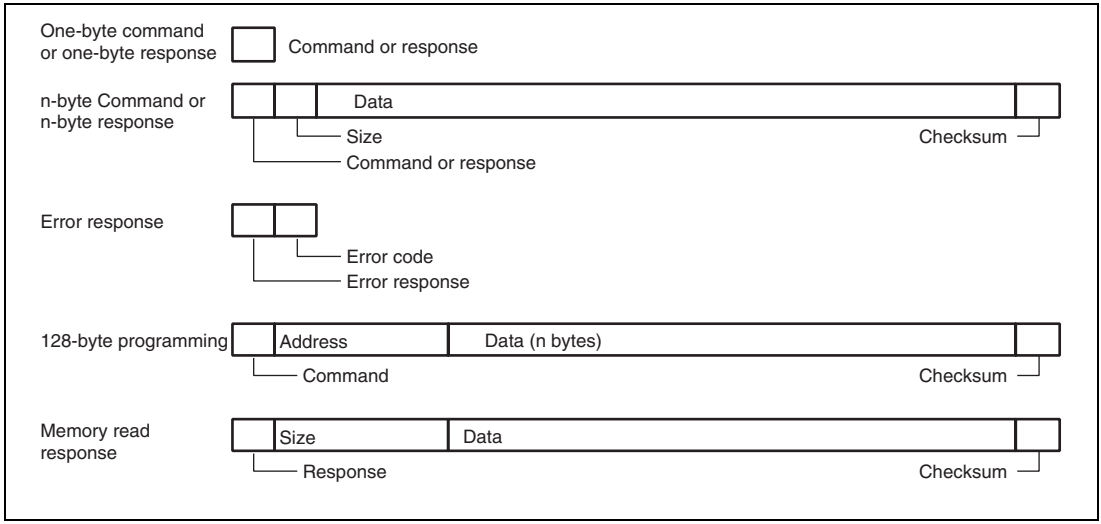


Figure 24.20 Communication Protocol Format

- Command (one byte): Commands including inquiries, selection, programming, erasing, and checking
- Response (one byte): Response to an inquiry
- Size (one byte): The amount of data for transmission excluding the command, amount of data, and checksum
- Checksum (one byte): The checksum is calculated so that the total of all values from the command byte to the SUM byte becomes H'00.
- Data (n bytes): Detailed data of a command or response
- Error response (one byte): Error response to a command
- Error code (one byte): Type of the error
- Address (four bytes): Address for programming
- Data (n bytes): Data to be programmed (the size is indicated in the response to the programming unit inquiry.)
- Size (four bytes): Four-byte response to a memory read

(3) Inquiry and Selection States

The boot program returns information from the flash memory in response to the host's inquiry commands and sets the device code, clock mode, and bit rate in response to the host's selection command.

Table 24.17 lists the inquiry and selection commands.

Table 24.17 Inquiry and Selection Commands

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 modes and values of each mode
H'11	Clock mode selection	Indication of the selected clock mode
H'22	Division ratio inquiry	Inquiry regarding the number of frequency-divided clock types, the number of division ratios and the values of each division
H'23	Operating clock frequency inquiry	Inquiry regarding the maximum and minimum values of the main clock and peripheral clocks
H'24	User boot MAT information inquiry	Inquiry regarding the a number of user boot MATs and the start and last addresses of each MAT
H'25	User MAT information inquiry	Inquiry regarding the a number of user MATs and the start and last addresses of each MAT
H'26	Block for erasing information Inquiry	Inquiry regarding the number of blocks and the start and last addresses of each block
H'27	Programming unit inquiry	Inquiry regarding the unit of program data
H'3F	New bit rate selection	Selection of new bit rate
H'40	Transition to programming/erasing state	Erasing of user MATs or user boot MATs, and entry to programming/erasing state
H'4F	Boot program status inquiry	Inquiry into the operated status of the boot program

The selection commands, which are device selection (H'10), clock mode selection (H'11), and new bit rate selection (H'3F), should be sent from the host in that order. When two or more selection commands are sent at once, the last command will be valid.

All of these commands, except for the boot program status inquiry command (H'4F), will be valid until the boot program receives the programming/erasing transition (H'40). The host can choose the needed commands and make inquiries while the above commands are being transmitted. H'4F is valid even after the boot program has received H'40.

(a) Supported Device Inquiry

The boot program will return the device codes of supported devices and the product code in response to the supported device inquiry.

Command

H'20

- Command, H'20, (one byte): Inquiry regarding supported devices

Response

H'30	Size	Number of devices	
Number of characters	Device code		Product name
…			
SUM			

- Response, H'30, (one byte): Response to the supported device inquiry
- Size (one byte): Number of bytes to be transmitted, excluding the command, size, and checksum, that is, the amount of data contributes by the number of devices, characters, device codes and product names
- Number of devices (one byte): The number of device types supported by the boot program
- Number of characters (one byte): The number of characters in the device codes and boot program's name
- Device code (four bytes): ASCII code of the supporting product
- Product name (n bytes): Type name of the boot program in ASCII-coded characters
- SUM (one byte): Checksum

  The checksum is calculated so that the total number of all values from the command byte to the SUM byte becomes H'00.

(b) Device Selection

The boot program will set the supported device to the specified device code. The program will return the selected device code in response to the inquiry after this setting has been made.

Command	H'10	Size	Device code	SUM

- Command, H'10, (one byte): Device selection
- Size (one byte): Amount of device-code data
  This is fixed at 4
- Device code (four bytes): Device code (ASCII code) returned in response to the supported device inquiry
- SUM (one byte): Checksum

Response	H'06

- Response, H'06, (one 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, (one byte): Error response to the device selection command
  ERROR : (one 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 inquiry.

Command	H'21

- Command, H'21, (one byte): Inquiry regarding clock mode

Response	H'31	Size	Number of modes	Mode	…	SUM

- Response, H'31, (one byte): Response to the clock-mode inquiry
- Size (one byte): Amount of data that represents the number of modes and modes
- Number of clock modes (one byte): The number of supported clock modes
  H'00 indicates no clock mode or the device allows to read the clock mode.
- Mode (one byte): Values of the supported clock modes (i.e. H'01 means clock mode 1.)
- SUM (one byte): Checksum

(d) Clock Mode Selection

The boot program will set the specified clock mode. The program will return the selected clock-mode information after this setting has been made.

The clock-mode selection command should be sent after the device-selection commands.

Command	H'11	Size	Mode	SUM

- Command, H'11, (one byte): Selection of clock mode
- Size (one byte): Amount of data that represents the modes
- Mode (one byte): A clock mode returned in reply to the supported clock mode inquiry.
- SUM (one byte): Checksum

Response	H'06

- Response, H'06, (one 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, (one byte): Error response to the clock mode selection command
- ERROR : (one 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 mode must be selected using these respective values.

(e) Division Ratio Inquiry

The boot program will return the supported division ratios in response to the inquiry.

Command

H'22

- Command, H'22, (one byte): Inquiry regarding division ratio

Response

H'32	Size	Number of types					
Number of division ratios	Division ratio	…					
…							
SUM							

- Response, H'32, (one byte): Response to the division ratio inquiry
- Size (one byte): The total amount of data that represents the number of types, the number of division ratios, and the division ratios
- Number of types (one byte): The number of supported divided clock types
  (e.g. when there are two divided clock types, which are the main and peripheral clocks, the number of types will be H'02.)
- Number of division ratios (one byte): The number of division ratios for each type
  (e.g. the number of division ratios to which the main clock can be set and the peripheral clock can be set.)
- Division ratio (one byte)

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

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

(f) Operating Clock Frequency Inquiry

The boot program will return the number of operating clock frequencies, and the maximum and minimum values.

Command

H'23

- Command, H'23, (one byte): Inquiry regarding operating clock frequencies

Response

H'33	Size	Number of operating clock frequencies
Minimum value of operating clock frequency		Maximum value of operating clock frequency
…		
SUM		

- Response, H'33, (one byte): Response to operating clock frequency inquiry
- Size (one byte): The number of bytes that represents the minimum values, maximum values, and the number of frequencies.
- Number of operating clock frequencies (one byte): The number of supported operating clock frequency types
  (e.g. when there are two operating clock frequency types, which are the main and peripheral clocks, the number of types will be H'02.)
- Minimum value of operating clock frequency (two bytes): The minimum value of the multiplied or divided clock frequency.

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

  There are as many pairs of minimum and maximum values as there are operating clock frequencies.
- SUM (one byte): Checksum

(g) User Boot MAT Information Inquiry

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

Command

H'24

- Command, H'24, (one byte): Inquiry regarding user boot MAT information

Response

H'34	Size	Number of areas	
Start address area			Last address area
…			
SUM			

- Response, H'34, (one byte): Response to the user boot MAT information inquiry
- Size (one byte): The number of bytes that represents the number of areas, area-start address and area-last address
- Number of areas (one byte): The number of consecutive user boot MAT areas
  When the user boot MAT areas are consecutive, the number of areas is H'01.
- Area-start address (four bytes): Start address of the area
- Area-last address (four bytes): Last address of the area
  There are as many groups of data representing the start and last addresses as there are areas.
- SUM (one byte): Checksum

(h) User MAT Information Inquiry

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

Command

H'25

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

Response

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

- Response, H'35, (one byte): Response to the user MAT information inquiry
- Size (one byte): The number of bytes that represents the number of areas, area-start address and area-last address
- Number of areas (one byte): The number of consecutive user MAT areas
  When the user MAT areas are consecutive, the number of areas is H'01.
- Area-start address (four bytes): Start address of the area
- Area-last address (four bytes): Last address of the area
  There are as many groups of data representing the start and last addresses as there are areas.
- SUM (one byte): Checksum

(i) Erased Block Information Inquiry

The boot program will return the number of erased blocks and their addresses.

Command

H'26

- Command, H'26, (two bytes): Inquiry regarding erased block information

Response

H'36	Size	Number of blocks	
Block start address			Block last address
…			
SUM			

- Response, H'36, (one byte): Response to the number of erased blocks and addresses
- Size (three bytes): The number of bytes that represents the number of blocks, block-start addresses, and block-last addresses.
- Number of blocks (one byte): The number of erased blocks
- Block start address (four bytes): Start address of a block

- Block last Address (four bytes): Last address of a block
  There are as many groups of data representing the start and last addresses as there are areas.
- SUM (one byte): Checksum

(j) Programming Unit Inquiry

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

Command

H'27

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

Response

H'37	Size	Programming unit	SUM

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

(k) New Bit-Rate Selection

The boot program will set a new bit rate and return the new bit rate.

This selection should be sent after sending the clock mode selection command.

Command

H'3F	Size	Bit rate	Input frequency
Number of division ratios	Division ratio 1	Division ratio 2	
SUM			

- Command, H'3F, (one byte): Selection of new bit rate
- Size (one byte): The total number of bytes that represents the bit rate, input frequency, number of division ratios, and division ratio
- Bit rate (two bytes): New bit rate
  One hundredth of the value (e.g. when the value is 19200 bps, it will be 192, which is H'00C0.)
- Input frequency (two bytes): Frequency of the clock input to the boot program
  This is valid to the hundredths place and represents the value in MHz multiplied by 100. (E.g. when the value is 20.00 MHz, it will be 2000, which is H'07D0.)

- Number of division ratios (one byte): The number of division ratios to which the device can be set.
  There are usually two division ratios, which are the main and peripheral module operating frequencies.
- Division ratio 1 (one byte): The value of division ratios for the main operating frequency
  Division ratio: The inverse of the division ratio, as a negative number (e.g. when the clock frequency is divided by two, the value of division ratio will be H'FE. H'FE = D'-2)
- Division ratio 2 (one byte): The value of division ratios for the peripheral frequency
  (Division ratio: The inverse of the division ratio, as a negative number (E.g. when the clock is divided by two, the value of division ratio will be H'FE. H'FE = D'-2)
- SUM (one byte): Checksum

Response

H'06

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

Error Response

H'BF	ERROR

- Error response, H'BF, (one byte): Error response to selection of new bit rate
- ERROR: (one byte): Error code

H'11:	Sum checking error
H'24:	Bit-rate selection error The rate is not available.
H'25:	Error in input frequency This input frequency is not within the specified range.
H'26:	Division ratio error The ratio does not match an available ratio.
H'27:	Operating frequency error The frequency is not within the specified range.

(4) Receive Data Check

The methods for checking of receive data are listed below.

1. Input frequency

   The received value of the input frequency is checked to ensure that it is within the range of minimum to maximum frequencies which matches the clock modes of the specified device. When the value is out of this range, an input-frequency error is generated.

2. Division ratio

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

3. Operating frequency error

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

   Operating frequency = Input frequency ÷ Division ratio

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

4. Bit rate

   To facilitate error checking, the value (n) of clock select (CKS) in the serial mode register (SMR), and the value (N) in the bit rate register (BRR), which are found from the peripheral operating clock frequency (ϕ) and bit rate (B), are used to calculate the error rate to ensure that it is less than 4%. If the error is more than 4%, a bit rate error is generated. The error is calculated using the following expression:

$$\text{Error (\%)} = \{[\frac{\phi \times 10^6}{(N + 1) \times B \times 64 \times 2^{(2 \times n - 1)}}] - 1\} \times 100$$

When the new bit rate is selectable, the rate will be set in the register after sending ACK in response. The host will send an ACK with the new bit rate for confirmation and the boot program will response with that rate.

Confirmation	H'06

- Confirmation, H'06, (one byte): Confirmation of a new bit rate

Response	H'06

- Response, H'06, (one byte): Response to confirmation of a new bit rate

The sequence of new bit-rate selection is shown in figure 24.21.

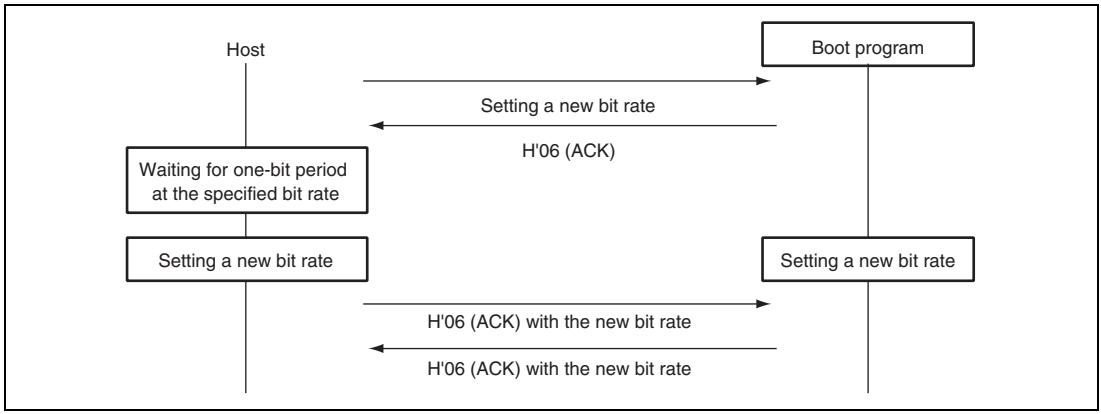


Figure 24.21 New Bit-Rate Selection Sequence

(5) Transition to Programming/Erasing State

The boot program will transfer the erasing program and erase the data in the user MATs first , then the data in the user boot MATs. On completion of this erasure, ACK will be returned and the program will enter the programming/erasing state.

The host should select the device code, clock mode, and new bit rate with device selection, clock-mode selection, and new bit-rate selection commands, and then send the command for the transition to programming/erasing state. These procedures should be carried out before sending of the programming selection command or program data.

Command	H'40

- Command, H'40, (one byte): Transition to programming/erasing state

Response	H'06

- Response, H'06, (one byte): Response to transition to programming/erasing state
  The boot program will send ACK when the user MATs and the user boot MATs have been erased by the transferred erasing program.

Error Response	H'C0	H'51

- Error response, H'C0, (one byte): Error response to the bland check of the user boot MATs
- Error code, H'51, (one byte): Erasing error
  An error occurred and erasure was not completed.

(6) Command Error

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

Error Response	H'80	H'xx

- Error response, H'80, (one byte): Command error
- Command, H'xx, (one byte): Received command

(7) Command Order

The order for commands in the inquiry selection state is shown below.

1. A supported device inquiry (H'20) should be made to inquire about the supported devices.
2. The device should be selected from among those described by the returned information and set with a device-selection (H'10) command.
3. A clock-mode inquiry (H'21) should be made to inquire about the supported clock modes.
4. The clock mode should be selected from among those described by the returned information and set.
5. After selection of the device and clock mode, inquiries for other required information should be made, such as the division-ratio inquiry (H'22) or operating frequency inquiry (H'23), 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, according to the returned information on division ratios and operating frequencies.
7. After selection of the device and clock mode, the information of the user boot MAT and the user MAT should be made to inquire about the user boot MATs information inquiry (H'24), user MATs information inquiry (H'25), erased block information inquiry (H'26), and programming 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.

(8) Programming/Erasing State

A programming selection command makes the boot program select the programming method, a 128-byte programming command makes it program the memory with data, and an erasing selection command and block erasing command make it erase the block. Table 24.18 lists the programming/erasing commands.

Table 24.18 Programming/Erasing Commands

Command	Command Name	Description
H'42	User boot MAT programming selection	Transfers the user boot MAT programming program
H'43	User MAT programming selection	Transfers the user MAT programming 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 boot MAT
H'4B	User MAT sum check	Checks the checksum of the user MAT
H'4C	User boot MAT blank check	Checks the blank data of the user boot MAT
H'4D	User MAT blank check	Checks the blank data of the user MAT
H'4F	Boot program status inquiry	Inquires into the boot program's status

1. Programming

   Programming is executed by the programming selection and 128-byte programming commands.

   Firstly, the host should send the programming selection command.

   After issuing the programming selection command, the host should send the 128-byte programming command. The 128-byte programming command that follows the selection command represents the data programmed according to the method specified by the selection command. When more than 128-byte data is programmed, 128-byte commands should repeatedly be executed. Sending a 128-byte programming command with H'FFFFFFFF as the address will stop the programming. On completion of programming, the boot program will wait for selection of programming or erasing.

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

   The sequence for the programming selection and 128-byte programming commands is shown in figure 24.22.

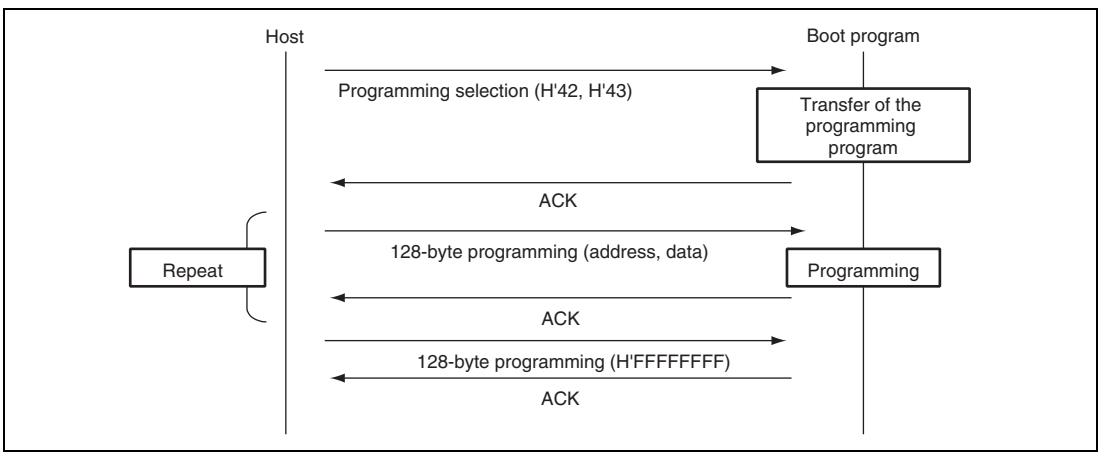


Figure 24.22 Programming Sequence

2. Erasure

Erasure is executed by the erasure selection and block erasure commands.

Firstly, erasure is selected by the erasure selection command and the boot program then erases the specified block. The command should be repeatedly executed if two or more blocks are to be erased. Sending a block erasure command from the host with the block number H'FF will stop the erasure operating. On completion of erasing, the boot program will wait for selection of programming or erasing.

The sequence for the erasure selection and block erasure commands is shown in figure 24.23.

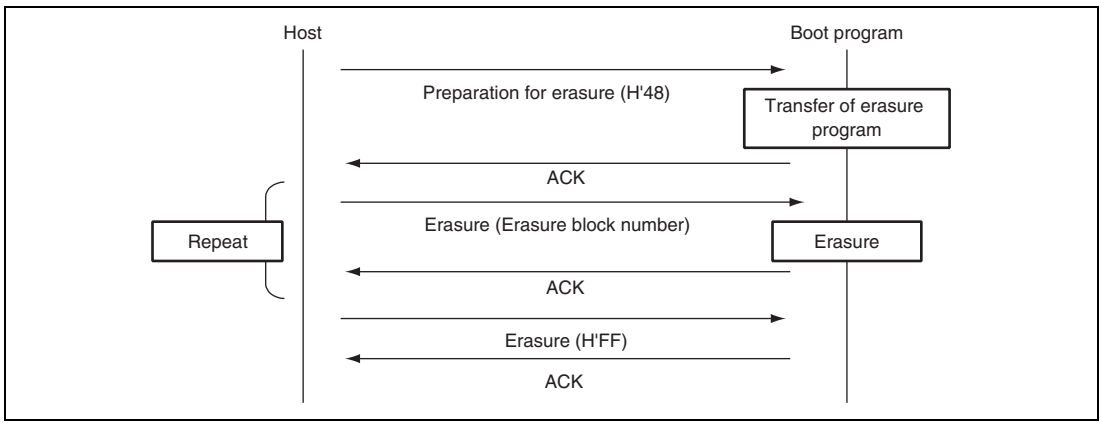


Figure 24.23 Erasure Sequence

3. Programming/Erasing State Information

(a) User Boot MAT Programming Selection

The boot program will transfer a program for user boot MAT programming selection. The data is programmed to the user boot MATs by the transferred program for programming.

Command	H'42

- Command, H'42, (one byte): User boot MAT programming selection

Response	H'06

- Response, H'06, (one byte): Response to user boot MAT programming selection
  When the programming program has been transferred, the boot program will return ACK.

Error Response	H'C2	ERROR

- Error response: H'C2 (1 byte): Error response to user boot MAT programming selection
- ERROR : (1 byte): Error code
  H'54: Selection processing error (transfer error occurs and processing is not completed)

(b) User MAT Programming Selection

The boot program will transfer a program for user MAT programming selection. The data is programmed to the user MATs by the transferred program for programming.

Command	H'43

- Command, H'43, (one byte): User-program programming selection

Response	H'06

- Response, H'06, (one byte): Response to user-program programming selection
  When the programming program has been transferred, the boot program will return ACK.

Error Response	H'C3	ERROR

- Error response : H'C3 (1 byte): Error response to user-program programming selection
- ERROR : (1 byte): Error code
  H'54: Selection processing error (transfer error occurs and processing is not completed)

(c) 128-Byte Programming

The boot program will use the programming program transferred by the programming selection to program the user MATs in response to 128-byte programming.

Command	H'50	Address						
	Data	…						
	…							
	SUM							

- Command, H'50, (one byte): 128-byte programming
- Programming Address (four bytes): Start address for programming
  Multiple of the size specified in response to the programming unit inquiry
  (i.e. H'00, H'01, H'00, H'00 : H'00010000)
- Program data (128 bytes): Data to be programmed
  The size is specified in the response to the programming unit inquiry.
- SUM (one byte): Checksum

Response	H'06

- Response, H'06, (one byte): Response to 128-byte programming
  On completion of programming, the boot program will return ACK.

Error Response	H'D0	ERROR

- Error response, H'D0, (one byte): Error response for 128-byte programming
- ERROR: (one byte): Error code

  H'11: Checksum Error

  H'53: Programming error

  A programming error has occurred and programming cannot be continued.

The specified address should match the unit for programming of data. For example, when the programming is in 128-byte units, the lower eight bits of the address should be H'00 or H'80. When there are less than 128 bytes of data to be programmed, the host should fill the rest with H'FF.

Sending the 128-byte programming command with the address of H'FFFFFFFF will stop the programming operation. The boot program will interpret this as the end of the programming and wait for selection of programming or erasing.

Command	H'50	Address	SUM

- Command, H'50, (one byte): 128-byte programming
- Programming Address (four bytes): End code is H'FF, H'FF, H'FF, H'FF.
- SUM (one byte): Checksum

Response	H'06

- Response, H'06, (one byte): Response to 128-byte programming
  On completion of programming, the boot program will return ACK.

Error Response	H'D0	ERROR

- Error Response, H'D0, (one byte): Error response for 128-byte programming
- ERROR: (one byte): Error code
  H'11: Checksum error
  H'53: Programming error
  An error has occurred in programming and programming cannot be continued.

(d) Erasure Selection

The boot program will transfer the erasure program. User MAT data is erased by the transferred erasure program.

Command	H'48

- Command, H'48, (one byte): Erasure selection

Response	H'06

- Response, H'06, (one byte): Response for erasure selection
  After the erasure program has been transferred, the boot program will return ACK.

Error Response	H'C8	ERROR

- ERROR: (one byte): Error code
  H'54: Selection processing error (transfer error occurs and processing is not completed)

(e) Block Erasure

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

Command	H'58	Size	Block number	SUM

- Command, H'58, (one byte): Erasure
- Size (one byte): The number of bytes that represents the erase block number
  This is fixed to 1.
- Block number (one byte): Number of the block to be erased
- SUM (one byte): Checksum

Response	H'06

- Response, H'06, (one byte): Response to Erasure
  After erasure has been completed, the boot program will return ACK.

Error Response	H'D8	ERROR

- Error Response, H'D8, (one byte): Response to Erasure
- ERROR (one byte): Error code

  H'11: Sum check error
  H'29: Block number error
  Block number is incorrect.
  H'51: Erasure error
  An error has occurred during erasure.

On receiving block number H'FF, the boot program will stop erasure and wait for a selection command.

Command	H'58	Size	Block number	SUM

- Command, H'58, (one byte): Erasure
- Size, (one byte): The number of bytes that represents the block number
  This is fixed to 1.
- Block number (one byte): H'FF
  Stop code for erasure
- SUM (one byte): Checksum

Response	H'06

- Response, H'06, (one byte): Response to end of erasure (ACK)
  When erasure is to be performed after the block number H'FF has been sent, the procedure should be executed from the erasure selection command.

(f) 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 (fixed at 9)
- Area (1 byte)

  H'01: User MAT

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

Response

H'52	Read size						
Data	…						
SUM							

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

Error Response

H'D2	ERROR

- Error response: H'D2 (1 byte): Error response to memory read
- ERROR: (1 byte): Error code

  H'11: Sum check error

  H'2A: Address error

  The read address is not in the MAT.

  H'2B: Size error

  The read size exceeds the MAT.

(g) User Boot MAT Sum Check

The boot program will return the byte-by-byte total of the contents of the bytes of the user boot MAT.

Command	H'4A

- Command, H'4A, (one byte): Sum check for user boot MAT

Response	H'5A	Size	Checksum of MAT	SUM

- Response, H'5A, (one byte): Response to the sum check of the user boot MAT
- Size (one byte): The number of bytes that represents the checksum
  This is fixed to 4.
- Checksum of MAT (four bytes): Checksum of user boot MATs
  The total of the data is obtained in byte units.
- SUM (one byte): Sum check for data being transmitted

(h) User-Program Sum Check

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

Command	H'4B

- Command, H'4B, (one byte): Sum check for user program

Response	H'5B	Size	Checksum of user program	SUM

- Response, H'5B, (one byte): Response to the sum check of the user program
- Size (one byte): The number of bytes that represents the checksum
  This is fixed to 4.
- Checksum of user boot program (four bytes): Checksum of user MATs
  The total of the data is obtained in byte units.
- SUM (one byte): Sum check for data being transmitted

(i) User Boot MAT Blank Check

The boot program will check whether or not all user boot MATs are blank and return the result.

Command	H'4C

- Command, H'4C, (one byte): Blank check for user boot MATs

Response	H'06

- Response, H'06, (one byte): Response to the blank check for user boot MATs
  If the contents of all user boot MATs are blank (H'FF), the boot program will return ACK.

Error Response	H'CC	H'52

- Error Response, H'CC, (one byte): Error response to the blank check of user boot MATs.
- Error code, H'52, (one byte): Erasure has not been completed.

(j) User MAT Blank Check

The boot program will check whether or not all user MATs are blank and return the result.

Command	H'4D

- Command, H'4D, (one byte): Blank check for user MATs

Response	H'06

- Response, H'06, (one byte): Response to the blank check for user 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, (one byte): Error response to the blank check of user MATs.
- Error code, H'52, (one byte): Erasure has not been completed.

(k) Boot Program State Inquiry

The boot program will return indications of its present state and error condition. This inquiry can be made in the inquiry/selection state or the programming/erasing state.

Command	H'4F

- Command, H'4F, (one byte): Inquiry regarding boot program's state

Response	H'5F	Size	Status	ERROR	SUM

- Response, H'5F, (one byte): Response to boot program state inquiry
- Size (one byte): The number of bytes. This is fixed to 2.
- Status (one byte): State of the boot program
- ERROR (one byte): Error status

  ERROR = 0 indicates normal operation.
  ERROR = 1 indicates error has occurred.

- SUM (one byte): Sum check

Table 24.19 Status Codes

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	Program data receive wait
H'5F	Erase block specification wait (erasure is completed)

Table 24.20 Error Codes

Code	Description
H'00	No error
H'11	Sum check error
H'12	Program size error
H'21	Device code mismatch error
H'22	Clock mode mismatch error
H'24	Bit rate selection error
H'25	Input frequency error
H'26	Division 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

24.13 Usage Notes

1. The initial state of the product at its shipment is in the erased state. For the product whose revision of erasing is undefined, we recommend to execute automatic erasure for checking the initial state (erased state) and compensating.
2. For the PROM programmer suitable for programmer mode in this LSI and its program version, refer to the instruction manual of the socket adapter.
3. If the socket, socket adapter, or product index does not match the specifications, too much current flows and the product may be damaged.
4. Use a PROM programmer that supports the device with 256-Kbyte on-chip flash memory and 3.3 V programming voltage. Use only the specified socket adapter.
5. Do not power off the Vcc power supply (including the removal of the chip from the PROM programmer) during programming/erasing in which a high voltage is applied to the flash memory. Doing so may damage the flash memory permanently. If a reset is input, the reset must be released after the reset input period of at least 100µs.
6. The flash memory is not accessible until FKEY is cleared after programming/erasing starts. If the operating mode is changed and this LSI is restarted by a reset immediately after programming/erasing has finished, secure the reset input period (period of $\overline{RES}$ = 0) of at least 100µs. Transition to the reset state during programming/erasing is inhibited. If a reset is input accidentally, the reset must be released after the reset input period of at least 100µs.
7. At powering on the Vcc power supply, fix the $\overline{RES}$ pin to low and set the flash memory to hardware protection state. This power on timing must also be satisfied at a power-off and power-on caused by a power failure and other factors.
8. In on-board programming mode or programmer mode, programming of the 128-byte programming-unit block must be performed only once. Perform programming in the state where the programming-unit block is fully erased.
9. When the chip is to be reprogrammed with the programmer after execution of programming or erasure in on-board programming mode, it is recommended that automatic programming is performed after execution of automatic erasure.
10. To program the flash memory, the program data and program must be allocated to addresses which are higher than those of the external interrupt vector table and H'FF must be written to all the system reserved areas in the exception handling vector table.
11. If data other than H'FF (4 bytes) is written to the key code area (H'00003C to H'00003F) of the flash memory, reading cannot be performed in programmer mode. (In this case, data is read as H'00. Rewrite is possible after erasing the data.) For reading in programmer mode, make sure to write H'FF to the entire key code area.

12. If data other than H'FF is to be written to the key code area in programmer mode, a verification error will occur unless a software countermeasure is taken for the PROM programmer and version of program.
13. The programming program that includes the initialization routine and the erasing program that includes the initialization routine are each 3 kbytes or less. Accordingly, when the CPU clock frequency is 20 MHz, the download for each program takes approximately 200 μs at the maximum.
14. A programming/erasing program for the flash memory used in a conventional F-ZTAT H8, H8S microcomputer which does not support download of the on-chip program by setting the SCO bit in FCCS to 1 cannot run in this LSI. Be sure to download the on-chip program to execute programming/erasing of the flash memory in this F-ZTAT H8/H8S microcomputer.
15. Unlike a conventional F-ZTAT H8/H8S microcomputers, measures against a program crash are not taken by WDT while programming/erasing and downloading a programming/erasing program. When needed, measures should be taken by user. A periodic interrupt generated by the WDT can be used as the measures, as an example. In this case, the interrupt generation period should take into consideration time to program/erase the flash memory.
16. When downloading the programming/erasing program, do not clear the SCO bit in FCCS to 0 after immediately setting it to 1. Otherwise, download cannot be performed normally. Immediately after executing the instruction to set the SCO bit to 1, dummy read of the FCCS must be executed twice.
17. The contents of some registers are not saved in a programming/programming end/erasing program. When needed, save registers in the procedure program.

Section 25 Clock Pulse Generator

This LSI incorporates a clock pulse generator which generates the system clock (ϕ), internal clock, bus master clock, and subclock (ϕSUB). The clock pulse generator consists of an oscillator, duty correction circuit, system clock select circuit, subclock input circuit, and subclock waveform forming circuit. Figure 25.1 shows a block diagram of the clock pulse generator.

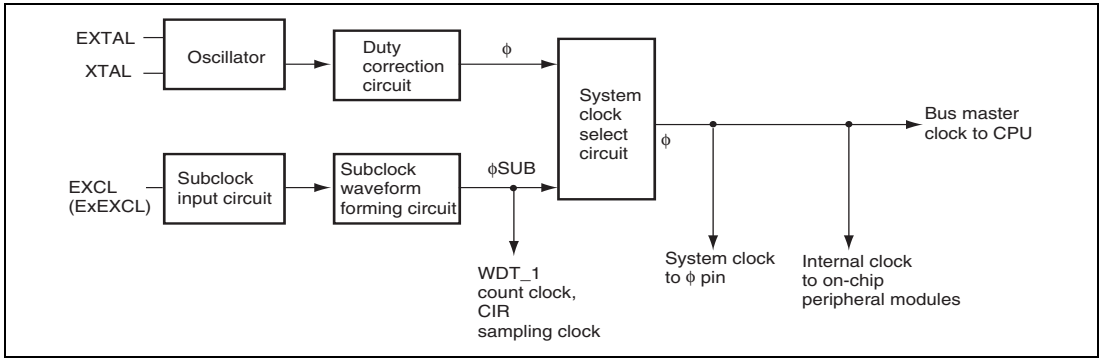


Figure 25.1 Block Diagram of Clock Pulse Generator

The subclock input is controlled by software according to the EXCLE bit and the EXCLS bit in the port control register (PTCNT0) settings in the low power control register (LPWRCR). For details on LPWRCR, see section 26.1.2, Low-Power Control Register (LPWRCR). For details on PTCNT0, see section 7.3.1, Port Control Register 0 (PTCNT0).

25.1 Oscillator

Clock pulses can be supplied either by connecting a crystal resonator or by providing external clock input.

25.1.1 Connecting Crystal Resonator

Figure 25.2 shows a typical method for connecting a crystal resonator. An appropriate damping resistance R_d, given in table 25.1 should be used. An AT-cut parallel-resonance crystal resonator should be used.

Figure 25.3 shows an equivalent circuit of a crystal resonator. A crystal resonator having the characteristics given in table 25.2 should be used.

The frequency of the crystal resonator should be the same as that of the system clock (ϕ).

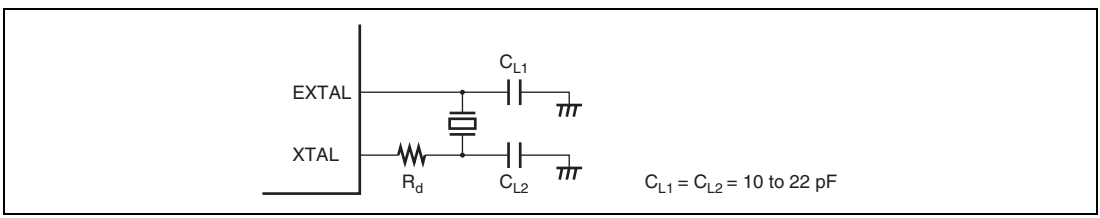


Figure 25.2 Typical Connection to Crystal Resonator

Table 25.1 Damping Resistor Values

Frequency (MHz)	8	10	12	16	20
R_d (Ω)	200	0	0	0	0

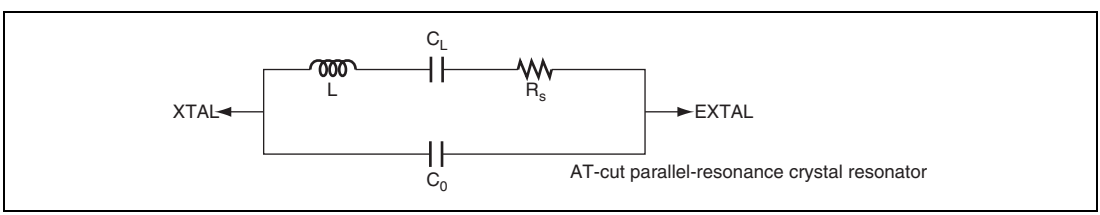


Figure 25.3 Equivalent Circuit of Crystal Resonator

Table 25.2 Crystal Resonator Parameters

Frequency (MHz)	8	10	12	16	20
R_S (max) (Ω)	80	70	60	50	40
C_0 (max) (pF)	7	7	7	7	7

25.1.2 External Clock Input Method

Figure 25.4 shows a typical method of inputting an external clock signal. To leave the XTAL pin open, incidental capacitance should be 10 pF or less. To input an inverted clock to the XTAL pin, the external clock should be set to high in standby mode or watch mode. External clock input conditions are shown in table 25.3. The frequency of the external clock should be the same as that of the system clock (ϕ).

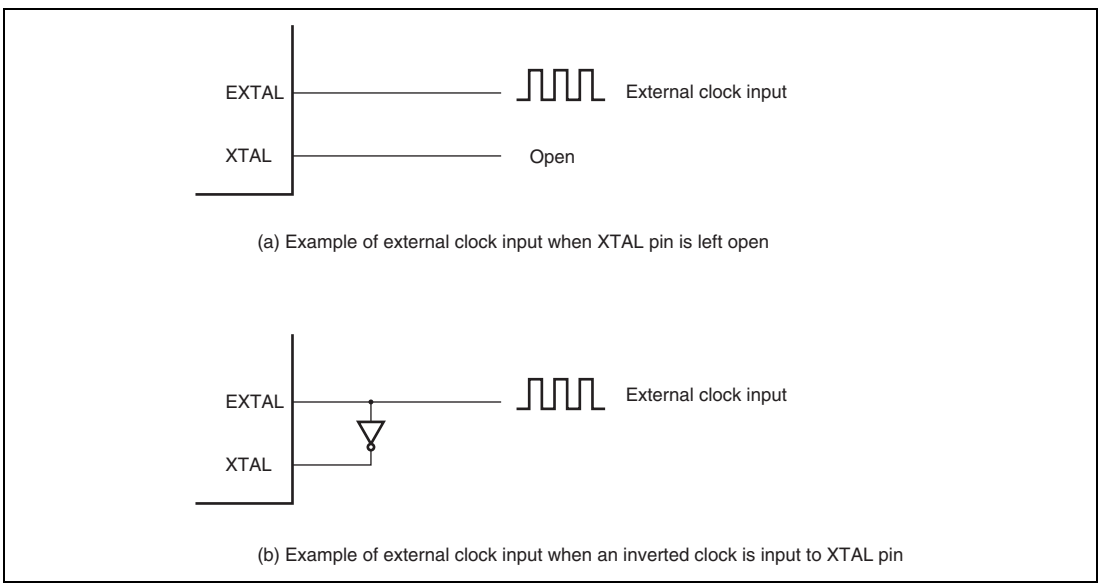


Figure 25.4 Example of External Clock Input

Table 25.3 External Clock Input Conditions

		VCC = 3.0 to 3.6 V			
Item	**Symbol**	**Min.**	**Max.**	**Unit**	**Test Conditions**
External clock input pulse width low level	t_{EXL}	20	—	ns	Figure 25.5
External clock input pulse width high level	t_{EXH}	20	—	ns	
External clock rising time	t_{EXr}	—	5	ns	
External clock falling time	t_{EXf}	—	5	ns	
Clock pulse width low level	t_{CL}	0.4	0.6	t_{cyc}	Figure 28.4
Clock pulse width high level	t_{CH}	0.4	0.6	t_{cyc}	

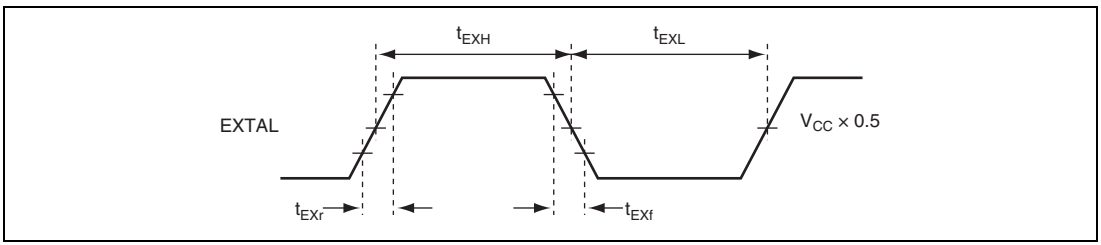


Figure 25.5 External Clock Input Timing

The oscillator and duty correction circuit can adjust the waveform of the external clock input that is input from the EXTAL pin.

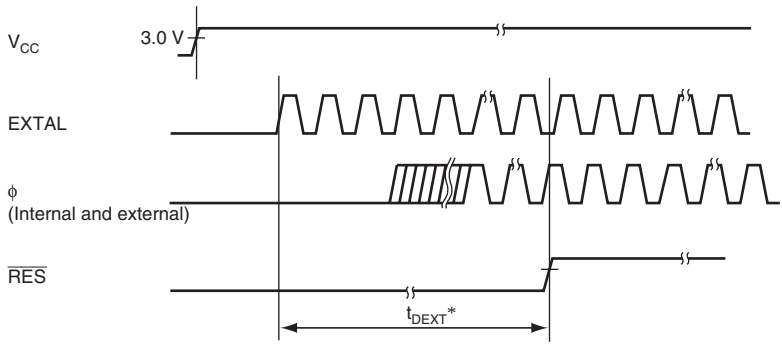
When a specified clock signal is input to the EXTAL pin, internal clock signal output is determined after the external clock output stabilization delay time (t_{DEXT}) has passed. As the clock signal output is not determined during the t_{DEXT} cycle, a reset signal should be set to low to maintain the reset state. Table 25.4 shows the external clock output stabilization delay time. Figure 25.6 shows the timing of the external clock output stabilization delay time.

Table 25.4 External Clock Output Stabilization Delay Time

Condition: VCC = 3.0 V to 3.6 V, AVCC = 3.0 V to 3.6 V, VSS = AVSS = 0 V

Item	Symbol	Min.	Max.	Unit	Remarks
External clock output stabilization delay time	t_{DEXT}*	500	—	μs	Figure 25.6

Note: * t_{DEXT} includes a $\overline{RES}$ pulse width (t_{RESW}).



Note: * The external clock output stabilization delay time (t_{DEXT}) includes a $\overline{RES}$ pulse width (t_{RESW}).

Figure 25.6 Timing of External Clock Output Stabilization Delay Time

25.2 Duty Correction Circuit

The duty correction circuit generates the system clock (ϕ) by correcting the duty of the clock output from the oscillator.

25.3 Subclock Input Circuit

The subclock input circuit controls subclock input from the EXCL or ExEXCL pin. To use the subclock, a 32.768-kHz external clock should be input from the EXCL or ExEXCL pin.

Figure 25.7 shows the relationship of subclock input from the EXCL pin and the ExEXCL pin. When using a pin to input the subclock, specify input for the pin by clearing the DDR bit of the pin to 0. The EXCL pin is specified as an input pin by clearing the EXCLS bit in PTCNT0 to 0. The ExEXCL pin is specified as an input pin by setting the EXCLS bit in PTCNT0 to 1. The subclock input is enabled by setting the EXCLE bit in LPWRCR to 1.

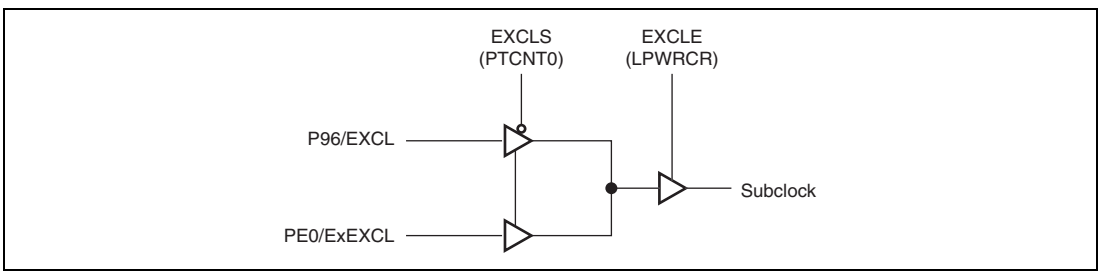


Figure 25.7 Subclock Input from EXCL Pin and ExEXCL Pin

Subclock input conditions are shown in table 25.5. When the subclock is not used, subclock input should not be enabled.

Table 25.5 Subclock Input Conditions

		VCC = 3.0 to 3.6 V				
Item	**Symbol**	**Min.**	**Typ.**	**Max.**	**Unit**	**Test Conditions**
Subclock input pulse width low level	t_{EXCLL}	—	15.26	—	μs	Figure 25.8
Subclock input pulse width high level	t_{EXCLH}	—	15.26	—	μs	
Subclock input rising time	t_{EXCLr}	—	—	10	ns	
Subclock input falling time	t_{EXCLf}	—	—	10	ns	

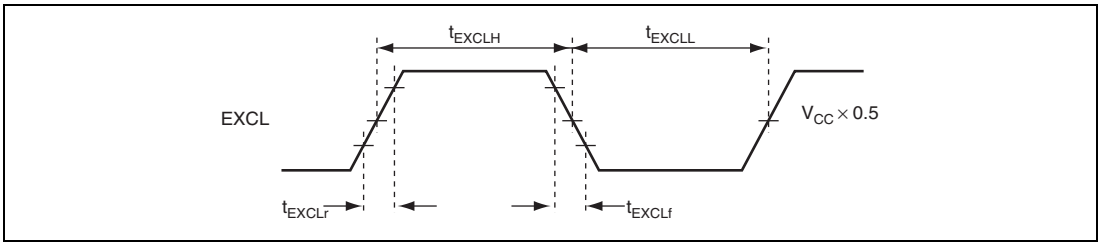


Figure 25.8 Subclock Input Timing

25.4 Subclock Waveform Forming Circuit

To remove noise from the subclock input at the EXCL (ExEXCL) pin, the subclock waveform forming circuit samples the subclock using a divided ϕ clock. The sampling frequency is set by the NESEL bit in LPWRCR.

The subclock is not sampled in watch mode.

25.5 Clock Select Circuit

The clock select circuit selects the system clock that is used in this LSI.

A clock generated by the oscillator to which the XTAL and EXTAL pins are connected is selected as a system clock (ϕ) when returning from high-speed mode, sleep mode, the reset state, or standby mode.

In watch mode, a subclock input from the EXCL (ExEXCL) pin is selected as a system clock when the EXCLE bit in LPWRCR is 1. At this time, on-chip peripheral modules such as WDT_1 and interrupt controller operate on the ϕSUB clock. The count clock and sampling clock for each timer are divided ϕSUB clocks.

25.6 Usage Notes

25.6.1 Notes on Resonator

Since all kinds of characteristics of the resonator are closely related to the board design by the user, use the example of resonator connection in this document for only reference; be sure to use an resonator that has been sufficiently evaluated by the user. Consult with the resonator manufacturer about the resonator circuit ratings that vary depending on the stray capacitances of the resonator and installation circuit. Make sure the voltage applied to the oscillation pins do not exceed the maximum rating.

25.6.2 Notes on Board Design

When using a crystal resonator, the crystal resonator and its load capacitors should be placed as close as possible to the XTAL and EXTAL pins. Other signal lines should be routed away from the oscillator to prevent inductive interference with correct oscillation as shown in figure 25.9.

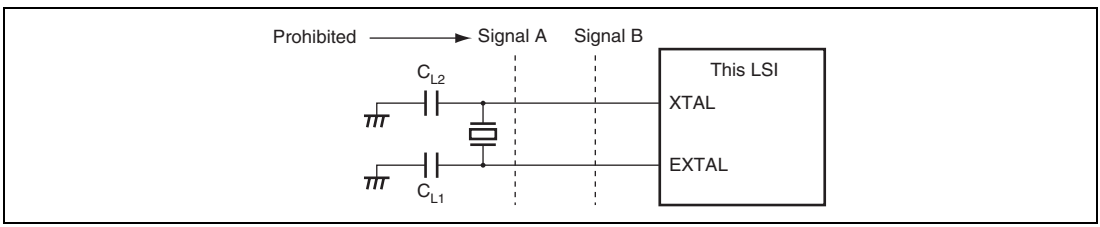


Figure 25.9 Note on Board Design of Oscillator Section

Section 26 Power-Down Modes

For operating modes after the reset state is cancelled, this LSI has four power-down operating modes in which power consumption is significantly reduced. In addition, there is also module stop mode in which reduced power consumption can be achieved by individually stopping on-chip peripheral modules.

- Medium-speed mode
  System clock frequency for the CPU operation can be selected as $\phi/2$, $\phi/4$, $\phi/8$, $\phi/16$ or $\phi/32$.
- Sleep mode
  The CPU stops but on-chip peripheral modules continue operating.
- Watch mode
  The CPU stops, but on-chip peripheral module WDT_1 and CIR continue operating.
- Software standby mode
  The clock pulse generator stops, and the CPU and on-chip peripheral modules stop operating.
- Module stop mode
  Independently of above operating modes, on-chip peripheral modules that are not used can be stopped individually.

26.1 Register Descriptions

Power-down modes are controlled by the following registers. To access SBYCR, LPWRCR, SYSCR2, MSTPCRH, and MSTPCRL the FLSHE bit in the serial timer control register (STCR) must be cleared to 0. For details on STCR, see section 3.2.3, Serial Timer Control Register (STCR). For details on the PSS bit in TSCR_1 (WDT_1), see TCSR_1 in section 13.3.5, Timer Control/Status Register (TCSR).

Table 26.1 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Data Bus Width
Standby control register	SBYCR	R/W	H'00	H'FF84	8
Low power control register	LPWRCR	R/W	H'00	H'FF85	8
Module stop control register H	MSTPCRH	R/W	H'3F	H'FF86	8
Module stop control register L	MSTPCRL	R/W	H'FF	H'FF87	8
Module stop control register A	MSTPCRA	R/W	H'FC	H'FE7E	8
Module stop control register B	MSTPCRB	R/W	H'FF	H'FE7F	8

26.1.1 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-speed mode or medium-speed mode: 0: Shifts to sleep mode 1: Shifts to software standby mode or watch mode Note that the SSBY bit is not changed even if a mode transition is made by an interrupt.
6	STS2	0	R/W	Standby Timer Select 2 to 0 On canceling software standby mode or watch mode, these bits select the wait time for clock stabilization from clock oscillation start. Select a wait time of 8 ms (oscillation stabilization time) or more, depending on the operating frequency. Table 26.2 shows the relationship between the STS2 to STS0 values and wait time. With an external clock, an arbitrary wait time can be selected. For normal cases, the minimum value is recommended.
5	STS1	0	R/W	
4	STS0	0	R/W	
3	—	0	R/W	Reserved The initial value should not be changed.

Bit	Bit Name	Initial Value	R/W	Description
2	SCK2	0	R/W	System Clock Select 2 to 0
1	SCK1	0	R/W	These bits select a clock for the bus master in high-speed mode or medium-speed mode.
0	SCK0	0	R/W	When making a transition to watch mode, these bits must be cleared to B'000.
				000: High-speed mode
				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: Setting prohibited

[Legend]
X: Don't care

Table 26.2 Operating Frequency and Wait Time

STS2	STS1	STS0	Wait Time	20 MHz	10 MHz	8 MHz	Unit
0	0	0	8192 states	0.4	0.8	1.0	ms
0	0	1	16384 states	0.8	1.6	2.0	
0	1	0	32768 states	1.6	3.3	4.1	
0	1	1	65536 states	3.3	6.6	8.2	
1	0	0	131072 states	6.6	13.1	16.4	
1	0	1	262144 states	13.1	26.2	32.8	
1	1	0/1	Reserved*	—	—	—	—

Recommended specification

Note: * Setting prohibited.

26.1.2 Low-Power Control Register (LPWRCR)

LPWRCR controls power-down modes.

Bit	Bit Name	Initial Value	R/W	Description
7	DTON	0	R/W	Direct Transfer On Flag The initial value should not be changed.
6	LSON	0	R/W	Low-Speed On Flag The initial value should not be changed.
5	NESEL	0	R/W	Noise Elimination Sampling Frequency Select Selects the frequency by which the subclock (ϕSUB) input from the EXCL or ExEXCL pin is sampled using the clock (ϕ) generated by the system clock pulse generator. Clear this bit to 0 when ϕ is 5 MHz or more. The initial value should not be changed. 0: Sampling using ϕ/32 clock 1: Sampling using ϕ/4 clock (not allowed)
4	EXCLE	0	R/W	Subclock Input Enable Enables or disables subclock input from the EXCL or ExEXCL pin. 0: Disables subclock input from the EXCL or ExEXCL pin 1: Enables subclock input from the EXCL or ExEXCL pin
3 to 0	—	All 0	R/W	Reserved The initial value should not be changed.

26.1.3 Module Stop Control Registers H, L, A, and B (MSTPCRH, MSTPCRL, MSTPCRA, MSTPCRB)

MSTPCR specifies on-chip peripheral modules to shift to module stop mode in module units. Each module can enter module stop mode by setting the corresponding bit to 1.

- MSTPCRH

Bit	Bit Name	Initial Value	R/W	Corresponding Module
7	MSTP15	0	R/W	Reserved The initial value should not be changed.
6	MSTP14	0	R/W	Reserved The initial value should not be changed.
5	MSTP13	1	R/W	Reserved The initial value should not be changed.
4	MSTP12	1	R/W	8-bit timers (TMR_0 and 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 and TMR_Y)

- MSTPCRL

Bit	Bit Name	Initial Value	R/W	Corresponding Module
7	MSTP7	1	R/W	Reserved The initial value should not be changed.
6	MSTP6	1	R/W	Serial communication interface 1 (SCI_1)
5	MSTP5	1	R/W	Serial communication interface 2 (SCI_2)
4	MSTP4	1	R/W	I^2C bus interface channel 0 (IIC_0)
3	MSTP3	1	R/W	I^2C bus interface channel 1 (IIC_1)
2	MSTP2	1	R/W	Keyboard buffer control unit_0 (PS2_0) Keyboard buffer control unit_1 (PS2_1) Keyboard buffer control unit_2 (PS2_2)
1	MSTP1	1	R/W	16-bit timer pulse unit (TPU)
0	MSTP0	1	R/W	LPC interface (LPC)

- MSTPCRA

Bit	Bit Name	Initial Value	R/W	Corresponding Module
7	MSTPA7	1	R/W	Reserved The initial value should not be changed.
6	MSTPA6	1	R/W	Duty period measurement timer_0 (TDP_0)
5	MSTPA5	1	R/W	Duty period measurement timer_1 (TDP_1)
4	MSTPA4	1	R/W	Duty period measurement timer_2 (TDP_2)
3	MSTPA3	1	R/W	CIR interface (CIR)
2	MSTPA2	1	R/W	FSI interface (FSI)*
1	MSTPA1	0	R/W	14-bit PWM timer (PWMX)
0	MSTPA0	0	R/W	Reserved The initial value should not be changed.

Note: * Before accessing registers of the FSI interface, clear bit 0 in MSTPCRL (MSTP0) and bit 2 in MSTPCRA (MSTPA2) to 0.

- MSTPCRB

Bit	Bit Name	Initial Value	R/W	Corresponding Module
7	MSTPB7	1	R/W	Reserved The initial value should not be changed.
6	MSTPB6	1	R/W	Reserved The initial value should not be changed.
5	MSTPB5	1	R/W	Keyboard buffer control unit_3 (PS2_3)
4	MSTPB4	1	R/W	I^2C bus interface_2 (IIC_2)
3	MSTPB3	1	R/W	Serial communication interface with FIFO (SCIF)
2	MSTPB2	1	R/W	Cycle measurement timer_2 (TCM_2) Cycle measurement timer_3 (TCM_3)
1	MSTPB1	1	R/W	Cycle measurement timer_0 (TCM_0) Cycle measurement timer_1 (TCM_1)
0	MSTPB0	1	R/W	8-bit PWMU timer_A (TWMU_A) 8-bit PWMU timer_B (TWMU_B)

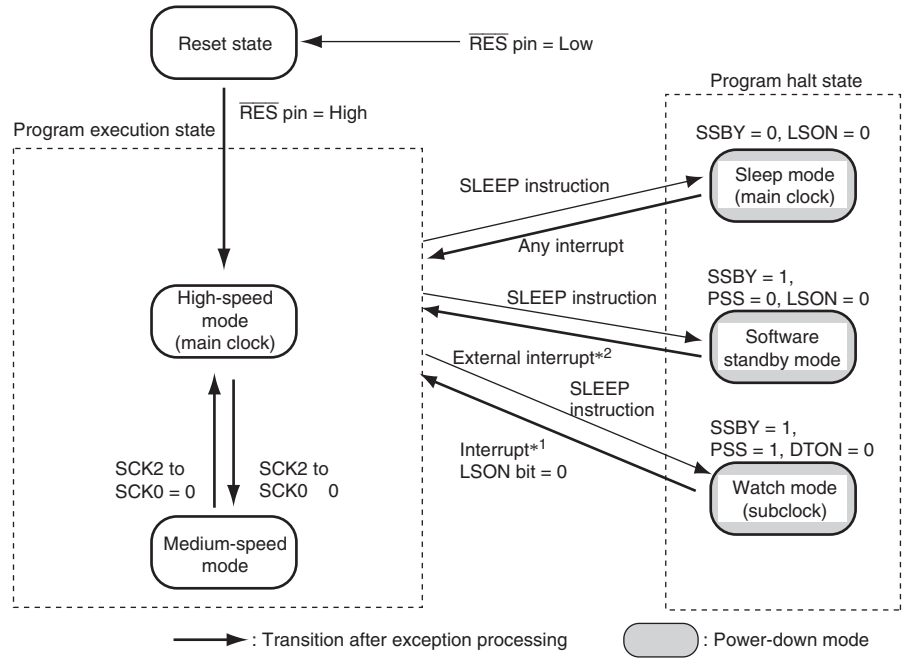
The PWMX sets operation or stop by a combination of bits as follows:

MSTPCRH: MSTP11	MSTPCRA: MSTPA1	Function
0	0	14-bit PWM timer (PWMX) operates.
0	1	14-bit PWM timer (PWMX) stops.
1	0	14-bit PWM timer (PWMX) stops.
1	1	14-bit PWM timer (PWMX) stops.

Note: The MSTP11 bit in MSTPCRH is a module stop bit of the PWMX.

26.2 Mode Transitions and LSI States

Figure 26.1 shows the possible mode transition diagram. The mode transition from program execution state to program halt state is performed by the SLEEP instruction. The mode transition from program halt state to program execution state is performed by an interrupt. The $\overline{\text{RES}}$ input causes a mode transition from any state to the reset state. Table 26.3 shows the LSI internal states in each operating mode.



Notes: When a transition is made between modes by means of an interrupt, the transition cannot be made on interrupt source generation alone. Ensure that interrupt handling is performed after accepting the interrupt request.

1. NMI, IRQ0 to IRQ15, KIN0 to KIN15, WUE8 to WUE15, and WDT_1, PS2 and CIR interrupts
2. NMI, IRQ0 to IRQ15, KIN0 to KIN15, WUE8 to WUE15, and PS2 interrupts

Figure 26.1 Mode Transition Diagram

Table 26.3 LSI Internal States in Each Operating Mode

Function		High Speed	Medium Speed	Sleep	Module Stop	Watch	Software Standby
System clock pulse generator		Functioning	Functioning	Functioning	Functioning	Stopped	Stopped
Subclock input		Functioning	Functioning	Functioning	Functioning	Functioning	Stopped
CPU	Instruction execution	Functioning	Medium-speed operation	Stopped	Functioning	Stopped	Stopped
	Registers			Retained		Retained	Retained
External interrupts	NMI	Functioning	Functioning	Functioning	Functioning	Functioning	Functioning
	IRQ0 to IRQ15						
	KIN0 to KIN15						
	WUE8 to WUE15						
On-chip peripheral modules	WDT_1	Functioning	Functioning	Functioning	Functioning	Subclock operation	Stopped (retained)
	CIR						
	WDT_0					Stopped (retained)	
	TMR_0, TMR_1				Functioning/ stopped (retained)		
	TPU TCM_0 to 3 TDP_0 to 2						
	TMR_X, TMR_Y						
	SCIF						
	IIC_0 to 2						
	LPC						
	FSI						
	PS2_0 to 3		Medium-speed operation/functioning				
	PWMU		Functioning		Functioning/ stopped (reset)	Stopped (reset)	Stopped (reset)
	PWM						
	PWMX						
	SCI_1, SCI_2						
	A/D converter						
	RAM	Functioning	Functioning	Functioning	Functioning	Retained	Retained
	I/O	Functioning	Functioning	Functioning	Functioning	Retained	Retained

Note: Stopped (retained) means that the internal register values are retained and the internal state is operation suspended. Stopped (reset) means that the internal register values and the internal state are initialized. In module stop mode, only modules for which a stop setting has been made are stopped (reset or retained).

26.3 Medium-Speed Mode

The operating mode changes to medium-speed mode as soon as the current bus cycle ends by the settings of the SCK2 to SCK0 bits in SBYCR. The operating clock can be selected from ϕ/2, ϕ/4, ϕ/8, ϕ/16, or ϕ/32. On-chip peripheral functions other than the bus masters and the PS2 operate on the system clock (ϕ).

In medium-speed mode, a bus access is executed in the specified number of states with respect to the bus master operating clock. For example, if ϕ/4 is selected as the operating clock, on-chip memory is accessed in four states, and internal I/O registers in eight states.

A transition is made from medium-speed mode to high-speed mode at the end of the current bus cycle by clearing all of bits SCK2 to SCK0 to 0.

If the SLEEP instruction is executed when the SSBY bit in SBYCR is 0 and the LSON bit in LPWRCR is 0, a transition is made to sleep mode. When sleep mode is canceled by an interrupt, medium-speed mode is restored. When the SLEEP instruction is executed with the SSBY bit set to 1, the LSON bit in LPWRCR set to 0, and the PSS bit in TCSR (WDT_1) set to 0, operation shifts to software standby mode. When software standby mode is canceled by an external interrupt, medium-speed mode is restored.

When the $\overline{\text{RES}}$ pin is driven low and medium-speed mode is cancelled, operation shifts to the reset state. The same applies to a reset caused by an overflow of the watchdog timer.

Figure 26.2 shows the timing of medium-speed mode.

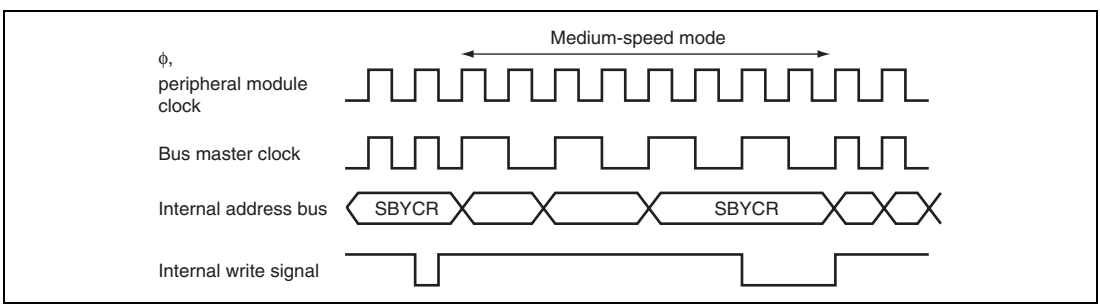


Figure 26.2 Timing of Medium-Speed Mode

26.4 Sleep Mode

The CPU makes a transition to sleep mode if the SLEEP instruction is executed when the SSBY bit in SBYCR is cleared to 0 and the LSON bit in LPWRCR is cleared to 0. In sleep mode, CPU operation stops but the on-chip peripheral modules do not. The contents of the CPU's internal registers are retained.

Sleep mode is cleared by any interrupt or the $\overline{\text{RES}}$ pin input.

When an interrupt occurs, sleep mode is cleared and interrupt exception handling starts. Sleep mode is not cleared if the interrupt is disabled, or interrupts other than NMI have been masked by the CPU.

When the $\overline{\text{RES}}$ pin is driven low and sleep mode is cleared, a transition is made to the reset state. After the specified reset input time has elapsed, driving the $\overline{\text{RES}}$ pin high causes the CPU to start reset exception handling.

26.5 Software Standby Mode

The CPU makes a transition to software standby mode when the SLEEP instruction is executed with the SSBY bit in SBYCR set to 1, the LSON bit in LPWRCR cleared to 0, and the PSS bit in TCSR (WDT_1) cleared to 0. In software standby mode, the CPU, on-chip peripheral modules, and clock pulse generator all stop. However, the contents of the CPU registers, on-chip RAM data, I/O ports, and the states of on-chip peripheral modules other than the SCI, PWMU, PWMX, and A/D converter are retained as long as the prescribed voltage is supplied.

Software standby mode is cleared by an external interrupt (NMI, IRQ0 to IRQ15, KIN0 to KIN15, or WUE8 to WUE15), PS2 interrupt, or $\overline{\text{RES}}$ pin input.

When an external interrupt request signal is input, system clock oscillation starts, and after the elapse of the time set in bits STS2 to STS0 in SBYCR, software standby mode is cleared, and interrupt exception handling is started. When clearing software standby mode with an IRQ0 to IRQ15 interrupt, set the corresponding enable bit to 1. When clearing software standby mode with a KIN0 to KIN15 or WUE8 to WUE15 interrupt, enable the input. In these cases, ensure that no interrupt with a higher priority than interrupts IRQ0 to IRQ15 is generated. In the case of an IRQ0 to IRQ15 interrupt, software standby mode is not cleared if the corresponding enable bit is cleared to 0 or if the interrupt has been masked by the CPU. In the case of a KIN0 to KIN15 or WUE8 to WUE15 interrupt, software standby mode is not cleared if the input is disabled or if the interrupt has been masked by the CPU.

When the $\overline{\text{RES}}$ pin is driven low, the clock pulse generator starts oscillation. Simultaneously with the start of system clock oscillation, the system clock is supplied to the entire LSI. Note that the $\overline{\text{RES}}$ pin must be held low until clock oscillation is stabilized. If the $\overline{\text{RES}}$ pin is driven high after the clock oscillation stabilization time has elapsed, the CPU starts reset exception handling.

Figure 26.2 shows an example in which a transition is made to software standby mode at the falling edge of the NMI pin, and software standby mode is cleared at the rising edge of the NMI pin.

In this example, an NMI interrupt is accepted with the NMIEG bit in SYSCR cleared to 0 (falling edge specification), then the NMIEG bit is set to 1 (rising edge specification), the SSBY bit is set 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.

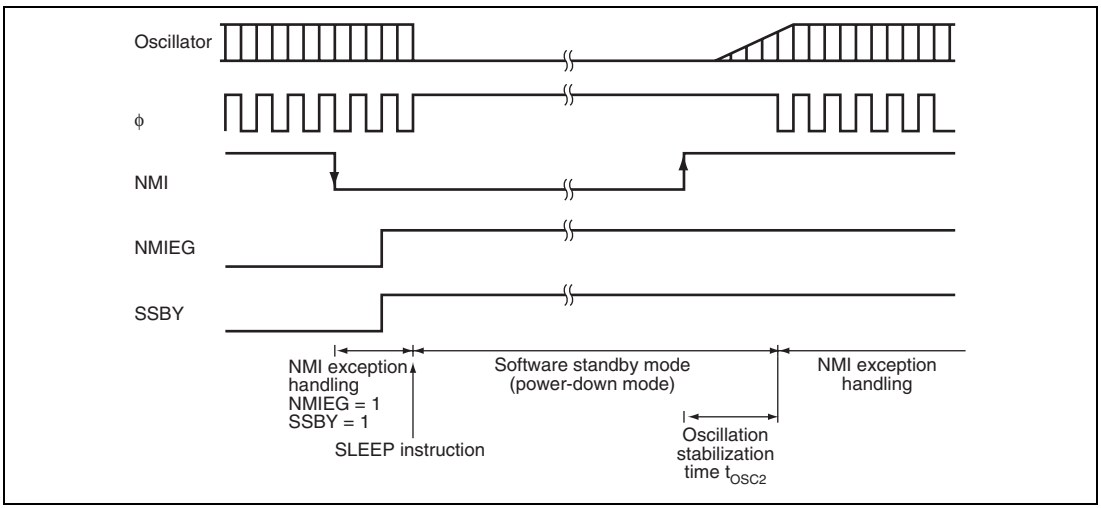


Figure 26.3 Software Standby Mode Application Example

26.6 Watch Mode

The CPU makes a transition to watch mode when the SLEEP instruction is executed in high-speed mode with the SSBY bit in SBYCR set to 1, the DTON bit in LPWRCR cleared to 0, and the PSS bit in TCSR (WDT_1) set to 1.

In watch mode, the CPU is stopped and on-chip peripheral modules other than CIR and WDT_1 are also stopped. The contents of the CPU's internal registers, several on-chip peripheral module registers, and on-chip RAM data are retained and the I/O ports retain their values before transition as long as the prescribed voltage is supplied.

Watch mode is cleared by an interrupt (WOVI1, NMI, IRQ0 to IRQ15, KIN0 to KIN15, or WUE8 to WUE15), PS2 interrupt, CIR interrupt, or $\overline{\text{RES}}$ pin input.

When an interrupt occurs, watch mode is cleared and a transition is made to high-speed mode or medium-speed mode. When a transition is made to high-speed mode, a stable clock is supplied to the entire LSI and interrupt exception handling starts after the time set in the STS2 to STS0 bits in SBYCR has elapsed. In the case of an IRQ0 to IRQ15 interrupt, watch mode is not cleared if the corresponding enable bit has been cleared to 0 or the interrupt has been masked by the CPU. In the case of a KIN0 to KIN15 or WUE8 to WUE15 interrupt, watch mode is not cleared if the input is disabled or the interrupt has been masked by the CPU. In the case of an interrupt from an on-chip peripheral module, watch mode is not cleared if the interrupt enable register has been set to disable the reception of that interrupt or the interrupt has been masked by the CPU.

When the $\overline{\text{RES}}$ pin is driven low, the clock pulse generator starts oscillation. Simultaneously with the start of system clock oscillation, the system clock is supplied to the entire LSI. Note that the $\overline{\text{RES}}$ pin must be held low until clock oscillation is stabilized. If the $\overline{\text{RES}}$ pin is driven high after the clock oscillation stabilization time has elapsed, the CPU starts reset exception handling.

26.7 Module Stop Mode

Module stop mode can be individually set for each on-chip peripheral module.

When the corresponding MSTP bit in MSTPCR is set to 1, module operation stops at the end of the bus cycle and a transition is made to module stop mode. In turn, when the corresponding MSTP bit is cleared to 0, module stop mode is cleared and module operation resumes at the end of the bus cycle. In module stop mode, the internal states of on-chip peripheral modules other than the SCI, PWM, PWMX, and A/D converter are retained.

After the reset state is cancelled, all on-chip peripheral modules are in module stop mode.

While an on-chip peripheral module is in module stop mode, its registers cannot be read from or written to.

26.8 Usage Notes

26.8.1 I/O Port Status

The status of the I/O ports is retained in software standby mode. Therefore, while a high level is output or the pull-up MOS is on, the current consumption is not reduced by the amount of current to support the high level output.

26.8.2 Current Consumption when Waiting for Oscillation Stabilization

The current consumption increases during oscillation stabilization.

Section 27 List of Registers

The list of registers gives information on the on-chip register addresses, how the register bits are configured, the register states in each operating mode, the register selection condition, and the register address of each module. The information is given as shown below.

1. Register addresses (address order)
- Registers are listed from the lower allocation addresses.
- For the addresses of 16 bits, the MSB is described.
- Registers are classified by functional modules.
- The access size is indicated.
- H8S/2140B Group compatible register addresses or extended register addresses are selected depending on the RELOCATE bit in system control register 3 (SYSCR3).

  When the extended register addresses are selected, the some register addresses of ICC_1, TMR_Y, PWMX_0, and PORT are changed. Therefore, the selection with other module registers that share the same addresses with these registers is not necessary.

2. Register bits
- Bit configurations of the registers are described in the same order as the register addresses in section 27.1, Register Addresses (Address Order).
- 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 as a counter or for holding data.
- Each line covers eight bits, and 16-bit register is shown as 2 lines, respectively.

3. Register states in each operating mode
- Register states are described in the same order as the register addresses in section 27.1, Register Addresses (Address Order).
- The register states described here are for the basic operating modes. If there is a specific reset for an on-chip peripheral module, see the section on that on-chip peripheral module.

4. Register selection conditions
- Register selection conditions are described in the same order as the register addresses in section 27.1, Register Addresses (Address Order).
- For register selection conditions, see section 3.2.2, System Control Register (SYSCR), section 3.2.3, Serial Timer Control Register (STCR), section 26.1.3, Module Stop Control Registers H, L, A, and B (MSTPCRH, MSTPCRL, MSTPCRA, MSTPCRB), or register descriptions for each module.

5. Register addresses (classification by type of module)
- The register addresses are described by modules
- The register addresses are described in channel order when the module has multiple channels.

27.1 Register Addresses (Address Order)

The data bus width indicates the numbers of bits.

The number of access states indicates the number of states based on the specified reference clock.

Register Name	Abbreviation	Number of bits	Address	Module	Data Width	Access States
Port 1 data direction register	P1DDR	8	H'F900 (PORTS = 1)	PORT	8	2
Port 2 data direction register	P2DDR	8	H'F901 (PORTS = 1)	PORT	8	2
Port 1 data register	P1DR	8	H'F902 (PORTS = 1)	PORT	8	2
Port 2 data register	P2DR	8	H'F903 (PORTS = 1)	PORT	8	2
Port 1 input data register	P1PIN	8	H'F904 (Read) (PORTS = 1)	PORT	8	2
Port 2 input data register	P2PIN	8	H'F905 (Read) (PORTS = 1)	PORT	8	2
Port 1 pull-up MOS control register	P1PCR	8	H'F906 (PORTS = 1)	PORT	8	2
Port 2 pull-up MOS control register	P2PCR	8	H'F907 (PORTS = 1)	PORT	8	2
Port 3 data direction register	P3DDR	8	H'F910 (PORTS = 1)	PORT	8	2
Port 4 data direction register	P4DDR	8	H'F911 (PORTS = 1)	PORT	8	2
Port 3 data register	P3DR	8	H'F912 (PORTS = 1)	PORT	8	2
Port 4 data register	P4DR	8	H'F913 (PORTS = 1)	PORT	8	2
Port 3 input data register	P3PIN	8	H'F914 (Read) (PORTS = 1)	PORT	8	2
Port 4 input data register	P4PIN	8	H'F915 (Read) (PORTS = 1)	PORT	8	2
Port 3 pull-up MOS control register	P3PCR	8	H'F916 (PORTS = 1)	PORT	8	2

Register Name	Abbreviation	Number of bits	Address	Module	Data Width	Access States
Port 5 data direction register	P5DDR	8	H'F920 (PORTS = 1)	PORT	8	2
Port 6 data direction register	P6DDR	8	H'F921 (PORTS = 1)	PORT	8	2
Port 5 data register	P5DR	8	H'F922 (PORTS = 1)	PORT	8	2
Port 6 data register	P6DR	8	H'F923 (PORTS = 1)	PORT	8	2
Port 5 input data register	P5PIN	8	H'F924 (Read) (PORTS = 1)	PORT	8	2
Port 6 input data register	P6PIN	8	H'F925 (Read) (PORTS = 1)	PORT	8	2
Port 6 noise canceler enable register	P6NCE	8	H'F92B (PORTS = 1)	PORT	8	2
Port 6 noise canceler decision control register	P6NCMC	8	H'F92D (PORTS = 1)	PORT	8	2
Port 6 noise cancel cycle setting register	P6NCCS	8	H'F92F (PORTS = 1)	PORT	8	2
Port 8 data direction register	P8DDR	8	H'F931 (PORTS = 1)	PORT	8	2
Port 8 data register	P8DR	8	H'F933 (PORTS = 1)	PORT	8	2
Port 7 input data register	P7PIN	8	H'F934 (Read) (PORTS = 1)	PORT	8	2
Port 8 input data register	P8PIN	8	H'F935 (Read) (PORTS = 1)	PORT	8	2
Port 9 data direction register	P9DDR	8	H'F940 (PORTS = 1)	PORT	8	2
Port 9 data register	P9DR	8	H'F942 (PORTS = 1)	PORT	8	2
Port 9 input data register	P9PIN	8	H'F944 (Read) (PORTS = 1)	PORT	8	2
Port 9 pull-up MOS control register	P9PCR	8	H'F946 (PORTS = 1)	PORT	8	2
Port A data direction register	PADDR	8	H'F950 (PORTS = 1)	PORT	8	2

Register Name	Abbreviation	Number of bits	Address	Module	Data Width	Access States
Port B data direction register	PBDDR	8	H'F951 (PORTS = 1)	PORT	8	2
Port A output data register	PAODR	8	H'F952 (PORTS = 1)	PORT	8	2
Port B output data register	PBODR	8	H'F953 (PORTS = 1)	PORT	8	2
Port A input data register	PAPIN	8	H'F954 (Read) (PORTS = 1)	PORT	8	2
Port B input data register	PBPIN	8	H'F955 (Read) (PORTS = 1)	PORT	8	2
Port B pull-up MOS control register	PBPCR	8	H'F957 (PORTS = 1)	PORT	8	2
Port C data direction register	PCDDR	8	H'F960 (PORTS = 1)	PORT	8	2
Port D data direction register	PDDDR	8	H'F961 (PORTS = 1)	PORT	8	2
Port C output data register	PCODR	8	H'F962 (PORTS = 1)	PORT	8	2
Port D output data register	PDODR	8	H'F963 (PORTS = 1)	PORT	8	2
Port C input data register	PCPIN	8	H'F964 (Read) (PORTS = 1)	PORT	8	2
Port D input data register	PDPIN	8	H'F965 (Read) (PORTS = 1)	PORT	8	2
Port C pull-up MOS control register	PCPCR	8	H'F966 (PORTS = 1)	PORT	8	2
Port D pull-up MOS control register	PDPCR	8	H'F967 (PORTS = 1)	PORT	8	2
Port C Nch-OD control register	PCNOCR	8	H'F968 (PORTS = 1)	PORT	8	2
Port D Nch-OD control register	PDNOCR	8	H'F969 (PORTS = 1)	PORT	8	2
Port C noise canceler enable register	PCNCE	8	H'F96A (PORTS = 1)	PORT	8	2
Port C noise canceler decision control register	PCNCMC	8	H'F96C (PORTS = 1)	PORT	8	2
Port C noise cancel cycle setting register	PCNCCS	8	H'F96E (PORTS = 1)	PORT	8	2

Register Name	Abbreviation	Number of bits	Address	Module	Data Width	Access States
Port F data direction register	PFDDR	8	H'F971 (PORTS = 1)	PORT	8	2
Port F output data register	PFODR	8	H'F973 (PORTS = 1)	PORT	8	2
Port E input data register	PEPIN	8	H'F974 (Read) (PORTS = 1)	PORT	8	2
Port F input data register	PFPIN	8	H'F975 (Read) (PORTS = 1)	PORT	8	2
Port F pull-up MOS control register	PFPCR	8	H'F977 (PORTS = 1)	PORT	8	2
Port F Nch-OD control register	PFNOCR	8	H'F979 (PORTS = 1)	PORT	8	2
Port G data direction register	PGDDR	8	H'F980 (PORTS = 1)	PORT	8	2
Port H data direction register	PHDDR	8	H'F981 (PORTS = 1)	PORT	8	2
Port G output data register	PGODR	8	H'F982 (PORTS = 1)	PORT	8	2
Port H output data register	PHODR	8	H'F983 (PORTS = 1)	PORT	8	2
Port G input data register	PGPIN	8	H'F984 (Read) (PORTS = 1)	PORT	8	2
Port H input data register	PHPIN	8	H'F985 (Read) (PORTS = 1)	PORT	8	2
Port H pull-up MOS control register	PHPCR	8	H'F987 (PORTS = 1)	PORT	8	2
Port G Nch-OD control register	PGNOCR	8	H'F988 (PORTS = 1)	PORT	8	2
Port H Nch-OD control register	PHNOCR	8	H'F989 (PORTS = 1)	PORT	8	2
Port G noise canceler enable register	PGNCE	8	H'F98A (PORTS = 1)	PORT	8	2
Port G noise canceler decision control register	PGNCMC	8	H'F98C (PORTS = 1)	PORT	8	2
Port G noise cancel cycle setting register	PGNCCS	8	H'F98E (PORTS = 1)	PORT	8	2

Register Name	Abbreviation	Number of bits	Address	Module	Data Width	Access States
Port I data direction register	PIDDR	8	H'F990	PORT	8	2
Port J data direction register	PJDDR	8	H'F991	PORT	8	2
Port I output data register	PIODR	8	H'F992	PORT	8	2
Port J output data register	PJODR	8	H'F993	PORT	8	2
Port I input data register	PIPIN	8	H'F994 (Read)	PORT	8	2
Port J input data register	PJPIN	8	H'F995 (Read)	PORT	8	2
Port J pull-up MOS control register	PJPCR	8	H'F997	PORT	8	2
Port I Nch-OD control register	PINOCR	8	H'F998	PORT	8	2
Port J Nch-OD control register	PJNOCR	8	H'F999	PORT	8	2
Receive control register 1	CCR1	8	H'FA40	CIR	8	2
Receive control register 2	CCR2	8	H'FA41	CIR	8	2
Receive status register	CSTR	8	H'FA42	CIR	8	2
Interrupt enable register	CEIR	8	H'FA43	CIR	8	2
Bit rate register	BRR	8	H'FA44	CIR	8	2
Receive data register 0 to 7	CIRRDR0 to CIRRDR7	8	H'FA45	CIR	8	2
Header minimum high-level period register	HHMIN	16	H'FA46	CIR	8	2
Header maximum high-level period register	HHMAX	16	H'FA48	CIR	8	2
Header minimum low-level period register	HLMIN	8	H'FA4A	CIR	8	2
Header maximum low-level period register	HLMAX	8	H'FA4B	CIR	8	2
Data level 0 minimum period register	DT0MIN	8	H'FA4C	CIR	8	2
Data level 0 maximum period register	DT0MAX	8	H'FA4D	CIR	8	2
Data level 1 minimum period register	DT1MIN	8	H'FA4E	CIR	8	2
Data level 1 maximum period register	DT1MAX	8	H'FA4F	CIR	8	2
Repeat header minimum low-level period register	RMIN	8	H'FA50	CIR	8	2
Repeat header maximum low-level period register	RMAX	8	H'FA51	CIR	8	2

Register Name	Abbreviation	Number of bits	Address	Module	Data Width	Access States
TDP timer counter_0	TDPCNT_0	16	H'FB40	TDP_0	16	2
TDP pulse width upper limit register_0	TDPWDMX_0	16	H'FB42	TDP_0	16	2
TDP pulse width lower limit register_0	TDPWDMN_0	16	H'FB44	TDP_0	16	2
TDP cycle upper limit register_0	TDPPDMX_0	16	H'FB46	TDP_0	16	2
TDP input capture register_0	TDPICR_0	16	H'FB48	TDP_0	16	2
TDP input capture buffer register_0	TDPICRF_0	16	H'FB4A	TDP_0	16	2
TDP status register_0	TDPCSR_0	8	H'FB4C	TDP_0	8	2
TDP control register 1_0	TDPCR1_0	8	H'FB4D	TDP_0	8	2
TDP interrupt enable register_0	TDPIER_0	8	H'FB4E	TDP_0	8	2
TDP control register 2_0	TDPCR2_0	8	H'FB4F	TDP_0	8	2
TDP cycle lower limit register_0	TDPPDMN_0	16	H'FB50	TDP_0	16	2
TDP timer counter_1	TDPCNT_1	16	H'FB60	TDP_1	16	2
TDP pulse width upper limit register_1	TDPWDMX_1	16	H'FB62	TDP_1	16	2
TDP pulse width lower limit register_1	TDPWDMN_1	16	H'FB64	TDP_1	16	2
TDP cycle upper limit register_1	TDPPDMX_1	16	H'FB66	TDP_1	16	2
TDP input capture register_1	TDPICR_1	16	H'FB68	TDP_1	16	2
TDP input capture buffer register_1	TDPICRF_1	16	H'FB6A	TDP_1	16	2
TDP status register_1	TDPCSR_1	8	H'FB6C	TDP_1	8	2
TDP control register 1_1	TDPCR1_1	8	H'FB6D	TDP_1	8	2
TDP interrupt enable register_1	TDPIER_1	8	H'FB6E	TDP_1	8	2
TDP control register 2_1	TDPCR2_1	8	H'FB6F	TDP_1	8	2
TDP cycle lower limit register_1	TDPPDMN_1	16	H'FB70	TDP_1	16	2
TDP timer counter_2	TDPCNT_2	16	H'FB80	TDP_2	16	2
TDP pulse width upper limit register_2	TDPWDMX_2	16	H'FB82	TDP_2	16	2
TDP pulse width lower limit register_2	TDPWDMN_2	16	H'FB84	TDP_2	16	2
TDP cycle upper limit register_2	TDPPDMX_2	16	H'FB86	TDP_2	16	2

Register Name	Abbreviation	Number of bits	Address	Module	Data Width	Access States
TDP input capture register_2	TDPICR_2	16	H'FB88	TDP_2	16	2
TDP input capture buffer register_2	TDPICRF_2	16	H'FB8A	TDP_2	16	2
TDP status register_2	TDPCSR_2	8	H'FB8C	TDP_2	8	2
TDP control register 1_2	TDPCR1_2	8	H'FB8D	TDP_2	8	2
TDP interrupt enable register_2	TDPIER_2	8	H'FB8E	TDP_2	8	2
TDP control register 2_2	TDPCR2_2	8	H'FB8F	TDP_2	8	2
TDP cycle lower limit register_2	TDPPDMN_2	16	H'FB90	TDP_2	16	2
TCM timer counter_0	TCMCNT_0	16	H'FBC0	TCM_0	16	2
TCM cycle upper limit register	TCMMLCM_0	16	H'FBC2	TCM_0	16	2
TCM input capture register_0	TCMICR_0	16	H'FBC4	TCM_0	16	2
TCM input capture buffer register_0	TCMICRF_0	16	H'FBC6	TCM_0	16	2
TCM status register_0	TCMCSR_0	8	H'FBC8	TCM_0	8	2
TCM control register_0	TCMCR_0	8	H'FBC9	TCM_0	8	2
TCM interrupt enable register_0	TCMIER_0	8	H'FBCA	TCM_0	8	2
TCM cycle lower limit register_0	TCMMINCM_0	16	H'FBCC	TCM_0	16	2
TCM timer counter_1	TCMCNT_1	16	H'FBD0	TCM_1	16	2
TCM cycle upper limit register_0	TCMMLCM_1	16	H'FBD2	TCM_1	16	2
TCM input capture register_1	TCMICR_1	16	H'FBD4	TCM_1	16	2
TCM input capture buffer register_1	TCMICRF_1	16	H'FBD6	TCM_1	16	2
TCM status register_1	TCMCSR_1	8	H'FBD8	TCM_1	8	2
TCM control register_1	TCMCR_1	8	H'FBD9	TCM_1	8	2
TCM interrupt enable register_1	TCMIER_1	8	H'FBDA	TCM_1	8	2
TCM cycle lower limit register_1	TCMMINCM_1	16	H'FBDC	TCM_1	16	2
TCM timer counter_2	TCMCNT_2	16	H'FBE0	TCM_2	16	2
TCM cycle upper limit register_2	TCMMLCM_2	16	H'FBE2	TCM_2	16	2
TCM input capture register_2	TCMICR_2	16	H'FBE4	TCM_2	16	2
TCM input capture buffer register_2	TCMICRF_2	16	H'FBE6	TCM_2	16	2
TCM status register_2	TCMCSR_2	8	H'FBE8	TCM_2	8	2
TCM control register_2	TCMCR_2	8	H'FBE9	TCM_2	8	2
TCM interrupt enable register_2	TCMIER_2	8	H'FBEA	TCM_2	8	2
TCM cycle lower limit register_2	TCMMINCM_2	16	H'FBEC	TCM_2	16	2
TCM timer counter_3	TCMCNT_3	16	H'FBF0	TCM_3	16	2
TCM cycle upper limit register_3	TCMMLCM_3	16	H'FBF2	TCM_3	16	2

Register Name	Abbreviation	Number of bits	Address	Module	Data Width	Access States
TCM input capture register_3	TCMICR_3	16	H'FBF4	TCM_3	16	2
TCM input capture buffer register_3	TCMICRF_3	16	H'FBF6	TCM_3	16	2
TCM status register_3	TCMCSR_3	8	H'FBF8	TCM_3	8	2
TCM control register_3	TCMCR_3	8	H'FBF9	TCM_3	8	2
TCM interrupt enable register_3	TCMIER_3	8	H'FBFA	TCM_3	8	2
TCM cycle lower limit register_3	TCMMINCM_3	16	H'FBFC	TCM_3	16	2
A/D data register A	ADDRA	16	H'FC00	A/D converter	16	2
A/D data register B	ADDRB	16	H'FC02	A/D converter	16	2
A/D data register C	ADDRC	16	H'FC04	A/D converter	16	2
A/D data register D	ADDRD	16	H'FC06	A/D converter	16	2
A/D data register E	ADDRE	16	H'FC08	A/D converter	16	2
A/D data register F	ADDRF	16	H'FC0A	A/D converter	16	2
A/D data register G	ADDRG	16	H'FC0C	A/D converter	16	2
A/D data register H	ADDRH	16	H'FC0E	A/D converter	16	2
A/D control/status register	ADCSR	16	H'FC10	A/D converter	16	2
A/D control register	ADCR	16	H'FC11	A/D converter	16	2
Receive buffer register	FRBR	8	H'FC20	SCIF	8	2
Transmitter holding register	FTHR	8	H'FC20	SCIF	8	2
Divisor latch L	FDLL	8	H'FC20	SCIF	8	2
Interrupt enable register	FIER	8	H'FC21	SCIF	8	2
Divisor latch H	FDLH	8	H'FC21	SCIF	8	2
Interrupt identification register	FIIR	8	H'FC22	SCIF	8	2
FIFO control register	FFCR	8	H'FC22	SCIF	8	2
Line control register	FLCR	8	H'FC23	SCIF	8	2
Modem control register	FMCR	8	H'FC24	SCIF	8	2
Line status register	FLSR	8	H'FC25	SCIF	8	2

Register Name	Abbreviation	Number of bits	Address	Module	Data Width	Access States
Modem status register	FMSR	8	H'FC26	SCIF	8	2
Scratch pad register	FSCR	8	H'FC27	SCIF	8	2
SCIF control register	SCIFCR	8	H'FC28	SCIF	8	2
FSI access host base address register H	FSIHBARH	8	H'FC50	FSI	8	2
FSI access host base address register L	FSIHBARL	8	H'FC51	FSI	8	2
FSI flash memory size register	FSISR	8	H'FC52	FSI	8	2
FSI command host base address register H	CMDHBARH	8	H'FC53	FSI	8	2
FSI command host base address register L	CMDHBARL	8	H'FC54	FSI	8	2
FSI command register	FSICMDR	8	H'FC55	FSI	8	2
FSILPC command status register 1	FSILSTR1	8	H'FC56	FSI	8	2
FSI general-purpose register 1	FSIGPR1	8	H'FC57	FSI	8	2
FSI general-purpose register 2	FSIGPR2	8	H'FC58	FSI	8	2
FSI general-purpose register 3	FSIGPR3	8	H'FC59	FSI	8	2
FSI general-purpose register 4	FSIGPR4	8	H'FC5A	FSI	8	2
FSI general-purpose register 5	FSIGPR5	8	H'FC5B	FSI	8	2
FSI general-purpose register 6	FSIGPR6	8	H'FC5C	FSI	8	2
FSI general-purpose register 7	FSIGPR7	8	H'FC5D	FSI	8	2
FSI general-purpose register 8	FSIGPR8	8	H'FC5E	FSI	8	2
FSI general-purpose register 9	FSIGPR9	8	H'FC5F	FSI	8	2
FSI general-purpose register A	FSIGPRA	8	H'FC60	FSI	8	2
FSI general-purpose register B	FSIGPRB	8	H'FC61	FSI	8	2
FSI general-purpose register C	FSIGPRC	8	H'FC62	FSI	8	2
FSI general-purpose register D	FSIGPRD	8	H'FC63	FSI	8	2
FSI general-purpose register E	FSIGPRE	8	H'FC64	FSI	8	2
FSI general-purpose register F	FSIGPRF	8	H'FC65	FSI	8	2
FSILPC control register	SLCR	8	H'FC66	FSI	8	2
FSI address register H	FSIARH	8	H'FC67	FSI	8	2
FSI address register M	FSIARM	8	H'FC68	FSI	8	2
FSI address register L	FSIARL	8	H'FC69	FSI	8	2

Register Name	Abbreviation	Number of bits	Address	Module	Data Width	Access States
FSI write data register HH	FSIWDRHH	8	H'FC6A	FSI	8	2
FSI write data register HL	FSIWDRHL	8	H'FC6B	FSI	8	2
FSI write data register LH	FSIWDRLH	8	H'FC6C	FSI	8	2
FSI write data register LL	FSIWDRLL	8	H'FC6D	FSI	8	2
FSILPC command status register 2	FSILSTR2	8	H'FC6E	FSI	8	2
FSI control register 1	FSICR1	8	H'FC90	FSI	8	2
FSI control register 2	FSICR2	8	H'FC91	FSI	8	2
FSI byte count register	FSIBNR	8	H'FC92	FSI	8	2
FSI instruction register	FSIINS	8	H'FC93	FSI	8	2
FSI read instruction register	FSIRDINS	8	H'FC94	FSI	8	2
FSI program instruction register	FSIPPINS	8	H'FC95	FSI	8	2
FSI status register	FSISTR	8	H'FC96	FSI	8	2
FSI transmit data register 0	FSITDR0	8	H'FC98	FSI	8	2
FSI transmit data register 1	FSITDR1	8	H'FC99	FSI	8	2
FSI transmit data register 2	FSITDR2	8	H'FC9A	FSI	8	2
FSI transmit data register 3	FSITDR3	8	H'FC9B	FSI	8	2
FSI transmit data register 4	FSITDR4	8	H'FC9C	FSI	8	2
FSI transmit data register 5	FSITDR5	8	H'FC9D	FSI	8	2
FSI transmit data register 6	FSITDR6	8	H'FC9E	FSI	8	2
FSI transmit data register 7	FSITDR7	8	H'FC9F	FSI	8	2
FSI receive data register	FSIRDR	8	H'FCA0	FSI	8	2
PWM duty setting register 0_A	PWMREG0_A	8	H'FD00	PWMU_A	8	2
PWM prescaler register 0_A	PWMPRE0_A	8	H'FD01	PWMU_A	8	2
PWM duty setting register 1_A	PWMREG1_A	8	H'FD02	PWMU_A	8	2
PWM prescaler register 1_A	PWMPRE1_A	8	H'FD03	PWMU_A	8	2
PWM duty setting register 2_A	PWMREG2_A	8	H'FD04	PWMU_A	8	2
PWM prescaler register 2_A	PWMPRE2_A	8	H'FD05	PWMU_A	8	2
PWM duty setting register 3_A	PWMREG3_A	8	H'FD06	PWMU_A	8	2
PWM prescaler register 3_A	PWMPRE3_A	8	H'FD07	PWMU_A	8	2
PWM duty setting register 4_A	PWMREG4_A	8	H'FD08	PWMU_A	8	2
PWM prescaler register 4_A	PWMPRE4_A	8	H'FD09	PWMU_A	8	2
PWM duty setting register 5_A	PWMREG5_A	8	H'FD0A	PWMU_A	8	2

Register Name	Abbreviation	Number of bits	Address	Module	Data Width	Access States
PWM prescaler register 5_A	PWMPRE5_A	8	H'FD0B	PWMU_A	8	2
PWM control register A_A	PWMCONA_A	8	H'FD0C	PWMU_A	8	2
PWM control register B_A	PWMCONB_A	8	H'FD0D	PWMU_A	8	2
PWM control register C_A	PWMCONC_A	8	H'FD0E	PWMU_A	8	2
PWM control register D_A	PWMCOND_A	8	H'FD0F	PWMU_A	8	2
PWM duty setting register 0_B	PWMREG0_B	8	H'FD10	PWMU_B	8	2
PWM prescaler register 0_B	PWMPRE0_B	8	H'FD11	PWMU_B	8	2
PWM duty setting register 1_B	PWMREG1_B	8	H'FD12	PWMU_B	8	2
PWM prescaler register 1_B	PWMPRE1_B	8	H'FD13	PWMU_B	8	2
PWM duty setting register 2_B	PWMREG2_B	8	H'FD14	PWMU_B	8	2
PWM prescaler register 2_B	PWMPRE2_B	8	H'FD15	PWMU_B	8	2
PWM duty setting register 3_B	PWMREG3_B	8	H'FD16	PWMU_B	8	2
PWM prescaler register 3_B	PWMPRE3_B	8	H'FD17	PWMU_B	8	2
PWM duty setting register 4_B	PWMREG4_B	8	H'FD18	PWMU_B	8	2
PWM prescaler register 4_B	PWMPRE4_B	8	H'FD19	PWMU_B	8	2
PWM duty setting register 5_B	PWMREG5_B	8	H'FD1A	PWMU_B	8	2
PWM prescaler register 5_B	PWMPRE5_B	8	H'FD1B	PWMU_B	8	2
PWM control register A_B	PWMCONA_B	8	H'FD1C	PWMU_B	8	2
PWM control register B_B	PWMCONB_B	8	H'FD1D	PWMU_B	8	2
PWM control register C_B	PWMCONC_B	8	H'FD1E	PWMU_B	8	2
PWM control register D_B	PWMCOND_B	8	H'FD1F	PWMU_B	8	2
Timer control register_1	TCR_1	8	H'FD40	TPU_1	8	2
Timer mode register_1	TMDR_1	8	H'FD41	TPU_1	8	2
Timer I/O control register_1	TIOR_1	8	H'FD42	TPU_1	8	2
Timer interrupt enable register_1	TIER_1	8	H'FD44	TPU_1	8	2
Timer status register_1	TSR_1	8	H'FD45	TPU_1	8	2
Timer counter _1	TCNT_1	16	H'FD46	TPU_1	16	2
Timer general register A_1	TGRA_1	16	H'FD48	TPU_1	16	2
Timer general register B_1	TGRB_1	16	H'FD4A	TPU_1	16	2
LPC channel 1 address register H	LADR1H	8	H'FDC0	LPC	8	2
LPC channel 1 address register L	LADR1L	8	H'FDC1	LPC	8	2
LPC channel 2 address register H	LADR2H	8	H'FDC2	LPC	8	2
LPC channel 2 address register L	LADR2L	8	H'FDC3	LPC	8	2

Register Name	Abbreviation	Number of bits	Address	Module	Data Width	Access States
SCIF address register H	SCIFADRH	8	H'FDC4	LPC	8	2
SCIF address register L	SCIFADRL	8	H'FDC5	LPC	8	2
LPC channel 4 address register H	LADR4H	8	H'FDD4	LPC	8	2
LPC channel 4 address register L	LADR4L	8	H'FDD5	LPC	8	2
Input data register 4	IDR4	8	H'FDD6	LPC	8	2
Output data register 4	ODR4	8	H'FDD7	LPC	8	2
Status register 4	STR4	8	H'FDD8	LPC	8	2
Host interface control register 4	HICR4	8	H'FDD9	LPC	8	2
SERIRQ control register 2	SIRQCR2	8	H'FDDA	LPC	8	2
SERIRQ control register 3	SIRQCR3	8	H'FDDB	LPC	8	2
Port 6 noise canceler enable register	P6NCE	8	H'FE00 (PORTS = 0)	PORT	8	2
Port 6 noise canceler decision control register	P6NCMC	8	H'FE01 (PORTS = 0)	PORT	8	2
Port 6 noise cancel cycle setting register	P6NCCS	8	H'FE02 (PORTS = 0)	PORT	8	2
Port C noise canceler enable register	PCNCE	8	H'FE03 (PORTS = 0)	PORT	8	2
Port C noise canceler decision control register	PCNCMC	8	H'FE04 (PORTS = 0)	PORT	8	2
Port C noise cancel cycle setting register	PCNCCS	8	H'FE05 (PORTS = 0)	PORT	8	2
Port G noise canceler enable register	PGNCE	8	H'FE06 (PORTS = 0)	PORT	8	2
Port G noise canceler decision control register	PGNCMC	8	H'FE07 (PORTS = 0)	PORT	8	2
Port G noise cancel cycle setting register	PGNCCS	8	H'FE08 (PORTS = 0)	PORT	8	2
Port H input data register	PHPIN	8	H'FE0C (Read) (PORTS = 0)	PORT	8	2
Port H data direction register	PHDDR	8	H'FE0C (Write) (PORTS = 0)	PORT	8	2
Port H output data register	PHODR	8	H'FE0D (PORTS = 0)	PORT	8	2
Port H Nch-OD control register	PHNOCR	8	H'FE0E (PORTS = 0)	PORT	8	2
Port control register 0	PTCNT0	8	H'FE10	PORT	8	2

Register Name	Abbreviation	Number of bits	Address	Module	Data Width	Access States
Port control register 1	PTCNT1	8	H'FE11	PORT	8	2
Port control register 2	PTCNT2	8	H'FE12	PORT	8	2
Port 9 pull-up MOS control register	P9PCR	8	H'FE14 (PORTS = 0)	PORT	8	2
Port G Nch-OD control register	PGNOCR	8	H'FE16 (PORTS = 0)	PORT	8	2
Port F Nch-OD control register	PFNOCR	8	H'FE19 (PORTS = 0)	PORT	8	2
Port C Nch-OD control register	PCNOCR	8	H'FE1C (PORTS = 0)	PORT	8	2
Port D Nch-OD control register	PDNOCR	8	H'FE1D (PORTS = 0)	PORT	8	2
Bidirectional data register 0MW	TWR0MW	8	H'FE20	LPC	8	2
Bidirectional data register 0SW	TWR0SW	8	H'FE20	LPC	8	2
Bidirectional data register 1	TWR1	8	H'FE21	LPC	8	2
Bidirectional data register 2	TWR2	8	H'FE22	LPC	8	2
Bidirectional data register 3	TWR3	8	H'FE23	LPC	8	2
Bidirectional data register 4	TWR4	8	H'FE24	LPC	8	2
Bidirectional data register 5	TWR5	8	H'FE25	LPC	8	2
Bidirectional data register 6	TWR6	8	H'FE26	LPC	8	2
Bidirectional data register 7	TWR7	8	H'FE27	LPC	8	2
Bidirectional data register 8	TWR8	8	H'FE28	LPC	8	2
Bidirectional data register 9	TWR9	8	H'FE29	LPC	8	2
Bidirectional data register 10	TWR10	8	H'FE2A	LPC	8	2
Bidirectional data register 11	TWR11	8	H'FE2B	LPC	8	2
Bidirectional data register 12	TWR12	8	H'FE2C	LPC	8	2
Bidirectional data register 13	TWR13	8	H'FE2D	LPC	8	2
Bidirectional data register 14	TWR14	8	H'FE2E	LPC	8	2
Bidirectional data register 15	TWR15	8	H'FE2F	LPC	8	2
Input data register 3	IDR3	8	H'FE30	LPC	8	2
Output data register 3	ODR3	8	H'FE31	LPC	8	2
Status register 3	STR3	8	H'FE32	LPC	8	2
Host interface control register 5	HICR5	8	H'FE33	LPC	8	2
LPC channel 3 address register H	LADR3H	8	H'FE34	LPC	8	2

Register Name	Abbreviation	Number of bits	Address	Module	Data Width	Access States
LPC channel 3 address register L	LADR3L	8	H'FE35	LPC	8	2
SERIRQ control register 0	SIRQCR0	8	H'FE36	LPC	8	2
SERIRQ control register 1	SIRQCR1	8	H'FE37	LPC	8	2
Input data register 1	IDR1	8	H'FE38	LPC	8	2
Output data register 1	ODR1	8	H'FE39	LPC	8	2
Status register 1	STR1	8	H'FE3A	LPC	8	2
Input data register 2	IDR2	8	H'FE3C	LPC	8	2
SERIRQ control register 4	SIRQCR4	8	H'FE3B	LPC	8	2
Output data register 2	ODR2	8	H'FE3D	LPC	8	2
Status register 2	STR2	8	H'FE3E	LPC	8	2
Host interface select register	HISEL	8	H'FE3F	LPC	8	2
Host interface control register 0	HICR0	8	H'FE40	LPC	8	2
Host interface control register 1	HICR1	8	H'FE41	LPC	8	2
Host interface control register 2	HICR2	8	H'FE42	LPC	8	2
Host interface control register 3	HICR3	8	H'FE43	LPC	8	2
Wakeup event interrupt mask register	WUEMR	8	H'FE45	INT	8	2
Port G output data register	PGODR	8	H'FE46 (PORTS = 0)	PORT	8	2
Port G input data register	PGPIN	8	H'FE47 (Read) (PORTS = 0)	PORT	8	2
Port G data direction register	PGDDR	8	H'FE47 (Write) (PORTS = 0)	PORT	8	2
Port F output data register	PFODR	8	H'FE49 (PORTS = 0)	PORT	8	2
Port E input data register	PEPIN	8	H'FE4A (Read) (write prohibited) (PORTS = 0)	PORT	8	2
Port F input data register	PFPIN	8	H'FE4B (Read) (PORTS = 0)	PORT	8	2
Port F data direction register	PFDDR	8	H'FE4B (Write) (PORTS = 0)	PORT	8	2
Port C output data register	PCODR	8	H'FE4C (PORTS = 0)	PORT	8	2
Port D output data register	PDODR	8	H'FE4D (PORTS = 0)	PORT	8	2

Register Name	Abbreviation	Number of bits	Address	Module	Data Width	Access States
Port C input data register	PCPIN	8	H'FE4E (Read) (PORTS = 0)	PORT	8	2
Port C data direction register	PCDDR	8	H'FE4E (Write) (PORTS = 0)	PORT	8	2
Port D input data register	PDPIN	8	H'FE4F (Read) (PORTS = 0)	PORT	8	2
Port D data direction register	PDDDR	8	H'FE4F (Write) (PORTS = 0)	PORT	8	2
Timer control register_0	TCR_0	8	H'FE50	TPU_0	8	2
Timer mode register_0	TMDR_0	8	H'FE51	TPU_0	8	2
Timer I/O control register H_0	TIORH_0	8	H'FE52	TPU_0	8	2
Timer I/O control register L_0	TIORL_0	8	H'FE53	TPU_0	8	2
Timer interrupt enable register_0	TIER_0	8	H'FE54	TPU_0	8	2
Timer status register_0	TSR_0	8	H'FE55	TPU_0	8	2
Timer counter _0	TCNT_0	16	H'FE56	TPU_0	16	2
Timer general register A_0	TGRA_0	16	H'FE58	TPU_0	16	2
Timer general register B_0	TGRB_0	16	H'FE5A	TPU_0	16	2
Timer general register C_0	TGRC_0	16	H'FE5C	TPU_0	16	2
Timer general register D_0	TGRD_0	16	H'FE5E	TPU_0	16	2
Timer control register_2	TCR_2	8	H'FE70	TPU_2	8	2
Timer mode register_2	TMDR_2	8	H'FE71	TPU_2	8	2
Timer I/O control register_2	TIOR_2	8	H'FE72	TPU_2	8	2
Timer interrupt enable register_2	TIER_2	8	H'FE74	TPU_2	8	2
Timer status register_2	TSR_2	8	H'FE75	TPU_2	8	2
Timer counter _2	TCNT_2	16	H'FE76	TPU_2	16	2
Timer general register A_2	TGRA_2	16	H'FE78	TPU_2	16	2
Timer general register B_2	TGRB_2	16	H'FE7A	TPU_2	16	2
System control register 3	SYSCR3	8	H'FE7D	SYSTEM	8	2
Module stop control register A	MSTPCRA	8	H'FE7E	SYSTEM	8	2
Module stop control register B	MSTPCRB	8	H'FE7F	SYSTEM	8	2
Keyboard matrix interrupt register	KMIMR	8	H'FE81 (RELOCATE = 1)	INT	8	2
Pull-up MOS control register	KMPCR	8	H'FE82 (RELOCATE = 1)	PORT	8	2

Register Name	Abbreviation	Number of bits	Address	Module	Data Width	Access States
Keyboard matrix interrupt register A	KMIMRA	8	H'FE83 (RELOCATE = 1)	INT	8	2
Wake-up sense control register	WUESCR	8	H'FE84	INT	8	2
Wake-up input interrupt status register	WUESR	8	H'FE85	INT	8	2
Wake-up enable register	WER	8	H'FE86	INT	8	2
Interrupt control register D	ICRD	8	H'FE87	INT	8	2
I^2C bus control register_2	ICCR_2	8	H'FE88	IIC_2	8	2
I^2C bus status register_2	ICSR_2	8	H'FE89	IIC_2	8	2
I^2C bus control Initialization register_2	ICRES_2	8	H'FE8A	IIC_2	8	2
I^2C bus control extended register_2	ICXR_2	8	H'FE8C	IIC_2	8	2
I^2C bus data register_2	ICDR_2	8	H'FE8E	IIC_2	8	2
Second slave address register_2	SARX_2	8	H'FE8E	IIC_2	8	2
I^2C bus mode register_2	ICMR_2	8	H'FE8F	IIC_2	8	2
Slave address register_2	SAR_2	8	H'FE8F	IIC_2	8	2
PWMX(D/A) control register	DACR	8	H'FEA0 (RELOCATE = 1)	PWMX	8	2
PWMX(D/A) data register AH	DADRAH	8	H'FEA0 (RELOCATE = 1)	PWMX	8	2
PWMX(D/A) data register AL	DADRAL	8	H'FEA1 (RELOCATE = 1)	PWMX	8	2
PWMX(D/A) data register BH	DADRBH	8	H'FEA6 (RELOCATE = 1)	PWMX	8	2
PWMX(D/A) counter H	DACNTH	8	H'FEA6 (RELOCATE = 1)	PWMX	8	2
PWMX(D/A) data register BL	DADRBL	8	H'FEA7 (RELOCATE = 1)	PWMX	8	2
PWMX(D/A) counter L	DACNTL	8	H'FEA7 (RELOCATE = 1)	PWMX	8	2
Flash code control status register	FCCS	8	H'FEA8	ROM	8	2
Flash program code select register	FPCS	8	H'FEA9	ROM	8	2
Flash erace code select register	FECS	8	H'FEAA	ROM	8	2
Flash key code register	FKEY	8	H'FEAC	ROM	8	2
Flash MAT select register	FMATS	8	H'FEAD	ROM	8	2

Register Name	Abbreviation	Number of bits	Address	Module	Data Width	Access States
Flash transfer destination address register	FTDAR	8	H'FEAE	ROM	8	2
Timer start register	TSTR	8	H'FEB0	TPU common	8	2
Timer synchro register	TSYR	8	H'FEB1	TPU common	8	2
Keyboard control register 1_0	KBCR1_0	8	H'FEC0	PS2_0	8	2
Keyboard data buffer transmit data register_0	KBTR_0	8	H'FEC1	PS2_0	8	2
Keyboard control register 1_1	KBCR1_1	8	H'FEC2	PS2_1	8	2
Keyboard data buffer transmit data register_1	KBTR_1	8	H'FEC3	PS2_1	8	2
Keyboard control register 1_2	KBCR1_2	8	H'FEC4	PS2_2	8	2
Keyboard data buffer transmit data register_2	KBTR_2	8	H'FEC5	PS2_2	8	2
Timer XY control register	TCRXY	8	H'FEC6	TMR_XY	8	2
Timer control register_Y	TCR_Y	8	H'FEC8 (RELOCATE = 1)	TMR_Y	8	2
Timer control/status register_Y	TCSR_Y	8	H'FEC9 (RELOCATE = 1)	TMR_Y	8	2
Time constant register A_Y	TCORA_Y	8	H'FECA (RELOCATE = 1)	TMR_Y	8	2
Time constant register B_Y	TCORB_Y	8	H'FECB (RELOCATE = 1)	TMR_Y	8	2
Timer counter _Y	TCNT_Y	8	H'FECC (RELOCATE = 1)	TMR_Y	8	2
I^2C bus data register_1	ICDR_1	8	H'FECE (RELOCATE = 1)	IIC_1	8	2
Second slave address register_1	SARX_1	8	H'FECE (RELOCATE = 1)	IIC_1	8	2
I^2C bus mode register_1	ICMR_1	8	H'FECF (RELOCATE = 1)	IIC_1	8	2
Slave address register_1	SAR_1	8	H'FECF (RELOCATE = 1)	IIC_1	8	2
I^2C bus control register_1	ICCR_1	8	H'FED0 (RELOCATE = 1)	IIC_1	8	2
I^2C bus status register_1	ICSR_1	8	H'FED1 (RELOCATE = 1)	IIC_1	8	2

Register Name	Abbreviation	Number of bits	Address	Module	Data Width	Access States
Keyboard control register 1_3	KBCR1_3	8	H'FED2	PS2_3	8	2
Keyboard data buffer transmit data register_3	KBTR_3	8	H'FED3	PS2_3	8	2
I^2C bus control extended register_0	ICXR_0	8	H'FED4	IIC_0	8	2
I^2C bus control extended register_1	ICXR_1	8	H'FED5	IIC_1	8	2
Keyboard control register H_0	KBCRH_0	8	H'FED8	PS2_0	8	2
Keyboard control register L_0	KBCRL_0	8	H'FED9	PS2_0	8	2
Keyboard data buffer register_0	KBBR_0	8	H'FEDA	PS2_0	8	2
Serial mode register_1	SMR_1	8	H'FF88	SCI_1	8	2
I^2C bus control register_1	ICCR_1	8	H'FF88 (RELOCATE = 0)	IIC_1	8	2
Bit rate register_1	BRR_1	8	H'FF89	SCI_1	8	2
I^2C bus status register_1	ICSR_1	8	H'FF89 (RELOCATE = 0)	IIC_1	8	2
Serial control register_1	SCR_1	8	H'FF8A	SCI_1	8	2
Transmit data register_1	TDR_1	8	H'FF8B	SCI_1	8	2
Serial status register_1	SSR_1	8	H'FF8C	SCI_1	8	2
Receive data register_1	RDR_1	8	H'FF8D	SCI_1	8	2
Smart card mode register_1	SCMR_1	8	H'FF8E	SCI_1	8	2
I^2C bus data register_1	ICDR_1	8	H'FF8E (RELOCATE = 0)	IIC_1	8	2
Second slave address register_1	SARX_1	8	H'FF8E (RELOCATE = 0)	IIC_1	8	2
I^2C bus mode register_1	ICMR_1	8	H'FF8F (RELOCATE = 0)	IIC_1	8	2
Slave address register_1	SAR_1	8	H'FF8F (RELOCATE = 0)	IIC_1	8	2
PWMX(D/A) control register	DACR	8	H'FFA0 (RELOCATE = 0)	PWMX	8	2
PWMX(D/A) data register AH	DADRAH	8	H'FFA0 (RELOCATE = 0)	PWMX	8	2
Serial mode register_2	SMR_2	8	H'FFA0	SCI_2	8	2
PWMX(D/A) data register AL	DADRAL	8	H'FFA1 (RELOCATE = 0)	PWMX	8	2
Bit rate register_2	BRR_2	8	H'FFA1	SCI_2	8	2
Serial control register_2	SCR_2	8	H'FFA2	SCI_2	8	2

Register Name	Abbreviation	Number of bits	Address	Module	Data Width	Access States
Transmit data register_2	TDR_2	8	H'FFA3	SCI_2	8	2
Serial status register_2	SSR_2	8	H'FFA4	SCI_2	8	2
Receive data register_2	RDR_2	8	H'FFA5	SCI_2	8	2
Smart card mode register_2	SCMR_2	8	H'FFA6	SCI_2	8	2
PWMX(D/A) counter H	DACNTH	8	H'FFA6 (RELOCATE = 0)	PWMX	8	2
PWMX(D/A) data register BH	DADRBH	8	H'FFA6 (RELOCATE = 0)	PWMX	8	2
PWMX(D/A) counter L	DACNTL	8	H'FFA7 (RELOCATE = 0)	PWMX	8	2
PWMX(D/A) data register BL	DADRBL	8	H'FFA7 (RELOCATE = 0)	PWMX	8	2
Timer control/status register_0	TCSR_0	8	H'FFA8 (Write)	WDT_0	16	2
Timer control/status register_0	TCSR_0	8	H'FFA8 (Read)	WDT_0	8	2
Timer counter _0	TCNT_0	8	H'FFA8 (Write)	WDT_0	16	2
Timer counter _0	TCNT_0	8	H'FFA9 (Read)	WDT_0	8	2
Port A output data register	PAODR	8	H'FFAA (PORTS = 0)	PORT	8	2
Port A input data register	PAPIN	8	H'FFAB (Read) (PORTS = 0)	PORT	8	2
Port A data direction register	PADDR	8	H'FFAB (Write) (PORTS = 0)	PORT	8	2
Port 1 pull-up MOS control register	P1PCR	8	H'FFAC (PORTS = 0)	PORT	8	2
Port 2 pull-up MOS control register	P2PCR	8	H'FFAD (PORTS = 0)	PORT	8	2
Port 3 pull-up MOS control register	P3PCR	8	H'FFAE (PORTS = 0)	PORT	8	2
Port 1 data direction register	P1DDR	8	H'FFB0 (PORTS = 0)	PORT	8	2
Port 2 data direction register	P2DDR	8	H'FFB1 (PORTS = 0)	PORT	8	2
Port 1 data register	P1DR	8	H'FFB2 (PORTS = 0)	PORT	8	2
Port 2 data register	P2DR	8	H'FFB3 (PORTS = 0)	PORT	8	2

Register Name	Abbreviation	Number of bits	Address	Module	Data Width	Access States
Port 3 data direction register	P3DDR	8	H'FFB4 (PORTS = 0)	PORT	8	2
Port 4 data direction register	P4DDR	8	H'FFB5 (PORTS = 0)	PORT	8	2
Port 3 data register	P3DR	8	H'FFB6 (PORTS = 0)	PORT	8	2
Port 4 data register	P4DR	8	H'FFB7 (PORTS = 0)	PORT	8	2
Port 5 data direction register	P5DDR	8	H'FFB8 (PORTS = 0)	PORT	8	2
Port 6 data direction register	P6DDR	8	H'FFB9 (PORTS = 0)	PORT	8	2
Port 5 data register	P5DR	8	H'FFBA (PORTS = 0)	PORT	8	2
Port 6 data register	P6DR	8	H'FFBB (PORTS = 0)	PORT	8	2
Port B output data register	PBODR	8	H'FFBC (PORTS = 0)	PORT	8	2
Port 8 data direction register	P8DDR	8	H'FFBD (Write) (PORTS = 0)	PORT	8	2
Port B input data register	PBPIN	8	H'FFBD (Read) (PORTS = 0)	PORT	8	2
Port 7 input data register	P7PIN	8	H'FFBE (Read) (PORTS = 0)	PORT	8	2
Port B data direction register	PBDDR	8	H'FFBE (Write) (PORTS = 0)	PORT	8	2
Port 8 data register	P8DR	8	H'FFBF (PORTS = 0)	PORT	8	2
Port 9 data direction register	P9DDR	8	H'FFC0 (PORTS = 0)	PORT	8	2
Port 9 data register	P9DR	8	H'FFC1 (PORTS = 0)	PORT	8	2
Interrupt enable register	IER	8	H'FFC2	INT	8	2
Serial timer control register	STCR	8	H'FFC3	SYSTEM	8	2
System control register	SYSCR	8	H'FFC4	SYSTEM	8	2
Mode control register	MDCR	8	H'FFC5	SYSTEM	8	2
Bus control register	BCR	8	H'FFC6	BSC	8	2
Wait state control register	WSCR	8	H'FFC7	BSC	8	2

Register Name	Abbreviation	Number of bits	Address	Module	Data Width	Access States
Timer control register_0	TCR_0	8	H'FFC8	TMR_0	8	2
Timer control register_1	TCR_1	8	H'FFC9	TMR_1	8	2
Timer control/status register_0	TCSR_0	8	H'FFCA	TMR_0	8	2
Timer control/status register_1	TCSR_1	8	H'FFCB	TMR_1	8	2
Time constant register A_0	TCORA_0	8	H'FFCC	TMR_0	16	2
Time constant register A_1	TCORA_1	8	H'FFCD	TMR_1	16	2
Time constant register B_0	TCORB_0	8	H'FFCE	TMR_0	16	2
Time constant register B_1	TCORB_1	8	H'FFCF	TMR_1	16	2
Timer counter _0	TCNT_0	8	H'FFD0	TMR_0	16	2
Timer counter _1	TCNT_1	8	H'FFD1	TMR_1	16	2
I^2C bus control register_0	ICCR_0	8	H'FFD8	IIC_0	8	2
I^2C bus status register_0	ICSR_0	8	H'FFD9	IIC_0	8	2
I^2C bus data register_0	ICDR_0	8	H'FFDE	IIC_0	8	2
Second slave address register_0	SARX_0	8	H'FFDE	IIC_0	8	2
I^2C bus mode register_0	ICMR_0	8	H'FFDF	IIC_0	8	2
Slave address register_0	SAR_0	8	H'FFDF	IIC_0	8	2
Keyboard control register H_3	KBCRH_3	8	H'FFE0	PS2_3	8	2
Keyboard control register L_3	KBCRL_3	8	H'FFE1	PS2_3	8	2
Keyboard data buffer register_3	KBBR_3	8	H'FFE2	PS2_3	8	2
Keyboard control register 2_3	KBCR2_3	8	H'FFE3	PS2_3	8	2
Timer control/status register	TCSR_1	8	H'FFEA (Write)	WDT_1	16	2
Timer control/status register	TCSR_1	8	H'FFEA (Read)	WDT_1	8	2
Timer counter _1	TCNT_1	8	H'FFEA (Write)	WDT_1	16	2
Timer counter _1	TCNT_1	8	H'FFEB (Read)	WDT_1	8	2
Timer control register_X	TCR_X	8	H'FFF0	TMR_X	8	2
Timer control register_Y	TCR_Y	8	H'FFF0 (RELOCATE = 0)	TMR_Y	8	2
Keyboard matrix interrupt register	KMIMR	8	H'FFF1 (RELOCATE = 0)	INT	8	2
Timer control/status register_X	TCSR_X	8	H'FFF1	TMR_X	8	2
Timer control/status register_Y	TCSR_Y	8	H'FFF1 (RELOCATE = 0)	TMR_Y	8	2
Pull-up MOS control register	KMPCR	8	H'FFF2 (RELOCATE = 0)	PORT	8	2

Register Name	Abbreviation	Number of bits	Address	Module	Data Width	Access States
Input capture register R	TICRR	8	H'FFF2	TMR_X	8	2
Time constant register A_Y	TCORA_Y	8	H'FFF2 (RELOCATE = 0)	TMR_Y	8	2
Input capture register F	TICRF	8	H'FFF3	TMR_X	8	2
Time constant register B_Y	TCORB_Y	8	H'FFF3 (RELOCATE = 0)	TMR_Y	8	2
Keyboard matrix interrupt register A	KMIMRA	8	H'FFF3 (RELOCATE = 0)	INT	8	2
Timer counter _X	TCNT_X	8	H'FFF4	TMR_X	8	2
Timer counter _Y	TCNT_Y	8	H'FFF4 (RELOCATE = 0)	TMR_Y	8	2
Time constant register C	TCORC	8	H'FFF5	TMR_X	8	2
Time constant register A_X	TCORA_X	8	H'FFF6	TMR_X	8	2
Time constant register B_X	TCORB_X	8	H'FFF7	TMR_X	8	2
Timer connection register I	TCONRI	8	H'FFFC	TMR_X	8	2
Timer connection register S	TCONRS	8	H'FFFE	TMR_X, TMR_Y	8	2

27.2 Register Bits

Register addresses and bit names of the on-chip peripheral modules are described below.

Each line covers eight bits, and 16-bit registers are shown as 2 lines.

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
PIDDR	PI7DDR	PI6DDR	PI5DDR	PI4DDR	PI3DDR	PI2DDR	PI1DDR	PI0DDR	PORT
PJDDR	PJ7DDR	PJ6DDR	PJ5DDR	PJ4DDR	PJ3DDR	PJ2DDR	PJ1DDR	PJ0DDR	
PIODR	PI7ODR	PI6ODR	PI5ODR	PI4ODR	PI3ODR	PI2ODR	PI1ODR	PI0ODR	
PJODR	PJ7ODR	PJ6ODR	PJ5ODR	PJ4ODR	PJ3ODR	PJ2ODR	PJ1ODR	PJ0ODR	
PIPIN	PI7PIN	PI6PIN	PI5PIN	PI4PIN	PI3PIN	PI2PIN	PI1PIN	PI0PIN	
PJPIN	PJ7PIN	PJ6PIN	PJ5PIN	PJ4PIN	PJ3PIN	PJ2PIN	PJ1PIN	PJ0PIN	
PJPCR	PJ7PCR	PJ6PCR	PJ5PCR	PJ4PCR	PJ3PCR	PJ2PCR	PJ1PCR	PJ0PCR	
PINOCR	PI7NOCR	PI6NOCR	PI5NOCR	PI4NOCR	PI3NOCR	PI2NOCR	PI1NOCR	PI0NOCR	
PJNOCR	PJ7NOCR	PJ6NOCR	PJ5NOCR	PJ4NOCR	PJ3NOCR	PJ2NOCR	PJ1NOCR	PJ0NOCR	
CCR1	CIRE	SRES	CPHS	MLS	REPRCVE	—	CLK1	CLK0	CIR
CCR2	TFM1	TFM0	—	—	—	—	—	—	
CSTR	CIRBUSY	CIRRDRF	REPF	OVRF	REND	ABF	FRF	HEADF	
CEIR	—	—	REPIE	OVEIE	RENDIE	ABIE	FREIE	HEADFIE	
BRR	BRR7	BRR6	BRR5	BRR4	BRR3	BRR2	BRR1	BRR0	
CIRRDR0 to 7	CIRRDR7	CIRRDR6	CIRRDR5	CIRRDR4	CIRRDR3	CIRRDR2	CIRRDR1	CIRRDR0	
HHMIN	RFMBIN4	RFMBIN3	RFMBIN2	RFMBIN1	RFMBIN0	—	HHMIN9	HHMIN8	
	HHMIN7	HHMIN6	HHMIN5	HHMIN4	HHMIN3	HHMIN2	HHMIN1	HHMIN0	
HHMAX	FLT1	FLT0	FLTE	FLTCK1	FLTCK0	—	HHMAX9	HHMAX8	
	HHMAX7	HHMAX6	HHMAX5	HHMAX4	HHMAX3	HHMAX2	HHMAX1	HHMAX0	
HLMIN	HLMIN7	HLMIN6	HLMIN5	HLMIN4	HLMIN3	HLMIN2	HLMIN1	HLMIN0	
HLMAX	HLMAX7	HLMAX6	HLMAX5	HLMAX4	HLMAX3	HLMAX2	HLMAX1	HLMAX0	
DT1MIN	DT1MIN7	DT1MIN6	DT1MIN5	DT1MIN4	DT1MIN3	DT1MIN2	DT1MIN1	DT1MIN0	
DT1MAX	DT1MAX7	DT1MAX6	DT1MAX5	DT1MAX4	DT1MAX3	DT1MAX2	DT1MAX1	DT1MAX0	
DT0MIN	DT0MIN7	DT0MIN6	DT0MIN5	DT0MIN4	DT0MIN3	DT0MIN2	DT0MIN1	DT0MIN0	
DT0MAX	DT0MAX7	DT0MAX6	DT0MAX5	DT0MAX4	DT0MAX3	DT0MAX2	DT0MAX1	DT0MAX0	
RMIN	RMIN7	RMIN6	RMIN5	RMIN4	RMIN3	RMIN2	RMIN1	RMIN0	
RMAX	RMAX7	RMAX6	RMAX5	RMAX4	RMAX3	RMAX2	RMAX1	RMAX0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
TDPCNT_0	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	TDP_0
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TDPPDMX_0	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TDPPDMN_0	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TDPWDMX_0	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TDPICR_0	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TDPICRF_0	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TDPCSR_0	OVF	TWDMXOVF	TWDMNUDF	TPDMXOVF	ICPF	CMF	CKSEG	TPDMNUDF	
TDPCR1_0	CST	POCTL	CPSPE	IEDG	TDPMDS	CKS2	CKS1	CKS0	
TDPIER_0	OVIE	TWDMXIE	TWDMNIE	TPDMXIE	ICPIE	CMIE	TDPIPE	TPDMNIE	
TDPCR2_0	PMMS	MCICTL	—	—	—	—	—	—	
TDPWDMN_0	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TDPCNT_1	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	TDP_1
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TDPPDMX_1	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TDPPDMN_1	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TDPWDMX_1	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TDPICR_1	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TDPICRF_1	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
TDPCSR_1	OVF	TWDMXOVF	TWDMNUDF	TPDMXOVF	ICPF	CMF	CKSEG	TPDMNUDF	TDP_1
TDPCR1_1	CST	POCTL	CPSPE	IEDG	TDPMDS	CKS2	CKS1	CKS0	
TDPIER_1	OVIE	TWDMXIE	TWDMNIE	TPDMXIE	ICPIE	CMIE	TDPIPE	TPDMNIE	
TDPCR2_1	PMMS	MCICTL	—	—	—	—	—	—	
TDPWDMN_1	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TDPCNT_2	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	TDP_2
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TDPPDMX_2	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TDPPDMN_2	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TDPWDMX_2	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TDPICR_2	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TDPICRF_2	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TDPCSR_2	OVF	TWDMXOVF	TWDMNUDF	TPDMXOVF	ICPF	CMF	CKSEG	TPDMNUDF	
TDPCR1_2	CST	POCTL	CPSPE	IEDG	TDPMDS	CKS2	CKS1	CKS0	
TDPIER_2	OVIE	TWDMXIE	TWDMNIE	TPDMXIE	ICPIE	CMIE	TDPIPE	TPDMNIE	
TDPCR2_2	PMMS	MCICTL	—	—	—	—	—	—	
TDPWDMN_2	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TCMCNT_0	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	TCM_0
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TCMMLCM_0	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TCMICR_0	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TCMICRF_0	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
TCMCSR_0	OVF	MAXOVF	CMF	CKSEG	ICPF	MINUDF	MCICTL	—	TCM_0
TCMCR_0	CST	POCTL	CPSPE	IEDG	TCMMDS	CKS2	CKS1	CKS0	
TCMIER_0	OVIE	MAXOVIE	CMIE	TCMIPE	ICPIE	MINUDIE	CMMS	—	
TCMMINCM_0	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TCMCNT_1	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	TCM_1
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TCMMLCM_1	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TCMICR_1	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TCMICRF_1	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TCMCSR_1	OVF	MAXOVF	CMF	CKSEG	ICPF	MINUDF	MCICTL	—	
TCMCR_1	CST	POCTL	CPSPE	IEDG	TCMMDS	CKS2	CKS1	CKS0	
TCMIER_1	OVIE	MAXOVIE	CMIE	TCMIPE	ICPIE	MINUDIE	CMMS	—	
TCMMINCM_1	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TCMCNT_2	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	TCM_2
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TCMMLCM_2	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TCMICR_2	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TCMICRF_2	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TCMCSR_2	OVF	MAXOVF	CMF	CKSEG	ICPF	MINUDF	MCICTL	—	
TCMCR_2	CST	POCTL	CPSPE	IEDG	TCMMDS	CKS2	CKS1	CKS0	
TCMIER_2	OVIE	MAXOVIE	CMIE	TCMIPE	ICPIE	MINUDIE	CMMS	—	
TCMMINCM_2	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
TCMCNT_3	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	TCM_3
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TCMMLCM_3	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TCMICR_3	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TCMICRF_3	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TCMCSR_3	OVF	MAXOVF	CMF	CKSEG	ICPF	MINUDF	MCICTL	—	
TCMCR_3	CST	POCTL	CPSPE	IEDG	TCMMDS	CKS2	CKS1	CKS0	
TCMIER_3	OVIE	MAXOVIE	CMIE	TCMIPE	ICPIE	MINUDIE	CMMS	—	
TCMMINCM_3	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
ADDRA	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	A/D converter
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
ADDRB	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
ADDRC	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
ADDRD	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
ADDRE	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
ADDRF	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
ADDRG	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
ADDRH	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
ADCSR	ADF	ADIE	ADST	—	CH3	CH2	CH1	CH0	
ADCR	TRGS1	TRGS0	SCANE	SCANS	CKS1	CKS0	ADSTCLR	—	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
FRBR	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	SCIF
FTHR	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
FDLL	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
FIER	—	—	—	—	EDSSI	ELSI	ETBEI	ERBFI	
FDLH	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
FIIR	FIFOE1	FIFOE0	—	—	INTID2	INTID1	INTID0	INTPEND	
FFCR	RCVRTRIG1	RCVRTRIG0	—	—	DMAMODE	XMITFRST	RCVRFRST	FIFOE	
FLCR	DLAB	BREAK	STICKPARITY	EPS	PEN	STOP	CLS1	CLS0	
FMCR	—	—	—	LOOPBACK	OUT2	OUT1	RTS	DTR	
FLSR	RXFIFOERR	TEMT	THRE	BI	FE	PE	OE	DR	
FMSR	DCD	RI	DSR	CTS	DDCD	TERI	DDSR	DCTS	
FSCR	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
SCIFCR	SCIFOE1	SCIFOE0	—	OUT2LOOP	CKSEL1	CKSEL0	SCIFRST	REGRST	
FSIHBARH	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24	FSI
FSIHBARL	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16	
FSISR	—	—	—	—	—	—	FSIMS1	FSIMS0	
CMDHBARH	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24	
CMDHBARL	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16	
FSICMDR	bit7	bt6	bit5	bit4	bit3	bit2	bit1	bit0	
FSILSTR1	CMDBUSY	FSICMDI	FSIDMYE	FSIWBUS	YFSIWI	LFBUSY	—	—	
FSILSTR2	—	—	—	FSIDWBUSY	FSIDRBUSY	SIZE2	SIZE1	SIZE0	
FSIGPR1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
FSIGPR2	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
FSIGPR3	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
FSIGPR4	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
FSIGPR5	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
FSIGPR6	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
FSIGPR7	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
FSIGPR8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
FSIGPR9	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
FSIGPRA	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
FSIGPRB	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
FSIGPRC	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	FSI
FSIGPRD	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
FSIGPRE	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
FSIGPRF	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
SLCR	FSILE	FSICMDIE	FSIWIE	FSIWDRLL	FSICR1	—	—	—	
FSIARH	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16	
FSIARM	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
FSIARL	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
FSIWDRHH	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24	
FSIWDRHL	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16	
FSIWDRLH	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
FSIWDRLL	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
FSICR1	SRES	FSIE	FRDE	AAIE	CPHS	CPOS	—	CKSEL	
FSICR2	TE	RE	FSITEIE	FSIRXIE	—	—	—	—	
FSIBNR	TBN3	TBN2	TBN1	TBN0	—	RBN2	RBN1	RBN0	
FSINS	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
FSIRDINS	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
FSIPPINS	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
FSISTR	FSITEI	OBF	FSIRXI	—	—	—	—	—	
FSITDR0	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
FSITDR1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
FSITDR2	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
FSITDR3	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
FSITDR4	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
FSITDR5	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
FSITDR6	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
FSITDR7	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
FSIRDR	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
PWMREG0_A	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	PWMU_A
PWMPRE0_A	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
PWMREG1_A	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
PWMPRE1_A	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
PWMREG2_A	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
PWMPRE2_A	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
PWMREG3_A	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
PWMPRE3_A	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
PWMREG4_A	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
PWMPRE4_A	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
PWMREG5_A	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
PWMPRE5_A	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
PWMCONA_A	CLK1	CLK0	—	—	—	—	—	—	
PWMCONB_A	—	—	PWM5E	PWM4E	PWM3E	PWM2E	PWM1E	PWM0E	
PWMCONC_A	—	CNTMD01	PWMSL5	PWMSL4	PWMSL3	PWMSL2	PWMSL1	PWMSL0	
PWMCOND_A	PH5S	PH4S	PH3S	PH2S	PH1S	PH0S	CNTMD45	CNTMD23	
PWMREG0_B	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	PWMU_B
PWMPRE0_B	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
PWMREG1_B	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
PWMPRE1_B	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
PWMREG2_B	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
PWMPRE2_B	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
PWMREG3_B	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
PWMPRE3_B	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
PWMREG4_B	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
PWMPRE4_B	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
PWMREG5_B	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
PWMPRE5_B	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
PWMCONA_B	CLK1	CLK0	—	—	—	—	—	—	
PWMCONB_B	—	—	PWM5E	PWM4E	PWM3E	PWM2E	PWM1E	PWM0E	
PWMCONC_B	—	CNTMD01	PWMSL5	PWMSL4	PWMSL3	PWMSL2	PWMSL1	PWMSL0	
PWMCOND_B	PH5S	PH4S	PH3S	PH2S	PH1S	PH0S	CNTMD45	CNTMD23	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
TCR_1	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_1
TMDR_1	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_1	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_1	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_1	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNT_1	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TGRA_1	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TGRB_1	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
LADR1H	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	LPC
LADR1L	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
LADR2H	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
LADR2L	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
SCIFADRH	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
SCIFADRL	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
LADR4H	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
LADR4L	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
IDR4	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
ODR4	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
STR4	DBU47	DBU46	DBU45	DBU44	C/$\overline{D}$4	DBU42	IBF4	OBF4	
HICR4	—	LPC4E	IBFIE4	—	—	—	—	—	
SIRQCR2	IEDIR3	IEDIR4	IRQ11E4	IRQ10E4	IRQ9E4	IRQ6E4	SMIE4	—	PORT
SIRQCR3	SELIRQ15	SELIRQ14	SELIRQ13	SELIRQ8	SELIRQ7	SELIRQ5	SELIRQ4	SELIRQ3	
P6NCE	P67NCE	P66NCE	P65NCE	P64NCE	P63NCE	P62NCE	P61NCE	P60NCE	
P6NCMC	P67NCMC	P66NCMC	P65NCMC	P64NCMC	P63NCMC	P62NCMC	P61NCMC	P60NCMC	
P6NCCS	—	—	—	—	—	P6NCCK2	P6NCCK1	P6NCCK0	
PCNCE	PC7NCE	PC6NCE	PC5NCE	PC4NCE	PC3NCE	PC2NCE	PC1NCE	PC0NCE	
PCNCMC	PC7NCMC	PC6NCMC	PC5NCMC	PC4NCMC	PC3NCMC	PC2NCMC	PC1NCMC	PC0NCMC	
PCNCCS	—	—	—	—	—	PCNCCK2	PCNCCK1	PCNCCK0	
PGNCE	PG7NCE	PG6NCE	PG5NCE	PG4NCE	PG3NCE	PG2NCE	PG1NCE	PG0NCE	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
PGNCMC	PG7NCMC	PG6NCMC	PG5NCMC	PG4NCMC	PG3NCMC	PG2NCMC	PG1NCMC	PG0NCMC	PORT
PGNCCS	—	—	—	—	—	PGNCCK2	PGNCCK1	PGNCCK0	
PHPIN	—	—	PH5PIN	PH4PIN	PH3PIN	PH2PIN	PH1PIN	PH0PIN	
PHDDR	—	—	PH5DDR	PH4DDR	PH3DDR	PH2DDR	PH1DDR	PH0DDR	
PHODR	—	—	PH5ODR	PH4ODR	PH3ODR	PH2ODR	PH1ODR	PH0ODR	
PHNOCR	—	—	PH5NOCR	PH4NOCR	PH3NOCR	PH2NOCR	PH1NOCR	PH0NOCR	
PTCNT0	—	—	—	—	—	—	—	EXCLS	
PTCNT1	IIC1BS	IIC1AS	—	—	IIC0BS	IIC0AS	—	—	
PTCNT2	—	TxD2RS	RxD2RS	TxD1RS	RxD1RS	—	PORTS	—	
P9PCR	—	—	P95PCR	P94PCR	P93PCR	P92PCR	P91PCR	P90PCR	
PGNOCR	PG7NOCR	PG6NOCR	PG5NOCR	PG4NOCR	PG3NOCR	PG2NOCR	PG1NOCR	PG0NOCR	
PFNOCR	PF7NOCR	PF6NOCR	PF5NOCR	PF4NOCR	PF3NOCR	PF2NOCR	PF1NOCR	PF0NOCR	
PCNOCR	PC7NOCR	PC6NOCR	PC5NOCR	PC4NOCR	PC3NOCR	PC2NOCR	PC1NOCR	PC0NOCR	
PDNOCR	PD7NOCR	PD6NOCR	PD5NOCR	PD4NOCR	PD3NOCR	PD2NOCR	PD1NOCR	PD0NOCR	
TWR0MW	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	LPC
TWR0SW	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TWR1	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TWR2	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TWR3	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TWR4	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TWR5	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TWR6	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TWR7	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TWR8	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TWR9	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TWR10	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TWR11	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TWR12	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TWR13	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TWR14	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TWR15	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
IDR3	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
ODR3	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	LPC
STR3*[2]	IBF3B	OBF3B	MWMF	SWMF	C/$\overline{D}$3	DBU32	IBF3A	OBF3A	
STR3*[3]	DBU37	DBU36	DBU35	DBU34	C/$\overline{D}$3	DBU32	IBF3	OBF3	
HICR5	OBEIE	OBEI	—	—	SCIFE	—	—	—	
LADR3H	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
LADR3L	bit 7	bit 6	bit 5	bit 4	bit 3	—	bit 1	TWRE	
SIRQCR0	Q/$\overline{C}$	UPSEL	IEDIR	SMIE3B	SMIE3A	SMIE2	IRQ12E1	IRQ1E1	
SIRQCR1	IRQ11E3	IRQ10E3	IRQ9E3	IRQ6E3	IRQ11E2	IRQ10E2	IRQ9E2	IRQ6E2	
IDR1	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
ODR1	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
STR1	DBU17	DBU16	DBU15	DBU14	C/$\overline{D}$1	DBU12	IBF1	OBF1	
SIRQCR4	—	—	—	—	SCSIRQ3	SCSIRQ2	SCSIRQ1	SCSIRQ0	
IDR2	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
ODR2	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
STR2	DBU27	DBU26	DBU25	DBU24	C/$\overline{D}$2	DBU22	IBF2	OBF2	
HISEL	SELSTR3	SELIRQ11	SELIRQ10	SELIRQ9	SELIRQ6	SELSMI	SELIRQ12	SELIRQ1	
HICR0	LPC3E	LPC2E	LPC1E	FGA20E	SDWNE	PMEE	LSMIE	LSCIE	
HICR1	LPCBSY	CLKREQ	IRQBSY	LRSTB	SDWNB	PMEB	LSMIB	LSCIB	
HICR2	GA20	LRST	SDWN	ABRT	IBFIE3	IBFIE2	IBFIE1	ERRIE	
HICR3	LFRAME	CLKRUN	SERIRQ	LRESET	LPCPD	PME	LSMI	LSCI	
WUEMR	WUEMR15	WUEMR14	WUEMR13	WUEMR12	WUEMR11	WUEMR10	WUEMR9	WUEMR8	INT
PGODR	PG7ODR	PG6ODR	PG5ODR	PG4ODR	PG3ODR	PG2ODR	PG1ODR	PG0ODR	PORT
PGPIN	PG7PIN	PG6PIN	PG5PIN	PG4PIN	PG3PIN	PG2PIN	PG1PIN	PG0PIN	
PGDDR	PG7DDR	PG6DDR	PG5DDR	PG4DDR	PG3DDR	PG2DDR	PG1DDR	PG0DDR	
PFODR	PF7ODR	PF6ODR	PF5ODR	PF4ODR	PF3ODR	PF2ODR	PF1ODR	PF0ODR	
PEPIN	—	—	—	PE4PIN	PE3PIN	PE2PIN	PE1PIN	PE0PIN	
PFPIN	PF7PIN	PF6PIN	PF5PIN	PF4PIN	PF3PIN	PF2PIN	PF1PIN	PF0PIN	
PFDDR	PF7DDR	PF6DDR	PF5DDR	PF4DDR	PF3DDR	PF2DDR	PF1DDR	PF0DDR	
PCODR	PC7ODR	PC6ODR	PC5ODR	PC4ODR	PC3ODR	PC2ODR	PC1ODR	PC0ODR	
PDODR	PD7ODR	PD6ODR	PD5ODR	PD4ODR	PD3ODR	PD2ODR	PD1ODR	PD0ODR	
PCPIN	PC7PIN	PC6PIN	PC5PIN	PC4PIN	PC3PIN	PC2PIN	PC1PIN	PC0PIN	
PCDDR	PC7DDR	PC6DDR	PC5DDR	PC4DDR	PC3DDR	PC2DDR	PC1DDR	PC0DDR	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
PDPIN	PD7PIN	PD6PIN	PD5PIN	PD4PIN	PD3PIN	PD2PIN	PD1PIN	PD0PIN	PORT
PDDDR	PD7DDR	PD6DDR	PD5DDR	PD4DDR	PD3DDR	PD2DDR	PD1DDR	PD0DDR	
TCR_0	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_0
TMDR_0	—	—	BFB	BFA	MD3	MD2	MD1	MD0	
TIORH_0	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIORL_0	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0	
TIER_0	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	
TSR_0	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
TCNT_0	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TGRA_0	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TGRB_0	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TGRC_0	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TGRD_0	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TCR_2	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_2
TMDR_2	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_2	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_2	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_2	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNT_2	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TGRA_2	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TGRB_2	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
SYSCR3	—	EIVS	RELOCATE	—	—	—	—	—	SYSTEM
MSTPCRA	MSTPA7	MSTPA6	MSTPA5	MSTPA4	MSTPA3	MSTPA2	MSTPA1	MSTPA0	
MSTPCRB	MSTPB7	MSTPB6	MSTPB5	MSTPB4	MSTPB3	MSTPB2	MSTPB1	MSTPB0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
KMIMR	KMIMR7	KMIMR6	KMIMR5	KMIMR4	KMIMR3	KMIMR2	KMIMR1	KMIMR0	INT
KMPCR	KM7PCR	KM6PCR	KM5PCR	KM4PCR	KM3PCR	KM2PCR	KM1PCR	KM0PCR	PORT
KMIMRA	KMIMR15	KMIMR14	KMIMR13	KMIMR12	KMIMR11	KMIMR10	KMIMR9	KMIMR8	INT
WUESCR	WUE15SC	WUE14SC	WUE13SC	WUE12SC	WUE11SC	WUE10SC	WUE9SC	WUE8SC	
WUESR	WUE15F	WUE14F	WUE13F	WUE12F	WUE11F	WUE10F	WUE9F	WUE8F	
WER	WUEE	—	—	—	—	—	—	—	
ICRD	ICRD7	ICRD6	ICRD5	ICRD4	ICRD3	ICRD2	ICRD1	ICRD0	
ICCR_2	ICE	IEIC	MST	TRS	ACKE	BBSY	IRIC	SCP	IIC_2
ICSR_2	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB	
ICRES_2	—	—	—	—	CLR3	CLR2	CLR1	CLR0	
ICXR_2	STOPIM	HNDS	ICDRF	ICDRE	ALIE	ALSL	FNC1	FNC0	
SARX_2	SVAX6	SVAX5	SVAX4	SVAX3	SVAX2	SVAX1	SVAX0	FSX	
ICDR_2	ICDR7	ICDR6	ICDR5	ICDR4	ICDR3	ICDR2	ICDR1	ICDR0	
SAR_2	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS	
ICMR_2	MLS	WAIT	CKS2	CKS1	CKS0	BC2	BC1	BC0	
DACR	—	PWME	—	—	OEB	OEA	OS	CKS	PWMX
DADRA	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6	
	DA5	DA4	DA3	DA2	DA1	DA0	CFS	—	
DADRB	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6	
	DA5	DA4	DA3	DA2	DA1	DA0	CFS	REGS	
DACNTH	DACNT7	DACNT6	DACNT5	DACNT4	DACNT3	DACNT2	DACNT1	DACNT0	
DACNTL	DACNT8	DACNT9	DACNT10	DACNT11	DACNT12	DACNT13	—	REGS	
FCCS	—	—	—	FLER	—	—	—	SCO	ROM
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	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
TSTR	—	—	—	—	—	CST2	CST1	CST0	TPU common
TSYR	—	—	—	—	—	SYNC2	SYNC1	SYNC0	
KBCR1_0	KBTS	PS	KCIE	KTIE	—	KCIF	KBTE	KTER	PS2
KBTR_0	KBT7	KBT6	KBT5	KBT4	KBT3	KBT2	KBT1	KBT0	
KBCR1_1	KBTS	PS	KCIE	KTIE	—	KCIF	KBTE	KTER	
KBTR_1	KBT7	KBT6	KBT5	KBT4	KBT3	KBT2	KBT1	KBT0	
KBCR1_2	KBTS	PS	KCIE	KTIE	—	KCIF	KBTE	KTER	
KBTR_2	KBT7	KBT6	KBT5	KBT4	KBT3	KBT2	KBT1	KBT0	
TCRXY	—	—	CKSX	CKSY	—	—	—	—	TMR_XY
TCR_Y	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	TMR_Y
TCSR_Y	CMFB	CMFA	OVF	ICIE	OS3	OS2	OS1	OS0	
TCORA_Y	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TCORB_Y	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TCNT_Y	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
ICDR_1	ICDR7	ICDR6	ICDR5	ICDR4	ICDR3	ICDR2	ICDR1	ICDR0	IIC_1
SARX_1	SVAX6	SVAX5	SVAX4	SVAX3	SVAX2	SVAX1	SVAX0	FSX	
ICMR_1	MLS	WAIT	CKS2	CKS1	CKS0	BC2	BC1	BC0	
SAR_1	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS	
ICCR_1	ICE	IEIC	MST	TRS	ACKE	BBSY	IRIC	SCP	
ICSR_1	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB	
KBCR1_3	KBTS	PS	KCIE	KTIE	—	KCIF	KBTE	KTER	PS2_3
KBTR_3	KBT7	KBT6	KBT5	KBT4	KBT3	KBT2	KBT1	KBT0	
ICXR_0	STOPIM	HNDS	ICDRF	ICDRE	ALIE	ALSL	FNC1	FNC0	IIC_0
ICXR_1	STOPIM	HNDS	ICDRF	ICDRE	ALIE	ALSL	FNC1	FNC0	IIC_1
KBCRH_0	KBIOE	KCLKI	KDI	KBFSEL	KBIE	KBF	PER	KBS	PS2_0
KBCRL_0	KBE	KCLKO	KDO	—	RXCR3	RXCR2	RXCR1	RXCR0	
KBBR_0	KB7	KB6	KB5	KB4	KB3	KB2	KB1	KB0	
KBCR2_0	—	—	—	—	TXCR3	TXCR2	TXCR1	TXCR0	
KBCRH_1	KBIOE	KCLKI	KDI	KBFSEL	KBIE	KBF	PER	KBS	PS2_1
KBCRL_1	KBE	KCLKO	KDO	—	RXCR3	RXCR2	RXCR1	RXCR0	
KBBR_1	KB7	KB6	KB5	KB4	KB3	KB2	KB1	KB0	
KBCR2_1	—	—	—	—	TXCR3	TXCR2	TXCR1	TXCR0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
KBCRH_2	KBIOE	KCLKI	KDI	KBFSEL	KBIE	KBF	PER	KBS	PS2_2
KBCRL_2	KBE	KCLKO	KDO	—	RXCR3	RXCR2	RXCR1	RXCR0	
KBBR_2	KB7	KB6	KB5	KB4	KB3	KB2	KB1	KB0	
KBCR2_2	—	—	—	—	TXCR3	TXCR2	TXCR1	TXCR0	
ICRES_0	—	—	—	—	CLR3	CLR2	CLR1	CLR0	IIC_0
ICRA	ICRA7	ICRA6	ICRA5	ICRA4	ICRA3	ICRA2	ICRA1	ICRA0	INT
ICRB	ICRB7	ICRB6	ICRB5	ICRB4	ICRB3	ICRB2	ICRB1	ICRB0	
ICRC	ICRC7	ICRC6	ICRC5	ICRC4	ICRC3	ICRC2	ICRC1	ICRC0	
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	
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	—	—	—	—	—	—	—	
PCSR	—	—	PWCKXB	PWCKXA	—	—	—	PWCKXC	PWMX
SBYCR	SSBY	STS2	STS1	STS0	—	SCK2	SCK1	SCK0	SYSTEM
LPWRCR	DTON	LSON	NESEL	EXCLE	—	—	—	—	
MSTPCRH	MSTP15	MSTP14	MSTP13	MSTP12	MSTP11	MSTP10	MSTP9	MSTP8	
MSTPCRL	MSTP7	MSTP6	MSTP5	MSTP4	MSTP3	MSTP2	MSTP1	MSTP0	
SMR_1*[1]	C/$\overline{A}$ (GM)	CHR (BLK)	PE (PE)	O/$\overline{E}$ (O/$\overline{E}$)	STOP (BCP1)	MP (BCP0)	CKS1 (CKS1)	CKS0 (CKS0)	SCI_1
BRR_1	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
SCR_1	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR_1	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
SSR_1*[1]	TDRE (TDRE)	RDRF (RDRF)	ORER (ORER)	FER (ERS)	PER (PER)	TEND (TEND)	MPB (MPB)	MPBT (MPBT)	SCI_1
RDR_1	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
SCMR_1	—	—	—	—	SDIR	SINV	—	SMIF	
SMR_2*[1]	C/$\overline{A}$ (GM)	CHR (BLK)	PE (PE)	O/$\overline{E}$ (O/$\overline{E}$)	STOP (BCP1)	MP (BCP0)	CKS1 (CKS1)	CKS0 (CKS0)	SCI_2
BRR_2	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
SCR_2	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR_2	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
SSR_2*[1]	TDRE (TDRE)	RDRF (RDRF)	ORER (ORER)	FER (ERS)	PER (PER)	TEND (TEND)	MPB (MPB)	MPBT (MPBT)	
RDR_2	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
SCMR_2	—	—	—	—	SDIR	SINV	—	SMIF	
TCSR_0	OVF	WT/$\overline{IT}$	TME	—	RST/$\overline{NMI}$	CKS2	CKS1	CKS0	WDT_0
TCNT_0	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
PAODR	PA7ODR	PA6ODR	PA5ODR	PA4ODR	PA3ODR	PA2ODR	PA1ODR	PA0ODR	PORT
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	P27PCR	P26PCR	P25PCR	P24PCR	P23PCR	P22PCR	P21PCR	P20PCR	
P3PCR	P37PCR	P36PCR	P35PCR	P34PCR	P33PCR	P32PCR	P31PCR	P30PCR	
P1DDR	P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DDR	
P2DDR	P27DDR	P26DDR	P25DDR	P24DDR	P23DDR	P22DDR	P21DDR	P20DDR	
P1DR	P17DR	P16DR	P15DR	P14DR	P13DR	P12DR	P11DR	P10DR	
P2DR	P27DR	P26DR	P25DR	P24DR	P23DR	P22DR	P21DR	P20DR	
P3DDR	P37DDR	P36DDR	P35DDR	P34DDR	P33DDR	P32DDR	P31DDR	P30DDR	
P4DDR	P47DDR	P46DDR	P45DDR	P44DDR	P43DDR	P42DDR	P41DDR	P40DDR	
P3DR	P37DR	P36DR	P35DR	P34DR	P33DR	P32DR	P31DR	P30DR	
P4DR	P47DR	P46DR	P45DR	P44DR	P43DR	P42DR	P41DR	P40DR	
P5DDR	—	—	—	—	—	P52DDR	P51DDR	P50DDR	
P6DDR	P67DDR	P66DDR	P65DDR	P64DDR	P63DDR	P62DDR	P61DDR	P60DDR	
P5DR	—	—	—	—	—	P52DR	P51DR	P50DR	
P6DR	P67DR	P66DR	P65DR	P64DR	P63DR	P62DR	P61DR	P60DR	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
PBODR	PB7ODR	PB6ODR	PB5ODR	PB4ODR	PB3ODR	PB2ODR	PB1ODR	PB0ODR	PORT
PBPIN	PB7PIN	PB6PIN	PB5PIN	PB4PIN	PB3PIN	PB2PIN	PB1PIN	PB0PIN	
P8DDR	—	P86DDR	P85DDR	P84DDR	P83DDR	P82DDR	P81DDR	P80DDR	
P7PIN	P77PIN	P76PIN	P75PIN	P74PIN	P73PIN	P72PIN	P71PIN	P70PIN	
PBDDR	PB7DDR	PB6DDR	PB5DDR	PB4DDR	PB3DDR	PB2DDR	PB1DDR	PB0DDR	
P8DR	—	P86DR	P85DR	P84DR	P83DR	P82DR	P81DR	P80DR	
P9DDR	P97DDR	P96DDR	P95DDR	P94DDR	P93DDR	P92DDR	P91DDR	P90DDR	
P9DR	P97DR	P96DR	P95DR	P94DR	P93DR	P92DR	P91DR	P90DR	
IER	IRQ7E	IRQ6E	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IRQ0E	INT
STCR	IICX2	IICX1	IICX0	IICE	FLSHE	IICS	ICKS1	ICKS0	SYSTEM
SYSCR	—	—	INTM1	INTM0	XRST	NMIEG	KINWUE	RAME	
MDCR	EXPE	—	—	—	—	MDS2	MDS1	MDS0	
BCR	—	ICIS0	BRSTRM	BRSTS1	BRSTS0	—	IOS1	IOS0	BSC
WSCR	—	—	ABW	AST	WMS1	WMS0	WC1	WC0	
TCR_0	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	TMR_0, TMR_1
TCR_1	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	
TCSR_0	CMFB	CMFA	OVF	ADTE	OS3	OS2	OS1	OS0	
TCSR_1	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0	
TCORA_0	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TCORA_1	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TCORB_0	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TCORB_1	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TCNT_0	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TCNT_1	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
ICCR_0	ICE	IEIC	MST	TRS	ACKE	BBSY	IRIC	SCP	IIC_0
ICSR_0	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB	
ICDR_0	ICDR7	ICDR6	ICDR5	ICDR4	ICDR3	ICDR2	ICDR1	ICDR0	
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	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
KBCRH_3	KBIOE	KCLKI	KDI	KBFSEL	KBIE	KBF	PER	KBS	PS2_3
KBCRL_3	KBE	KCLKO	KDO	—	RXCR3	RXCR2	RXCR1	RXCR0	
KBBR_3	KB7	KB6	KB5	KB4	KB3	KB2	KB1	KB0	
KBCR2_3	—	—	—	—	TXCR3	TXCR2	TXCR1	TXCR0	
TCSR_1	OVF	WT/$\overline{IT}$	TME	PSS	RST/$\overline{NMI}$	CKS2	CKS1	CKS0	WDT_1
TCNT_1	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TCR_X	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	TMR_X
TCSR_X	CMFB	CMFA	OVF	ICF	OS3	OS2	OS1	OS0	
TICRR	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TICRF	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TCNT_X	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TCORC	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TCORA_X	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TCORB_X	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
TCONRI	—	—	—	ICST	—	—	—	—	
TCONRS	TMRX/Y	—	—	—	—	—	—	—	TMR_X, TMR_Y

Notes: 1. In normal mode and smart card interface mode, bit names differ in part.
( ) : Bit name in smart card interface mode.
2. When TWRE = 1 or SELSTR3 = 0.
3. When TWRE = 0 and SELSTR3 = 1.

27.3 Register States in Each Operating Mode

Register Abbreviation	Reset	High-Speed/Medium speed	Watch	Sleep	Module Stop	Software Standby	Module
PIDDR	Initialized	—	—	—	—	—	PORT
PJDDR	Initialized	—	—	—	—	—	
PIODR	Initialized	—	—	—	—	—	
PJODR	Initialized	—	—	—	—	—	
PIPIN	—	—	—	—	—	—	
PJPIN	—	—	—	—	—	—	
PJPCR	Initialized	—	—	—	—	—	
PINOCR	Initialized	—	—	—	—	—	
PJNOCR	Initialized	—	—	—	—	—	
CCR1	Initialized	—	—	—	—	—	CIR
CCR2	Initialized	—	—	—	—	—	
CSTR	Initialized	—	—	—	—	—	
CEIR	Initialized	—	—	—	—	—	
BRR	Initialized	—	—	—	—	—	
CIRRDR0 to 7	Initialized	—	—	—	—	—	
HHMIN	Initialized	—	—	—	—	—	
HHMAX	Initialized	—	—	—	—	—	
HLMIN	Initialized	—	—	—	—	—	
HLMAX	Initialized	—	—	—	—	—	
DT1MIN	Initialized	—	—	—	—	—	
DT1MAX	Initialized	—	—	—	—	—	
DT0MIN	Initialized	—	—	—	—	—	
DT0MAX	Initialized	—	—	—	—	—	
RMIN	Initialized	—	—	—	—	—	
RMAX	Initialized	—	—	—	—	—	

Register Abbreviation	Reset	High-Speed/Medium speed	Watch	Sleep	Module Stop	Software Standby	Module
TDPCNT_0	Initialized	—	—	—	—	—	TDP_0
TDPPDMX_0	Initialized	—	—	—	—	—	
TDPPDMN_0	Initialized	—	—	—	—	—	
TDPWDMX_0	Initialized	—	—	—	—	—	
TDPICR_0	Initialized	—	—	—	—	—	
TDPICRF_0	Initialized	—	—	—	—	—	
TDPCSR_0	Initialized	—	—	—	—	—	
TDPCR1_0	Initialized	—	—	—	—	—	
TDPIER_0	Initialized	—	—	—	—	—	
TDPCR2_0	Initialized	—	—	—	—	—	
TDPWDMN_0	Initialized	—	—	—	—	—	
TDPCNT_1	Initialized	—	—	—	—	—	TDP_1
TDPPDMX_1	Initialized	—	—	—	—	—	
TDPPDMN_1	Initialized	—	—	—	—	—	
TDPWDMX_1	Initialized	—	—	—	—	—	
TDPICR_1	Initialized	—	—	—	—	—	
TDPICRF_1	Initialized	—	—	—	—	—	
TDPCSR_1	Initialized	—	—	—	—	—	
TDPCR1_1	Initialized	—	—	—	—	—	
TDPIER_1	Initialized	—	—	—	—	—	
TDPCR2_1	Initialized	—	—	—	—	—	
TDPWDMN_1	Initialized	—	—	—	—	—	
TDPCNT_2	Initialized	—	—	—	—	—	TDP_2
TDPPDMX_2	Initialized	—	—	—	—	—	
TDPPDMN_1	Initialized	—	—	—	—	—	
TDPWDMX_2	Initialized	—	—	—	—	—	
TDPICR_2	Initialized	—	—	—	—	—	
TDPICRF_2	Initialized	—	—	—	—	—	
TDPCSR_2	Initialized	—	—	—	—	—	

Register Abbreviation	Reset	High-Speed/Medium speed	Watch	Sleep	Module Stop	Software Standby	Module
TDPCR1_2	Initialized	—	—	—	—	—	TDP_2
TDPIER_2	Initialized	—	—	—	—	—	
TDPCR2_2	Initialized	—	—	—	—	—	
TDPWDMN_2	Initialized	—	—	—	—	—	
TCMCNT_0	Initialized	—	—	—	—	—	TCM_0
TCMMLCM_0	Initialized	—	—	—	—	—	
TCMICR_0	Initialized	—	—	—	—	—	
TCMICRF_0	Initialized	—	—	—	—	—	
TCMCSR_0	Initialized	—	—	—	—	—	
TCMCR_0	Initialized	—	—	—	—	—	
TCMIER_0	Initialized	—	—	—	—	—	
TCMMINCM_0	Initialized	—	—	—	—	—	
TCMCNT_1	Initialized	—	—	—	—	—	TCM_1
TCMMLCM_1	Initialized	—	—	—	—	—	
TCMICR_1	Initialized	—	—	—	—	—	
TCMICRF_1	Initialized	—	—	—	—	—	
TCMCSR_1	Initialized	—	—	—	—	—	
TCMCR_1	Initialized	—	—	—	—	—	
TCMIER_1	Initialized	—	—	—	—	—	
TCMMINCM_1	Initialized	—	—	—	—	—	
TCMCNT_2	Initialized	—	—	—	—	—	TCM_2
TCMMLCM_2	Initialized	—	—	—	—	—	
TCMICR_2	Initialized	—	—	—	—	—	
TCMICRF_2	Initialized	—	—	—	—	—	
TCMCSR_2	Initialized	—	—	—	—	—	
TCMCR_2	Initialized	—	—	—	—	—	
TCMIER_2	Initialized	—	—	—	—	—	
TCMMINCM_2	Initialized	—	—	—	—	—	

Register Abbreviation	Reset	High-Speed/Medium speed	Watch	Sleep	Module Stop	Software Standby	Module
TCMCNT_3	Initialized	—	—	—	—	—	TCM_3
TCMMLCM_3	Initialized	—	—	—	—	—	
TCMICR_3	Initialized	—	—	—	—	—	
TCMICRF_3	Initialized	—	—	—	—	—	
TCMCSR_3	Initialized	—	—	—	—	—	
TCMCR_3	Initialized	—	—	—	—	—	
TCMIER_3	Initialized	—	—	—	—	—	
TCMMINCM_3	Initialized	—	—	—	—	—	
ADDRA	Initialized	—	Initialized	—	Initialized	Initialized	A/D converter
ADDRB	Initialized	—	Initialized	—	Initialized	Initialized	
ADDRC	Initialized	—	Initialized	—	Initialized	Initialized	
ADDRD	Initialized	—	Initialized	—	Initialized	Initialized	
ADDRE	Initialized	—	Initialized	—	Initialized	Initialized	
ADDRF	Initialized	—	Initialized	—	Initialized	Initialized	
ADDRG	Initialized	—	Initialized	—	Initialized	Initialized	
ADDRH	Initialized	—	Initialized	—	Initialized	Initialized	
ADCSR	Initialized	—	Initialized	—	Initialized	Initialized	
ADCR	Initialized	—	Initialized	—	Initialized	Initialized	
FRBR	Initialized	—	—	—	—	—	SCIF
FTHR	—	—	—	—	—	—	
FDLL	Initialized	—	—	—	—	—	
FIER	Initialized	—	—	—	—	—	
FDLH	Initialized	—	—	—	—	—	
FIIR	Initialized	—	—	—	—	—	
FFCR	Initialized	—	—	—	—	—	
FLCR	Initialized	—	—	—	—	—	
FMCR	Initialized	—	—	—	—	—	
FLSR	Initialized	—	—	—	—	—	
FMSR	—	—	—	—	—	—	
FSCR	Initialized	—	—	—	—	—	
SCIFCR	Initialized	—	—	—	—	—	

Register Abbreviation	Reset	High-Speed/Medium speed	Watch	Sleep	Module Stop	Software Standby	Module
FSIHBARH	Initialized	—	—	—	—	—	FSI
FSIHBARL	Initialized	—	—	—	—	—	
FSISR	Initialized	—	—	—	—	—	
CMDHRARH	Initialized	—	—	—	—	—	
CMDHRARL	Initialized	—	—	—	—	—	
FSICMDR	Initialized	—	—	—	—	—	
FSILSTR1	Initialized	—	—	—	—	—	
FSILSTR2	Initialized	—	—	—	—	—	
FSIGPR1	Initialized	—	—	—	—	—	
FSIGPR2	Initialized	—	—	—	—	—	
FSIGPR3	Initialized	—	—	—	—	—	
FSIGPR4	Initialized	—	—	—	—	—	
FSIGPR5	Initialized	—	—	—	—	—	
FSIGPR6	Initialized	—	—	—	—	—	
FSIGPR7	Initialized	—	—	—	—	—	
FSIGPR8	Initialized	—	—	—	—	—	
FSIGPR9	Initialized	—	—	—	—	—	
FSIGPRA	Initialized	—	—	—	—	—	
FSIGPRB	Initialized	—	—	—	—	—	
FSIGPRC	Initialized	—	—	—	—	—	
FSIGPRD	Initialized	—	—	—	—	—	
FSIGPRE	Initialized	—	—	—	—	—	
FSIGPRF	Initialized	—	—	—	—	—	
SLCR	Initialized	—	—	—	—	—	
FSIARH	Initialized	—	—	—	—	—	
FSIARM	Initialized	—	—	—	—	—	
FSIARL	Initialized	—	—	—	—	—	
FSIWDRHH	Initialized	—	—	—	—	—	
FSIWDRHL	Initialized	—	—	—	—	—	

Register Abbreviation	Reset	High-Speed/Medium speed	Watch	Sleep	Module Stop	Software Standby	Module
FSIWDRLH	Initialized	—	—	—	—	—	FSI
FSIWDRLL	Initialized	—	—	—	—	—	
FSICR1	Initialized	—	—	—	—	—	
FSICR2	Initialized	—	—	—	—	—	
FSIBNR	Initialized	—	—	—	—	—	
FSINS	Initialized	—	—	—	—	—	
FSIRDINS	Initialized	—	—	—	—	—	
FSIPPINS	Initialized	—	—	—	—	—	
FSISTR	Initialized	—	—	—	—	—	
FSITDR0	Initialized	—	—	—	—	—	
FSITDR1	Initialized	—	—	—	—	—	
FSITDR2	Initialized	—	—	—	—	—	
FSITDR3	Initialized	—	—	—	—	—	
FSITDR4	Initialized	—	—	—	—	—	
FSITDR5	Initialized	—	—	—	—	—	
FSITDR6	Initialized	—	—	—	—	—	
FSITDR7	Initialized	—	—	—	—	—	
FSIRDR	Initialized	—	—	—	—	—	
PWMREG0_A	Initialized	—	Initialized	—	Initialized	Initialized	PWMU_A
PWMPRE0_A	Initialized	—	Initialized	—	Initialized	Initialized	
PWMREG1_A	Initialized	—	Initialized	—	Initialized	Initialized	
PWMPRE1_A	Initialized	—	Initialized	—	Initialized	Initialized	
PWMREG2_A	Initialized	—	Initialized	—	Initialized	Initialized	
PWMPRE2_A	Initialized	—	Initialized	—	Initialized	Initialized	
PWMREG3_A	Initialized	—	Initialized	—	Initialized	Initialized	
PWMPRE3_A	Initialized	—	Initialized	—	Initialized	Initialized	
PWMREG4_A	Initialized	—	Initialized	—	Initialized	Initialized	
PWMPRE4_A	Initialized	—	Initialized	—	Initialized	Initialized	
PWMREG5_A	Initialized	—	Initialized	—	Initialized	Initialized	
PWMPRE5_A	Initialized	—	Initialized	—	Initialized	Initialized	

Register Abbreviation	Reset	High-Speed/Medium speed	Watch	Sleep	Module Stop	Software Standby	Module
PWMCONA_A	Initialized	—	Initialized	—	Initialized	Initialized	PWMU_A
PWMCONB_A	Initialized	—	Initialized	—	Initialized	Initialized	
PWMCONC_A	Initialized	—	Initialized	—	Initialized	Initialized	
PWMCOND_A	Initialized	—	Initialized	—	Initialized	Initialized	
PWMREG0_B	Initialized	—	Initialized	—	Initialized	Initialized	PWMU_B
PWMPRE0_B	Initialized	—	Initialized	—	Initialized	Initialized	
PWMREG1_B	Initialized	—	Initialized	—	Initialized	Initialized	
PWMPRE1_B	Initialized	—	Initialized	—	Initialized	Initialized	
PWMREG2_B	Initialized	—	Initialized	—	Initialized	Initialized	
PWMPRE2_B	Initialized	—	Initialized	—	Initialized	Initialized	
PWMREG3_B	Initialized	—	Initialized	—	Initialized	Initialized	
PWMPRE3_B	Initialized	—	Initialized	—	Initialized	Initialized	
PWMREG4_B	Initialized	—	Initialized	—	Initialized	Initialized	
PWMPRE4_B	Initialized	—	Initialized	—	Initialized	Initialized	
PWMREG5_B	Initialized	—	Initialized	—	Initialized	Initialized	
PWMPRE5_B	Initialized	—	Initialized	—	Initialized	Initialized	
PWMCONA_B	Initialized	—	Initialized	—	Initialized	Initialized	
PWMCONB_B	Initialized	—	Initialized	—	Initialized	Initialized	
PWMCONC_B	Initialized	—	Initialized	—	Initialized	Initialized	
PWMCOND_B	Initialized	—	Initialized	—	Initialized	Initialized	
TCR_1	Initialized	—	—	—	—	—	TPU_1
TMDR_1	Initialized	—	—	—	—	—	
TIOR_1	Initialized	—	—	—	—	—	
TIER_1	Initialized	—	—	—	—	—	
TSR_1	Initialized	—	—	—	—	—	
TCNT_1	Initialized	—	—	—	—	—	
TGRA_1	Initialized	—	—	—	—	—	
TGRB_1	Initialized	—	—	—	—	—	

Register Abbreviation	Reset	High-Speed/Medium speed	Watch	Sleep	Module Stop	Software Standby	Module
LADR1H	Initialized	—	—	—	—	—	LPC
LADR1L	Initialized	—	—	—	—	—	
LADR2H	Initialized	—	—	—	—	—	
LADR2L	Initialized	—	—	—	—	—	
SCIFADRH	Initialized	—	—	—	—	—	
SCIFADRL	Initialized	—	—	—	—	—	
LADR4H	Initialized	—	—	—	—	—	
LADR4L	Initialized	—	—	—	—	—	
IDR4	Initialized	—	—	—	—	—	
ODR4	Initialized	—	—	—	—	—	
STR4	Initialized	—	—	—	—	—	
HICR4	Initialized	—	—	—	—	—	
SIRQCR2	Initialized	—	—	—	—	—	
SIRQCR3	Initialized	—	—	—	—	—	
P6NCE	Initialized	—	—	—	—	—	PORT
P6NCMC	Initialized	—	—	—	—	—	
P6NCCS	Initialized	—	—	—	—	—	
PCNCE	Initialized	—	—	—	—	—	
PCNCMC	Initialized	—	—	—	—	—	
PCNCCS	Initialized	—	—	—	—	—	
PGNCE	Initialized	—	—	—	—	—	
PGNCMC	Initialized	—	—	—	—	—	
PGNCCS	Initialized	—	—	—	—	—	
PHPIN	Initialized	—	—	—	—	—	
PHDDR	Initialized	—	—	—	—	—	
PHODR	Initialized	—	—	—	—	—	
PHNOCR	Initialized	—	—	—	—	—	
PTCNT0	Initialized	—	—	—	—	—	
PTCNT1	Initialized	—	—	—	—	—	
PTCNT2	Initialized	—	—	—	—	—	

Register Abbreviation	Reset	High-Speed/Medium speed	Watch	Sleep	Module Stop	Software Standby	Module
P9PCR	Initialized	—	—	—	—	—	PORT
PGNOCR	Initialized	—	—	—	—	—	
PFNOCR	Initialized	—	—	—	—	—	
PCNOCR	Initialized	—	—	—	—	—	
PDNOCR	Initialized	—	—	—	—	—	
TWR0MW	Initialized	—	—	—	—	—	LPC
TWR0SW	Initialized	—	—	—	—	—	
TWR1	Initialized	—	—	—	—	—	
TWR2	Initialized	—	—	—	—	—	
TWR3	Initialized	—	—	—	—	—	
TWR4	Initialized	—	—	—	—	—	
TWR5	Initialized	—	—	—	—	—	
TWR6	Initialized	—	—	—	—	—	
TWR7	Initialized	—	—	—	—	—	
TWR8	Initialized	—	—	—	—	—	
TWR9	Initialized	—	—	—	—	—	
TWR10	Initialized	—	—	—	—	—	
TWR11	Initialized	—	—	—	—	—	
TWR12	Initialized	—	—	—	—	—	
TWR13	Initialized	—	—	—	—	—	
TWR14	Initialized	—	—	—	—	—	
TWR15	Initialized	—	—	—	—	—	
IDR3	Initialized	—	—	—	—	—	
ODR3	Initialized	—	—	—	—	—	
STR3	Initialized	—	—	—	—	—	
HICR5	Initialized	—	—	—	—	—	
LADR3H	Initialized	—	—	—	—	—	
LADR3L	Initialized	—	—	—	—	—	
SIRQCR0	Initialized	—	—	—	—	—	
SIRQCR1	Initialized	—	—	—	—	—	

Register Abbreviation	Reset	High-Speed/Medium speed	Watch	Sleep	Module Stop	Software Standby	Module
IDR1	Initialized	—	—	—	—	—	LPC
ODR1	Initialized	—	—	—	—	—	
STR1	Initialized	—	—	—	—	—	
SIRQCR4	Initialized	—	—	—	—	—	
IDR2	Initialized	—	—	—	—	—	
ODR2	Initialized	—	—	—	—	—	
STR2	Initialized	—	—	—	—	—	
HISEL	Initialized	—	—	—	—	—	
HICR0	Initialized	—	—	—	—	—	
HICR1	Initialized	—	—	—	—	—	
HICR2	—	—	—	—	—	—	
HICR3	—	—	—	—	—	—	
WUEMR	Initialized	—	—	—	—	—	INT
PGODR	Initialized	—	—	—	—	—	PORT
PGPIN	—	—	—	—	—	—	
PGDDR	Initialized	—	—	—	—	—	
PFODR	Initialized	—	—	—	—	—	
PEPIN	—	—	—	—	—	—	
PFPIN	—	—	—	—	—	—	
PFDDR	Initialized	—	—	—	—	—	
PCODR	Initialized	—	—	—	—	—	
PDODR	Initialized	—	—	—	—	—	
PCPIN	—	—	—	—	—	—	
PCDDR	Initialized	—	—	—	—	—	
PDPIN	—	—	—	—	—	—	
PDDDR	Initialized	—	—	—	—	—	
TCR_0	Initialized	—	—	—	—	—	TPU_0
TMDR_0	Initialized	—	—	—	—	—	
TIORH_0	Initialized	—	—	—	—	—	
TIORL_0	Initialized	—	—	—	—	—	
TIER_0	Initialized	—	—	—	—	—	
TSR_0	Initialized	—	—	—	—	—	
TCNT_0	Initialized	—	—	—	—	—	

Register Abbreviation	Reset	High-Speed/Medium speed	Watch	Sleep	Module Stop	Software Standby	Module
TGRA_0	Initialized	—	—	—	—	—	TPU_0
TGRB_0	Initialized	—	—	—	—	—	
TGRC_0	Initialized	—	—	—	—	—	
TGRD_0	Initialized	—	—	—	—	—	
TCR_2	Initialized	—	—	—	—	—	TPU_2
TMDR_2	Initialized	—	—	—	—	—	
TIOR_2	Initialized	—	—	—	—	—	
TIER_2	Initialized	—	—	—	—	—	
TSR_2	Initialized	—	—	—	—	—	
TCNT_2	Initialized	—	—	—	—	—	
TGRA_2	Initialized	—	—	—	—	—	
TGRB_2	Initialized	—	—	—	—	—	
SYSCR3	Initialized	—	—	—	—	—	SYSTEM
MSTPCRA	Initialized	—	—	—	—	—	
MSTPCRB	Initialized	—	—	—	—	—	
KMIMR	Initialized	—	—	—	—	—	INT
KMPCR	Initialized	—	—	—	—	—	PORT
KMIMRA	Initialized	—	—	—	—	—	INT
WUESCR	Initialized	—	—	—	—	—	
WUESR	Initialized	—	—	—	—	—	
WER	Initialized	—	—	—	—	—	
ICRD	Initialized	—	—	—	—	—	
ICCR_2	Initialized	—	—	—	—	—	IIC_2
ICSR_2	Initialized	—	—	—	—	—	
ICRES_2	Initialized	—	—	—	—	—	
ICXR_2	Initialized	—	—	—	—	—	
ICDR_2	—	—	—	—	—	—	
SARX_2	Initialized	—	—	—	—	—	
ICMR_2	Initialized	—	—	—	—	—	
SAR_2	Initialized	—	—	—	—	—	

Register Abbreviation	Reset	High-Speed/Medium speed	Watch	Sleep	Module Stop	Software Standby	Module
DACR	Initialized	—	Initialized	—	Initialized	Initialized	PWMX
DADRA	Initialized	—	Initialized	—	Initialized	Initialized	
DADRB	Initialized	—	Initialized	—	Initialized	Initialized	
DACNT	Initialized	—	Initialized	—	Initialized	Initialized	
FCCS	Initialized	—	—	—	—	—	ROM
FPCS	Initialized	—	—	—	—	—	
FECS	Initialized	—	—	—	—	—	
FKEY	Initialized	—	—	—	—	—	
FMATS	Initialized	—	—	—	—	—	
FTDAR	Initialized	—	—	—	—	—	
TSTR	Initialized	—	—	—	—	—	TPU common
TSYR	Initialized	—	—	—	—	—	
KBCR1_0	Initialized	—	—	—	—	—	PS2
KBTR_0	Initialized	—	—	—	—	—	
KBCR1_1	Initialized	—	—	—	—	—	
KBTR_1	Initialized	—	—	—	—	—	
KBCR1_2	Initialized	—	—	—	—	—	
KBTR_2	Initialized	—	—	—	—	—	
TCRXY	Initialized	—	—	—	—	—	TMR_XY
TCR_Y	Initialized	—	—	—	—	—	TMR_Y
TCSR_Y	Initialized	—	—	—	—	—	
TCORA_Y	Initialized	—	—	—	—	—	
TCORB_Y	Initialized	—	—	—	—	—	
TCNT_Y	Initialized	—	—	—	—	—	
ICDR_1	—	—	—	—	—	—	IIC_1
SARX_1	Initialized	—	—	—	—	—	
ICMR_1	Initialized	—	—	—	—	—	
SAR_1	Initialized	—	—	—	—	—	
ICCR_1	Initialized	—	—	—	—	—	
ICSR_1	Initialized	—	—	—	—	—	

Register Abbreviation	Reset	High-Speed/Medium speed	Watch	Sleep	Module Stop	Software Standby	Module
KBCR1_3	Initialized	—	—	—	—	—	PS2_3
KBTR_3	Initialized	—	—	—	—	—	
ICXR_0	Initialized	—	—	—	—	—	IIC_0
ICXR_1	Initialized	—	—	—	—	—	IIC_1
KBCRH_0	Initialized	—	—	—	—	—	PS2_0
KBCRL_0	Initialized	—	—	—	—	—	
KBBR_0	Initialized	—	—	—	—	—	
KBCR2_0	Initialized	—	—	—	—	—	
KBCRH_1	Initialized	—	—	—	—	—	PS2_1
KBCRL_1	Initialized	—	—	—	—	—	
KBBR_1	Initialized	—	—	—	—	—	
KBCR2_1	Initialized	—	—	—	—	—	
KBCRH_2	Initialized	—	—	—	—	—	PS2_2
KBCRL_2	Initialized	—	—	—	—	—	
KBBR_2	Initialized	—	—	—	—	—	
KBCR2_2	Initialized	—	—	—	—	—	
ICRES_0	Initialized	—	—	—	—	—	IIC_0
ICRA	Initialized	—	—	—	—	—	INT
ICRB	Initialized	—	—	—	—	—	
ICRC	Initialized	—	—	—	—	—	
ISR	Initialized	—	—	—	—	—	
ISCRH	Initialized	—	—	—	—	—	
ISCRL	Initialized	—	—	—	—	—	
ABRKCR	Initialized	—	—	—	—	—	
BARA	Initialized	—	—	—	—	—	
BARB	Initialized	—	—	—	—	—	
BARC	Initialized	—	—	—	—	—	
IER16	Initialized	—	—	—	—	—	
ISR16	Initialized	—	—	—	—	—	
ISCR16H	Initialized	—	—	—	—	—	

Register Abbreviation	Reset	High-Speed/Medium speed	Watch	Sleep	Module Stop	Software Standby	Module
ISCR16L	Initialized	—	—	—	—	—	INT
ISSR16	Initialized	—	—	—	—	—	
ISSR	Initialized	—	—	—	—	—	
PCSR	Initialized	—	—	—	—	—	PWMX
SBYCR	Initialized	—	—	—	—	—	SYSTEM
LPWRCR	Initialized	—	—	—	—	—	
MSTPCRH	Initialized	—	—	—	—	—	
MSTPCRL	Initialized	—	—	—	—	—	
SMR_1	Initialized	—	—	—	—	—	SCI_1
BRR_1	Initialized	—	—	—	—	—	
SCR_1	Initialized	—	—	—	—	—	
TDR_1	Initialized	—	Initialized	—	Initialized	Initialized	
SSR_1	Initialized	—	Initialized	—	Initialized	Initialized	
RDR_1	Initialized	—	Initialized	—	Initialized	Initialized	
SCMR_1	Initialized	—	—	—	—	—	
SMR_2	Initialized	—	—	—	—	—	SCI_2
BRR_2	Initialized	—	—	—	—	—	
SCR_2	Initialized	—	—	—	—	—	
TDR_2	Initialized	—	Initialized	—	Initialized	Initialized	
SSR_2	Initialized	—	Initialized	—	Initialized	Initialized	
RDR_2	Initialized	—	Initialized	—	Initialized	Initialized	
SCMR_2	Initialized	—	—	—	—	—	
TCSR_0	Initialized	—	—	—	—	—	WDT_0
TCNT_0	Initialized	—	—	—	—	—	
PAODR	Initialized	—	—	—	—	—	PORT
PAPIN	—	—	—	—	—	—	
PADDR	Initialized	—	—	—	—	—	
P1PCR	Initialized	—	—	—	—	—	
P2PCR	Initialized	—	—	—	—	—	
P3PCR	Initialized	—	—	—	—	—	

Register Abbreviation	Reset	High-Speed/Medium speed	Watch	Sleep	Module Stop	Software Standby	Module
P1DDR	Initialized	—	—	—	—	—	PORT
P2DDR	Initialized	—	—	—	—	—	
P1DR	Initialized	—	—	—	—	—	
P2DR	Initialized	—	—	—	—	—	
P3DDR	Initialized	—	—	—	—	—	
P4DDR	Initialized	—	—	—	—	—	
P3DR	Initialized	—	—	—	—	—	
P4DR	Initialized	—	—	—	—	—	
P5DDR	Initialized	—	—	—	—	—	
P6DDR	Initialized	—	—	—	—	—	
P5DR	Initialized	—	—	—	—	—	
P6DR	Initialized	—	—	—	—	—	
PBODR	Initialized	—	—	—	—	—	
PBPIN	—	—	—	—	—	—	
P8DDR	Initialized	—	—	—	—	—	
P7PIN	—	—	—	—	—	—	
PBDDR	Initialized	—	—	—	—	—	
P8DR	Initialized	—	—	—	—	—	
P9DDR	Initialized	—	—	—	—	—	
P9DR	Initialized	—	—	—	—	—	
IER	Initialized	—	—	—	—	—	INT
STCR	Initialized	—	—	—	—	—	SYSTEM
SYSCR	Initialized	—	—	—	—	—	
MDCR	Initialized	—	—	—	—	—	
BCR	Initialized	—	—	—	—	—	BSC
WSCR	Initialized	—	—	—	—	—	
TCR_0	Initialized	—	—	—	—	—	TMR_0, TMR_1
TCR_1	Initialized	—	—	—	—	—	
TCSR_0	Initialized	—	—	—	—	—	
TCSR_1	Initialized	—	—	—	—	—	

Register Abbreviation	Reset	High-Speed/Medium speed	Watch	Sleep	Module Stop	Software Standby	Module
TCORA_0	Initialized	—	—	—	—	—	TMR_0, TMR_1
TCORA_1	Initialized	—	—	—	—	—	
TCORB_0	Initialized	—	—	—	—	—	
TCORB_1	Initialized	—	—	—	—	—	
TCNT_0	Initialized	—	—	—	—	—	
TCNT_1	Initialized	—	—	—	—	—	
ICCR_0	Initialized	—	—	—	—	—	IIC_0
ICSR_0	Initialized	—	—	—	—	—	
ICDR_0	—	—	—	—	—	—	
SARX_0	Initialized	—	—	—	—	—	
ICMR_0	Initialized	—	—	—	—	—	
SAR_0	Initialized	—	—	—	—	—	
KBCRH_3	Initialized	—	—	—	—	—	PS2_3
KBCRL_3	Initialized	—	—	—	—	—	
KBBR_3	Initialized	—	—	—	—	—	
KBCR2_3	Initialized	—	—	—	—	—	
TCSR_1	Initialized	—	—	—	—	—	WDT_1
TCNT_1	Initialized	—	—	—	—	—	
TCR_X	Initialized	—	—	—	—	—	TMR_X
TCSR_X	Initialized	—	—	—	—	—	
TICRR	Initialized	—	—	—	—	—	
TICRF	Initialized	—	—	—	—	—	
TCNT_X	Initialized	—	—	—	—	—	
TCORC	Initialized	—	—	—	—	—	
TCORA_X	Initialized	—	—	—	—	—	
TCORB_X	Initialized	—	—	—	—	—	
TCONRI	Initialized	—	—	—	—	—	
TCONRS	Initialized	—	—	—	—	—	TMR_X, TMR_Y

27.4 Register Selection Condition

Lower Address	Register Abbreviation	Register Selection Condition	Module
H'F900	P1DDR	PORTS = 1	PORT
H'F901	P2DDR		
H'F902	P1DR		
H'F903	P2DR		
H'F904	P1PIN (Read)		
H'F905	P2PIN (Read)		
H'F907	P2PCR		
H'F910	P3DDR		
H'F911	P4DDR		
H'F912	P3DR		
H'F913	P4DR		
H'F914	P3PIN (Read)		
H'F915	P4PIN (Read)		
H'F916	P3PCR		
H'F920	P5DDR		
H'F921	P6DDR		
H'F922	P5DR		
H'F923	P6DR		
H'F924	P5PIN (Read)		
H'F925	P6PIN (Read)		
H'F92B	P6NCE		
H'F92D	P6NCMC		
H'F92F	P6NCCS		
H'F931	P8DDR		
H'F933	P8DR		
H'F934	P7PIN (Read)		
H'F935	P8PIN (Read)		
H'F940	P9DDR		
H'F942	P9DR		

Lower Address	Register Abbreviation	Register Selection Condition	Module
H'F944	P9PIN (Read)	PORTS = 1	PORT
H'F946	P9PCR		
H'F950	PADDR		
H'F951	PBDDR		
H'F952	PAODR		
H'F953	PBODR		
H'F954	PAPIN (Read)		
H'F955	PBPIN (Read)		
H'F957	PBPCR		
H'F960	PCDDR		
H'F961	PDDDR		
H'F962	PCODR		
H'F963	PDODR		
H'F964	PCPIN (Read)		
H'F965	PDPIN (Read)		
H'F966	PCPCR		
H'F967	PDPCR		
H'F968	PCNOCR		
H'F969	PDNOCR		
H'F96A	PCNCE		
H'F96C	PCNCMC		
H'F96E	PCNCCS		
H'F971	PFDDR		
H'F973	PFODR		
H'F974	PEPIN (Read)		
H'F975	PFPIN (Read)		
H'F977	PFPCR		
H'F979	PFNOCR		
H'F980	PGDDR		
H'F981	PHDDR		
H'F982	PGODR		

Lower Address	Register Abbreviation	Register Selection Condition	Module
H'F983	PHODR	PORTS = 1	PORT
H'F984	PGPIN (Read)		
H'F985	PHPIN (Read)		
H'F987	PHPCR		
H'F988	PGNOCR		
H'F989	PHNOCR		
H'F98A	PGNCE		
H'F98C	PGNCMC		
H'F98E	PGNCCS		
H'F990	PIDDR	No condition	
H'F991	PJDDR		
H'F992	PIODR		
H'F993	PJODR		
H'F994	PIPIN (Read)		
H'F995	PJPIN (Read)		
H'F996	PIPCR		
H'F997	PJPCR		
H'F998	PINOCR		
H'F999	PJNOCR		
H'FA40	CCR1	MSTPA3 = 0	CIR
H'FA41	CCR2		
H'FA42	CSTR		
H'FA43	CEIR		
H'FA44	BRR		
H'FA45	CIRRDR0 to 7		
H'FA46	HHMIN		
H'FA48	HHMAX		
H'FA4A	HLMIN		
H'FA4B	HLMAX		
H'FA4C	DT0MIN		
H'FA4D	DT0MAX		
H'FA4E	DT1MIN		

Lower Address	Register Abbreviation	Register Selection Condition	Module
H'FA4F	DT1MAX	MSTPA3 = 0	CIR
H'FA50	RMIN		
H'FA51	RMAX		
H'FB40	TDPCNT_0	MSTPA6 = 0	TDP_0
H'FB42	TDPWDMX_0		
H'FB44	TDPWDMN_0		
H'FB46	TDPPDMX_0		
H'FB48	TDPICR_0		
H'FB4A	TDPICRF_0		
H'FB4C	TDPCSR_0		
H'FB4D	TDPCR1_0		
H'FB4E	TDPIER_0		
H'FB4F	TDPCR2_0		
H'FB50	TDPPDMN_0		
H'FB60	TDPCNT_1	MSTPA5 = 0	TDP_1
H'FB62	TDPWDMX_1		
H'FB64	TDPWDMN_1		
H'FB66	TDPPDMX_1		
H'FB68	TDPICR_1		
H'FB6A	TDPICRF_1		
H'FB6C	TDPCSR_1		
H'FB6D	TDPCR1_1		
H'FB6E	TDPIER_1		
H'FB6F	TDPCR2_1		
H'FB70	TDPPDMN_1		
H'FB80	TDPCNT_1	MSTPA4 = 0	TDP_2
H'FB82	TDPWDMX_1		
H'FB84	TDPWDMN_1		
H'FB86	TDPPDMX_1		
H'FB88	TDPICR_1		

Lower Address	Register Abbreviation	Register Selection Condition	Module
H'FB8A	TDPICRF_1		
H'FB8C	TDPCSR_1		
H'FB8D	TDPCR1_1		
H'FB8E	TDPIER_1		
H'FB8F	TDPCR2_1		
H'FB90	TDPPDMN_1		
H'FBC0	TCMCNT_0	MSTPB1 = 0	TCM_0
H'FBC2	TCMMLCM_0		
H'FBC4	TCMICR_0		
H'FBC6	TCMICRF_0		
H'FBC8	TCMCSR_0		
H'FBC9	TCMCR_0		
H'FBCA	TCMIER_0		
H'FBCC	TCMMINCM_0		
H'FBD0	TCMCNT_1	MSTPB1 = 0	TCM_1
H'FBD2	TCMMLCM_1		
H'FBD4	TCMICR_1		
H'FBD6	TCMICRF_1		
H'FBD8	TCMCSR_1		
H'FBD9	TCMCR_1		
H'FBDA	TCMIER_1		
H'FBDC	TCMMINCM_1		
H'FBE0	TCMCNT_2	MSTPB2 = 0	TCM_2
H'FBE2	TCMMLCM_2		
H'FBE4	TCMICR_2		
H'FBE6	TCMICRF_2		
H'FBE8	TCMCSR_2		
H'FBE9	TCMCR_2		
H'FBEA	TCMIER_2		
H'FBEC	TCMMINCM_2		

Lower Address	Register Abbreviation	Register Selection Condition	Module
H'FBF0	TCMCNT_3	MSTPB2 = 0	TCM_3
H'FBF2	TCMMLCM_3		
H'FBF4	TCMICR_3		
H'FBF6	TCMICRF_3		
H'FBF8	TCMCSR_3		
H'FBF9	TCMCR_3		
H'FBFA	TCMIER_3		
H'FBFC	TCMMINCM_3		
H'FC00	ADDRA	MSTP9 = 0	A/D converter
H'FC02	ADDRB		
H'FC04	ADDRC		
H'FC06	ADDRD		
H'FC08	ADDRE		
H'FC0A	ADDRF		
H'FC0C	ADDRG		
H'FC0E	ADDRH		
H'FC10	ADCSR		
H'FC11	ADCR		
H'FC20	FRBR	MSTPB3 = 0	SCIF
H'FC20	FTHR		
H'FC20	FDLL		
H'FC21	FIER		
H'FC21	FDLH		
H'FC22	FIIR		
H'FC22	FFCR		
H'FC23	FLCR		
H'FC24	FMCR		
H'FC25	FLSR		
H'FC26	FMSR		
H'FC27	FSCR		
H'FC28	SCIFCR		

Lower Address	Register Abbreviation	Register Selection Condition	Module
H'FC50	FSIHBARH	MSTP0 = 0	FSI
H'FC51	FSIHBARL	MSTPA2 = 0	
H'FC52	FSISR		
H'FC53	CMDHBARH		
H'FC54	CMDHBARL		
H'FC55	FSICMDR		
H'FC56	FSILSTR1		
H'FC57	FSIGPR1		
H'FC58	FSIGPR2		
H'FC59	FSIGPR3		
H'FC5A	FSIGPR4		
H'FC5B	FSIGPR5		
H'FC5C	FSIGPR6		
H'FC5D	FSIGPR7		
H'FC5E	FSIGPR8		
H'FC5F	FSIGPR9		
H'FC60	FSIGPRA		
H'FC61	FSIGPRB		
H'FC62	FSIGPRC		
H'FC63	FSIGPRD		
H'FC64	FSIGPRE		
H'FC65	FSIGPRF		
H'FC66	SLCR		
H'FC67	FSIARH		
H'FC68	FSIARM		
H'FC69	FSIARL		
H'FC6A	FSIWDRHH		
H'FC6B	FSIWDRHL		
H'FC6C	FSIWDRLH		
H'FC6D	FSIWDRLL		
H'FC6E	FSILSTR2		

Lower Address	Register Abbreviation	Register Selection Condition	Module
H'FC90	FSICR1	MSTP0 = 0	FSI
H'FC91	FSICR2	MSTPA2 = 0	
H'FC92	FSIBNR		
H'FC93	FSIINS		
H'FC94	FSIRDINS		
H'FC95	FSIPPINS		
H'FC96	FSISTR		
H'FC98	FSITDR0		
H'FC99	FSITDR1		
H'FC9A	FSITDR2		
H'FC9B	FSITDR3		
H'FC9C	FSITDR4		
H'FC9D	FSITDR5		
H'FC9E	FSITDR6		
H'FC9F	FSITDR7		
H'FCA0	FSIRDR		
H'FD00	PWMREG0_A	MSTPB0 = 0	PWMU_A
H'FD01	PWMPRE0_A		
H'FD02	PWMREG1_A		
H'FD03	PWMPRE1_A		
H'FD04	PWMREG2_A		
H'FD05	PWMPRE2_A		
H'FD06	PWMREG3_A		
H'FD07	PWMPRE3_A		
H'FD08	PWMREG4_A		
H'FD09	PWMPRE4_A		
H'FD0A	PWMREG5_A		
H'FD0B	PWMPRE5_A		
H'FD0C	PWMCONA_A		
H'FD0D	PWMCONB_A		
H'FD0E	PWMCONC_A		
H'FD0F	PWMCOND_A		

Lower Address	Register Abbreviation	Register Selection Condition	Module
H'FD10	PWMREG0_B	MSTPB0 = 0	PWMU_B
H'FD11	PWMPRE0_B		
H'FD12	PWMREG1_B		
H'FD13	PWMPRE1_B		
H'FD14	PWMREG2_B		
H'FD15	PWMPRE2_B		
H'FD16	PWMREG3_B		
H'FD17	PWMPRE3_B		
H'FD18	PWMREG4_B		
H'FD19	PWMPRE4_B		
H'FD1A	PWMREG5_B		
H'FD1B	PWMPRE5_B		
H'FD1C	PWMCONA_B		
H'FD1D	PWMCONB_B		
H'FD1E	PWMCONC_B		
H'FD1F	PWMCOND_B		
H'FD3A	SYTSR0	No condition	SYSTEM
H'FD3B	SYTSR1		
H'FD40	TCR_1	MSTP1 = 0	TPU_1
H'FD41	TMDR_1		
H'FD42	TIOR_1		
H'FD44	TIER_1		
H'FD45	TSR_1		
H'FD46	TCNT_1		
H'FD48	TGRA_1		
H'FD4A	TGRB_1		

Lower Address	Register Abbreviation	Register Selection Condition	Module
H'FDC0	LADR1H	MSTP0 = 0	LPC
H'FDC1	LADR1L		
H'FDC2	LADR2H		
H'FDC3	LADR2L		
H'FDC4	SCIFADRH		
H'FDC5	SCIFADRL		
H'FDD4	LADR4H		
H'FDD5	LADR4L		
H'FDD6	IDR4		
H'FDD7	ODR4		
H'FDD8	STR4		
H'FDD9	HICR4		
H'FDDA	SIRQCR2		
H'FDDB	SIRQCR3		
H'FE00	P6NCE	PORTS = 0	PORT
H'FE01	P6NCMC		
H'FE02	P6NCCS		
H'FE03	PCNCE		
H'FE04	PCNCMC		
H'FE05	PCNCCS		
H'FE06	PGNCE		
H'FE07	PGNCMC		
H'FE08	PGNCCS		
H'FE0C	PHPIN (Read)		
	PHDDR (Write)		
H'FE0D	PHODR		
H'FE0E	PHNOCR		

Lower Address	Register Abbreviation	Register Selection Condition	Module
H'FE10	PTCNT0	No condition	PORT
H'FE11	PTCNT1		
H'FE12	PTCNT2		
H'FE14	P9PCR	PORTS = 0	
H'FE16	PGNOCR		
H'FE19	PFNOCR		
H'FE1C	PCNOCR		
H'FE1D	PDNOCR		
H'FE20	TWR0MW	MSTP0 = 0	LPC
	TWR0SW		
H'FE21	TWR1		
H'FE22	TWR2		
H'FE23	TWR3		
H'FE24	TWR4		
H'FE25	TWR5		
H'FE26	TWR6		
H'FE27	TWR7		
H'FE28	TWR8		
H'FE29	TWR9		
H'FE2A	TWR10		
H'FE2B	TWR11		
H'FE2C	TWR12		
H'FE2D	TWR13		
H'FE2E	TWR14		
H'FE2F	TWR15		
H'FE30	IDR3		
H'FE31	ODR3		

Lower Address	Register Abbreviation	Register Selection Condition	Module
H'FE32	STR3	MSTP0 = 0	LPC
H'FE33	HICR5		
H'FE34	LADR3H		
H'FE35	LADR3L		
H'FE36	SIRQCR0		
H'FE37	SIRQCR1		
H'FE38	IDR1		
H'FE39	ODR1		
H'FE3A	STR1		
H'FE3B	SIRQCR4		
H'FE3C	IDR2		
H'FE3D	ODR2		
H'FE3E	STR2		
H'FE3F	HISEL		
H'FE40	HICR0		
H'FE41	HICR1		
H'FE42	HICR2		
H'FE43	HICR3		
H'FE45	WUEMR	No condition	INT
H'FE46	PGODR	PORTS = 0	PORT
H'FE47	PGPIN (Read)		
	PGDDR (Write)		
H'FE49	PFODR		
H'FE4A	PEPIN (Read) (write prohibited)		
H'FE4B	PFPIN (Read)		
H'FE4C	PCODR		
H'FE4D	PDODR		
H'FE4E	PCPIN (Read)		
	PCDDR (Write)		
H'FE4F	PDPIN (Read)		
	PDDDR (Write)		

Lower Address	Register Abbreviation	Register Selection Condition	Module
H'FE50	TCR_0	MSTP1 = 0	TPU_0
H'FE51	TMDR_0		
H'FE52	TIORH_0		
H'FE53	TIORL_0		
H'FE54	TIER_0		
H'FE55	TSR_0		
H'FE56	TCNT_0		
H'FE58	TGRA_0		
H'FE5A	TGRB_0		
H'FE5C	TGRC_0		
H'FE5E	TGRD_0		
H'FE70	TCR_2		TPU_2
H'FE71	TMDR_2		
H'FE72	TIOR_2		
H'FE74	TIER_2		
H'FE75	TSR_2		
H'FE76	TCNT_2		
H'FE78	TGRA_2		
H'FE7A	TGRB_2		
H'FE7D	SYSCR3	No condition	SYSTEM
H'FE7E	MSTPCRA		
H'FE7F	MSTPCRB		
H'FE81	KMIMR (RELOCATE = 1)		INT
H'FE82	KMPCR (RELOCATE = 1)		PORT
H'FE83	KMIMRA (RELOCATE = 1)		INT
H'FE84	WUESCR		
H'FE85	WUESR		
H'FE86	WER		
H'FE87	ICRD		

Lower Address	Register Abbreviation	Register Selection Condition		Module
H'FE88	ICCR_2	MSTPB4 = 0		IIC_2
H'FE89	ICSR_2			
H'FE8A	ICRES_2			
H'FE8C	ICXR_2			
H'FE8E	ICDR_2		ICE in ICCR_2 = 1	
H'FE8E	SARX_2		ICE in ICCR_2 = 0	
H'FE8F	ICMR_2		ICE in ICCR_2 = 1	
H'FE8F	SAR_2		ICE in ICCR_2 = 0	
H'FEA0	DACR (RELOCATE = 1)	MSTP11 = 0 MSTPA1 = 0	REGS in DACNT/DADRB = 1	PWMX
	DADRAH (RELOCATE = 1)		REGS in DACNT/DADRB = 0	
H'FEA1	DADRAL (RELOCATE = 1)			
H'FEA6	DADRBH (RELOCATE = 1)			
	DACNTH (RELOCATE = 1)		REGS in DACNT/DADRB = 1	
H'FEA7	DADRBL (RELOCATE = 1)		REGS in DACNT/DADRB = 0	
	DACNTL (RELOCATE = 1)		REGS in DACNT/DADRB = 1	
H'FEA8	FCCS	FLSHE = 1		ROM
H'FEA9	FPCS			
H'FEAA	FECS			
H'FEAC	FKEY			
H'FEAD	FMATS			
H'FEAE	FTDAR			

Lower Address	Register Abbreviation	Register Selection Condition		Module
H'FEB0	TSTR	MSTP1 = 0		TPU common
H'FEB1	TSYR			
H'FEC0	KBCR1_0	MSTP2 = 0		PS2
H'FEC1	KBTR_0			
H'FEC2	KBCR1_1			
H'FEC3	KBTR_1			
H'FEC4	KBCR1_2			
H'FEC5	KBTR_2			
H'FEC6	TCRXY	MSTP8 = 0		TMR_XY
H'FEC8	TCR_Y (RELOCATE = 1)			TMR_Y
H'FEC9	TCSR_Y (RELOCATE = 1)			
H'FECA	TCORA_Y (RELOCATE = 1)			
H'FECB	TCORB_Y (RELOCATE = 1)			
H'FECC	TCNT_Y (RELOCATE = 1)			
H'FECE	ICDR_1 (RELOCATE = 1)	MSTP3 = 0	ICE in ICCR_1 = 1	IIC_1
	SARX_1 (RELOCATE = 1)		ICE in ICCR_1 = 0	
H'FECF	ICMR_1 (RELOCATE = 1)		ICE in ICCR_1 = 1	
	SAR_1 (RELOCATE = 1)		ICE in ICCR_1 = 0	
H'FED0	ICCR_1 (RELOCATE = 1)			
H'FED1	ICSR_1 (RELOCATE = 1)			

Lower Address	Register Abbreviation	Register Selection Condition	Module
H'FED2	KBCR1_3	MSTPB5	PS2_3
H'FED3	KBTR_3		
H'FED4	ICXR_0	MSTP4 = 0	IIC_0
H'FED5	ICXR_1	MSTP3 = 0	IIC_1
H'FED8	KBCRH_0	MSTP2 = 0	PS2
H'FED9	KBCRL_0		
H'FEDA	KBBR_0		
H'FEDB	KBCR2_0		
H'FEDC	KBCRH_1		
H'FEDD	KBCRL_1		
H'FEDE	KBBR_1		
H'FEDF	KBCR2_1		
H'FEE0	KBCRH_2		
H'FEE1	KBCRL_2		
H'FEE2	KBBR_2		
H'FEE3	KBCR2_2		
H'FEE6	ICRES_0	MSTP4 = 0, IICE in STCR = 1	IIC_0
H'FEE8	ICRA	No condition	INT
H'FEE9	ICRB		
H'FEEA	ICRC		
H'FEEB	ISR		
H'FEEC	ISCRH		
H'FEED	ISCRL		
H'FEF4	ABRKCR		
H'FEF5	BARA		
H'FEF6	BARB		
H'FEF7	BARC		
H'FEF8	IER16		
H'FEF9	ISR16		
H'FEFA	ISCR16H		
H'FEFB	ISCR16L		

Lower Address	Register Abbreviation	Register Selection Condition	Module
H'FEFC	ISSR16	No condition	INT
H'FEFD	ISSR		
H'FF82	PCSR (RELOCATE = 0)	FLSHE in STCR = 0	PWMX
	PCSR (RELOCATE = 1)	No condition	
H'FF84	SBYCR (RELOCATE = 0)	FLSHE in STCR = 0	SYSTEM
	SBYCR (RELOCATE = 1)	No condition	
H'FF85	LPWRCR (RELOCATE = 0)	FLSHE in STCR = 0	
	LPWRCR (RELOCATE = 1)	No condition	
H'FF86	MSTPCRH (RELOCATE = 0)	FLSHE in STCR = 0	
	MSTPCRH (RELOCATE = 1)	No condition	
H'FF87	MSTPCRL (RELOCATE = 0)	FLSHE in STCR = 0	SYSTEM
	MSTPCRL (RELOCATE = 1)	No condition	
H'FF88	SMR_1 (RELOCATE = 1)	MSTP6 = 0	SCI_1
	SMR_1 (RELOCATE = 0)	MSTP6 = 0, IICE in STCR = 0	
	ICCR_1 (RELOCATE = 0)	MSTP3 = 0, IICE in STCR = 1	IIC_1
H'FF89	BRR_1 (RELOCATE = 1)	MSTP6 = 0	SCI_1
	BRR_1 (RELOCATE = 0)	MSTP6 = 0, IICE in STCR = 0	
	ICSR_1 (RELOCATE = 0)	MSTP3 = 0, IICE in STCR = 1	IIC_1

Lower Address	Register Abbreviation	Register Selection Condition		Module
H'FF8A	SCR_1	MSTP6 = 0		SCI_1
H'FF8B	TDR_1			
H'FF8C	SSR_1			
H'FF8D	RDR_1			
H'FF8E	SCMR_1 (RELOCATE = 1)	MSTP6 = 0		
	SCMR_1 (RELOCATE = 0)	MSTP6 = 0, IICE in STCR = 0		
	ICDR_1 (RELOCATE = 0)	MSTP3 = 0 IICE in STCR = 1	ICE in ICCR_1 = 1	IIC_1
	SARX_1 (RELOCATE = 0)		ICE in ICCR_1 = 0	
H'FF8F	ICMR_1 (RELOCATE = 0)		ICE in ICCR_1 = 1	
	SAR_1 (RELOCATE = 0)		ICE in ICCR_1 = 0	
H'FFA0	DADRAH (RELOCATE = 0)	MSTP11 = 0 MSTPA1 = 0 IICE in STCR = 1	REGS in DACNT/ DADRB = 0	PWMX
	DACR (RELOCATE = 0)		REGS in DACNT/ DADRB = 1	
	SMR_2 (RELOCATE = 0)	MSTP5 = 0, IICE in STCR = 0		SCI_2
H'FFA1	DADRAL (RELOCATE = 0)	MSTP11 = 0 MSTPA1 = 0 IICE in STCR = 1	REGS in DACNT/ DADRB = 0	PWMX
	DADRBH (RELOCATE = 0)	MSTP11 = 0 MSTPA1 = 0 IICE in STCR = 1	REGS in DACNT/ DADRB = 0	
	DACNTH (RELOCATE = 0)		REGS in DACNT/ DADRB = 1	
	BRR_2 (RELOCATE = 0)	MSTP5 = 0, IICE in STCR = 0		SCI_2
H'FFA2	SCR_2	MSTP5 = 0		
H'FFA3	TDR_2			
H'FFA4	SSR_2			

Lower Address	Register Abbreviation	Register Selection Condition		Module
H'FFA5	RDR_2	MSTP5 = 0		SCI_2
H'FFA6	SCMR_2			
H'FFA7	DADRBL (RELOCATE = 0)	MSTP11 = 0 MSTPA1 = 0 IICE in STCR = 1	REGS in DACNT/ DADRB = 0	PWMX
	DACNTL (RELOCATE = 0)		REGS in DACNT/ DADRB = 1	
H'FFA8	TCSR_0	No condition		WDT_0
	TCNT_0 (Write)			
H'FFA9	TCNT_0 (Read)			
H'FFAA	PAODR	PORTS = 0		PORT
H'FFAB	PAPIN (Read)			
	PADDR (Write)			
H'FFAC	P1PCR			
H'FFAD	P2PCR	PORTS = 0		PORT
H'FFAE	P3PCR			
H'FFB0	P1DDR			
H'FFB1	P2DDR			
H'FFB2	P1DR			
H'FFB3	P2DR			
H'FFB4	P3DDR			
H'FFB5	P4DDR			
H'FFB6	P3DR			
H'FFB7	P4DR			
H'FFB8	P5DDR			
H'FFB9	P6DDR			
H'FFBA	P5DR			
H'FFBB	P6DR			
H'FFBC	PBODR			
H'FFBD	P8DDR (Write)			
	PBPIN (Read)			

Lower Address	Register Abbreviation	Register Selection Condition			Module
H'FFBE	P7PIN (Read)	PORTS = 0			PORT
	PBDDR (Write)				
H'FFBF	P8DR				
H'FFC0	P9DDR				
H'FFC1	P9DR				
H'FFC2	IER	No condition			INT
H'FFC3	STCR	No condition			SYSTEM
H'FFC4	SYSCR				
H'FFC5	MDCR				
H'FFC6	BCR	No condition			BSC
H'FFC7	WSCR				
H'FFC8	TCR_0	MSTP12 = 0			TMR_0, TMR_1
H'FFC9	TCR_1				
H'FFCA	TCSR_0				
H'FFCB	TCSR_1	MSTP12 = 0			TMR_0, TMR_1
H'FFCC	TCORA_0				
H'FFCD	TCORA_1				
H'FFCE	TCORB_0				
H'FFCF	TCORB_1				
H'FFD8	ICCR_0	MSTP4 = 0 IICE in STCR = 1 (RELOCATE = 0)		No condition of IICE = 1 when RELOCATE = 1	IIC_0
H'FFD9	ICSR_0				
H'FFDE	ICDR_0	MSTP4 = 0 IICE in STCR = 1 (RELOCATE = 0)	ICE in ICCR_0 = 1		
	SARX_0		ICE in ICCR_0 = 0		
H'FFDF	ICMR_0	MSTP4 = 0 IICE in STCR = 1	ICE in ICCR_0 = 1		
	SAR_0	(RELOCATE = 0)	ICE in ICCR_0 = 0		

Lower Address	Register Abbreviation	Register Selection Condition		Module
H'FFE0	KBCRH_3	MSTPB5 = 0		PS2_3
H'FFE1	KBCRL_3			
H'FFE2	KBBR_3			
H'FFE3	KBCR2_3			
H'FFEA	TCSR_1	No condition		WDT_1
	TCNT_1 (Write)			
H'FFEB	TCNT_1 (Read)			
H'FFF0	TCR_X (RELOCATE = 1)	MSTP8 = 0		TMR_X
	TCR_X (RELOCATE = 0)	MSTP8 = 0 KINWUE in SYSCR = 0	TMRX/Y in TCONRS = 0	
	TCR_Y (RELOCATE = 0)		TMRX/Y in TCONRS = 1	TMR_Y
H'FFF1	KMIMR (RELOCATE = 0)	MSTP2 = 0 KINWUE in SYSCR = 1		INT
	TCSR_X (RELOCATE = 1)	MSTP8 = 0		TMR_X
	TCSR_X (RELOCATE = 0)	MSTP8 = 0 KINWUE in SYSCR = 0	TMRX/Y in TCONRS = 0	
	TCSR_Y (RELOCATE = 0)		TMRX/Y in TCONRS = 1	TMR_Y
H'FFF2	KMPCR (RELOCATE = 0)	MSTP2 = 0 KINWUE in SYSCR = 1		PORT
	TICRR (RELOCATE = 1)	MSTP8 = 0		TMR_X
	TICRR (RELOCATE = 0)	MSTP8 = 0 KINWUE in SYSCR = 0	TMRX/Y in TCONRS = 0	
	TCORA_Y (RELOCATE = 0)		TMRX/Y in TCONRS = 1	TMR_Y

Lower Address	Register Abbreviation	Register Selection Condition		Module
H'FFF3	KMIMRA (RELOCATE = 0)	MSTP2 = 0, KINWUE in SYSCR = 1		INT
	TICRF (RELOCATE = 1)	MSTP8 = 0		TMR_X
	TICRF (RELOCATE = 0)	MSTP8 = 0 KINWUE in SYSCR = 0	TMRX/Y in TCONRS = 0	
	TCORB_Y (RELOCATE = 0)		TMRX/Y in TCONRS = 1	TMR_Y
H'FFF4	TCNT_X (RELOCATE = 1)	MSTP8 = 0		TMR_X
	TCNT_X (RELOCATE = 0)	MSTP8 = 0 KINWUE in SYSCR = 0	TMRX/Y in TCONRS = 0	
	TCNT_Y (RELOCATE = 0)		TMRX/Y in TCONRS = 1	TMR_Y
H'FFF5	TCORC (RELOCATE = 1)	MSTP8 = 0		TMR_X
	TCORC (RELOCATE = 0)	MSTP8 = 0 KINWUE in SYSCR = 0	TMRX/Y in TCONRS = 0	
H'FFF6	TCORA_X (RELOCATE = 1)	MSTP8 = 0		TMR_X
	TCORA_X (RELOCATE = 0)	MSTP8 = 0 KINWUE in SYSCR = 0	TMRX/Y in TCONRS = 0	
H'FFF7	TCORB_X (RELOCATE = 1)	MSTP8 = 0		TMR_X
	TCORB_X (RELOCATE = 0)	MSTP8 = 0 KINWUE in SYSCR = 0	TMRX/Y in TCONRS = 0	
H'FFFC	TCONRI (RELOCATE = 1)	MSTP8 = 0		TMR_X
	TCONRI (RELOCATE = 0)	MSTP8 = 0, KINWUE in SYSCR = 0		
H'FFFE	TCONRS (RELOCATE = 1)	MSTP8 = 0		TMR_X, TMR_Y
	TCONRS (RELOCATE = 0)	MSTP8 = 0, KINWUE in SYSCR = 0		

27.5 Register Addresses (Classification by Type of Module)

Module	Register Abbreviation	Number of Bits	Address	Data Bus Width	Access States
INT	WUEMR	8	H'FE45	8	2
INT	KMIMR	8	H'FE81 (RELOCATE = 1)	8	2
INT	KMIMRA	8	H'FE83 (RELOCATE = 1)	8	2
INT	WUESCR	8	H'FE84	8	2
INT	WUESR	8	H'FE85	8	2
INT	WER	8	H'FE86	8	2
INT	ICRD	8	H'FE87	8	2
INT	ICRA	8	H'FEE8	8	2
INT	ICRB	8	H'FEE9	8	2
INT	ICRC	8	H'FEEA	8	2
INT	ISR	8	H'FEEB	8	2
INT	ISCRH	8	H'FEEC	8	2
INT	ISCRL	8	H'FEED	8	2
INT	KMIMR	8	H'FFF1 (RELOCATE = 0)	8	2
INT	ABRKCR	8	H'FEF4	8	2
INT	BARA	8	H'FEF5	8	2
INT	BARB	8	H'FEF6	8	2
INT	BARC	8	H'FEF7	8	2
INT	IER16	8	H'FEF8	8	2
INT	ISR16	8	H'FEF9	8	2
INT	ISCR16H	8	H'FEFA	8	2
INT	ISCR16L	8	H'FEFB	8	2
INT	ISSR16	8	H'FEFC	8	2
INT	ISSR	8	H'FEFD	8	2
INT	IER	8	H'FFC2	8	2
INT	KMIMRA	8	H'FFF3 (RELOCATE = 0)	8	2

Module	Register Abbreviation	Number of Bits	Address	Data Bus Width	Access States
BSC	BCR	8	H'FFC6	8	2
BSC	WSCR	8	H'FFC7	8	2
PORT	P1DDR	8	H'F900 (PORTS = 1)	8	2
PORT	P2DDR	8	H'F901 (PORTS = 1)	8	2
PORT	P1DR	8	H'F902 (PORTS = 1)	8	2
PORT	P2DR	8	H'F903 (PORTS = 1)	8	2
PORT	P1PIN	8	H'F904 (Read) (PORTS = 1)	8	2
PORT	P2PIN	8	H'F905 (Read) (PORTS = 1)	8	2
PORT	P1PCR	8	H'F906 (PORTS = 1)	8	2
PORT	P2PCR	8	H'F907 (PORTS = 1)	8	2
PORT	P3DDR	8	H'F910 (PORTS = 1)	8	2
PORT	P4DDR	8	H'F911 (PORTS = 1)	8	2
PORT	P3DR	8	H'F912 (PORTS = 1)	8	2
PORT	P4DR	8	H'F913 (PORTS = 1)	8	2
PORT	P3PIN	8	H'F914 (Read) (PORTS = 1)	8	2
PORT	P4PIN	8	H'F915 (Read) (PORTS = 1)	8	2
PORT	P3PCR	8	H'F916 (PORTS = 1)	8	2
PORT	P5DDR	8	H'F920 (PORTS = 1)	8	2
PORT	P6DDR	8	H'F921 (PORTS = 1)	8	2

Module	Register Abbreviation	Number of Bits	Address	Data Bus Width	Access States
PORT	P5DR	8	H'F922 (PORTS = 1)	8	2
PORT	P6DR	8	H'F923 (PORTS = 1)	8	2
PORT	P5PIN	8	H'F924 (Read) (PORTS = 1)	8	2
PORT	P6PIN	8	H'F925 (Read) (PORTS = 1)	8	2
PORT	P6NCE	8	H'F92B (PORTS = 1)	8	2
PORT	P6NCMC	8	H'F92D (PORTS = 1)	8	2
PORT	P6NCCS	8	H'F92F (PORTS = 1)	8	2
PORT	P8DDR	8	H'F931 (PORTS = 1)	8	2
PORT	P8DR	8	H'F933 (PORTS = 1)	8	2
PORT	P7PIN	8	H'F934 (Read) (PORTS = 1)	8	2
PORT	P8PIN	8	H'F935 (Read) (PORTS = 1)	8	2
PORT	P9DDR	8	H'F940 (PORTS = 1)	8	2
PORT	P9DR	8	H'F942 (PORTS = 1)	8	2
PORT	P9PIN	8	H'F944 (Read) (PORTS = 1)	8	2
PORT	P9PCR	8	H'F946 (PORTS = 1)	8	2
PORT	PADDR	8	H'F950 (PORTS = 1)	8	2
PORT	PBDDR	8	H'F951 (PORTS = 1)	8	2
PORT	PAODR	8	H'F952 (PORTS = 1)	8	2

Module	Register Abbreviation	Number of Bits	Address	Data Bus Width	Access States
PORT	PBODR	8	H'F953 (PORTS = 1)	8	2
PORT	PAPIN	8	H'F954 (Read) (PORTS = 1)	8	2
PORT	PBPIN	8	H'F955 (Read) (PORTS = 1)	8	2
PORT	PBPCR	8	H'F957 (PORTS = 1)	8	2
PORT	PCDDR	8	H'F960 (PORTS = 1)	8	2
PORT	PDDDR	8	H'F961 (PORTS = 1)	8	2
PORT	PCODR	8	H'F962 (PORTS = 1)	8	2
PORT	PDODR	8	H'F963 (PORTS = 1)	8	2
PORT	PCPIN	8	H'F964 (Read) (PORTS = 1)	8	2
PORT	PDPIN	8	H'F965 (Read) (PORTS = 1)	8	2
PORT	PCPCR	8	H'F966 (PORTS = 1)	8	2
PORT	PDPCR	8	H'F967 (PORTS = 1)	8	2
PORT	PCNOCR	8	H'F968 (PORTS = 1)	8	2
PORT	PDNOCR	8	H'F969 (PORTS = 1)	8	2
PORT	PCNCE	8	H'F96A (PORTS = 1)	8	2
PORT	PCNCMC	8	H'F96C (PORTS = 1)	8	2
PORT	PCNCCS	8	H'F96E (PORTS = 1)	8	2
PORT	PFDDR	8	H'F971 (PORTS = 1)	8	2

Module	Register Abbreviation	Number of Bits	Address	Data Bus Width	Access States
PORT	PFODR	8	H'F973 (PORTS = 1)	8	2
PORT	PEPIN	8	H'F974 (Read) (PORTS = 1)	8	2
PORT	PFPIN	8	H'F975 (Read) (PORTS = 1)	8	2
PORT	PFPCR	8	H'F977 (PORTS = 1)	8	2
PORT	PFNOCR	8	H'F979 (PORTS = 1)	8	2
PORT	PGDDR	8	H'F980 (PORTS = 1)	8	2
PORT	PHDDR	8	H'F981 (PORTS = 1)	8	2
PORT	PGODR	8	H'F982 (PORTS = 1)	8	2
PORT	PHODR	8	H'F983 (PORTS = 1)	8	2
PORT	PGPIN	8	H'F984 (Read) (PORTS = 1)	8	2
PORT	PHPIN	8	H'F985 (Read) (PORTS = 1)	8	2
PORT	PHPCR	8	H'F987 (PORTS = 1)	8	2
PORT	PGNOCR	8	H'F988 (PORTS = 1)	8	2
PORT	PHNOCR	8	H'F989 (PORTS = 1)	8	2
PORT	PGNCE	8	H'F98A (PORTS = 1)	8	2
PORT	PGNCMC	8	H'F98C (PORTS = 1)	8	2
PORT	PGNCCS	8	H'F98E (PORTS = 1)	8	2
PORT	PIDDR	8	H'F990	8	2
PORT	PJDDR	8	H'F991	8	2

Module	Register Abbreviation	Number of Bits	Address	Data Bus Width	Access States
PORT	PIODR	8	H'F992	8	2
PORT	PJODR	8	H'F993	8	2
PORT	PIPIN	8	H'F994 (Read)	8	2
PORT	PJPIN	8	H'F995 (Read)	8	2
PORT	PJPCR	8	H'F997	8	2
PORT	PINOCR	8	H'F998	8	2
PORT	PJNOCR	8	H'F999	8	2
PORT	P6NCE	8	H'FE00 (PORTS = 0)	8	2
PORT	P6NCMC	8	H'FE01 (PORTS = 0)	8	2
PORT	P6NCCS	8	H'FE02 (PORTS = 0)	8	2
PORT	PCNCE	8	H'FE03 (PORTS = 0)	8	2
PORT	PCNCMC	8	H'FE04 (PORTS = 0)	8	2
PORT	PCNCCS	8	H'FE05 (PORTS = 0)	8	2
PORT	PGNCE	8	H'FE06 (PORTS = 0)	8	2
PORT	PGNCMC	8	H'FE07 (PORTS = 0)	8	2
PORT	PGNCCS	8	H'FE08 (PORTS = 0)	8	2
PORT	PHPIN	8	H'FE0C (Read) (PORTS = 0)	8	2
PORT	PHDDR	8	H'FE0C (Write) (PORTS = 0)	8	2
PORT	PHODR	8	H'FE0D (PORTS = 0)	8	2
PORT	PHNOCR	8	H'FE0E (PORTS = 0)	8	2

Module	Register Abbreviation	Number of Bits	Address	Data Bus Width	Access States
PORT	PTCNT0	8	H'FE10 (PORTS = 0)	8	2
PORT	PTCNT1	8	H'FE11 (PORTS = 0)	8	2
PORT	PTCNT2	8	H'FE12 (PORTS = 0)	8	2
PORT	P9PCR	8	H'FE14 (PORTS = 0)	8	2
PORT	PGNOCR	8	H'FE16 (PORTS = 0)	8	2
PORT	PFNOCR	8	H'FE19 (PORTS = 0)	8	2
PORT	PCNOCR	8	H'FE1C (PORTS = 0)	8	2
PORT	PDNOCR	8	H'FE1D (PORTS = 0)	8	2
PORT	PGODR	8	H'FE46 (PORTS = 0)	8	2
PORT	PGPIN	8	H'FE47 (Read) (PORTS = 0)	8	2
PORT	PGDDR	8	H'FE47 (Write) (PORTS = 0)	8	2
PORT	PFODR	8	H'FE49 (PORTS = 0)	8	2
PORT	PEPIN	8	H'FE4A (Read) (write prohibited) (PORTS = 0)	8	2
PORT	PFPIN	8	H'FE4B (Read) (PORTS = 0)	8	2
PORT	PFDDR	8	H'FE4B (Write) (PORTS = 0)	8	2

Module	Register Abbreviation	Number of Bits	Address	Data Bus Width	Access States
PORT	PCODR	8	H'FE4C (PORTS = 0)	8	2
PORT	PDODR	8	H'FE4D (PORTS = 0)	8	2
PORT	PCPIN	8	H'FE4E (Read) (PORTS = 0)	8	2
PORT	PCDDR	8	H'FE4E (Write) (PORTS = 0)	8	2
PORT	PDPIN	8	H'FE4F (Read) (PORTS = 0)	8	2
PORT	PDDDR	8	H'FE4F (Write) (PORTS = 0)	8	2
PORT	KMPCR	8	H'FE82 (RELOCATE = 1) (PORTS = 0)	8	2
PORT	PAODR	8	H'FFAA (PORTS = 0)	8	2
PORT	PAPIN	8	H'FFAB (Read) (PORTS = 0)	8	2
PORT	PADDR	8	H'FFAB (Write) (PORTS = 0)	8	2
PORT	P1PCR	8	H'FFAC (PORTS = 0)	8	2
PORT	P2PCR	8	H'FFAD (PORTS = 0)	8	2
PORT	P3PCR	8	H'FFAE (PORTS = 0)	8	2
PORT	P1DDR	8	H'FFB0 (PORTS = 0)	8	2
PORT	P2DDR	8	H'FFB1 (PORTS = 0)	8	2
PORT	P1DR	8	H'FFB2 (PORTS = 0)	8	2

Module	Register Abbreviation	Number of Bits	Address	Data Bus Width	Access States
PORT	P2DR	8	H'FFB3 (PORTS = 0)	8	2
PORT	P3DDR	8	H'FFB4 (PORTS = 0)	8	2
PORT	P4DDR	8	H'FFB5 (PORTS = 0)	8	2
PORT	P3DR	8	H'FFB6 (PORTS = 0)	8	2
PORT	P4DR	8	H'FFB7 (PORTS = 0)	8	2
PORT	P5DDR	8	H'FFB8 (PORTS = 0)	8	2
PORT	P6DDR	8	H'FFB9 (PORTS = 0)	8	2
PORT	P5DR	8	H'FFBA (PORTS = 0)	8	2
PORT	P6DR	8	H'FFBB (PORTS = 0)	8	2
PORT	PBODR	8	H'FFBC (PORTS = 0)	8	2
PORT	P8DDR	8	H'FFBD (Write) (PORTS = 0)	8	2
PORT	PBPIN	8	H'FFBD (Read) (PORTS = 0)	8	2
PORT	P7PIN	8	H'FFBE (Read) (PORTS = 0)	8	2
PORT	PBDDR	8	H'FFBE (Write) (PORTS = 0)	8	2
PORT	P8DR	8	H'FFBF (PORTS = 0)	8	2
PORT	P9DDR	8	H'FFC0 (PORTS = 0)	8	2
PORT	P9DR	8	H'FFC1 (PORTS = 0)	8	2
PORT	KMPCR	8	H'FFF2 (RELOCATE = 0) (PORTS = 0)	8	2

Module	Register Abbreviation	Number of Bits	Address	Data Bus Width	Access States
TDP_0	TDPCNT_0	8	H'FB40	8	2
TDP_0	TDPWDMX_0	8	H'FB42	8	2
TDP_0	TDPWDMN_0	8	H'FB44	8	2
TDP_0	TDPPDMX_0	8	H'FB46	8	2
TDP_0	TDPICR_0	8	H'FB48	8	2
TDP_0	TDPICRF_0	8	H'FB8A	8	2
TDP_0	TDPCSR_0	8	H'FB8C	8	2
TDP_0	TDPCR1_0	8	H'FB4D	8	2
TDP_0	TDPIER_0	8	H'FB4E	8	2
TDP_0	TDPCR2_0	8	H'FB4F	8	2
TDP_0	TDPPDMN_0	16	H'FB50	16	2
TDP_1	TDPCNT_1	16	H'FB60	16	2
TDP_1	TDPWDMX_1	16	H'FB62	6	2
TDP_1	TDPWDMN_1	16	H'FB64	16	2
TDP_1	TDPPDMX_1	16	H'FB66	16	2
TDP_1	TDPICR_1	16	H'FB68	16	2
TDP_1	TDPICRF_1	16	H'FB6A	16	2
TDP_1	TDPCSR_1	8	H'FB6C	8	2
TDP_1	TDPCR1_1	8	H'FB6D	8	2
TDP_1	TDPIER_1	8	H'FB6E	8	2
TDP_1	TDPCR2_1	8	H'FB6F	8	2
TDP_1	TDPPDMN_1	16	H'FB70	16	2
TDP_2	TDPCNT_2	16	H'FB80	16	2
TDP_2	TDPWDMX_2	16	H'FB82	16	2
TDP_2	TDPWDMN_2	16	H'FB84	16	2
TDP_2	TDPPDMX_2	16	H'FB86	16	2
TDP_2	TDPICR_2	16	H'FB88	16	2
TDP_2	TDPICRF_2	16	H'FB8A	16	2
TDP_2	TDPCSR_2	8	H'FB8C	8	2
TDP_2	TDPCR1_2	8	H'FB8D	8	2
TDP_2	TPDIER_2	8	H'FB8E	8	2

Module	Register Abbreviation	Number of Bits	Address	Data Bus Width	Access States
TDP_2	TDPCR2_2	8	H'FB8F	8	2
TDP_2	TDPPDMN_2	16	H'FB90	16	2
TCM_0	TCMCNT_0	16	H'FBC0	16	2
TCM_0	TCMMLCM_0	16	H'FBC2	16	2
TCM_0	TCMICR_0	16	H'FBC4	16	2
TCM_0	TCMICRF_0	16	H'FBC6	16	2
TCM_0	TCMCSR_0	8	H'FBC8	8	2
TCM_0	TCMCR_0	8	H'FBC9	8	2
TCM_0	TCMIER_0	8	H'FBCA	8	2
TCM_0	TCMMINCM_0	16	H'FBCC	16	2
TCM_1	TCMCNT_1	16	H'FBD0	16	2
TCM_1	TCMMLCM_1	16	H'FBD2	16	2
TCM_1	TCMICR_1	16	H'FBD4	16	2
TCM_1	TCMICRF_1	16	H'FBD6	16	2
TCM_1	TCMCSR_1	8	H'FBD8	8	2
TCM_1	TCMCR_1	8	H'FBD9	8	2
TCM_1	TCMIER_1	8	H'FBDA	8	2
TCM_1	TCMMINCM_1	16	H'FBDC	16	2
TCM_2	TCMCNT_2	16	H'FBE0	16	2
TCM_2	TCMMLCM_2	16	H'FBE2	16	2
TCM_2	TCMICR_2	16	H'FBE4	16	2
TCM_2	TCMICRF_2	16	H'FBE6	16	2
TCM_2	TCMCSR_2	8	H'FBE8	8	2
TCM_2	TCMCR_2	8	H'FBE9	8	2
TCM_2	TCMIER_2	8	H'FBEA	8	2
TCM_2	TCMMINCM_2	16	H'FBEC	16	2
TCM_3	TCMCNT_3	16	H'FBF0	16	2
TCM_3	TCMMLCM_3	16	H'FBF2	16	2
TCM_3	TCMICR_3	16	H'FBF4	16	2
TCM_3	TCMICRF_3	16	H'FBF6	16	2
TCM_3	TCMCSR_3	8	H'FBF8	8	2

Module	Register Abbreviation	Number of Bits	Address	Data Bus Width	Access States
TCM_3	TCMCR_3	8	H'FBF9	8	2
TCM_3	TCMIER_3	8	H'FBFA	8	2
TCM_3	TCMMINCM_3	16	H'FBFC	16	2
FSI	FSIHBARH	8	H'FC50	8	2
FSI	FSIHBARL	8	H'FC51	8	2
FSI	FSISR	8	H'FC52	8	2
FSI	CMDHBARH	8	H'FC53	8	2
FSI	CMDHBARH	8	H'FC54	8	2
FSI	FSICMDR	8	H'FC55	8	2
FSI	FSILSTR1	8	H'FC56	8	2
FSI	FSIGPR1	8	H'FC57	8	2
FSI	FSIGPR2	8	H'FC58	8	2
FSI	FSIGPR3	8	H'FC59	8	2
FSI	FSIGPR4	8	H'FC5A	8	2
FSI	FSIGPR5	8	H'FC5B	8	2
FSI	FSIGPR6	8	H'FC5C	8	2
FSI	FSIGPR7	8	H'FC5D	8	2
FSI	FSIGPR8	8	H'FC5E	8	2
FSI	FSIGPR9	8	H'FC5F	8	2
FSI	FSIGPRA	8	H'FC60	8	2
FSI	FSIGPRB	8	H'FC61	8	2
FSI	FSIGPRC	8	H'FC62	8	2
FSI	FSIGPRD	8	H'FC63	8	2
FSI	FSIGPRE	8	H'FC64	8	2
FSI	FSIGPRF	8	H'FC65	8	2
FSI	SLCR	8	H'FC66	8	2
FSI	FSIARH	8	H'FC67	8	2
FSI	FSIARM	8	H'FC68	8	2
FSI	FSIARL	8	H'FC69	8	2
FSI	FSIWDRHH	8	H'FC6A	8	2
FSI	FSIWDRHL	8	H'FC6B	8	2

Module	Register Abbreviation	Number of Bits	Address	Data Bus Width	Access States
FSI	FSIWDRLH	8	H'FC6C	8	2
FSI	FSIWDRLL	8	H'FC6D	8	2
FSI	FSILSTR2	8	H'FC6E	8	2
FSI	FSICR1	8	H'FC90	8	2
FSI	FSICR2	8	H'FC91	8	2
FSI	FSIBNR	8	H'FC92	8	2
FSI	FSINS	8	H'FC93	8	2
FSI	FSIRDINS	8	H'FC94	8	2
FSI	FSIPPINS	8	H'FC95	8	2
FSI	FSISTR	8	H'FC96	8	2
FSI	FSITDR0	8	H'FC98	8	2
FSI	FSITDR1	8	H'FC99	8	2
FSI	FSITDR2	8	H'FC9A	8	2
FSI	FSITDR3	8	H'FC9B	8	2
FSI	FSITDR4	8	H'FC9C	8	2
FSI	FSITDR5	8	H'FC9D	8	2
FSI	FSITDR6	8	H'FC9E	8	2
FSI	FSITDR7	8	H'FC9F	8	2
FSI	FSIRDR	8	H'FCA0	8	2
CIR	CCR1	8	H'FA40	8	2
CIR	CCR2	8	H'FA41	8	2
CIR	CSTR	8	H'FA42	8	2
CIR	CEIR	8	H'FA43	8	2
CIR	BRR	8	H'FA44	8	2
CIR	CIRRDR0 to 7	8	H'FA45	8	2
CIR	HHMIN	16	H'FA46	8	2
CIR	HHMAX	16	H'FA48	8	2
CIR	HLMIN	8	H'FA4A	8	2
CIR	HLMAX	8	H'FA4B	8	2
CIR	DT0MIN	8	H'FA4C	8	2
CIR	DT0MAX	8	H'FA4D	8	2

Module	Register Abbreviation	Number of Bits	Address	Data Bus Width	Access States
CIR	DT1MIN	8	H'FA4E	8	2
CIR	DT1MAX	8	H'FA4F	8	2
CIR	RMIN	8	H'FA50	8	2
CIR	RMAX	8	H'FA51	8	2
PWMU_A	PWMREG0	8	H'FD00	8	2
PWMU_A	PWMPRE0	8	H'FD01	8	2
PWMU_A	PWMREG1	8	H'FD02	8	2
PWMU_A	PWMPRE1	8	H'FD03	8	2
PWMU_A	PWMREG2	8	H'FD04	8	2
PWMU_A	PWMPRE2	8	H'FD05	8	2
PWMU_A	PWMREG3	8	H'FD06	8	2
PWMU_A	PWMPRE3	8	H'FD07	8	2
PWMU_A	PWMREG4	8	H'FD08	8	2
PWMU_A	PWMPRE4	8	H'FD09	8	2
PWMU_A	PWMREG5	8	H'FD0A	8	2
PWMU_A	PWMPRE5	8	H'FD0B	8	2
PWMU_A	PWMCONA	8	H'FD0C	8	2
PWMU_A	PWMCONB	8	H'FD0D	8	2
PWMU_A	PWMCONC	8	H'FD0E	8	2
PWMU_A	PWMCOND	8	H'FD0F	8	2
PWMU_B	PWMREG0	8	H'FD10	8	2
PWMU_B	PWMPRE0	8	H'FD11	8	2
PWMU_B	PWMREG1	8	H'FD12	8	2
PWMU_B	PWMPRE1	8	H'FD13	8	2
PWMU_B	PWMREG2	8	H'FD14	8	2
PWMU_B	PWMPRE2	8	H'FD15	8	2
PWMU_B	PWMREG3	8	H'FD16	8	2
PWMU_B	PWMPRE3	8	H'FD17	8	2
PWMU_B	PWMREG4	8	H'FD18	8	2
PWMU_B	PWMPRE4	8	H'FD19	8	2
PWMU_B	PWMREG5	8	H'FD1A	8	2
PWMU_B	PWMPRE5	8	H'FD1B	8	2

Module	Register Abbreviation	Number of Bits	Address	Data Bus Width	Access States
PWMU_B	PWMCONA	8	H'FD1C	8	2
PWMU_B	PWMCONB	8	H'FD1D	8	2
PWMU_B	PWMCONC	8	H'FD1E	8	2
PWMU_B	PWMCOND	8	H'FD1F	8	2
PWMX	DACR	8	H'FEA0 (RELOCATE = 1)	8	2
PWMX	DADRAH	8	H'FEA0 (RELOCATE = 1)	8	2
PWMX	DADRAL	8	H'FEA1 (RELOCATE = 1)	8	2
PWMX	DADRBH	8	H'FEA6 (RELOCATE = 1)	8	2
PWMX	DACNTH	8	H'FEA6 (RELOCATE = 1)	8	2
PWMX	DADRBL	8	H'FEA7 (RELOCATE = 1)	8	2
PWMX	DACNTL	8	H'FEA7 (RELOCATE = 1)	8	2
PWMX	PCSR	8	H'FF82	8	2
PWMX	DACR	8	H'FFA0 (RELOCATE = 0)	8	2
PWMX	DADRAH	8	H'FFA0 (RELOCATE = 0)	8	2
PWMX	DADRAL	8	H'FFA1 (RELOCATE = 0)	8	2
PWMX	DACNTH	8	H'FFA6 (RELOCATE = 0)	8	2
PWMX	DADRBH	8	H'FFA6 (RELOCATE = 0)	8	2
PWMX	DACNTL	8	H'FFA7 (RELOCATE = 0)	8	2
PWMX	DADRBL	8	H'FFA7 (RELOCATE = 0)	8	2
TPU_0	TCR_0	8	H'FE50	8	2
TPU_0	TMDR_0	8	H'FE51	8	2

Module	Register Abbreviation	Number of Bits	Address	Data Bus Width	Access States
TPU_0	TIORH_0	8	H'FE52	8	2
TPU_0	TIORL_0	8	H'FE53	8	2
TPU_0	TIER_0	8	H'FE54	8	2
TPU_0	TSR_0	8	H'FE55	8	2
TPU_0	TCNT_0	16	H'FE56	16	2
TPU_0	TGRA_0	16	H'FE58	16	2
TPU_0	TGRB_0	16	H'FE5A	16	2
TPU_0	TGRC_0	16	H'FE5C	16	2
TPU_0	TGRD_0	16	H'FE5E	16	2
TPU_1	TCR_1	8	H'FD40	8	2
TPU_1	TMDR_1	8	H'FD41	8	2
TPU_1	TIOR_1	8	H'FD42	8	2
TPU_1	TIER_1	8	H'FD44	8	2
TPU_1	TSR_1	8	H'FD45	8	2
TPU_1	TCNT_1	16	H'FD46	16	2
TPU_1	TGRA_1	16	H'FD48	16	2
TPU_1	TGRB_1	16	H'FD4A	16	2
TPU_2	TCR_2	8	H'FE70	8	2
TPU_2	TMDR_2	8	H'FE71	8	2
TPU_2	TIOR_2	8	H'FE72	8	2
TPU_2	TIER_2	8	H'FE74	8	2
TPU_2	TSR_2	8	H'FE75	8	2
TPU_2	TCNT_2	16	H'FE76	16	2
TPU_2	TGRA_2	16	H'FE78	16	2
TPU_2	TGRB_2	16	H'FE7A	16	2
TPU common	TSTR	8	H'FEB0	8	2
TPU common	TSYR	8	H'FEB1	8	2
TMR_0	TCR_0	8	H'FFC8	8	2
TMR_0	TCSR_0	8	H'FFCA	8	2
TMR_0	TCORA_0	8	H'FFCC	16	2

Module	Register Abbreviation	Number of Bits	Address	Data Bus Width	Access States
TMR_0	TCORB_0	8	H'FFCE	16	2
TMR_0	TCNT_0	8	H'FFD0	16	2
TMR_1	TCR_1	8	H'FFC9	8	2
TMR_1	TCSR_1	8	H'FFCB	16	2
TMR_1	TCORA_1	8	H'FFCD	16	2
TMR_1	TCORB_1	8	H'FFCF	16	2
TMR_1	TCNT_1	8	H'FFD1	16	2
TMR_X	TCR_X	8	H'FFF0	8	2
TMR_X	TCSR_X	8	H'FFF1	8	2
TMR_X	TICRR	8	H'FFF2	8	2
TMR_X	TICRF	8	H'FFF3	8	2
TMR_X	TCNT_X	8	H'FFF4	8	2
TMR_X	TCORC	8	H'FFF5	8	2
TMR_X	TCORA_X	8	H'FFF6	8	2
TMR_X	TCORB_X	8	H'FFF7	8	2
TMR_X	TCONRI	8	H'FFFC	8	2
TMR_Y	TCR_Y	8	H'FEC8 (RELOCATE = 1)	8	2
TMR_Y	TCSR_Y	8	H'FEC9 (RELOCATE = 1)	8	2
TMR_Y	TCORA_Y	8	H'FECA (RELOCATE = 1)	8	2
TMR_Y	TCORB_Y	8	H'FECB (RELOCATE = 1)	8	2
TMR_Y	TCNT_Y	8	H'FECC (RELOCATE = 1)	8	2
TMR_Y	TCR_Y	8	H'FFF0 (RELOCATE = 0)	8	2
TMR_Y	TCSR_Y	8	H'FFF1 (RELOCATE = 0)	8	2
TMR_Y	TCORA_Y	8	H'FFF2 (RELOCATE = 0)	8	2
TMR_Y	TCORB_Y	8	H'FFF3 (RELOCATE = 0)	8	2

Module	Register Abbreviation	Number of Bits	Address	Data Bus Width	Access States
TMR_Y	TCNT_Y	8	H'FFF4 (RELOCATE = 0)	8	2
TMR_XY	TCRXY	8	H'FEC6	8	2
TMR_X	TCONRI	8	H'FFFC	8	2
TMR_X, TMR_Y	TCONRS	8	H'FFFE	8	2
WDT_0	TCSR_0	8	H'FFA8 (Write)	16	2
WDT_0	TCSR_0	8	H'FFA8 (Read)	8	2
WDT_0	TCNT_0	8	H'FFA8 (Write)	16	2
WDT_0	TCNT_0	8	H'FFA9 (Read)	8	2
WDT_1	TCSR_1	8	H'FFEA (Write)	16	2
WDT_1	TCSR_1	8	H'FFEA (Read)	8	2
WDT_1	TCNT_1	8	H'FFEA (Write)	16	2
WDT_1	TCNT_1	8	H'FFEB (Read)	8	2
SCI_1	SMR_1	8	H'FF88	8	2
SCI_1	BRR_1	8	H'FF89	8	2
SCI_1	SCR_1	8	H'FF8A	8	2
SCI_1	TDR_1	8	H'FF8B	8	2
SCI_1	SSR_1	8	H'FF8C	8	2
SCI_1	RDR_1	8	H'FF8D	8	2
SCI_1	SCMR_1	8	H'FF8E	8	2
SCI2	SMR_2	8	H'FFA0	8	2
SCI2	BRR_2	8	H'FFA1	8	2
SCI2	SCR_2	8	H'FFA2	8	2
SCI2	TDR_2	8	H'FFA3	8	2
SCI2	SSR_2	8	H'FFA4	8	2
SCI2	RDR_2	8	H'FFA5	8	2
SCI2	SCMR_2	8	H'FFA6	8	2
IIC_0	ICXR_0	8	H'FED4	8	2
IIC_0	ICCR_0	8	H'FFD8	8	2
IIC_0	ICSR_0	8	H'FFD9	8	2
IIC_0	ICDR_0	8	H'FFDE	8	2

Module	Register Abbreviation	Number of Bits	Address	Data Bus Width	Access States
IIC_0	SARX_0	8	H'FFDE	8	2
IIC_0	ICMR_0	8	H'FFDF	8	2
IIC_0	SAR_0	8	H'FFDF	8	2
IIC_1	ICDR_1	8	H'FECE (RELOCATE = 1)	8	2
IIC_1	SARX_1	8	H'FECE (RELOCATE = 1)	8	2
IIC_1	ICMR_1	8	H'FECF (RELOCATE = 1)	8	2
IIC_1	SAR_1	8	H'FECF (RELOCATE = 1)	8	2
IIC_1	ICCR_1	8	H'FED0 (RELOCATE = 1)	8	2
IIC_1	ICSR_1	8	H'FED1 (RELOCATE = 1)	8	2
IIC_1	ICXR_1	8	H'FED5	8	2
IIC_1	ICCR_1	8	H'FF88 (RELOCATE = 0)	8	2
IIC_1	ICSR_1	8	H'FF89 (RELOCATE = 0)	8	2
IIC_1	ICDR_1	8	H'FF8E (RELOCATE = 0)	8	2
IIC_1	SARX_1	8	H'FF8E (RELOCATE = 0)	8	2
IIC_1	ICMR_1	8	H'FF8F (RELOCATE = 0)	8	2
IIC_1	SAR_1	8	H'FF8F (RELOCATE = 0)	8	2
IIC_2	ICCR_2	8	H'FE88	8	2
IIC_2	ICSR_2	8	H'FE89	8	2
IIC_2	ICRES_2	8	H'FE8A	8	2
IIC_2	ICXR_2	8	H'FE8C	8	2
IIC_2	ICDR_2	8	H'FE8E	8	2
IIC_2	SARX_2	8	H'FE8E	8	2

Module	Register Abbreviation	Number of Bits	Address	Data Bus Width	Access States
IIC_2	ICMR_2	8	H'FE8F	8	2
IIC_2	SAR_2	8	H'FE8F	8	2
IIC_0	ICRES_0	8	H'FEE6	8	2
PS2_0	KBCR1_0	8	H'FEC0	8	2
PS2_0	KBTR_0	8	H'FEC1	8	2
PS2_0	KBCRH_0	8	H'FED8	8	2
PS2_0	KBCRL_0	8	H'FED9	8	2
PS2_0	KBBR_0	8	H'FEDA	8	2
PS2_0	KBCR2_0	8	H'FEDB	8	2
PS2_1	KBCR1_1	8	H'FEC2	8	2
PS2_1	KBTR_1	8	H'FEC3	8	2
PS2_1	KBCRH_1	8	H'FEDC	8	2
PS2_1	KBCRL_1	8	H'FEDD	8	2
PS2_1	KBBR_1	8	H'FEDE	8	2
PS2_1	KBCR2_1	8	H'FEDF	8	2
PS2_2	KBCR1_2	8	H'FEC4	8	2
PS2_2	KBTR_2	8	H'FEC5	8	2
PS2_2	KBCRH_2	8	H'FEE0	8	2
PS2_2	KBCRL_2	8	H'FEE1	8	2
PS2_2	KBBR_2	8	H'FEE2	8	2
PS2_2	KBCR2_2	8	H'FEE3	8	2
PS2_3	KBCR1_3	8	H'FED2	8	2
PS2_3	KBTR_3	8	H'FED3	8	2
PS2_3	KBCRH_3	8	H'FFE0	8	2
PS2_3	KBCRL_3	8	H'FFE1	8	2
PS2_3	KBBR_3	8	H'FFE2	8	2
PS2_3	KBCR2_3	8	H'FFE3	8	2
LPC	LADR1H	8	H'FDC0	8	2
LPC	LADR1L	8	H'FDC1	8	2
LPC	LADR2H	8	H'FDC2	8	2
LPC	LADR2L	8	H'FDC3	8	2

Module	Register Abbreviation	Number of Bits	Address	Data Bus Width	Access States
LPC	SCIFADRH	8	H'FDC4	8	2
LPC	SCIFADRL	8	H'FDC5	8	2
LPC	LADR4H	8	H'FDD4	8	2
LPC	LADR4L	8	H'FDD5	8	2
LPC	IDR4	8	H'FDD6	8	2
LPC	ODR4	8	H'FDD7	8	2
LPC	STR4	8	H'FDD8	8	2
LPC	HICR4	8	H'FDD9	8	2
LPC	SIRQCR2	8	H'FDDA	8	2
LPC	SIRQCR3	8	H'FDDB	8	2
LPC	TWR0MW	8	H'FE20	8	2
LPC	TWR0SW	8	H'FE20	8	2
LPC	TWR1	8	H'FE21	8	2
LPC	TWR2	8	H'FE22	8	2
LPC	TWR3	8	H'FE23	8	2
LPC	TWR4	8	H'FE24	8	2
LPC	TWR5	8	H'FE25	8	2
LPC	TWR6	8	H'FE26	8	2
LPC	TWR7	8	H'FE27	8	2
LPC	TWR8	8	H'FE28	8	2
LPC	TWR9	8	H'FE29	8	2
LPC	TWR10	8	H'FE2A	8	2
LPC	TWR11	8	H'FE2B	8	2
LPC	TWR12	8	H'FE2C	8	2
LPC	TWR13	8	H'FE2D	8	2
LPC	TWR14	8	H'FE2E	8	2
LPC	TWR15	8	H'FE2F	8	2
LPC	IDR3	8	H'FE30	8	2
LPC	ODR3	8	H'FE31	8	2
LPC	STR3	8	H'FE32	8	2
LPC	HICR5	8	H'FE33	8	2

Module	Register Abbreviation	Number of Bits	Address	Data Bus Width	Access States
LPC	LADR3H	8	H'FE34	8	2
LPC	LADR3L	8	H'FE35	8	2
LPC	SIRQCR0	8	H'FE36	8	2
LPC	SIRQCR1	8	H'FE37	8	2
LPC	IDR1	8	H'FE38	8	2
LPC	ODR1	8	H'FE39	8	2
LPC	STR1	8	H'FE3A	8	2
LPC	SIRQCR4	8	H'FE3B	8	2
LPC	IDR2	8	H'FE3C	8	2
LPC	ODR2	8	H'FE3D	8	2
LPC	STR2	8	H'FE3E	8	2
LPC	HISEL	8	H'FE3F	8	2
LPC	HICR0	8	H'FE40	8	2
LPC	HICR1	8	H'FE41	8	2
LPC	HICR2	8	H'FE42	8	2
LPC	HICR3	8	H'FE43	8	2
A/D converter	ADDRA	16	H'FC00	16	2
A/D converter	ADDRB	16	H'FC02	16	2
A/D converter	ADDRC	16	H'FC04	16	2
A/D converter	ADDRD	16	H'FC06	16	2
A/D converter	ADDRE	16	H'FC08	16	2
A/D converter	ADDRF	16	H'FC0A	16	2
A/D converter	ADDRG	16	H'FC0C	16	2
A/D converter	ADDRH	16	H'FC0E	16	2
A/D converter	ADCSR	8	H'FC10	8	2
A/D converter	ADCR	8	H'FC11	8	2
SCIF	FRBR	8	H'FC20	8	2
SCIF	FTHR	8	H'FC20	8	2
SCIF	FDLL	8	H'FC20	8	2
SCIF	FIER	8	H'FC21	8	2
SCIF	FDLH	8	H'FC21	8	2
SCIF	FIIR	8	H'FC22	8	2

Module	Register Abbreviation	Number of Bits	Address	Data Bus Width	Access States
SCIF	FFCR	8	H'FC22	8	2
SCIF	FLCR	8	H'FC23	8	2
SCIF	FMCR	8	H'FC24	8	2
SCIF	FLSR	8	H'FC25	8	2
SCIF	FMSR	8	H'FC26	8	2
SCIF	FSCR	8	H'FC27	8	2
SCIF	SCIFCR	8	H'FC28	8	2
ROM	FCCS	8	H'FEA8	8	2
ROM	FPCS	8	H'FEA9	8	2
ROM	FECS	8	H'FEAA	8	2
ROM	FKEY	8	H'FEAC	8	2
ROM	FMATS	8	H'FEAD	8	2
ROM	FTDAR	8	H'FEAE	8	2
SYSTEM	SYSCR3	8	H'FE7D	8	2
SYSTEM	MSTPCRA	8	H'FE7E	8	2
SYSTEM	MSTPCRB	8	H'FE7F	8	2
SYSTEM	SBYCR	8	H'FF84	8	2
SYSTEM	LPWRCR	8	H'FF85	8	2
SYSTEM	MSTPCRH	8	H'FF86	8	2
SYSTEM	MSTPCRL	8	H'FF87	8	2
SYSTEM	STCR	8	H'FFC3	8	2
SYSTEM	SYSCR	8	H'FFC4	8	2
SYSTEM	MDCR	8	H'FFC5	8	2

Section 28 Electrical Characteristics

28.1 Absolute Maximum Ratings

Table 28.1 lists the absolute maximum ratings.

Table 28.1 Absolute Maximum Ratings

Item	Symbol	Value	Unit
Power supply voltage*	V_{CC}	–0.3 to +4.3	V
Input voltage (except ports 7, D, A, G, I, PE4, PE2 to PE0, P97, P86, P52, and P42)	V_{in}	–0.3 to V_{CC} + 0.3	
Input voltage (ports A, G, I, PE4, PE2 to PE0, P97, P86, P52, and P42)	V_{in}	–0.3 to +7.0	
Input voltage (AN input is not selected for port D)	V_{in}	–0.3 to V_{CC} + 0.3	
Input voltage (AN input is selected for port D)	V_{in}	–0.3 to V_{CC} +0.3 or –0.3 to AV_{CC} +0.3 whichever is lower	
Input voltage (port 7)	V_{in}	–0.3 to AV_{CC} + 0.3	
Reference power supply voltage	AVref	–0.3 to AV_{CC} + 0.3	
Analog power supply voltage	AV_{CC}	–0.3 to +4.3	
Analog input voltage	V_{AN}	–0.3 to AV_{CC} + 0.3	
Operating temperature	T_{opr}	–20 to +75	°C
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 exceeded. Make sure the applied power supply does not exceed 4.3 V.

Note: * Voltage applied to the VCC pin. The VCL pin should not be applied a voltage.

28.2 DC Characteristics

Table 28.2 lists the DC characteristics. Table 28.3 lists the permissible output currents. Table 28.4 lists the bus drive characteristics.

Table 28.2 DC Characteristics (1)

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC}*[1] = 3.0 V to 3.6 V, AVref*[1] = 3.0 V to AV_{CC}, V_{SS} = AV_{SS}*[1] = 0 V

Item			Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Schmitt trigger input voltage	P67 to P60, $\overline{IRQ7}$ to $\overline{IRQ0}$, $\overline{IRQ15}$ to $\overline{IRQ8}$	(1)	V_T^-	$V_{CC} \times 0.2$	—	—	V	
	$\overline{KIN7}$ to $\overline{KIN0}$, $\overline{KIN15}$ to $\overline{KIN8}$, $\overline{WUE15}$ to $\overline{WUE8}$		V_T^+	—	—	$V_{CC} \times 0.7$		
	$\overline{ExIRQ7}$ to $\overline{ExIRQ6}$, and $\overline{ExIRQ15}$ to $\overline{ExIRQ8}$		$V_T^+ - V_T^-$	$V_{CC} \times 0.05$	—	—		
Input high voltage	$\overline{RES}$, NMI, MD2, MD1, and $\overline{ETRST}$	(2)	V_{IH}	$V_{CC} \times 0.9$	—	$V_{CC} + 0.3$		
	EXTAL			$V_{CC} \times 0.7$	—	$V_{CC} + 0.3$		
	Port 7			$V_{CC} \times 0.7$	—	$AV_{CC} + 0.3$		
	Ports A, G, I, PE4, PE2 to PE0, P97, P86, P52, and P42			$V_{CC} \times 0.7$	—	5.5		
	Input pins other than (1) and (2) above			$V_{CC} \times 0.7$	—	$V_{CC} + 0.3$		
Input low voltage	$\overline{RES}$, MD2, MD1, and $\overline{ETRST}$	(3)	V_{IL}	– 0.3	—	$V_{CC} \times 0.1$		
	NMI, EXTAL, and input pins other than (1) and (3) above			–0.3	—	$V_{CC} \times 0.2$		
Output high voltage	All output pins (except for ports A, G, I, P97, P86, P52, and P42)		V_{OH}	V_{CC} – 0.5	—	—		I_{OH} = –200 μA
				V_{CC} – 1.0	—	—		I_{OH} = –1 mA
	Ports A, G, I, P97, P86, P52, and P42*[2]			0.5	—	—		I_{OH} = - 200 μA
Output low voltage	All output pins *[3]		V_{OL}	—	—	0.4		I_{OL} = 1.6 mA
	Ports 1, 2, 3, C, and D			—	—	1.0		I_{OL} = 5 mA

Table 28.2 DC Characteristics (2)

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC}*[1] = 3.0 V to 3.6 V, AVref*[1] = 3.0 V to AV_{CC},
V_{SS} = AV_{SS}*[1] = 0 V

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input leakage current	$\overline{RES}$	$\lvert I_{in} \rvert$	—	—	10.0	µA	Vin = 0.5 to V_{CC} – 0.5 V
	NMI, MD2, MD1, $\overline{ETRST}$ PE0 to PE2, and PE4		—	—	1.0		
	Port 7		—	—	1.0		Vin = 0.5 to AV_{CC} – 0.5 V
Three-state leakage current (off state)	Ports 1 to 6, 8, 9, A to D, PE3, and F to J	$\lvert I_{TSI} \rvert$	—	—	1.0		Vin = 0.5 to V_{CC} – 0.5 V
Input pull-up MOS current	Ports 1 to 3, P95 to P90, ports 6, B to D, F, H, and J	$-I_p$	20	—	150	µA	V_{in} = 0 V
Input capacitance	All pins	C_{in}	—	—	10	pF	V_{in} = 0 V f = 1 MHz T_a = 25 °C
Supply current*[4]	Normal operation	I_{CC}	—	25	40	mA	V_{CC} = 3.0 V to 3.6 V f = 20 MHz, all modules operating, high-speed mode
	Sleep mode		—	20	35		V_{CC} = 3.0 V to 3.6 V f = 20 MHz
	Standby mode		—	35	70	µA	Ta ≤ 50 °C
			—	—	200		50 °C < Ta
Analog power supply current	During A/D conversion	AI_{CC}	—	1	2	mA	
	A/D conversion standby		—	0.01	5	µA	AV_{CC} = 3.0 V to 3.6 V
Reference power supply current	During A/D conversion	AI_{ref}	—	1	2	mA	
	A/D conversion standby		—	0.01	5	µA	AVref = 3.0 V to AV_{CC}
VCC start voltage		VCC_{START}	—	0	0.8	V	
VCC rising edge		SVCC	—	—	20	ms/V	

Notes: 1. Do not leave the AVCC, AVref, and AVSS pins open even if the A/D converter is not used.
Even if the A/D converter is not used, apply a value in the range from 3.0 V to 3.6 V to the AVCC and AVref pins by connecting to the power supply (V_{CC}). The relationship between these two pins should be AVref ≤ AV_{CC}.

2. Ports A, G, I, P97, P86, P52, P42, and peripheral module outputs multiplexed on the pin are NMOS push-pull outputs.
An external pull-up resistor is necessary to provide high-level output from SCL0, SCL1, SDA0, SDA1, SDA2, SCL2, ExSCLA, ExSCLB, ExSDAA, and ExSDAB (ICE bit in ICCR is 1).
Ports A, G, I, P97, P86, P52, and P42 (ICE bit in ICCR is 0) high levels are driven by NMOS. An external pull-up resistor is necessary to provide high-level output from these pins when they are used as an output.

3. Indicates values when ICCS = 0, ICE = 0, and KBIOE = 0. Low level output when the bus drive function is selected is rated separately.

4. Current consumption values are for V_{IH} min = V_{CC} – 0.2 V and V_{IL} max = 0.2 V with all output pins unloaded and the on-chip pull-up MOSs in the off state.

Table 28.2 DC Characteristics (3) Using LPC Function

Conditions: V_{CC} = 3.0 V to 3.6 V, V_{SS} = 0 V

Item		Symbol	Min.	Max.	Unit	Test Conditions
Input high voltage	P37 to P30, P82 to P80, PB1, PB0	V_{IH}	V_{CC} × 0.5	—	V	
Input low voltage	P37 to P30, P82 to P80, PB1, PB0	V_{IL}	—	V_{CC} × 0.3	V	
Output high voltage	P37, P33 to P30, P82 to P80, PB1, PB0	V_{OH}	V_{CC} × 0.9	—	V	I_{OH} = - 0.5 mA
Output low voltage	P37, P33 to P30, P82 to P80, PB1, PB0	V_{OL}	—	V_{CC}×0.1	V	I_{OL} = 1.5 mA

Table 28.2 DC Characteristics (4) Using FSI Function

Conditions: V_{CC} = 3.0 V to 3.6 V, V_{SS} = 0 V

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input high voltage	PB7 to PB4	V_{IH}	$V_{CC} \times 0.7$	—	V_{CC} + 0.3	V	—
Input low voltage		V_{IL}	– 0.3	—	$V_{CC} \times 0.2$		—
Output high voltage		V_{OH}	V_{CC} – 0.5	—	—		I_{OH} = –200 μA
			V_{CC} – 0.3	—	—		I_{OH} = –1 mA
Output low voltage		V_{OL}	—	—	0.4		I_{OL} = 1.6 mA
Input pull-up MOS current		$-I_P$	30	—	300	μA	V_{in} = 0 V
Input capacitance		C_{in}	—	—	10	pF	V_{in} = 0 V, f = 1 MHz, Ta = 25 °C

Table 28.3 Permissible Output Currents

Conditions: V_{CC} = 3.0 V to 3.6 V, V_{SS} = 0V

Item		Symbol	Min.	Typ.	Max.	Unit
Permissible output low current (per pin)	SCL0, SDA0, SCL1, SDA1, SCL2, SDA2, ExSCLA, ExSDAA, ExSCLB, ExSDAB, PS2AC to PS2DC, PS2AD to S2DD, and PA7 to PA4 (bus drive function selected)	I_{OL}	—	—	8	mA
	Ports 1, 2, 3, C, and D		—	—	5	
	Other output pins		—	—	2	
Permissible output low current (total)	Total of ports 1, 2, 3, C, and D	ΣI_{OL}	—	—	40	
	Total of all output pins, including the above		—	—	60	
Permissible output high current (per pin)	All output pins	$-I_{OH}$	—	—	2	
Permissible output high current (total)	Total of all output pins	$\Sigma - I_{OH}$	—	—	30	

Notes: 1. To protect LSI reliability, do not exceed the output current values in table 28.3.
2. When driving a Darlington transistor or LED, always insert a current-limiting resistor in the output line, as show in figures 28.1 and 28.2.

Table 28.4 Bus Drive Characteristics

Conditions: V_{CC} = 3.0 V to 3.6V, V_{SS} = 0 V

Applicable Pins: SCL0, SDA0, SCL1, SDA1, SCL2, SDA2, ExSCLA, ExSDAA, ExSCLB, and ExSDAB (bus drive function selected)

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Schmitt trigger input voltage	V_T^-	$V_{CC} \times 0.3$	—	—	V	
	V_T^+	—	—	$V_{CC} \times 0.7$		
	$V_T^+ - V_T^-$	$V_{CC} \times 0.05$	—	—		
Input high voltage	V_{IH}	$V_{CC} \times 0.7$	—	5.5		
Input low voltage	V_{IL}	–0.5	—	$V_{CC} \times 0.3$		
Output low voltage	V_{OL}	—	—	0.5		I_{OL} = 8 mA
		—	—	0.4		I_{OL} = 3 mA
Input capacitance	C_{in}	—	—	10	pF	V_{in} = 0 V, f = 1 MHz, Ta = 25 °C
Three-state leakage current (off state)	$\lvert I_{TSI} \rvert$	—	—	1.0	μA	V_{in} = 0.5 to V_{CC} – 0.5 V

Conditions: V_{CC} = 3.0 V to 3.6V, V_{SS} = 0 V

Applicable Pins: PS2AC to PS2DC, PS2AD to PS2DD, and PA7 to PA4 (bus drive function selected)

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Output low voltage	V_{OL}	—	—	0.5	V	I_{OL} = 8 mA
		—	—	0.4		I_{OL} = 3 mA

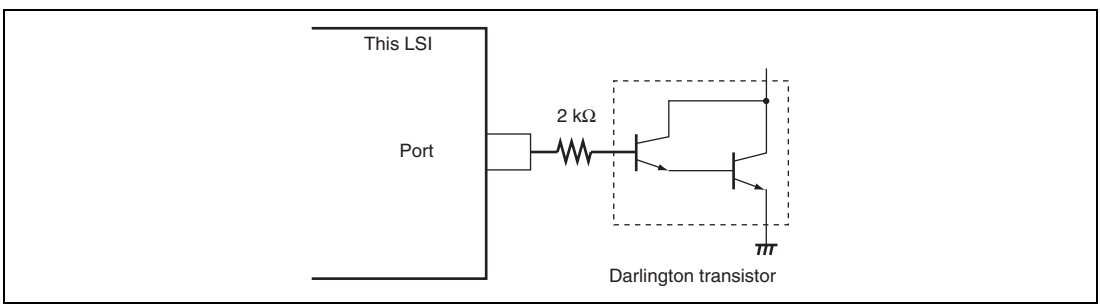


Figure 28.1 Darlington Transistor Drive Circuit (Example)

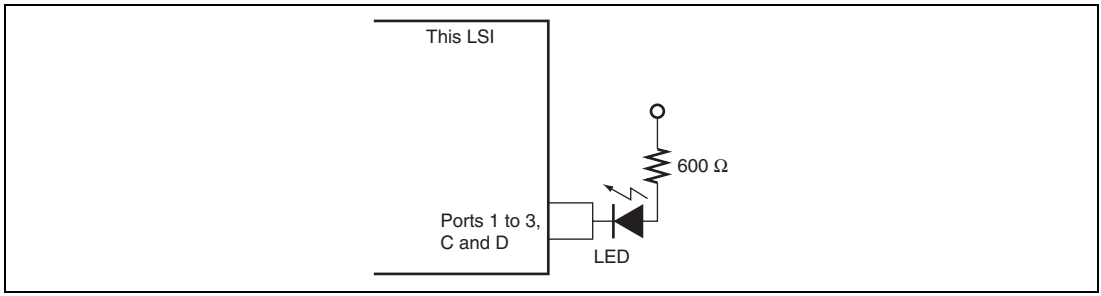


Figure 28.2 LED Drive Circuit (Example)

28.3 AC Characteristics

Figure 28.3 shows the test conditions for the AC characteristics.

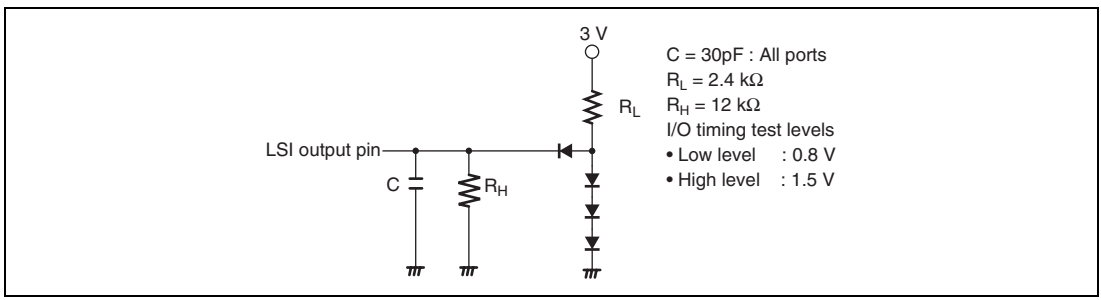


Figure 28.3 Output Load Circuit

28.3.1 Clock Timing

Table 28.5 shows the clock timing. The clock timing specified here covers clock output (ϕ) and clock pulse generator (crystal) and external clock input (EXTAL pin) oscillation stabilization times. For details of external clock input (EXTAL pin and EXCL pin) timing, see section 25, Clock Pulse Generator.

Table 28.5 Clock Timing

Condition A: V_{CC} = 3.0 V to 3.6 V, V_{SS} = 0 V, ϕ = 8 MHz to 10 MHz

Condition B: V_{CC} = 3.0 V to 3.6 V, V_{SS} = 0 V, ϕ = 10 MHz to 20 MHz

		Condition A		Condition B			
Item	**Symbol**	**Min.**	**Max.**	**Min.**	**Max.**	**Unit**	**Reference**
Clock cycle time	t_{cyc}	100	125	50	100	ns	Figure 28.4
Clock high pulse width	t_{CH}	30	—	20	—		
Clock low pulse width	t_{CL}	30	—	20	—		
Clock rise time	t_{Cr}	—	20	—	5		
Clock fall time	t_{Cf}	—	20	—	5		
Reset oscillation stabilization (crystal)	t_{OSC1}	20	—	20	—	ms	Figure 28.5
Software standby oscillation stabilization time (crystal)	t_{OSC2}	8	—	8	—		Figure 28.6
External clock output stabilization delay time	t_{DEXT}	500	—	500	—	µs	Figure 28.5

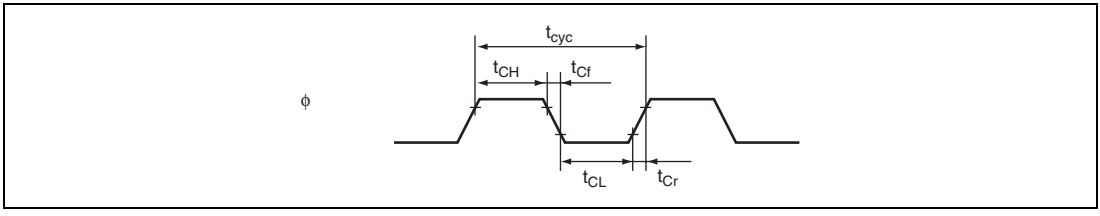


Figure 28.4 System Clock Timing

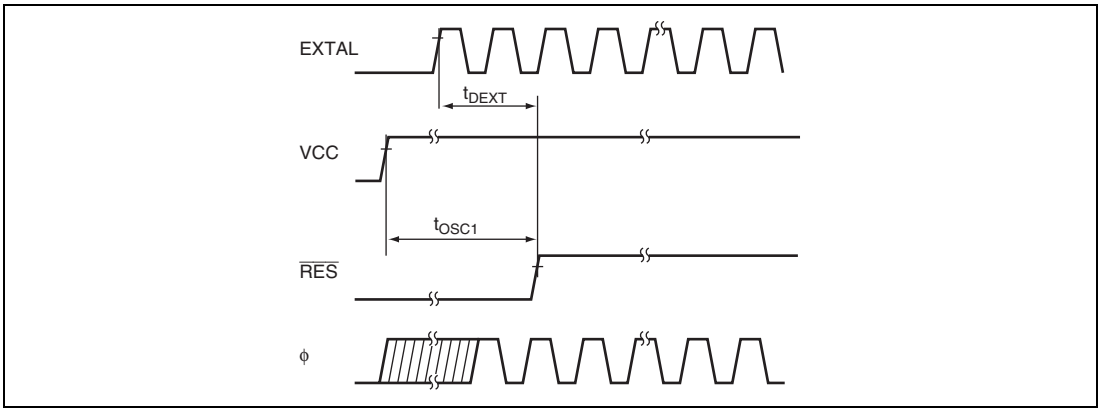


Figure 28.5 Oscillation Stabilization Timing

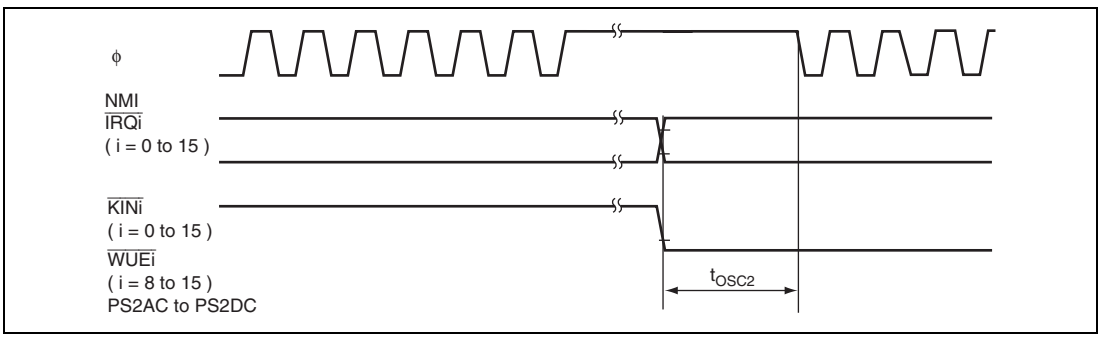


Figure 28.6 Oscillation Stabilization Timing (Exiting Software Standby Mode)

28.3.2 Control Signal Timing

Table 28.6 shows the control signal timing. Only external interrupts NMI, IRQ0 to IRQ15, KIN0 to KIN15, WUE8 to WUE15, and KBCA to KBCD can be operated based on the subclock (ϕ = 32.768 kHz).

Table 28.6 Control Signal Timing

Conditions: V_{CC} = 3.0 V to 3.6 V, V_{SS} = 0 V, ϕ = 32.768 kHz, 8 MHz to maximum operating frequency

Item	Symbol	Min.	Max.	Unit	Test Conditions
$\overline{\text{RES}}$ setup time	t_{RESS}	200	—	ns	Figure 28.7
$\overline{\text{RES}}$ pulse width	t_{RESW}	20	—	t_{cyc}	
NMI setup time	t_{NMIS}	150	—	ns	Figure 28.8
NMI hold time	t_{NMIH}	10	—		
NMI pulse width (exiting software standby mode)	t_{NMIW}	200	—		
IRQ setup time ($\overline{\text{IRQ15}}$ to $\overline{\text{IRQ0}}$, $\overline{\text{KIN15}}$ to $\overline{\text{KIN0}}$, $\overline{\text{WUE15}}$ to $\overline{\text{WUE8}}$)	t_{IRQS}	150	—		
IRQ hold time ($\overline{\text{IRQ15}}$ to $\overline{\text{IRQ0}}$, $\overline{\text{KIN15}}$ to $\overline{\text{KIN0}}$, $\overline{\text{WUE15}}$ to $\overline{\text{WUE8}}$)	t_{IRQH}	10	—		
IRQ pulse width ($\overline{\text{IRQ15}}$ to $\overline{\text{IRQ0}}$, $\overline{\text{KIN15}}$ to $\overline{\text{KIN0}}$, $\overline{\text{WUE15}}$ to $\overline{\text{WUE8}}$) (exiting software standby mode)	t_{IRQW}	200	—		

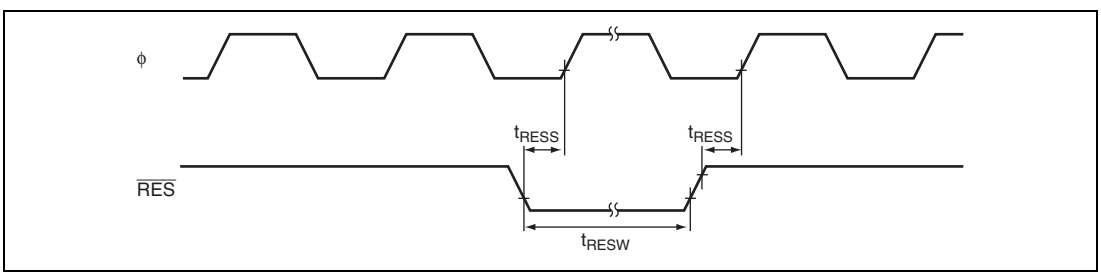


Figure 28.7 Reset Input Timing

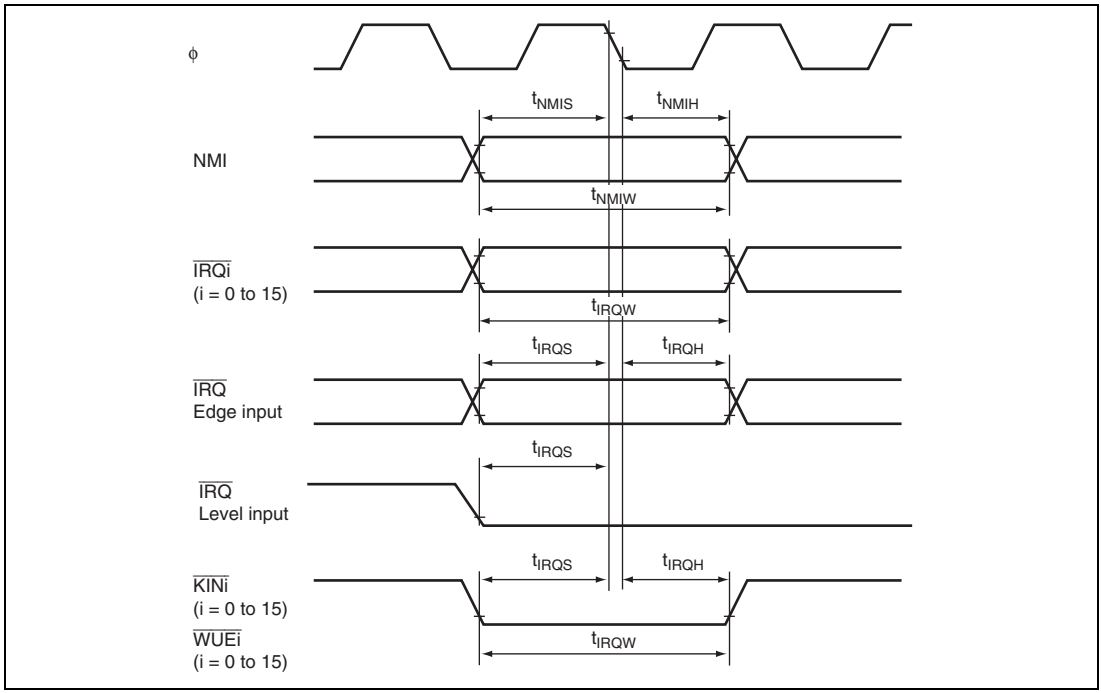


Figure 28.8 Interrupt Input Timing

28.3.3 Timing of On-Chip Peripheral Modules

Table 28.7 shows the on-chip peripheral module timing. The on-chip peripheral modules that can be operated by the subclock (ϕ = 32.768 kHz) are I/O ports, external interrupts (NMI, IRQ0 to IRQ15, KIN0 to KIN15, WUE8 to WUE15, and KBCA to KBCD) and watchdog timer (WDT_1) only. The system clock or LCLK operation can be used in the FSI.

Table 28.7 Timing of On-Chip Peripheral Modules

Conditions: V_{CC} = 3.0 V to 3.6 V, V_{SS} = 0 V, ϕ = 32.768 kHz*[1], ϕ = 8 MHz to maximum operating frequency, FSICK = 8 MHz to maximum operating frequency or LCLK (33 MHz)

Item			Symbol	Min.	Max.	Unit	Test Conditions
I/O ports	Output data delay time*[2]		t_{PWD}	—	50	ns	Figure 28.9
	Input data setup time		t_{PRS}	30	—		
	Input data hold time		t_{PRH}	30	—		
TPU	Timer output delay time		t_{TOCD}	—	50	ns	Figure 28.10
	Timer input setup time		t_{TICS}	30	—		
	Timer clock input setup time		t_{TCKS}	30	—		Figure 28.11
	Timer clock pulse width	Single edge	t_{TCKWH}	1.5	—	t_{cyc}	
		Both edges	t_{TCKWL}	2.5	—		
TMR	Timer output delay time		t_{TMOD}	—	50	ns	Figure 28.12
	Timer reset input setup time		t_{TMRS}	30	—		Figure 28.14
	Timer clock input setup time		t_{TMCS}	30	—		Figure 28.13
	Timer clock pulse width	Single edge	t_{TMCWH}	1.5	—	t_{cyc}	
		Both edges	t_{TMCWL}	2.5	—		
TCM	TCM input setup time		t_{TCMS}	30	—	ns	Figure 28.15
	TCM clock input setup time		t_{TCMCKS}	30	—		Figure 28.16
	TCM clock pulse width		t_{TCMCKW}	1.5	—	t_{cyc}	
TDP	TDP input setup time		t_{TDPS}	30	—	ns	Figure 28.17
	TDP clock input setup time		t_{TDPCKS}	30	—		Figure 28.18
	TDP clock pulse width		t_{TDPCKW}	1.5	—	t_{cyc}	
PWMU, PWMX	Pulse output delay time		t_{PWOD}	—	50	ns	Figure 28.19
SCI	Input clock cycle	Asynchronous	t_{Scyc}	4	—	t_{cyc}	Figure 28.20
		Synchronous		6	—		
	Input clock pulse width		t_{SCKW}	0.4	0.6	t_{Scyc}	

Item		Symbol	Min.	Max.	Unit	Test Conditions
SCI	Input clock rise time	t_{SCKr}	—	1.5	t_{cyc}	Figure 28.20
	Input clock fall time	t_{SCKf}	—	1.5		
	Transmit data delay time (synchronous)	t_{TXD}	—	50	ns	Figure 28.21
	Receive data setup time (synchronous)	t_{RXS}	50	—		
	Receive data hold time (synchronous)	t_{RXH}	50	—		
FSI	Clock cycle	t_{CYC}	30	—	ns	Figure 28.22
	Clock pulse width (high)	t_{CKH}	13	—		
	Clock pulse width (low)	t_{CKL}	13	—		
	SS signal rise delay time	t_{SSH}	12	—		
	SS signal fall delay time	t_{SSL}	12	—		
	Transmit signal delay time	t_{TXD}	—	12		
	Receive signal setup time	t_{RXS}	5	—		
	Receive signal hold time	t_{RXH}	5	—		

Notes: 1. Applied only for the peripheral modules that are available during subclock operation.
2. Other than P52, P97, P86, P42, port A, port G, and port I.

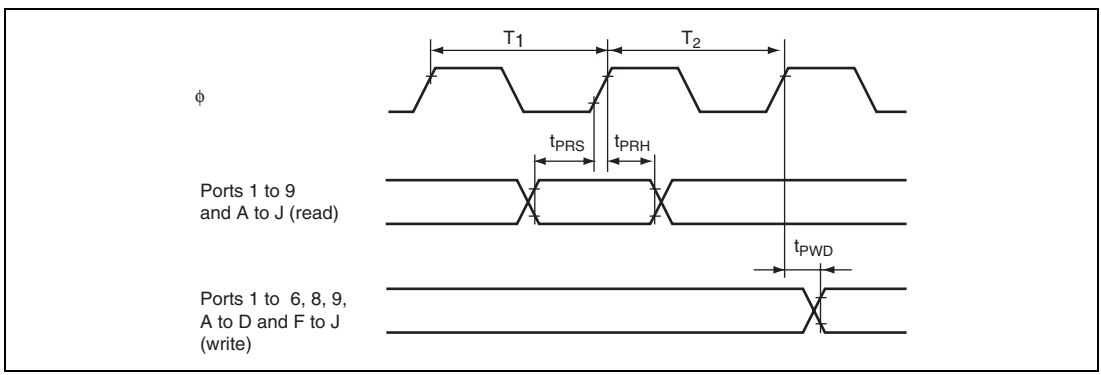


Figure 28.9 I/O Port Input/Output Timing

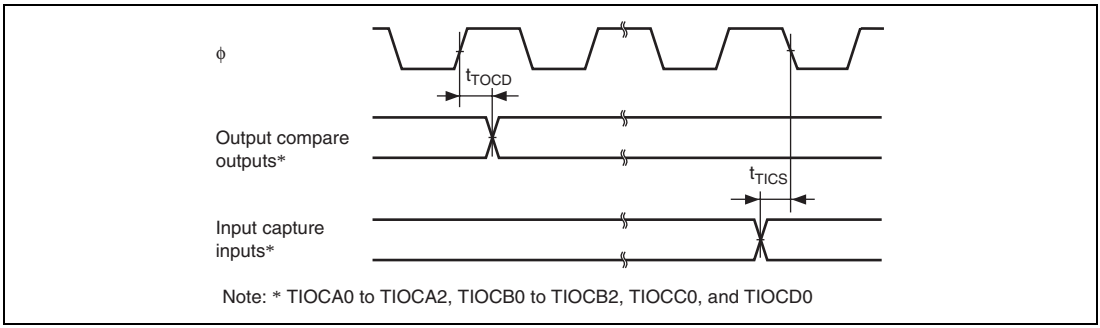


Figure 28.10 TPU Input/Output Timing

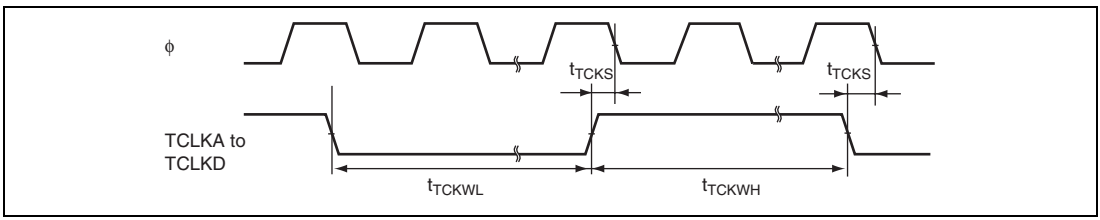


Figure 28.11 TPU Clock Input Timing

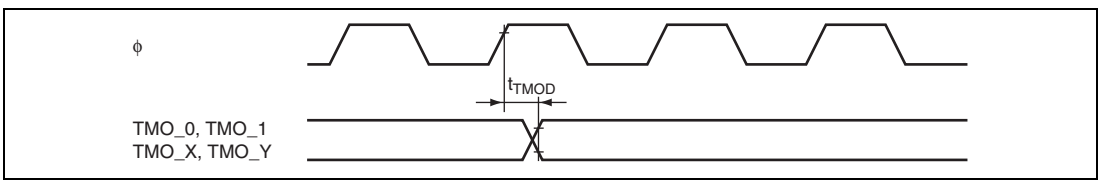


Figure 28.12 8-Bit Timer Output Timing

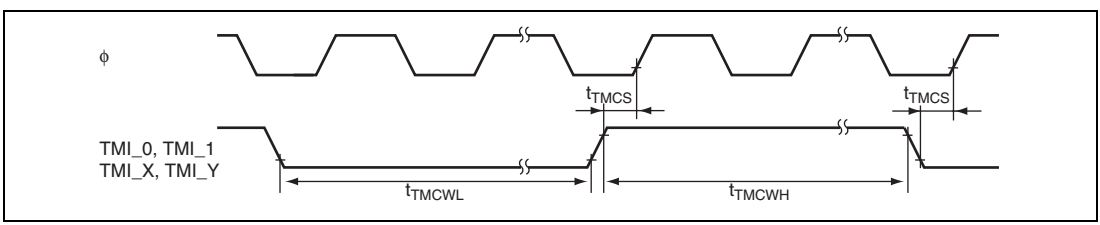


Figure 28.13 8-Bit Timer Clock Input Timing

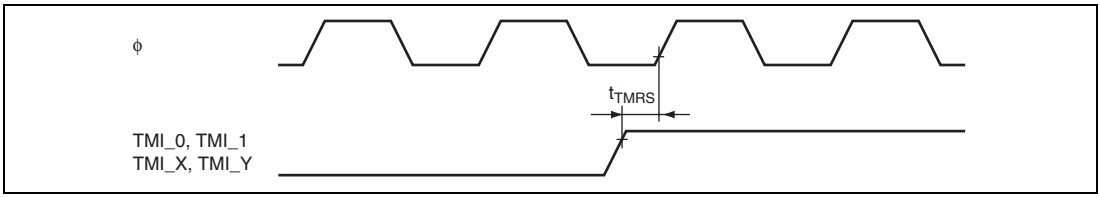


Figure 28.14 8-Bit Timer Reset Input Timing

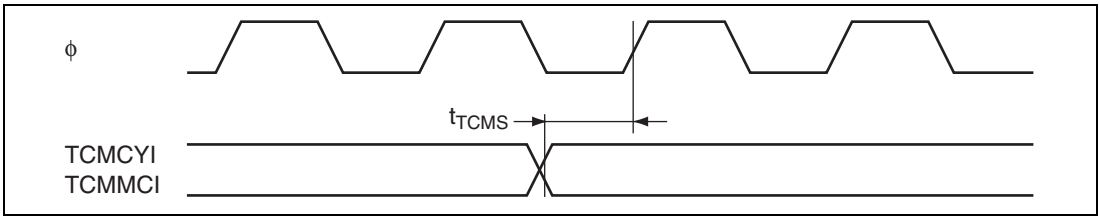


Figure 28.15 TCM Input Setup Time

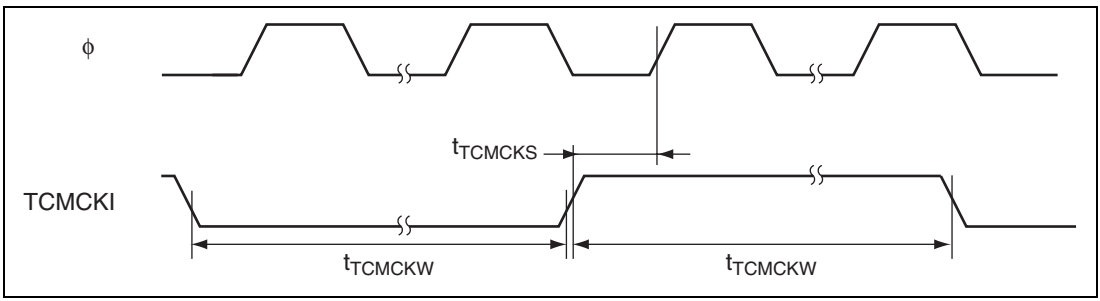


Figure 28.16 TCM Clock Input Timing

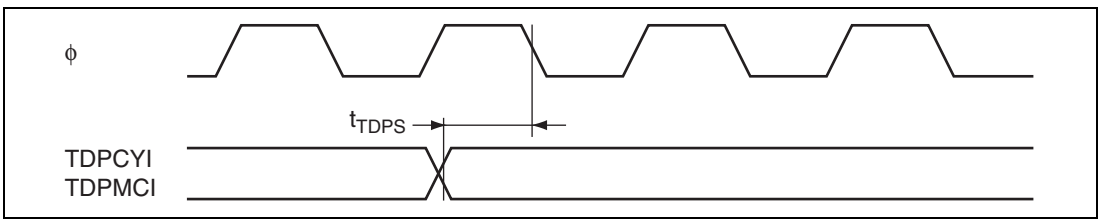


Figure 28.17 TDP Input Setup Time

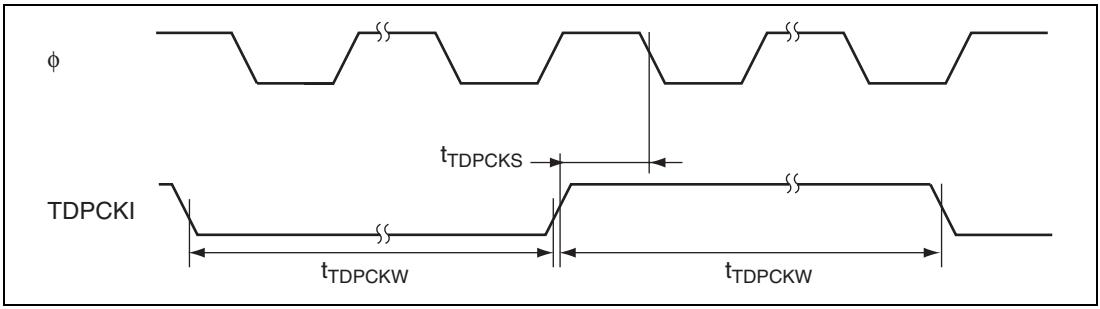


Figure 28.18 TDP Clock Input Timing

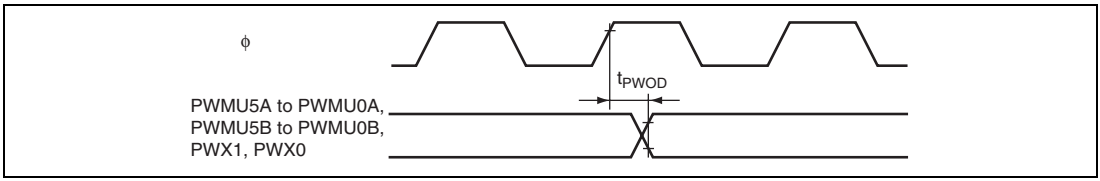


Figure 28.19 PWMU, PWMX Output Timing

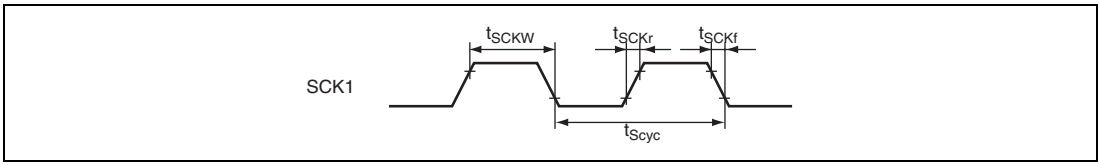


Figure 28.20 SCK Clock Input Timing

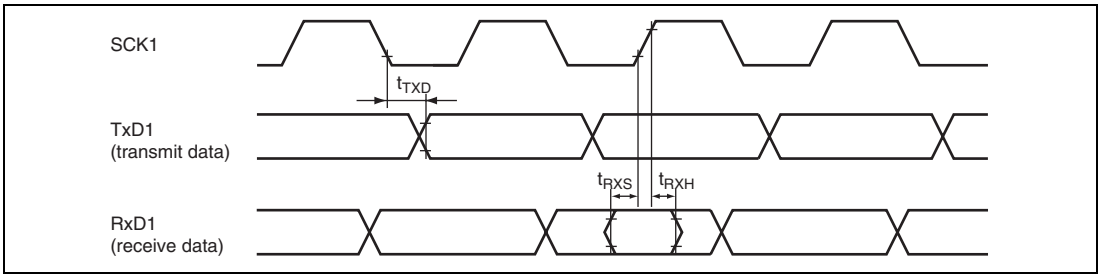


Figure 28.21 SCI Input/Output Timing (Clock Synchronous Mode)

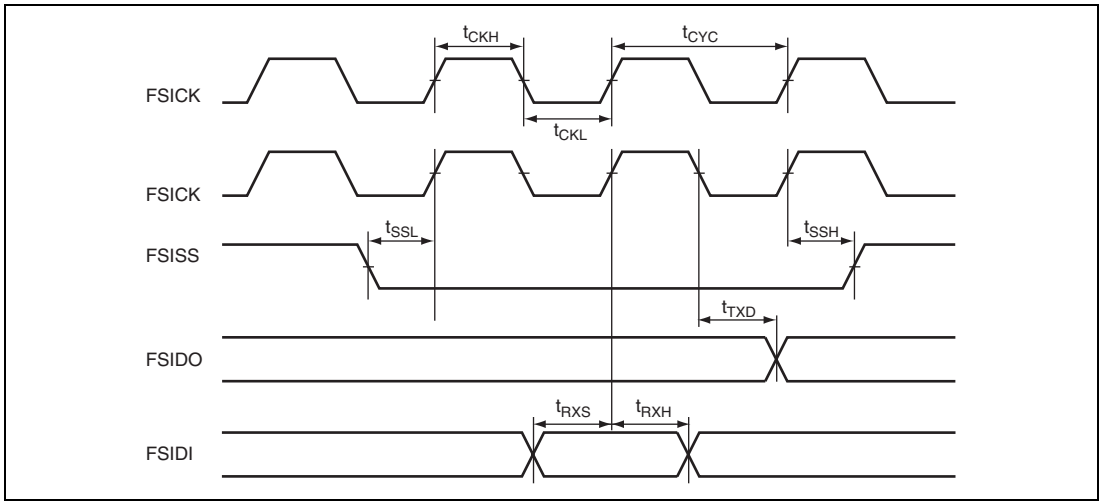


Figure 28.22 FSI Input/Output Timing

Table 28.8 PS2 Timing

Conditions: V_{CC} = 3.0 V to 3.6 V, V_{SS} = 0 V, ϕ = 8 MHz to maximum operating frequency

Item	Symbol	Standard Value			Unit	Test Conditions	Remarks
		Min.	Typ.	Max.			
KCLK, KD output fall time	t_{KBF}	—	—	250	ns		Figure 28.23
KCLK, KD input data hold time	t_{KBIH}	150	—	—			
KCLK, KD input data setup time	t_{KBIS}	150	—	—			
KCLK, KD output delay time	t_{KBOD}	—	—	450			
KCLK, KD capacitive load	C_b	—	—	400	pF		

Note: * When KCLK and KD are output, an external pull-up register must be connected, as shown in figure 28.23.

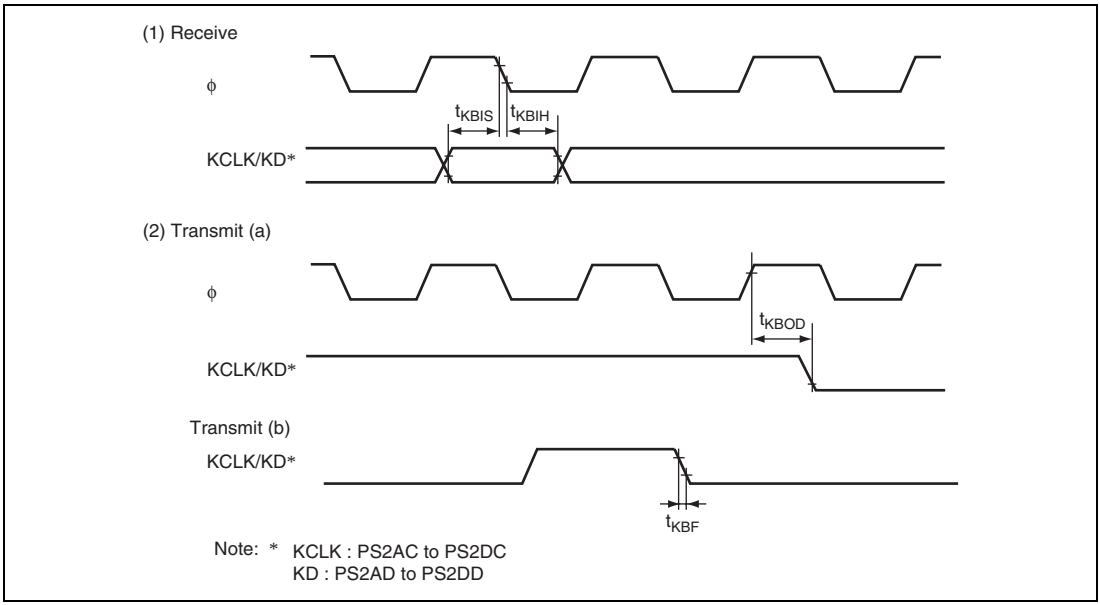


Figure 28.23 PS2 Timing

Table 28.9 I²C Bus Timing

Conditions: V_{CC} = 3.0 V to 3.6 V, V_{SS} = 0 V, ϕ = 8 MHz to maximum operating frequency

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
SCL input cycle time	t_{SCL}	12	—	—	t_{cyc}	Figure 28.24
SCL input high pulse width	t_{SCLH}	3	—	—		
SCL input low pulse width	t_{SCLL}	5	—	—		
SCL, SDA input rise time	t_{Sr}	—	—	7.5*		
SCL, SDA input fall time	t_{Sf}	—	—	300	ns	
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} can be set according to the clock selected for use by the I²C module.

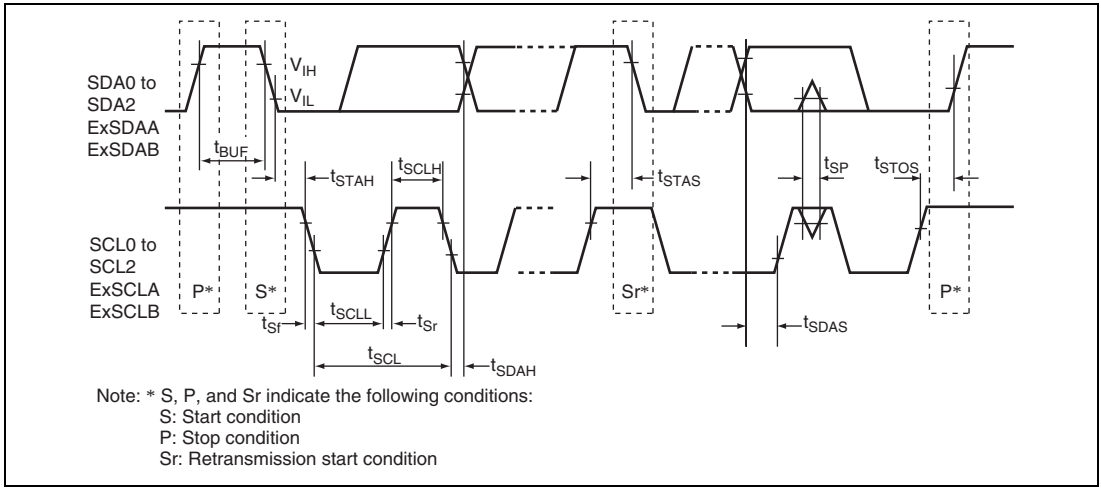


Figure 28.24 I²C Bus Interface Input/Output Timing

Table 28.10 LPC Timing

Conditions: V_{CC} = 3.0 V to 3.6V, V_{SS} = 0 V, ϕ = 8 MHz to maximum operating frequency, Ta = –20 to +75°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input clock cycle	t_{Lcyc}	30	—	—	ns	Figure 28.25
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	—	—		

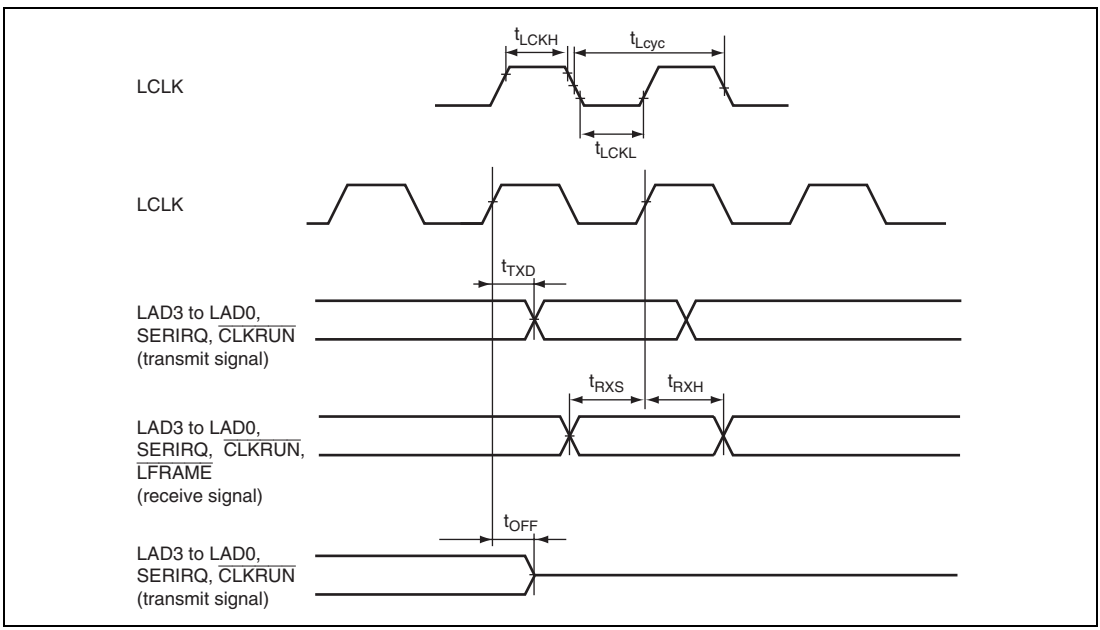


Figure 28.25 LPC Interface Timing

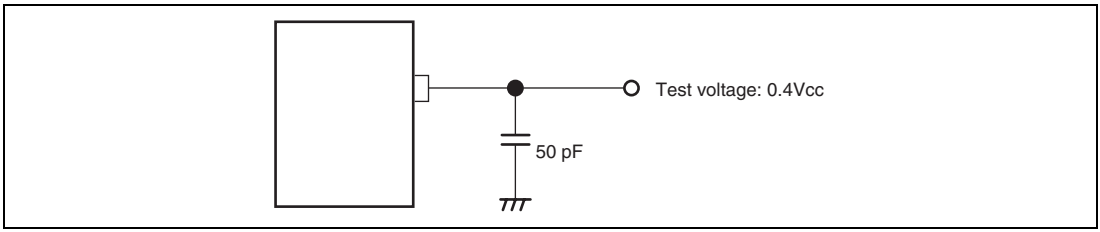


Figure 28.26 Test Conditions for Tester

Table 28.11 JTAG Timing

Conditions: V_{CC} = 3.0 V to 3.6 V, V_{SS} = 0 V, ϕ = 8 MHz to 20 MHz

Item	Symbol	Min.	Max.	Unit	Test Conditions
ETCK clock cycle time	t_{TCKcyc}	50*	125*	ns	Figure 28.27
ETCK clock high pulse width	t_{TCKH}	20	—		
ETCK clock low pulse width	t_{TCKL}	20	—		
ETCK clock rise time	t_{TCKr}	—	5		
ETCK clock fall time	t_{TCKf}	—	5		
$\overline{\text{ETRST}}$ pulse width	t_{TRSTW}	20	—	t_{cyc}	Figure 28.28
Reset hold transition pulse width	t_{RSTHW}	3	—		
ETMS setup time	t_{TMSS}	20	—	ns	Figure 28.29
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 $t_{cyc} \leq t_{TCKcyc}$

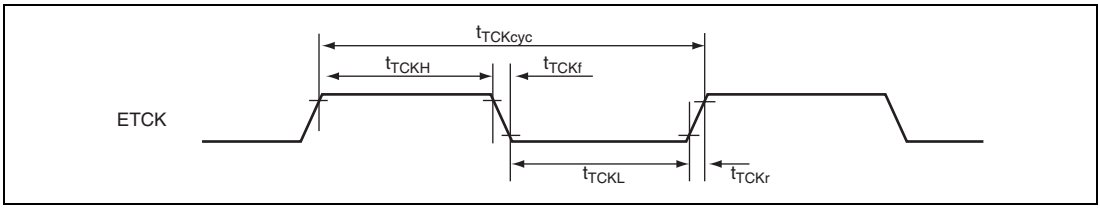


Figure 28.27 JTAG ETCK Timing

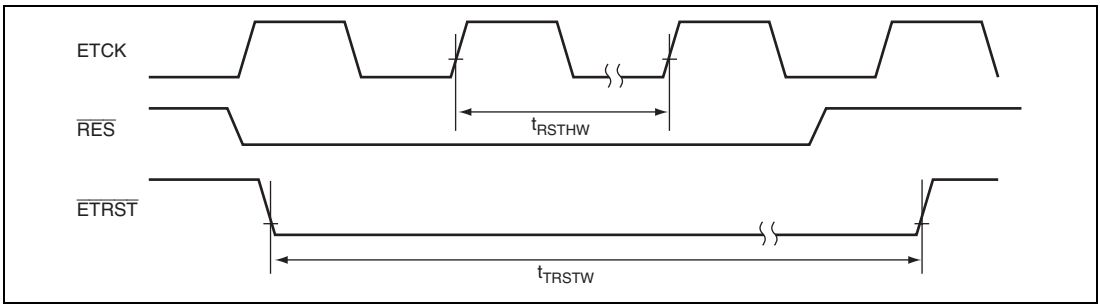


Figure 28.28 Reset Hold Timing

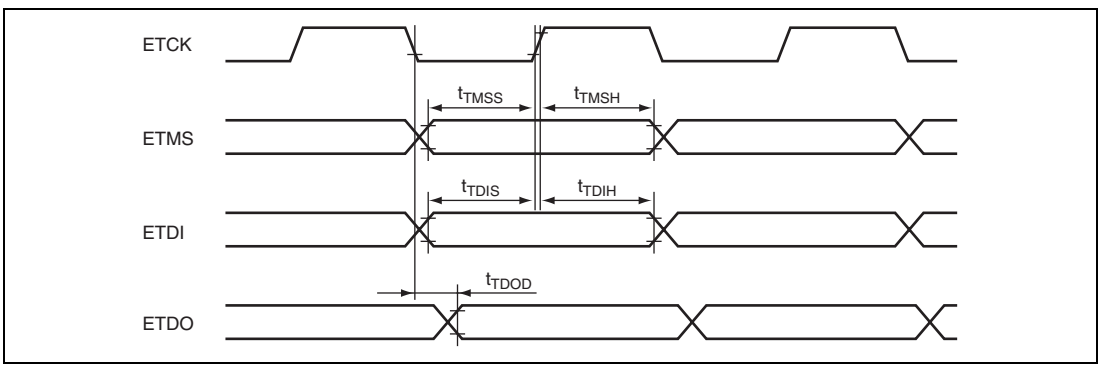


Figure 28.29 JTAG Input/Output Timing

28.4 A/D Conversion Characteristics

Table 28.12 lists the A/D conversion characteristics.

Table 28.12 A/D Conversion Characteristics (AN15 to AN0 Input: 134/266-State Conversion)

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, AVref = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 8 MHz to 20 MHz

	Condition			
Item	**Min.**	**Typ.**	**Max.**	**Unit**
Resolution	10			Bits
Conversion time	—	—	4.0*	µs
Analog input capacitance	—	—	20	pF
Permissible signal-source impedance	—	—	5	kΩ
Nonlinearity error	—	—	±7.0	LSB
Offset error	—	—	±7.5	
Full-scale error	—	—	±7.5	
Quantization error	—	—	±0.5	
Absolute accuracy	—	—	±8.0	

Note: The power supply to Avref must either be made simultaneously with or follow the power supply to Avcc.

* Value when using the maximum operating frequency of 40 states (ADCLK = 10 MHz).

28.5 Flash Memory Characteristics

Table 28.13 lists the flash memory characteristics.

Table 28.13 Flash Memory Characteristics

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, Avref = 3.0 V to AV_{CC},
V_{SS} = AV_{SS} = 0 V
Ta = 0°C to +75°C (operating temperature range for programming/erasing)

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
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	Σ_{tP}	—	1.4	4	s/160 Kbytes	Ta = 25 °C
Erase time (total)*1*2*4	Σ_{tE}	—	1.4	4	s/160 Kbytes	Ta = 25 °C
Programming and Erase time (total)*1*2*4	Σ_{tPE}	—	2.9	8	s/160 Kbytes	Ta = 25 °C
Reprogramming count	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 ranges from 1 to the minimum number.)
4. This value indicates the characteristics while the flash memory is reprogrammed within the specified range (including the minimum number).

28.6 Usage Notes

It is necessary to connect a bypass capacitor between the VCC pin and VSS pin, and a capacitor between the VCL pin and VSS pin for stable internal step-down power. An example of connection is shown in figure 28.30.

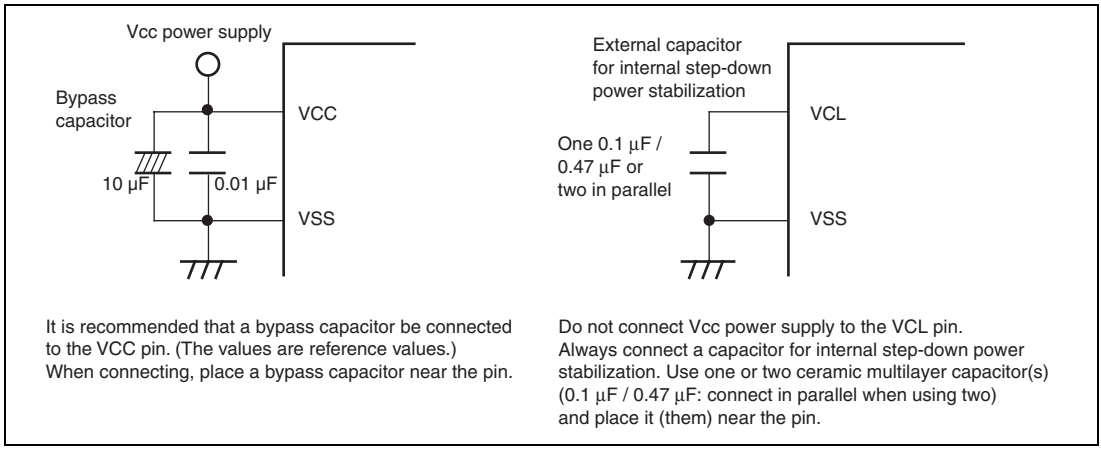


Figure 28.30 Connection of VCC and VCL Capacitors

Appendix

A. I/O Port States in Each Pin State

Table A.1 I/O Port States in Each Pin State

Port Name Pin Name	Reset	Software Standby Mode	Watch Mode	Sleep Mode	Program Execution State
Port 1	T	keep	keep	keep	I/O port
Port 2	T	keep	keep	keep	I/O port
Port 3	T	keep	keep	keep	I/O port
Port 4	T	keep	keep	keep	I/O port
Ports 52 to 50	T	keep	keep	keep	I/O port
Port 6	T	keep	keep	keep	I/O port
Ports 7 and E4 to E1	T	T	T	T	Input port
Port 8	T	keep	keep	keep	I/O port
Port 97	T	keep	keep	keep	I/O port
Port 96 ϕ, EXCL	T	[DDR = 1]H [DDR = 0]T	EXCL input/ keep	[DDR = 1] Clock output [DDR = 0]T	Clock output/ EXCL input/ Input port
Ports 95 to 90	T	keep	keep	keep	I/O port
Ports A to D, F, G, and H5 to H0	T	keep	keep	keep	I/O port
Port E0	T	T	ExEXCL input/T	T	ExEXCL input/ input port
Port I	T	keep	keep	keep	I/O port
Port J	T	keep	keep	keep	I/O port

[Legend]

H: High level

L: Low level

T: High impedance

keep: Input ports are in the high-impedance state (when DDR = 0 and PCR = 1, the input pull-up MOS remains on).

Output ports maintain their previous state.

Depending on the pins, the on-chip peripheral modules may be initialized and the I/O port function determined by DDR and DR.

DDR: Data direction register

B. Product Lineup

Product Type		Type Code	Mark Code	Package (Code)
H8S/2117R	Flash memory version	R4F2117R	F2117RTE20V	PTQP0144LC-A (TFP-144V)
			F2117RBG20V	PLBG0176GA-A (BP-176V)
			F2117RLP20V	PTLG0145JB-A (TLP-145V)

C. Package Dimensions

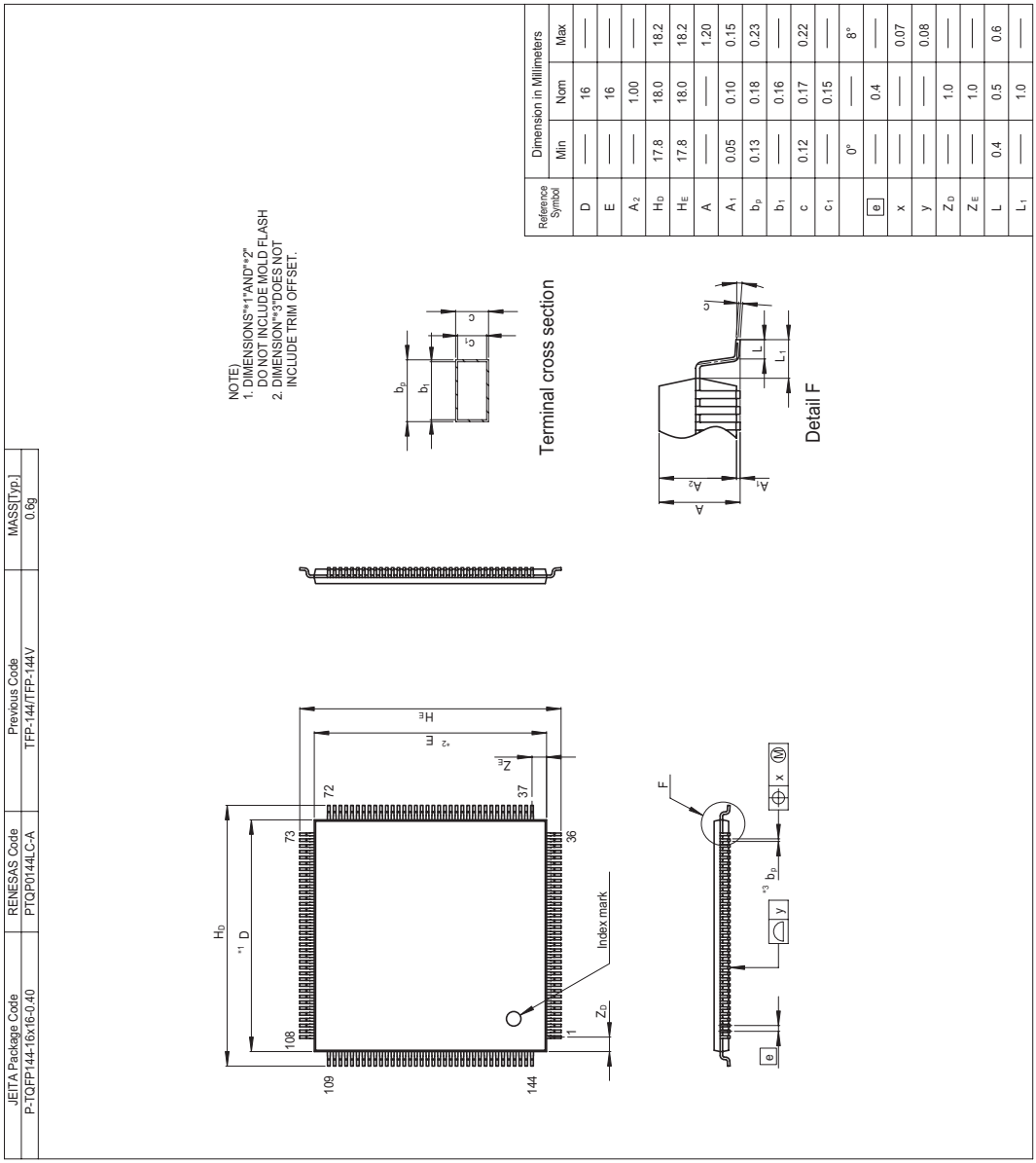
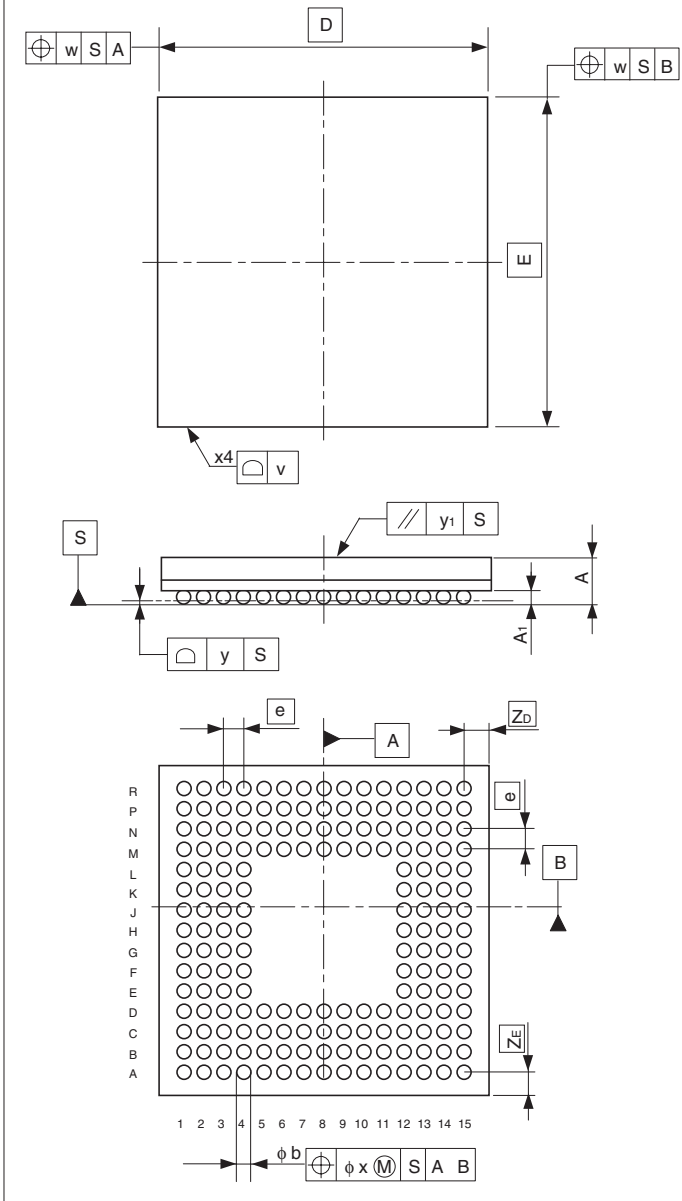


Figure C.1 Package Dimensions (TFP-144V)

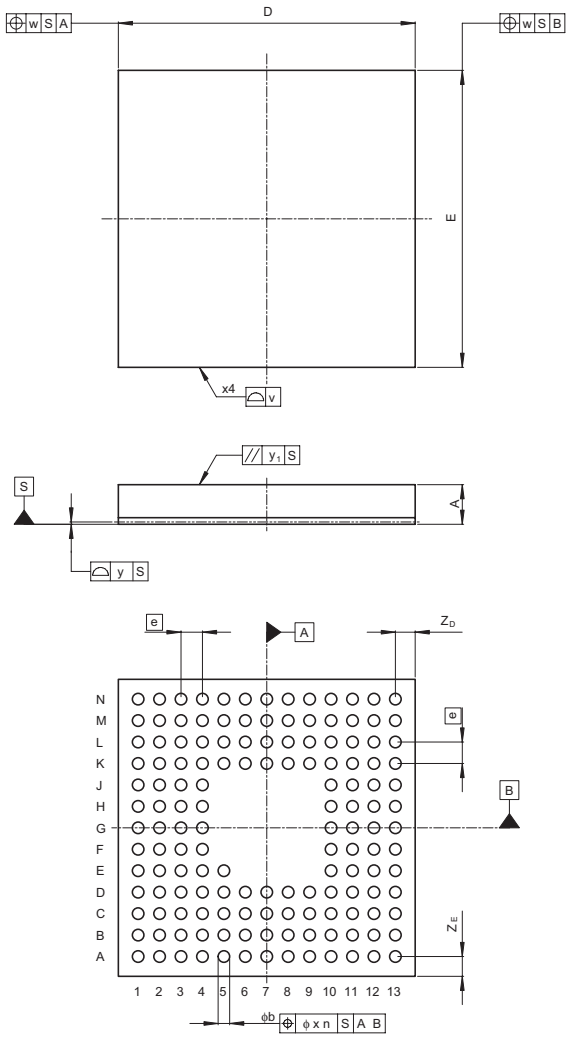
JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LFBGA176-13x13-0.80	PLBG0176GA-A	BP-176/BP-176V	0.45g



Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	—	13.0	—
E	—	13.0	—
v	—	—	0.15
w	—	—	0.20
A	—	—	1.40
A_1	0.35	0.40	0.45
e	—	0.80	—
b	0.45	0.50	0.55
x	—	—	0.08
y	—	—	0.10
y_1	—	—	0.2
S_D	—	—	—
S_E	—	—	—
Z_D	—	0.90	—
Z_E	—	0.90	—

Figure C.2 Package Dimensions (BP-176V)

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-TFLGA145-9x9-0.65	PTLG0145JB-A	-	0.15g



Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	—	9.0	—
E	—	9.0	—
v	—	—	0.15
w	—	—	0.20
A	—	—	1.2
A_1	—	—	—
e	—	0.65	—
b	0.30	0.35	0.40
x	—	—	0.08
y	—	—	0.1
y_1	—	—	0.20
S_D	—	—	—
S_E	—	—	—
Z_D	—	0.6	—
Z_E	—	0.6	—

Figure C.3 Package Dimensions (TLP-145V)

D. Treatment of Unused Pins

The treatments of unused pins are listed in table D.1.

Table D.1 Treatment of Unused Pins

Pin Name	Example of Pin Treatment
$\overline{\text{RES}}$	(Always used as a reset pin)
$\overline{\text{ETRST}}$	(Always used as a reset pin)
MD2, MD1	(Always used as mode pins)
NMI	• Connect to V_{CC} via a pull-up resistor
EXTAL	(Always used as a clock pin)
XTAL	(Always used as a clock pin)
Port 1 Port 2 Port 3 Port 4 Port 5 Port 6 Port 8 Port 9 Port A Port B Port C Port D Port F Port G Port H Port I Port J	• Connect each pin to V_{CC} via a pull-up resistor or to V_{SS} via a pull-down resistor
Port 7	• Connect each pin to AV_{CC} via a pull-up resistor or to AV_{SS} via a pull-down resistor
Port E	• Connect each pin to V_{CC} via a pull-up resistor

Index

Numerics

14-bit PWM timer (PWMX) 219
16-bit count mode 380
16-bit cycle measurement timer (TCM) 305
16-bit timer pulse unit 237
8-bit timer (TMR) 355

A

A/D converter 727
A/D converter activation 289
Absolute address 58
Address map 74
Address space 36
Addressing modes 57
ADI 739
Advanced mode 34
Arithmetic operations instructions 48
Asynchronous mode 424

B

Base cycle 229
Bcc 45, 53
Bit manipulation instructions 51
Bit rate 419
Block sata transfer instructions 55
Block structure 753
Boot mode 750, 776
Branch instructions 53
Buffer operation 275
Bus controller (BSC) 133

C

Cascaded connection 380
Clock pulse generator 833
Clocked synchronous mode 441
CMIA 384
CMIB 384
Communications protocol 804
Compare-match count mode 380
Condition field 56
Condition-code register (CCR) 40
Conversion cycle 229
Conversion time 737
CPU operating modes 32
Crystal resonator 834
Cycle measurement mode 319

D

Data transfer instructions 47
Download pass/fail result parameter 767

E

Effective address 57, 61
Effective address extension 56
ERI1 460
Error protection 799
Exception handling 75
Exception handling vector table 76, 78
Extended control register (EXR) 39
Extended vector mode 107
External clock 835

F

Flash erase block select parameter 775
Flash memory 329, 749
Flash multipurpose address area parameter 774
Flash multipurpose data destination parameter 774
Flash pass and fail parameter 768
Flash program/erase frequency parameter 773
FOVI 350
Framing error 431

G

General registers 38

H

H8S/2140B group compatible vector mode 101
Hardware protection 798

I

I^2C bus data format 560
I^2C bus interface (IIC) 529
ICIA 350
ICIX 384
IICI 581
Immediate 59
Input capture operation 382
Instruction set 45
Interface 403
Internal block diagram 8
Interrupt controller 85
Interrupt exception handling 81
interrupt exception handling vector table 108
Interrupt mask bit 40
interrupt mask level 39
Interval timer mode 399

K

Keyboard buffer control unit (PS2) 587

L

Logic operations instructions 50
LPC interface (LPC) 615
LPC interface clock start request 673
LSI internal states in each operating mode 849

M

Memory indirect 60
Mode transition diagram 848
Module stop mode 854
Multiply-accumulate register (MAC) 41
Multiprocessor communication function 435

N

Noise canceler 579
Normal mode 32

O

OCIA 350
OCIB 350
On-board programming 776
On-board programming mode 776
Operation field 56
Output buffer control 154
Overflow 398
Overrun error 431

OVI 384

P

Parity error 431
Pin arrangement in each operating mode 12
Pin assignments 9
Pin functions 19
Power-down modes 841
Program counter (PC) 39
program-counter relative 59
Programmer mode 750, 802
Programming/erasing interface 754
Programming/erasing interface parameters 765
Programming/erasing interface register 759
Protection 798
PWM modes 279

R

RAM 747
Register direct 57
Register field 56
Register indirect 57
Register indirect with displacement 58
Register indirect with post-Increment 58
Register indirect with pre-decrement 58
Registers
ABRKCR 91
ADCR 734
ADCSR 732
ADDR 731
BAR 92
BCR 133
BRR 419
DACNT 221
DACR 224
DADR 222
DPFR 767
FCCS 759
FDLH 498
FDLL 498
FEBS 775
FECS 761
FFCR 502
FIER 499
FIIR 500
FKEY 762
FLCR 503
FLSR 506
FMATS 763
FMCR 504
FMPAR 774
FMPDR 774
FMSR 510
FPCS 761
FPEFEQ 773
FPFR 768
FRBR 497
FRSR 497
FSCR 511
FTDAR 764
FTHR 498
FTSR 498
HICR 620
HISEL 659
ICCR 543
ICDR 536
ICMR 540
ICR 90
ICRES 555
ICSR 551
ICXR 556
IDR 637
IER 96
ISCR 93
ISR 97

ISSR 103
KBBR 598
KBCR1 591
KBCR2 593
KBCRH 594
KBCRL 596
KBTR 598
LADR 634
LPWRCR 844
MDCR 68
MSTPCR 845
ODR 637
PTCNT0 194
PTCNT2 196
RDR 407
RSR 407
SAR 537
SARX 538
SBYCR 842
SCIFCR 512
SCMR 418
SCR 411
SIRQCR 645
SMR 408
SSR 414
STCR 71
STR 638
SYSCR 69
SYSCR3 73
TCMCNT 309
TCMCR 313
TCMCSR 311
TCMICR 310
TCMICRF 310
TCMIER 315
TCMMINCM 310
TCMMLCM 309
TCNT 263, 360, 394
TCONRI 373
TCONRS 373
TCOR 361
TCR 243, 362
TCSR 367
TDPCNT 333
TDPCR1 338
TDPCR2 340
TDPCSR 335
TDPICR 335
TDPICRF 335
TDPIER 341
TDPPDMN 334
TDPPDMX 334
TDPWDMN 334
TDPWDMX 333
TDR 407
TGR 263
TICRF 372
TICRR 372
TIOR 249
TMDR 247
TSR 260
TSTR 263
TSYR 264
TWR 638
WER 104
WSCR 134
WUESCR 104
WUESR 104
Reset 79
Reset exception handling 79
RXI1 460

S

Scan mode 736
Serial communication interface (SCI) 403
Serial communication interface with FIFO (SCIF) 493
Serial data reception 431, 445
Serial data transmission 429, 443

Serial formats 560
Shift instructions 50
Single mode 735
Sleep mode 851
Smart card 403
Smart card interface 449
Software protection 799
Software standby mode 851
Stack pointer (SP) 38
Stack status 83
Standard serial communication interface specifications for boot mode 802
Synchronous operation 273
System control instructions 54

T

TCI0V 288
TCI1U 288
TCI1V 288
TCI2U 288
TCI2V 288
TEI1 460
TGI0A 288
TGI0B 288
TGI0C 288
TGI0D 288
TGI1A 288
TGI1B 288
TGI2A 288
TGI2B 288
Toggle output 270
Trace bit 39
Trap instruction exception handling 81
TRAPA instruction 59, 81
TXI1 460

U

USB 2.0 host/function module (USB) 469
User boot MAT 801
User boot mode 789
User MAT 801
User program mode 750, 780

V

Vector address switching 130

W

Watch mode 853
Watchdog timer (WDT) 391
Watchdog timer mode 398
Waveform output by compare match 269
WOVI 400

Renesas 16-Bit Single-Chip Microcomputer
Hardware Manual
H8S/2117R Group

Publication Date:	Rev.1.00, Apr. 28, 2008
Published by:	Sales Strategic Planning Div. Renesas Technology Corp.
Edited by:	Customer Support Department Global Strategic Communication Div. Renesas Solutions Corp.

© 2008. Renesas Technology Corp., All rights reserved. Printed in Japan.

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan

RENESAS SALES OFFICES

http://www.renesas.com

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, Malaysia
Tel: <603> 7955-9390, Fax: <603> 7955-9510

H8S/2117R Group
Hardware Manual



Renesas Technology Corp.
2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan