REJ09B0098-0300

Everywhere you imagine. RENESAS

The revision list can be viewed directly by clicking the title page.

The revision list summarizes the locations of revisions and additions. Details should always be checked by referring to the relevant text.

16

H8S/2114R Group

Hardware Manual

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

H8S/2114R R4F2114R

Hardware Manual

Rev.3.00
Revision Date: Jul. 14, 2005

Renesas Technology
www.renesas.com

Keep safety first in your circuit designs!

1. Renesas Technology Corp. puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage.
   Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corp. product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corp. or a third party.
2. Renesas Technology Corp. assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corp. without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor for the latest product information before purchasing a product listed herein.
   The information described here may contain technical inaccuracies or typographical errors.
   Renesas Technology Corp. assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.
   Please also pay attention to information published by Renesas Technology Corp. by various means, including the Renesas Technology Corp. Semiconductor home page (http://www.renesas.com).
4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corp. assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corp. semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
6. The prior written approval of Renesas Technology Corp. is necessary to reprint or reproduce in whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
   Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
8. Please contact Renesas Technology Corp. for further details on these materials or the products contained therein.

General Precautions on Handling of Product

1. Treatment of NC Pins

Note: Do not connect anything to the NC pins.
The NC (not connected) pins are either not connected to any of the internal circuitry or are used as test pins or to reduce noise. If something is connected to the NC pins, the operation of the LSI is not guaranteed.

2. Treatment of Unused Input Pins

Note: Fix all unused input pins to high or low level.
Generally, the input pins of CMOS products are high-impedance input pins. If unused pins are in their open states, intermediate levels are induced by noise in the vicinity, a pass-through current flows internally, and a malfunction may occur.

3. Processing before Initialization

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

4. Prohibition of Access to Undefined or Reserved Addresses

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

Configuration of This Manual

This manual comprises the following items:

1. General Precautions on Handling of Product
2. Configuration of This Manual
3. Preface
4. Contents
5. Overview
6. Description of Functional Modules
   - CPU and System-Control Modules
   - On-Chip Peripheral Modules

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

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

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

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

    Product code, Package dimensions, etc.

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

11. Index

Preface

This H8S/2114R Group is a series of microcomputers (MCUs) made up of the H8S/2000 CPU with Renesas Technology's original architecture as its core, and the peripheral functions required to configure a system.

The H8S/2000 CPU has an internal 32-bit configuration, sixteen 16-bit general registers, and a simple and optimized instruction set for high-speed operation. The H8S/2000 CPU can handle a 16-Mbyte linear address space. The instruction set of the H8S/2000 CPU maintains upward compatibility at the object level with the H8/300 and H8/300H CPUs. This allows the transition from the H8/300, H8/300L, or H8/300H to the H8S/2000 CPU.

This LSI is equipped with ROM, RAM, two kinds of PWM timers (PWM and PWMX), a 16-bit free running timer (FRT), a 16-bit timer pulse unit (TPU), 8-bit timers (TMR), watchdog timer (WDT), serial communication interface (SCI), I^2C bus interface (IIC), a LPC interface (LPC), a keyboard buffer control units (KBU), an A/D converter, and I/O ports as on-chip peripheral modules required for system configuration.

A data transfer controller (DTC) and LPC interface (LPC) are included as bus masters.

A flash memory (F-ZTAT™*) is available for this LSI's 1 Mbyte ROM. The CPU and ROM are connected to a 16-bit bus, enabling byte data and word data to be accessed in a single state. This improves the instruction fetch and process speeds.

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

Target Users: This manual was written for users who use the H8S/2114R in the design of application systems. Target users are expected to understand the fundamentals of electrical circuits, logic circuits, and microcomputers.

Objective: This manual was written to explain the hardware functions and electrical characteristics of the H8S/2114R Group to the target users.
Refer to the H8S/2600 Series, H8S/2000 Series Programming Manual for a detailed description of the instruction set.

Notes on reading this manual:

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

- In order to understand the details of the CPU's functions

  Read the H8S/2600 Series, H8S/2000 Series Programming Manual.
- In order to understand the detailed function of a register whose name is known

  Read the index that is the final part of the manual to find the page number of the entry on the register. The addresses, bits, and initial values of the registers are summarized in section 25, List of Registers.

Rules:	Register name:	The following notation is used for cases when the same or a similar function, e.g., serial communication interface, is implemented on more than one channel: XXX_N (XXX is the register name and N is the channel number)
	Bit order:	The MSB is on the left and the LSB is on the right.
	Number notation:	Binary is B'xxxx, hexadecimal is H'xxxx, decimal is xxxx.
	Signal notation:	An overbar is added to a low-active signal: $\overline{\text{xxxx}}$

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

H8S/2114R Group manuals:

Document Title	Document No.
H8S/2114R Group Hardware Manual	This manual
H8S/2600 Series, H8S/2000 Series Programming Manual	REJ09B0139

User's manuals for development tools:

Document Title	Document No.
H8S, H8/300 Series C/C++ Compiler, Assembler, Optimizing Linkage Editor User's Manual	REJ10B0058
Microcomputer Development Environment System H8S, H8/300 Series Simulator/Debugger User's Manual	ADE-702-282
H8S, H8/300 Series High-performance Embedded Workshop 3 Tutorial	REJ10B0024
H8S, H8/300 Series High-performance Embedded Workshop 3 User's Manual	REJ10B0026

Main Revisions and Additions in this Edition

Item	Page	Revisions (See Manual for Details)
All pages	—	Suffix R is added to group name and product code. • H8S/2114 Group→ H8S/2114R Group • R4F2114 → R4F2114R
Appendix C. Package Dimensions Figure C.1 Package Dimensions (TFP-144)	977	Replaced.

Contents

Section 1 Overview 1
1.1 Overview 1
1.2 Internal Block Diagram 3
1.3 Pin Description 4
1.3.1 Pin Arrangement 4
1.3.2 Pin Arrangement in Each Operating Mode 5
1.3.3 Pin Functions 10

Section 2 CPU 19
2.1 Features 19
2.1.1 Differences between H8S/2600 CPU and H8S/2000 CPU 20
2.1.2 Differences from H8/300 CPU 21
2.1.3 Differences from H8/300H CPU 21
2.2 CPU Operating Modes 22
2.2.1 Normal Mode 22
2.2.2 Advanced Mode 24
2.3 Address Space 26
2.4 Register Configuration 27
2.4.1 General Registers 28
2.4.2 Program Counter (PC) 29
2.4.3 Extended Control Register (EXR) 29
2.4.4 Condition-Code Register (CCR) 30
2.4.5 Initial Register Values 31
2.5 Data Formats 32
2.5.1 General Register Data Formats 32
2.5.2 Memory Data Formats 34
2.6 Instruction Set 35
2.6.1 Table of Instructions Classified by Function 36
2.6.2 Basic Instruction Formats 47
2.7 Addressing Modes and Effective Address Calculation 48
2.7.1 Register Direct—Rn 48
2.7.2 Register Indirect—@ERn 48
2.7.3 Register Indirect with Displacement—@(d:16, ERn) or @(d:32, ERn) 49
2.7.4 Register Indirect with Post-Increment or Pre-Decrement—@ERn+ or @-ERn 49
2.7.5 Absolute Address—@aa:8, @aa:16, @aa:24, or @aa:32 49
2.7.6 Immediate—#xx:8, #xx:16, or #xx:32 50

2.7.7 Program-Counter Relative—@(d:8, PC) or @(d:16, PC) 50
2.7.8 Memory Indirect—@@aa:8 51
2.7.9 Effective Address Calculation 52
2.8 Processing States 54
2.9 Usage Notes 56
2.9.1 Note on TAS Instruction Usage 56
2.9.2 Note on STM/LDM Instruction Usage 56
2.9.3 Note on Bit Manipulation Instructions 56
2.9.4 EEPMOV Instruction 57

Section 3 MCU Operating Modes 59
3.1 Operating Mode Selection 59
3.2 Register Descriptions 60
3.2.1 Mode Control Register (MDCR) 60
3.2.2 System Control Register (SYSCR) 61
3.2.3 Serial Timer Control Register (STCR) 63
3.2.4 System Control Register 3 (SYSCR3) 66
3.3 Operating Mode Descriptions 67
3.3.1 Mode 2 67
3.3.2 Mode 3 67
3.4 Address Map 67

Section 4 Exception Handling 69
4.1 Exception Handling Types and Priority 69
4.2 Exception Sources and Exception Vector Table 70
4.3 Reset 74
4.3.1 Reset Exception Handling 74
4.3.2 Interrupts Immediately after Reset 75
4.3.3 On-Chip Peripheral Modules after Reset is Cancelled 75
4.4 Interrupt Exception Handling 76
4.5 Trap Instruction Exception Handling 76
4.6 Stack Status after Exception Handling 77
4.7 Usage Note 78

Section 5 Interrupt Controller 79
5.1 Features 79
5.2 Input/Output Pins 81
5.3 Register Descriptions 82
5.3.1 Interrupt Control Registers A to D (ICRA to ICRD) 83
5.3.2 Address Break Control Register (ABRKCR) 85

5.3.3 Break Address Registers A to C (BARA to BARC) ... 86
5.3.4 IRQ Sense Control Registers (ISCR16H, ISCR16L, ISCRH, ISCRL) ... 87
5.3.5 IRQ Enable Registers (IER16, IER) ... 90
5.3.6 IRQ Status Registers (ISR16, ISR) ... 91
5.3.7 Keyboard Matrix Interrupt Mask Registers (KMIMRA, KMIMR) Wake-Up Event Interrupt Mask Registers (WUEMR, WUEMRB) ... 93
5.3.8 IRQ Sense Port Select Register 16 (ISSR16), IRQ Sense Port Select Register (ISSR) ... 97
5.4 Interrupt Sources ... 99
5.4.1 External Interrupt Sources ... 99
5.4.2 Internal Interrupt Sources ... 102
5.5 Interrupt Exception Handling Vector Tables ... 102
5.6 Interrupt Control Modes and Interrupt Operation ... 109
5.6.1 Interrupt Control Mode 0 ... 112
5.6.2 Interrupt Control Mode 1 ... 114
5.6.3 Interrupt Exception Handling Sequence ... 117
5.6.4 Interrupt Response Times ... 119
5.6.5 DTC Activation by Interrupt ... 120
5.7 Address Breaks ... 122
5.7.1 Features ... 122
5.7.2 Block Diagram ... 122
5.7.3 Operation ... 123
5.7.4 Usage Notes ... 123
5.8 Usage Notes ... 125
5.8.1 Conflict between Interrupt Generation and Disabling ... 125
5.8.2 Instructions for Disabling Interrupts ... 126
5.8.3 Interrupts during Execution of EEPMOV Instruction ... 126
5.8.4 Vector Address Switching ... 126
5.8.5 External Interrupt Pin in Software Standby Mode and Watch Mode ... 127
5.8.6 Noise Canceller Switching ... 127
5.8.7 IRQ Status Register (ISR) ... 127

Section 6 Bus Controller (BSC) ... 129
6.1 Features ... 129
6.2 Register Descriptions ... 130
6.2.1 Bus Control Register (BCR) ... 130
6.2.2 Wait State Control Register (WSCR) ... 131
6.3 Bus Arbitration ... 132
6.3.1 Priority of Bus Masters ... 132
6.3.2 Bus Transfer Timing ... 132

Section 7 Data Transfer Controller (DTC) 135
7.1 Features 136
7.2 Register Descriptions 137
7.2.1 DTC Mode Register A (MRA) 138
7.2.2 DTC Mode Register B (MRB) 139
7.2.3 DTC Source Address Register (SAR) 139
7.2.4 DTC Destination Address Register (DAR) 140
7.2.5 DTC Transfer Count Register A (CRA) 140
7.2.6 DTC Transfer Count Register B (CRB) 140
7.2.7 DTC Enable Registers (DTCER) 141
7.2.8 DTC Vector Register (DTVECR) 142
7.3 Activation Sources 143
7.4 Location of Register Information and DTC Vector Table 144
7.5 Operation 147
7.5.1 Normal Mode 148
7.5.2 Repeat Mode 149
7.5.3 Block Transfer Mode 150
7.5.4 Chain Transfer 151
7.5.5 Interrupt Sources 152
7.5.6 Operation Timing 152
7.5.7 Number of DTC Execution States 154
7.6 Procedures for Using DTC 155
7.6.1 Activation by Interrupt 155
7.6.2 Activation by Software 155
7.7 Examples of Use of the DTC 156
7.7.1 Normal Mode 156
7.7.2 Software Activation 157
7.8 Usage Notes 158
7.8.1 Module Stop Mode Setting 158
7.8.2 On-Chip RAM 158
7.8.3 DTCE Bit Setting 158
7.8.4 Setting Required on Entering Subactive Mode or Watch Mode 158
7.8.5 DTC Activation by Interrupt Sources of SCI, IIC, LPC, or A/D Converter 158

Section 8 I/O Ports 159
8.1 Port 1 164
8.1.1 Port 1 Data Direction Register (P1DDR) 164
8.1.2 Port 1 Data Register (P1DR) 165
8.1.3 Port 1 Pull-Up MOS Control Register (P1PCR) 165
8.1.4 Pin Functions 166

8.1.5 Port 1 Input Pull-Up MOS 166
8.2 Port 2 167
8.2.1 Port 2 Data Direction Register (P2DDR) 167
8.2.2 Port 2 Data Register (P2DR) 168
8.2.3 Port 2 Pull-Up MOS Control Register (P2PCR) 168
8.2.4 Pin Functions 169
8.2.5 Port 2 Input Pull-Up MOS 170
8.3 Port 3 171
8.3.1 8.3.1 Port 3 Data Direction Register (P3DDR) 171
8.3.2 Port 3 Data Register (P3DR) 172
8.3.3 Port 3 Pull-Up MOS Control Register (P3PCR) 172
8.3.4 Pin Functions 173
8.3.5 Port 3 Input Pull-Up MOS 173
8.4 Port 4 174
8.4.1 Port 4 Data Direction Register (P4DDR) 174
8.4.2 Port 4 Data Register (P4DR) 175
8.4.3 Pin Functions 175
8.5 Port 5 178
8.5.1 Port 5 Data Direction Register (P5DDR) 178
8.5.2 Port 5 Data Register (P5DR) 178
8.5.3 Pin Functions 179
8.6 Port 6 180
8.6.1 Port 6 Data Direction Register (P6DDR) 180
8.6.2 Port 6 Data Register (P6DR) 181
8.6.3 Pull-Up MOS Control Register (KMPCR) 181
8.6.4 Noise Canceller Enable Register (P6NCE) 182
8.6.5 Noise Canceller Mode Control Register (P6NCMC) 182
8.6.6 Noise Cancel Cycle Setting Register (P6NCCS) 183
8.6.7 System Control Register 2 (SYSCR2) 185
8.6.8 Pin Functions 185
8.6.9 Port 6 Input Pull-Up MOS 188
8.7 Port 7 189
8.7.1 Port 7 Input Data Register (P7PIN) 189
8.7.2 Pin Functions 190
8.8 Port 8 191
8.8.1 Port 8 Data Direction Register (P8DDR) 191
8.8.2 Port 8 Data Register (P8DR) 192
8.8.3 Pin Functions 193
8.9 Port 9 196
8.9.1 Port 9 Data Direction Register (P9DDR) 196

8.9.2 Port 9 Data Register (P9DR) 197
8.9.3 Port 9 Pull-Up MOS Control Register (P9PCR) 197
8.9.4 Pin Functions 198
8.9.5 Port 9 Input Pull-Up MOS 200
8.10 Port A 201
8.10.1 Port A Data Direction Register (PADDR) 201
8.10.2 Port A Output Data Register (PAODR) 202
8.10.3 Port A Input Data Register (PAPIN) 202
8.10.4 Pin Functions 203
8.11 Port B 204
8.11.1 Port B Data Direction Register (PBDDR) 204
8.11.2 Port B Output Data Register (PBODR) 205
8.11.3 Port B Input Data Register (PBPIN) 205
8.11.4 Pin Functions 206
8.11.5 Port B Input Pull-Up MOS 208
8.12 Port C 209
8.12.1 Port C Data Direction Register (PCDDR) 209
8.12.2 Port C Output Data Register (PCODR) 210
8.12.3 Port C Input Data Register (PCPIN) 210
8.12.4 Noise Canceller Enable Register (PCNCE) 211
8.12.5 Noise Canceller Mode Control Register (PCNCMC) 211
8.12.6 Noise Cancel Cycle Setting Register (PCNCCS) 212
8.12.7 Pin Functions 212
8.12.8 Port C Nch-OD control register (PCNOCR) 215
8.12.9 Pin Functions 215
8.12.10 Port C Input Pull-Up MOS 216
8.13 Port D 217
8.13.1 Port D Data Direction Register (PDDDR) 217
8.13.2 Port D Output Data Register (PDODR) 218
8.13.3 Port D Input Data Register (PDPIN) 218
8.13.4 Pin Functions 219
8.13.5 Port D Nch-OD control register (PDNOCR) 223
8.13.6 Pin Functions 223
8.13.7 Port D Input Pull-Up MOS 224
8.14 Port E 225
8.14.1 Port E Input Pull-Up MOS Control Register (PEPCR) 225
8.14.2 Port E Input Data Register (PEPIN) 225
8.14.3 Pin Functions 226
8.14.4 Port E Input Pull-Up MOS 226
8.15 Port F 227

8.15.1 Port F Data Direction Register (PFDDR) 227
8.15.2 Port F Output Data Register (PFODR) 228
8.15.3 Port F Input Data Register (PFPIN) 228
8.15.4 Pin Functions 229
8.15.5 Port F Nch-OD control register (PFNOCR) 231
8.15.6 Pin Functions 231
8.15.7 Port F Input Pull-Up MOS 232
8.16 Port G 233
8.16.1 Port G Data Direction Register (PGDDR) 233
8.16.2 Port G Output Data Register (PGODR) 234
8.16.3 Port G Input Data Register (PGPIN) 234
8.16.4 Noise Canceller Enable Register (PGNCE) 235
8.16.5 Noise Canceller Mode Control Register (PGNCMC) 235
8.16.6 Noise Cancel Cycle Setting Register (PGNCCS) 236
8.16.7 Pin Functions 237
8.16.8 Port G Nch-OD control register (PGNOCR) 242
8.16.9 Pin Functions 242
8.17 Change of Peripheral Function Pins 243
8.17.1 Port Control Register 0 (PTCNT0) 243
8.17.2 Port Control Register 1 (PTCNT1) 244
8.17.3 Port Control Register 2 (PTCNT2) 245

Section 9 8-Bit PWM Timer (PWM) 247
9.1 Features 247
9.2 Input/Output Pins 249
9.3 Register Descriptions 249
9.3.1 PWM Register Select (PWSL) 250
9.3.2 PWM Data Registers 15 to 8 (PWDR15 to PWDR8) 251
9.3.3 PWM Data Polarity Register B (PWDPRB) 252
9.3.4 PWM Output Enable Register B (PWOERB) 252
9.3.5 Peripheral Clock Select Register (PCSR) 253
9.4 Operation 254
9.4.1 PWM Setting Example 256
9.4.2 Diagram of PWM Used as D/A Converter 256
9.5 Usage Notes 257
9.5.1 Module Stop Mode Setting 257

Section 10 14-Bit PWM Timer (PWMX) 259
10.1 Features 259
10.2 Input/Output Pins 260
10.3 Register Descriptions 260
10.3.1 PWMX (D/A) Counter (DACNT) 261
10.3.2 PWMX (D/A) Data Registers A and B (DADRA and DADRB) 262
10.3.3 PWMX (D/A) Control Register (DACR) 264
10.3.4 Peripheral Clock Select Register (PCSR) 265
10.4 Bus Master Interface 266
10.5 Operation 269
10.6 Usage Notes 276
10.6.1 Module Stop Mode Setting 276

Section 11 16-Bit Free-Running Timer (FRT) 277
11.1 Features 277
11.2 Input/Output Pins 279
11.3 Register Descriptions 279
11.3.1 Free-Running Counter (FRC) 280
11.3.2 Output Compare Registers A and B (OCRA and OCRB) 280
11.3.3 Input Capture Registers A to D (ICRA to ICRD) 280
11.3.4 Output Compare Registers AR and AF (OCRAR and OCRAF) 281
11.3.5 Output Compare Register DM (OCRDM) 281
11.3.6 Timer Interrupt Enable Register (TIER) 282
11.3.7 Timer Control/Status Register (TCSR) 283
11.3.8 Timer Control Register (TCR) 286
11.3.9 Timer Output Compare Control Register (TOCR) 287
11.4 Operation 289
11.4.1 Pulse Output 289
11.5 Operation Timing 290
11.5.1 FRC Increment Timing 290
11.5.2 Output Compare Output Timing 291
11.5.3 FRC Clear Timing 291
11.5.4 Input Capture Input Timing 292
11.5.5 Buffered Input Capture Input Timing 293
11.5.6 Timing of Input Capture Flag (ICF) Setting 294
11.5.7 Timing of Output Compare Flag (OCF) setting 295
11.5.8 Timing of FRC Overflow Flag Setting 295
11.5.9 Automatic Addition Timing 296
11.5.10 Mask Signal Generation Timing 297

11.6 Interrupt Sources 298
11.7 Usage Notes 299
11.7.1 Conflict between FRC Write and Clear 299
11.7.2 Conflict between FRC Write and Increment 300
11.7.3 Conflict between OCR Write and Compare-Match 301
11.7.4 Switching of Internal Clock and FRC Operation 302
11.7.5 Module Stop Mode Setting 304

Section 12 16-Bit Timer Pulse Unit (TPU) 305
12.1 Features 305
12.2 Input/Output Pins 309
12.3 Register Descriptions 310
12.3.1 Timer Control Register (TCR) 311
12.3.2 Timer Mode Register (TMDR) 315
12.3.3 Timer I/O Control Register (TIOR) 317
12.3.4 Timer Interrupt Enable Register (TIER) 326
12.3.5 Timer Status Register (TSR) 328
12.3.6 Timer Counter (TCNT) 331
12.3.7 Timer General Register (TGR) 331
12.3.8 Timer Start Register (TSTR) 331
12.3.9 Timer Synchro Register (TSYR) 332
12.4 Interface to Bus Master 333
12.4.1 16-Bit Registers 333
12.4.2 8-Bit Registers 333
12.5 Operation 335
12.5.1 Basic Functions 335
12.5.2 Synchronous Operation 341
12.5.3 Buffer Operation 343
12.5.4 PWM Modes 347
12.5.5 Phase Counting Mode 352
12.6 Interrupts 357
12.6.1 Interrupt Source and Priority 357
12.6.2 DTC Activation 359
12.6.3 A/D Converter Activation 359
12.7 Operation Timing 360
12.7.1 Input/Output Timing 360
12.7.2 Interrupt Signal Timing 364
12.8 Usage Notes 368
12.8.1 Input Clock Restrictions 368
12.8.2 Caution on Period Setting 368

12.8.3 Conflict between TCNT Write and Clear Operations 369
12.8.4 Conflict between TCNT Write and Increment Operations 369
12.8.5 Conflict between TGR Write and Compare Match 370
12.8.6 Conflict between Buffer Register Write and Compare Match 371
12.8.7 Conflict between TGR Read and Input Capture 372
12.8.8 Conflict between TGR Write and Input Capture 373
12.8.9 Conflict between Buffer Register Write and Input Capture 374
12.8.10 Conflict between Overflow/Underflow and Counter Clearing 375
12.8.11 Conflict between TCNT Write and Overflow/Underflow 376
12.8.12 Multiplexing of I/O Pins 376
12.8.13 Module Stop Mode Setting 376

Section 13 8-Bit Timer (TMR) 377
13.1 Features 377
13.2 Input/Output Pins 381
13.3 Register Descriptions 382
13.3.1 Timer Counter (TCNT) 383
13.3.2 Time Constant Register A (TCORA) 383
13.3.3 Time Constant Register B (TCORB) 384
13.3.4 Timer Control Register (TCR) 384
13.3.5 Timer Control/Status Register (TCSR) 389
13.3.6 Time Constant Register C (TCORC) 394
13.3.7 Input Capture Registers R and F (TICRR and TICRF) 394
13.3.8 Timer Input Select Register (TISR) 395
13.3.9 Timer Connection Register I (TCONRI) 395
13.3.10 Timer Connection Register S (TCONRS) 396
13.3.11 Timer XY Control Register (TCRXY) 396
13.4 Operation 397
13.4.1 Pulse Output 397
13.5 Operation Timing 398
13.5.1 TCNT Count Timing 398
13.5.2 Timing of CMFA and CMFB Setting at Compare-Match 399
13.5.3 Timing of Timer Output at Compare-Match 399
13.5.4 Timing of Counter Clear at Compare-Match 400
13.5.5 TCNT External Reset Timing 400
13.5.6 Timing of Overflow Flag (OVF) Setting 401
13.6 TMR_0 and TMR_1 Cascaded Connection 402
13.6.1 16-Bit Count Mode 402
13.6.2 Compare-Match Count Mode 402
13.7 TMR_Y and TMR_X Cascaded Connection 403

13.7.1 16-Bit Count Mode 403
13.7.2 Compare-Match Count Mode 403
13.7.3 Input Capture Operation 404
13.8 Interrupt Sources 406
13.9 Usage Notes 407
13.9.1 Conflict between TCNT Write and Counter Clear 407
13.9.2 Conflict between TCNT Write and Count-Up 408
13.9.3 Conflict between TCOR Write and Compare-Match 409
13.9.4 Conflict between Compare-Matches A and B 410
13.9.5 Switching of Internal Clocks and TCNT Operation 410
13.9.6 Mode Setting with Cascaded Connection 412
13.9.7 Module Stop Mode Setting 412

Section 14 Watchdog Timer (WDT) 413
14.1 Features 413
14.2 Input/Output Pins 415
14.3 Register Descriptions 415
14.3.1 Timer Counter (TCNT) 415
14.3.2 Timer Control/Status Register (TCSR) 416
14.4 Operation 420
14.4.1 Watchdog Timer Mode 420
14.4.2 Interval Timer Mode 421
14.4.3 $\overline{\text{RESO}}$ Signal Output Timing 422
14.5 Interrupt Sources 423
14.6 Usage Notes 424
14.6.1 Notes on Register Access 424
14.6.2 Conflict between Timer Counter (TCNT) Write and Increment 425
14.6.3 Changing Values of CKS2 to CKS0 Bits 426
14.6.4 Changing Value of PSS Bit 426
14.6.5 Switching between Watchdog Timer Mode and Interval Timer Mode 426
14.6.6 System Reset by $\overline{\text{RESO}}$ Signal 426

Section 15 Serial Communication Interface (SCI, IrDA) 427
15.1 Features 427
15.2 Input/Output Pins 430
15.3 Register Descriptions 431
15.3.1 Receive Shift Register (RSR) 431
15.3.2 Receive Data Register (RDR) 431
15.3.3 Transmit Data Register (TDR) 432
15.3.4 Transmit Shift Register (TSR) 432

15.3.5 Serial Mode Register (SMR) 432
15.3.6 Serial Control Register (SCR) 436
15.3.7 Serial Status Register (SSR) 439
15.3.8 Smart Card Mode Register (SCMR) 444
15.3.9 Bit Rate Register (BRR) 445
15.3.10 Keyboard Comparator Control Register (KBCOMP) 453
15.4 Operation in Asynchronous Mode 455
15.4.1 Data Transfer Format 455
15.4.2 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode 457
15.4.3 Clock 458
15.4.4 SCI Initialization (Asynchronous Mode) 459
15.4.5 Serial Data Transmission (Asynchronous Mode) 460
15.4.6 Serial Data Reception (Asynchronous Mode) 462
15.5 Multiprocessor Communication Function 466
15.5.1 Multiprocessor Serial Data Transmission 468
15.5.2 Multiprocessor Serial Data Reception 469
15.6 Operation in Clocked Synchronous Mode 472
15.6.1 Clock 472
15.6.2 SCI Initialization (Clocked Synchronous Mode) 473
15.6.3 Serial Data Transmission (Clocked Synchronous Mode) 474
15.6.4 Serial Data Reception (Clocked Synchronous Mode) 477
15.6.5 Simultaneous Serial Data Transmission and Reception (Clocked Synchronous Mode) 479
15.7 Smart Card Interface Description 481
15.7.1 Sample Connection 481
15.7.2 Data Format (Except in Block Transfer Mode) 481
15.7.3 Block Transfer Mode 483
15.7.4 Receive Data Sampling Timing and Reception Margin 483
15.7.5 Initialization 484
15.7.6 Serial Data Transmission (Except in Block Transfer Mode) 485
15.7.7 Serial Data Reception (Except in Block Transfer Mode) 488
15.7.8 Clock Output Control 490
15.8 IrDA Operation 492
15.9 Interrupt Sources 496
15.9.1 Interrupts in Normal Serial Communication Interface Mode 496
15.9.2 Interrupts in Smart Card Interface Mode 497
15.10 Usage Notes 498
15.10.1 Module Stop Mode Setting 498
15.10.2 Break Detection and Processing 498

15.10.3 Mark State and Break Sending ... 498
15.10.4 Receive Error Flags and Transmit Operations (Clocked Synchronous Mode Only) ... 498
15.10.5 Relation between Writing to TDR and TDRE Flag ... 498
15.10.6 Restrictions on Using DTC ... 499
15.10.7 SCI Operations during Mode Transitions ... 499
15.10.8 Notes on Switching from SCK Pins to Port Pins ... 503

Section 16 I²C Bus Interface (IIC) ... 505
16.1 Features ... 505
16.2 Input/Output Pins ... 509
16.3 Register Descriptions ... 510
16.3.1 I²C Bus Data Register (ICDR) ... 510
16.3.2 Slave Address Register (SAR) ... 511
16.3.3 Second Slave Address Register (SARX) ... 512
16.3.4 I²C Bus Mode Register (ICMR) ... 514
16.3.5 I²C Bus Control Register (ICCR) ... 517
16.3.6 I²C Bus Status Register (ICSR) ... 526
16.3.7 DDC Switch Register (DDCSWR) ... 530
16.3.8 I²C Bus Extended Control Register (ICXR) ... 531
16.4 Operation ... 535
16.4.1 I²C Bus Data Format ... 535
16.4.2 Initialization ... 537
16.4.3 Master Transmit Operation ... 537
16.4.4 Master Receive Operation ... 541
16.4.5 Slave Receive Operation ... 551
16.4.6 Slave Transmit Operation ... 559
16.4.7 IRIC Setting Timing and SCL Control ... 562
16.4.8 Operation by Using DTC ... 565
16.4.9 Noise Canceller ... 567
16.4.10 Initialization of Internal State ... 568
16.5 Interrupt Sources ... 569
16.6 Usage Notes ... 570
16.6.1 Module Stop Mode Setting ... 580

Section 17 Keyboard Buffer Control Unit (KBU) ... 581
17.1 Features ... 581
17.2 Input/Output Pins ... 584
17.3 Register Descriptions ... 585
17.3.1 Keyboard Control Register 1 (KBCR1) ... 585

17.3.2 Keyboard Buffer Control Register 2 (KBCR2) 587
17.3.3 Keyboard Control Register H (KBCRH) 588
17.3.4 Keyboard Control Register L (KBCRL) 590
17.3.5 Keyboard Data Buffer Register (KBBR) 592
17.3.6 Keyboard Buffer Transmit Data Register (KBTR) 592
17.4 Operation 593
17.4.1 Receive Operation 593
17.4.2 Transmit Operation 595
17.4.3 Receive Abort 597
17.4.4 KCLKI and KDI Read Timing 600
17.4.5 KCLKO and KDO Write Timing 601
17.4.6 KBF Setting Timing and KCLK Control 602
17.4.7 Receive Timing 603
17.4.8 Operation during Data Reception 604
17.4.9 KCLK Fall Interrupt Operation 605
17.4.10 First KCLK Falling Interrupt 606
17.5 Usage Notes 611
17.5.1 KBIOE Setting and KCLK Falling Edge Detection 611
17.5.2 KD Output by KDO bit (KBCRL) and by Automatic Transmission 612
17.5.3 Module Stop Mode Setting 612
17.5.4 Medium Speed Mode 612
17.5.5 Transmit Completion Flag (KBTE) 612

Section 18 LPC Interface (LPC) 613
18.1 Features 613
18.2 Input/Output Pins 616
18.3 Register Descriptions 617
18.3.1 Host Interface Control Registers 0 and 1 (HICR0 and HICR1) 619
18.3.2 Host Interface Control Registers 2 and 3 (HICR2 and HICR3) 625
18.3.3 Host Interface Control Register 4 (HICR4) 628
18.3.4 LPC Channel 3 Address Registers H and L (LADR3H and LADR3L) 629
18.3.5 LPC Channel 4 Address Registers H and L (LADR4H and LADR4L) 631
18.3.6 Input Data Registers 1 to 4 (IDR1 to IDR4) 632
18.3.7 Output Data Registers 1 to 4 (ODR1 to ODR4) 633
18.3.8 Bidirectional Data Registers 0 to 15 (TWR0 to TWR15) 633
18.3.9 Status Registers 1 to 4 (STR1 to STR4) 634
18.3.10 SERIRQ Control Register 0 (SIRQCR0) 640
18.3.11 SERIRQ Control Register 1 (SIRQCR1) 644
18.3.12 SERIRQ Control Register 2 (SIRQCR2) 649
18.3.13 Host Interface Select Register (HISEL) 653

18.3.14 RAM Buffer Address Register (RBUFAR) 654
18.3.15 Flash Memory Programming Address Registers H and L (FLWARH and FLARL) 655
18.3.16 Manufacture ID Code Register (LMCMIDCR) and Device ID Code Register (LMCDIDCR) 656
18.3.17 Erase Block Register (EBLKR) 657
18.3.18 LMC Status Registers 1 and 2 (LMCST1 and LMCST2) 658
18.3.19 LMC Control Registers 1 and 2 (LMCCR1 and LMCCR2) 662
18.3.20 Host Base Address Registers 1H and 1L (HBAR1H and HBAR1L) 665
18.3.21 Host Base Address Registers 2H and 2L (HBAR2H and HBAR2L) 666
18.3.22 On-Chip RAM Host Base Address Registers H and L (RAMBARH and RAMBARL) 667
18.3.23 Address Space Set Register (ASSR) 668
18.3.24 On-Chip RAM Address Space Set Register (RAMASSR) 669
18.3.25 Slave Address Register 1 (SAR1) 670
18.3.26 Slave Address Register 2 (SAR2) 671
18.3.27 On-Chip RAM Slave Address Register (RAMAR) 671
18.3.28 Flash Memory Write Protect Registers H, M, and L (FWPRH, FWPRM, and FWPRL) 672
18.3.29 Flash Memory Read Protect Registers H, M, and L (FRPRH, FRPRM, and FRPRL) 674
18.3.30 On-Chip RAM Protect Control Register (MPCR) 676
18.3.31 User Command Register (UCMDTR) 676
18.4 Operation 677
18.4.1 LPC interface Activation 677
18.4.2 LPC I/O Cycles 677
18.4.3 Gate A20 680
18.4.4 LPC Interface Shutdown Function (LPCPD) 683
18.4.5 LPC Interface Serialized Interrupt Operation (SERIRQ) 687
18.4.6 LPC Interface Clock Start Request 689
18.4.7 LPC/FW Memory Cycle 689
18.4.8 LPC/FW Memory Access Command 692
18.4.9 Flash Memory Address Translation (Host → Slave) 700
18.4.10 On-Chip RAM Address Translation (Host → Slave) 701
18.4.11 Address Space Priority 702
18.4.12 Example 1 of Address Space Priority 703
18.4.13 Example 2 of Address Space Priority 704
18.4.14 Flash Memory Protection 705
18.4.15 On-Chip RAM Protection 707
18.4.16 Flash Memory Programming 707

18.4.17 Flash Memory Erasing 709
18.5 Interrupt Sources 710
18.5.1 IBFI1, IBFI2, IBFI3, IBFI4, LMC, LMCUI, and ERRI 710
18.5.2 SMI, HIRQ1, HIRQ6, HIRQ9, HIRQ10, HIRQ11, and HIRQ12 711
18.6 Usage Note 714
18.6.1 Data Conflict 714
18.6.2 Module Stop Mode Setting 715
18.6.3 Operating Mode in LPC/FW Memory Write Cycle 715

Section 19 A/D Converter 717
19.1 Features 717
19.2 Input/Output Pins 719
19.3 Register Descriptions 720
19.3.1 A/D Data Registers A to D (ADDRA to ADDRD) 720
19.3.2 A/D Control/Status Register (ADCSR) 721
19.3.3 A/D Control Register (ADCR) 722
19.4 Operation 723
19.4.1 Single Mode 723
19.4.2 Scan Mode 723
19.4.3 Input Sampling and A/D Conversion Time 724
19.4.4 External Trigger Input Timing 726
19.5 Interrupt Source 727
19.6 A/D Conversion Accuracy Definitions 727
19.7 Usage Notes 729
19.7.1 Permissible Signal Source Impedance 729
19.7.2 Influences on Absolute Accuracy 729
19.7.3 Setting Range of Analog Power Supply and Other Pins 730
19.7.4 Notes on Board Design 730
19.7.5 Notes on Noise Countermeasures 730
19.7.6 Module Stop Mode Setting 731

Section 20 RAM 733

Section 21 Flash Memory (0.18-µm F-ZTAT Version) 735
21.1 Features 735
21.1.1 Mode Transitions 737
21.1.2 Mode Comparison 738
21.1.3 Flash Memory MAT Configuration 739
21.1.4 Block Division 739
21.1.5 Programming/Erasing Interface 742

21.2 Input/Output Pins 744
21.3 Register Descriptions 744
21.3.1 Programming/Erasing Interface Registers 746
21.3.2 Programming/Erasing Interface Parameters 753
21.4 On-Board Programming 764
21.4.1 Boot Mode 764
21.4.2 User Program Mode 768
21.4.3 User Boot Mode 779
21.4.4 Storable Areas for Procedure Program and Program Data 783
21.5 Protection 791
21.5.1 Hardware Protection 791
21.5.2 Software Protection 793
21.5.3 Error Protection 793
21.6 Switching between User MAT and User Boot MAT 795
21.7 Programmer Mode 796
21.8 Serial Communication Interface Specifications for Boot Mode 797
21.9 Usage Notes 824

Section 22 Boundary Scan (JTAG) 827
22.1 Features 827
22.2 Input/Output Pins 829
22.3 Register Descriptions 830
22.3.1 Instruction Register (SDIR) 830
22.3.2 Bypass Register (SDBPR) 832
22.3.3 Boundary Scan Register (SDBSR) 832
22.3.4 ID Code Register (SDIDR) 842
22.4 Operation 842
22.4.1 TAP Controller State Transitions 842
22.4.2 JTAG Reset 843
22.5 Boundary Scan 844
22.5.1 Supported Instructions 844
22.5.2 Notes 846
22.6 Usage Notes 846

Section 23 Clock Pulse Generator 849
23.1 Oscillator 850
23.1.1 Connecting Crystal Resonator 850
23.1.2 External Clock Input Method 851
23.2 Duty Correction Circuit 854
23.3 Medium-Speed Clock Divider 854

23.4 Bus Master Clock Select Circuit 854
23.5 Subclock Input Circuit 855
23.6 Subclock Waveform Forming Circuit 856
23.7 Clock Select Circuit 856
23.8 Handling of X1 and X2 Pins 857
23.9 Usage Notes 857
23.9.1 Notes on Resonator 857
23.9.2 Notes on Board Design 857

Section 24 Power-Down Modes 859
24.1 Register Descriptions 860
24.1.1 Standby Control Register (SBYCR) 860
24.1.2 Low-Power Control Register (LPWRCR) 862
24.1.3 Module Stop Control Registers H, L, and A (MSTPCRH, MSTPCRL, MSTPCRA) 864
24.2 Mode Transitions and LSI States 866
24.3 Medium-Speed Mode 870
24.4 Sleep Mode 871
24.5 Software Standby Mode 872
24.6 Hardware Standby Mode 874
24.7 Watch Mode 875
24.8 Subsleep Mode 876
24.9 Subactive Mode 877
24.10 Module Stop Mode 878
24.11 Direct Transitions 878
24.12 Usage Notes 879
24.12.1 I/O Port Status 879
24.12.2 Current Consumption when Waiting for Oscillation Stabilization 879
24.12.3 DTC Module Stop Mode 879

Section 25 List of Registers 881
25.1 Register Addresses (Address Order) 883
25.2 Register Bits 897
25.3 Register States in Each Operating Mode 909
25.4 Register Selection Condition 920
25.5 Register Addresses (Classification by Type of Module) 933

Section 26 Electrical Characteristics 947
26.1 Absolute Maximum Ratings 947
26.2 DC Characteristics 948

26.3 AC Characteristics 955
26.3.1 Clock Timing 956
26.3.2 Control Signal Timing 958
26.3.3 Timing of On-Chip Peripheral Modules 959
26.3.4 A/D Conversion Characteristics 971
26.4 Flash Memory Characteristics 972
26.5 Usage Notes 973

Appendix 975
A. I/O Port States in Each Pin State 975
B. Product Lineup 976
C. Package Dimensions 977

Index 979

Figures

Section 1 Overview
Figure 1.1 H8S/2114R Group Internal Block Diagram 3
Figure 1.2 H8S/2114R Group Pin Arrangement (TFP-144) 4
Figure 1.3 Sample Design of Reset Signals with no Affection Each Other 17

Section 2 CPU
Figure 2.1 Exception Vector Table (Normal Mode) 23
Figure 2.2 Stack Structure in Normal Mode 23
Figure 2.3 Exception Vector Table (Advanced Mode) 24
Figure 2.4 Stack Structure in Advanced Mode 25
Figure 2.5 Memory Map 26
Figure 2.6 CPU Internal Registers 27
Figure 2.7 Usage of General Registers 28
Figure 2.8 Stack 29
Figure 2.9 General Register Data Formats (1) 32
Figure 2.9 General Register Data Formats (2) 33
Figure 2.10 Memory Data Formats 34
Figure 2.11 Instruction Formats (Examples) 47
Figure 2.12 Branch Address Specification in Memory Indirect Addressing Mode 51
Figure 2.13 State Transitions 55

Section 3 MCU Operating Modes
Figure 3.1 Address Map 68

Section 4 Exception Handling
Figure 4.1 Reset Sequence (Mode 2) 75
Figure 4.2 Stack Status after Exception Handling 77
Figure 4.3 Operation when SP Value Is Odd 78

Section 5 Interrupt Controller
Figure 5.1 Block Diagram of Interrupt Controller 80
Figure 5.2 Relation between IRQ7 and IRQ6 Interrupts, KIN15 to KIN0 Interrupts, WUE7 to WUE0 Interrupts, KMIMR, KMIMRA, and WUEMRB (H8S/2140B Group Compatible Vector Mode: EIVS = 0) 95
Figure 5.3 Relation between IRQ7 and IRQ6 Interrupts, KIN15 to KIN0 Interrupts, WUE7 to WUE0 Interrupts, KMIMR, KMIMRA, and WUEMRB (Extended Vector Mode: EIVS = 1) 96
Figure 5.4 Block Diagram of Interrupts IRQ15 to IRQ0 100

Figure 5.5 Block Diagram of Interrupts KIN15 to KIN0 and WUE15 to WUE0 (Example of WUE15 to WUE8) 101
Figure 5.6 Block Diagram of Interrupt Control Operation 110
Figure 5.7 Flowchart of Procedure up to Interrupt Acceptance in Interrupt Control Mode 0 113
Figure 5.8 State Transition in Interrupt Control Mode 1 114
Figure 5.9 Flowchart of Procedure up to Interrupt Acceptance in Interrupt Control Mode 1 116
Figure 5.10 Interrupt Exception Handling 118
Figure 5.11 Interrupt Control for DTC 120
Figure 5.12 Block Diagram of Address Break Function 122
Figure 5.13 Examples of Address Break Timing 124
Figure 5.14 Conflict between Interrupt Generation and Disabling 125

Section 6 Bus Controller (BSC)
Figure 6.1 Block Diagram of BSC 129

Section 7 Data Transfer Controller (DTC)
Figure 7.1 Block Diagram of DTC 136
Figure 7.2 Block Diagram of DTC Activation Source Control 143
Figure 7.3 DTC Register Information Location in Address Space 144
Figure 7.4 DTC Operation Flowchart 147
Figure 7.5 Memory Mapping in Normal Mode 148
Figure 7.6 Memory Mapping in Repeat Mode 149
Figure 7.7 Memory Mapping in Block Transfer Mode 150
Figure 7.8 Chain Transfer Operation 151
Figure 7.9 DTC Operation Timing (Example in Normal Mode or Repeat Mode) 152
Figure 7.10 DTC Operation Timing (Example of Block Transfer Mode, with Block Size of 2) 153
Figure 7.11 DTC Operation Timing (Example of Chain Transfer) 153

Section 8 I/O Ports
Figure 8.1 Noise Cancel Circuit 184
Figure 8.2 Noise Cancel Operation 184

Section 9 8-Bit PWM Timer (PWM)
Figure 9.1 Block Diagram of PWM Timer 248
Figure 9.2 Example of Additional Pulse Timing (When Upper 4 Bits of PWDR = B'1000) 255
Figure 9.3 Example of PWM Setting 256
Figure 9.4 Example when PWM is Used as D/A Converter 256

Section 10 14-Bit PWM Timer (PWMX)
Figure 10.1 PWMX (D/A) Block Diagram 259
Figure 10.2 (1) DACNT Access Operation (1) [CPU → DACNT(H'AA57) Writing] 267
Figure 10.2 (2) DACNT Access Operation (2) [DACNT → CPU(H'AA57) Reading] 268

Figure 10.3 PWMX (D/A) Operation 269
Figure 10.4 Output Waveform (OS = 0, DADR corresponds to T_L) 272
Figure 10.5 Output Waveform (OS = 1, DADR corresponds to T_H) 273
Figure 10.6 D/A Data Register Configuration when CFS = 1 273
Figure 10.7 Output Waveform when DADR = H'0207 (OS = 1) 274

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

Figure 11.1 Block Diagram of 16-Bit Free-Running Timer 278
Figure 11.2 Example of Pulse Output 289
Figure 11.3 Increment Timing with Internal Clock Source 290
Figure 11.4 Increment Timing with External Clock Source 290
Figure 11.5 Timing of Output Compare A Output 291
Figure 11.6 Clearing of FRC by Compare-Match A Signal 291
Figure 11.7 Input Capture Input Signal Timing (Usual Case) 292
Figure 11.8 Input Capture Input Signal Timing (When ICRA to ICRD is Read) 292
Figure 11.9 Buffered Input Capture Timing 293
Figure 11.10 Buffered Input Capture Timing (BUFEA = 1) 294
Figure 11.11 Timing of Input Capture Flag (ICFA, ICFB, ICFC, or ICFD) Setting 294
Figure 11.12 Timing of Output Compare Flag (OCFA or OCFB) Setting 295
Figure 11.13 Timing of Overflow Flag (OVF) Setting 296
Figure 11.14 OCRA Automatic Addition Timing 296
Figure 11.15 Timing of Input Capture Mask Signal Setting 297
Figure 11.16 Timing of Input Capture Mask Signal Clearing 297
Figure 11.17 Conflict between FRC Write and Clear 299
Figure 11.18 Conflict between FRC Write and Increment 300
Figure 11.19 Conflict between OCR Write and Compare-Match (When Automatic Addition Function is Not Used) 301
Figure 11.20 Conflict between OCR Write and Compare-Match (When Automatic Addition Function is Used) 302

Section 12 16-Bit Timer Pulse Unit (TPU)

Figure 12.1 Block Diagram of TPU 306
Figure 12.2 16-Bit Register Access Operation [Bus Master ↔ TCNT (16 Bits)] 333
Figure 12.3 8-Bit Register Access Operation [Bus Master ↔ TCR (Upper 8 Bits)] 334
Figure 12.4 8-Bit Register Access Operation [Bus Master ↔ TMDR (Lower 8 Bits)] 334
Figure 12.5 8-Bit Register Access Operation [Bus Master ↔ TCR and TMDR (16 Bits)] 334
Figure 12.6 Example of Counter Operation Setting Procedure 335
Figure 12.7 Free-Running Counter Operation 336
Figure 12.8 Periodic Counter Operation 337
Figure 12.9 Example of Setting Procedure for Waveform Output by Compare Match 337
Figure 12.10 Example of 0 Output/1 Output Operation 338

Figure 12.11 Example of Toggle Output Operation 338
Figure 12.12 Example of Input Capture Operation Setting Procedure 339
Figure 12.13 Example of Input Capture Operation 340
Figure 12.14 Example of Synchronous Operation Setting Procedure 341
Figure 12.15 Example of Synchronous Operation 342
Figure 12.16 Compare Match Buffer Operation 343
Figure 12.17 Input Capture Buffer Operation 343
Figure 12.18 Example of Buffer Operation Setting Procedure 344
Figure 12.19 Example of Buffer Operation (1) 345
Figure 12.20 Example of Buffer Operation (2) 346
Figure 12.21 Example of PWM Mode Setting Procedure 348
Figure 12.22 Example of PWM Mode Operation (1) 349
Figure 12.23 Example of PWM Mode Operation (2) 350
Figure 12.24 Example of PWM Mode Operation (3) 351
Figure 12.25 Example of Phase Counting Mode Setting Procedure 352
Figure 12.26 Example of Phase Counting Mode 1 Operation 353
Figure 12.27 Example of Phase Counting Mode 2 Operation 354
Figure 12.28 Example of Phase Counting Mode 3 Operation 355
Figure 12.29 Example of Phase Counting Mode 4 Operation 356
Figure 12.30 Count Timing in Internal Clock Operation 360
Figure 12.31 Count Timing in External Clock Operation 360
Figure 12.32 Output Compare Output Timing 361
Figure 12.33 Input Capture Input Signal Timing 361
Figure 12.34 Counter Clear Timing (Compare Match) 362
Figure 12.35 Counter Clear Timing (Input Capture) 362
Figure 12.36 Buffer Operation Timing (Compare Match) 363
Figure 12.37 Buffer Operation Timing (Input Capture) 363
Figure 12.38 TGI Interrupt Timing (Compare Match) 364
Figure 12.39 TGI Interrupt Timing (Input Capture) 365
Figure 12.40 TCIV Interrupt Setting Timing 366
Figure 12.41 TCIU Interrupt Setting Timing 366
Figure 12.42 Timing for Status Flag Clearing by CPU 367
Figure 12.43 Timing for Status Flag Clearing by DTC Activation 367
Figure 12.44 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode 368
Figure 12.45 Conflict between TCNT Write and Clear Operations 369
Figure 12.46 Conflict between TCNT Write and Increment Operations 370
Figure 12.47 Conflict between TGR Write and Compare Match 370
Figure 12.48 Conflict between Buffer Register Write and Compare Match 371
Figure 12.49 Conflict between TGR Read and Input Capture 372
Figure 12.50 Conflict between TGR Write and Input Capture 373

Figure 12.51 Conflict between Buffer Register Write and Input Capture 374
Figure 12.52 Conflict between Overflow and Counter Clearing 375
Figure 12.53 Conflict between TCNT Write and Overflow 376

Section 13 8-Bit Timer (TMR)

Figure 13.1 Block Diagram of 8-Bit Timer (TMR_0 and TMR_1) 379
Figure 13.2 Block Diagram of 8-Bit Timer (TMR_Y and TMR_X) 380
Figure 13.3 Pulse Output Example 397
Figure 13.4 Count Timing for Internal Clock Input 398
Figure 13.5 Count Timing for External Clock Input (Both Edges) 398
Figure 13.6 Timing of CMF Setting at Compare-Match 399
Figure 13.7 Timing of Toggled Timer Output by Compare-Match A Signal 399
Figure 13.8 Timing of Counter Clear by Compare-Match 400
Figure 13.9 Timing of Counter Clear by External Reset Input 400
Figure 13.10 Timing of OVF Flag Setting 401
Figure 13.11 Timing of Input Capture Operation 404
Figure 13.12 Timing of Input Capture Signal
(Input capture signal is input during TICRR and TICRF read) 405
Figure 13.13 Conflict between TCNT Write and Clear 407
Figure 13.14 Conflict between TCNT Write and Count-Up 408
Figure 13.15 Conflict between TCOR Write and Compare-Match 409

Section 14 Watchdog Timer (WDT)

Figure 14.1 Block Diagram of WDT 414
Figure 14.2 Watchdog Timer Mode (RST/$\overline{\text{NMI}}$ = 1) Operation 421
Figure 14.3 Interval Timer Mode Operation 421
Figure 14.4 OVF Flag Set Timing 422
Figure 14.5 Output Timing of $\overline{\text{RESO}}$ signal 422
Figure 14.6 Writing to TCNT and TCSR (WDT_0) 424
Figure 14.7 Conflict between TCNT Write and Increment 425
Figure 14.8 Sample Circuit for Resetting the System by the $\overline{\text{RESO}}$ Signal 426

Section 15 Serial Communication Interface (SCI, IrDA)

Figure 15.1 Block Diagram of SCI 429
Figure 15.2 Data Format in Asynchronous Communication
(Example with 8-Bit Data, Parity, Two Stop Bits) 455
Figure 15.3 Receive Data Sampling Timing in Asynchronous Mode 457
Figure 15.4 Relation between Output Clock and Transmit Data Phase
(Asynchronous Mode) 458
Figure 15.5 Sample SCI Initialization Flowchart 459
Figure 15.6 Example of Operation in Transmission in Asynchronous Mode
(Example with 8-Bit Data, Parity, One Stop Bit) 460

Figure 15.7 Sample Serial Transmission Flowchart 461
Figure 15.8 Example of SCI Operation in Reception
(Example with 8-Bit Data, Parity, One Stop Bit) 462
Figure 15.9 Sample Serial Reception Flowchart (1) 464
Figure 15.9 Sample Serial Reception Flowchart (2) 465
Figure 15.10 Example of Communication Using Multiprocessor Format
(Transmission of Data H'AA to Receiving Station A) 467
Figure 15.11 Sample Multiprocessor Serial Transmission Flowchart 468
Figure 15.12 Example of SCI Operation in Reception
(Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit) 469
Figure 15.13 Sample Multiprocessor Serial Reception Flowchart (1) 470
Figure 15.13 Sample Multiprocessor Serial Reception Flowchart (2) 471
Figure 15.14 Data Format in Synchronous Communication (LSB-First) 472
Figure 15.15 Sample SCI Initialization Flowchart 473
Figure 15.16 Sample SCI Transmission Operation in Clocked Synchronous Mode 475
Figure 15.17 Sample Serial Transmission Flowchart 476
Figure 15.18 Example of SCI Receive Operation in Clocked Synchronous Mode 477
Figure 15.19 Sample Serial Reception Flowchart 478
Figure 15.20 Sample Flowchart of Simultaneous Serial Transmission and Reception 480
Figure 15.21 Pin Connection for Smart Card Interface 481
Figure 15.22 Data Formats in Normal Smart Card Interface Mode 482
Figure 15.23 Direct Convention (SDIR = SINV = O/$\overline{E}$ = 0) 482
Figure 15.24 Inverse Convention (SDIR = SINV = O/$\overline{E}$ = 1) 482
Figure 15.25 Receive Data Sampling Timing in Smart Card Interface Mode
(When Clock Frequency is 372 Times the Bit Rate) 484
Figure 15.26 Data Re-transfer Operation in SCI Transmission Mode 486
Figure 15.27 TEND Flag Set Timings during Transmission 486
Figure 15.28 Sample Transmission Flowchart 487
Figure 15.29 Data Re-transfer Operation in SCI Reception Mode 488
Figure 15.30 Sample Reception Flowchart 489
Figure 15.31 Clock Output Fixing Timing 490
Figure 15.32 Clock Stop and Restart Procedure 491
Figure 15.33 IrDA Block Diagram 492
Figure 15.34 IrDA Transmission and Reception 493
Figure 15.35 Sample Transmission using DTC in Clocked Synchronous Mode 499
Figure 15.36 Sample Flowchart for Mode Transition during Transmission 500
Figure 15.37 Pin States during Transmission in Asynchronous Mode (Internal Clock) 501
Figure 15.38 Pin States during Transmission in Clocked Synchronous Mode
(Internal Clock) 501
Figure 15.39 Sample Flowchart for Mode Transition during Reception 502

Figure 15.40 Switching from SCK Pins to Port Pins 503
Figure 15.41 Prevention of Low Pulse Output at Switching from SCK Pins to Port Pins 504

Section 16 I^2C Bus Interface (IIC)

Figure 16.1 Block Diagram of I^2C Bus Interface 507
Figure 16.2 I^2C Bus Interface Connections (Example: This LSI as Master) 508
Figure 16.3 I^2C Bus Data Format (I^2C Bus Format) 535
Figure 16.4 I^2C Bus Data Format (Serial Format) 535
Figure 16.5 I^2C Bus Timing 536
Figure 16.6 Sample Flowchart for IIC Initialization 537
Figure 16.7 Sample Flowchart for Operations in Master Transmit Mode 538
Figure 16.8 Example of Operation Timing in Master Transmit Mode (MLS = WAIT = 0) 540
Figure 16.9 Example of Stop Condition Issuance Operation Timing in Master Transmit Mode (MLS = WAIT = 0) 541
Figure 16.10 Sample Flowchart for Operations in Master Receive Mode (HNDS = 1) 542
Figure 16.11 Example of Operation Timing in Master Receive Mode (MLS = WAIT = 0, HNDS = 1) 544
Figure 16.12 Example of Stop Condition Issuance Operation Timing in Master Receive Mode (MLS = WAIT = 0, HNDS = 1) 544
Figure 16.13 Sample Flowchart for Operations in Master Receive Mode (receiving multiple bytes) (WAIT = 1) 546
Figure 16.14 Sample Flowchart for Operations in Master Receive Mode (receiving a single byte) (WAIT = 1) 547
Figure 16.15 Example of Master Receive Mode Operation Timing (MLS = ACKB = 0, WAIT = 1) 550
Figure 16.16 Example of Stop Condition Issuance Timing in Master Receive Mode (MLS = ACKB = 0, WAIT = 1) 550
Figure 16.17 Sample Flowchart for Operations in Slave Receive Mode (HNDS = 1) 552
Figure 16.18 Example of Slave Receive Mode Operation Timing (1) (MLS = 0, HNDS= 1) 554
Figure 16.19 Example of Slave Receive Mode Operation Timing (2) (MLS = 0, HNDS= 1) 555
Figure 16.20 Sample Flowchart for Operations in Slave Receive Mode (HNDS = 0) 556
Figure 16.21 Example of Slave Receive Mode Operation Timing (1) (MLS = ACKB = 0, HNDS = 0) 558
Figure 16.22 Example of Slave Receive Mode Operation Timing (2) (MLS = ACKB = 0, HNDS = 0) 558
Figure 16.23 Sample Flowchart for Slave Transmit Mode 559
Figure 16.24 Example of Slave Transmit Mode Operation Timing (MLS = 0) 561
Figure 16.25 IRIC Setting Timing and SCL Control (1) 562
Figure 16.26 IRIC Setting Timing and SCL Control (2) 563
Figure 16.27 IRIC Setting Timing and SCL Control (3) 564
Figure 16.28 Block Diagram of Noise Canceller 567

Figure 16.29 Notes on Reading Master Receive Data 573
Figure 16.30 Flowchart for Start Condition Issuance Instruction for Retransmission and Timing 574
Figure 16.31 Stop Condition Issuance Timing 575
Figure 16.32 IRIC Flag Clearing Timing when WAIT = 1 576
Figure 16.33 ICDR Read and ICCR Access Timing in Slave Transmit Mode 577
Figure 16.34 TRS Bit Set Timing in Slave Mode 578
Figure 16.35 Diagram of Erroneous Operation when Arbitration is Lost 580

Section 17 Keyboard Buffer Control Unit (KBU)

Figure 17.1 Block Diagram of KBU 582
Figure 17.2 KBU Connection 583
Figure 17.3 Sample Receive Processing Flowchart 594
Figure 17.4 Receive Timing 595
Figure 17.5 Sample Transmit Processing Flowchart 596
Figure 17.6 Transmit Timing 597
Figure 17.7 (1) Sample Receive Abort Processing Flowchart 598
Figure 17.7 (2) Sample Receive Abort Processing Flowchart 599
Figure 17.8 Receive Abort and Transmit Start (Transmission/Reception Switchover) Timing 599
Figure 17.9 KCLKI and KDI Read Timing 600
Figure 17.10 KCLKO and KDO Write Timing 601
Figure 17.11 KBF Setting and KCLK Automatic I/O Inhibit Generation Timing 602
Figure 17.12 Receive Counter and KBBR Data Load Timing 603
Figure 17.13 Receive Timing and KCLK 604
Figure 17.14 Example of KCLK Input Fall Interrupt Operation 605
Figure 17.15 Timing of First KCLK Interrupt 606
Figure 17.16 First KCLK Interrupt Path 608
Figure 17.17 Interrupt Timing in Software Standby Mode, Watch Mode, and Subsleep Mode. 609
Figure 17.18 Internal Flag of First KCLK Falling Interrupt in Software Standby mode, Watch mode, and Subsleep mode 610
Figure 17.19 KBIOE Setting and KCLK Falling Edge Detection Timing 611
Figure 17.20 KDO Output 612

Section 18 LPC Interface (LPC)

Figure 18.1 Block Diagram of LPC 615
Figure 18.2 Typical $\overline{\text{LFRAME}}$ Timing 679
Figure 18.3 Abort Mechanism 679
Figure 18.4 GA20 Output 681
Figure 18.5 Power-Down State Termination Timing 686
Figure 18.6 SERIRQ Timing 687

Figure 18.7 Clock Start Request Timing 689
Figure 18.8 Example of Command Space Setting 692
Figure 18.9 Example of Flash Memory Address Translation 700
Figure 18.10 Example of On-Chip RAM Address Translation 701
Figure 18.11 Example 1 of Address Space Priority 703
Figure 18.12 Example 2 of Address Space Priority 704
Figure 18.13 Flash Memory Protection 706
Figure 18.14 Protected Address Space in On-Chip RAM 707
Figure 18.15 Example of Programming Flash Memory 708
Figure 18.16 Example of Erasing Flash Memory 709
Figure 18.17 HIRQ Flowchart (Example of Channel 1) 713

Section 19 A/D Converter

Figure 19.1 Block Diagram of A/D Converter 718
Figure 19.2 A/D Conversion Timing 725
Figure 19.3 External Trigger Input Timing 726
Figure 19.4 A/D Conversion Accuracy Definitions 728
Figure 19.5 A/D Conversion Accuracy Definitions 728
Figure 19.6 Example of Analog Input Circuit 729
Figure 19.7 Example of Analog Input Protection Circuit 731
Figure 19.8 Analog Input Pin Equivalent Circuit 731

Section 21 Flash Memory (0.18-µm F-ZTAT Version)

Figure 21.1 Block Diagram of Flash Memory 736
Figure 21.2 Mode Transition for Flash Memory 737
Figure 21.3 Flash Memory Configuration 739
Figure 21.4 Block Division of User MAT (1) 740
Figure 21.4 Block Division of User MAT (2) 741
Figure 21.5 Overview of User Procedure Program 742
Figure 21.6 System Configuration in Boot Mode 765
Figure 21.7 Automatic-Bit-Rate Adjustment Operation of SCI 765
Figure 21.8 Overview of Boot Mode State Transition Diagram 767
Figure 21.9 Programming/Erasing Overview Flow 768
Figure 21.10 RAM Map when Programming/Erasing is Executed 769
Figure 21.11 Programming Procedure 770
Figure 21.12 Erasing Procedure 776
Figure 21.13 Repeating Procedure of Erasing and Programming 778
Figure 21.14 Procedure for Programming User MAT in User Boot Mode 780
Figure 21.15 Procedure for Erasing User MAT in User Boot Mode 782
Figure 21.16 Transitions to Error-Protection State 794
Figure 21.17 Switching between User MAT and User Boot MAT 795

Figure 21.18 Memory Map in Programmer Mode 796
Figure 21.19 Boot Program States 798
Figure 21.20 Bit-Rate-Adjustment Sequence 799
Figure 21.21 Communication Protocol Format 800
Figure 21.22 Sequence of New Bit Rate Selection 811
Figure 21.23 Programming Sequence 814
Figure 21.24 Erasure Sequence 818

Section 22 Boundary Scan (JTAG)

Figure 22.1 JTAG Block Diagram 828
Figure 22.2 TAP Controller State Transitions 843
Figure 22.3 Reset Signal Circuit without Reset Signal Interference 847
Figure 22.4 Serial Data Input/Output (1) 848
Figure 22.5 Serial Data Input/Output (2) 848

Section 23 Clock Pulse Generator

Figure 23.1 Block Diagram of Clock Pulse Generator 849
Figure 23.2 Typical Connection to Crystal Resonator 850
Figure 23.3 Equivalent Circuit of Crystal Resonator 850
Figure 23.4 Example of External Clock Input 851
Figure 23.5 External Clock Input Timing 852
Figure 23.6 Timing of External Clock Output Stabilization Delay Time 853
Figure 23.7 Subclock Input from EXCL Pin and ExEXCL Pin 855
Figure 23.8 Subclock Input Timing 856
Figure 23.9 Handling of X1 and X2 Pins 857
Figure 23.10 Note on Board Design of Oscillator Section 857

Section 24 Power-Down Modes

Figure 24.1 Mode Transition Diagram 867
Figure 24.2 Medium-Speed Mode Timing 871
Figure 24.3 Software Standby Mode Application Example 873
Figure 24.4 Hardware Standby Mode Timing 874

Section 26 Electrical Characteristics

Figure 26.1 Darlington Transistor Drive Circuit (Example) 954
Figure 26.2 LED Drive Circuit (Example) 954
Figure 26.3 Output Load Circuit 955
Figure 26.4 System Clock Timing 956
Figure 26.5 Oscillation Stabilization Timing 957
Figure 26.6 Oscillation Stabilization Timing (Exiting Software Standby Mode) 957
Figure 26.7 Reset Input Timing 958
Figure 26.8 Interrupt Input Timing 959

Figure 26.9 I/O Port Input/Output Timing 961
Figure 26.10 FRT Input/Output Timing 961
Figure 26.11 FRT Clock Input Timing 961
Figure 26.12 TPU Input/Output Timing 962
Figure 26.13 TPU Clock Input Timing 962
Figure 26.14 8-Bit Timer Output Timing 962
Figure 26.15 8-Bit Timer Clock Input Timing 962
Figure 26.16 8-Bit Timer Reset Input Timing 963
Figure 26.17 PWM, PWMX Output Timing 963
Figure 26.18 SCK Clock Input Timing 963
Figure 26.19 SCI Input/Output Timing (Clock Synchronous Mode) 963
Figure 26.20 A/D Converter External Trigger Input Timing 964
Figure 26.21 WDT Output Timing ($\overline{\text{RESO}}$) 964
Figure 26.22 Keyboard Buffer Control Unit Timing 965
Figure 26.23 I^2C Bus Interface Input/Output Timing 967
Figure 26.24 LPC Interface Timing 968
Figure 26.25 Test Conditions for Tester 968
Figure 26.26 JTAG ETCK Timing 969
Figure 26.27 Reset Hold Timing 970
Figure 26.28 JTAG Input/Output Timing 970
Figure 26.29 Connection of VCL Capacitor 973

Appendix

Figure C.1 Package Dimensions (TFP-144) 977

Tables

Section 1 Overview
Table 1.1 H8S/2114R Group Pin Arrangement in Each Operating Mode 5
Table 1.2 Pin Functions 10

Section 2 CPU
Table 2.1 Instruction Classification 35
Table 2.2 Operation Notation 36
Table 2.3 Data Transfer Instructions 37
Table 2.4 Arithmetic Operations Instructions (1) 38
Table 2.4 Arithmetic Operations Instructions (2) 39
Table 2.5 Logic Operations Instructions 40
Table 2.6 Shift Instructions 41
Table 2.7 Bit Manipulation Instructions (1) 42
Table 2.7 Bit Manipulation Instructions (2) 43
Table 2.8 Branch Instructions 44
Table 2.9 System Control Instructions 45
Table 2.10 Block Data Transfer Instructions 46
Table 2.11 Addressing Modes 48
Table 2.12 Absolute Address Access Ranges 50
Table 2.13 Effective Address Calculation (1) 52
Table 2.13 Effective Address Calculation (2) 53

Section 3 MCU Operating Modes
Table 3.1 MCU Operating Mode Selection 59

Section 4 Exception Handling
Table 4.1 Exception Types and Priority 69
Table 4.2 Exception Handling Vector Table (H8S/2140B Group Compatible Vector Mode) 70
Table 4.3 Exception Handling Vector Table (Extended Vector Mode) 72
Table 4.4 Status of CCR after Trap Instruction Exception Handling 76

Section 5 Interrupt Controller
Table 5.1 Pin Configuration 81
Table 5.2 Correspondence between Interrupt Source and ICR (H8S/2140B Group Compatible Vector Mode: EIVS = 0) 83
Table 5.3 Correspondence between Interrupt Source and ICR (Extended Vector Mode: EIVS = 1) 84

Table 5.4 Interrupt Sources, Vector Addresses, and Interrupt Priorities (H8S/2140B Group Compatible Vector Mode) 103
Table 5.5 Interrupt Sources, Vector Addresses, and Interrupt Priorities (Extended Vector Mode) 106
Table 5.6 Interrupt Control Modes 109
Table 5.7 Interrupts Selected in Each Interrupt Control Mode 111
Table 5.8 Operations and Control Signal Functions in Each Interrupt Control Mode 112
Table 5.9 Interrupt Response Times 119
Table 5.10 Interrupt Source Selection and Clearing Control 121

Section 7 Data Transfer Controller (DTC)
Table 7.1 Correspondence between Interrupt Sources and DTCER 141
Table 7.2 Interrupt Sources, DTC Vector Addresses, and Corresponding DTCEs 145
Table 7.3 Register Functions in Normal Mode 148
Table 7.4 Register Functions in Repeat Mode 149
Table 7.5 Register Functions in Block Transfer Mode 150
Table 7.6 DTC Execution Status 154
Table 7.7 Number of States Required for Each Execution Status 154

Section 8 I/O Ports
Table 8.1 Port Functions 159
Table 8.2 Port 1 Input Pull-Up MOS States 166
Table 8.3 Port 2 Input Pull-Up MOS States 170
Table 8.4 Port 3 Input Pull-Up MOS States 173
Table 8.5 Port 6 Input Pull-Up MOS States 188
Table 8.6 Port 9 Input Pull-Up MOS States 200
Table 8.7 Port B Input Pull-Up MOS States 208
Table 8.8 Port C Input Pull-Up MOS States 216
Table 8.9 Port D Input Pull-Up MOS States 224
Table 8.10 Port E Input Pull-Up MOS States 226
Table 8.11 Port F Input Pull-Up MOS States 232

Section 9 8-Bit PWM Timer (PWM)
Table 9.1 Pin Configuration 249
Table 9.2 Internal Clock Selection 251
Table 9.3 Resolution, PWM Conversion Period, and Carrier Frequency when ϕ = 20 MHz 251
Table 9.4 Duty Cycle of Basic Pulse 254
Table 9.5 Position of Pulses Added to Basic Pulses 255

Section 10 14-Bit PWM Timer (PWMX)

Table 10.1 Pin Configuration 260
Table 10.2 Clock Select of PWMX 265
Table 10.3 Reading/Writing to 16-bit Registers 267
Table 10.4 Settings and Operation (Examples when ϕ = 20 MHz) 270
Table 10.5 Locations of Additional Pulses Added to Base Pulse (When CFS = 1) 275

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

Table 11.1 Pin Configuration 279
Table 11.2 FRT Interrupt Sources 298
Table 11.3 Switching of Internal Clock and FRC Operation 303

Section 12 16-Bit Timer Pulse Unit (TPU)

Table 12.1 TPU Functions 307
Table 12.2 Pin Configuration 309
Table 12.3 CCLR2 to CCLR0 (channel 0) 312
Table 12.4 CCLR2 to CCLR0 (channels 1 and 2) 312
Table 12.5 TPSC2 to TPSC0 (channel 0) 313
Table 12.6 TPSC2 to TPSC0 (channel 1) 313
Table 12.7 TPSC2 to TPSC0 (channel 2) 314
Table 12.8 MD3 to MD0 316
Table 12.9 TIORH_0 (channel 0) 318
Table 12.10 TIORH_0 (channel 0) 319
Table 12.11 TIORL_0 (channel 0) 320
Table 12.12 TIORL_0 (channel 0) 321
Table 12.13 TIOR_1 (channel 1) 322
Table 12.14 TIOR_1 (channel 1) 323
Table 12.15 TIOR_2 (channel 2) 324
Table 12.16 TIOR_2 (channel 2) 325
Table 12.17 Register Combinations in Buffer Operation 343
Table 12.18 PWM Output Registers and Output Pins 348
Table 12.19 Phase Counting Mode Clock Input Pins 352
Table 12.20 Up/Down-Count Conditions in Phase Counting Mode 1 353
Table 12.21 Up/Down-Count Conditions in Phase Counting Mode 2 354
Table 12.22 Up/Down-Count Conditions in Phase Counting Mode 3 355
Table 12.23 Up/Down-Count Conditions in Phase Counting Mode 4 356
Table 12.24 TPU Interrupts 358

Section 13 8-Bit Timer (TMR)

Table 13.1 Pin Configuration 381
Table 13.2 Clock Input to TCNT and Count Condition (1) 386

Table 13.2 Clock Input to TCNT and Count Condition (2) 387
Table 13.3 Registers Accessible by TMR_X/TMR_Y 396
Table 13.4 Input Capture Signal Selection 405
Table 13.5 Interrupt Sources of 8-Bit Timers TMR_0, TMR_1, TMR_Y, and TMR_X 406
Table 13.6 Timer Output Priorities 410
Table 13.7 Switching of Internal Clocks and TCNT Operation 411

Section 14 Watchdog Timer (WDT)
Table 14.1 Pin Configuration 415
Table 14.2 WDT Interrupt Source 423

Section 15 Serial Communication Interface (SCI, IrDA)
Table 15.1 Pin Configuration 430
Table 15.2 Relationships between N Setting in BRR and Bit Rate B 445
Table 15.3 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (1) 446
Table 15.3 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (2) 448
Table 15.4 Maximum Bit Rate for Each Frequency (Asynchronous Mode) 450
Table 15.5 Maximum Bit Rate with External Clock Input (Asynchronous Mode) 450
Table 15.6 BRR Settings for Various Bit Rates (Clocked Synchronous Mode) 451
Table 15.7 Maximum Bit Rate with External Clock Input (Clocked Synchronous Mode) 452
Table 15.8 BRR Settings for Various Bit Rates (Smart Card Interface Mode, n = 0, s = 372) 452
Table 15.9 Maximum Bit Rate for Each Frequency (Smart Card Interface Mode, S = 372) 452
Table 15.10 Serial Transfer Formats (Asynchronous Mode) 456
Table 15.11 SSR Status Flags and Receive Data Handling 463
Table 15.12 IrCKS2 to IrCKS0 Bit Settings 495
Table 15.13 SCI Interrupt Sources 496
Table 15.14 SCI Interrupt Sources 497

Section 16 I^2C Bus Interface (IIC)
Table 16.1 Pin Configuration 509
Table 16.2 Communication Format 513
Table 16.3 I^2C Transfer Rate 516
Table 16.4 Flags and Transfer States (Master Mode) 522
Table 16.5 Flags and Transfer States (Slave Mode) 524
Table 16.6 I^2C Bus Data Format Symbols 536
Table 16.7 Operation by Using DTC 566
Table 16.8 IIC Interrupt Sources 569
Table 16.9 I^2C Bus Timing (SCL and SDA Outputs) 570
Table 16.10 Permissible SCL Rise Time (t_{sr}) Values 571
Table 16.11 I^2C Bus Timing (with Maximum Influence of t_{Sr}/t_{Sf}) 572

Section 17 Keyboard Buffer Control Unit (KBU)
Table 17.1 Pin Configuration ... 584

Section 18 LPC Interface (LPC)
Table 18.1 Pin Configuration ... 616
Table 18.2 LPC I/O Cycle ... 678
Table 18.3 GA20 (P81) Setting/Clearing Timing ... 680
Table 18.4 Fast Gate A20 Output Signals ... 682
Table 18.5 Scope of LPC Interface Pin Shutdown ... 684
Table 18.6 Scope of Initialization in Each LPC interface Mode ... 685
Table 18.7 Serialized Interrupt Transfer Cycle Frame Configuration ... 688
Table 18.8 LPC Memory Cycle ... 690
Table 18.9 FW Memory Cycle (Byte Transfer) ... 691
Table 18.10 List of LPC/FW Memory Access Commands ... 693
Table 18.11 List of Factors that Prevents SYNC Field being Sent Back ... 697
Table 18.12 Receive Complete Interrupts and Error Interrupt ... 710
Table 18.13 HIRQ Setting and Clearing Conditions ... 712
Table 18.14 Host Address Example ... 715

Section 19 A/D Converter
Table 19.1 Pin Configuration ... 719
Table 19.2 Analog Input Channels and Corresponding ADDR ... 720
Table 19.3 A/D Conversion Time (Single Mode) ... 726
Table 19.4 A/D Converter Interrupt Source ... 727

Section 21 Flash Memory (0.18-µm F-ZTAT Version)
Table 21.1 Comparison of Programming Modes ... 738
Table 21.2 Pin Configuration ... 744
Table 21.3 Register/Parameter and Target Mode ... 745
Table 21.4 Parameters and Target Modes ... 754
Table 21.5 On-Board Programming Mode Setting ... 764
Table 21.6 System Clock Frequency for Automatic-Bit-Rate Adjustment by This LSI ... 766
Table 21.7 Executable MAT ... 784
Table 21.8 (1) Usable Area for Programming in User Program Mode ... 785
Table 21.8 (2) Usable Area for Erasure in User Program Mode ... 787
Table 21.8 (3) Usable Area for Programming in User Boot Mode ... 788
Table 21.8 (4) Usable Area for Erasure in User Boot Mode ... 790
Table 21.9 Hardware Protection ... 792
Table 21.10 Software Protection ... 793
Table 21.11 Inquiry and Selection Commands ... 801
Table 21.12 Programming/Erasing Commands ... 813
Table 21.13 Status Code ... 823

Table 21.14 Error Code 823

Section 22 Boundary Scan (JTAG)
Table 22.1 Pin Configuration 829
Table 22.2 JTAG Register Serial Transfer 830
Table 22.3 Correspondence between Pins and Boundary Scan Register 832

Section 23 Clock Pulse Generator
Table 23.1 Damping Resistor Values 850
Table 23.2 Crystal Resonator Parameters 851
Table 23.3 External Clock Input Conditions 852
Table 23.4 External Clock Output Stabilization Delay Time 853
Table 23.5 Subclock Input Conditions 855

Section 24 Power-Down Modes
Table 24.1 Operating Frequency and Wait Time 862
Table 24.2 LSI Internal States in Each Operating Mode 868

Section 26 Electrical Characteristics
Table 26.1 Absolute Maximum Ratings 947
Table 26.2 DC Characteristics (1) 948
Table 26.2 DC Characteristics (2) 950
Table 26.2 DC Characteristics (3) Using LPC Function 951
Table 26.3 Permissible Output Currents 952
Table 26.4 Bus Drive Characteristics 953
Table 26.5 Clock Timing 956
Table 26.6 Control Signal Timing 958
Table 26.7 Timing of On-Chip Peripheral Modules 960
Table 26.8 KBU Bus Timing 964
Table 26.9 I^2C Bus Timing 966
Table 26.10 LPC Timing 967
Table 26.11 JTAG Timing 969
Table 26.12 A/D Conversion Characteristics (AN7 to AN0 Input: 134/266-State Conversion) 971
Table 26.13 Flash Memory Characteristics 972

Appendix
Table A.1 I/O Port States in Each Pin State 975

Section 1 Overview

1.1 Overview

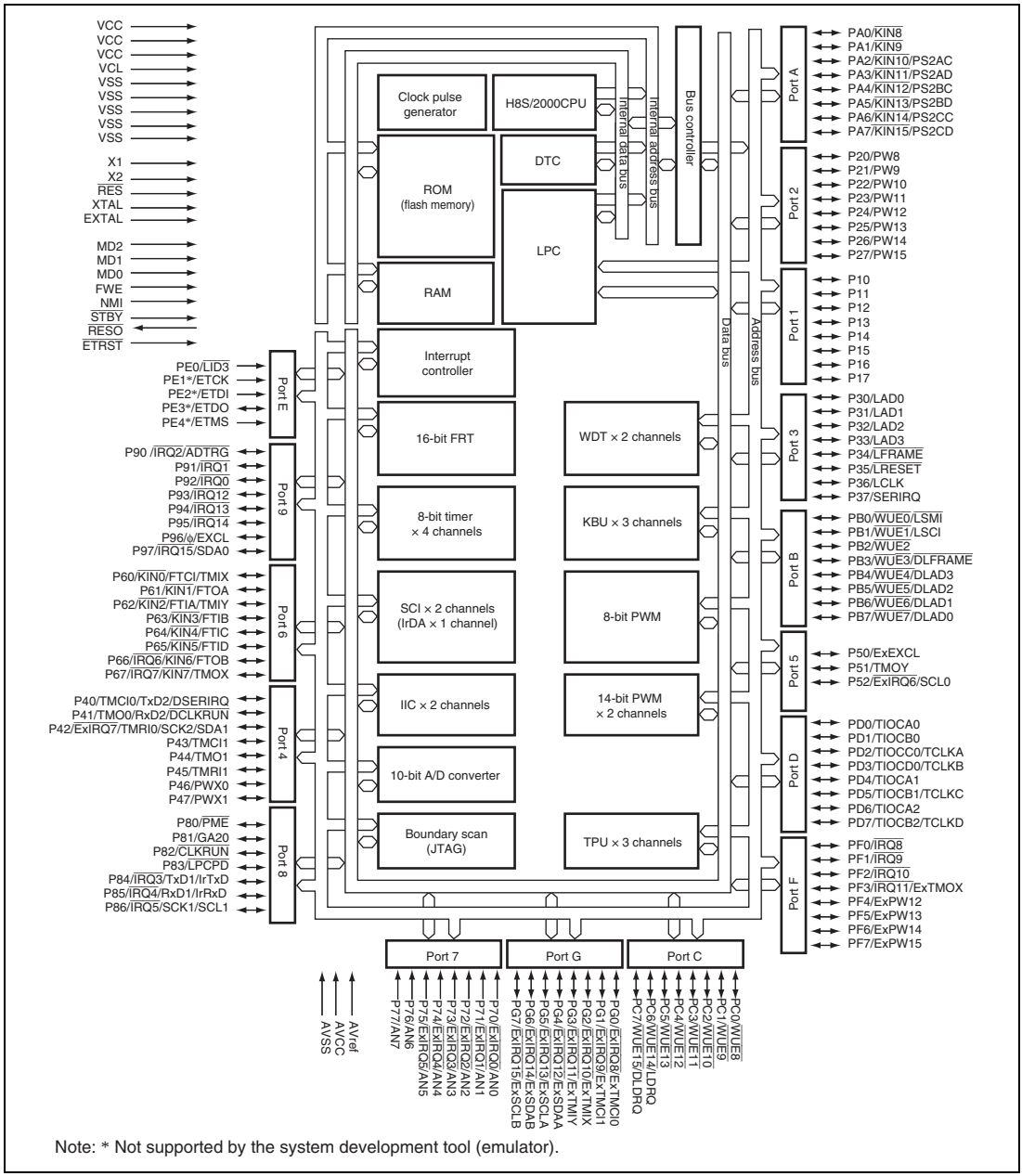
- 16-bit high-speed H8S/2000 CPU
  Upward-compatible with the H8/300 and H8/300H CPUs on an object level
  Sixteen 16-bit general registers
  65 basic instructions
- Various peripheral functions
  Data transfer controller (DTC)
  8-bit PWM timer (PWM)
  14-bit PWM timer (PWMX)
  16-bit timer pulse unit (TPU)
  16-bit free-running timer (FRT)
  8-bit timer (TMR)
  Watchdog timer (WDT)
  Asynchronous or clocked synchronous serial communication interface (SCI)
  I^2C bus interface (IIC)
  Keyboard buffer control unit (KBU)
  LPC interface (LPC)
  10-bit A/D converter
  Boundary scan (JTAG)
  Clock pulse generator
- On-chip memory

ROM Type	Model	ROM	RAM	Remarks
Flash memory version	R4F2114R	1 Mbyte	8 kbytes	Being developed

- General I/O ports
  I/O pins: 106
  Input-only pins: 13
- Supports various power-down states
- Compact package

Package	Code	Body Size	Pin Pitch
TQFP-144	TFP-144	16.0 × 16.0 mm	0.4 mm

1.2 Internal Block Diagram

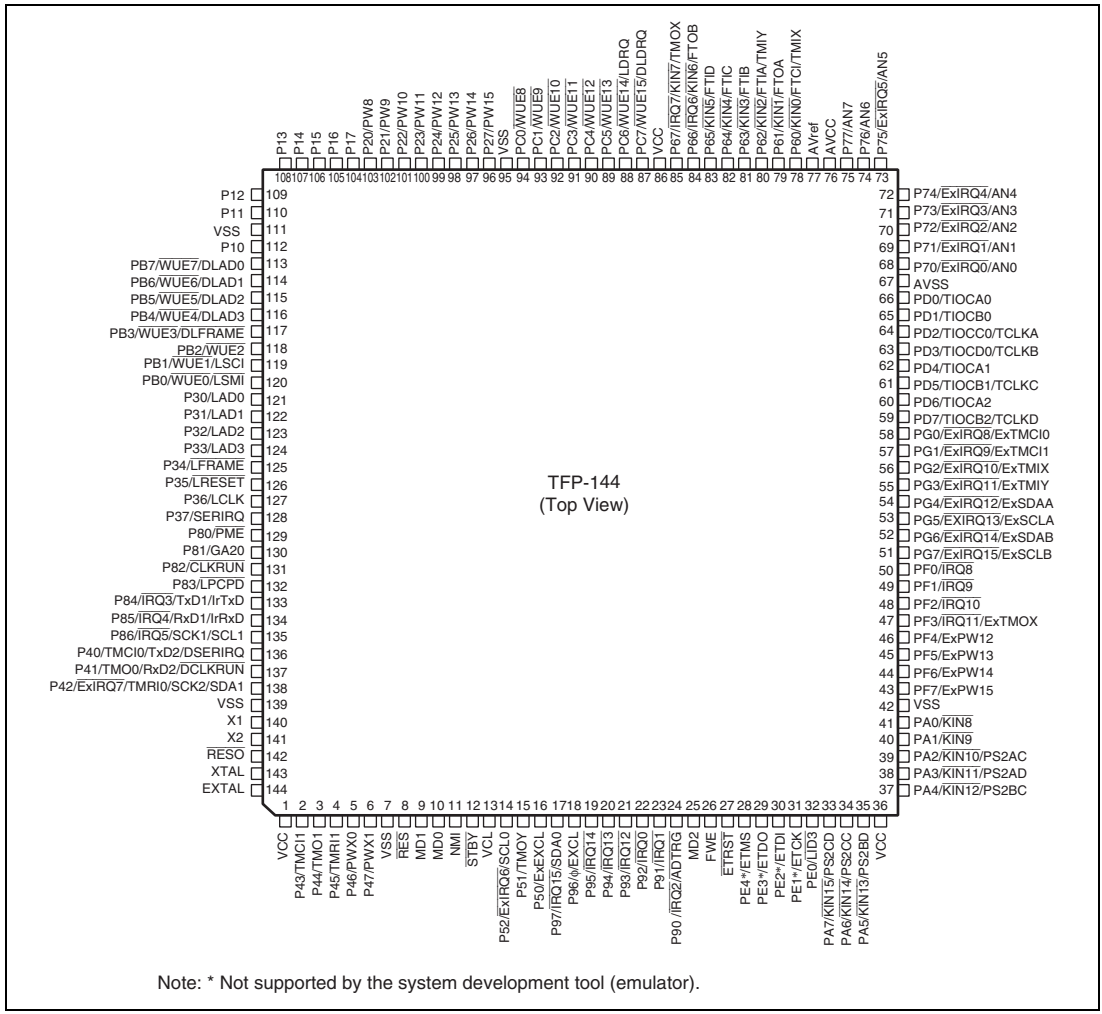


Note: * Not supported by the system development tool (emulator).

Figure 1.1 H8S/2114R Group Internal Block Diagram

1.3 Pin Description

1.3.1 Pin Arrangement



Note: * Not supported by the system development tool (emulator).

Figure 1.2 H8S/2114R Group Pin Arrangement (TFP-144)

1.3.2 Pin Arrangement in Each Operating Mode

Table 1.1 H8S/2114R Group Pin Arrangement in Each Operating Mode

Pin No.	Pin Name	
	Single-Chip Mode	
TFP-144	Mode 2 and Mode 3 (EXPE = 0)	Flash Memory Programmer Mode
1	VCC	VCC
2	P43/TMCI1	NC
3	P44/TMO1	NC
4	P45/TMRI1	NC
5	P46/PWX0	NC
6	P47/PWX1	NC
7	VSS	VSS
8	$\overline{\text{RES}}$	$\overline{\text{RES}}$
9	MD1	VSS
10	MD0	VSS
11	NMI	FA9
12	$\overline{\text{STBY}}$	VCC
13	VCL	VCL
14 (N)	P52/$\overline{\text{ExIRQ6}}$/SCL0	FA18
15	P51/TMOY	FA17
16	P50/ExEXCL	FA19
17 (N)	P97/$\overline{\text{IRQ15}}$/SDA0	VCC
18	P96/ϕ/EXCL	NC
19	P95/$\overline{\text{IRQ14}}$	FA16
20	P94/$\overline{\text{IRQ13}}$	FA15
21	P93/$\overline{\text{IRQ12}}$	$\overline{\text{WE}}$
22	P92/$\overline{\text{IRQ0}}$	VSS
23	P91/$\overline{\text{IRQ1}}$	VCC
24	P90/$\overline{\text{IRQ2}}$/$\overline{\text{ADTRG}}$	VCC
25	MD2	VSS
26	FWE	FWE
27	$\overline{\text{ETRST}}$	$\overline{\text{RES}}$

Pin No.	Pin Name	
	Single-Chip Mode	
TFP-144	Mode 2 and Mode 3 (EXPE = 0)	Flash Memory Programmer Mode
28	PE4*/ETMS	NC
29	PE3*/ETDO	NC
30	PE2*/ETDI	NC
31	PE1*/ETCK	NC
32	PE0/$\overline{\text{LID3}}$	NC
33 (N)	PA7/$\overline{\text{KIN15}}$/PS2CD	NC
34 (N)	PA6/$\overline{\text{KIN14}}$/PS2CC	NC
35 (N)	PA5/$\overline{\text{KIN13}}$/PS2BD	NC
36	VCC	VCC
37 (N)	PA4/$\overline{\text{KIN12}}$/PS2BC	NC
38 (N)	PA3/$\overline{\text{KIN11}}$/PS2AD	NC
39 (N)	PA2/$\overline{\text{KIN10}}$/PS2AC	NC
40 (N)	PA1/$\overline{\text{KIN9}}$	NC
41 (N)	PA0/$\overline{\text{KIN8}}$	NC
42	VSS	VSS
43	PF7/ExPW15	NC
44	PF6/ExPW14	NC
45	PF5/ExPW13	NC
46	PF4/ExPW12	NC
47	PF3/$\overline{\text{IRQ11}}$/ExTMOX	NC
48	PF2/$\overline{\text{IRQ10}}$	NC
49	PF1/$\overline{\text{IRQ9}}$	NC
50	PF0/$\overline{\text{IRQ8}}$	NC
51 (N)	PG7/$\overline{\text{ExIRQ15}}$/ExSCLB	NC
52 (N)	PG6/$\overline{\text{ExIRQ14}}$/ExSDAB	NC
53 (N)	PG5/$\overline{\text{ExIRQ13}}$/ExSCLA	NC
54 (N)	PG4/$\overline{\text{ExIRQ12}}$/ExSDAA	NC
55 (N)	PG3/$\overline{\text{ExIRQ11}}$/ExTMIY	NC
56 (N)	PG2/$\overline{\text{ExIRQ10}}$/ExTMIX	NC
57 (N)	PG1/$\overline{\text{ExIRQ9}}$/ExTMCI1	NC

Pin No.	Pin Name	
	Single-Chip Mode	
TFP-144	Mode 2 and Mode 3 (EXPE = 0)	Flash Memory Programmer Mode
58 (N)	PG0/$\overline{\text{ExIRQ8}}$/ExTMCI0	NC
59	PD7/TIOCB2/TCLKD	NC
60	PD6/TIOCA2	NC
61	PD5/TIOCB1/TCLKC	NC
62	PD4/TIOCA1	NC
63	PD3/TIOCD0/TCLKB	NC
64	PD2/TIOCC0/TCLKA	NC
65	PD1/TIOCB0	NC
66	PD0/TIOCA0	NC
67	AVSS	VSS
68	P70/$\overline{\text{ExIRQ0}}$/AN0	NC
69	P71/$\overline{\text{ExIRQ1}}$/AN1	NC
70	P72/$\overline{\text{ExIRQ2}}$/AN2	NC
71	P73/$\overline{\text{ExIRQ3}}$/AN3	NC
72	P74/$\overline{\text{ExIRQ4}}$/AN4	NC
73	P75/$\overline{\text{ExIRQ5}}$/AN5	NC
74	P76/AN6	NC
75	P77/AN7	NC
76	AVCC	VCC
77	AVref	VCC
78	P60/FTCI/$\overline{\text{KIN0}}$/TMIX	NC
79	P61/FTOA/$\overline{\text{KIN1}}$	NC
80	P62/FTIA/$\overline{\text{KIN2}}$/TMIY	NC
81	P63/FTIB/$\overline{\text{KIN3}}$	NC
82	P64/FTIC/$\overline{\text{KIN4}}$	NC
83	P65/FTID/$\overline{\text{KIN5}}$	NC
84	P66/$\overline{\text{IRQ6}}$/FTOB/$\overline{\text{KIN6}}$	NC
85	P67/$\overline{\text{IRQ7}}$/TMOX/$\overline{\text{KIN7}}$	VSS
86	VCC	VCC
87	PC7/$\overline{\text{WUE15}}$/DLDRQ	NC

Pin No.	Pin Name	
	Single-Chip Mode	
TFP-144	Mode 2 and Mode 3 (EXPE = 0)	Flash Memory Programmer Mode
88	PC6/$\overline{\text{WUE14}}$/LDRQ	NC
89	PC5/$\overline{\text{WUE13}}$	NC
90	PC4/$\overline{\text{WUE12}}$	NC
91	PC3/$\overline{\text{WUE11}}$	NC
92	PC2/$\overline{\text{WUE10}}$	NC
93	PC1/$\overline{\text{WUE9}}$	NC
94	PC0/$\overline{\text{WUE8}}$	NC
95	VSS	VSS
96	P27/PW15	$\overline{\text{CE}}$
97	P26/PW14	FA14
98	P25/PW13	FA13
99	P24/PW12	FA12
100	P23/PW11	FA11
101	P22/PW10	FA10
102	P21/PW9	$\overline{\text{OE}}$
103	P20/PW8	FA8
104	P17	FA7
105	P16	FA6
106	P15	FA5
107	P14	FA4
108	P13	FA3
109	P12	FA2
110	P11	FA1
111	VSS	VSS
112	P10	FA0
113	PB7/$\overline{\text{WUE7}}$/DLAD0	NC
114	PB6/$\overline{\text{WUE6}}$/DLAD1	NC
115	PB5/$\overline{\text{WUE5}}$/DLAD2	NC
116	PB4/$\overline{\text{WUE4}}$/DLAD3	NC
117	PB3/$\overline{\text{WUE3}}$/$\overline{\text{DLFRAME}}$	NC

Pin No.	Pin Name	
	Single-Chip Mode	
TFP-144	Mode 2 and Mode 3 (EXPE = 0)	Flash Memory Programmer Mode
118	PB2/$\overline{\text{WUE2}}$	NC
119	PB1/$\overline{\text{WUE1}}$/LSCI	NC
120	PB0/$\overline{\text{WUE0}}$/$\overline{\text{LSMI}}$	NC
121	P30/LAD0	FO0
122	P31/LAD1	FO1
123	P32/LAD2	FO2
124	P33/LAD3	FO3
125	P34/$\overline{\text{LFRAME}}$	FO4
126	P35/$\overline{\text{LRESET}}$	FO5
127	P36/LCLK	FO6
128	P37/SERIRQ	FO7
129	P80/$\overline{\text{PME}}$	NC
130	P81/GA20	NC
131	P82/$\overline{\text{CLKRUN}}$	NC
132	P83/$\overline{\text{LPCPD}}$	NC
133	P84/$\overline{\text{IRQ3}}$/TxD1/IrTxD	NC
134	P85/$\overline{\text{IRQ4}}$/RxD1/IrRxD	NC
135 (N)	P86/$\overline{\text{IRQ5}}$/SCK1/SCL1	NC
136	P40/TMCI0/TxD2/DSERIRQ	NC
137	P41/TMO0/RxD2/$\overline{\text{DCLKRUN}}$	NC
138 (N)	P42/$\overline{\text{ExIRQ7}}$/TMRI0/SCK2/SDA1	NC
139	VSS	VSS
140	X1	NC
141	X2	NC
142	$\overline{\text{RESO}}$	NC
143	XTAL	XTAL
144	EXTAL	EXTAL

Notes: (N) indicates the pin is driven by NMOS push-pull/open drain.

* Not supported by the system development tool (emulator).

1.3.3 Pin Functions

Table 1.2 Pin Functions

Type	Symbol	Pin No.	I/O	Name and Function
Power supply	VCC	1, 36, 86	Input	Power supply pins. Connect all these pins to the system power supply. Connect the bypass capacitor between VCC and VSS (near VCC).
	VCL	13	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	Input	Ground pins. Connect all these pins to the system power supply (0 V).
Clock	XTAL	143	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 23, Clock Pulse Generator.
	EXTAL	144	Input	
	ϕ	18	Output	Supplies the system clock to external devices.
	EXCL	18	Input	32.768-kHz external clock for sub clock should be supplied. To which pin the external clock is input can be selected from the EXCL and ExEXCL pins.
	ExEXCL	16	Input	
	X2 X1	141 140	Input	These pins should be left open.
Operating mode control	MD2 MD1 MD0	25 9 10	Input	These pins set the operating mode. Inputs at these pins should not be changed during operation.
System control	$\overline{\text{RES}}$	8	Input	Reset pin. When this pin is low, the chip is reset.
	$\overline{\text{RESO}}$	142	Output	Outputs a reset signal to an external device.
	$\overline{\text{STBY}}$	12	Input	When this pin is low, a transition is made to hardware standby mode.
	FWE	26	Input	Control pin for use by flash memory

Type	Symbol	Pin No.	I/O	Name and Function
Interrupts	NMI	11	Input	Nonmaskable interrupt request input pin
	$\overline{\text{IRQ15}}$ to $\overline{\text{IRQ0}}$	17, 19, 20, 21, 47 to 50, 85, 84, 135, 134, 133, 24, 23, 22	Input	These pins request a maskable interrupt. To which pin an IRQ interrupt is input can be selected from the $\overline{\text{IRQn}}$ and $\overline{\text{ExIRQn}}$ pins. (n = 15 to 0)
	$\overline{\text{ExIRQ15}}$ to $\overline{\text{ExIRQ0}}$	51 to 58 138 14 73 to 68		
Boundary scan (JTAG)	$\overline{\text{ETRST}}$*2	27	Input	Interface pins for boundary scan
	ETMS	28	Input	Reset by holding the $\overline{\text{ETRST}}$ pin to low regardless of the JTAG activation. At this time, the $\overline{\text{ETRST}}$ pin should be held low for 20 clocks of ETCK. For details, see section 26, Electrical Characteristics. Then, to activate the JTAG, the $\overline{\text{ETRST}}$ pin should be set to high and the pins ETCK, ETMS, and ETDI should be set appropriately. When in the normal operation without activating the JTAG, pins $\overline{\text{ETRST}}$, ETCK, ETMS, and ETDI are set to high or high-impedance. As these pins are pulled up inside the chip, take care during standby state.
	ETDO	29	Output	
	ETDI	30	Input	
	ETCK	31	Input	

Type	Symbol	Pin No.	I/O	Name and Function
PWM timer (PWM)	PW15 to PW8	96 to 103	Output	PWM timer pulse output pins. From which pin pulses are output can be selected from the PWn and ExPWn pins. (n = 15 to 12)
	ExPW15 to ExPW12	43 to 46		
14-bit PWM timer (PWMX)	PWX1 PWX0	6 5	Output	PWMX pulse output pins
16-bit free running timer (FRT)	FTCI	78	Input	External event input pin
	FTOA FTOB	79 84	Output	Output compare output pins
	FTIA to FTID	80 to 83	Input	Input capture input pins
16-bit timer pulse unit (TPU)	TCLKD TCLKC TCLKB TCLKA	59 61 63 64	Input	Timer external clock input/output pins
	TIOCA0 TIOCB0 TIOCC0 TIOCD0	66 65 64 63	Input/ Output	Input capture input/output compare output/PWM output pins for TGRA_0 to TGRD_0
	TIOCA1 TIOCB1	62 61	Input/ Output	Input capture input/output compare output/PWM output pins for TGRA_1 and TGRB_1
	TIOCA2 TIOCB2	60 59	Input/ Output	Input capture input/output compare output/PWM output pins for TGRA_2 and TGRB_2
8-bit timer (TMR_0, TMR_1, TMR_X, TMR_Y)	TMO0 TMO1 TMOX ExTMOX TMOY	137 3 85 47 15	Output	Waveform output pins with output compare function. From which pin waveforms are output can be selected from the TMOX and ExTMOX pins.
	TMCI0 TMCI1 ExTMCI0 ExTMCI1	136 2 58 57	Input	Input pins for the external clock input to the counter. To which pin the external clock is input can be selected from the TMCIn and ExTMCIn pins. (n = 1 or 0)

Type	Symbol	Pin No.	I/O	Name and Function
8-bit timer (TMR_0, TMR_1, TMR_X, TMR_Y)	TMRI0 TMRI1	138 4	Input	External event input pin and counter reset input pin
	TMIX TMIY ExTMIX ExTMIY	78 80 56 55	Input	External event input pins and counter reset input pins. To which pin an external event or counter reset is input can be selected from the TMIn and ExTMIn pins. (n = X or Y)
Serial communi-cation interface (SCI_1, SCI_2)	TxD1 TxD2	133 136	Output	Transmit data output pins
	RxD1 RxD2	134 137	Input	Receive data input pins
	SCK1 SCK2	135 138	Input/ Output	Clock input/output pins. Output type is NMOS push-pull output.
SCI with IrDA (SCI)	IrTxD	133	Output	Encoded data output pin for IrDA
	IrRxD	134	Input	Encoded data input pin for IrDA
I^2C bus interface (IIC)	SCL0 SCL1 ExSCLA ExSCLB	14 135 53 51	Input/ Output	I^2C clock input/output pins. These pins can drive a bus directly with the NMOS open drain output. To which pin the I^2C clock is input or output can be selected from the SCLn, ExSCLA, and ExSCLB pins. (n = 1 or 0)
	SDA0 SDA1 ExSDAA ExSDAB	17 138 54 52	Input/ Output	I^2C data input/output pins. These pins can drive a bus directly with the NMOS open drain output. To which pin the I^2C data is input or output can be selected from the SDAn, ExSDAA, and ExSDAB pins. (n = 1 or 0)
Keyboard buffer control unit (KBU)	PS2AC PS2BC PS2CC	39 37 34	Input/ Output	Synchronous clock input/output pins for the keyboard buffer control unit
	PS2AD PS2BD PS2CD	38 35 33	Input/ Output	Data input/output pins for the keyboard buffer control unit.

Type	Symbol	Pin No.	I/O	Name and Function
Keyboard control	$\overline{KIN15}$ to $\overline{KIN0}$	33 to 35, 37 to 41, 85 to 78	Input	Matrix keyboard input pins. All pins have a wake-up function. Normally, $\overline{KIN0}$ to $\overline{KIN15}$ function as key scan inputs, and P10 to P17 and P20 to P27 function as key scan outputs. Thus, composed with a maximum of 16 outputs x 16 inputs, a 256-key matrix can be configured.
	$\overline{WUE15}$ to $\overline{WUE8}$	87 to 94	Input	Wake-up event input pins. Same wake up as key wake up can be performed with various sources.
	$\overline{WUE7}$ to $\overline{WUE0}$	113 to 120		
A/D converter	AN7 to AN0	75 to 68	Input	Analog input pins
	$\overline{ADTRG}$	24	Input	External trigger input pin to start A/D conversion
	AVCC	76	Input	Analog power supply pin. When the A/D converter is not used, this pin should be connected to the system power supply (+3.3 V).
	AVref	77	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 (+3.3 V).
	AVSS	67	Input	Ground pin for the A/D converter. This pin should be connected to the system power supply (0 V).

Type	Symbol	Pin No.	I/O	Name and Function
LPC Interface (LPC)	LAD3 to LAD0	124 to 121	Input/ Output	Transfer cycle type, address, and data input/output pins
	$\overline{\text{LFRAME}}$	125	Input	Input pin indicating transfer cycle start and forced termination of an abnormal transfer cycle
	$\overline{\text{LRESET}}$	126	Input	LPC reset pin. When this pin is low, a reset state is entered.
	LCLK	127	Input	LPC clock input pin
	SERIRQ	128	Input/ Output	LPC serial host interrupt (HIRQ1, SMI, HIRQ6, or HIRQ9 to HIRQ12) input/output pin
	LSCI $\overline{\text{LSMI}}$ $\overline{\text{PME}}$	119 120 129	Input/ Output	General input/output ports of LSCI, LSMI, and PME
	GA20	130	Output	GATE A20 control signal output pin
	$\overline{\text{CLKRUN}}$	131	Input/ Output	LCLK operation start request input/output pin
	$\overline{\text{LPCPD}}$	132	Input	LPC module shutdown control input pin
	$\overline{\text{LID3}}$	32	Input	Input pin for setting host address 31
	DLAD3 to DLAD0	116 to 113	Input/ Output	LAD input/output pins for the docking LPC
	$\overline{\text{DLFRAME}}$	117	Output	$\overline{\text{LFRAME}}$ output pin for the docking LPC
	DSERIRQ	136	Input/ Output	SERIRQ input/output pin for the docking LPC
	$\overline{\text{DCLKRUN}}$	137	Input/ Output	$\overline{\text{CLKRUN}}$ input/output pin for the docking LPC
	LDRQ	88	Output	Encoded DMA request output pin for the docking LPC
	DLDRQ	87	Input	Encoded DMA request input pin for the docking LPC

Type	Symbol	Pin No.	I/O	Name and Function
I/O ports	P17 to P10	104 to 110, 112	Input/ Output	Eight input/output pins
	P27 to P20	96 to 103	Input/ Output	Eight input/output pins
	P37 to P30	128 to 121	Input/ Output	Eight input/output pins
	P47 to P40	6 to 2, 138 to 136	Input/ Output	Eight input/output pins
	P52 to P50	14 to 16	Input/ Output	Three input/output pins
	P67 to P60	85 to 78	Input/ Output	Eight input/output pins
	P77 to P70	75 to 68	Input	Eight input pins
	P86 to P80	135 to 129	Input/ Output	Seven input/output pins
	P97 to P90	17 to 24	Input/ Output	Eight input/output pins
	PA7 to PA0	33 to 35, 37 to 41	Input/ Output	Eight input/output pins
	PB7 to PB0	113 to 120	Input/ Output	Eight input/output pins
	PC7 to PC0	87 to 94	Input/ Output	Eight input/output pins
	PD7 to PD0	59 to 66	Input/ Output	Eight input/output pins
	PE4 to PE0*[1]	28 to 32	Input	Five input pins
	PF7 to PF0	43 to 50	Input/ Output	Eight input/output pins
	PG7 to PG0	51 to 58	Input/ Output	Eight 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.
Apart the power on reset circuit from this LSI to prevent the $\overline{\text{ETRST}}$ pin of the board tester from affecting the operation of this LSI.
Apart the power on reset circuit from this LSI to prevent the system reset of this LSI from affecting the ETRST pin of the board tester.

Figure1.3 shows an example of design in which signals for reset do not affect each other.

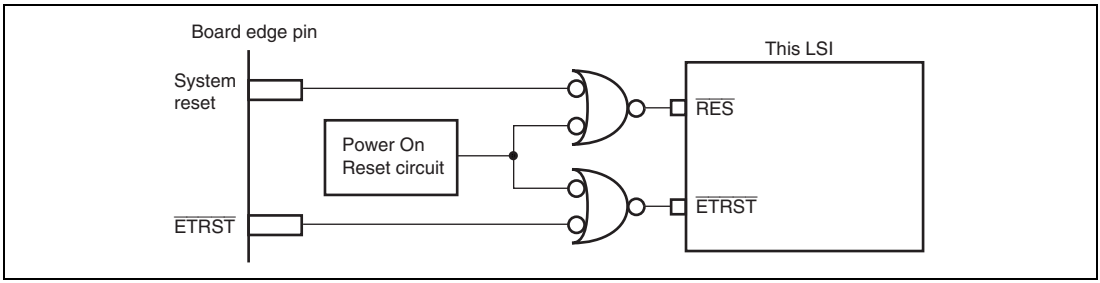


Figure 1.3 Sample Design of Reset Signals with no Affection Each Other

Section 2 CPU

The H8S/2000 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/2000 CPU has sixteen 16-bit general registers, can address a 16 Mbytes linear address space, and is ideal for realtime control.

This section describes the H8S/2000 CPU. The usable modes and address spaces differ depending on the product. For details on each product, see section 3, MCU Operating Modes.

2.1 Features

- Upward-compatibility with H8/300 and H8/300H CPUs
  - Can execute H8/300 CPU and H8/300H CPU object programs
- General-register architecture
  - Sixteen 16-bit general registers also usable as sixteen 8-bit registers or eight 32-bit registers
- Sixty-five basic instructions
  - 8/16/32-bit arithmetic and logic instructions
  - Multiply and divide instructions
  - Powerful bit-manipulation instructions
- Eight addressing modes
  - Register direct [Rn]
  - Register indirect [@ERn]
  - Register indirect with displacement [@(d:16,ERn) or @(d:32,ERn)]
  - Register indirect with post-increment or pre-decrement [@ERn+ or @–ERn]
  - Absolute address [@aa:8, @aa:16, @aa:24, or @aa:32]
  - Immediate [#xx:8, #xx:16, or #xx:32]
  - Program-counter relative [@(d:8,PC) or @(d:16,PC)]
  - Memory indirect [@@aa:8]
- 16 Mbytes address space
  - Program: 16 Mbytes
  - Data: 16 Mbytes
- High-speed operation
  - All frequently-used instructions are executed in one or two states
  - 8/16/32-bit register-register add/subtract: 1 state
  - 8 × 8-bit register-register multiply: 12 states (MULXU.B), 13 states (MULXS.B)
  - 16 ÷ 8-bit register-register divide: 12 states (DIVXU.B)

— 16 × 16-bit register-register multiply: 20 states (MULXU.W), 21 states (MULXS.W)
— 32 ÷ 16-bit register-register divide: 20 states (DIVXU.W)

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

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 as shown below.

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

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

In addition, there are differences in address space, CCR and EXR register functions, 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/2000 CPU has the following enhancements.

- More general registers and control registers
  - Eight 16-bit extended registers and one 8-bit control register have been added.
- Extended address space
  - Normal mode supports the same 64 kbytes address space as the H8/300 CPU.
  - Advanced mode supports a maximum 16 Mbytes address space.
- Enhanced addressing
  - The addressing modes have been enhanced to make effective use of the 16 Mbytes address space.
- Enhanced instructions
  - Addressing modes of bit-manipulation instructions have been enhanced.
  - Signed multiply and divide instructions have been added.
  - Two-bit shift and two-bit rotate instructions have been added.
  - Instructions for saving and restoring multiple registers have been added.
  - A test and set instruction has been added.
- Higher speed
  - Basic instructions are executed twice as fast.

2.1.3 Differences from H8/300H CPU

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

- Additional control register
  - One 8-bit control register has been added.
- Enhanced instructions
  - Addressing modes of bit-manipulation instructions have been enhanced.
  - Two-bit shift and two-bit rotate instructions have been added.
  - Instructions for saving and restoring multiple registers have been added.
  - A test and set instruction has been added.
- Higher speed
  - Basic instructions are executed twice as fast.

2.2 CPU Operating Modes

The H8S/2000 CPU has two operating modes: normal and advanced. Normal mode supports a maximum 64 kbytes address space. Advanced mode supports a maximum 16 Mbytes address space. The mode is selected by the LSI's mode pins.

2.2.1 Normal Mode

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

- Address space

  Linear access to a maximum address space of 64 kbytes is possible.
- Extended registers (En)

  The extended registers (E0 to E7) can be used as 16-bit registers, or as the upper 16-bit segments of 32-bit registers.

  When extended register 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 general register Rn is referenced in the register indirect addressing mode with pre-decrement (@–Rn) or post-increment (@Rn+) and a carry or borrow occurs, 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 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 top area from H'0000 to H'00FF. Note that this area is also used for the exception vector table.
- Stack structure

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

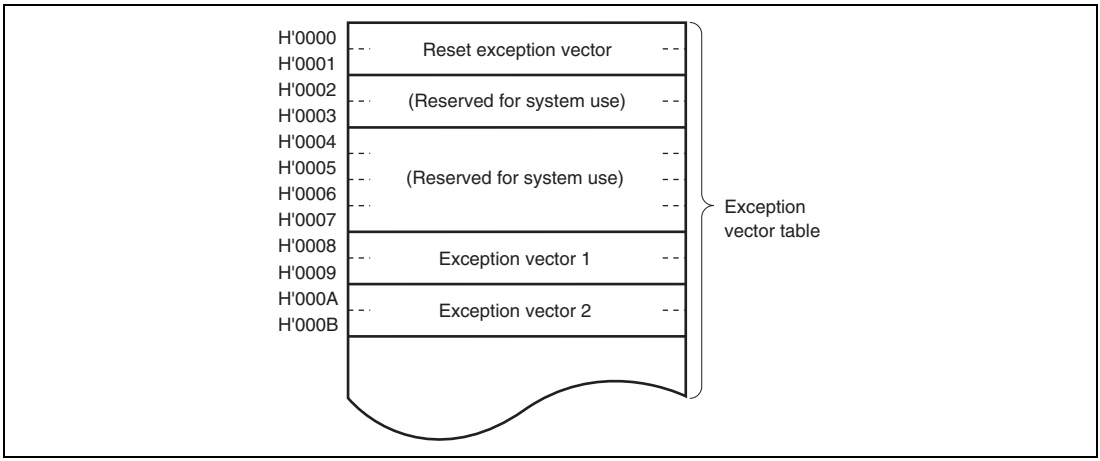


Figure 2.1 Exception Vector Table (Normal Mode)

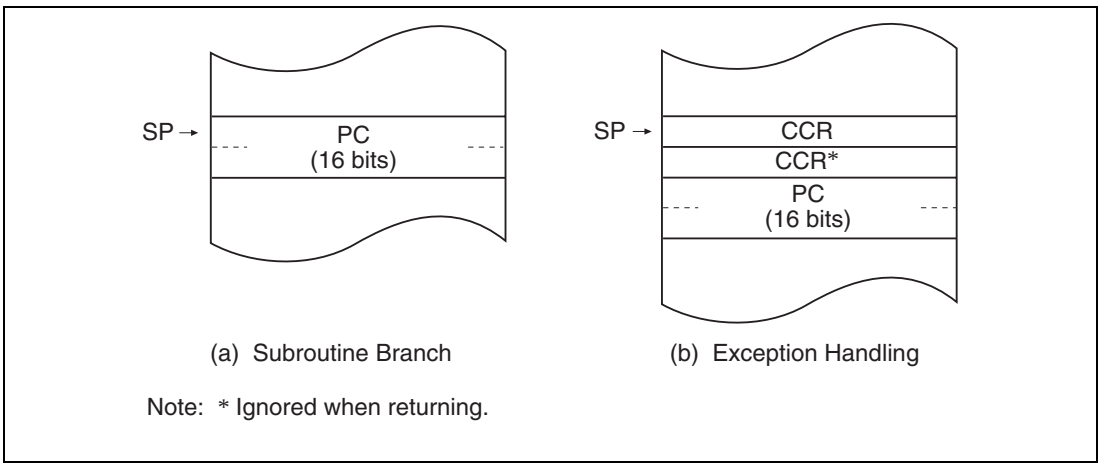


Figure 2.2 Stack Structure in Normal Mode

2.2.2 Advanced Mode

- Address space

  Linear access to a maximum address space of 16 Mbytes is possible.
- Extended registers (En)

  The extended registers (E0 to E7) can be used as 16-bit registers. They can also be used 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 32-bit units. In each 32 bits, the upper eight bits are ignored and a branch address is stored in the lower 24 bits (see figure 2.3). For details of the exception vector table, see section 4, Exception Handling.

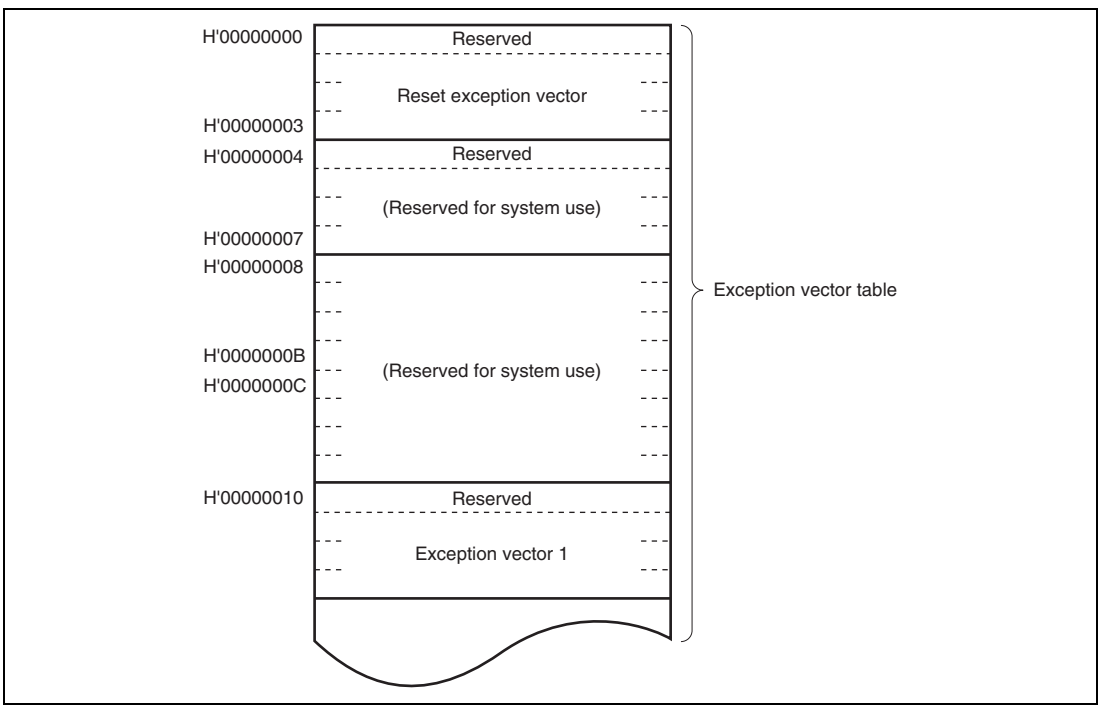


Figure 2.3 Exception Vector Table (Advanced Mode)

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 eight bits of these 32 bits are 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 top area 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 and condition-code register (CCR) are pushed onto the stack in exception handling, they are stored as shown in figure 2.4. The extended control register (EXR) is not pushed onto the stack. For details, see section 4, Exception Handling.

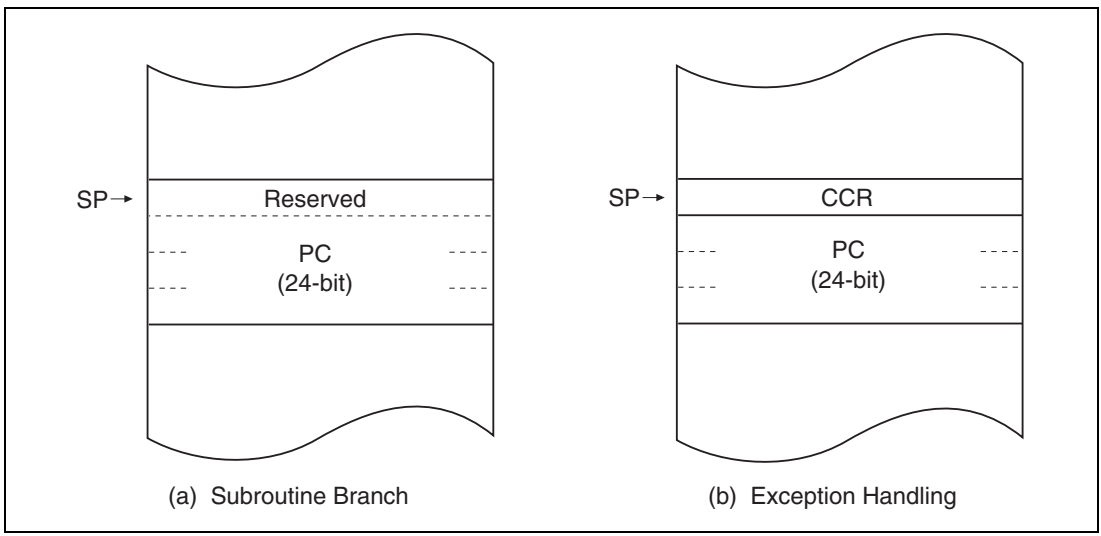


Figure 2.4 Stack Structure in Advanced Mode

2.3 Address Space

Figure 2.5 shows a memory map of the H8S/2000 CPU. The H8S/2000 CPU provides linear access to a maximum 64 kbytes address space in normal mode, and a maximum 16 Mbytes (architecturally 4 Gbytes) address space in advanced mode. The usable modes and address spaces differ depending on the product. For details on each product, see section 3, MCU Operating Modes.

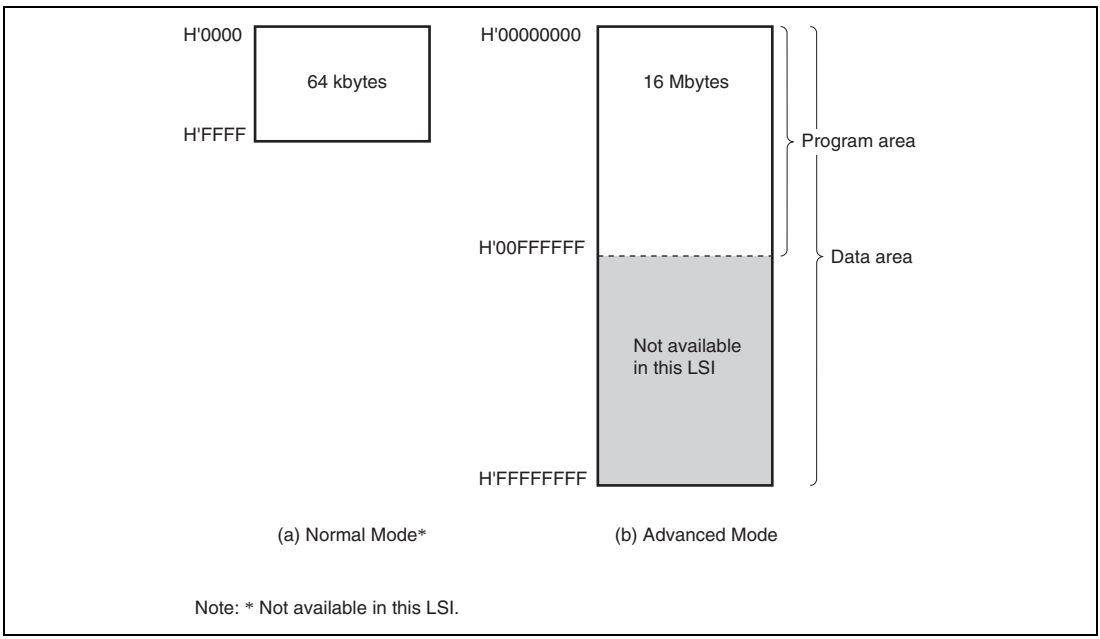


Figure 2.5 Memory Map

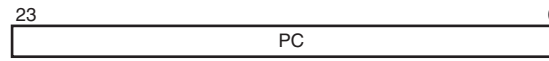
2.4 Register Configuration

The H8S/2000 CPU has the internal registers shown in figure 2.6. These are classified into two types of registers: general registers and control registers. Control registers refer to a 24-bit program counter (PC), an 8-bit extended control register (EXR), and an 8-bit condition code register (CCR).

General Registers (Rn) and Extended Registers (En)

	15 0	7 0	7 0
ER0	E0	R0H	R0L
ER1	E1	R1H	R1L
ER2	E2	R2H	R2L
ER3	E3	R3H	R3L
ER4	E4	R4H	R4L
ER5	E5	R5H	R5L
ER6	E6	R6H	R6L
ER7 (SP)	E7	R7H	R7L

Control Registers



	7	6	5	4	3	2	1	0
EXR*	T	-	-	-	-	I2	I1	I0

	7	6	5	4	3	2	1	0
CCR	I	UI	H	U	N	Z	V	C

Legend

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

Note: * Does not affect operation in this LSI.

Figure 2.6 CPU Internal Registers

2.4.1 General Registers

The H8S/2000 CPU has eight 32-bit general registers. These general registers are all functionally alike 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).

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

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

The usage of each register can be selected independently.

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

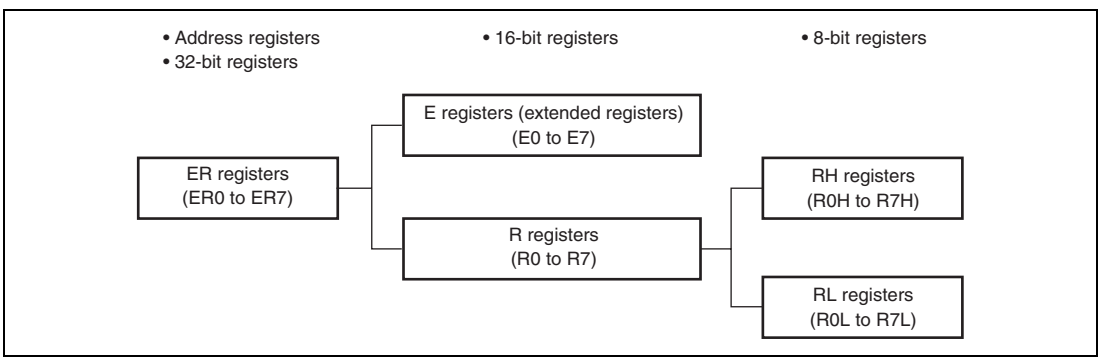


Figure 2.7 Usage of General Registers

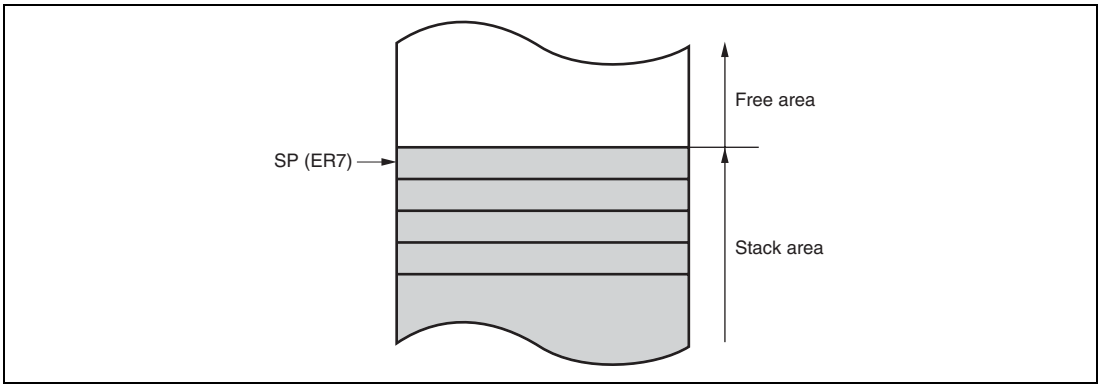


Figure 2.8 Stack

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 for read, the least significant PC bit is regarded as 0.)

2.4.3 Extended Control Register (EXR)

EXR does not affect operation in this LSI.

Bit	Bit Name	Initial Value	R/W	Description
7	T	0	R/W	Trace Bit Does not affect operation in this LSI.
6 to 3	–	All 1	R	Reserved These bits are always read as 1.
2 to 0	I2	1	R/W	Interrupt Mask Bits 2 to 0
	I1	1	R/W	Do not affect operation in this LSI.
	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, see section 5, Interrupt Controller.
6	UI	Undefined	R/W	User Bit or Interrupt Mask Bit Can be written to and read from by software using the LDC, STC, ANDC, ORC, and XORC instructions.
5	H	Undefined	R/W	Half-Carry Flag When the ADD.B, ADDX.B, SUB.B, SUBX.B, 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 written to and read from 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 when data is zero, and cleared to 0 when data is not zero.

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 otherwise.
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 Initial Register Values

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

2.5 Data Formats

The H8S/2000 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 of general registers.

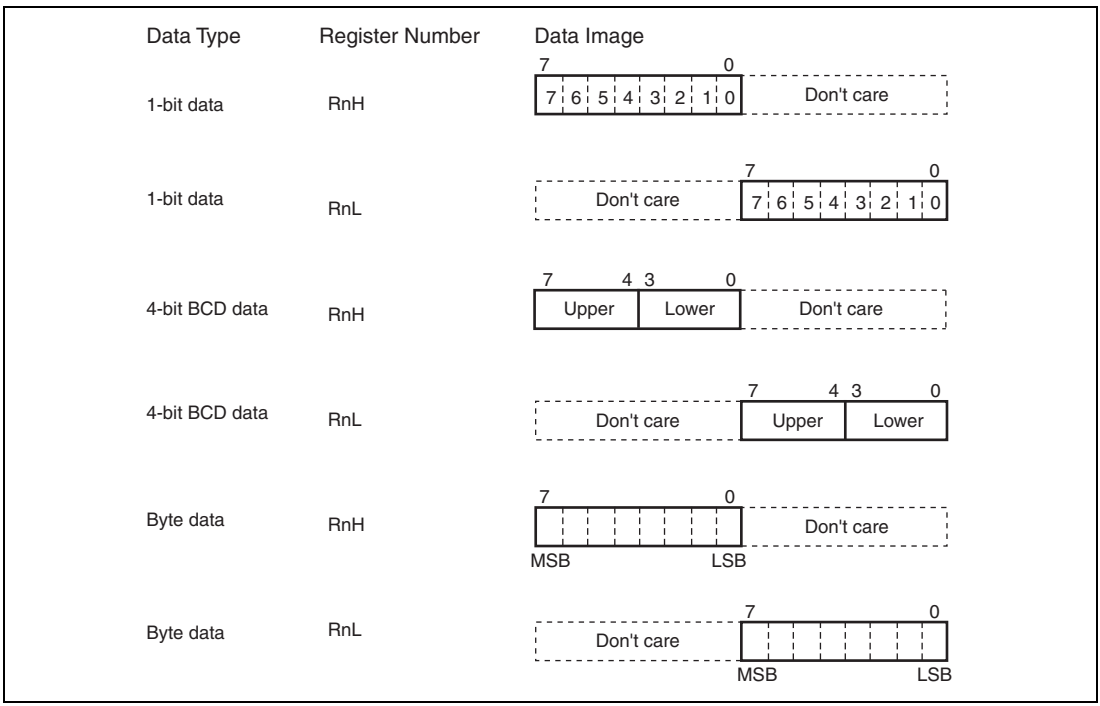


Figure 2.9 General Register Data Formats (1)

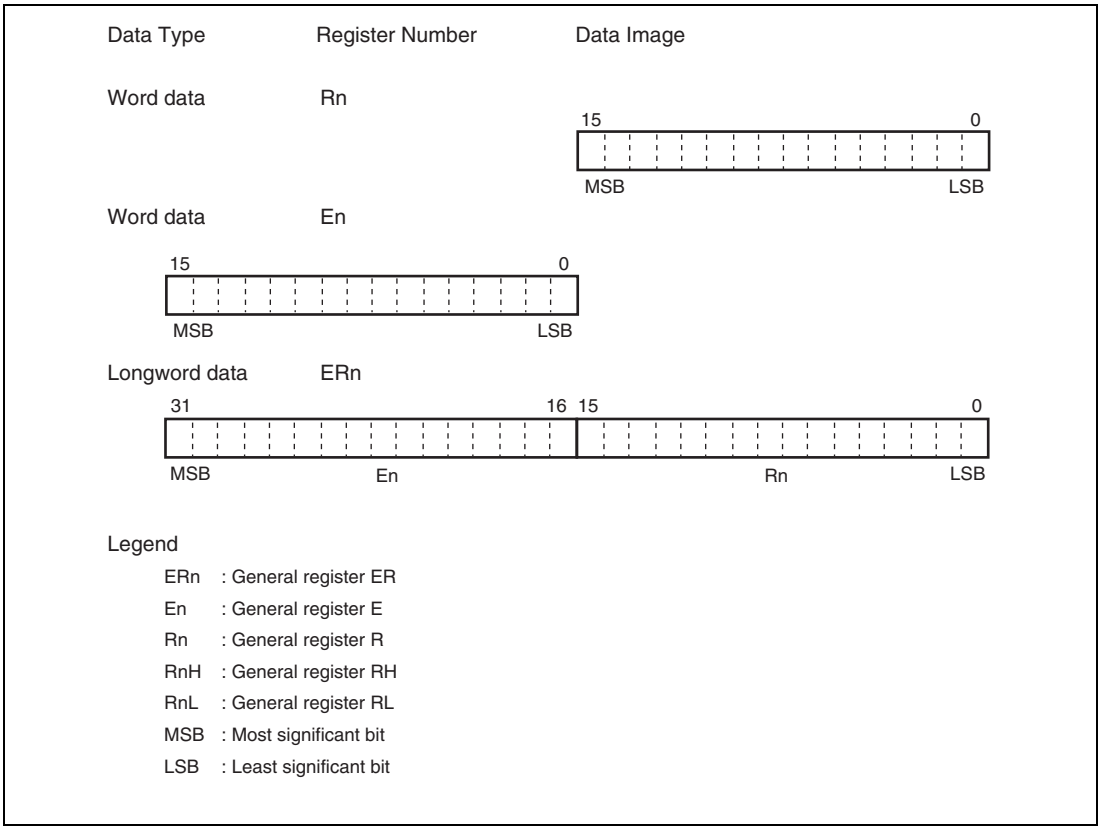


Figure 2.9 General Register Data Formats (2)

2.5.2 Memory Data Formats

Figure 2.10 shows the data formats in memory. The H8S/2000 CPU can access word data and longword data in memory, but 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, no address error occurs but the least significant bit of the address is regarded as 0, so the access starts at the preceding address. This also applies to instruction fetches.

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

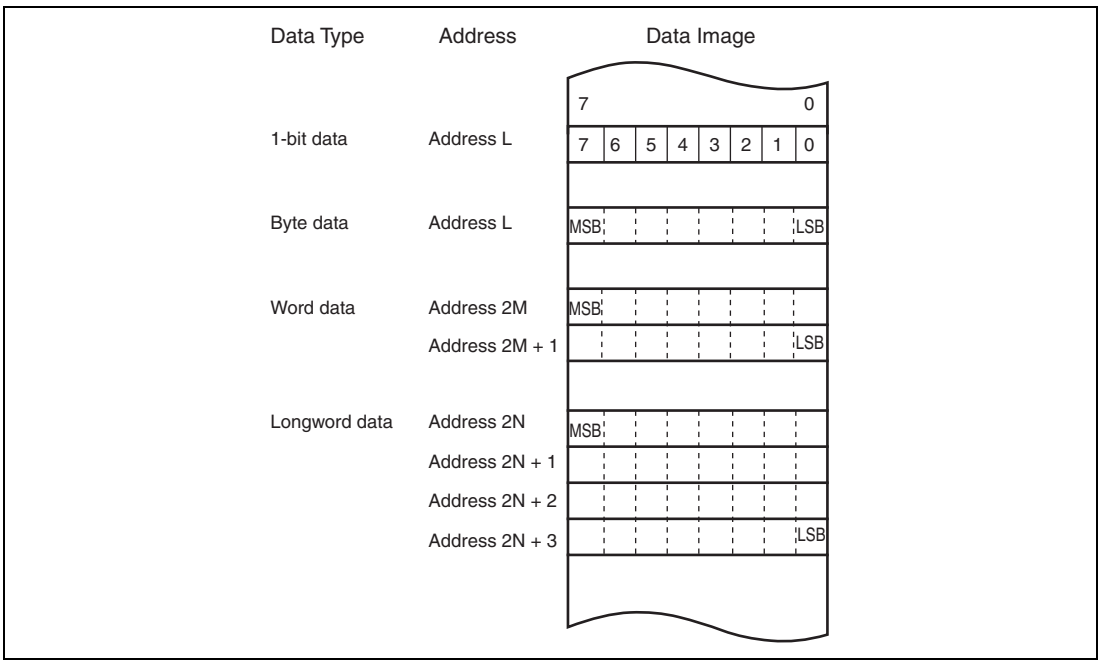


Figure 2.10 Memory Data Formats

2.6 Instruction Set

The H8S/2000 CPU has 65 types of instructions. The instructions are classified by function as shown 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*[5], STM*[5]	L	
	MOVFPE*[3], MOVTPE*[3]	B	
Arithmetic operations	ADD, SUB, CMP, NEG	B/W/L	19
	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	
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	B_{CC}*[2], JMP, BSR, JSR, RTS	–	5
System control	TRAPA, RTE, SLEEP, LDC, STC, ANDC, ORC, XORC, NOP	–	9
Block data transfer	EEPMOV	–	1
			Total: 65

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

1. POP.W Rn and PUSH.W Rn are identical to MOV.W @SP+, Rn and MOV.W Rn, @-SP. POP.L ERn and PUSH.L ERn are identical to MOV.L @SP+, ERn and MOV.L ERn, @-SP.
2. B_{CC} is the generic name for conditional branch instructions.
3. Cannot be used in this LSI.
4. To use the TAS instruction, use registers ER0, ER1, ER4, and ER5.
5. Since register ER7 functions as the stack pointer in an STM/LDM instruction, it cannot be used as an STM/LDM register.

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)
(EAd)	Destination operand
(EAs)	Source operand
EXR	Extended control register
CCR	Condition-code register
N	N (negative) flag in CCR
Z	Z (zero) flag in CCR
V	V (overflow) flag in CCR
C	C (carry) flag in CCR
PC	Program counter
SP	Stack pointer
#IMM	Immediate data
disp	Displacement
+	Addition
–	Subtraction
×	Multiplication
÷	Division
∧	Logical AND
∨	Logical OR
⊕	Logical exclusive OR
→	Move
~	NOT (logical complement)
:8/:16/:24/:32	8-, 16-, 24-, or 32-bit length

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

Table 2.3 Data Transfer Instructions

Instruction	Size*[1]	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*[2]	L	@SP+ → Rn (register list) Pops two or more general registers from the stack.
STM*[2]	L	Rn (register list) → @-SP Pushes two or more general registers onto the stack.

Notes: 1. Size refers to the operand size.
B: Byte
W: Word
L: Longword
2. Since register ER7 functions as the stack pointer in an STM/LDM instruction, it cannot be used as an STM/LDM register.

Table 2.4 Arithmetic Operations Instructions (1)

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

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

Table 2.4 Arithmetic Operations Instructions (2)

Instruction	Size*	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 the 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.

Notes: 1. Size refers to the operand size.
B: Byte
W: Word
L: Longword
2. To use the TAS instruction, use registers ER0, ER1, ER4, and ER5.

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 data in a general register.

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

Table 2.6 Shift Instructions

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

Note: * Size 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 Logically 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 Logically 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 Logically 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 Logically 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: * Size refers to the operand size.
B: Byte

Table 2.7 Bit Manipulation Instructions (2)

Instruction	Size*	Function
BXOR	B	C ⊕ (<bit-No.> of <EAd>) → C Logically exclusive-ORs 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 Logically exclusive-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.
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: * Size 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 the memory operand 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 eight bits are valid.
STC	B/W	CCR → (EAd), EXR → (EAd) Transfers CCR or EXR contents to a general register or memory operand. Although CCR and EXR are 8-bit registers, word-size transfers are performed between them and memory. The upper eight bits are valid.
ANDC	B	CCR ∧ #IMM → CCR, EXR ∧ #IMM → EXR Logically ANDs the CCR or EXR contents with immediate data.
ORC	B	CCR ∨ #IMM → CCR, EXR ∨ #IMM → EXR Logically ORs the CCR or EXR contents with immediate data.
XORC	B	CCR ⊕ #IMM → CCR, EXR ⊕ #IMM → EXR Logically exclusive-ORs the CCR or EXR contents with immediate data.
NOP	–	PC + 2 → PC Only increments the program counter.

Note: * Size 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/2000 CPU instructions consist of 2-byte (1-word) units. An instruction consists of an operation field (op), a register field (r), an effective address extension (EA), 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-bit, and data registers by 3-bit or 4-bit. Some instructions have two register fields, and some have no register field.
- Effective address extension

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

  Specifies the branching condition of Bcc instructions.

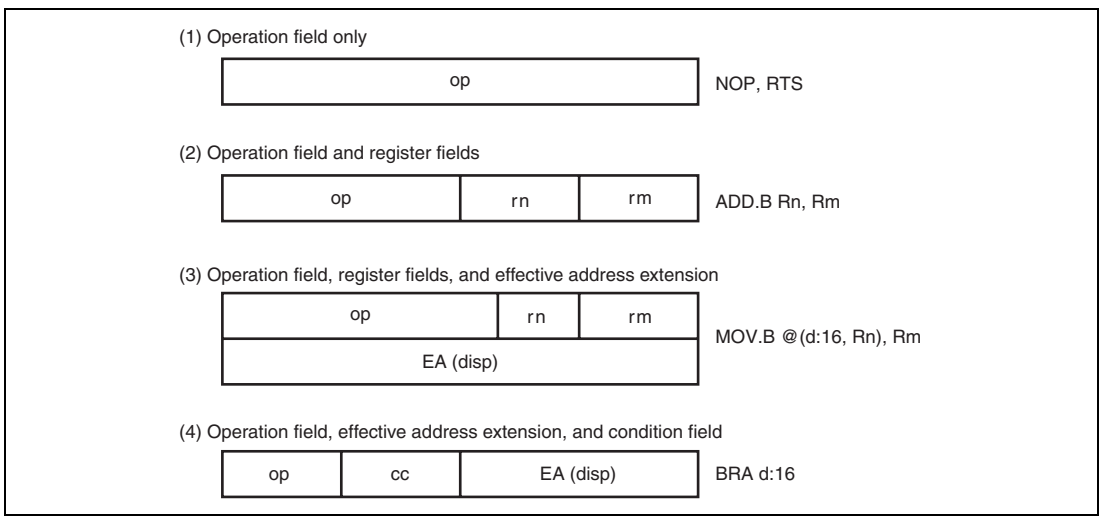


Figure 2.11 Instruction Formats (Examples)

2.7 Addressing Modes and Effective Address Calculation

The H8S/2000 CPU supports the eight addressing modes listed in table 2.11. Each instruction uses a subset of these addressing modes.

Arithmetic and logic operations instructions can use the register direct and immediate addressing modes. Data transfer instructions can use all addressing modes except program-counter relative and memory indirect. Bit manipulation instructions can use register direct, register indirect, or 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	@ERn+
	Register indirect with pre-decrement	@–ERn
5	Absolute address	@aa:8/@aa:16/@aa:24/@aa:32
6	Immediate	#xx:8/#xx:16/#xx:32
7	Program-counter relative	@(d:8,PC)/@(d:16,PC)
8	Memory indirect	@@aa:8

2.7.1 Register Direct—Rn

The register field of the instruction code specifies an 8-, 16-, or 32-bit general register which contains 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 a memory operand. If the address is a program instruction address, the lower 24 bits are valid and the upper eight 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 code 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 access, and 4 for longword access. For word or longword transfer instructions, the register value should be even.

Register Indirect with Pre-Decrement—@-ERn: The value 1, 2, or 4 is subtracted from an address register (ERn) specified by the register field in the instruction code, and the result becomes 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 access, and 4 for longword access. For 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. For a 32-bit absolute address, the entire address space is accessed.

A 24-bit absolute address (@aa:24) indicates the address of a program instruction. The upper eight 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)		

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

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

The ADDS, SUBS, INC, and DEC instructions implicitly contain immediate data in their instruction codes. 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 can be used by the Bcc and BSR instructions. An 8-bit or 16-bit displacement contained in the instruction code is sign-extended to 24-bit and added to the 24-bit address indicated by the PC value to generate a 24-bit branch address. Only the lower 24-bit of this branch address are valid; the upper eight 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-byte (–63 to +64 words) or –32766 to +32768-byte (–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 which contains a branch address. The upper bits of the 8-bit 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 top area of the address range in which the branch address is stored is also used for the exception vector area. For further details, see 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 the instruction code to be fetched at the address preceding the specified address. (For further information, see section 2.5.2, Memory Data Formats.)

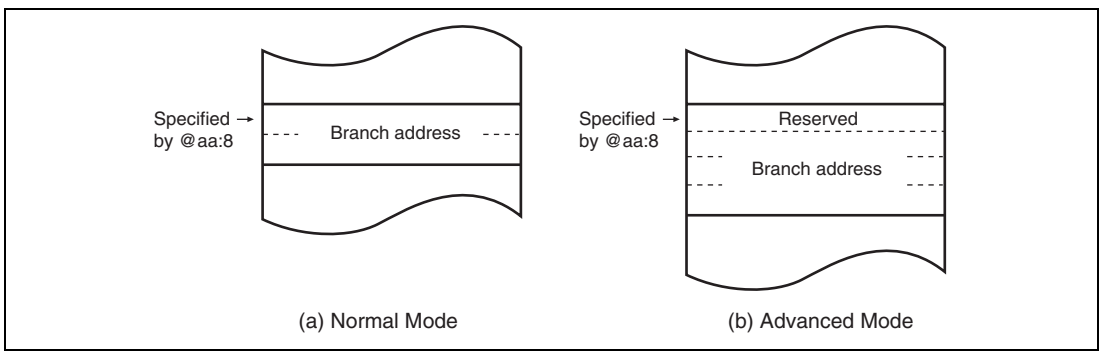


Figure 2.12 Branch Address Specification in Memory Indirect Addressing Mode

2.7.9 Effective Address Calculation

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

Table 2.13 Effective Address Calculation (1)

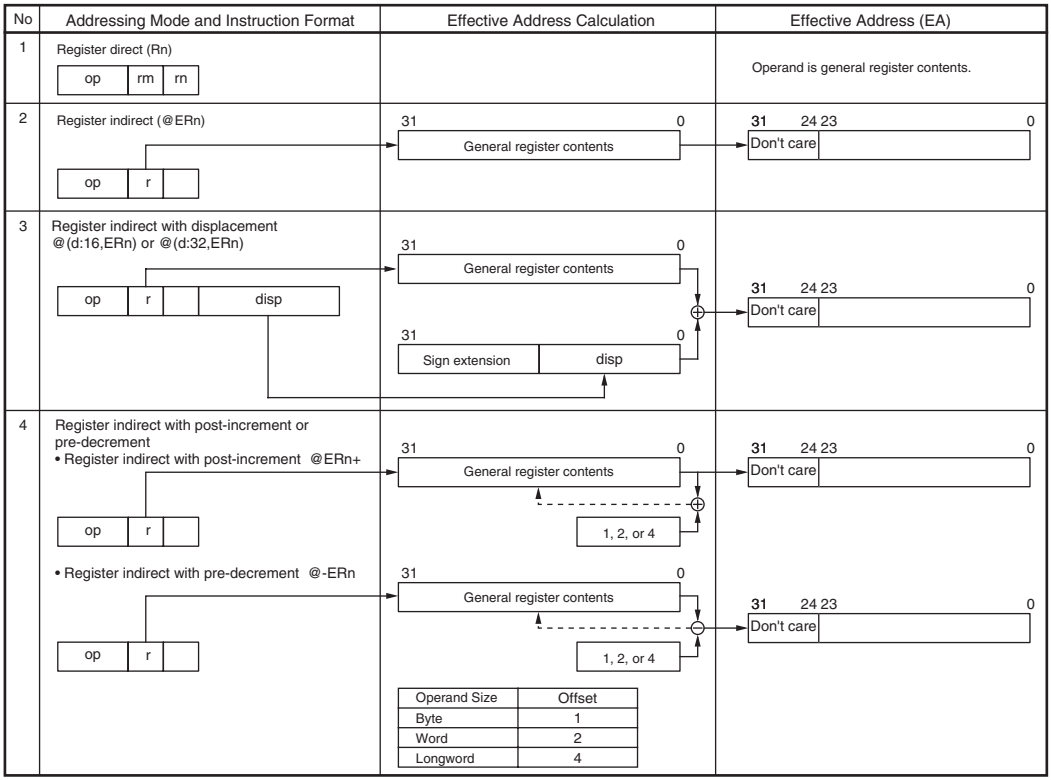
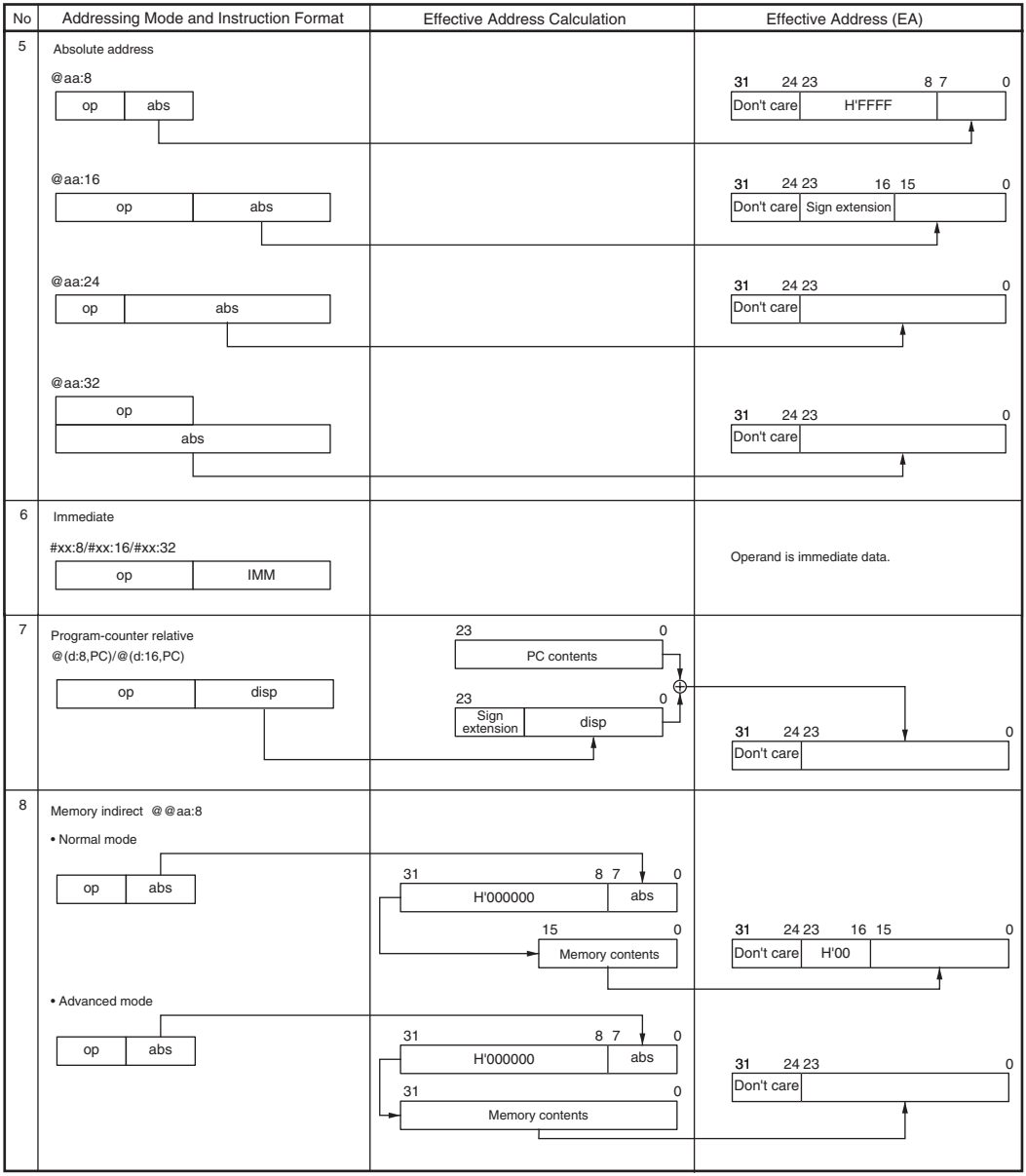


Table 2.13 Effective Address Calculation (2)



2.8 Processing States

The H8S/2000 CPU has five main processing states: the reset state, exception handling state, program execution state, bus-released state, and program stop state. Figure 2.13 indicates the state transitions.

- Reset state

  In this state the CPU and on-chip peripheral modules are all initialized and stopped. 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, see 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, see section 4, Exception Handling.

- Program execution state

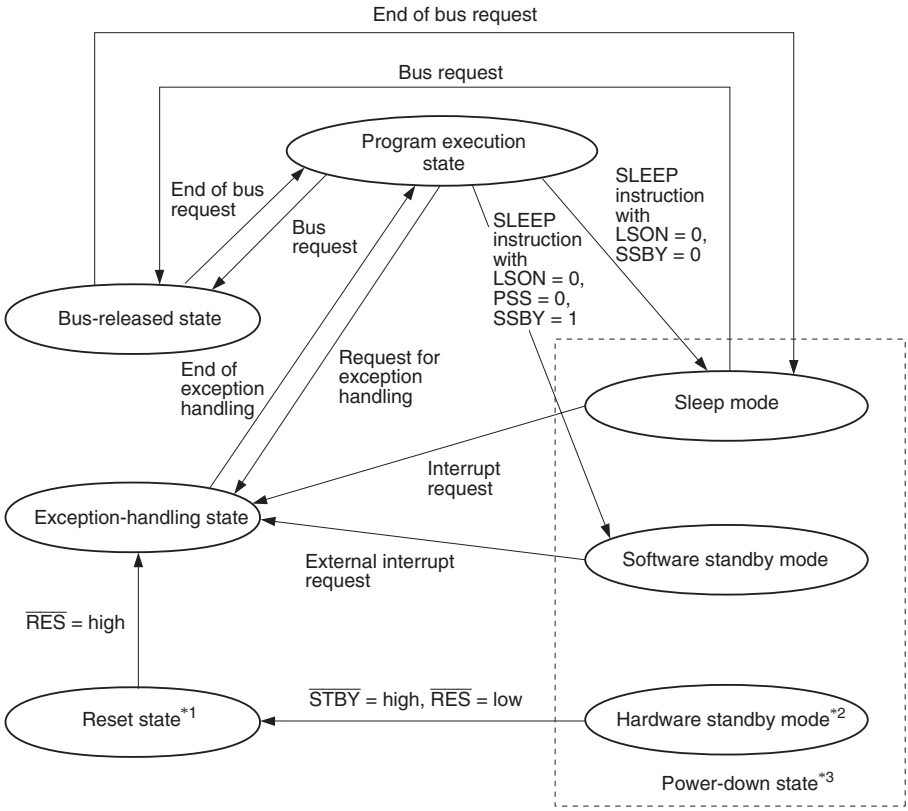
  In this state the CPU executes program instructions in sequence.

- Bus-released state

  In a product which has a bus master other than the CPU, the bus-released state occurs when the bus has been released in response to a bus request from a bus master other than the CPU. While the bus is released, the CPU halts operations. For details, see section 6, Bus Controller (BSC).

- 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 hardware standby mode. For details, see section 24, Power-Down Modes.



Notes: 1. From any state except hardware standby mode, a transition to the reset state occurs whenever $\overline{RES}$ goes low. A transition can also be made to the reset state when the watchdog timer overflows.
2. From any state, a transition to hardware standby mode occurs when $\overline{STBY}$ goes low.
3. The power-down state also includes watch mode, subactive mode, subsleep mode, etc. For details, refer to section 24, Power-Down Modes.

Figure 2.13 State Transitions

2.9 Usage Notes

2.9.1 Note on TAS Instruction Usage

To use the TAS instruction, use registers ER0, ER1, ER4, and ER5.

The TAS instruction is not generated by the Renesas Technology H8S and H8/300 series C/C++ compilers. When the TAS instruction is used as a user-defined intrinsic function, registers ER0, ER1, ER4, and ER5 should be used.

2.9.2 Note on STM/LDM Instruction Usage

Since the ER7 register is used as the stack pointer in an STM/LDM instruction, it cannot be used as a register that allows save (STM) or restore (LDM) operation. Two to four registers can be saved/restored by single STM/LDM instruction. Available registers are listed below.

Two: ER0 and ER1, ER2 and ER3, ER4 and ER5
Three: ER0 to ER2, ER4 to ER6
Four: ER0 to ER3

The STM/LDM instruction with ER7 is not created by the Renesas Technology H8S or H8/300 series C/C++ compilers.

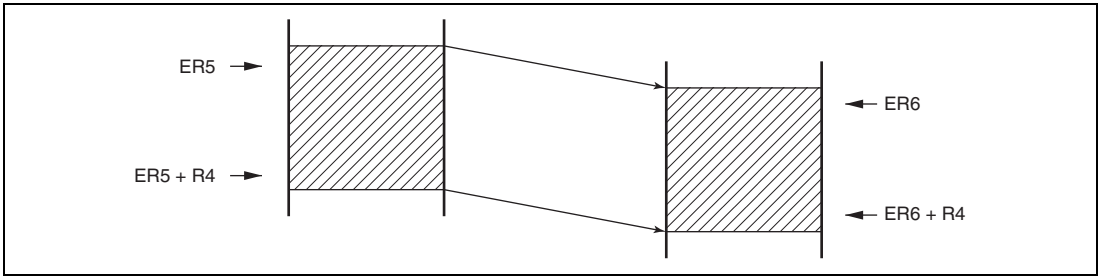
2.9.3 Note on Bit Manipulation Instructions

The BSET, BCLR, BNOT, BST, and BIST instructions read data in byte units, manipulate the data of the target bit, and write data in byte units. Special care is required when using these instructions in cases where a register containing a write-only bit is used or a bit is directly manipulated for a port.

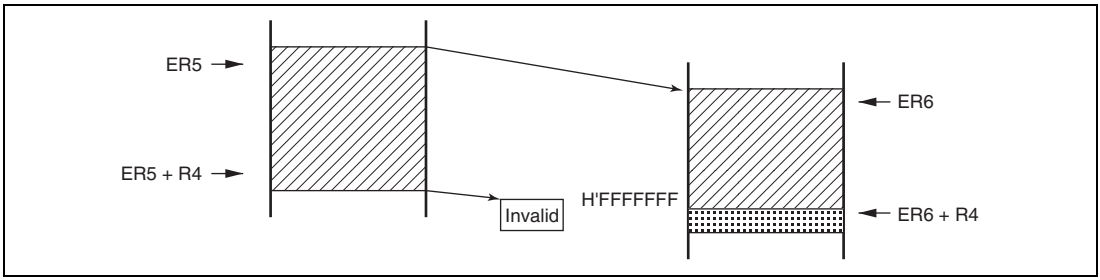
In addition, the BCLR instruction can be used to clear the flag of the internal I/O register. In this case, if the flag to be cleared has been set to 1 by an interrupt processing routine, the flag need not be read before executing the BCLR instruction.

2.9.4 EEPMOV Instruction

1. EEPMOV is a block-transfer instruction and transfers the byte size of data indicated by R4*, which starts from the address indicated by ER5, to the address indicated by ER6.



2. Set R4 and ER6 so that the end address of the destination address (value of ER6 + R4) does not exceed H'00FFFFFF (the value of ER6 must not change from H'00FFFFFF to H'01000000 during execution).



Section 3 MCU Operating Modes

3.1 Operating Mode Selection

This LSI supports five operating modes (modes 2 to 4, 6, and 7). The operating mode is determined by the setting of the mode pins (MD2, MD1, and MD0). 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
3	0	1	1	Normal	Single-chip mode	Enabled
4	1	0	0	—	Flash memory programming/erasing	—
6	1	1	0	Emulation	On-chip emulation mode	Enabled
7	1	1	1	Emulation	On-chip emulation mode	Enabled

Modes 2 and 3 are single-chip mode.

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

Mode 4 is a boot mode for programming or erasing the flash memory. For details, see section 21, Flash Memory (0.18-µm F-ZTAT Version).

Modes 6 and 7 are on-chip emulation modes. In these modes, 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. For details on the bus control register (BCR), see section 6.2.1, Bus Control Register (BCR).

- Mode control register (MDCR)
- System control register (SYSCR)
- Serial timer control register (STCR)
- System control register 3 (SYSCR3)

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 1 0	MDS2 MDS1 MDS0	—* —* —*	R R R	Mode Select 2 to 0 These bits indicate the input levels at mode pins (MD2, MD1, and MD0) (the current operating mode). The MDS2, MDS1, and MDS0 bits correspond to the MD2, MD1, and MD0 pins, respectively. These bits are read-only bits and cannot be written to. The input levels of the mode pins (MD2, MD1, and MD0) are latched into these bits when MDCR is read. These latches are canceled by a reset.

Note: * The initial values are determined by the settings of the MD2, MD1, and MD0 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, 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/TISR, TCORA_X, TCORB_X, TCONRI, and CONRS) 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 25, 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	IICS	0	R/W	I^2C Extra Buffer Select Sets bits 7 to 4 of port A to form an output buffer similar to SCL and SDA. This function is used to realize the I^2C interface only by software. 0: PA7 to PA4 are normal I/O pins 1: PA7 to PA4 are I/O pins that can be bus driven
6	IICX1	0	R/W	I^2C Transfer Rate Select 1, 0 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 16.3.
5	IICX0	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
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 DDCSWR), 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. DDCSWR 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 25, 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 (BCR2, WSCR, PCSR, and SYSCR2). 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. 1: 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.
2	—	0	R/(W)	Reserved The initial value should not be changed.
1	ICKS1	0	R/W	Internal Clock Source Select 1, 0 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).
0	ICKS0	0	R/W	

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*	0	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	0	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 25, 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.

3.3.2 Mode 3

The CPU can access a 64-kbyte address space in either normal mode or single-chip mode. The on-chip ROM is enabled. The size of ROM and RAM that can be used in mode 3 is 56 kbytes and 4 kbytes, respectively.

3.4 Address Map

Figures 3.1 shows the address map in each operating mode.

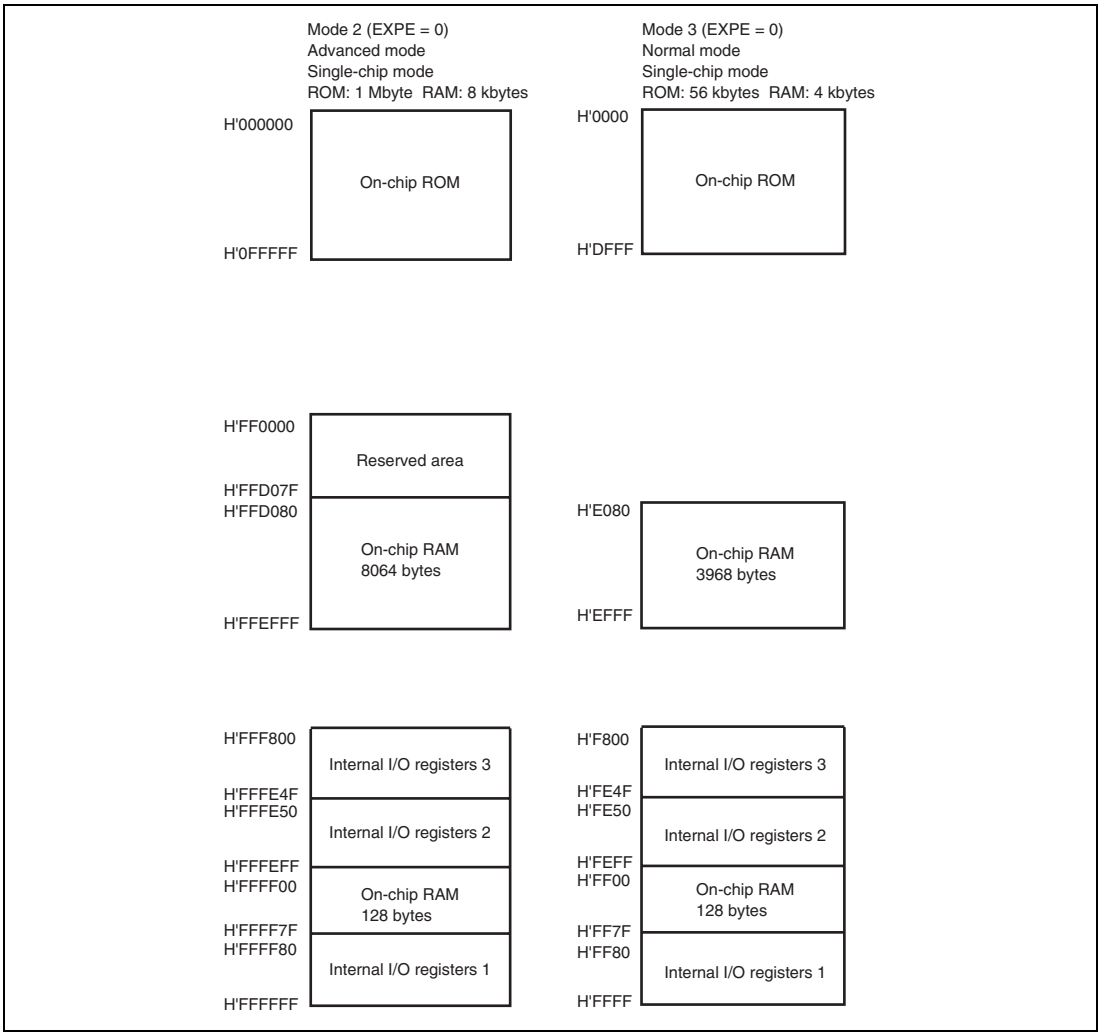


Figure 3.1 Address Map

Section 4 Exception Handling

4.1 Exception Handling Types and Priority

As table 4.1 indicates, exception handling may be caused by a reset, 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 ↑ Low	Reset	Starts immediately after a low-to-high transition of the $\overline{\text{RES}}$ pin, or when the watchdog timer overflows.
	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.

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	
			Normal Mode	Advanced Mode
Reset		0	H'0000 to H'0001	H'000000 to H'000003
Reserved for system use		1 ⋮ 5	H'0002 to H'0003 ⋮ H'000A to H'000B	H'000004 to H'000007 ⋮ H'000014 to H'000017
Direct transition		6	H'000C to H'000D	H'000018 to H'00001B
External interrupt (NMI)		7	H'000E to H'000F	H'00001C to H'00001F
Trap instruction (four sources)		8	H'0010 to H'0011	H'000020 to H'000023
		9	H'0012 to H'0013	H'000024 to H'000027
		10	H'0014 to H'0015	H'000028 to H'00002B
		11	H'0016 to H'0017	H'00002C to H'00002F
Reserved for system use		12 ⋮ 15	H'0018 to H'0019 ⋮ H'001E to H'001F	H'000030 to H'000033 ⋮ H'00003C to H'00003F
External interrupt	IRQ0	16	H'0020 to H'0021	H'000040 to H'000043
	IRQ1	17	H'0022 to H'0023	H'000044 to H'000047
	IRQ2	18	H'0024 to H'0025	H'000048 to H'00004B
	IRQ3	19	H'0026 to H'0027	H'00004C to H'00004F
	IRQ4	20	H'0028 to H'0029	H'000050 to H'000053
	IRQ5	21	H'002A to H'002B	H'000054 to H'000057
	IRQ6, KIN7 to KIN0	22	H'002C to H'002D	H'000058 to H'00005B
	IRQ7, KIN15 to KIN8, WUE7 to WUE0	23	H'002E to H'002F	H'00005C to H'00005F
Internal interrupt*		24 ⋮ 29	H'0030 to H'0031 ⋮ H'003A to H'003B	H'000060 to H'000063 ⋮ H'000074 to H'000077

Exception Source		Vector Number	Vector Addresses	
			Normal Mode	Advanced Mode
Reserved for system use		30	H'003C to H'003D	H'000078 to H'00007B
Reserved for system use		31	H'003E to H'003F	H'00007C to H'00007F
Reserved for system use		32	H'0040 to H'0041	H'000080 to H'000083
External interrupt	WUE15 to WUE8	33	H'0042 to H'0043	H'000084 to H'000087
Internal interrupt*		34	H'0044 to H'0045	H'000088 to H'00008B
		⋮	⋮	⋮
		55	H'006E to H'006F	H'0000DC to H'0000DF
External interrupt	IRQ8	56	H'0070 to H'0071	H'0000E0 to H'0000E3
	IRQ9	57	H'0072 to H'0073	H'0000E4 to H'0000E7
	IRQ10	58	H'0074 to H'0075	H'0000E8 to H'0000EB
	IRQ11	59	H'0076 to H'0077	H'0000EC to H'0000EF
	IRQ12	60	H'0078 to H'0079	H'0000F0 to H'0000F3
	IRQ13	61	H'007A to H'007B	H'0000F4 to H'0000F7
	IRQ14	62	H'007C to H'007D	H'0000F8 to H'0000FB
	IRQ15	63	H'007E to H'007F	H'0000FC to H'0000FF
Internal interrupt*		64	H'0080 to H'0081	H'000100 to H'000103
		⋮	⋮	⋮
		127	H'00FE to H'00FF	H'0001FC to H'0001FF

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

Table 4.3 Exception Handling Vector Table (Extended Vector Mode)

Exception Source		Vector Number	Vector Addresses	
			Normal Mode	Advanced Mode
Reset		0	H'0000 to H'0001	H'000000 to H'000003
Reserved for system use		1 ⋮ 5	H'0002 to H'0003 ⋮ H'000A to H'000B	H'000004 to H'000007 ⋮ H'000014 to H'000017
Direct transition		6	H'000C to H'000D	H'000018 to H'00001B
External interrupt (NMI)		7	H'000E to H'000F	H'00001C to H'00001F
Trap instruction (four sources)		8	H'0010 to H'0011	H'000020 to H'000023
		9	H'0012 to H'0013	H'000024 to H'000027
		10	H'0014 to H'0015	H'000028 to H'00002B
		11	H'0016 to H'0017	H'00002C to H'00002F
Reserved for system use		12 ⋮ 15	H'0018 to H'0019 ⋮ H'001E to H'001F	H'000030 to H'000033 ⋮ H'00003C to H'00003F
External interrupt	IRQ0	16	H'0020 to H'0021	H'000040 to H'000043
	IRQ1	17	H'0022 to H'0023	H'000044 to H'000047
	IRQ2	18	H'0024 to H'0025	H'000048 to H'00004B
	IRQ3	19	H'0026 to H'0027	H'00004C to H'00004F
	IRQ4	20	H'0028 to H'0029	H'000050 to H'000053
	IRQ5	21	H'002A to H'002B	H'000054 to H'000057
	IRQ6	22	H'002C to H'002D	H'000058 to H'00005B
	IRQ7	23	H'002E to H'002F	H'00005C to H'00005F
Internal interrupt*		24 ⋮ 29	H'0030 to H'0031 ⋮ H'003A to H'003B	H'000060 to H'000063 ⋮ H'000074 to H'000077
External interrupt	KIN7 to KIN0	30	H'003C to H'003D	H'000078 to H'00007B
External interrupt	KIN15 to KIN8	31	H'003E to H'003F	H'00007C to H'00007F
External interrupt	WUE7 to WUE0	32	H'0040 to H'0041	H'000080 to H'000083
External interrupt	WUE15 to WUE8	33	H'0042 to H'0043	H'000084 to H'000087

Exception Source		Vector Number	Vector Addresses	
			Normal Mode	Advanced Mode
Internal interrupt*		34 ⎮ 55	H'0044 to H'0045 ⎮ H'006E to H'006F	H'000088 to H'00008B ⎮ H'0000DC to H'0000DF
External interrupt	IRQ8	56	H'0070 to H'0071	H'0000E0 to H'0000E3
	IRQ9	57	H'0072 to H'0073	H'0000E4 to H'0000E7
	IRQ10	58	H'0074 to H'0075	H'0000E8 to H'0000EB
	IRQ11	59	H'0076 to H'0077	H'0000EC to H'0000EF
	IRQ12	60	H'0078 to H'0079	H'0000F0 to H'0000F3
	IRQ13	61	H'007A to H'007B	H'0000F4 to H'0000F7
	IRQ14	62	H'007C to H'007D	H'0000F8 to H'0000FB
	IRQ15	63	H'007E to H'007F	H'0000FC to H'0000FF
Internal interrupt*		64 ⎮ 127	H'0080 to H'0081 ⎮ H'00FE to H'00FF	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 Table.

4.3 Reset

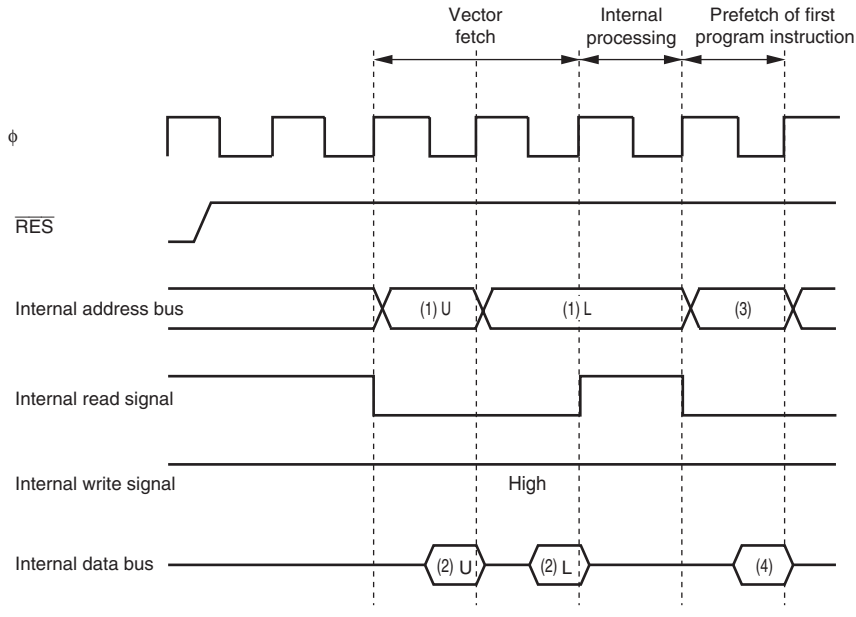
A reset has the highest exception priority. When the $\overline{\text{RES}}$ pin goes low, all processing halts and this LSI enters the reset state. To ensure that this LSI is reset, hold the $\overline{\text{RES}}$ pin low for at least 20 ms at power-on. To reset the chip during operation, hold the $\overline{\text{RES}}$ 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, and MSTPCRA) 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 24, 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 WUE0) 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 Stack Status after Exception Handling

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

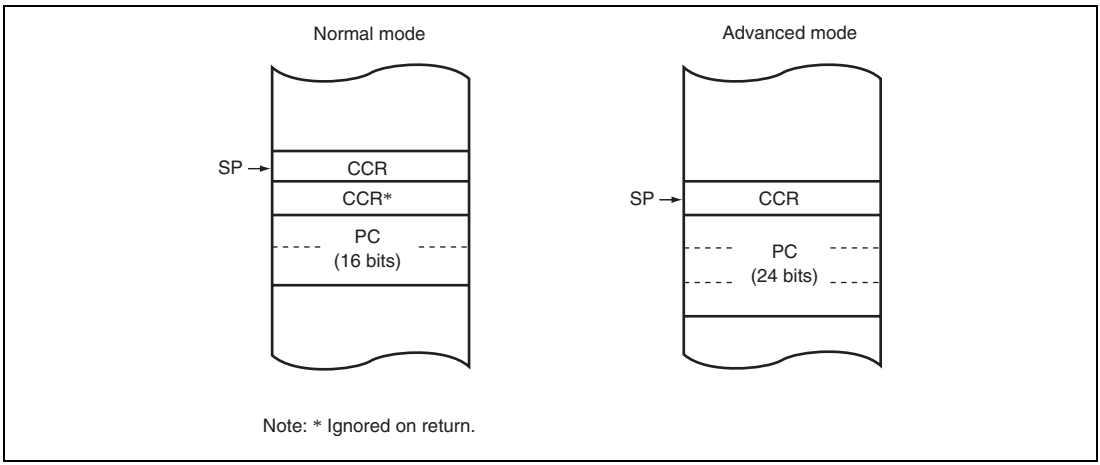


Figure 4.2 Stack Status after Exception Handling

4.7 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.

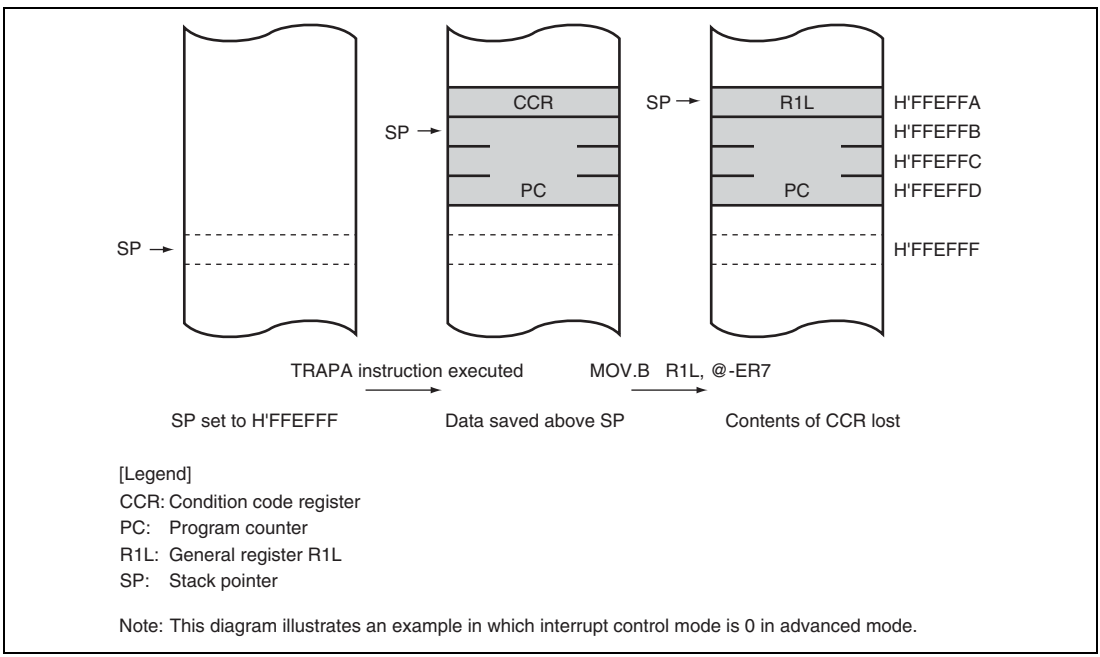


Figure 4.3 Operation when SP Value Is Odd

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.
- Independent vector addresses

  All interrupt sources are assigned independent vector addresses, making it unnecessary for the source to be identified in the interrupt handling routine.
- Forty-nine 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 $\overline{\text{IRQ7}}$, $\overline{\text{KIN15}}$ to $\overline{\text{KIN8}}$, or $\overline{\text{WUE7}}$ to $\overline{\text{WUE0}}$. When the EIVS bit in the system control register 3 (SYSCR3) is set to 1, an interrupt is requested at the falling edge of $\overline{\text{KIN15}}$ to $\overline{\text{KIN0}}$ and $\overline{\text{WUE15}}$ to $\overline{\text{WUE0}}$.
- DTC control

  The DTC can be activated by an interrupt request.
- 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, KIN15 to KIN8, and WUE7 to WUE0 interrupts.
- General ports for $\overline{\text{IRQ15}}$ to $\overline{\text{IRQ0}}$ input are selectable

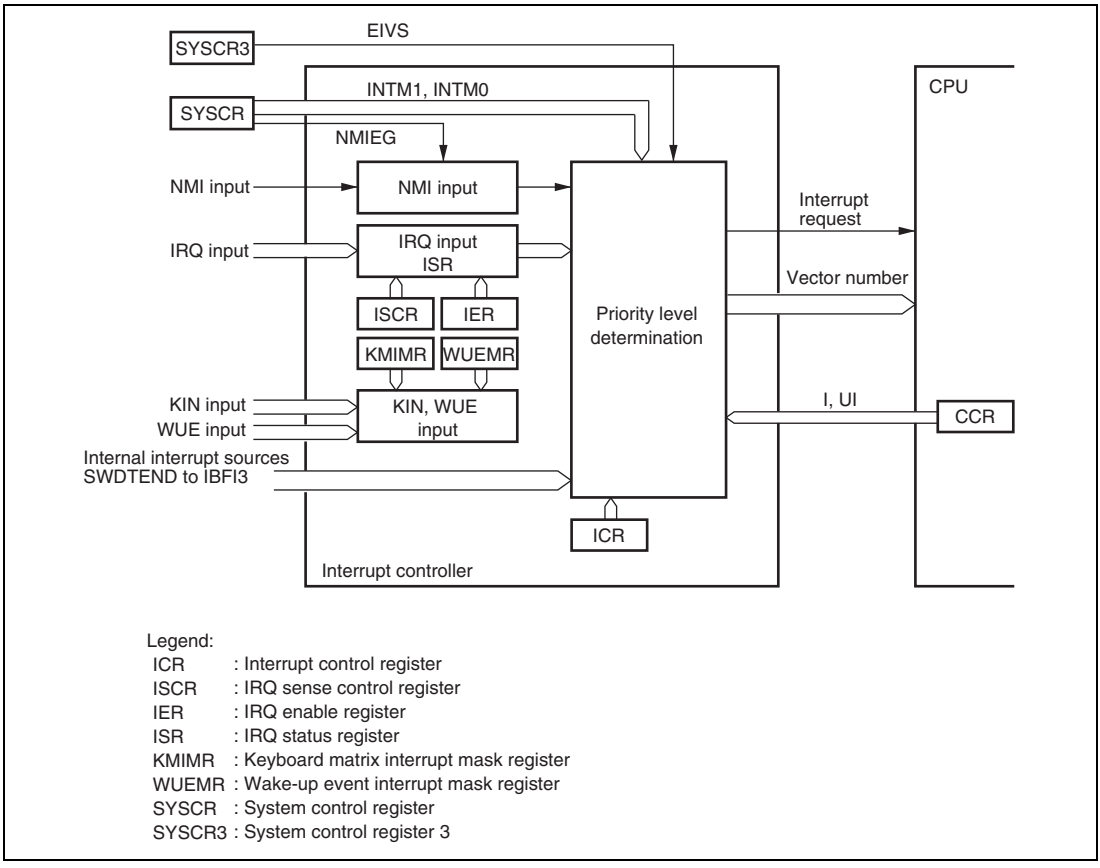


Figure 5.1 Block Diagram of Interrupt Controller

5.2 Input/Output Pins

Table 5.1 summarizes the pins of the interrupt controller.

Table 5.1 Pin Configuration

Symbol	I/O	Function
NMI	Input	Nonmaskable external interrupt pin Rising edge or falling edge can be selected
$\overline{\text{IRQ15}}$ to $\overline{\text{IRQ0}}$, $\overline{\text{ExIRQ15}}$ to $\overline{\text{ExIRQ0}}$	Input	Maskable external 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 IRQ0 interrupt is input can be selected from the $\overline{\text{IRQn}}$ and $\overline{\text{ExIRQn}}$ pins. (n = 15 to 0)
$\overline{\text{KIN15}}$ 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 An interrupt is requested at the falling edge.
$\overline{\text{WUE7}}$ to $\overline{\text{WUE0}}$	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.

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).

- Interrupt control registers A to D (ICRA to ICRD)
- Address break control register (ABRKCR)
- Break address registers A to C (BARA to BARC)
- IRQ sense control registers (ISCR16H, ISCR16L, ISCRH, ISCRL)
- IRQ enable registers (IER16, IER)
- IRQ status registers (ISR16, ISR)
- Keyboard matrix interrupt mask registers (KMIMRA, KMIMR)
- Wake-up event interrupt mask registers (WUEMR, WUEMRB)
- IRQ sense port select registers (ISSR16, ISSR)

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 5.2 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	—	IRQ8 to IRQ11
6	ICRn6	IRQ1	FRT	SCI_1	IRQ12 to IRQ15
5	ICRn5	IRQ2, IRQ3	—	SCI_2	—
4	ICRn4	IRQ4, IRQ5	—	IIC_0	WUE8 to WUE15
3	ICRn3	IRQ6, IRQ7	TMR_0	IIC_1	TPU_0
2	ICRn2	DTC	TMR_1	—	TPU_1
1	ICRn1	WDT_0	TMR_X, TMR_Y	LPC	TPU_2
0	ICRn0	WDT_1	KBU	—	—

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

Table 5.3 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	—	IRQ8 to IRQ11
6	ICRn6	IRQ1	FRT	SCI_1	IRQ12 to IRQ15
5	ICRn5	IRQ2, IRQ3	—	SCI_2	KIN0 to KIN15
4	ICRn4	IRQ4, IRQ5	—	IIC_0	WUE0 to WUE15
3	ICRn3	IRQ6, IRQ7	TMR_0	IIC_1	TPU channel 0
2	ICRn2	DTC	TMR_1	—	TPU channel 1
1	ICRn1	WDT_0	TMR_X, TMR_Y	LPC	TPU channel 2
0	ICRn0	WDT_1	KBU	—	—

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. In normal mode, addresses A23 to A16 are not compared.

- 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 $\overline{\text{ExIRQ0}}$.

- ISCR16H

Bit	Bit Name	Initial Value	R/W	Description
76	IRQ15SCBIRQ15SCA	00	R/WR/W	IRQn Sense Control B IRQn Sense Control A
54	IRQ14SCBIRQ14SCA	00	R/WR/W	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).
32	IRQ13SCBIRQ13SCA	00	R/WR/W	
10	IRQ12SCBIRQ12SCA	00	R/WR/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).

- ISCRL

Bit	Bit Name	Initial Value	R/W	Description
7	IRQ3SCB	0	R/W	IRQn Sense Control B
6	IRQ3SCA	0	/W	IRQn Sense Control A
5	IRQ2SCB	0	R/W	BA
4	IRQ2SCA	0	R/W	00: Interrupt request generated at low level of $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ input
3	IRQ1SCB	0	R/W	01: Interrupt request generated at falling edge of $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ input
2	IRQ1SCA	0	R/W	10: Interrupt request generated at rising edge of $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ input
1	IRQ0SCB	0	R/W	11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ input
0	IRQ0SCA	0	R/W	(n = 3 to 0)
				Note: The $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ pin is selected by the IRQ sense port select register (ISSR).

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 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)*	
4	IRQ4F	0	R/(W)*	[Clearing conditions]
3	IRQ3F	0	R/(W)*	• When writing 0 to IRQnF flag after reading IRQnF = 1
2	IRQ2F	0	R/(W)*	
1	IRQ1F	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	IRQ0F	0	R/(W)*	• When IRQn interrupt exception handling is executed when falling-edge, rising-edge, or both-edge detection is set
				(n = 7 to 0)
				Note: The $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ pin is selected by the IRQ sense port select register (ISSR).

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, WUEMRB)

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 $\overline{\text{WUE0}}$).

- 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	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	

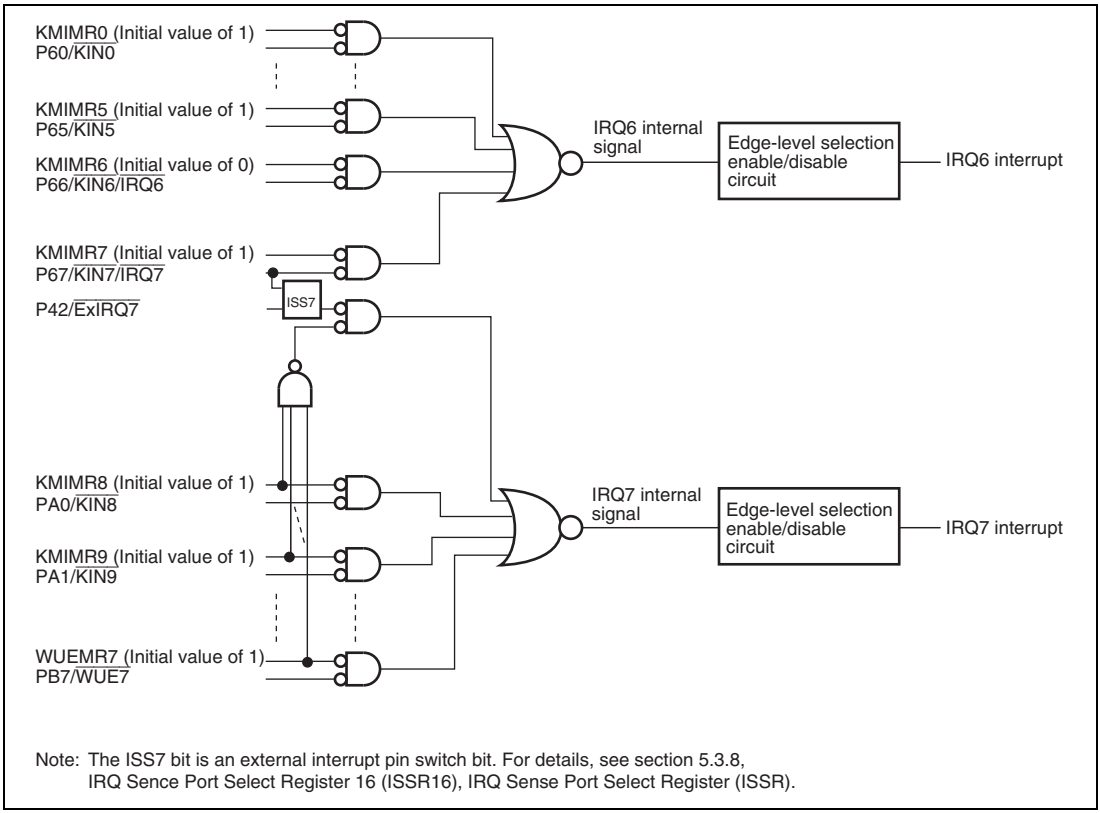
- WUEMR

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	

- WUEMRB

Bit	Bit Name	Initial Value	R/W	Description
7	WUEMR7	1	R/W	Wake-Up Event Interrupt Mask
6	WUEMR6	1	R/W	These bits enable or disable a wake-up event input interrupt request (WUE7 to WUE0).
5	WUEMR5	1	R/W	
4	WUEMR4	1	R/W	0: Enables a wake-up event input interrupt request
3	WUEMR3	1	R/W	1: Disables a wake-up event input interrupt request
2	WUEMR2	1	R/W	
1	WUEMR1	1	R/W	
0	WUEMR0	1	R/W	

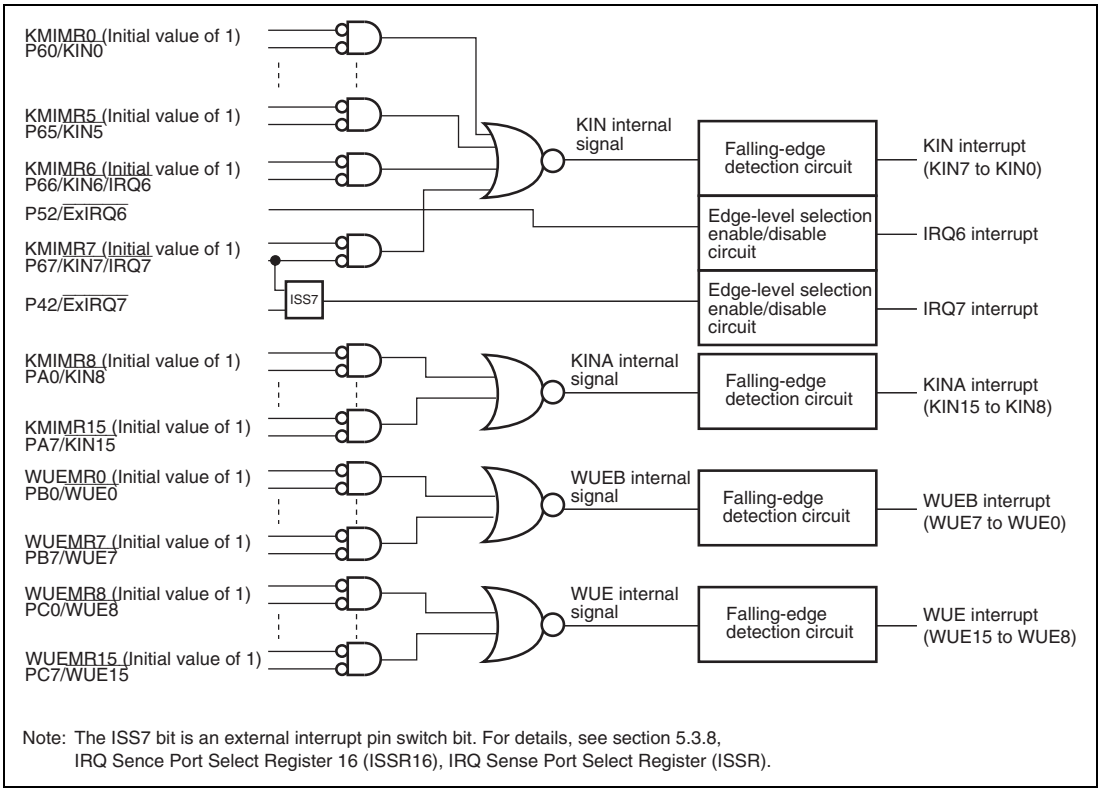
Figure 5.2 shows the relation between the IRQ7 and IRQ6 interrupts, KIN15 to KIN0 interrupts, WUE7 to WUE0 interrupts, KMIMR, KMIMRA, and WUEMRB in H8S/2140B Group compatible vector mode. The relation in extended vector mode is shown in figure 5.3.



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

Figure 5.2 Relation between IRQ7 and IRQ6 Interrupts, KIN15 to KIN0 Interrupts, WUE7 to WUE0 Interrupts, KMIMR, KMIMRA, and WUEMRB (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 and WUEMR7 to WUEMR0 bits is cleared to 0. If the $\overline{\text{KIN7}}$ to $\overline{\text{KIN0}}$ pins or $\overline{\text{KIN15}}$ to $\overline{\text{KIN8}}$ pins, and $\overline{\text{WUE7}}$ to $\overline{\text{WUE0}}$ 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 Sence Port Select Register 16 (ISSR16), IRQ Sense Port Select Register (ISSR).

Figure 5.3 Relation between IRQ7 and IRQ6 Interrupts, KIN15 to KIN0 Interrupts, WUE7 to WUE0 Interrupts, KMIMR, KMIMRA, and WUEMRB (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 $\overline{\text{IRQ15}}$ to $\overline{\text{IRQ0}}$ pins and $\overline{\text{ExIRQ15}}$ to $\overline{\text{ExIRQ0}}$ 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: P42/$\overline{\text{ExIRQ7}}$ is selected
6	—	0	R/W	Reserved The initial values should not be changed.
5	ISS5	0	R/W	0: P86/$\overline{\text{IRQ5}}$ is selected 1: P75/$\overline{\text{ExIRQ5}}$ is selected
4	ISS4	0	R/W	0: P85/$\overline{\text{IRQ4}}$ is selected 1: P74/$\overline{\text{ExIRQ4}}$ is selected
3	ISS3	0	R/W	0: P84/$\overline{\text{IRQ3}}$ is selected 1: P73/$\overline{\text{ExIRQ3}}$ is selected
2	ISS2	0	R/W	0: P90/$\overline{\text{IRQ2}}$ is selected 1: P72/$\overline{\text{ExIRQ2}}$ is selected
1	ISS1	0	R/W	0: P91/$\overline{\text{IRQ1}}$ is selected 1: P71/$\overline{\text{ExIRQ1}}$ is selected
0	ISS0	0	R/W	0: P92/$\overline{\text{IRQ0}}$ is selected 1: P70/$\overline{\text{ExIRQ0}}$ is selected

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 WUE0. 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 $\overline{\text{ExIRQ0}}$. 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 $\overline{\text{ExIRQ0}}$.
- Enabling or disabling of interrupt requests IRQ15 to IRQ0 can be selected with IER.
- The status of interrupt requests IRQ15 to IRQ0 is indicated in ISR. ISR flags can be 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 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.

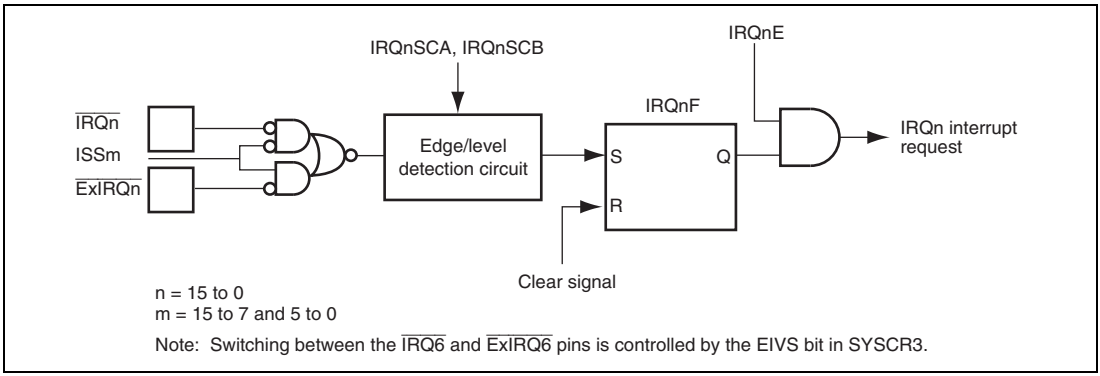


Figure 5.4 Block Diagram of Interrupts IRQ15 to IRQ0

(3) KIN15 to KIN0 Interrupts and WUE15 to WUE0 Interrupts

Interrupts KIN15 to KIN0 and WUE15 to WUE0 are requested by an input signal at pins $\overline{\text{KIN15}}$ to $\overline{\text{KIN0}}$ and $\overline{\text{WUE15}}$ to $\overline{\text{WUE0}}$. Interrupts KIN15 to KIN0 and WUE15 to WUE0 have the following features 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 WUE7 to WUE0 and 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. Interrupt settings for interrupts WUE15 to WUE8 can be made regardless of the settings for interrupts IRQ7 and IRQ6.
  - — Enabling or disabling of interrupt requests KIN15 to KIN0 and WUE15 to WUE0 can be selected using KMIMRA, KMIMR, WUEMRB, and WUEMR.
  - — If the $\overline{\text{KIN7}}$ to $\overline{\text{KIN0}}$ pins or $\overline{\text{WUE15}}$ to $\overline{\text{WUE8}}$ pins, and $\overline{\text{WUE7}}$ to $\overline{\text{WUE0}}$ 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.
  - — 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 and WUEMR7 to WUEMR0 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, KIN7 to KIN0, WUE15 to WUE8, and WUE7 to WUE0 each form a group. The interrupt exception handling for an interrupt request from the same group is started at the same vector address.
  - An interrupt request is generated by a falling edge at pins $\overline{\text{KIN15}}$ to $\overline{\text{KIN0}}$ and $\overline{\text{WUE15}}$ to $\overline{\text{WUE0}}$.
  - Enabling or disabling of interrupt requests KIN15 to KIN0 and WUE15 to WUE0 can be selected using KMIMRA, KMIMR, WUEMRB, and WUEMR.
  - The status of interrupt requests KIN15 to KIN0 and WUE15 to WUE0 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 the KMIMR15 to KMIMR8 and WUEMR7 to WUEMR0 bits.

The detection of interrupts KIN15 to KIN0 and WUE15 to WUE0 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 KIN15 to KIN0 and WUE15 to WUE0 is shown in figure 5.5.

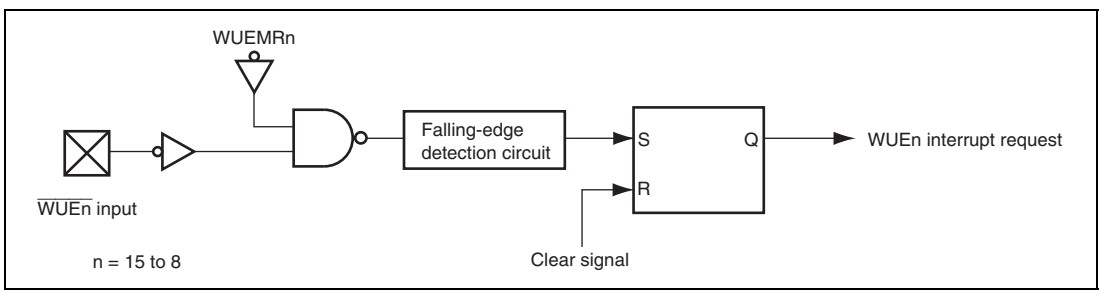


Figure 5.5 Block Diagram of Interrupts KIN15 to KIN0 and WUE15 to WUE0 (Example of WUE15 to WUE8)

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.
- The DTC can be activated by an interrupt request from an on-chip peripheral module.
- An interrupt request that activates the DTC is not affected by the interrupt control mode or the status of the CPU interrupt mask bits.

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

Origin of Interrupt Source	Name	Vector Number	Vector Address: Normal Mode	Vector Address: Advanced Mode	ICR	Priority
External pin	NMI	7	H'000E	H'00001C	—	High
	IRQ0	16	H'0020	H'000040	ICRA7	
	IRQ1	17	H'0022	H'000044	ICRA6	
	IRQ2	18	H'0024	H'000048	ICRA5	
	IRQ3	19	H'0026	H'00004C		
	IRQ4	20	H'0028	H'000050	ICRA4	
	IRQ5	21	H'002A	H'000054		
	IRQ6, KIN7 to KIN0	22	H'002C	H'000058	ICRA3	
	IRQ7, KIN15 to KIN8, WUE7 to WUE0	23	H'002E	H'00005C		
DTC	SWDTEND (Software activation data transfer end)	24	H'0030	H'000060	ICRA2	
WDT_0	WOVI0 (Interval timer)	25	H'0032	H'000064	ICRA1	
WDT_1	WOVI1 (Interval timer)	26	H'0034	H'000068	ICRA0	
—	Address break	27	H'0036	H'00006C	—	
A/D converter	ADI (A/D conversion end)	28	H'0038	H'000070	ICRB7	
	Reserved for system use	29	H'003A	H'000074	—	
	Reserved for system use	30	H'003C	H'000078		
	Reserved for system use	31	H'003E	H'00007C		
External pin	Reserved for system use	32	H'0040	H'000080	ICRD4	
	WUE15 to WUE8	33	H'0042	H'000084		
TPU_0	TGI0A (TGR0A input capture/compare match)	34	H'0044	H'000088	ICRD3	
	TGI0B (TGR0B input capture/compare match)	35	H'0046	H'00008C		
	TGI0C (TGR0C input capture/compare match)	36	H'0048	H'000090		
	TGI0D (TGR0D input capture/compare match)	37	H'004A	H'000094		
	TGI0V (Overflow 0)	38	H'004C	H'000098		
TPU_1	TGI1A (TGR1A input capture/compare match)	39	H'004E	H'00009C	ICRD2	
	TGI1B (TGR1B input capture/compare match)	40	H'0050	H'0000A0		
	TGI1V (Overflow 1)	41	H'0052	H'0000A4		
	TGI1U (Underflow 1)	42	H'0054	H'0000A8		Low

Origin of Interrupt Source	Name	Vector Number	Vector Address: Normal Mode	Vector Address: Advanced Mode	ICR	Priority
TPU_2	TGI2A (TGR2A input capture/compare match)	43	H'0056	H'0000AC	ICRD1	High
	TGI2B (TGR2B input capture/compare match)	44	H'0058	H'0000B0		
	TGI2V (Overflow 2)	45	H'005A	H'0000B4		
	TGI2U (Underflow 2)	46	H'005C	H'0000B8		
	Reserved for system use	47	H'005E	H'0000BC		
FRT	ICIA (Input capture A)	48	H'0060	H'0000C0	ICRB6	
	ICIB (Input capture B)	49	H'0062	H'0000C4		
	ICIC (Input capture C)	50	H'0064	H'0000C8		
	ICID (Input capture D)	51	H'0066	H'0000CC		
	OCIA (Output compare A)	52	H'0068	H'0000D0		
	OCIB (Output compare B)	53	H'006A	H'0000D4		
	FOVI (Overflow)	54	H'006C	H'0000D8		
	Reserved for system use	55	H'006E	H'0000DC		
External pin	IRQ8	56	H'0070	H'0000E0	ICRD7	
	IRQ9	57	H'0072	H'0000E4		
	IRQ10	58	H'0074	H'0000E8		
	IRQ11	59	H'0076	H'0000EC		
	IRQ12	60	H'0078	H'0000F0	ICRD6	
	IRQ13	61	H'007A	H'0000F4		
	IRQ14	62	H'007C	H'0000F8		
	IRQ15	63	H'007E	H'0000FC		
TMR_0	CMIA0 (Compare match A)	64	H'0080	H'000100	ICRB3	
	CMIB0 (Compare match B)	65	H'0082	H'000104		
	OVI0 (Overflow)	66	H'0084	H'000108		
	Reserved for system use	67	H'0086	H'00010C		
TMR_1	CMIA1 (Compare match A)	68	H'0088	H'000110	ICRB2	
	CMIB1 (Compare match B)	69	H'008A	H'000114		
	OVI1 (Overflow)	70	H'008C	H'000118		
	Reserved for system use	71	H'008E	H'00011C		
TMR_X TMR_Y	CMIAY (Compare match A)	72	H'0090	H'000120	ICRB1	
	CMIBY (Compare match B)	73	H'0092	H'000124		
	OVIY (Overflow)	74	H'0094	H'000128		
	ICIX (Input capture)	75	H'0096	H'00012C		
	CMIAX (Compare match A)	76	H'0098	H'000130		
	CMIBX (Compare match B)	77	H'009A	H'000134		
	OVIX (Overflow)	78	H'009C	H'000138		
	Reserved for system use	79	H'009E	H'00013C	—	
	Reserved for system use	80	H'00A0	H'000140		
	Reserved for system use	81	H'00A2	H'000144		
	Reserved for system use	82	H'00A4	H'000148		
	Reserved for system use	83	H'00A6	H'00014C		
SCI_1	ERI1 (Reception error 1)	84	H'00A8	H'000150	ICRC6	
	RXI1 (Reception completion 1)	85	H'00AA	H'000154		
	TXI1 (Transmission data empty 1)	86	H'00AC	H'000158		
	TEI1 (Transmission end 1)	87	H'00AE	H'00015C		Low

Origin of Interrupt Source	Name	Vector Number	Vector Address: Normal Mode	Vector Address: Advanced Mode	ICR	Priority
SCI_2	ERI2 (Reception error 2)	88	H'00B0	H'000160	ICRC5	High
	RXI2 (Reception completion 2)	89	H'00B2	H'000164		
	TXI2 (Transmission data empty 2)	90	H'00B4	H'000168		
	TEI2 (Transmission end 2)	91	H'00B6	H'00016C		
IIC_0	IICI0 (1-byte transmission/reception completion)	92	H'00B8	H'000170	ICRC4	
	Reserved for system use	93	H'00BA	H'000174		
IIC_1	IICI1 (1-byte transmission/reception completion)	94	H'00BC	H'000178	ICRC3	
	Reserved for system use	95	H'00BE	H'00017C		
KBU	KBIA (Reception completion A)	96	H'00C0	H'000180	ICRB0	
	KBIB (Reception completion B)	97	H'00C2	H'000184		
	KBIC (Reception completion C)	98	H'00C4	H'000188		
	KBTIA (Transmission completion A)/ KBCA (1st KCLKA)	99	H'00C6	H'00018C		
	KBTIB (Transmission completion B)/ KBCB (1st KCLKB)	100	H'00C8	H'000190		
	KBTIC (Transmission completion C)/ KBCC (1st KCLKC)	101	H'00CA	H'000194		
	Reserved for system use	102	H'00CC	H'000198		
	Reserved for system use	103	H'00CE	H'00019C		
LPC	Reserved for system use	104	H'00D0	H'0001A0	ICRC1	
	LMCI (LPC/FW command reception completion)	105	H'00D2	H'0001A4		
	LMCUI (LPC/FW user command reception completion)	106	H'00D4	H'0001A8		
	IBFI4 (IDR4 reception completion)	107	H'00D6	H'0001AC		
	ERR1 (Transfer error, etc.)	108	H'00D8	H'0001B0		
	IBFI1 (IDR1 reception completion)	109	H'00DA	H'0001B4		
	IBFI2 (IDR2 reception completion)	110	H'00DC	H'0001B8		
	IBFI3 (IDR3 reception completion)	111	H'00DE	H'0001BC		
	Reserved for system use	112 to 127	H'00E0 to H'00FE	H'0001C0 to H'0001FC	—	Low

Table 5.5 Interrupt Sources, Vector Addresses, and Interrupt Priorities (Extended Vector Mode)

Origin of Interrupt Source	Name	Vector Number	Vector Address: Normal Mode	Vector Address: Advanced Mode	ICR	Priority
External pin	NMI	7	H'000E	H'00001C	—	High
	IRQ0	16	H'0020	H'000040	ICRA7	
	IRQ1	17	H'0022	H'000044	ICRA6	
	IRQ2	18	H'0024	H'000048	ICRA5	
	IRQ3	19	H'0026	H'00004C		
	IRQ4	20	H'0028	H'000050	ICRA4	
	IRQ5	21	H'002A	H'000054		
	IRQ6	22	H'002C	H'000058	ICRA3	
	IRQ7	23	H'002E	H'00005C		
DTC	SWDTEND (Software activation data transfer end)	24	H'0030	H'000060	ICRA2	
WDT_0	WOVI0 (Interval timer)	25	H'0032	H'000064	ICRA1	
WDT_1	WOVI1 (Interval timer)	26	H'0034	H'000068	ICRA0	
—	Address break	27	H'0036	H'00006C	—	
A/D converter	ADI (A/D conversion end)	28	H'0038	H'000070	ICRB7	
	Reserved for system use	29	H'003A	H'000074	—	
External pin	KIN7 to KIN0	30	H'003C	H'000078	ICRD5	
	KIN15 to KIN8	31	H'003E	H'00007C		
	WUE7 to WUE0	32	H'0040	H'000080	ICRD4	
	WUE15 to WUE8	33	H'0042	H'000084		
TPU_0	TGI0A (TGR0A input capture/compare match)	34	H'0044	H'000088	ICRD3	
	TGI0B (TGR0B input capture/compare match)	35	H'0046	H'00008C		
	TGI0C (TGR0C input capture/compare match)	36	H'0048	H'000090		
	TGI0D (TGR0D input capture/compare match)	37	H'004A	H'000094		
	TGI0V (Overflow 0)	38	H'004C	H'000098		
TPU_1	TGI1A (TGR1A input capture/compare match)	39	H'004E	H'00009C	ICRD2	
	TGI1B (TGR1B input capture/compare match)	40	H'0050	H'0000A0		
	TGI1V (Overflow 1)	41	H'0052	H'0000A4		
	TGI1U (Underflow 1)	42	H'0054	H'0000A8		Low

Origin of Interrupt Source	Name	Vector Number	Vector Address: Normal Mode	Vector Address: Advanced Mode	ICR	Priority
TPU_2	TGI2A (TGR2A input capture/compare match)	43	H'0056	H'0000AC	ICRD1	High
	TGI2B (TGR2B input capture/compare match)	44	H'0058	H'0000B0		
	TGI2V (Overflow 1)	45	H'005A	H'0000B4		
	TGI2U (Underflow 2)	46	H'005C	H'0000B8		
	Reserved for system use	47	H'005E	H'0000BC		
FRT	ICIA (Input capture A)	48	H'0060	H'0000C0	ICRB6	
	ICIB (Input capture B)	49	H'0062	H'0000C4		
	ICIC (Input capture C)	50	H'0064	H'0000C8		
	ICID (Input capture D)	51	H'0066	H'0000CC		
	OCIA (Output compare A)	52	H'0068	H'0000D0		
	OCIB (Output compare B)	53	H'006A	H'0000D4		
	FOVI (Overflow)	54	H'006C	H'0000D8		
	Reserved for system use	55	H'006E	H'0000DC		
External pin	IRQ8	56	H'0070	H'0000E0	ICRD7	
	IRQ9	57	H'0072	H'0000E4		
	IRQ10	58	H'0074	H'0000E8		
	IRQ11	59	H'0076	H'0000EC		
	IRQ12	60	H'0078	H'0000F0	ICRD6	
	IRQ13	61	H'007A	H'0000F4		
	IRQ14	62	H'007C	H'0000F8		
	IRQ15	63	H'007E	H'0000FC		
TMR_0	CMIA0 (Compare match A)	64	H'0080	H'000100	ICRB3	
	CMIB0 (Compare match B)	65	H'0082	H'000104		
	OVI0 (Overflow)	66	H'0084	H'000108		
	Reserved for system use	67	H'0086	H'00010C		
TMR_1	CMIA1 (Compare match A)	68	H'0088	H'000110	ICRB2	
	CMIB1 (Compare match B)	69	H'008A	H'000114		
	OVI1 (Overflow)	70	H'008C	H'000118		
	Reserved for system use	71	H'008E	H'00011C		
TMR_X	CMIAY (Compare match A)	72	H'0090	H'000120	ICRB1	
TMR_Y	CMIBY (Compare match B)	73	H'0092	H'000124		
	OVIY (Overflow)	74	H'0094	H'000128		
	ICIX (Input capture)	75	H'0096	H'00012C		
	CMIAX (Compare match A)	76	H'0098	H'000130		
	CMIBX (Compare match B)	77	H'009A	H'000134		
	OVIX (Overflow)	78	H'009C	H'000138		
—	Reserved for system use	79	H'009E	H'00013C	—	
	Reserved for system use	80	H'00A0	H'000140		
	Reserved for system use	81	H'00A2	H'000144		
	Reserved for system use	82	H'00A4	H'000148		
	Reserved for system use	83	H'00A6	H'00014C		Low

Origin of Interrupt Source	Name	Vector Number	Vector Address: Normal Mode	Vector Address: Advanced Mode	ICR	Priority
SCI_1	ERI1 (Reception error 1)	84	H'00A8	H'000150	ICRC6	High
	RXI1 (Reception completion 1)	85	H'00AA	H'000154		
	TXI1 (Transmission data empty 1)	86	H'00AC	H'000158		
	TEI1 (Transmission end 1)	87	H'00AE	H'00015C		
SCI_2	ERI2 (Reception error 2)	88	H'00B0	H'000160	ICRC5	
	RXI2 (Reception completion 2)	89	H'00B2	H'000164		
	TXI2 (Transmission data empty 2)	90	H'00B4	H'000168		
	TEI2 (Transmission end 2)	91	H'00B6	H'00016C		
IIC_0	IICI0 (1-byte transmission/reception completion)	92	H'00B8	H'000170	ICRC4	
	Reserved for system use	93	H'00BA	H'000174		
IIC_1	IICI1 (1-byte transmission/reception completion)	94	H'00BC	H'000178	ICRC3	
	Reserved for system use	95	H'00BE	H'00017C		
KBU	KBIA (Reception completion A)	96	H'00C0	H'000180	ICRB0	
	KBIB (Reception completion B)	97	H'00C2	H'000184		
	KBIC (Reception completion C)	98	H'00C4	H'000188		
	KBTIA (Transmission completion A)/ KBCA (1st KCLKA)	99	H'00C6	H'00018C		
	KBTIB (Transmission completion B)/ KBCB (1st KCLKB)	100	H'00C8	H'000190		
	KBTIC (Transmission completion C)/ KBCC (1st KCLKC)	101	H'00CA	H'000194		
	Reserved for system use	102	H'00CC	H'000198		
	Reserved for system use	103	H'00CE	H'00019C		
LPC	Reserved for system use	104	H'00D0	H'0001A0	ICRC1	
	LMCI (LPC/FW command reception completion)	105	H'00D2	H'0001A4		
	LMCUI (LPC/FW user command reception completion)	106	H'00D4	H'0001A8		
	IBFI4 (IDR4 reception completion)	107	H'00D6	H'0001AC		
	ERR1 (Transfer error, etc.)	108	H'00D8	H'0001B0		
	IBFI1 (IDR1 reception completion)	109	H'00DA	H'0001B4		
	IBFI2 (IDR2 reception completion)	110	H'00DC	H'0001B8		
	IBFI3 (IDR3 reception completion)	111	H'00DE	H'0001BC		
	Reserved for system use	112 to 127	H'00E0 to H'00FE	H'0001C0 to 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 or in hardware standby mode. The interrupt control mode is selected by SYSCR. Table 5.6 shows the interrupt control modes.

Table 5.6 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.

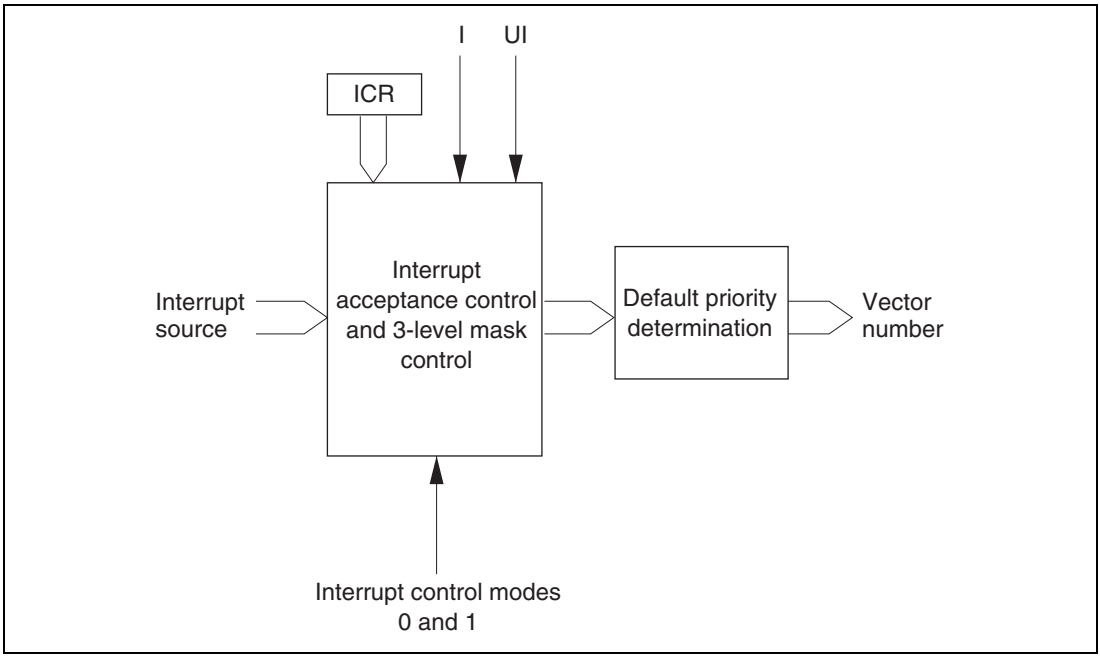


Figure 5.6 Block Diagram of Interrupt Control Operation

(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 5.7 shows the interrupts selected in each interrupt control mode.

Table 5.7 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 5.8 shows operations and control signal functions in each interrupt control mode.

Table 5.8 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.

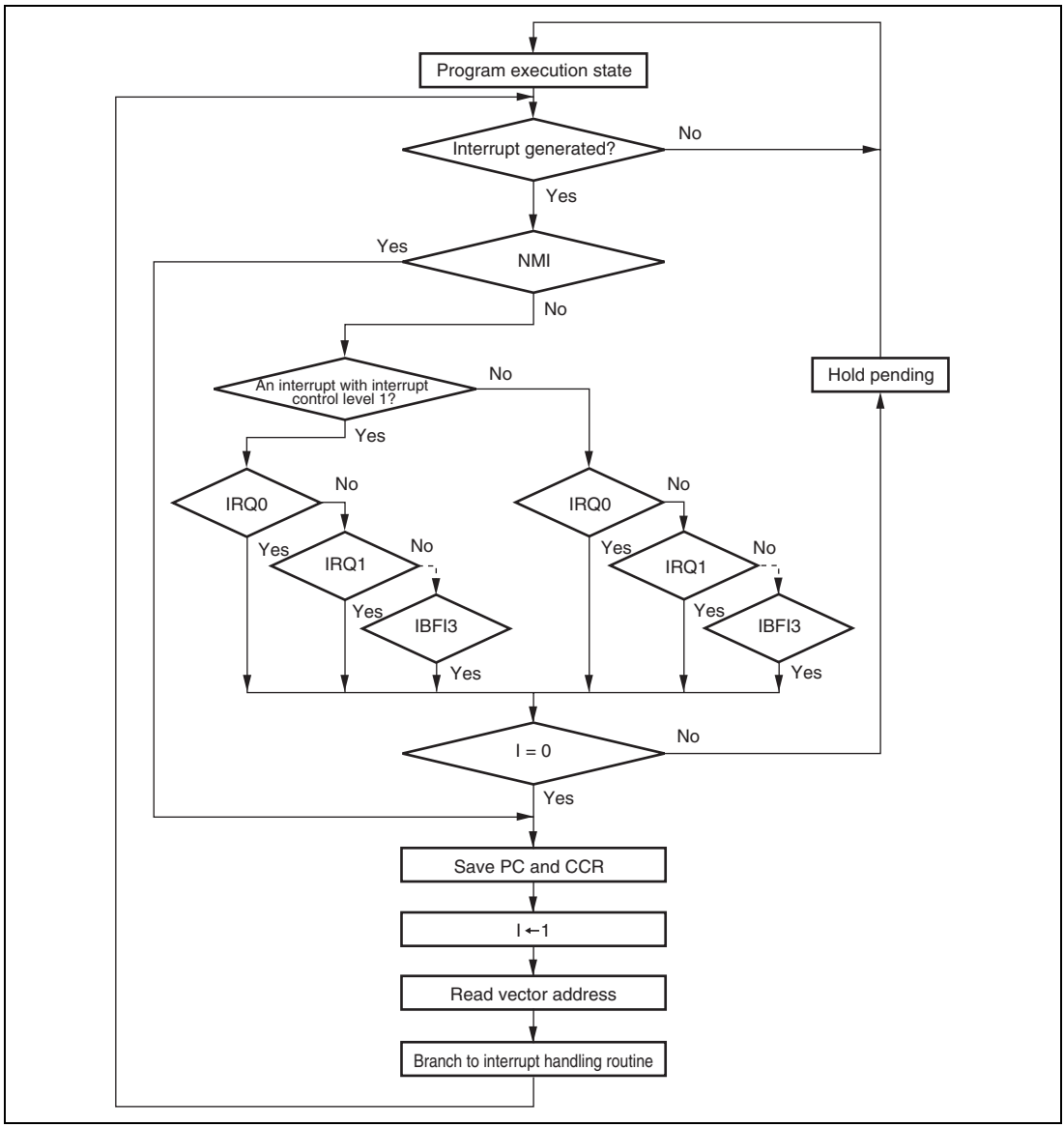


Figure 5.7 Flowchart of Procedure up to Interrupt Acceptance in Interrupt Control Mode 0

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.

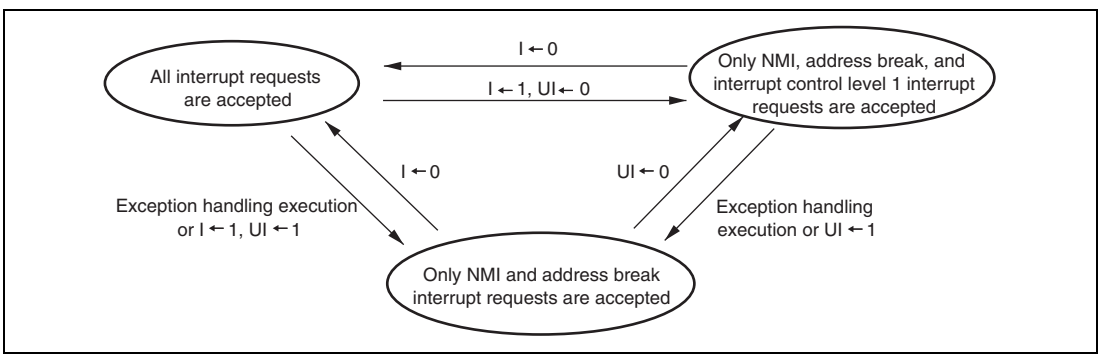


Figure 5.8 State Transition in Interrupt Control Mode 1

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.

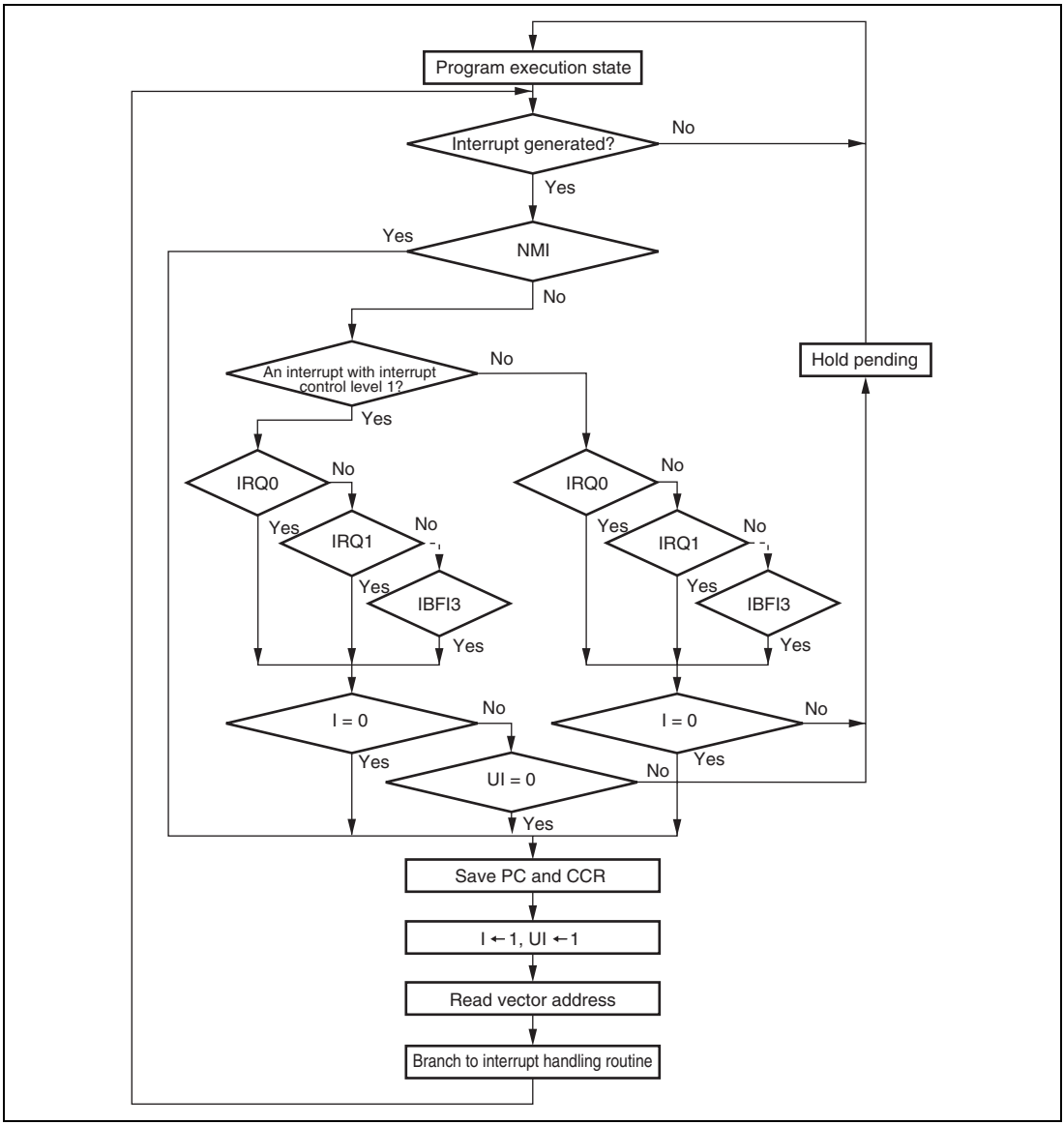


Figure 5.9 Flowchart of Procedure up to Interrupt Acceptance in Interrupt Control Mode 1

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.

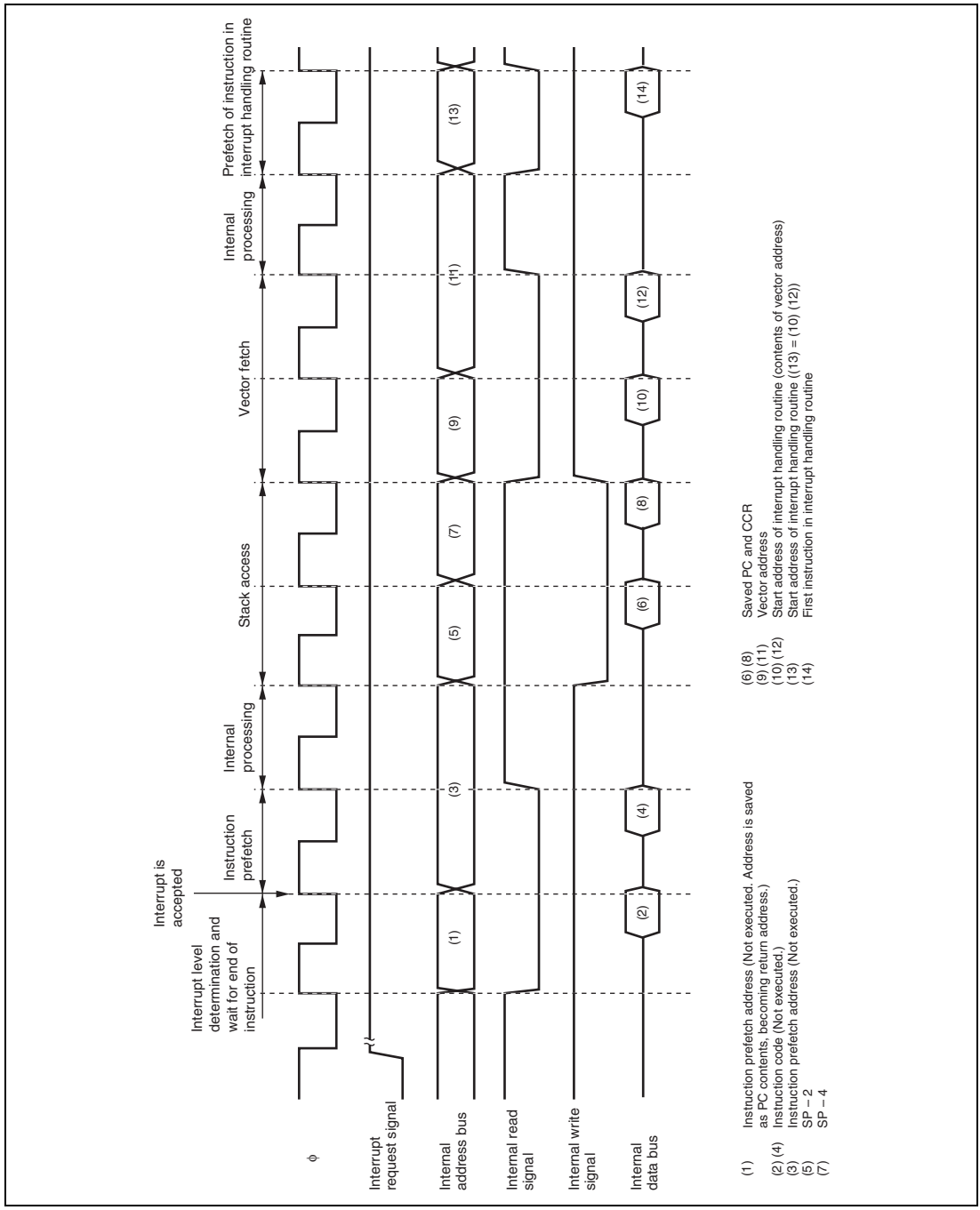


Figure 5.10 Interrupt Exception Handling

5.6.4 Interrupt Response Times

Table 5.9 shows interrupt response times – the intervals between generation of an interrupt request and execution of the first instruction in the interrupt handling routine.

Table 5.9 Interrupt Response Times

No.	Execution Status	Normal Mode	Advanced Mode
1	Interrupt priority determination*[1]	3	3
2	Number of wait states until executing instruction ends*[2]	1 to 21	1 to 21
3	Saving of PC and CCR in stack	2	2
4	Vector fetch	1	2
5	Instruction fetch*[3]	2	2
6	Internal processing*[4]	2	2
	Total (using on-chip memory)	11 to 31	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.

5.6.5 DTC Activation by Interrupt

The DTC can be activated by an interrupt. In this case, the following options are available:

- Interrupt request to CPU
- Activation request to DTC
- Both of the above

For details on interrupt requests that can be used to activate the DTC, see section 7, Data Transfer Controller (DTC). Figure 5.11 shows a block diagram of the DTC and interrupt controller.

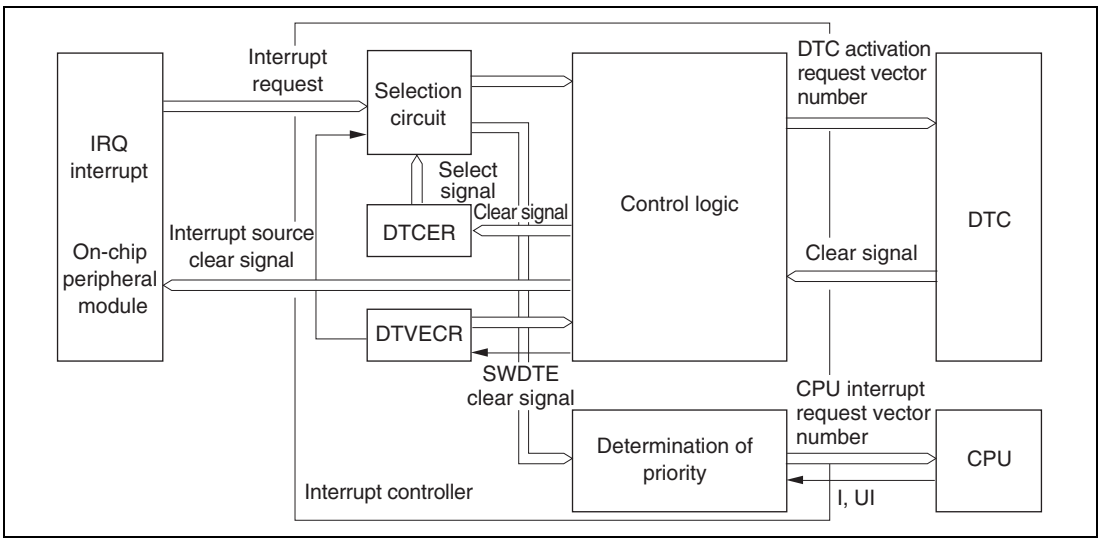


Figure 5.11 Interrupt Control for DTC

The interrupt controller has three main functions in DTC control.

(1) Selection of Interrupt Source

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

(2) Determination of Priority

The DTC activation source is selected in accordance with the default priority order, and is not affected by mask or priority levels. See section 7.4, Location of Register Information and DTC Vector Table, for the respective priorities.

(3) Operation Order

If the same interrupt is selected as a DTC activation source and a CPU interrupt source, the DTC data transfer is performed first, followed by CPU interrupt exception handling.

Table 5.10 summarizes interrupt source selection and interrupt source clearance control according to the settings of the DTCE bit in DTCERA to DTCERE of the DTC and the DISEL bit in MRB of the DTC.

Table 5.10 Interrupt Source Selection and Clearing Control

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

[Legend]

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

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

×: The relevant interrupt cannot be used.

*: Don't care

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 5.12 shows a block diagram of the address break function.

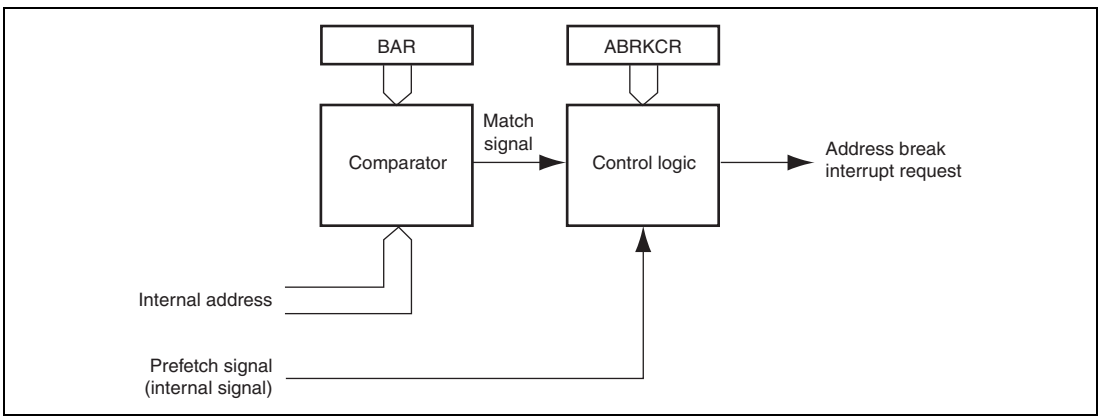


Figure 5.12 Block Diagram of Address Break Function

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.

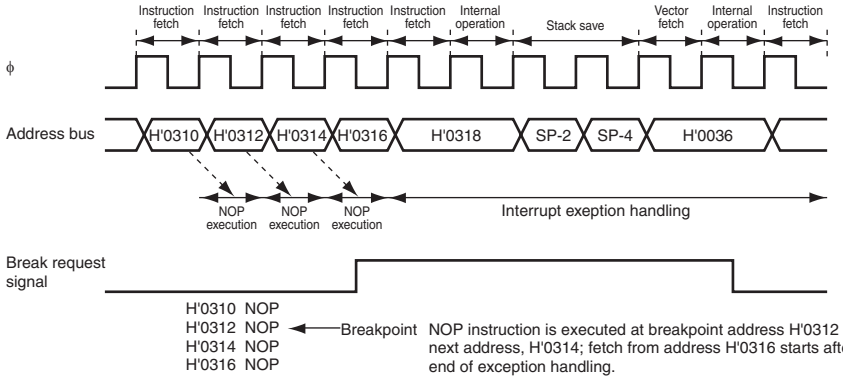
1. Set the break address in bits A23 to A1 in BAR.
2. Set the BIE bit in ABRKCR to 1 to enable address breaks. An address break will not be requested if the BIE bit is cleared to 0.

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.

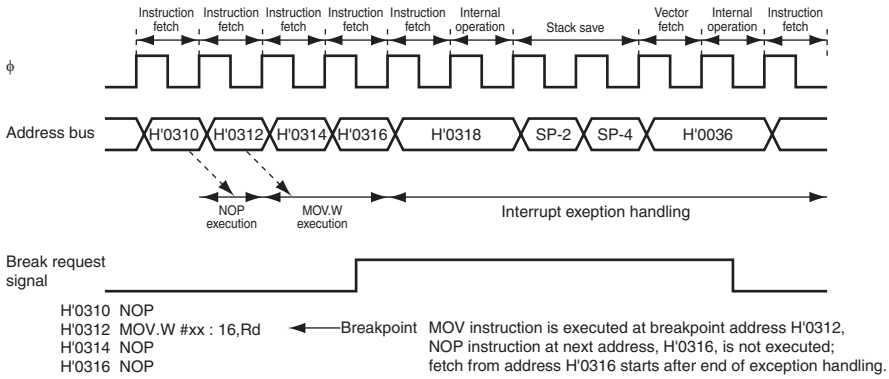
5.7.4 Usage Notes

- 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.
- In normal mode, no comparison is made with address lines A23 to A16.
- 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 5.13 shows some address timing examples.

• Program area in on-chip memory, 1-state execution instruction at specified break address



• Program area in on-chip memory, 2-state execution instruction at specified break address



• Program area in external memory (2-state access, 16-bit-bus access), 1-state execution instruction at specified break address (Not available in this LSI)

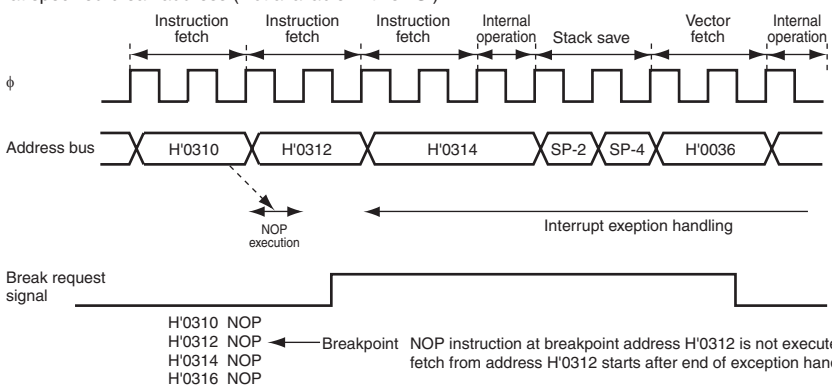


Figure 5.13 Examples of Address Break Timing

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 5.12 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.

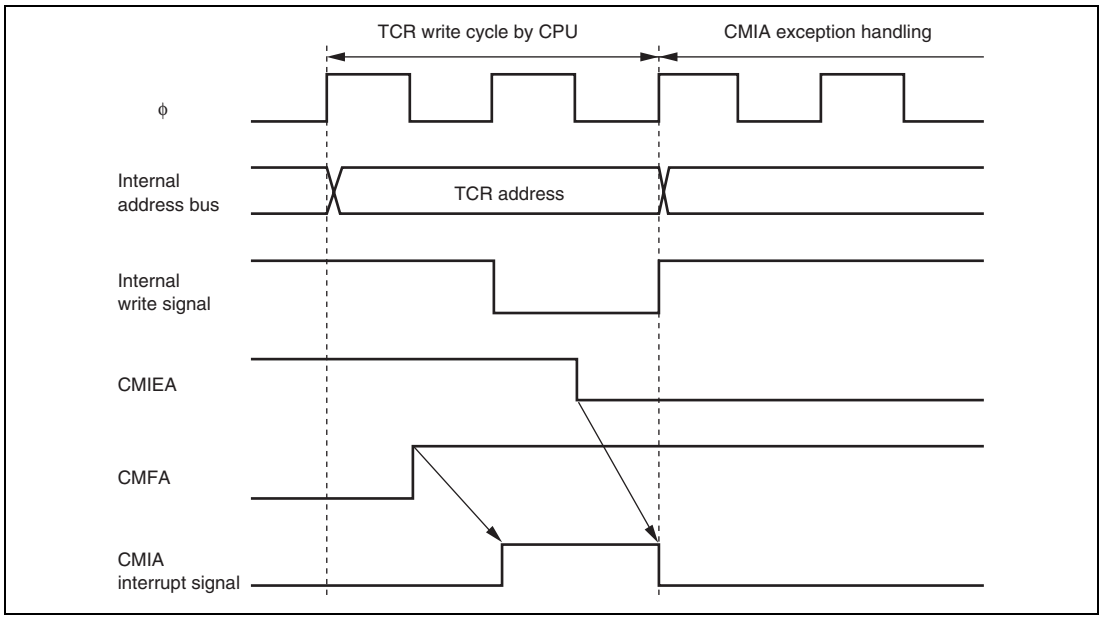


Figure 5.14 Conflict between Interrupt Generation and Disabling

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 $\overline{\text{WUE0}}$ 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 $\overline{\text{WUE0}}$ 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 $\overline{\text{ExIRQ0}}$, $\overline{\text{KIN15}}$ to $\overline{\text{KIN0}}$, and $\overline{\text{WUE15}}$ to $\overline{\text{WUE0}}$) 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)

This LSI has an on-chip bus controller (BSC).

The BSC has a bus arbitration function, and controls the operation of the internal bus masters – CPU, data transfer controller (DTC), and LPC interface (LPC).

Though this LSI does not have external extended functions, take care not to set inappropriate values in the control registers related to the bus controller when utilizing software with other similar products.

6.1 Features

- Bus arbitration function
  Includes a bus arbiter that arbitrates bus mastership between the CPU, DTC, and LPC.

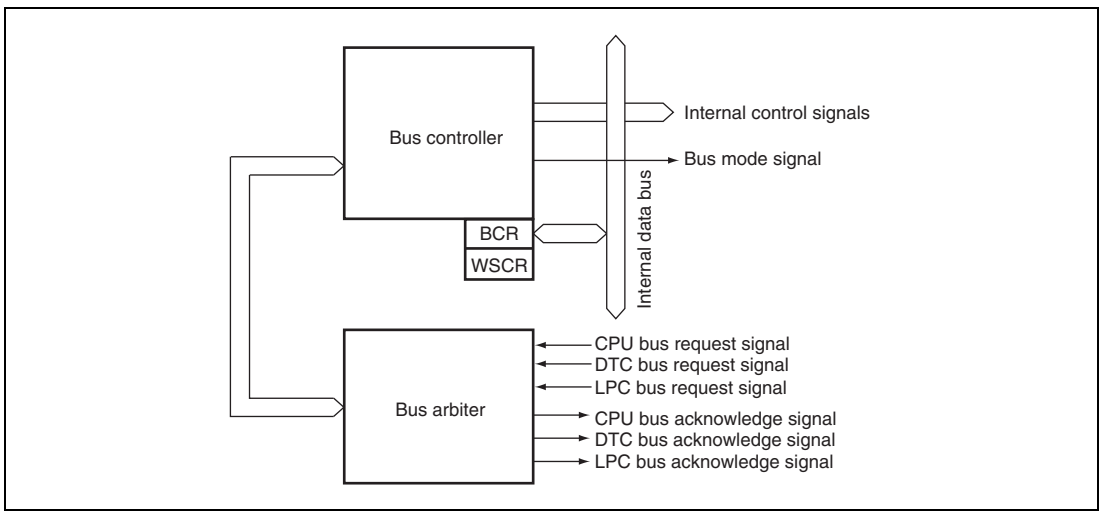


Figure 6.1 Block Diagram of BSC

6.2 Register Descriptions

The bus controller has the following registers.

- Bus control register (BCR)
- Wait state control register (WSCR)

6.2.1 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, 0
0	IOS0	1	R/W	The initial value should not be changed.

6.2.2 Wait State Control Register (WSCR)

Bit	Bit Name	Initial Value	R/W	Description
7	—	1	R/W	Reserved
6	—	1	R/W	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, 0
2	WMS0	0	R/W	The initial value should not be changed.
1	WC1	1	R/W	Wait Count 1, 0
0	WC0	1	R/W	The initial value should not be changed.

6.3 Bus Arbitration

The BSC has a bus arbiter that arbitrates bus master operations. There are three bus masters – CPU, DTC, and LPC – which perform read/write operations when they have possession of the bus.

6.3.1 Priority of Bus Masters

Each bus master requests the bus by means of a bus request signal. The bus arbiter detects the bus masters' bus request signals, and sends a bus request acknowledge signal to the bus master making the request at the designated timing. If there are bus requests from more than one bus master, the bus request acknowledge signal is sent to the one with the highest priority. When a bus master receives the bus request acknowledge signal, it takes possession of the bus until that signal is canceled. The priority order of the bus masters is as follows:

(High) LPC > DTC > CPU (Low)

6.3.2 Bus Transfer Timing

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

(1) CPU

The CPU is the lowest-priority bus master, and if a bus request is received from the DTC or LPC, the bus arbiter transfers the bus to the DTC or LPC.

- DTC and LPC bus transfer timing
  - The bus is transferred at a break between bus cycles.

    However, if a bus cycle is executed in discrete operations, as in the case of a longword-size access, the bus is not transferred between the component operations. For details, see section 2.7, Bus States During Instruction Execution, in the H8S/2600 Series, H8S/2000 Series Programming Manual.
  - If the CPU is in sleep mode, the bus is transferred immediately.

(2) DTC

The DTC sends the bus arbiter a request for the bus when an activation request is generated.

The DTC is a bus master with lower priority than the LPC, and if a bus request is received from the LPC, the bus arbiter transfers the bus to the LPC.

- LPC bus transfer timing

  The bus is transferred at a break between bus cycles.

  If a bus cycle is executed in discrete operations, as in the case of a longword-size access, the bus is not transferred between the component operations. Similarly, in case of a 32-bit access by the DTC, the bus is not transferred between the component operations for each longword.

(3) LPC

The LPC has the highest bus master priority. The LPC sends the bus arbiter a request for the bus when an activation request is generated. The LPC does not release the bus until it completes reading/writing the on-chip memory. For details, see section 18, LPC Interface (LPC).

Section 7 Data Transfer Controller (DTC)

This LSI includes a data transfer controller (DTC). The DTC can be activated by an interrupt or software, to transfer data.

Figure 7.1 shows a block diagram of the DTC. The DTC's register information is stored in the on-chip RAM. When the DTC is used, the RAME bit in SYSCR must be set to 1. A 32-bit bus connects the DTC to addresses H'(FF)EC00 to H'(FF)EFFF in on-chip RAM (1 kbyte), enabling 32-bit/1-state reading and writing of the DTC register information.

7.1 Features

- Transfer is possible over any number of channels
- Three transfer modes

  Normal, repeat, and block transfer modes are available
- One activation source can trigger a number of data transfers (chain transfer)
- Direct specification of 16 Mbytes address space is possible
- Activation by software is possible
- Transfer can be set in byte or word units
- A CPU interrupt can be requested for the interrupt that activated the DTC

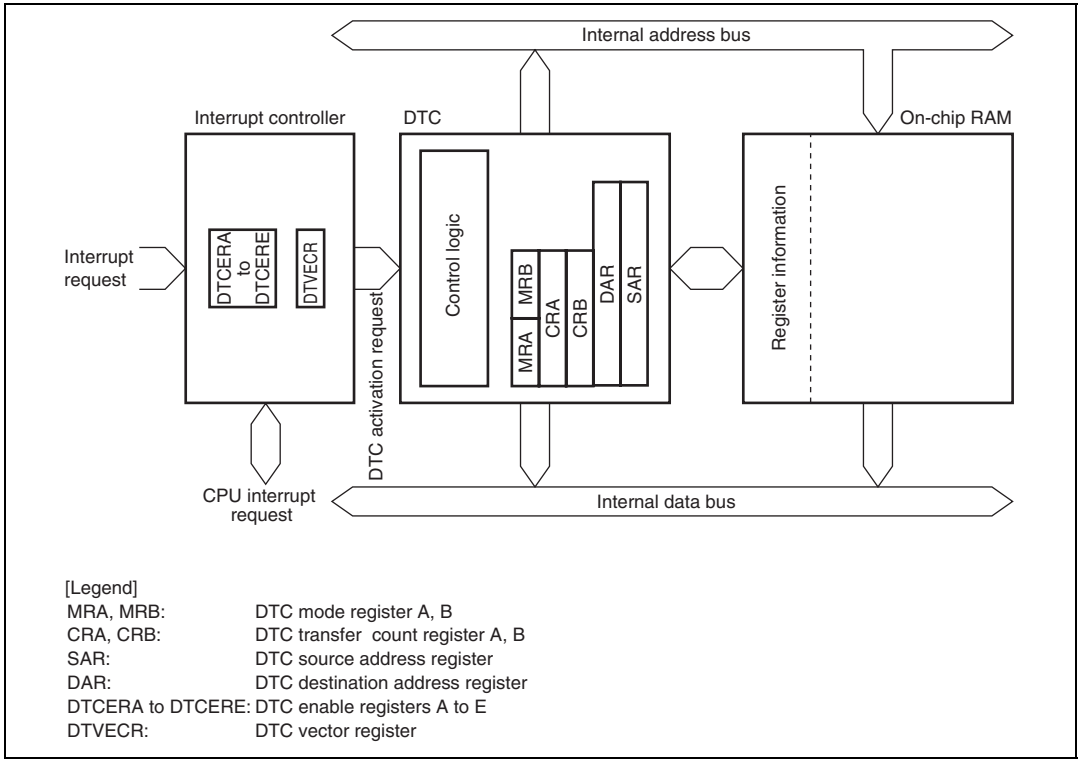


Figure 7.1 Block Diagram of DTC

7.2 Register Descriptions

The DTC has the following registers.

- DTC mode register A (MRA)
- DTC mode register B (MRB)
- DTC source address register (SAR)
- DTC destination address register (DAR)
- DTC transfer count register A (CRA)
- DTC transfer count register B (CRB)

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

- DTC enable register (DTCER)
- DTC vector register (DTVECR)

7.2.1 DTC Mode Register A (MRA)

MRA selects the DTC operating mode.

Bit	Bit Name	Initial Value	R/W	Description
7	SM1	Undefined	—	Source Address Mode 1, 0
6	SM0	Undefined	—	These bits specify an SAR operation after a data transfer. 0*: SAR is fixed 10: SAR is incremented after a transfer (by +1 when Sz = 0, by +2 when Sz = 1) 11: SAR is decremented after a transfer (by –1 when Sz = 0, by –2 when Sz = 1)
5	DM1	Undefined	—	Destination Address Mode 1, 0
4	DM0	Undefined	—	These bits specify a DAR operation after a data transfer. 0*: DAR is fixed 10: DAR is incremented after a transfer (by +1 when Sz = 0, by +2 when Sz = 1) 11: DAR is decremented after a transfer (by –1 when Sz = 0, by –2 when Sz = 1)
3	MD1	Undefined	—	DTC Mode
2	MD0	Undefined	—	These bits specify the DTC transfer mode. 00: Normal mode 01: Repeat mode 10: Block transfer mode 11: Setting prohibited
1	DTS	Undefined	—	DTC Transfer Mode Select Specifies whether the source side or the destination side is set to be a repeat area or block area in repeat mode or block transfer mode. 0: Destination side is repeat area or block area 1: Source side is repeat area or block area
0	Sz	Undefined	—	DTC Data Transfer Size Specifies the size of data to be transferred. 0: Byte-size transfer 1: Word-size transfer

[Legend]

*: Don't care

7.2.2 DTC Mode Register B (MRB)

MRB selects the DTC operating mode.

Bit	Bit Name	Initial Value	R/W	Description
7	CHNE	Undefined	—	DTC Chain Transfer Enable When this bit is set to 1, a chain transfer will be performed. For details, see section 7.5.4, Chain Transfer. In data transfer with CHNE set to 1, determination of the end of the specified number of data transfers, clearing of the interrupt source flag, and clearing of DTCER are not performed.
6	DISEL	Undefined	—	DTC Interrupt Select When this bit is set to 1, a CPU interrupt request is generated every time data transfer ends. (DTC does not clear the interrupt source flag which is as an activation source, to 0.) When this bit is cleared to 0, a CPU interrupt request is generated only when the specified number of data transfers ends. (DTC does not clear the interrupt source flag which is as an activation source, to 0.)
5 to 0	—	All undefined	—	Reserved These bits have no effect on DTC operation. The write value should always be 0.

7.2.3 DTC Source Address Register (SAR)

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

7.2.4 DTC Destination Address Register (DAR)

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

7.2.5 DTC Transfer Count Register A (CRA)

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

In normal mode, the entire CRA functions as a 16-bit transfer counter (1 to 65536). It is decremented by 1 every time data is transferred, and transfer ends when the count reaches H'0000.

In repeat mode or block transfer mode, the CRA is divided into two parts; the upper eight bits (CRAH) and the lower 8 bits (CRAL). CRAH holds the number of transfers while CRAL functions as an 8-bit transfer counter (1 to 256). CRAL is decremented by 1 every time data is transferred, and the contents of CRAH are sent when the count reaches H'00.

7.2.6 DTC Transfer Count Register B (CRB)

CRB is a 16-bit register that designates the number of times data is to be transferred by the DTC in block transfer mode. It functions as a 16-bit transfer counter (1 to 65536) that is decremented by 1 every time data is transferred, and transfer ends when the count reaches H'0000.

7.2.7 DTC Enable Registers (DTCER)

DTCER specifies DTC activation interrupt sources. DTCER is comprised of five registers: DTCERA to DTCERE. The correspondence between interrupt sources and DTCE bits is shown in tables 7.1 and 7.2. For DTCE bit setting, use bit manipulation instructions such as BSET and BCLR. Multiple DTC activation sources can be set at one time (only at the initial setting) by masking all interrupts and writing data after executing a dummy read on the relevant register.

Bit	Bit Name	Initial Value	R/W	Description
7	DTCEn7	0	R/W	DTC Activation Enable
6	DTCEn6	0	R/W	Setting this bit to 1 specifies a relevant interrupt source as a DTC activation source.
5	DTCEn5	0	R/W	[Clearing conditions]
4	DTCEn4	0	R/W	• When data transfer has ended with the DISEL bit in MRB set to 1
3	DTCEn3	0	R/W	• When the specified number of transfers have ended
2	DTCEn2	0	R/W	These bits are not cleared when the DISEL bit is 0 and the specified number of transfers have not been completed
1	DTCEn1	0	R/W	
0	DTCEn0	0	R/W	

Note: n: A to E

Table 7.1 Correspondence between Interrupt Sources and DTCER

		Register				
Bit	Bit Name	DTCERA	DTCERB	DTCERC	DTCERD	DTCERE
7	DTCEn7	(16)IRQ0	(53)OCIB	(69)CMIB1	(86)TXI1	(34)TGI0A
6	DTCEn6	(17)IRQ1	(39)TGI1A	(72)CMIAY	(89)RXI2	(35)TGI0B
5	DTCEn5	(18)IRQ2	(40)TGI1B	(73)CMIBY	(90)TXI2	(36)TGI0C
4	DTCEn4	(19)IRQ3	(43)TGI2A	(76)CMIAX	(92)IICI0	(37)TGI0D
3	DTCEn3	(28)ADI	(44)TGI2B	(77)CMIBX	(94)IICI1	(108)ERR1
2	DTCEn2	(48)ICIA	(64)CMIA0	—	—	(109)IBFI1
1	DTCEn1	(49)ICIB	(65)CMIB0	—	—	(110)IBFI2
0	DTCEn0	(52)OCIA	(68)CMIA1	(85)RXI1	—	(111)IBFI3

Note: n : A to E
( ) : Vector number

7.2.8 DTC Vector Register (DTVECR)

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

Bit	Bit Name	Initial Value	R/W	Description
7	SWDTE	0	R/W	DTC Software Activation Enable Setting this bit to 1 activates DTC. Only 1 can always be written to this bit. Writing 0 is enabled only after 1 has been read. [Clearing conditions] • When the DISEL bit is 0 and the specified number of transfers have not ended • When 0 is written to the DISEL bit after a software-activated data transfer end interrupt (SWDTEND) request has been sent to the CPU. [Holding conditions] • When the DISEL bit is set to 1 and data transfer has ended • The specified number of transfers have ended • On data transfer by software activation
6	DTVEC6	0	R/W	DTC Software Activation Vectors 6 to 0 These bits specify a vector number for DTC software activation. The vector address is expressed as H'0400 + (vector number × 2). For example, when DTVEC6 to DTVEC0 = H'10, the vector address is H'0420. When the SWDTE bit is 0, these bits can be written to.
5	DTVEC5	0	R/W	
4	DTVEC4	0	R/W	
3	DTVEC3	0	R/W	
2	DTVEC2	0	R/W	
1	DTVEC1	0	R/W	
0	DTVEC0	0	R/W	

7.3 Activation Sources

The DTC is activated by an interrupt request or by a write to DTVECR by software. The interrupt request source to activate the DTC is selected by DTCER. At the end of a data transfer (or the last consecutive transfer in the case of chain transfer), the interrupt flag that became the activation source or the corresponding DTCER bit is cleared. The activation source flag, in the case of RXI1, for example, is the RDRF flag in SCI_1.

When an interrupt has been designated as a DTC activation source, the existing CPU mask level and interrupt controller priorities have no effect. If there is more than one activation source at the same time, the DTC operates in accordance with the default priorities. Figure 7.2 shows a block diagram of DTC activation source control. For details on the interrupt controller, see section 5, Interrupt Controller.

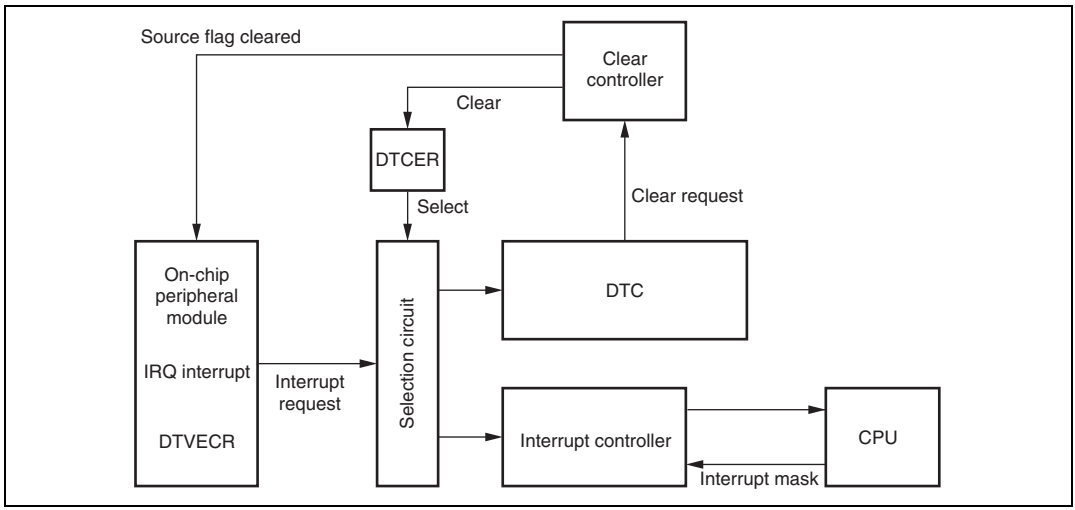


Figure 7.2 Block Diagram of DTC Activation Source Control

7.4 Location of Register Information and DTC Vector Table

Locate the register information in the on-chip RAM (addresses: H'(FF)EC00 to H'(FF)EFFF). Register information should be located at an address that is a multiple of four within the range. The method for locating the register information in address space is shown in figure 7.3. Locate MRA, SAR, MRB, DAR, CRA, and CRB, in that order, from the start address of the register information. In the case of chain transfer, register information should be located in consecutive areas as shown in figure 7.3, and the register information start address should be located at the vector address corresponding to the interrupt source in the DTC vector table. The DTC reads the start address of the register information from the vector table set for each activation source, and then reads the register information from that start address.

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

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

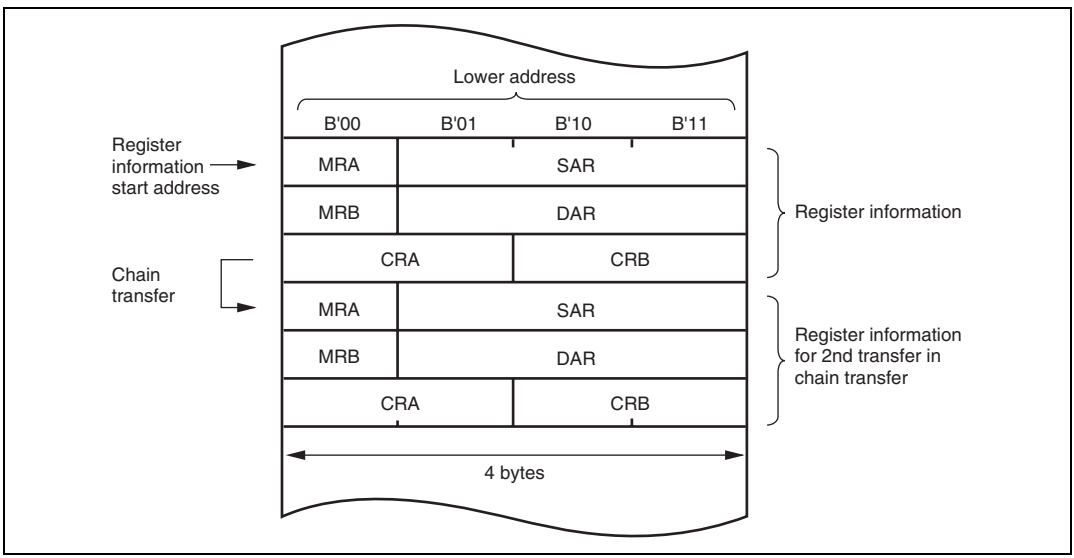


Figure 7.3 DTC Register Information Location in Address Space

Table 7.2 Interrupt Sources, DTC Vector Addresses, and Corresponding DTCEs

Activation Source Origin	Activation Source	Vector Number	DTC Vector Address	DTCE*	Priority
Software	Write to DTVECR	DTVECR	H'0400 + (vector number x 2)	—	High
External pins	IRQ0	16	H'0420	DTCEA7	
	IRQ1	17	H'0422	DTCEA6	
	IRQ2	18	H'0424	DTCEA5	
	IRQ3	19	H'0426	DTCEA4	
A/D converter	ADI	28	H'0438	DTCEA3	
TPU_0	TGIA	34	H'0444	DTCEE7	
	TGIB	35	H'0446	DTCEE6	
	TGIC	36	H'0448	DTCEE5	
	TGID	37	H'044A	DTCEE4	
TPU_1	TGIA	39	H'044E	DTCEB6	
	TGIB	40	H'0450	DTCEB5	
TPU_2	TGIA	43	H'0456	DTCEB4	
	TGIB	44	H'0458	DTCEB3	
FRT	ICIA	48	H'0460	DTCEA2	
	ICIB	49	H'0462	DTCEA1	
	OCIA	52	H'0468	DTCEA0	
	OCIB	53	H'046A	DTCEB7	
TMR_0	CMIA0	64	H'0480	DTCEB2	
	CMIB0	65	H'0482	DTCEB1	
TMR_1	CMIA1	68	H'0488	DTCEB0	
	CMIB1	69	H'048A	DTCEC7	
TMR_Y	CMIAY	72	H'0490	DTCEC6	
	CMIBY	73	H'0492	DTCEC5	
TMR_X	CMIAX	76	H'0498	DTCEC4	
	CMIBX	77	H'049A	DTCEC3	
SCI_1	RXI1	85	H'04AA	DTCEC0	
	TXI1	86	H'04AC	DTCED7	
SCI_2	RXI2	89	H'04B2	DTCED6	
	TXI2	90	H'04B4	DTCED5	
IIC_0	IICI0	92	H'04B8	DTCED4	
IIC_1	IICI1	94	H'04BC	DTCED3	Low

Activation Source Origin	Activation Source	Vector Number	DTC Vector Address	DTCE*	Priority
LPC	ERRI	108	H'04D8	DTCEE3	High
	IBFI1	109	H'04DA	DTCEE2	
	IBFI2	110	H'04DC	DTCEE1	
	IBFI3	111	H'04DE	DTCEE0	Low

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

7.5 Operation

The DTC stores register information in on-chip RAM. When activated, the DTC reads register information in on-chip RAM and transfers data. After the data transfer, the DTC writes updated register information back to on-chip RAM. The pre-storage of register information in memory makes it possible to transfer data over any required number of channels. The transfer mode can be specified as normal, repeat, or block transfer mode. Setting the CHNE bit in MRB to 1 makes it possible to perform a number of transfers with a single activation source (chain transfer).

The 24-bit SAR designates the DTC transfer source address, and the 24-bit DAR designates the transfer destination address. After each transfer, SAR and DAR are independently incremented, decremented, or left fixed depending on its register information.

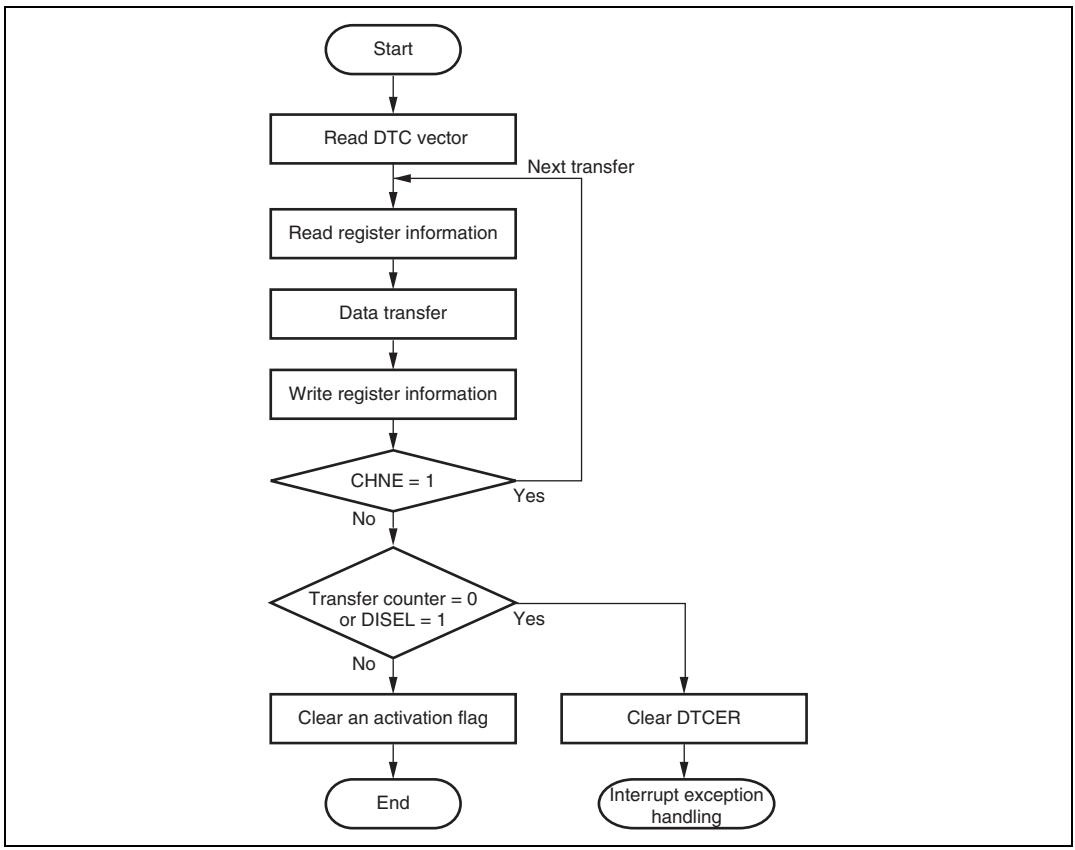


Figure 7.4 DTC Operation Flowchart

7.5.1 Normal Mode

In normal mode, one activation source transfers one byte or one word of data. Table 7.3 lists the register functions in normal mode. From 1 to 65,536 transfers can be specified. Once the specified number of transfers has been completed, a CPU interrupt can be requested.

Table 7.3 Register Functions in Normal Mode

Name	Abbreviation	Function
DTC source address register	SAR	Transfer source address
DTC destination address register	DAR	Transfer destination address
DTC transfer count register A	CRA	Transfer counter
DTC transfer count register B	CRB	Not used

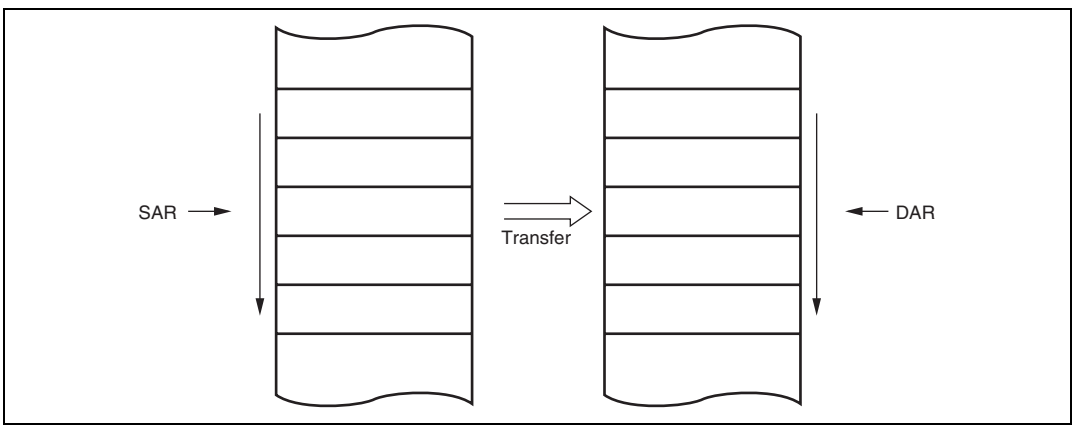


Figure 7.5 Memory Mapping in Normal Mode

7.5.2 Repeat Mode

In repeat mode, one activation source transfers one byte or one word of data. Table 7.4 lists the register functions in repeat mode. From 1 to 256 transfers can be specified. Once the specified number of transfers has been completed, the initial states of the transfer counter and the address register that is specified as the repeat area is restored, and transfer is repeated. In repeat mode, the transfer counter value does not reach H'00, and therefore CPU interrupts cannot be requested when the DISEL bit in MRB is cleared to 0.

Table 7.4 Register Functions in Repeat Mode

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

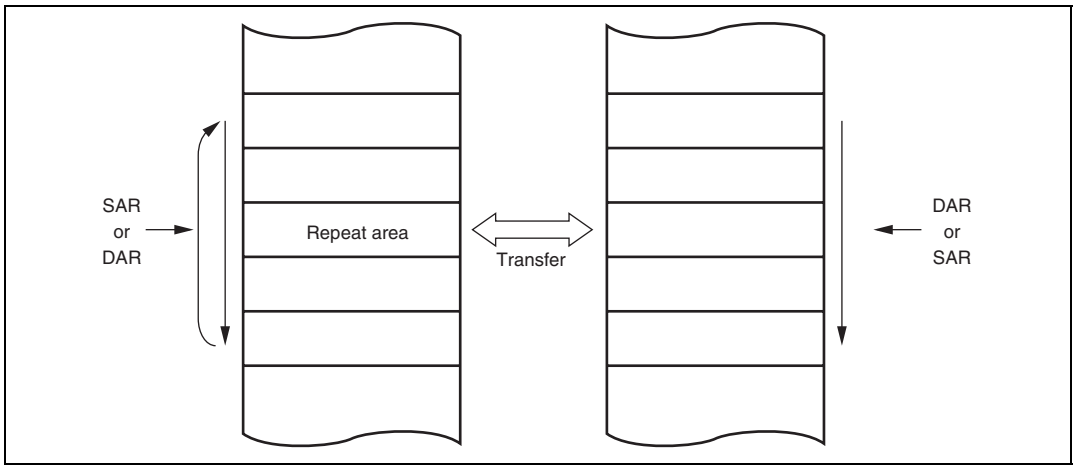


Figure 7.6 Memory Mapping in Repeat Mode

7.5.3 Block Transfer Mode

In block transfer mode, one activation source transfers one block of data. Either the transfer source or the transfer destination is designated as a block area. Table 7.5 lists the register functions in block transfer mode. The block size can be between 1 and 256. When the transfer of one block ends, the initial state of the block size counter and the address register that is specified as the block area is restored. The other address register is then incremented, decremented, or left fixed according to the register information. From 1 to 65,536 transfers can be specified. Once the specified number of transfers has been completed, a CPU interrupt is requested.

Table 7.5 Register Functions in Block Transfer Mode

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

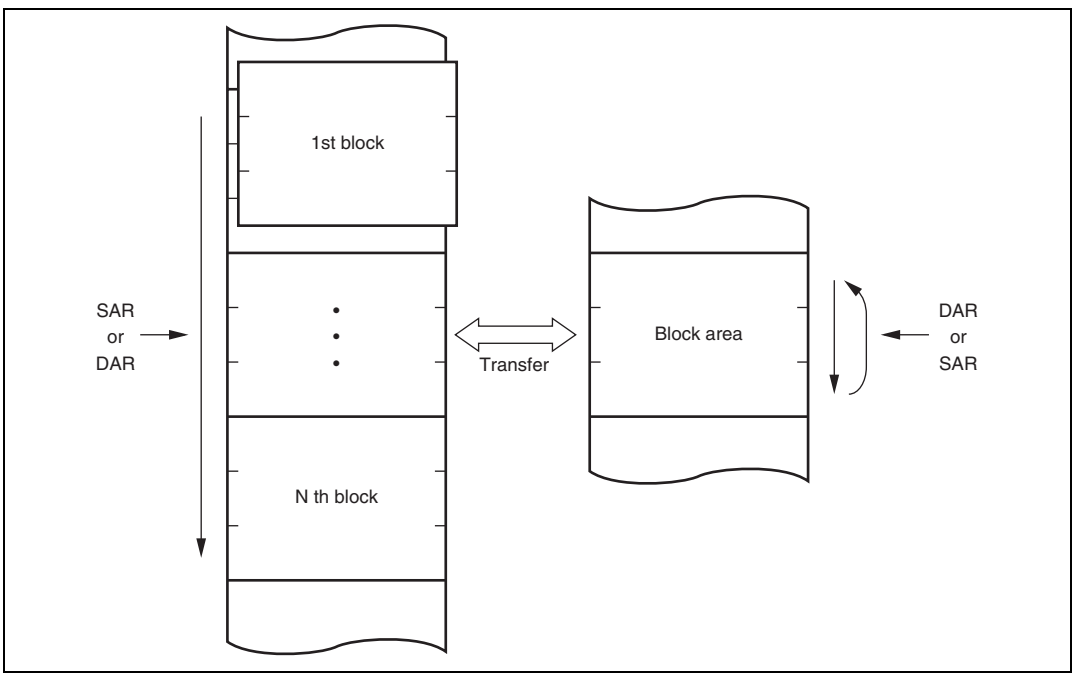


Figure 7.7 Memory Mapping in Block Transfer Mode

7.5.4 Chain Transfer

Setting the CHNE bit in MRB to 1 enables a number of data transfers to be performed consecutively in response to a single transfer request. SAR, DAR, CRA, CRB, MRA, and MRB, which define data transfers, can be set independently.

Figure 7.8 shows the overview of chain transfer operation. When activated, the DTC reads the register information start address stored at the DTC vector address, and then reads the first register information at that start address. After the data transfer, the CHNE bit will be tested. When it has been set to 1, DTC reads the next register information located in a consecutive area and performs the data transfer. These sequences are repeated until the CHNE bit is cleared to 0.

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

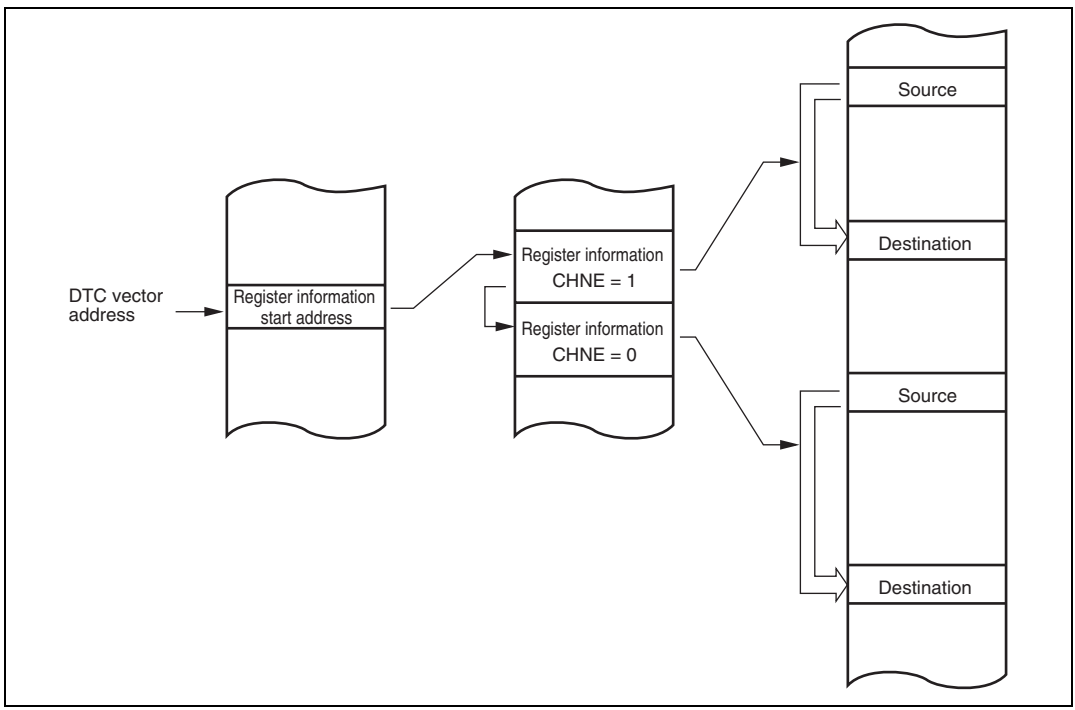


Figure 7.8 Chain Transfer Operation

7.5.5 Interrupt Sources

An interrupt request is issued to the CPU when the DTC has completed the specified number of data transfers, or a data transfer for which the DISEL bit was set to 1. In the case of interrupt activation, the interrupt set as the activation source is generated. These interrupts to the CPU are subject to CPU mask level and priority level control by the interrupt controller.

In the case of software activation, a software-activated data transfer end interrupt (SWDTEND) is generated.

When the DISEL bit is 1 and one data transfer has been completed, or the specified number of transfers have been completed, after data transfer ends, the SWDTE bit is held at 1 and an SWDTEND interrupt is generated. The interrupt handling routine will then clear the SWDTE bit to 0.

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

7.5.6 Operation Timing

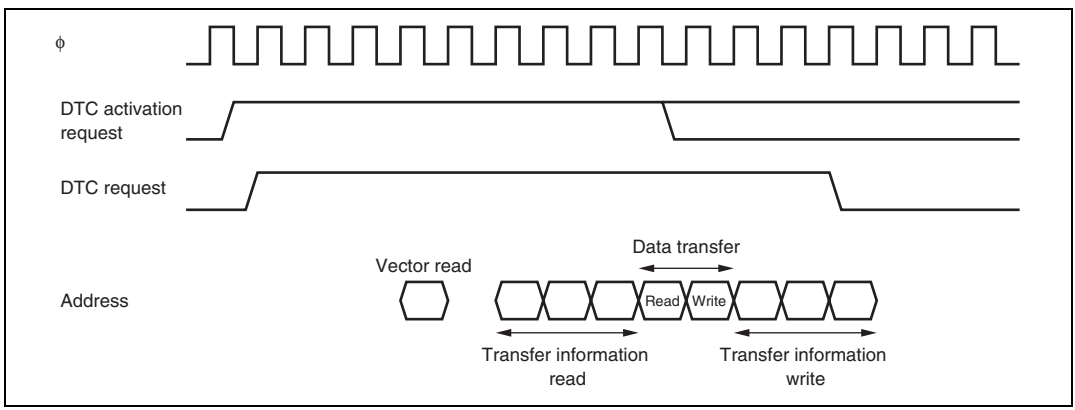


Figure 7.9 DTC Operation Timing (Example in Normal Mode or Repeat Mode)

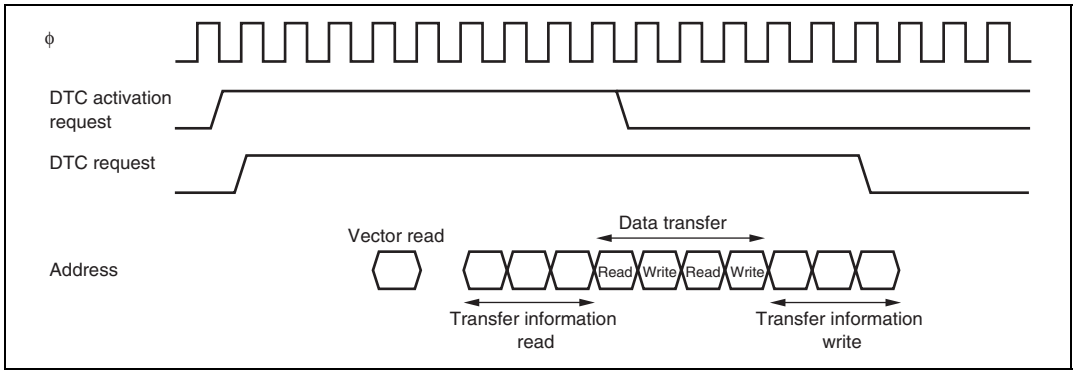


Figure 7.10 DTC Operation Timing (Example of Block Transfer Mode, with Block Size of 2)

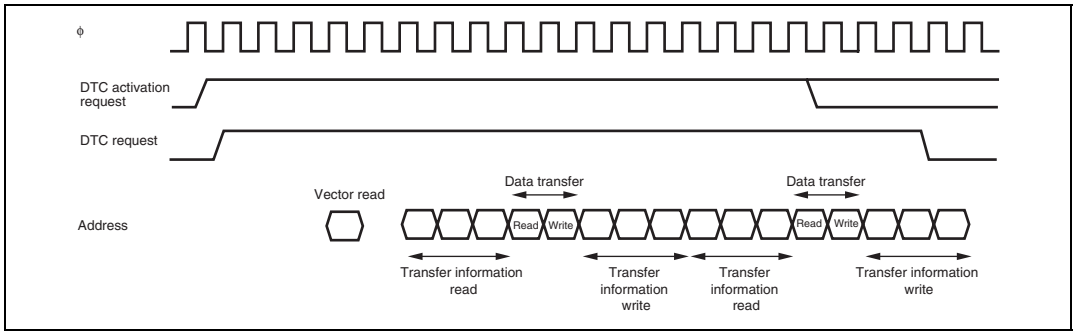


Figure 7.11 DTC Operation Timing (Example of Chain Transfer)

7.5.7 Number of DTC Execution States

Table 7.6 lists the execution status for a single DTC data transfer, and table 7.7 shows the number of states required for each execution status.

Table 7.6 DTC Execution Status

Mode	Vector Read I	Register Information Read/Write J	Data Read K	Data Write L	Internal Operations M
Normal	1	6	1	1	3
Repeat	1	6	1	1	3
Block transfer	1	6	N	N	3

[Legend]

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

Table 7.7 Number of States Required for Each Execution Status

Object to be Accessed		On-Chip RAM (H'(FF)EC00 to H'(FF)EFFF)	On-Chip RAM (On-chip RAM area other than H'(FF)EC00 to H'(FF)EFFF)	On-Chip ROM	On-Chip I/O Registers	
Bus width		32	16	16	8	16
Access states		1	1	1	2	2
Execution status	Vector read S_I	—	—	1	—	—
	Register information read/write S_J	1	—	—	—	—
	Byte data read S_K	1	1	1	2	2
	Word data read S_K	1	1	1	4	2
	Byte data write S_L	1	1	1	2	2
	Word data write S_L	1	1	1	4	2
	Internal operation S_M	1	1	1	1	1

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

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

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

7.6 Procedures for Using DTC

7.6.1 Activation by Interrupt

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

[1] Set the MRA, MRB, SAR, DAR, CRA, and CRB register information in on-chip RAM.

[2] Set the start address of the register information in the DTC vector address.

[3] Set the corresponding bit in DTCER to 1.

[4] Set the enable bits for the interrupt sources to be used as the activation sources to 1. The DTC is activated when an interrupt used as an activation source is generated.

[5] After one data transfer has been completed, or after the specified number of data transfers have been completed, the DTCE bit is cleared to 0 and a CPU interrupt is requested. If the DTC is to continue transferring data, set the DTCE bit to 1.

7.6.2 Activation by Software

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

[1] Set the MRA, MRB, SAR, DAR, CRA, and CRB register information in on-chip RAM.

[2] Set the start address of the register information in the DTC vector address.

[3] Check that the SWDTE bit is 0.

[4] Write 1 to the SWDTE bit and the vector number to DTVECR.

[5] Check the vector number written to DTVECR.

[6] After one data transfer has been completed, if the DISEL bit is 0 and a CPU interrupt is not requested, the SWDTE bit is cleared to 0. If the DTC is to continue transferring data, set the SWDTE bit to 1. When the DISEL bit is 1 or after the specified number of data transfers have been completed, the SWDTE bit is held at 1 and a CPU interrupt is requested.

7.7 Examples of Use of the DTC

7.7.1 Normal Mode

An example is shown in which the DTC is used to receive 128 bytes of data via the SCI.

[1] Set MRA to a fixed source address (SM1 = SM0 = 0), incrementing destination address (DM1 = 1, DM0 = 0), normal mode (MD1 = MD0 = 0), and byte size (Sz = 0). The DTS bit can have any value. Set MRB for one data transfer by one interrupt (CHNE = 0, DISEL = 0). Set the SCI, RDR address in SAR, the start address of the RAM area where the data will be received in DAR, and 128 (H'0080) in CRA. CRB can be set to any value.

[2] Set the start address of the register information at the DTC vector address.

[3] Set the corresponding bit in DTCER to 1.

[4] Set the SCI to the appropriate receive mode. Set the RIE bit in SCR to 1 to enable the reception complete (RXI) interrupt. Since the generation of a receive error during the SCI reception operation will disable subsequent reception, the CPU should be enabled to accept receive error interrupts.

[5] Each time the reception of one byte of data has been completed on the SCI, the RDRF flag in SSR is set to 1, an RXI interrupt is generated, and the DTC is activated. The receive data is transferred from RDR to RAM by the DTC. DAR is incremented and CRA is decremented. The RDRF flag is automatically cleared to 0.

[6] When CRA becomes 0 after 128 data transfers have been completed, the RDRF flag is held at 1, the DTCE bit is cleared to 0, and an RXI interrupt request is sent to the CPU. The interrupt handling routine will perform wrap-up processing.

7.7.2 Software Activation

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

[1] Set MRA to incrementing source address (SM1 = 1, SM0 = 0), incrementing destination address (DM1 = 1, DM0 = 0), block transfer mode (MD1 = 1, MD0 = 0), and byte size (Sz = 0). The DTS bit can have any value. Set MRB for one block transfer by one interrupt (CHNE = 0). Set the transfer source address (H'1000) in SAR, the transfer destination address (H'2000) in DAR, and 128 (H'8080) in CRA. Set 1 (H'0001) in CRB.

[2] Set the start address of the register information at the DTC vector address (H'04C0).

[3] Check that the SWDTE bit in DTVECR is 0. Check that there is currently no transfer activated by software.

[4] Write 1 to the SWDTE bit and the vector number (H'60) to DTVECR. The write data is H'E0.

[5] Read DTVECR again and check that it is set to the vector number (H'60). If it is not, this indicates that the write failed. This is presumably because an interrupt occurred between steps 3 and 4 and led to a different software activation. To activate this transfer, go back to step 3.

[6] If the write was successful, the DTC is activated and a block of 128 bytes of data is transferred.

[7] After the transfer, an SWDTEND interrupt occurs. The interrupt handling routine should clear the SWDTE bit to 0 and perform wrap-up processing.

7.8 Usage Notes

7.8.1 Module Stop Mode Setting

DTC operation can be enabled or disabled by the module stop control register (MSTPCR). In the initial state, DTC operation is enabled. Access to DTC registers is disabled when module stop mode is set. Note that when the DTC is being activated, module stop mode can not be specified. For details, see section 24, Power-Down Modes.

7.8.2 On-Chip RAM

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

7.8.3 DTCE Bit Setting

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

7.8.4 Setting Required on Entering Subactive Mode or Watch Mode

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

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

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

Section 8 I/O Ports

Table 8.1 is a summary of the port functions. The pins of each port also function as input/output pins of peripheral modules and interrupt input pins. Each input/output port includes a data direction register (DDR) that controls input/output and data registers (DR and ODR) that store output data. DDR, DR, and ODR are not provided for an input-only port.

Ports 1 to 3, 6, and B to F have built-in input pull-up MOSs. Port 1 to 3, C, and D can drive LEDs (with 5-mA current sink).

P52, P97, P86, P42, and ports A and G are NMOS push-pull output.

Table 8.1 Port Functions

Port	Description	Mode 2, Mode 3	I/O Status
Port 1	General I/O port	P17	Built-in input pull-up MOSs
		P16	LED drive capability
		P15	(sink current 5 mA)
		P14	
		P13	
		P12	
		P11	
		P10	
Port 2	General I/O port also functioning as PWM output	P27/PW15	Built-in input pull-up MOSs
		P26/PW14	LED drive capability
		P25/PW13	(sink current 5 mA)
		P24/PW12	
		P23/PW11	
		P22/PW10	
		P21/PW9	
		P20/PW8	

Port	Description	Mode 2, Mode 3	I/O Status
Port 3	General I/O port also functioning as LPC input/output	P37/SERIRQ	Built-in input pull-up MOSs
		P36/LCLK	LED drive capability
		P35/$\overline{\text{LRESET}}$	(sink current 5 mA)
		P34/$\overline{\text{LFRAME}}$	
		P33/LAD3	
		P32/LAD2	
		P31/LAD1	
		P30/LAD0	
Port 4	General I/O port also functioning as interrupt input, PWMX output, TMR_0, and TMR_1, SCI_2, IIC_1, and LPC inputs/outputs	P47/PWX1	
		P46/PWX0	
		P45/TMRI1	
		P44/TMO1	
		P43/TMCI1	
		P42/$\overline{\text{ExIRQ7}}$/TMRI0/SCK2/SDA1	
		P41/TMO0/RxD2/$\overline{\text{DCLKRUN}}$	
		P40/TMCI0/TxD2/DSERIRQ	
Port 5	General I/O port also functioning as interrupt input, IIC_0 input/output, TMR_Y output, and external sub-clock input	P52/$\overline{\text{ExIRQ6}}$/SCL0	
		P51/TMOY	
		P50/ExEXCL	
Port 6	General I/O port also functioning as interrupt input, TMR_Y, keyboard input, FRT, and TMR_X inputs/outputs	P67/$\overline{\text{IRQ7}}$/$\overline{\text{KIN7}}$/TMOX	Built-in input pull-up MOSs and noise canceller
		P66/$\overline{\text{IRQ6}}$/$\overline{\text{KIN6}}$/FTOB	
		P65/$\overline{\text{KIN5}}$/FTID	
		P64/$\overline{\text{KIN4}}$/FTIC	
		P63/$\overline{\text{KIN3}}$/FTIB	
		P62/$\overline{\text{KIN2}}$/FTIA/TMIY	
		P61/$\overline{\text{KIN1}}$/FTOA	
		P60/$\overline{\text{KIN0}}$/FTCI/TMIX	

Port	Description	Mode 2, Mode 3	I/O Status
Port 7	General input port also functioning as interrupt input and A/D converter analog input	P77/AN7 P76/AN6 P75/$\overline{\text{ExIRQ5}}$/AN5 P74/$\overline{\text{ExIRQ4}}$/AN4 P73/$\overline{\text{ExIRQ3}}$/AN3 P72/$\overline{\text{ExIRQ2}}$/AN2 P71/$\overline{\text{ExIRQ1}}$/AN1 P70/$\overline{\text{ExIRQ0}}$/AN0	
Port 8	General I/O port also functioning as interrupt input, SCI_1, IrDA interface, IIC_1, and LPC inputs/outputs	P86/$\overline{\text{IRQ5}}$/SCK1/SCL1 P85/$\overline{\text{IRQ4}}$/RxD1/IrRxD P84/$\overline{\text{IRQ3}}$/TxD1/IrTxD P83/$\overline{\text{LPCPD}}$ P82/$\overline{\text{CLKRUN}}$ P81/GA20 P80/$\overline{\text{PME}}$	
Port 9	General I/O port also functioning as A/D converter external trigger, external sub-clock, interrupt input, system clock output, and IIC_0 input/output	P97/$\overline{\text{IRQ15}}$/SDA0 P96/ϕ/EXCL P95/$\overline{\text{IRQ14}}$ P94/$\overline{\text{IRQ13}}$ P93/$\overline{\text{IRQ12}}$ P92/$\overline{\text{IRQ0}}$ P91/$\overline{\text{IRQ1}}$ P90/$\overline{\text{IRQ2}}$/$\overline{\text{ADTRG}}$	Built-in input pull-up MOSs (P95 to P90)
Port A	General I/O port also functioning as keyboard input and KBU input/output	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/$\overline{\text{KIN9}}$ PA0/$\overline{\text{KIN8}}$	

Port	Description	Mode 2, Mode 3	I/O Status
Port B	General I/O port also functioning as wake-up event input and LPC input/output	PB7/$\overline{\text{WUE7}}$/DLAD0 PB6/$\overline{\text{WUE6}}$/DLAD1 PB5/$\overline{\text{WUE5}}$/DLAD2 PB4/$\overline{\text{WUE4}}$/DLAD3 PB3/$\overline{\text{WUE3}}$/$\overline{\text{DLFRAME}}$ PB2/$\overline{\text{WUE2}}$ PB1/$\overline{\text{WUE1}}$/$\overline{\text{LSCI}}$ PB0/$\overline{\text{WUE0}}$/$\overline{\text{LSMI}}$	Built-in input pull-up MOSs
Port C	General I/O port also functioning wake-up event input and LPC input/output	PC7/$\overline{\text{WUE15}}$/DLDRQ PC6/$\overline{\text{WUE14}}$/LDRQ PC5/$\overline{\text{WUE13}}$ PC4/$\overline{\text{WUE12}}$ PC3/$\overline{\text{WUE11}}$ PC2/$\overline{\text{WUE10}}$ PC1/$\overline{\text{WUE9}}$ PC0/$\overline{\text{WUE8}}$	Built-in input pull-up MOSs and noise canceller LED drive capability (sink current 5 mA)
Port D	General I/O port also functioning as TPU input/output	PD7/TIOCB2/TCLKD PD6/TIOCA2 PD5/TIOCB1/TCLKC PD4/TIOCA1 PD3/TIOCD0/TCLKB PD2/TIOCC0/TCLKA PD1/TIOCB0 PD0/TIOCA0	Built-in input pull-up MOSs LED drive capability (sink current 5 mA)
Port E	General input port also functioning as LPC input and emulator input	PE4*/ETMS PE3*/ETDO PE2*/ETDI PE1*/ETCK PE0/$\overline{\text{LID3}}$	Built-in input pull-up MOSs

Port	Description	Mode 2, Mode 3	I/O Status
Port F	General I/O port also functioning as interrupt input, and PWM and TMR_X outputs	PF7/ExPW15	Built-in input pull-up MOSs
		PF6/ExPW14	
		PF5/ExPW13	
		PF4/ExPW12	
		PF3/$\overline{\text{IRQ11}}$/ExTMOX	
		PF2/$\overline{\text{IRQ10}}$	
		PF1/$\overline{\text{IRQ9}}$	
		PF0/$\overline{\text{IRQ8}}$	
Port G	General I/O port also interrupt input, TMR_0, TMR_1, TMR_X, and TMR_Y inputs, and IIC_0 and IIC_1 inputs/outputs	PG7/$\overline{\text{ExIRQ15}}$/ExSCLB	Built-in noise canceller
		PG6/$\overline{\text{ExIRQ14}}$/ExSDAB	
		PG5/$\overline{\text{ExIRQ13}}$/ExSCLA	
		PG4/$\overline{\text{ExIRQ12}}$/ExSDAA	
		PG3/$\overline{\text{ExIRQ11}}$/ExTMIY	
		PG2/$\overline{\text{ExIRQ10}}$/ExTMIX	
		PG1/$\overline{\text{ExIRQ9}}$/ExTMCI1	
		PG0/$\overline{\text{ExIRQ8}}$/ExTMCI0	

Note: * Not supported in the system development tool (emulator).

8.1 Port 1

Port 1 is an 8-bit I/O port. Port 1 has a built-in input pull-up MOS that can be controlled by software. Port 1 has the following registers.

- Port 1 data direction register (P1DDR)
- Port 1 data register (P1DR)
- Port 1 pull-up MOS control register (P1PCR)

8.1.1 Port 1 Data Direction Register (P1DDR)

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

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

8.1.2 Port 1 Data Register (P1DR)

P1DR stores output data for the port 1 pins.

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

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

P1PCR controls the on/off state of the input pull-up MOS for port 1 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P17PCR	0	R/W	When the pins are in input state, the corresponding input pull-up MOS is turned on when a P1PCR bit is set to 1.
6	P16PCR	0	R/W	
5	P15PCR	0	R/W	
4	P14PCR	0	R/W	
3	P13PCR	0	R/W	
2	P12PCR	0	R/W	
1	P11PCR	0	R/W	
0	P10PCR	0	R/W	

8.1.4 Pin Functions

- P17, P16, P15, P14, P13, P12, P11, P10

The function of port 1 pins is switched as shown below according to the P1nDDR bit.

P1nDDR	0	1
Pin function	P1n input pin	P1n output pin

Note: n = 7 to 0

8.1.5 Port 1 Input Pull-Up MOS

Port 1 has a built-in input pull-up MOS that can be controlled by software. Table 8.2 summarizes the input pull-up MOS states.

Table 8.2 Port 1 Input Pull-Up MOS States

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

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

8.2 Port 2

Port 2 is an 8-bit I/O port. Port 2 pins also functions as PWM output pins. Port 2 has a built-in input pull-up MOS that can be controlled by software. Port 2 has the following registers.

- Port 2 data direction register (P2DDR)
- Port 2 data register (P2DR)
- Port 2 pull-up MOS control register (P2PCR)

8.2.1 Port 2 Data Direction Register (P2DDR)

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

Bit	Bit Name	Initial Value	R/W	Description
7	P27DDR	0	W	The corresponding port 2 pins are output ports or PWM outputs when the P2DDR bits are set to 1, and input ports when cleared to 0.
6	P26DDR	0	W	
5	P25DDR	0	W	
4	P24DDR	0	W	
3	P23DDR	0	W	
2	P22DDR	0	W	
1	P21DDR	0	W	
0	P20DDR	0	W	

8.2.2 Port 2 Data Register (P2DR)

P2DR stores output data for the port 2 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P27DR	0	R/W	P2DR stores output data for the port 2 pins that are used as the general output port. If a port 2 read is performed while the P2DDR bits are set to 1, the P2DR values are read. If a port 2 read is performed while the P2DDR bits are cleared to 0, the pin states are read.
6	P26DR	0	R/W	
5	P25DR	0	R/W	
4	P24DR	0	R/W	
3	P23DR	0	R/W	
2	P22DR	0	R/W	
1	P21DR	0	R/W	
0	P20DR	0	R/W	

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

P2PCR controls the on/off state of the input pull-up MOS for port 2 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P27PCR	0	R/W	When the pins are in input state, the corresponding input pull-up MOS is turned on when a P2PCR bit is set to 1.
6	P26PCR	0	R/W	
5	P25PCR	0	R/W	
4	P24PCR	0	R/W	
3	P23PCR	0	R/W	
2	P22PCR	0	R/W	
1	P21PCR	0	R/W	
0	P20PCR	0	R/W	

8.2.4 Pin Functions

- P27/PW15, P26/PW14

  The function of port 2 pins is switched as shown below according to the combination of the PWMAS bit in PTCNT0, the OEm bit in PWOERB of PWM, and the P2nDDR bit.

PWMAS	0			1	
P2nDDR	0	1		0	1
OEm	—	0	1	—	
Pin function	P2n input pin	P2n output pin	PWm output pin	P2n input pin	P2n output pin

Note: n = 7 to 6
m = 15 to 14

- P25/PW13, P24/PW12

  The function of port 2 pins is switched as shown below according to the combination of the PWMBS bit in PTCNT0, the OEm bit in PWOERB of PWM, and the P2nDDR bit.

PWMBS	0			1	
P2nDDR	0	1		0	1
OEm	—	0	1	—	
Pin function	P2n input pin	P2n output pin	PWm output pin	P2n input pin	P2n output pin

Note: n = 5 to 4
m = 13 to 12

- P23/PW11, P22/PW10, P21/PW9, P20/PW8

  The function of port 2 pins is switched as shown below according to the combination of the OEm bit in PWOERA of PWM and the P2nDDR bit.

P2nDDR	0	1	
OEm	—	0	1
Pin function	P2n input pin	P2n output pin	PWm output pin

Note: n = 3 to 0
m = 11 to 8

8.2.5 Port 2 Input Pull-Up MOS

Port 2 has a built-in input pull-up MOS that can be controlled by software. Table 8.3 summarizes the input pull-up MOS states.

Table 8.3 Port 2 Input Pull-Up MOS States

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

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

8.3 Port 3

Port 3 is an 8-bit I/O port. Port 3 pins also function as LPC input/output pins. Port 3 has a built-in input pull-up MOS that can be controlled by software. Port 3 has the following registers.

- Port 3 data direction register (P3DDR)
- Port 3 data register (P3DR)
- Port 3 pull-up MOS control register (P3PCR)

8.3.1 Port 3 Data Direction Register (P3DDR)

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

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

8.3.2 Port 3 Data Register (P3DR)

P3DR stores output data for the port 3 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P37DR	0	R/W	P3DR stores output data for the port 3 pins that are used as the general output port. If a port 3 read is performed while the P3DDR bits are set to 1, the P3DR values are read. If a port 3 read is performed while the P3DDR bits are cleared to 0, the pin states are read.
6	P36DR	0	R/W	
5	P35DR	0	R/W	
4	P34DR	0	R/W	
3	P33DR	0	R/W	
2	P32DR	0	R/W	
1	P31DR	0	R/W	
0	P30DR	0	R/W	

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

P3PCR controls the on/off state of the input pull-up MOS for port 3 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P37PCR	0	R/W	When the pins are in input state, the corresponding input pull-up MOS is turned on when a P3PCR bit is set to 1.
6	P36PCR	0	R/W	
5	P35PCR	0	R/W	
4	P34PCR	0	R/W	
3	P33PCR	0	R/W	
2	P32PCR	0	R/W	
1	P31PCR	0	R/W	
0	P30PCR	0	R/W	

8.3.4 Pin Functions

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

  The function of port 3 pins is switched as shown below according to the combination of the LPC4E bit in HICR4 of LPC, LPC3E to LPC1E bits in HICR0, LMCE bit in LMCCR1, and the P3nDDR bit. LPCENABLE in the following table is expressed by the following logical expressions.

  LPCENABLE = 1 : LPC4E + LPC3E + LPC2E + LPC1E + LMCE

LPCENABLE	0		1
P3nDDR	0	1	—
Pin function	P3n input pins	P3n output pins	LPC input/output pin

Note: n = 7 to 0

8.3.5 Port 3 Input Pull-Up MOS

Port 3 has a built-in input pull-up MOS that can be controlled by software. Table 8.4 summarizes the input pull-up MOS states.

Table 8.4 Port 3 Input Pull-Up MOS States

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

[Legend]

Off: Always off.

On/Off: On when P3DDR = 0 and P3PCR = 1; otherwise off.

8.4 Port 4

Port 4 is an 8-bit I/O port. Port 4 pins also function as interrupt input, PWMX output, TMR_0, TMR_1, SCI_2, IIC_1, and LPC input/output pins. The output format for P42 and SCK2 is NMOS push-pull output. The output format for SDA1 is NMOS open-drain output. Port 4 has the following registers.

- Port 4 data direction register (P4DDR)
- Port 4 data register (P4DR)

8.4.1 Port 4 Data Direction Register (P4DDR)

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

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

8.4.2 Port 4 Data Register (P4DR)

P4DR stores output data for the port 4 pins.

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

8.4.3 Pin Functions

- P47/PWX1

  The pin function is switched as shown below according to the combination of the OEB bit in DACR of PWMX, and P47DDR bit.

OEB	0		1
P47DDR	0	1	—
Pin function	P47 input pin	P47 output pin	PWX1 output pin

- P46/PWMX0

  The pin function is switched as shown below according to the combination of the OEA bit in DACR of PWMX, and the P46DDR bit.

OEA	0		1
P46DDR	0	1	—
Pin function	P46 input pin	P46 output pin	PWX0 output pin

- P45/TMRI1

  The pin function is switched as shown below according to the P45DDR bit.

  When the CCLR1 and CCLR0 bits in TCR of TMR_1 are set to 1, this pin is used as the TMRI1 input pin.

P45DDR	0	1
Pin function	P45 input pin	P45 output pin
	TMRI1 input pin	

- P44/TMO1

  The pin function is switched as shown below according to the combination of the OS3 to OS0 bits in TCR of TMR_1 and the P44DDR bit.

OS3 to OS0	All 0		One bit is set as 1
P44DDR	0	1	—
Pin function	P44 input pin	P44 output pin	TMO1 output pin

- P43/TMCI1

  The pin function is switched as shown below according to the P43DDR bit. When the external clock is selected by the CKS2 to CKS0 bits in TCR of TMR_1, this pin can be used as the TMCII input pin.

P43DDR	0	1
Pin function	P43 input pin	P43 output pin
	TMCI1 input pin	

- P42/$\overline{\text{ExIRQ7}}$/TMRI0/SCK2/SDA1

  The pin function is switched as shown below according to the combination of the SDA1AS and SDA1BS bits in PTCNT1, ICE bit in ICCR of IIC_1, CKE1 and CKE0 bits in SCR of SCI_2, C/$\overline{\text{A}}$ bit in SMR, and the P42DDR bit. When the CCLR1 and CCLR0 bits in TCR of TMR_0 are set to 1, this pin is used as the TMRI0 input pin. When the ISS7 bit in ISSR and the IRQ7E bit in IER of the interrupt controller are set to 1, this pin can be used as the $\overline{\text{ExIRQ7}}$ interrupt input pin. IICENABLE in the following table is expressed by the following logical expressions.

  IICENABLE = 1 : ICE • $\overline{\text{SDA1AS}}$ • $\overline{\text{SDA1BS}}$

IICENABLE	0					1
CKE1	0				1	0
C/$\overline{A}$	0			1	—	0
CKE0	0		1	—	—	0
P42DDR	0	1	—	—	—	—
Pin function	P42 input pin	P42 output pin	SCK2 output pin	SCK2 output pin	SCK2 input pin	SDA1 input/output pin
	$\overline{ExIRQ7}$ input pin/TMRI0 input pin					

Note: To use this pin as the SDA1 input/output pin, clear the SDA1AS and SDA1BS bits in PTCNT1, CKE1 and CKE0 bits in SCR of SCI_2, and C/$\overline{A}$ bit in SMR 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 or SCK2 output pin, the output format is NMOS push-pull output.

- P41/TMO0/RxD2/$\overline{DCLKRUN}$

  The pin function is switched as shown below according to the combination of the OS3 to OS0 bits in TCSR of TMR_0, RE bit in SCR of SCI_2, LPCS bit in PTCNT2 and the P41DDR bits.

LPCS	0				1
OS3 to OS0	All 0			One bit is set as 1	—
RE	0		1	0	—
P41DDR	0	1	—	—	—
Pin function	P41 input pin	P41 output pin	RxD2 input pin	TMO0 output pin	$\overline{DCLKRUN}$ input/output pin

Note: To use this pin as the TMO0 output pin, clear the RE bit in SCR of SCI_2 to 0.

- P40/TMCI0/TxD2/DSERIRQ

  The pin function is switched as shown below according to the combination of the TE bit in SCR of SCI_2, LPCS bit in PTCNT2, and the P40DDR bits. When the TMI0S bit in PTCNT0 is cleared to 0 and the external clock is selected by the CKS2 to CKS0 bits in TCR of TMR_0, this bit is used as the TMCI0 input pin.

LPCS	0			1
TE	0		1	—
P40DDR	0	1	—	—
Pin function	P40 input pin	P40 output pin	TxD2 output pin	DSERIRQ input/output pin
	TMCI0 input pin			

8.5 Port 5

Port 5 is a 3-bit I/O port. Port 5 pins also function as interrupt input pins, IIC_0 input/output pin, TMR_Y output pin, and the external sub-clock input pin. The output format for P52 is NMOS push-pull output. Port 5 has the following registers.

- Port 5 data direction register (P5DDR)
- Port 5 data register (P5DR)

8.5.1 Port 5 Data Direction Register (P5DDR)

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

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	Undefined	—	Reserved These bits cannot be modified.
2	P52DDR	0	W	If port 5 pins are specified for use as the general I/O port, the corresponding port 5 pins are output ports when the P5DDR bits are set to 1, and input ports when cleared to 0.
1	P51DDR	0	W	
0	P50DDR	0	W	

8.5.2 Port 5 Data Register (P5DR)

P5DR stores output data for the port 5 pins.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
2	P52DR	0	R/W	P5DR stores output data for the port 5 pins that are used as the general output port. If a port 5 read is performed while the P5DDR bits are set to 1, the P5DR values are read. If a port 5 read is performed while the P5DDR bits are cleared to 0, the pin states are read.
1	P51DR	0	R/W	
0	P50DR	0	R/W	

8.5.3 Pin Functions

- P52/$\overline{\text{ExIRQ6}}$/SCL0

  The pin function is switched as shown below according to the combination of the SCL0AS and SCL0BS bits in PTCNT1, ICE bit in ICCR of IIC_1, and the P52DDR bit.

  When the IRQ6E bit in IER of the interrupt controller is set to 1, this pin can be used as the $\overline{\text{ExIRQ6}}$ interrupt input pin. IICENABLE in the following table is expressed by the following logical expressions.

  IICENABLE = 1 : ICE • $\overline{\text{SCLOAS}}$ • $\overline{\text{SCLOBS}}$

IICENABLE	0		1
P52DDR	0	1	—
Pin function	P52 input pin	P52 output pin	SCL0 input/output pin
	$\overline{\text{ExIRQ6}}$ input pin		

Note: To use this pin as the SCL0 input/output pin, clear the SCL0AS and SCL0BS 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 output.

- P51/TMOY

  The pin function is switched as shown below according to the combination of the OS3 to OS0 bits in TCSR of TMR_Y and the P51DDR bit.

OS3 to OS0	All 0		One bit is set as 1
P51DDR	0	1	—
Pin function	P51 input pin	P51 output pin	TMOY output pin

- P50/ExEXCL

  The pin function is switched as shown below according to the combination of the EXCLS bit in PTCNT0, EXCLE bit in LPWRCR, and the P50DDR bit.

  To use this pin as the ExEXCL input pin, clear the P50DDR bit to 0.

EXCLS	0		1		
P50DDR	0	1	0		1
EXCLE	—		0	1	0
Pin function	P50 input pin	P50 output pin	P50 input pin	ExEXCL input pin	P50 output pin

8.6 Port 6

Port 6 is an 8-bit I/O port. Port 6 pins also function as the interrupt input pin, TMR_Y, keyboard and noise cancel input pins, FRT, and TMR_X input/output pin. Port 6 can change the input level for four levels. Port 6 has the following registers.

- Port 6 data direction register (P6DDR)
- Port 6 data register (P6DR)
- Pull-up MOS control register (KMPCR)
- System control register 2 (SYSCR2)
- Noise canceller enable register (P6NCE)
- Noise canceller decision control register (P6NCMC)
- Noise cancel cycle setting register (P6NCCS)

8.6.1 Port 6 Data Direction Register (P6DDR)

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

Bit	Bit Name	Initial Value	R/W	Description
7	P67DDR	0	W	The corresponding port 6 pins are output ports when P6DDR bits are set to 1, and input ports when cleared to 0.
6	P66DDR	0	W	
5	P65DDR	0	W	
4	P64DDR	0	W	
3	P63DDR	0	W	
2	P62DDR	0	W	
1	P61DDR	0	W	
0	P60DDR	0	W	

8.6.2 Port 6 Data Register (P6DR)

P6DR stores output data for the port 6 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P67DR	0	R/W	P6DR stores output data for the port 6 pins that are used as the general output port. If a port 6 read is performed while the P6DDR bits are set to 1, the P6DR values are read. If a port 6 read is performed while the P6DDR bits are cleared to 0, the pin states are read.
6	P66DR	0	R/W	
5	P65DR	0	R/W	
4	P64DR	0	R/W	
3	P63DR	0	R/W	
2	P62DR	0	R/W	
1	P61DR	0	R/W	
0	P60DR	0	R/W	

8.6.3 Pull-Up MOS Control Register (KMPCR)

KMPCR controls the on/off state of the input pull-up MOS for port 6 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	KM7PCR	0	R/W	When the pins are in input state, the corresponding input pull-up MOS is turned on when a KMPCR bit is set to 1.
6	KM6PCR	0	R/W	
5	KM5PCR	0	R/W	
4	KM4PCR	0	R/W	
3	KM3PCR	0	R/W	
2	KM2PCR	0	R/W	
1	KM1PCR	0	R/W	
0	KM0PCR	0	R/W	

8.6.4 Noise Canceller Enable Register (P6NCE)

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

Bit	Bit Name	Initial Value	R/W	Description
7	P67NCE	0	R/W	Noise cancel circuit is enabled when P6NCE bit is set to 1, and the pin state is fetched in the P6DR in the sampling cycle set by the P6NCCS.
6	P66NCE	0	R/W	
5	P65NCE	0	R/W	
4	P64NCE	0	R/W	
3	P63NCE	0	R/W	
2	P62NCE	0	R/W	
1	P61NCE	0	R/W	
0	P60NCE	0	R/W	

8.6.5 Noise Canceller Mode Control Register (P6NCMC)

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

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

8.6.6 Noise Cancel Cycle Setting Register (P6NCCS)

P6NCCS controls the sampling cycles of the noise canceller.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	Undefined	R/W	Reserved The read data is undefined. The write value should always be 0.
2	P6NCCK2	0	R/W	These bits set the sampling cycles of the noise canceller. 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	P6NCCK1	0	R/W	
0	P6NCCK0	0	R/W	

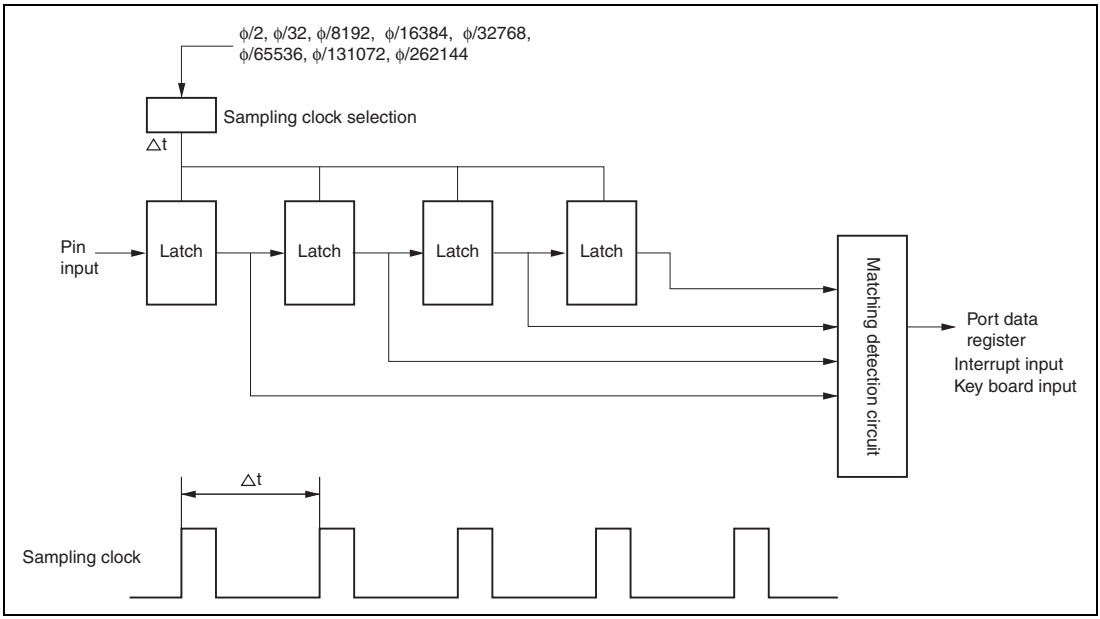


Figure 8.1 Noise Cancel Circuit

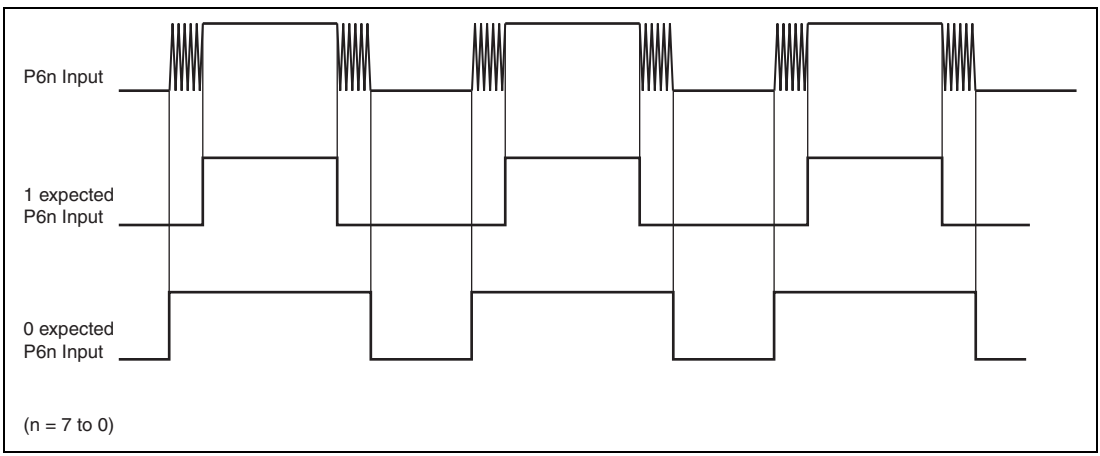


Figure 8.2 Noise Cancel Operation

8.6.7 System Control Register 2 (SYSCR2)

SYSCR2 controls the port 6 input level selection and the current specifications for the port 6 input pull-up MOSs.

Bit	Bit Name	Initial Value	R/W	Description
7	KWUL1	0	R/W	Key Wakeup Level 1, 0
6	KWUL0	0	R/W	Select the port 6 input level. 00: Standard input level is selected 01: Input level 1 is selected 10: Input level 2 is selected 11: Input level 3 is selected
5	P6PUE	0	R/W	Port 6 Input Pull-Up Extra Selects the current specification for the input pull-up MOS. 0: Standard current specification is selected 1: Current-limit specification is selected
4 to 0	—	All 0	R/W	Reserved The initial value should not be changed.

8.6.8 Pin Functions

- P67/$\overline{\text{IRQ7}}$/$\overline{\text{KIN7}}$/TMOX

  The function of port 6 pins is switched as shown below according to the combination of the TMOXS bit in PTCNT0, OS3 to OS0 bits in TCSR of TMR_X, and the P67DDR bit.

  When the KMIMR7 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 $\overline{\text{IRQ7}}$ interrupt input pin.

TMOXS	0			1	
OS3 to OS0	All 0		One bit is set as 1	—	
P67DDR	0	1	—	0	1
Pin function	P67 input pin	P67 output pin	TMOX output pin	P67 input pin	P67 output pin
	$\overline{\text{IRQ7}}$ input pin/$\overline{\text{KIN7}}$ input pin				

- P66/$\overline{\text{IRQ6}}$/$\overline{\text{KIN6}}$/FTOB

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

  When the KMIMR6 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 SYSCR 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 $\overline{\text{IRQ6}}$ interrupt input pin.

OEB	0		1
P66DDR	0	1	—
Pin function	P66 input pin	P66 output pin	FTOB output pin
	$\overline{\text{IRQ6}}$ input pin/$\overline{\text{KIN6}}$ input pin		

- P65/$\overline{\text{KIN5}}$/FTID

  The function of port 6 pins is switched as shown below according to the P65DDR bit.

  When the ICIDE bit in TIER of FRT is set to 1, this pin can be used as the FTID input pin. When the KMIMR5 bit in KMIMR of the interrupt controller is cleared to 0, this pin can be used as the $\overline{\text{KIN5}}$ input pin.

P65DDR	0	1
Pin function	P65 input pin	P65 output pin
	$\overline{\text{KIN5}}$ input pin/FTID input pin	

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

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

  When the ICICE bit in TIER of FRT is set to 1, this pin can be used as the FTIC input pin. When the KMIMR4 bit in KMIMR of the interrupt controller is cleared to 0, this pin can be used as the $\overline{\text{KIN4}}$ input pin.

P64DDR	0	1
Pin function	P64 input pin	P64 output pin
	$\overline{\text{KIN4}}$ input pin/FTIC input pin	

- P63/$\overline{\text{KIN3}}$/FTIB

  The function of port 6 pins is switched as shown below according to the P63DDR bit.

  When the ICIBE bit in TIER of FRT is set to 1, this pin can be used as the FTIB input pin.

  When the KMIMR3 bit in KMIMR of the interrupt controller is cleared to 0, this pin can be used as the $\overline{\text{KIN3}}$ input pin.

P63DDR	0	1
Pin function	P63 input pin	P63 output pin
	$\overline{\text{KIN3}}$ input pin/FTIB input pin	

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

  The function of port 6 pins is switched as shown below according to the P62DDR bit. When the ICIAE bit in TIER of FRT is set to 1, this pin can be used as the FTIA input pin. When the TMIYS bit in PTCNT0 is cleared to 0 and the CCLR1 and CCLR0 bits in TCR of TMR_Y are both set to 1, this pin is used as the TMIY (TMRIY) input pin. When the KMIMR2 bit in KMIMR of the interrupt controller is cleared to 0, this pin can be used as the $\overline{\text{KIN2}}$ input pin.

P62DDR	0	1
Pin function	P62 input pin	P62 output pin
	$\overline{\text{KIN2}}$ input pin/FTIA input pin/TMIY input pin	

- P61/$\overline{\text{KIN1}}$/FTOA

  The function of port 6 pins is switched as shown below according to the combination of the OEA bit in TOCR of FRT and the P61DDR bit.

  When the KMIMR1 bit in KMIMR of the interrupt controller is cleared to 0, this pin can be used as the $\overline{\text{KIN1}}$ input pin.

OEA	0		1
P61DDR	0	1	—
Pin function	P61 input pin	P61 output pin	FTOA output pin
	$\overline{\text{KIN1}}$ input pin		

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

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

  When the CKS1 and CKS0 bits in TCR of FRT are both set to 1, this pin can be used as the FTCI input pin. When the TMIXS bit in PTCNT0 is cleared to 0 and the CCLR1 and CCLR0 bits in TCR of TMR_X are both set to 1, this pin is used as the TMIX(TMRIX) input pin. When the KMIMR0 bit in KMIMR of the interrupt controller is cleared to 0, this pin can be used as the $\overline{\text{KIN0}}$ input pin.

P60DDR	0	1
Pin function	P60 input pin	P60 output pin
	$\overline{\text{KIN0}}$ input pin/FTCI input pin/TMIX input pin	

8.6.9 Port 6 Input Pull-Up MOS

Port 6 has a built-in input pull-up MOS that can be controlled by software. Port 6 can selects the current specification for the input pull-up MOSs by the P6PUE bit. When the pin functions as an output pin of the built-in peripheral function, the input pull-up MOS is always off. Table 8.5 summarizes the input pull-up MOS states.

Table 8.5 Port 6 Input Pull-Up MOS States

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

[Legend]

Off: Always off.

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

8.7 Port 7

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

- Port 7 input data register (P7PIN)

8.7.1 Port 7 Input Data Register (P7PIN)

P7PIN indicates the pin states.

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

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

8.7.2 Pin Functions

- P77/AN7, P76/AN6

Pin function	P7n input pin/ANn input pin

Note: n = 7, 6

- P75/$\overline{\text{ExIRQ5}}$/AN5, P74/$\overline{\text{ExIRQ4}}$/AN4, P73/$\overline{\text{ExIRQ3}}$/AN3, P72/$\overline{\text{ExIRQ2}}$/AN2, P71/$\overline{\text{ExIRQ1}}$/AN1, P70/$\overline{\text{ExIRQ0}}$/AN0

  When the ISS0n bit in ISSR and the IRQnE bit in IER of the interrupt controller are set to 1, this pin can be used as the $\overline{\text{ExIRQn}}$ interrupt input pin.

Pin function	P7n input pin/$\overline{\text{ExIRQn}}$ input pin/ANn input pin

Note: n = 5 to 0

When the interrupt input pin is set, do not use as the AN input pin.

8.8 Port 8

Port 8 is a 7-bit I/O port. Port 8 pins also function as the interrupt input pins, SCI_1 and IIC_1 input/output pins, and LPC input/output pin. The output format for P86 and SCK1 is NMOS push-pull output. The output format for SCL1 is NMOS open-drain output.

- Port 8 data direction register (P8DDR)
- Port 8 data register (P8DR)

8.8.1 Port 8 Data Direction Register (P8DDR)

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

Bit	Bit Name	Initial Value	R/W	Description
7	—	Undefined	—	Reserved This bit cannot be modified.
6	P86DDR	0	W	If port 8 pins are specified for use as the general I/O port, the corresponding port 8 pins are output ports when the P8DDR bits are set to 1, and input ports when cleared to 0.
5	P85DDR	0	W	
4	P84DDR	0	W	
3	P83DDR	0	W	
2	P82DDR	0	W	
1	P81DDR	0	W	
0	P80DDR	0	W	

8.8.2 Port 8 Data Register (P8DR)

P8DR stores output data for the port 8 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	—	1	—	Reserved The initial value should not be changed.
6	P86DR	0	R/W	P8DR stores output data for the port 8 pins that are used as the general output port. If a port 8 read is performed while the P8DDR bits are set to 1, the P8DR values are read. If a port 8 read is performed while the P8DDR bits are cleared to 0, the pin states are read.
5	P85DR	0	R/W	
4	P84DR	0	R/W	
3	P83DR	0	R/W	
2	P82DR	0	R/W	
1	P81DR	0	R/W	
0	P80DR	0	R/W	

8.8.3 Pin Functions

- P86/$\overline{\text{IRQ5}}$/SCK1/SCL1

  The pin function is switched as shown below according to the combination of the SCL1AS and SCL1BS bits in PTCNT1, ICE bit in ICCR of IIC_1, C/$\overline{\text{A}}$ bit in SMR of SCI_1, CKE0 and CKE1 bits in SCR, 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. IICENABLE in the following table is expressed by the following logical expressions.

  IICENABLE = 1 : ICE • $\overline{\text{SCL1AS}}$ • $\overline{\text{SCL1BS}}$

IICENABLE	0					1
CKE1	0				1	0
C/$\overline{\text{A}}$	0			1	—	0
CKE0	0		1	—	—	0
P86DDR	0	1	—	—	—	—
Pin function	P86 input pin	P86 output pin	SCK1 output pin	SCK1 input pin	SCL1 input/output pin	
	$\overline{\text{IRQ5}}$ input pin					

Note: To use this pin as the SCL1 input/output pin, clear the SCL1AS and SCL1BS bits in PTCNT1, CKE1, CKE0 bits in SCR of SCI_1 and C/$\overline{\text{A}}$ bit in SMR 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 output.

- P85/$\overline{\text{IRQ4}}$/RxD1/IrRxD

  The pin function is switched as shown below according to the combination of the RE bit in SCR of SCI_1 and the P85DDR bit. When the ISS4 bit in ISSR is cleared to 0 and the IRQ4E bit in IER of the interrupt controller is set to 1, this pin can be used as the $\overline{\text{IRQ4}}$ input pin.

RE	0		1
P85DDR	0	1	—
Pin function	P85 input pin	P85 output pin	RxD1 input pin/IrRxD input pin
	$\overline{\text{IRQ4}}$ input pin		

- P84/$\overline{\text{IRQ3}}$/TxD1/IrTxD

  The pin function is switched as shown below according to the combination of the TE bit in SCR of SCI_1 and the P84DDR bit. When the ISS3 bit in ISSR is cleared to 0 and the IRQ3E bit in IER of the interrupt controller is set to 1, this pin can be used as the $\overline{\text{IRQ3}}$ input pin.

TE	0		1
P84DDR	0	1	—
Pin function	P84 input pin	P84 output pin	TxD1 output pin/IrTxD output pin
	$\overline{\text{IRQ3}}$ input pin		

- P83/$\overline{\text{LPCPD}}$

  The pin function is switched as shown below according to the combination of the LPC4E bit in HICR4 of LPC, LPC3E to LPC1E bits in HICR0, LMCE bit in LMCCR1, and the P83DDR bit. LPCENABLE in the following table is expressed by the following logical expressions.

  LPCENABLE : LPC4E + LPC3E + LPC2E + LPC1E + LMCE

LPCENABLE	0		1
P83DDR	0	1	—
Pin function	P83 input pin	P83 output pin	$\overline{\text{LPCPD}}$ input pin

- P82/$\overline{\text{CLKRUN}}$

  The pin function is switched as shown below according to the combination of the LPC4E bit in HICR4 of LPC, LPC3E to LPC1E bits in HICR0, LMCE bit in LMCCR1, and the P82DDR bit. LPCENABLE in the following table is expressed by the following logical expressions.

  LPCENABLE : LPC4E + LPC3E + LPC2E + LPC1E + LMCE

LPCENABLE	0		1
P82DDR	0	1	—
Pin function	P82 input pin	P82 output pin	$\overline{\text{CLKRUN}}$ input/output pin

- P81/GA20

  The pin function is switched as shown below according to the combination of the FGA20E bit in HICR0 of LPC and the P81DDR bit.

FGA20E	0		1
P81DDR	0	1	—
Pin function	P81 input pin	P81 output pin	GA20 output pin

- P80/$\overline{\text{PME}}$

  The pin function is switched as shown below according to the combination of the PMEE bit in HICR0 of LPC and the P80DDR bit.

PMEE	0		1
P80DDR	0	1	—
Pin function	P80 input pin	P80 output pin	$\overline{\text{PME}}$ output pin

8.9 Port 9

Port 9 is an 8-bit I/O port. Port 9 pins also function as the interrupt input pins, A/D converter inputs, sub-clock input pin, IIC_0 I/O pin, and the system clock output pin (φ). The output format for P97 is NMOS push-pull output. The output format for SDA0 is NMOS open-drain output, and direct bus drive is possible. Port 9 has the following registers.

- Port 9 data direction register (P9DDR)
- Port 9 data register (P9DR)
- Port 9 pull-up MOS control register (P9PCR)

8.9.1 Port 9 Data Direction Register (P9DDR)

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

Bit	Bit Name	Initial Value	R/W	Description
7	P97DDR	0	W	The corresponding port 9 pins are output ports when the P9DDR bits are set to 1, and input ports when cleared to 0.
6	P96DDR	0	W	When this bit is set to 1, the corresponding port 96 pin is the system clock output pin (φ).
5	P95DDR	0	W	The corresponding port 9 pins are output ports when the P9DDR bits are set to 1, and input ports when cleared to 0.
4	P94DDR	0	W	
3	P93DDR	0	W	
2	P92DDR	0	W	
1	P91DDR	0	W	
0	P90DDR	0	W	

8.9.2 Port 9 Data Register (P9DR)

P9DR stores output data for the port 9 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P97DR	0	R/W	P9DR stores output data for the port 9 pins that are used as the general output port except for bit 6. If a port 9 read is performed while the P9DDR bits are set to 1, the P9DR values are read. If a port 9 read is performed while the P9DDR bits are cleared to 0, the pin states are read.
6	P96DR	Undefined*	R	
5	P95DR	0	R/W	
4	P94DR	0	R/W	
3	P93DR	0	R/W	
2	P92DR	0	R/W	
1	P91DR	0	R/W	
0	P90DR	0	R/W	

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

8.9.3 Port 9 Pull-Up MOS Control Register (P9PCR)

P9PCR controls the on/off state of the input pull-up MOS for port 9 pins.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	—	Reserved The initial value should not be changed.
5	P95PCR	0	R/W	When the pins are in input state, the corresponding input pull-up MOS is turned on when a P9PCR bit is set to 1.
4	P94PCR	0	R/W	
3	P93PCR	0	R/W	
2	P92PCR	0	R/W	
1	P91PCR	0	R/W	
0	P90PCR	0	R/W	

8.9.4 Pin Functions

- P97/$\overline{\text{IRQ15}}$/SDA0

  The pin function is switched as shown below according to the combination of the SDA0AS and SDA0BS 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. IICENABLE in the following table is expressed by the following logical expressions.

  IICENABLE = 1 : ICE • $\overline{\text{SDA0AS}}$ • $\overline{\text{SDA0BS}}$

IICENABLE	0		1
P97DDR	0	1	—
Pin function	P97 input pin	P97 output pin	SDA0 I/O pin
	$\overline{\text{IRQ15}}$ input pin		

Note: 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 output.

- P96/ϕ/EXCL

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

EXCLS	0			1	
P96DDR	0		1	0	1
EXCLE	0	1	—	—	
Pin function	P96 input pin	EXCL input pin	ϕ output pin*	P96 input pin	ϕ output pin*

Note: * The subclock is output in subactive, subsleep, and watch modes.

- P95/$\overline{\text{IRQ14}}$

  The pin function is switched as shown below according to the P95DDR bit. When the ISS14 bit in ISSR16 is cleared to 0 and the IRQ14E bit in IER16 of the interrupt controller is set to 1, this pin can be used as the $\overline{\text{IRQ14}}$ input pin.

P95DDR	0	1
Pin function	P95 input pin	P95 output pin
	$\overline{\text{IRQ14}}$ input pin	

- P94/$\overline{\text{IRQ13}}$

  The pin function is switched as shown below according to the P94DDR bit. When the ISS13 bit in ISSR16 is cleared to 0 and the IRQ13E bit in IER16 of the interrupt controller is set to 1, this pin can be used as the $\overline{\text{IRQ13}}$ input pin.

P94DDR	0	1
Pin function	P94 input pin	P94 output pin
	$\overline{\text{IRQ13}}$ input pin	

- P93/$\overline{\text{IRQ12}}$

  The pin function is switched as shown below according to the P93DDR bit. When the ISS12 bit in ISSR16 is cleared to 0 and the IRQ12E bit in IER16 of the interrupt controller is set to 1, this pin can be used as the $\overline{\text{IRQ12}}$ input pin.

P93DDR	0	1
Pin function	P93 input pin	P93 output pin
	$\overline{\text{IRQ12}}$ input pin	

- P92/$\overline{\text{IRQ0}}$

  The pin function is switched as shown below according to the P92DDR bit. When the ISS0 bit in ISSR is cleared to 0 and the IRQ0E bit in IER of the interrupt controller is set to 1, this pin can be used as the $\overline{\text{IRQ0}}$ input pin.

P92DDR	0	1
Pin function	P92 input pin	P92 output pin
	$\overline{\text{IRQ0}}$ input pin	

- P91/$\overline{\text{IRQ1}}$

  The pin function is switched as shown below according to the P91DDR bit. When the ISS1 bit in ISSR is cleared to 0 and the IRQ1E bit in IER of the interrupt controller is set to 1, this pin can be used as the $\overline{\text{IRQ1}}$ input pin.

P91DDR	0	1
Pin function	P91 input pin	P91 output pin
	$\overline{\text{IRQ1}}$ input pin	

- P90/$\overline{\text{IRQ2}}$/$\overline{\text{ADTRG}}$

  The pin function is switched as shown below according to the P90DDR bit.

  When the TRGS1 and TRGS0 bits in ADCR are both set to 1, this pin can be used as the $\overline{\text{ADTRG}}$ input pin.

  When the ISS2 bit in ISSR is cleared to 0 and the IRQ2E bit in IER of the interrupt controller is set to 1, this pin can be used as the $\overline{\text{IRQ2}}$ input pin.

P90DDR	0	1
Pin function	P90 input pin	P90 output pin
	$\overline{\text{IRQ2}}$ input pin/$\overline{\text{ADTRG}}$ input pin	

8.9.5 Port 9 Input Pull-Up MOS

P95 to P90 have built-in input pull-up MOSs that can be controlled by software. Table 8.6 summarizes the input pull-up MOS states.

Table 8.6 Port 9 Input Pull-Up MOS States

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

[Legend]

Off: Always off.

On/Off On when P9DDR = 0 and P9PCR = 1; otherwise off.

8.10 Port A

Port A is an 8-bit I/O port. Port A pins also function as the keyboard input pins and KBU input/output pins. The output format for port A is NMOS push-pull output.

Port A has the following registers. PADDR and PAPIN have the same address.

- Port A data direction register (PADDR)
- Port A output data register (PAODR)
- Port A input data register (PAPIN)

8.10.1 Port A Data Direction Register (PADDR)

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

Bit	Bit Name	Initial Value	R/W	Description
7	PA7DDR	0	W	The corresponding port A pins are output ports when the PADDR bits are set to 1, and input ports when cleared to 0.
6	PA6DDR	0	W	
5	PA5DDR	0	W	
4	PA4DDR	0	W	
3	PA3DDR	0	W	
2	PA2DDR	0	W	
1	PA1DDR	0	W	
0	PA0DDR	0	W	

8.10.2 Port A Output Data Register (PAODR)

PAODR stores output data for the port A pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PA7ODR	0	R/W	PAODR stores output data for the port A pins that are used as the general output port.
6	PA6ODR	0	R/W	
5	PA5ODR	0	R/W	
4	PA4ODR	0	R/W	
3	PA3ODR	0	R/W	
2	PA2ODR	0	R/W	
1	PA1ODR	0	R/W	
0	PA0ODR	0	R/W	

8.10.3 Port A Input Data Register (PAPIN)

PAPIN indicates the pin states.

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

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

8.10.4 Pin Functions

- 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

  The function of port A pins is switched according to the combination of the KBIOE bit in KBCRH of KBU 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 $\overline{\text{KINm}}$ input pin.

KBIOE	0		1
PAnDDR	0	1	—
Pin function	PAn input pin	PAn output pin	KBU input/output pin
	$\overline{\text{KINm}}$ input pin		

Notes: n = 7 to 2

m = 15 to 10

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

When the KBIOE bit is set to 1, the output format for PA3 and PA2 is NMOS open-drain output, and direct bus drive is possible.

- PA1/$\overline{\text{KIN9}}$, PA0/$\overline{\text{KIN8}}$

  The function of port A pins is switched as shown below according to the PAnDDR bit.

  When the KMIMRm bit in KMIMRA of the interrupt controller is cleared to 0, this pin can be used as the $\overline{\text{KINm}}$ input pin.

PAnDDR	0	1
Pin function	PAn input pin	PAn output pin
	$\overline{\text{KINm}}$ input pin	

Note: n = 1, 0

m = 9, 8

8.11 Port B

Port B is an 8-bit I/O port. Port B pins also function as the wake-up event input pins and LPC input/output pins. Port B has the following registers. PBDDR and PBPIN have the same address.

- Port B data direction register (PBDDR)
- Port B output data register (PBODR)
- Port B input data register (PBPIN)

8.11.1 Port B Data Direction Register (PBDDR)

PBDDR is used to specify the input/output attribute of each pin of port B.

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

8.11.2 Port B Output Data Register (PBODR)

PBODR stores output data for the port B pins.

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

8.11.3 Port B Input Data Register (PBPIN)

PBPIN indicates the pin states.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7PIN	Undefined*	R	When a PBPIN read is performed, the pin states are read. This register is assigned to the same address as that of PBDDR. When this register is written to, data is written to PBDDR and the port B setting is then changed.
6	PB6PIN	Undefined*	R	
5	PB5PIN	Undefined*	R	
4	PB4PIN	Undefined*	R	
3	PB3PIN	Undefined*	R	
2	PB2PIN	Undefined*	R	
1	PB1PIN	Undefined*	R	
0	PB0PIN	Undefined*	R	

Note: * The initial value of these pins is determined in accordance with the state of pins PB7 to PB0.

8.11.4 Pin Functions

- PB7/$\overline{\text{WUE7}}$/DLAD0

  The pin function is switched as shown below according to the combination of the LPCS bit in PTCNT2 and the PB7DDR bit. When the WUEM7 bit in WUEMRB of the interrupt controller is cleared to 0, this pin can be used as the $\overline{\text{WUE7}}$ input pin.

LPCS	0		1
PB7DDR	0	1	—
Pin function	PB7 input pin	PB7 output pin	DLAD0 input/output pin
	$\overline{\text{WUE7}}$ input pin		

- PB6/$\overline{\text{WUE6}}$/DLAD1

  The pin function is switched as shown below according to the combination of the LPCS bit in PTCNT2 and the PB6DDR bit. When the WUEM6 bit in WUEMRB of the interrupt controller is cleared to 0, this pin can be used as the $\overline{\text{WUE6}}$ input pin.

LPCS	0		1
PB6DDR	0	1	—
Pin function	PB6 input pin	PB6 output pin	DLAD1 input/output pin
	$\overline{\text{WUE6}}$ input pin		

- PB5/$\overline{\text{WUE5}}$/DLAD2

  The pin function is switched as shown below according to the combination of the LPCS bit in PTCNT2 and the PB5DDR bit. When the WUEM5 bit in WUEMRB of the interrupt controller is cleared to 0, this pin can be used as the $\overline{\text{WUE5}}$ input pin.

LPCS	0		1
PB5DDR	0	1	—
Pin function	PB5 input pin	PB5 output pin	DLAD2 input/output pin
	$\overline{\text{WUE5}}$ input pin		

- PB4/$\overline{\text{WUE4}}$/DLAD3

  The pin function is switched as shown below according to the combination of the LPCS bit in PTCNT2 and the PB4DDR bit. When the WUEM4 bit in WUEMRB of the interrupt controller is cleared to 0, this pin can be used as the $\overline{\text{WUE4}}$ input pin.

LPCS	0		1
PB4DDR	0	1	—
Pin function	PB4 input pin	PB4 output pin	DLAD3 input/output pin
	$\overline{\text{WUE4}}$ input pin		

- PB3/$\overline{\text{WUE3}}$/$\overline{\text{DLFRAME}}$

  The pin function is switched as shown below according to the combination of the LPCS bit in PTCNT2 and the PB3DDR bit. When the WUEM3 bit in WUEMRB of the interrupt controller is cleared to 0, this pin can be used as the $\overline{\text{WUE3}}$ input pin.

LPCS	0		1
PB3DDR	0	1	—
Pin function	PB3 input pin	PB3 output pin	$\overline{\text{DLFRAME}}$ output pin
	$\overline{\text{WUE3}}$ input pin		

- PB2/$\overline{\text{WUE2}}$

  The pin function is switched as shown below according to the PB2DDR bit. When the WUEM2 bit in WUEMRB of the interrupt controller is cleared to 0, this pin can be used as the $\overline{\text{WUE2}}$ input pin.

PB2DDR	0	1
Pin function	PB2 input pin	PB2 output pin
	$\overline{\text{WUE2}}$ input pin	

- PB1/ $\overline{\text{WUE1}}$/LSCI

  The pin function is switched as shown below according to the combination of the LSCIE bit in HICR0 of LPC and the PB1DDR bit. When the WUEM1 bit in WUEMRB of the interrupt controller is cleared to 0, this pin can be used as the $\overline{\text{WUE1}}$ input pin.

LSCIE	0		1
PB1DDR	0	1	—
Pin function	PB1 input pin	PB1 output pin	LSCI output pin
	$\overline{\text{WUE1}}$ input pin		

- PB0/$\overline{\text{WUE0}}$/$\overline{\text{LSMI}}$

  The pin function is switched as shown below according to the combination of the LSMIE bit in HICR0 of LPC and the PB0DDR bit. When the WUEM0 bit in WUEMRB of the interrupt controller is cleared to 0, this pin can be used as the $\overline{\text{WUE0}}$ input pin.

LSMIE	0		1
PB0DDR	0	1	—
Pin function	PB0 input pin	PB0 output pin	$\overline{\text{LSMI}}$ output pin
	$\overline{\text{WUE0}}$ input pin		

8.11.5 Port B Input Pull-Up MOS

Port B has a built-in input pull-up MOS that can be controlled by software. Table 8.7 summarizes the input pull-up MOS states.

Table 8.7 Port B Input Pull-Up MOS States

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

[Legend]

Off: Always off.

On/Off: On when PBDDR = 0 and PBODR = 1; otherwise off.

8.12 Port C

Port C is an 8-bit I/O port. Port C pins also function as the wake-up event inputs, noise cancel input pins, and LPC input/output pins. Port C has the following registers. PCDDR and PCPIN have the same address.

- Port C data direction register (PCDDR)
- Port C output data register (PCODR)
- Port C input data register (PCPIN)
- Port C Nch-OD control register (PCNOCR)
- Noise canceller enable register (PCNCE)
- Noise canceller decision control register (PCNCMC)
- Noise cancel cycle setting register (PCNCCS)

8.12.1 Port C Data Direction Register (PCDDR)

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

Bit	Bit Name	Initial Value	R/W	Description
7	PC7DDR	0	W	The corresponding port C pins are output ports when the PCDDR bits are set to 1, and input ports when cleared to 0.
6	PC6DDR	0	W	
5	PC5DDR	0	W	
4	PC4DDR	0	W	
3	PC3DDR	0	W	
2	PC2DDR	0	W	
1	PC1DDR	0	W	
0	PC0DDR	0	W	

8.12.2 Port C Output Data Register (PCODR)

PCODR stores output data for the port C pins.

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

8.12.3 Port C Input Data Register (PCPIN)

PCPIN indicates the pin states.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7PIN	Undefined*	R	When a PCPIN read is performed, the pin states are read. This register is assigned to the same address as that of PCDDR. When this register is written to, data is written to PCDDR and the port C setting is then changed.
6	PC6PIN	Undefined*	R	
5	PC5PIN	Undefined*	R	
4	PC4PIN	Undefined*	R	
3	PC3PIN	Undefined*	R	
2	PC2PIN	Undefined*	R	
1	PC1PIN	Undefined*	R	
0	PC0PIN	Undefined*	R	

Note: * The initial value of these pins is determined in accordance with the state of pins PC7 to PC0.

8.12.4 Noise Canceller Enable Register (PCNCE)

PCNCE enables or disables the noise cancel circuit at port C.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7NCE	0	R/W	Noise cancel circuit is enabled when PCNCE bit is set to 1, and the pin state is fetched in the PCPIN in the sampling cycle set by the PCNCCS.
6	PC6NCE	0	R/W	
5	PC5NCE	0	R/W	
4	PC4NCE	0	R/W	
3	PC3NCE	0	R/W	
2	PC2NCE	0	R/W	
1	PC1NCE	0	R/W	
0	PC0NCE	0	R/W	

8.12.5 Noise Canceller Mode Control Register (PCNCMC)

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

Bit	Bit Name	Initial Value	R/W	Description
7	PC7NCMC	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	PC6NCMC	0	R/W	
5	PC5NCMC	0	R/W	
4	PC4NCMC	0	R/W	
3	PC3NCMC	0	R/W	
2	PC2NCMC	0	R/W	
1	PC1NCMC	0	R/W	
0	PC0NCMC	0	R/W	

8.12.6 Noise Cancel Cycle Setting Register (PCNCCS)

PCNCCS controls the sampling cycles of the noise canceller.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	Undefined	R/W	Reserved The read data is undefined. The initial value should not be changed.
2	PCNCCK2	0	R/W	These bits set the sampling cycles of the noise canceller. When φ is 10 MHz 000: 0.88 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	PCNCCK1	0	R/W	
0	PCNCCK0	0	R/W	

8.12.7 Pin Functions

- PC7/$\overline{\text{WUE15}}$/DLDRQ

  The pin function is switched as shown below according to the combination of the LDRQS bit in PTCNT2 and the PC7DDR. 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.

LDRQS	0		1
PC7DDR	0	1	—
Pin Function	PC7 input pin	PC7 output pin	DLDRQ input pin
	$\overline{\text{WUE15}}$ input pin		

- PC6/$\overline{\text{WUE14}}$/LDRQ

  The pin function is switched as shown below according to the combination of the LDRQS bit in PTCNT2 and the PC6DDR. 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.

LDRQS	0		1
PC6DDR	0	1	—
Pin Function	PC6 input pin	PC6 output pin	LDRQ output pin
	$\overline{\text{WUE14}}$ input pin		

- PC5/$\overline{\text{WUE13}}$

  The pin function is switched as shown below according to the PC5DDR. 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.

PC5DDR	0	1
Pin Function	PC5 input pin	PC5 output pin
	$\overline{\text{WUE13}}$ input pin	

- PC4/$\overline{\text{WUE12}}$

  The pin function is switched as shown below according to the PC4DDR. 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.

PC4DDR	0	1
Pin Function	PC4 input pin	PC4 output pin
	$\overline{\text{WUE12}}$ input pin	

- PC3/$\overline{\text{WUE11}}$

  The pin function is switched as shown below according to the PC3DDR. 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.

PC3DDR	0	1
Pin Function	PC3 input pin	PC3 output pin
	$\overline{\text{WUE11}}$ input pin	

- PC2/$\overline{\text{WUE10}}$

  The pin function is switched as shown below according to the PC2DDR. 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.

PC2DDR	0	1
Pin Function	PC2 input pin	PC2 output pin
	$\overline{\text{WUE10}}$ input pin	

- PC1/$\overline{\text{WUE9}}$

  The pin function is switched as shown below according to the PC1DDR. 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.

PC1DDR	0	1
Pin Function	PC1 input pin	PC1 output pin
	$\overline{\text{WUE9}}$ input pin	

- PC0/$\overline{\text{WUE8}}$

  The pin function is switched as shown below according to the PC0DDR. 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.

PC0DDR	0	1
Pin Function	PC0 input pin	PC0 output pin
	$\overline{\text{WUE8}}$ input pin	

8.12.8 Port C Nch-OD control register (PCNOCR)

The individual bits of PCNOCR specify output driver type for the pins of port C that is specified to output.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7NOCR	0	R/W	0: CMOS
6	PC6NOCR	0	R/W	(P channel driver is enable)
5	PC5NOCR	0	R/W	1: N channel open-drain
4	PC4NOCR	0	R/W	(P channel driver is disable)
3	PC3NOCR	0	R/W	
2	PC2NOCR	0	R/W	
1	PC1NOCR	0	R/W	
0	PC0NOCR	0	R/W	

8.12.9 Pin Functions

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			

8.12.10 Port C Input Pull-Up MOS

Port C has a built-in input pull-up MOS that can be controlled by software. Input pull-up MOS can be specified as on or off on an individual bit basis. Table 8.8 summarizes the input pull-up MOS states.

Table 8.8 Port C Input Pull-Up MOS States

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

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

8.13 Port D

Port D is an 8-bit I/O port. Port D pins also function as the TPU I/O pins. Port D has the following registers. PDDDR and PDPIN have the same address.

- Port D data direction register (PDDDR)
- Port D output data register (PDODR)
- Port D input data register (PDPIN)
- Port D Nch-OD control register (PDNOCR)

8.13.1 Port D Data Direction Register (PDDDR)

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

Bit	Bit Name	Initial Value	R/W	Description
7	PD7DDR	0	W	The corresponding port D pins are output ports when the PDDDR bits are set to 1, and input ports when cleared to 0.
6	PD6DDR	0	W	
5	PD5DDR	0	W	
4	PD4DDR	0	W	
3	PD3DDR	0	W	
2	PD2DDR	0	W	
1	PD1DDR	0	W	
0	PD0DDR	0	W	

8.13.2 Port D Output Data Register (PDODR)

PDODR stores output data for the port D pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7ODR	0	R/W	The PDODR register stores the output data for the pins that are used as the general output port.
6	PD6ODR	0	R/W	
5	PD5ODR	0	R/W	
4	PD4ODR	0	R/W	
3	PD3ODR	0	R/W	
2	PD2ODR	0	R/W	
1	PD1ODR	0	R/W	
0	PD0ODR	0	R/W	

8.13.3 Port D Input Data Register (PDPIN)

PDPIN indicates the pin states of port D.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7PIN	Undefined*	R	When a PDPIN read is performed, the pin states are read.
6	PD6PIN	Undefined*	R	
5	PD5PIN	Undefined*	R	
4	PD4PIN	Undefined*	R	
3	PD3PIN	Undefined*	R	
2	PD2PIN	Undefined*	R	
1	PD1PIN	Undefined*	R	
0	PD0PIN	Undefined*	R	

Note: * The initial value of these pins is determined in accordance with the state of pins PD7 to PD0.

8.13.4 Pin Functions

- PD7/TIOCB2/TCLKD

  The pin function is switched as shown below according to the combination of the TPU channel 2 setting, TPSC2 to TPSC0 bits in TCR_0 of TPU, and the PD7DDR.

TPU Channel 2 Setting	Input or Initial Value		Output
PD7DDR	0	1	—
Pin Function	PD7 input pin	PD7 output pin	TIOCB2 output pin
	TIOCB2 input pin*[2]		
	TCLKD input pin*[1]		

Notes: 1. This pin functions as TCLKD input when TPSC2 to TPSC0 in TCR_0 are set to 111 or when channel 2 is set to phase counting mode.
2. 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.

- PD6/TIOCA2

  The pin function is switched as shown below according to the combination of the TPU channel 2 setting and the PD6DDR.

TPU Channel 2 Setting	Input or Initial Value		Output
PD6DDR	0	1	—
Pin Function	PD6 input pin	PD6 output pin	TIOCA2 output pin
	TIOCA2 input pin*		

Note: * 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.

- PD5/TIOCB1/TCLKC

  The pin function is switched as shown below according to the combination of the TPU channel 1 setting, TPSC2 to TPSC0 bits in TCR_0 and TCR_2 of TPU, and the PD5DDR.

TPU Channel 1 Setting	Input or Initial Value		Output
PD5DDR	0	1	—
Pin Function	PD5 input pin	PD5 output pin	TIOCB1 output pin
	TIOCB1 input pin*[2]		
	TCLKC input pin*[1]		

Notes: 1. This pin functions as TCLKC input when TPSC2 to TPSC0 in TCR_0 or TCR_2 are set to 110 or when channel 2 is set to phase counting mode.
2. 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 10xx.

- PD4/TIOCA1

  The pin function is switched as shown below according to the combination of the TPU channel 1 setting and the PD4DDR.

TPU Channel 1 Setting	Input or Initial Value		Output
PD4DDR	0	1	—
Pin Function	PD4 input pin	PD4 output pin	TIOCA1 output pin
	TIOCA1 input pin*		

Note: * 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_2 are set to 10xx.

- PD3/TIOCD0/TCLKB

  The pin function is switched as shown below according to the combination of the TPU channel 0 setting, TPSC2 to TPSC0 bits in TCR_0 to TCR_2 of TPU, and the PD3DDR.

TPU Channel 0 Setting	Input or Initial Value		Output
PD3DDR	0	1	—
Pin Function	PD3 input pin	PD3 output pin	TIOCD0 output pin
	TIOCD0 input pin*[2]		
	TCLKB input pin*[1]		

Notes: 1. This pin functions as TCLKB input when TPSC2 to TPSC0 in any of TCR_0, TCR_1, and TCR_2 are set to 101 or when channel 1 is set to phase counting mode.
2. This pin functions as TIOCD0 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 10xx.

- PD2/TIOCC0/TCLKA

  The pin function is switched as shown below according to the combination of the TPU channel 0 setting, TPSC2 to TPSC0 bits in TCR_0 to TCR_2 of TPU, and the PD2DDR.

TPU Channel 0 Setting	Input or Initial Value		Output
PD2DDR	0	1	—
Pin Function	PD2 input pin	PD2 output pin	TIOCC0 output pin
	TIOCC0 input pin*[2]		
	TCLKA input pin*[1]		

Notes: 1. This pin functions as TCLKA input when TPSC2 to TPSC0 in any of TCR_0, TCR_1, and TCR_2 are set to 100 or when channel 1 is set to phase counting mode.
2. 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 10xx.

- PD1/TIOCB0

  The pin function is switched as shown below according to the combination of the TPU channel 0 setting and the PD1DDR.

TPU Channel 0 Setting	Input or Initial Value		Output
PD1DDR	0	1	—
Pin Function	PD1 input pin	PD1 output pin	TIOCB0 output pin
	TIOCB0 input pin*		

Note: * This pin functions as TIOC$_{B0}$ 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 10xx.

- PD0/TIOCA0

  The pin function is switched as shown below according to the combination of the TPU channel 0 setting and the PD0DDR.

TPU Channel 0 Setting	Input or Initial Value		Output
PD0DDR	0	1	—
Pin Function	PD0 input pin	PD0 output pin	TIOCA0 output pin
	TIOCA0 input pin*		

Note: * This pin functions as TIOC$_{A0}$ 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 10xx.

For the setting of the TPU channel, see section 12, 16-bit Timer Pulse Unit (TPU).

8.13.5 Port D Nch-OD control register (PDNOCR)

The individual bits of PDNOCR specify output driver type for the pins of port D that is specified to output.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7NOCR	0	R/W	0: CMOS
6	PD6NOCR	0	R/W	(P channel driver is enable)
5	PD5NOCR	0	R/W	1: N channel open-drain
4	PD4NOCR	0	R/W	(P channel driver is disable)
3	PD3NOCR	0	R/W	
2	PD2NOCR	0	R/W	
1	PD1NOCR	0	R/W	
0	PD0NOCR	0	R/W	

8.13.6 Pin Functions

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			

8.13.7 Port D Input Pull-Up MOS

Port D has a built-in input pull-up MOS that can be controlled by software. Input pull-up MOS can be specified as on or off on an individual bit basis. Table 8.9 summarizes the input pull-up MOS states.

Table 8.9 Port D Input Pull-Up MOS States

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

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

8.14 Port E

Port E is a 5-bit input port. Port E pins also function as the LPC input pins and emulator input/output pins. Port E has the following registers.

- Port E input pull-up MOS control register (PEPCR)
- Port E input data register (PEPIN)

8.14.1 Port E Input Pull-Up MOS Control Register (PEPCR)

PEPCR specifies each bit in input pull-up MOS on/off.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R/W	Reserved The initial value should not be changed.
4	PE4PCR	0	R/W	0: Input pull-up MOS is off 1: Input pull-up MOS is on
3	PE3PCR	0	R/W	
2	PE2PCR	0	R/W	
1	PE1PCR	0	R/W	
0	PE0PCR	0	R/W	

8.14.2 Port E Input Data Register (PEPIN)

PEPIN indicates the pin states of port E.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	—	Reserved These bits are always read as 0.
4	PE4PIN	Undefined*	R	When these bits are read, the pin states are returned. These bits cannot be modified.
3	PE3PIN	Undefined*	R	
2	PE2PIN	Undefined*	R	
1	PE1PIN	Undefined*	R	
0	PE0PIN	Undefined*	R	

Note: * The initial value of these pins is determined in accordance with the state of pins PE4 to PE0.

8.14.3 Pin Functions

- PE4, PE3, PE2, PE1

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

Pin Function	PEn input pin

Note: n = 4 to 1

The PE5 to PE1 pins are not supported in the system development tool (emulator).

- PE0/$\overline{\text{LID3}}$

  The function of port E pin is switched as shown below according to the combination of the LMCE bit in LMCCR of LPC.

LMCE	0	1
Pin function	PE0 input pin	PE0, $\overline{\text{LID3}}$ input pin

8.14.4 Port E Input Pull-Up MOS

Port E has a built-in input pull-up MOS that can be controlled by software. Input pull-up MOS can be specified as on or off on an individual bit basis. Table 8.10 summarizes the input pull-up MOS states.

Table 8.10 Port E Input Pull-Up MOS States

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

[Legend]

Off: Always off.

On/Off On when PEPCR = 1; otherwise off.

8.15 Port F

Port F is an 8-bit I/O port. Port F pins also function as the interrupt input pins and TMR_X and PWM output pins. Port F has the following registers. PFDDR and PFPIN have the same address.

- Port F data direction register (PFDDR)
- Port F output data register (PFODR)
- Port F input data register (PFPIN)
- Port F Nch-OD control register (PFNOCR)

8.15.1 Port F Data Direction Register (PFDDR)

The individual bits of PFDDR specify input or output for the pins of port F.

Bit	Bit Name	Initial Value	R/W	Description
7	PF7DDR	0	W	The corresponding port F pins are output ports when the PFDDR bits are set to 1, and input ports when cleared to 0.
6	PF6DDR	0	W	
5	PF5DDR	0	W	
4	PF4DDR	0	W	
3	PF3DDR	0	W	
2	PF2DDR	0	W	
1	PF1DDR	0	W	
0	PF0DDR	0	W	

8.15.2 Port F Output Data Register (PFODR)

PFODR stores output data for the port F pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PF7ODR	0	R/W	The PFODR register stores the output data for the pins that are used as the general output port.
6	PF6ODR	0	R/W	
5	PF5ODR	0	R/W	
4	PF4ODR	0	R/W	
3	PF3ODR	0	R/W	
2	PF2ODR	0	R/W	
1	PF1ODR	0	R/W	
0	PF0ODR	0	R/W	

8.15.3 Port F Input Data Register (PFPIN)

PFPIN indicates the pin states of port F.

Bit	Bit Name	Initial Value	R/W	Description
7	PF7PIN	Undefined*	R	When PFPIN is read, the pin states are returned.
6	PF6PIN	Undefined*	R	
5	PF5PIN	Undefined*	R	
4	PF4PIN	Undefined*	R	
3	PF3PIN	Undefined*	R	
2	PF2PIN	Undefined*	R	
1	PF1PIN	Undefined*	R	
0	PF0PIN	Undefined*	R	

Note: * The initial value of these pins is determined in accordance with the state of pins PF7 to PF0.

8.15.4 Pin Functions

- PF7/ExPW15, PF6/ExPW14

  The function of port F pins is switched as shown below according to the combination of the PWMAS bit in PTCNT0, the OEm bit in PWOERB of PWM, and the PFnDDR bit.

PWMAS	0		1		
PFnDDR	0	1	0	1	
OEm	—		—	0	1
Pin function	PFn input pin	PFn output pin	PFn input pin	PFn output pin	ExPWm output pin

Note: n = 7, 6
m = 15, 14

- PF5/ExPW13, PF4/ExPW12

  The function of port F pins is switched as shown below according to the combination of the PWMBS bit in PTCNT0, the OEm bit in PWOERB of PWM, and the PFnDDR bit.

PWMBS	0		1		
PFnDDR	0	1	0	1	
OEm	—		—	0	1
Pin function	PFn input pin	PFn output pin	PFn input pin	PFn output pin	ExPWm output pin

Note: n = 5, 4
m = 13, 12

- PF3/$\overline{\text{IRQ11}}$/ExTMOX

  The pin function is switched as shown below according to the combination of the TMOXS bit in PTCNT0, OS3 to OS0 bits in TCSR of TMR_X, and PF3DDR bit. 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.

TMOXS	0		1		
OS3 to OS0	—		All 0		One bit is set as 1
PF3DDR	0	1	0	1	—
Pin Function	PF3 input pin	PF3 output pin	PF3 input pin	PF3 output pin	ExTMOX output pin
	$\overline{\text{IRQ11}}$ input pin				

- PF2/$\overline{\text{IRQ10}}$, PF1/$\overline{\text{IRQ9}}$, PF0/$\overline{\text{IRQ8}}$

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

PFnDDR	0	1
Pin function	PFn input pin	PFn output pin
	$\overline{\text{IRQm}}$ input pin	

Note: n = 2 to 0
m = 10 to 8

8.15.5 Port F Nch-OD control register (PFNOCR)

The individual bits of PFNOCR specify output driver type for the pins of port F that is specified to output.

Bit	Bit Name	Initial Value	R/W	Description
7	PF7NOCR	0	R/W	0: CMOS
6	PF6NOCR	0	R/W	(P channel driver is enable)
5	PF5NOCR	0	R/W	1: N channel open-drain
4	PF4NOCR	0	R/W	(P channel driver is disable)
3	PF3NOCR	0	R/W	
2	PF2NOCR	0	R/W	
1	PF1NOCR	0	R/W	
0	PF0NOCR	0	R/W	

8.15.6 Pin Functions

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			

8.15.7 Port F Input Pull-Up MOS

Port F has a built-in input pull-up MOS that can be controlled by software. Input pull-up MOS can be specified as on or off on an individual bit basis. Table 8.11 summarizes the input pull-up MOS states.

Table 8.11 Port F Input Pull-Up MOS States

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

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

8.16 Port G

Port G is an 8-bit I/O port. Port G pins also function as the interrupt input pins, and TMR_0, TMR_1, TMR_X, TMR_Y input pins and IIC_0, and IIC_1 input/output pins. The output format for port G is NMOS push-pull output.

Port G has the following registers. PGDDR and PGPIN have the same address.

- Port G data direction register (PGDDR)
- Port G output data register (PGODR)
- Port G input data register (PGPIN)
- Port G Nch-OD control register (PGNOCR)
- Noise canceller enable register (PGNCE)
- Noise canceller decision control register (PGNCMC)
- Noise cancel cycle setting register (PGNCCS)

8.16.1 Port G Data Direction Register (PGDDR)

The individual bits of PGDDR specify input or output for the pins of port G.

Bit	Bit Name	Initial Value	R/W	Description
7	PG7DDR	0	W	The corresponding port G pins are output ports when the PGDDR bits are set to 1, and input ports when cleared to 0.
6	PG6DDR	0	W	
5	PG5DDR	0	W	
4	PG4DDR	0	W	
3	PG3DDR	0	W	
2	PG2DDR	0	W	
1	PG1DDR	0	W	
0	PG0DDR	0	W	

8.16.2 Port G Output Data Register (PGODR)

PGODR stores output data for the port G pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PG7ODR	0	R/W	The PGODR register stores the output data for the pins that are used as the general output port.
6	PG6ODR	0	R/W	
5	PG5ODR	0	R/W	
4	PG4ODR	0	R/W	
3	PG3ODR	0	R/W	
2	PG2ODR	0	R/W	
1	PG1ODR	0	R/W	
0	PG0ODR	0	R/W	

8.16.3 Port G Input Data Register (PGPIN)

PGPIN indicates the pin states of port G.

Bit	Bit Name	Initial Value	R/W	Description
7	PG7PIN	Undefined*	R	When PGPIN is read, the pin states are returned.
6	PG6PIN	Undefined*	R	This register is assigned to the same address as that of PGDDR. When this register is written to, data is written to PGDDR and the port G setting is then changed.
5	PG5PIN	Undefined*	R	
4	PG4PIN	Undefined*	R	
3	PG3PIN	Undefined*	R	
2	PG2PIN	Undefined*	R	
1	PG1PIN	Undefined*	R	
0	PG0PIN	Undefined*	R	

Note: * The initial value of these pins is determined in accordance with the state of pins PG7 to PG0.

8.16.4 Noise Canceller Enable Register (PGNCE)

PGNCE enables or disables the noise cancel circuit at port G. To use the port G pins as the IIC_0 and IIC_1input/output pins, these bits in PGNCE should be disabled.

Bit	Bit Name	Initial Value	R/W	Description
7	PG7NCE	0	R/W	Noise cancel circuit is enabled when PGNCE bit is set to 1, and the pin state is fetched in the PGPIN in the sampling cycle set by the PGNCCS.
6	PG6NCE	0	R/W	
5	PG5NCE	0	R/W	
4	PG4NCE	0	R/W	
3	PG3NCE	0	R/W	
2	PG2NCE	0	R/W	
1	PG1NCE	0	R/W	
0	PG0NCE	0	R/W	

8.16.5 Noise Canceller Mode Control Register (PGNCMC)

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

Bit	Bit Name	Initial Value	R/W	Description
7	PG7NCMC	0	R/W	1 expected: 1 is stored in the port data register when 1 is input 0 expected: 0 is stored in the port data register when 0 is input
6	PG6NCMC	0	R/W	
5	PG5NCMC	0	R/W	
4	PG4NCMC	0	R/W	
3	PG3NCMC	0	R/W	
2	PG2NCMC	0	R/W	
1	PG1NCMC	0	R/W	
0	PG0NCMC	0	R/W	

8.16.6 Noise Cancel Cycle Setting Register (PGNCCS)

PGNCCS controls the sampling cycles of the noise canceller.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	Undefined	R/W	Reserved The read data is undefined. The initial value should not be changed.
2	PGNCCK2	0	R/W	These bits set the sampling cycles of the noise canceller. When ϕ is 10 MHz 000: 0.88 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	PGNCCK1	0	R/W	
0	PGNCCK0	0	R/W	

8.16.7 Pin Functions

- PG7/$\overline{\text{ExIRQ15}}$/ExSCLB

  The pin function is switched as shown below according to the combination of the SCL1BS and SCL0BS bits in PTCNT1, ICE bit in ICCR of IIC_1 and IIC_0, 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.

SCL1BS	0		1			0		
SCL0BS	0					1		
ICE_1	—		0		1	—		
ICE_0	—					0		1
PG7DDR	0	1	0	1	—	0	1	—
Pin function	PG7 input pin	PG7 output pin	PG7 input pin	PG7 output pin	ExSCLB (SCL1) input/output pin	PG7 input pin	PG7 output pin	ExSCLB (SCL0) input/output pin
	$\overline{\text{ExIRQ15}}$ input pin							

Note: SCL1BS and SCL0BS, SCL1BS and SCL1AS, and SCL0BS and SCL0AS should not be set to 1 at the same time. The output format for ExSCLB is NMOS open-drain output, and direct bus drive is possible.

- PG6/$\overline{\text{ExIRQ14}}$/ExSDAB

  The pin function is switched as shown below according to the combination of the SDA1BS and SDA0BS bits in PTCNT1, ICE bit in ICCR of IIC_1 and IIC_0, 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.

SDA1BS	0		1			0		
SDA0BS	0					1		
ICE_1	—		0		1	—		
ICE_0	—					0		1
PG6DDR	0	1	0	1	—	0	1	—
Pin function	PG6 input pin	PG6 output pin	PG6 input pin	PG6 output pin	ExSDAB (SDA1) input/output pin	PG6 input pin	PG6 output pin	ExSDAB (SDA0) input/output pin
	$\overline{\text{ExIRQ14}}$ input pin							

Note: SDA1BS and SDA0BS, SDA1BS and SDA1AS, and SDA0BS and SDA0AS should not be set to 1 at the same time. The output format for ExSDAB is NMOS open-drain output, and direct bus drive is possible.

- PG5/$\overline{\text{ExIRQ13}}$/ExSCLA

  The pin function is switched as shown below according to the combination of the SCL1AS and SCL0AS bits in PTCNT1, ICE bit in ICCR of IIC_1 and IIC_0, 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.

SCL1AS	0		1			0		
SCL0AS	0					1		
ICE_1	—		0		1	—		
ICE_0	—					0		1
PG5DDR	0	1	0	1	—	0	1	—
Pin function	PG5 input pin	PG5 output pin	PG5 input pin	PG5 output pin	ExSCLA (SCL1) input/output pin	PG5 input pin	PG5 output pin	ExSCLA (SCL0) input/output pin
	$\overline{\text{ExIRQ13}}$ input pin							

Note: SCL1AS and SCL0AS, SCL1AS and SCL1BS, and SCL0AS and SCL0BS should not be set to 1 at the same time. The output format for ExSCLA is NMOS open-drain output, and direct bus drive is possible.

- PG4/$\overline{\text{ExIRQ12}}$/ExSDAA

  The pin function is switched as shown below according to the combination of the SDA1AS and SDA0AS bits in PTCNT1, ICE bit in ICCR of IIC_1 and IIC_0, 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.

SDA1AS	0		1			0		
SDA0AS	0					1		
ICE_1	—		0		1	—		
ICE_0	—					0		1
PG4DDR	0	1	0	1	—	0	1	—
Pin function	PG4 input pin	PG4 output pin	PG4 input pin	PG4 output pin	ExSDAA (SDA1) input/output pin	PG4 input pin	PG4 output pin	ExSDAA (SDA0) input/output pin
	$\overline{\text{ExIRQ12}}$ input pin							

Note: SDA1AS and SDA0AS, SDA1AS and SDA1BS, and SDA0AS and SDA0BS should not be set to 1 at the same time. The output format for ExSDAA is NMOS open-drain output, and direct bus drive is possible.

- PG3/$\overline{\text{ExIRQ11}}$/ExTMIY

  The pin function is switched as shown below according to the PG3DDR bit. When the TMIYS bit in PTCNT0 and the CCLR1 and CCLR0 bits in TCR of TMR_Y are cleared to 0, this pin is used as the ExTMIY (ExTMRIY) input pin. 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.

PG3DDR	0	1
Pin function	PG3 input pin	PG3 output pin
	$\overline{\text{ExIRQ11}}$ input pin/ExTMIY input pin	

- PG2/$\overline{\text{ExIRQ10}}$/ExTMIX

  The pin function is switched as shown below according to the PG2DDR bit. When the TMIXS bit in PTCNT0 and the CCLR1 and CCLR0 bits in TCR of TMR_X are cleared to 0, this pin is used as the ExTMIX (ExTMRIX) input pin. 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.

PG2DDR	0	1
Pin function	PG2 input pin	PG2 output pin
	$\overline{\text{ExIRQ10}}$ input pin/ExTMIX input pin	

- PG1/$\overline{\text{ExIRQ9}}$/ExTMCI1

  The pin function is switched as shown below according to the PG1DDR bit. When the TMCI1S bit in PTCNT0 is set to 1 and the external clock is selected by the CKS2 to CKS0 bits in TCR of TMR_1, this bit is used as the ExTMCI1 input pin. 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.

PG1DDR	0	1
Pin function	PG1 input pin	PG1 output pin
	$\overline{\text{ExIRQ9}}$ input pin/ExTMCI1 input pin	

- PG0/$\overline{\text{ExIRQ8}}$/ExTMCI0

  The pin function is switched as shown below according to the PG0DDR bit. When the TMCI0S bit in PTCNT0 is set to 1 and the external clock is selected by the CKS2 to CKS0 bits in TCR of TMR_0, this bit is used as the ExTMCI0 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.

PG0DDR	0	1
Pin function	PG0 input pin	PG0 output pin
	$\overline{\text{ExIRQ8}}$ input pin/ExTMCI0 input pin	

8.16.8 Port G Nch-OD control register (PGNOCR)

The individual bits of PGNOCR specify output driver type for the pins of port G that is specified to output.

Bit	Bit Name	Initial Value	R/W	Description
7	PG7NOCR	0	R/W	0: NMOS push-pull
6	PG6NOCR	0	R/W	(N channel driver in V_{cc} side is enable)
5	PG5NOCR	0	R/W	1: N channel open-drain in V_{ss} side
4	PG4NOCR	0	R/W	(N channel driver in V_{cc} side is disable)
3	PG3NOCR	0	R/W	
2	PG2NOCR	0	R/W	
1	PG1NOCR	0	R/W	
0	PG0NOCR	0	R/W	

8.16.9 Pin Functions

DDR	0		1			
NOCR	—		0		1	
ODR	0	1	0	1	0	1
N-ch driver in V_{ss} side	Off		On	Off	On	Off
N-ch driver in V_{cc} side	Off		Off	On	Off	
Pin function	Input pin		Output pin			

8.17 Change of Peripheral Function Pins

For the 8-bit timer input/output, 8-bit PWM timer output, and IIC input/output, the multi-function I/O ports can be changed. I/O ports that also function as the external sub-clock input pin, 8-bit timer input/output pins, and the 8-bit PWM timer output pins are changed according to the setting of PTCNT0. I/O ports that also function as the IIC input/output pins are changed according to the setting of PTCNT1. I/O ports that also function as the docking LPC input/output pins are changed according to the setting of PTCNT2. 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.

8.17.1 Port Control Register 0 (PTCNT0)

PTCNT0 selects ports that also function as the external sub-clock input pin, 8-bit timer input/output pins, and 14-bit PWM timer output pins.

Bit	Bit Name	Initial Value	R/W	Description
7	TMCI0S	0	R/W	0: P40/TMCI0 is selected 1: PG0/ExTMCI0 is selected
6	TMCI1S	0	R/W	0: P43/TMCI1 is selected 1: PG1/ExTMCI1 is selected
5	TMIXS	0	R/W	0: P60/TMIX is selected 1: PG2/ExTMIX is selected
4	TMIYS	0	R/W	0: P62/TMIY is selected 1: PG3/ExTMIY is selected
3	TMOXS	0	R/W	0: P67/TMOX is selected 1: PF3/ExTMOX is selected
2	PWMAS	0	R/W	0: P27/PW15 and P26/PW14 are selected 1: PF7/ExPW15 and PF6/ExPW14 are selected
1	PWMBS	0	R/W	0: P25/PW13 and P24/PW12 are selected 1: PF5/ExPW13 and PF4/ExPW12 are selected
0	EXCLS	0	R/W	0: P96/EXCLK is selected 1: P50/ExEXCL is selected

8.17.2 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	SCL0AS	0	R/W		IIC0	IIC1
6	SCL1AS	0	R/W	0000:	P52/SCL0	P86/SCL1
5	SCL0BS	0	R/W	1000:	PG5/ExSCLA	P86/SCL1
4	SCL1BS	0	R/W	0100:	P52/SCL0	PG5/ExSCLA
				0010:	PG7/ExSCLB	P86/SCL1
				0001:	P52/SCL0	PG7/ExSCLB
				1001:	PG5/ExSCLA	PG7/ExSCLB
				0110:	PG7/ExSCLB	PG5/ExSCLA
				Settings other than those shown above are prohibited.		
3	SDA0AS	0	R/W		IIC0	IIC1
2	SDA1AS	0	R/W	0000:	P97/SDA0	P42/SDA1
1	SDA0BS	0	R/W	1000:	PG4/ExSDAA	P42/SDA1
0	SDA1BS	0	R/W	0100:	P97/SDA0	PG4/ExSDAA
				0010:	PG6/ExSDAB	P42/SDA1
				0001:	P97/SDA0	PG6/ExSDAB
				1001:	PG4/ExSDAA	PG6/ExSDAB
				0110:	PG6/ExSDAB	PG4/ExSDAA
				Settings other than those shown above are prohibited.		

Note: PTCNT1 must be written to while the ICE bit in ICCR is cleared to 0.

8.17.3 Port Control Register 2 (PTCNT2)

PTCNT2 selects ports that also function as docking LPC input/output pins. Setting 1 to one of LPC3E to LPC1E in HICR0 and LPC4E in HICR4 enables this function.

Bit	Bit Name	Initial Value	R/W	Description
7	LPCS	0	R/W	0: PB7/$\overline{\text{WUE7}}$, PB6/$\overline{\text{WUE6}}$, PB5/$\overline{\text{WUE5}}$, PB4/$\overline{\text{WUE4}}$, PB3/$\overline{\text{WUE3}}$, P41/TMO0/RxD2, P40/TMCI0/TxD2 are selected 1: PB7/$\overline{\text{WUE7}}$/DLAD0, PB6/$\overline{\text{WUE6}}$/DLAD1, PB5/$\overline{\text{WUE5}}$/DLAD2, PB4/$\overline{\text{WUE4}}$/DLAD3, PB3/$\overline{\text{WUE3}}$/$\overline{\text{DLFRAME}}$, P41/$\overline{\text{DCLKRUN}}$, P40/$\overline{\text{TMCI0}}$/DSERIRQ are selected
6 to 1	—	All 0	R/W	Reserved The initial value should not be changed.
0	LDRQS	0	R/W	0: PC6/$\overline{\text{WUE14}}$, PC7/$\overline{\text{WUE15}}$ are selected 1: PC6/$\overline{\text{WUE14}}$/LDRQ, PC7/$\overline{\text{WUE15}}$/DLDRQ are selected

Section 9 8-Bit PWM Timer (PWM)

This LSI has an on-chip pulse width modulation (PWM) timer with eight outputs. Eight output waveforms are generated from a common time base, enabling PWM output with a high carrier frequency to be produced using pulse division. Connecting a low pass filter externally to the LSI enables the PWM to function as an 8-bit D/A converter.

9.1 Features

- Operable at a maximum carrier frequency of 1.25 MHz using pulse division (at 20 MHz operation)
- Duty cycles from 0 to 100% with 1/256 resolution (100% duty realized by port output)
- Direct or inverted PWM output, and PWM output enable/disable control
- Selection of general ports for PWM output
  PW15/PW14 or ExPW15/ExPW14
  PW13/PW12 or ExPW13/ExPW12

Figure 9.1 shows a block diagram of the PWM timer.

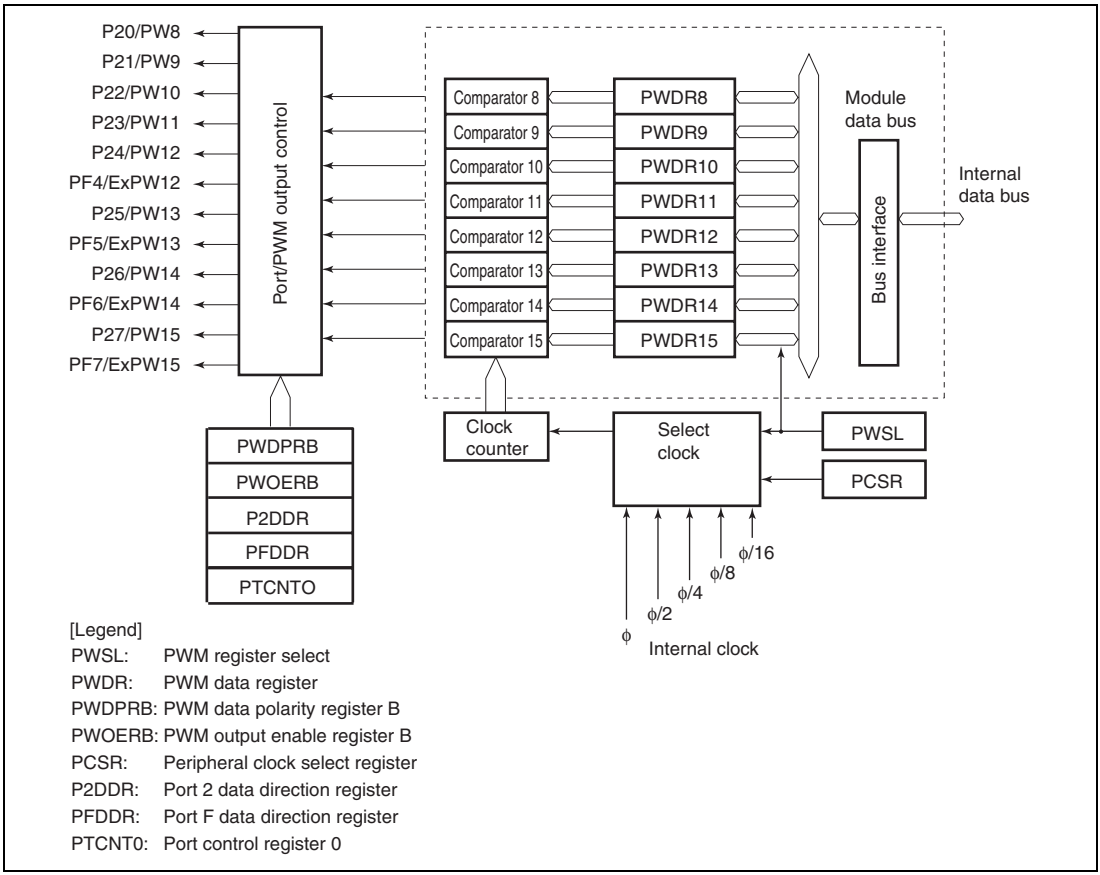


Figure 9.1 Block Diagram of PWM Timer

9.2 Input/Output Pins

Table 9.1 shows the PWM output pins.

Table 9.1 Pin Configuration

Name	Abbreviation	I/O	Function
PWM output 15 to 8	PW15 to PW8	Output	PWM timer pulse output 15 to 8
ExPWM output 15 to 12	ExPW15 to ExPW12		A pin for outputting is selected among PWn and ExPWn. (n = 15 to 12) For details, section 8.17.1, Port Control Register 0 (PTCNT0).

9.3 Register Descriptions

The PWM has the following registers. To access PCSR, the FLSHE bit in the serial timer control register (STCR) must be cleared to 0. For details on the serial timer control register (STCR), see section 3.2.3, Serial Timer Control Register (STCR).

- PWM register select (PWSL)
- PWM data registers 15 to 8 (PWDR15 to PWDR8)
- PWM data polarity register B (PWDPRB)
- PWM output enable register B (PWOERB)
- Peripheral clock select register (PCSR)

9.3.1 PWM Register Select (PWSL)

PWSL is used to select the input clock and the PWM data register.

Bit	Bit Name	Initial Value	R/W	Description
7	PWCKE	0	R/W	PWM Clock Enable
6	PWCKS	0	R/W	PWM Clock Select These bits, together with bits PWCKB and PWCKA in PCSR, select the internal clock input to TCNT in the PWM. For details, see table 9.2. The resolution, PWM conversion period, and carrier frequency depend on the selected internal clock, and can be obtained from the following equations. Resolution (minimum pulse width) = 1/internal clock frequency PWM conversion period = resolution × 256 Carrier frequency = 16/PWM conversion period With a 20 MHz system clock (φ), the resolution, PWM conversion period, and carrier frequency are as shown in table 9.3.
5	—	1	R	Reserved Always read as 1 and cannot be modified.
4	—	0	R	Reserved Always read as 0 and cannot be modified.
3	RS3	0	R/W	Register Select
2	RS2	0	R/W	These bits select the PWM data register.
1	RS1	0	R/W	0xxx: No effect on operation
0	RS0	0	R/W	1000: PWDR8 selected 1001: PWDR9 selected 1010: PWDR10 selected 1011: PWDR11 selected 1100: PWDR12 selected 1101: PWDR13 selected 1110: PWDR14 selected 1111: PWDR15 selected

[Legend]

x: Don't care.

Table 9.2 Internal Clock Selection

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

Table 9.3 Resolution, PWM Conversion Period, and Carrier Frequency when φ = 20 MHz

Internal Clock Frequency	Resolution	PWM Conversion Period	Carrier Frequency
φ	50 ns	1.28 µs	1250 kHz
φ/2	100 ns	25.6 µs	625 kHz
φ/4	200 ns	51.2 µs	312.5 kHz
φ/8	400 ns	102.4 µs	156.3 kHz
φ/16	800 ns	204.8 µs	78.1 kHz

9.3.2 PWM Data Registers 15 to 8 (PWDR15 to PWDR8)

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

9.3.3 PWM Data Polarity Register B (PWDPRB)

PWDPR selects the PWM output phase.

- PWDPRB

Bit	Bit Name	Initial Value	R/W	Description
7	OS15	0	R/W	Output Select 15 to 8
6	OS14	0	R/W	These bits select the PWM output phase. Bits OS15 to OS8 correspond to outputs PW15 to PW8.
5	OS13	0	R/W	
4	OS12	0	R/W	0: PWM direct output (PWDR value corresponds to high width of output)
3	OS11	0	R/W	
2	OS10	0	R/W	
1	OS9	0	R/W	1: PWM inverted output (PWDR value corresponds to low width of output)
0	OS8	0	R/W	

9.3.4 PWM Output Enable Register B (PWOERB)

PWOER switches between PWM output and port output.

- PWOERB

Bit	Bit Name	Initial Value	R/W	Description
7	OE15	0	R/W	Output Enable 15 to 8
6	OE14	0	R/W	These bits, together with P2DDR, specify the P2n/PWm pin state. Bits OE15 to OE8 correspond to outputs PW15 to PW8.
5	OE13	0	R/W	
4	OE12	0	R/W	
3	OE11	0	R/W	P2nDDR OEn: Pin state
2	OE10	0	R/W	
1	OE9	0	R/W	0x: Port input
0	OE8	0	R/W	10: Port output or PWM 256/256 output
				11: PWM output (0 to 255/256 output)

[Legend]

x: Don't care

Note: n = 7 to 0

m = 15 to 8

To perform PWM 256/256 output when DDR = 1 and OE = 0, the corresponding pin should be set to port output.

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

9.3.5 Peripheral Clock Select Register (PCSR)

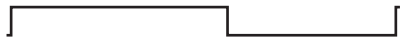
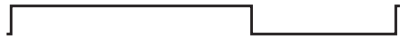
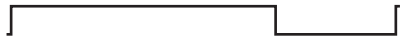
PCSR selects the PWM input clock.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R/W	See section 10.3.4, Peripheral Clock Select Register (PCSR).
6	—	0	R/W	
5	PWCKXB	0	R/W	
4	PWCKXA	0	R/W	
3	—	0	R/W	
2	PWCKB	0	R/W	PWM Clock Select B, A Together with bits PWCKE and PWCKS in PWSL, these bits select the internal clock input to the clock counter in the PWM. For details, see table 9.2.
1	PWCKA	0	R/W	
0	PWCKXC	0	R/W	See section 10.3.4, Peripheral Clock Select Register (PCSR).

9.4 Operation

The upper four bits of PWDR specify the duty cycle of the basic pulse as 0/16 to 15/16 with a resolution of 1/16. Table 9.4 shows the duty cycles of the basic pulse.

Table 9.4 Duty Cycle of Basic Pulse

Upper 4 Bits	Basic Pulse Waveform (Internal)
B'0000	H: 0 1 2 3 4 5 6 7 8 9 A B C D E F 0 L:
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 of PWDR specify the position of pulses added to the 16 basic pulses. An additional pulse adds a high period (when OS = 0) with a width equal to the resolution before the rising edge of a basic pulse. When the upper four bits of PWDR are B'0000, there is no rising edge of the basic pulse, but the timing for adding pulses is the same. Table 9.5 shows the positions of the additional pulses added to the basic pulses, and figure 9.2 shows an example of additional pulse timing.

Table 9.5 Position of Pulses Added to Basic Pulses

	Basic Pulse No.															
Lower 4 Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
B'0000																
B'0001																Yes
B'0010								Yes								Yes
B'0011								Yes				Yes				Yes
B'0100				Yes				Yes				Yes				Yes
B'0101				Yes				Yes				Yes		Yes		Yes
B'0110				Yes		Yes		Yes				Yes		Yes		Yes
B'0111				Yes		Yes		Yes		Yes		Yes		Yes		Yes
B'1000		Yes		Yes		Yes		Yes		Yes		Yes		Yes		Yes
B'1001		Yes		Yes		Yes		Yes		Yes		Yes		Yes	Yes	Yes
B'1010		Yes		Yes		Yes	Yes	Yes		Yes		Yes		Yes	Yes	Yes
B'1011		Yes		Yes		Yes	Yes	Yes		Yes	Yes	Yes		Yes	Yes	Yes
B'1100		Yes	Yes	Yes		Yes	Yes	Yes		Yes	Yes	Yes		Yes	Yes	Yes
B'1101		Yes	Yes	Yes		Yes	Yes	Yes		Yes	Yes	Yes	Yes	Yes	Yes	Yes
B'1110		Yes	Yes	Yes	Yes	Yes	Yes	Yes		Yes	Yes	Yes	Yes	Yes	Yes	Yes
B'1111		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

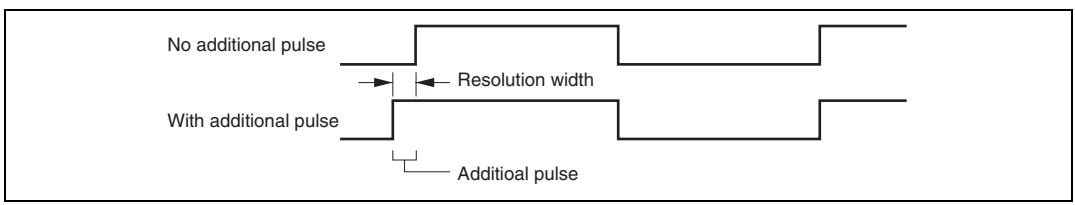


Figure 9.2 Example of Additional Pulse Timing (When Upper 4 Bits of PWDR = B'1000)

9.4.1 PWM Setting Example

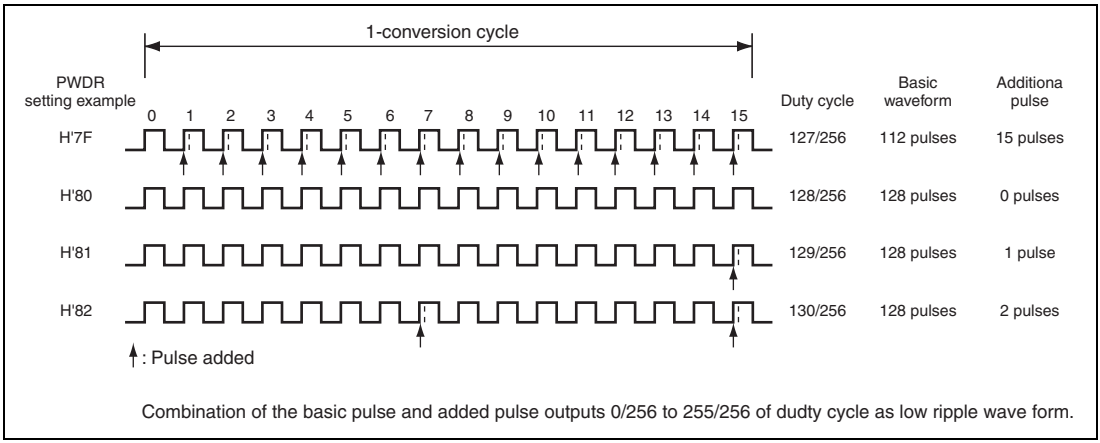


Figure 9.3 Example of PWM Setting

9.4.2 Diagram of PWM Used as D/A Converter

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

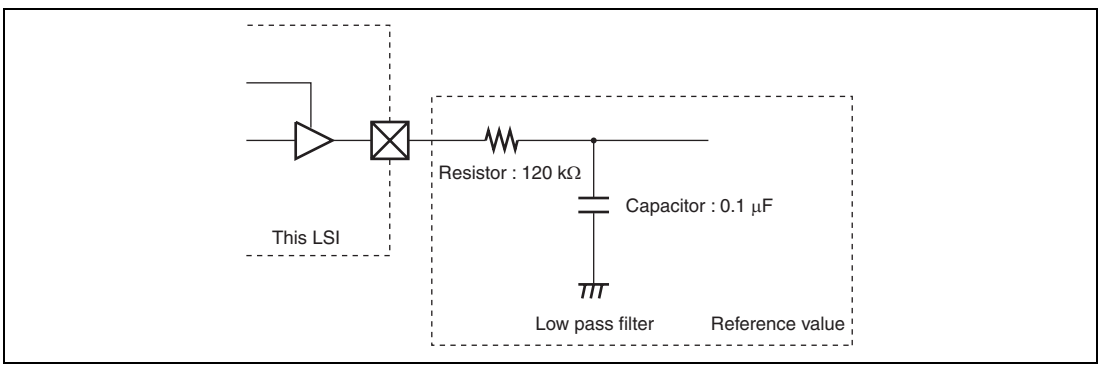


Figure 9.4 Example when PWM is Used as D/A Converter

9.5 Usage Notes

9.5.1 Module Stop Mode Setting

PWM operation can be enabled or disabled by the module stop control register. In the initial state, PWM operation is disabled. Access to PWM registers is enabled when module stop mode is cancelled. For details, see section 24, Power-Down Modes.

Section 10 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.

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

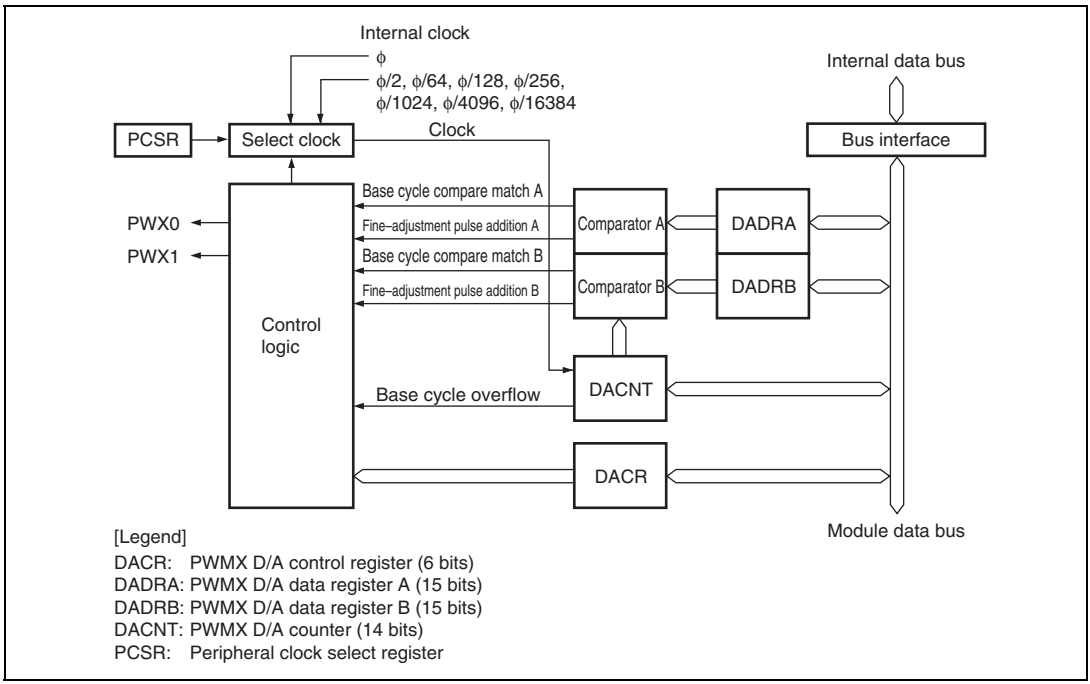


Figure 10.1 PWMX (D/A) Block Diagram

10.2 Input/Output Pins

Table 10.1 lists the PWMX (D/A) module input and output pins.

Table 10.1 Pin Configuration

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

10.3 Register Descriptions

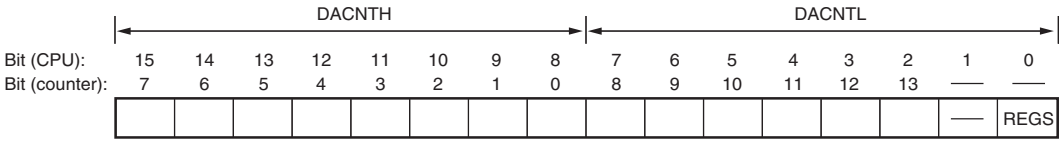
The PWMX (D/A) module has the following registers. The PWMX (D/A) registers are assigned to the same addresses with other registers. The registers are selected by the IICE bit in the serial timer control register (STCR). For details on the module stop control register, see section 24.1.3, Module Stop Control Register H, L, and A (MSTPCRH, MSTPCRL, MSTPCRA).

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

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

10.3.1 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 10.4, 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

10.3.2 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 10.4, 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

10.3.3 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.

10.3.4 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 10.2.
3	—	0	R/W	Reserved The initial value should not be changed.
2	PWCKB	0	R/W	PWM clock select B, A
1	PWCKA	0	R/W	See section 9.3.5, Peripheral Clock Select Register (PCSR).
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 10.2.

Table 10.2 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

10.4 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 10.3 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.

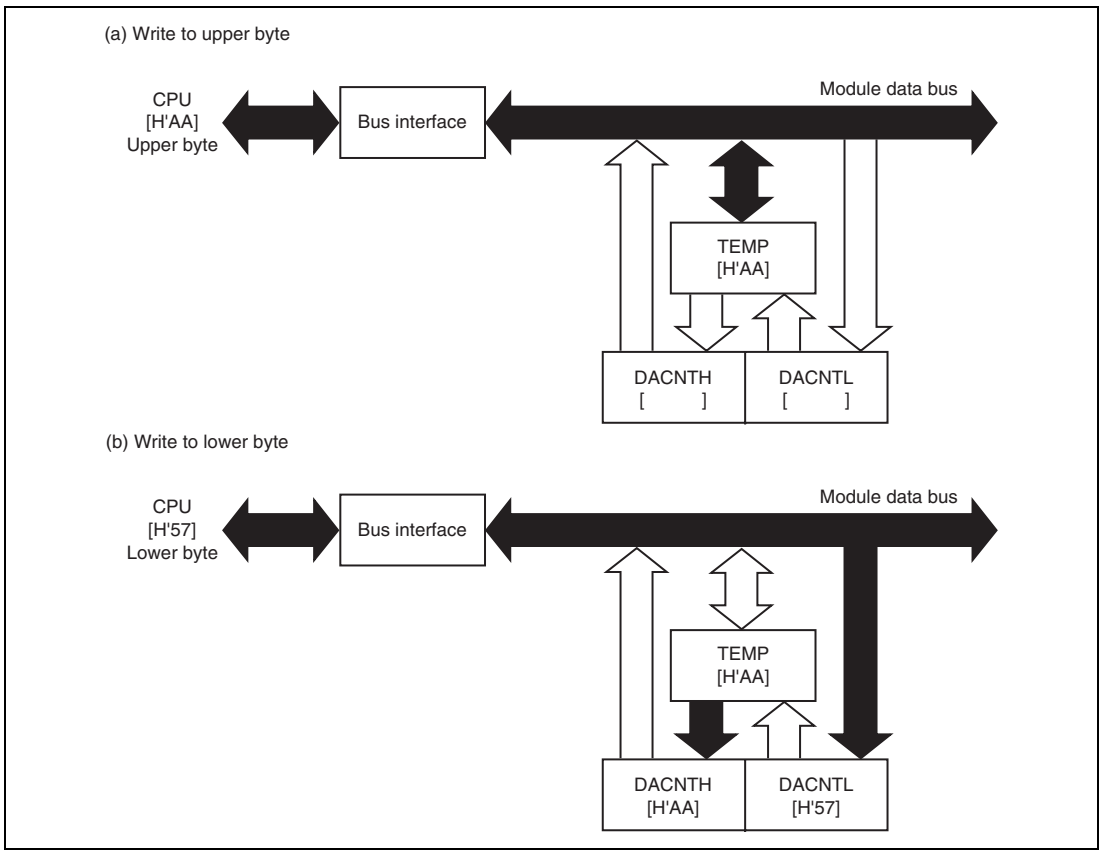


Figure 10.2 (1) DACNT Access Operation (1) [CPU → DACNT(H'AA57) Writing]

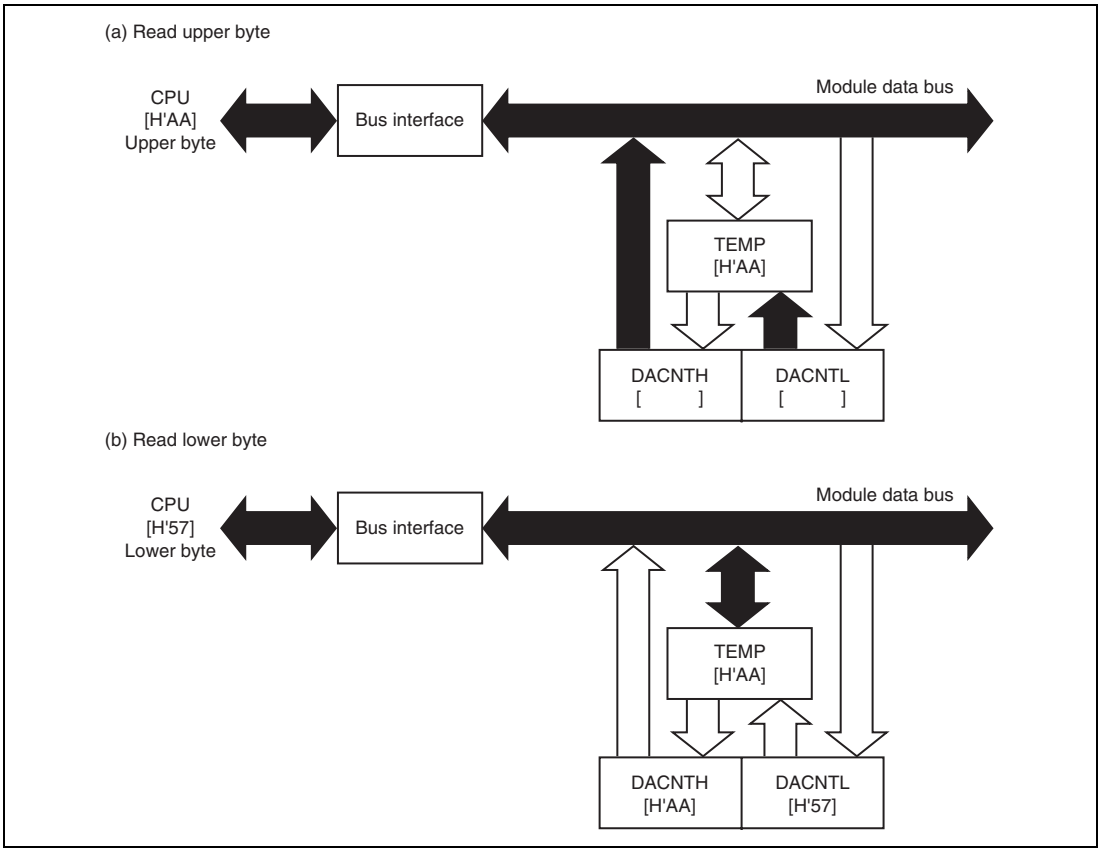


Figure 10.2 (2) DACNT Access Operation (2) [DACNT → CPU(H'AA57) Reading]

10.5 Operation

A PWM waveform like the one shown in figure 10.3 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 10.4 and 10.5 show the types of waveform output available.

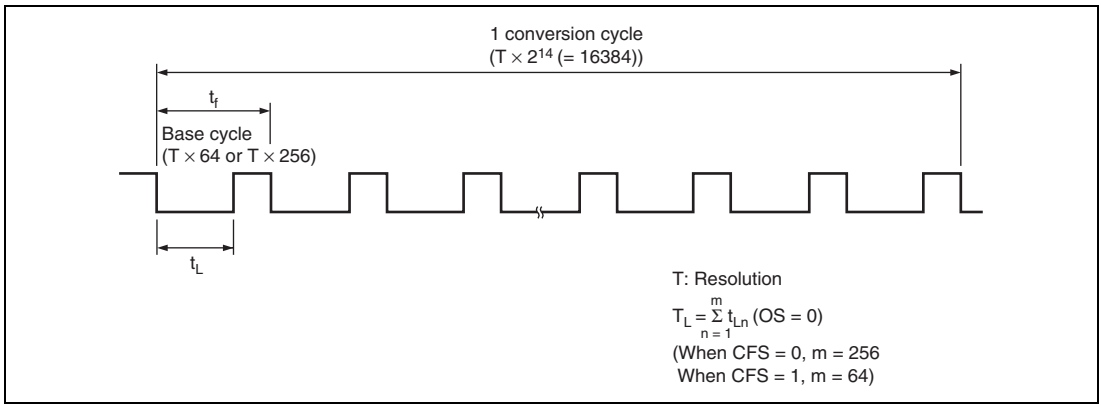


Figure 10.3 PWMX (D/A) Operation

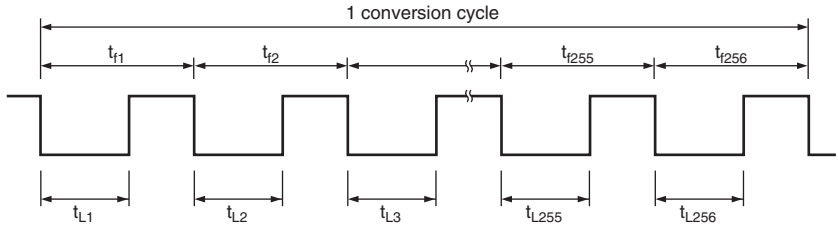
Table 10.4 summarizes the relationships between the CKS and CFS bit settings and the resolution, base cycle, and conversion cycle. The PWM output remains fixed unless DA13 to DA0 in DADR contain at least a certain minimum value. The relationship between the OS bit and the output waveform is shown in figures 10.4 and 10.5.

Table 10.4 Settings and Operation (Examples when ϕ = 20 MHz)

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

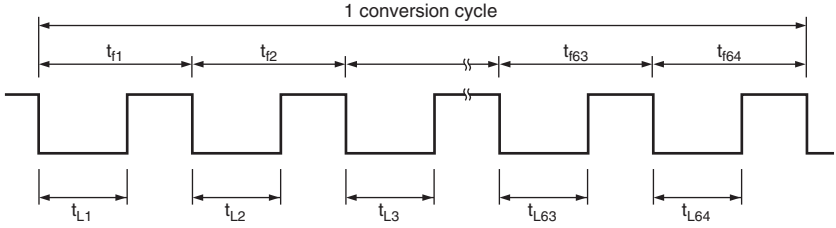
PCSR									Fixed DADR Bits					
PWCKX0 PWCKX1				Resolution T			Conversion Cycle	TL/TH	Accuracy (Bits)	Bit Data				Conversion Cycle*
C	B	A	CKS	(µs)	CFS	Base Cycle		(OS = 0/OS = 1)		DA3	DA2	DA1	DA0	
0	1	1	1	12.8	0	819.2 (µs) /1.2kHz	209.7 (ms)	Always low/high output	14					209.7 ms
								DA13 to 0 = 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	2.03 (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	496.48	0	52.4 (ms) /19.1Hz	8.13 (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.



$t_{f1} = t_{f2} = t_{f3} = \cdots = t_{f255} = t_{f256} = T \times 64$
$t_{L1} + t_{L2} + t_{L3} + \cdots + t_{L255} + t_{L256} = T_L$

a. CFS = 0 [base cycle = resolution (T) × 64]



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

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

Figure 10.4 Output Waveform (OS = 0, DADR corresponds to T_L)

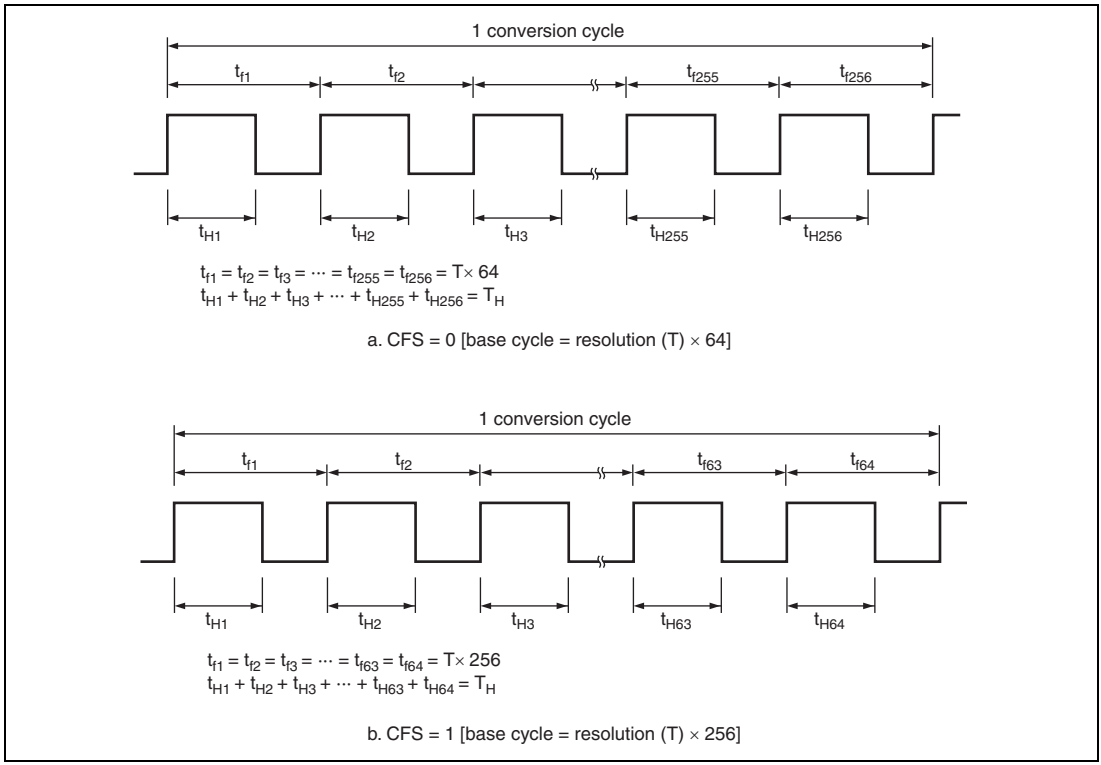


Figure 10.5 Output Waveform (OS = 1, DADR corresponds to T_H)

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 10.6.

Table 10.5 lists the locations of the additional pulses.

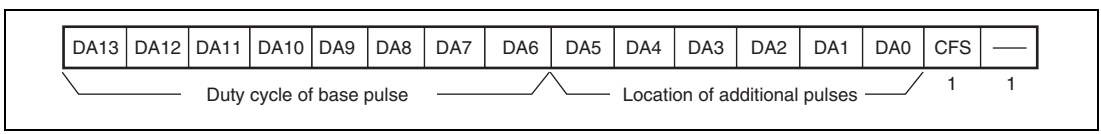


Figure 10.6 D/A Data Register Configuration when CFS = 1

In this example, DADR = H'0207 (B'0000 0010 0000 0111). The output waveform is shown in figure 10.7. 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 10.5. Thus, an additional pulse of 1/256 × (T) is to be added to the base pulse.

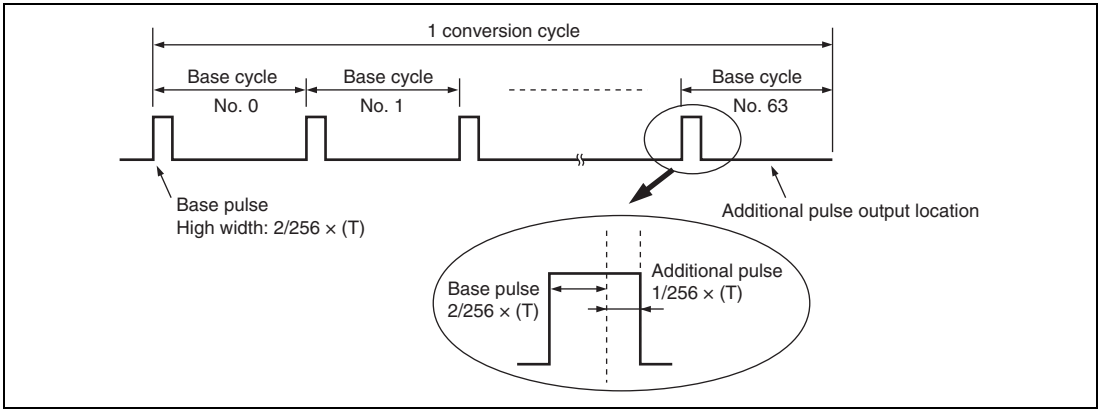
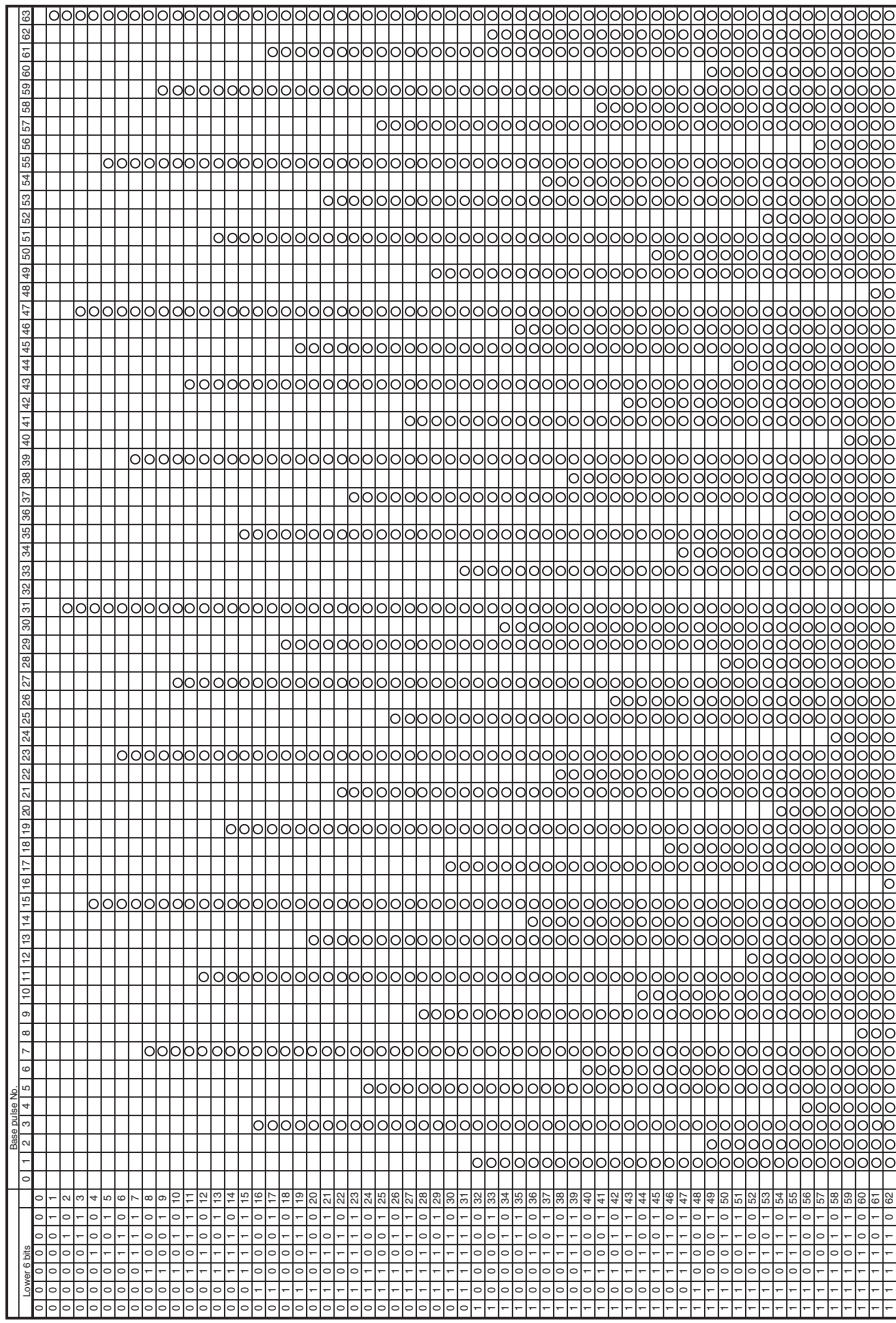


Figure 10.7 Output Waveform when DADR = H'0207 (OS = 1)

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



10.6 Usage Notes

10.6.1 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 24, Power-Down Modes.

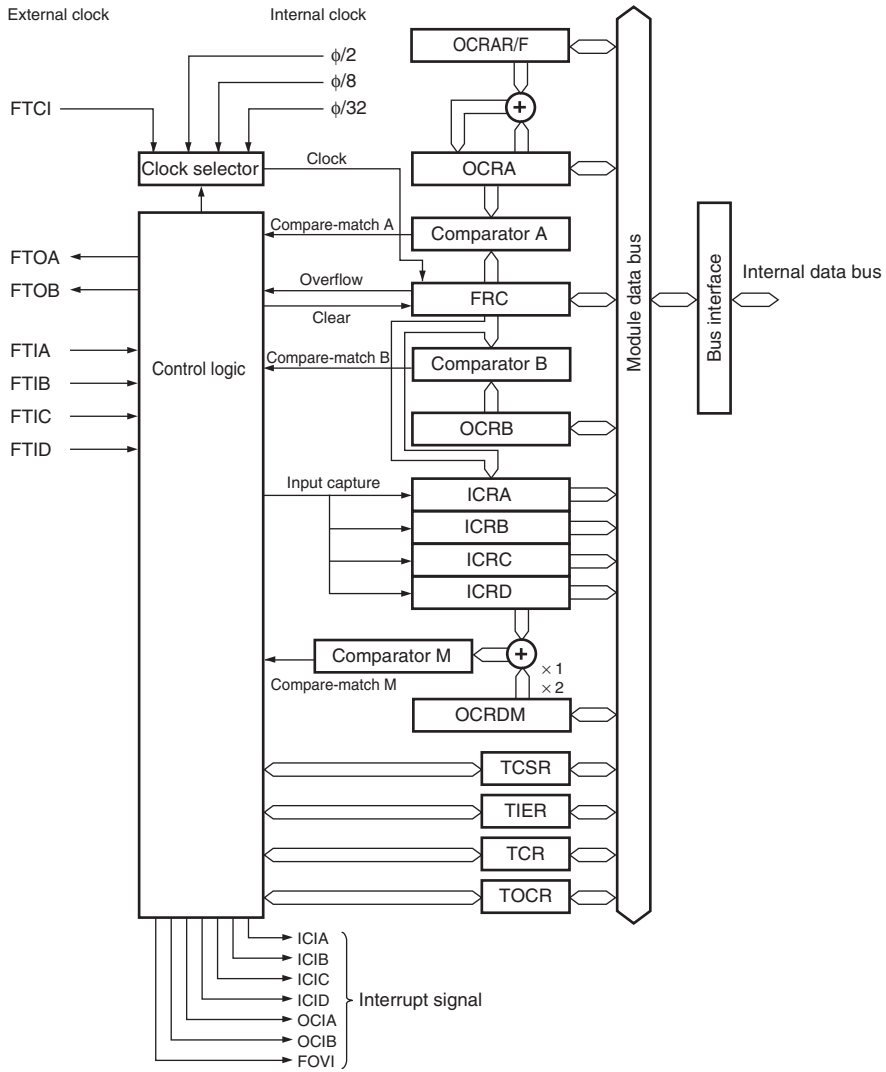
Section 11 16-Bit Free-Running Timer (FRT)

This LSI has an on-chip 16-bit free-running timer (FRT). The FRT operates on the basis of the 16-bit free-running counter (FRC), and outputs two independent waveforms, and measures the input pulse width and external clock periods.

11.1 Features

- Selection of four clock sources
  - One of the three internal clocks (φ/2, φ/8, or φ/32), or an external clock input can be selected (enabling use as an external event counter).
- Two independent comparators
  - Two independent waveforms can be output.
- Four independent input capture channels
  - The rising or falling edge can be selected.
  - Buffer modes can be specified.
- Counter clearing
  - The free-running counters can be cleared on compare-match A.
- Seven independent interrupts
  - Two compare-match interrupts, four input capture interrupts, and one overflow interrupt can be requested independently.
- Special functions provided by automatic addition function
  - The contents of OCRAR and OCRAF can be added to the contents of OCRA automatically, enabling a periodic waveform to be generated without software intervention. The contents of ICRD can be added automatically to the contents of OCRDM × 2, enabling input capture operations in this interval to be restricted.

Figure 11.1 shows a block diagram of the FRT.



[Legend]

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

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

11.2 Input/Output Pins

Table 11.1 lists the FRT input and output pins.

Table 11.1 Pin Configuration

Name	Abbreviation	I/O	Function
Counter clock input pin	FTCI	Input	FRC counter clock input
Output compare A output pin	FTOA	Output	Output compare A output
Output compare B output pin	FTOB	Output	Output compare B output
Input capture A input pin	FTIA	Input	Input capture A input
Input capture B input pin	FTIB	Input	Input capture B input
Input capture C input pin	FTIC	Input	Input capture C input
Input capture D input pin	FTID	Input	Input capture D input

11.3 Register Descriptions

The FRT has the following registers.

- Free-running counter (FRC)
- Output compare register A (OCRA)
- Output compare register B (OCRB)
- Input capture register A (ICRA)
- Input capture register B (ICRB)
- Input capture register C (ICRC)
- Input capture register D (ICRD)
- Output compare register AR (OCRAR)
- Output compare register AF (OCRAF)
- Output compare register DM (OCRDM)
- Timer interrupt enable register (TIER)
- Timer control/status register (TCSR)
- Timer control register (TCR)
- Timer output compare control register (TOCR)

Note: OCRA and OCRB share the same address. Register selection is controlled by the OCRS bit in TOCR. ICRA, ICRB, and ICRC share the same addresses with OCRAR, OCRAF, and OCRDM. Register selection is controlled by the ICRS bit in TOCR.

11.3.1 Free-Running Counter (FRC)

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

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

The FRT has two output compare registers, OCRA and OCRB, each of which is a 16-bit readable/writable register whose contents are continually compared with the value in FRC. When a match is detected (compare-match), the corresponding output compare flag (OCFA or OCFB) is set to 1 in TCSR. If the OEA or OEB bit in TOCR is set to 1, when the OCR and FRC values match, the output level selected by the OLVLA or OLVLB bit in TOCR is output at the output compare output pin (FTOA or FTOB). Following a reset, the FTOA and FTOB output levels are 0 until the first compare-match. OCR should always be accessed in 16-bit units; cannot be accessed in 8-bit units. OCR is initialized to H'FFFF.

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

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

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

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

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

11.3.4 Output Compare Registers AR and AF (OCRAR and OCRAF)

OCRAR and OCRAF are 16-bit readable/writable registers. When the OCRAMS bit in TOCR is set to 1, the operation of OCRA is changed to include the use of OCRAR and OCRAF. The contents of OCRAR and OCRAF are automatically added alternately to OCRA, and the result is written to OCRA. The write operation is performed on the occurrence of compare-match A. In the 1st compare-match A after setting the OCRAMS bit to 1, OCRAF is added. The operation due to compare-match A varies according to whether the compare-match follows addition of OCRAR or OCRAF. The value of the OLVLA bit in TOCR is ignored, and 1 is output on a compare-match A following addition of OCRAF, while 0 is output on a compare-match A following addition of OCRAR.

When using the OCRA automatic addition function, do not select internal clock $\phi/2$ as the FRC input clock together with a set value of H'0001 or less for OCRAR (or OCRAF).

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

11.3.5 Output Compare Register DM (OCRDM)

OCRDM is a 16-bit readable/writable register in which the upper eight bits are fixed at H'00. When the ICRDMS bit in TOCR is set to 1 and the contents of OCRDM are other than H'0000, the operation of ICRD is changed to include the use of OCRDM. The point at which input capture D occurs is taken as the start of a mask interval. Next, twice the contents of OCRDM is added to the contents of ICRD, and the result is compared with the FRC value. The point at which the values match is taken as the end of the mask interval. New input capture D events are disabled during the mask interval. A mask interval is not generated when the contents of OCRDM are H'0000 while the ICRDMS bit is set to 1.

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

11.3.6 Timer Interrupt Enable Register (TIER)

TIER enables and disables interrupt requests.

Bit	Bit Name	Initial Value	R/W	Description
7	ICIAE	0	R/W	Input Capture Interrupt A Enable Selects whether to enable input capture interrupt A request (ICIA) when input capture flag A (ICFA) in TCSR is set to 1. 0: ICIA requested by ICFA is disabled 1: ICIA requested by ICFA is enabled
6	ICIBE	0	R/W	Input Capture Interrupt B Enable Selects whether to enable input capture interrupt B request (ICIB) when input capture flag B (ICFB) in TCSR is set to 1. 0: ICIB requested by ICFB is disabled 1: ICIB requested by ICFB is enabled
5	ICICE	0	R/W	Input Capture Interrupt C Enable Selects whether to enable input capture interrupt C request (ICIC) when input capture flag C (ICFC) in TCSR is set to 1. 0: ICIC requested by ICFC is disabled 1: ICIC requested by ICFC is enabled
4	ICIDE	0	R/W	Input Capture Interrupt D Enable Selects whether to enable input capture interrupt D request (ICID) when input capture flag D (ICFD) in TCSR is set to 1. 0: ICID requested by ICFD is disabled 1: ICID requested by ICFD is enabled
3	OCIAE	0	R/W	Output Compare Interrupt A Enable Selects whether to enable output compare interrupt A request (OCIA) when output compare flag A (OCFA) in TCSR is set to 1. 0: OCIA requested by OCFA is disabled 1: OCIA requested by OCFA is enabled

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

11.3.7 Timer Control/Status Register (TCSR)

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

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

Bit	Bit Name	Initial Value	R/W	Description
6	ICFB	0	R/(W)*	Input Capture Flag B This status flag indicates that the FRC value has been transferred to ICRB by means of an input capture signal. When BUFEB = 1, ICFB indicates that the old ICRB value has been moved into ICRD and the new FRC value has been transferred to ICRB. [Setting condition] When an input capture signal causes the FRC value to be transferred to ICRB [Clearing condition] Read ICFB when ICFB = 1, then write 0 to ICFB
5	ICFC	0	R/(W)*	Input Capture Flag C This status flag indicates that the FRC value has been transferred to ICRC by means of an input capture signal. When BUFEA = 1, on occurrence of an input capture signal specified by the IEDGC bit at the FTIC input pin, ICFC is set but data is not transferred to ICRC. In buffer operation, ICFC can be used as an external interrupt signal by setting the ICICE bit to 1. [Setting condition] When an input capture signal is received [Clearing condition] Read ICFC when ICFC = 1, then write 0 to ICFC
4	ICFD	0	R/(W)*	Input Capture Flag D This status flag indicates that the FRC value has been transferred to ICRD by means of an input capture signal. When BUFEB = 1, on occurrence of an input capture signal specified by the IEDGD bit at the FTID input pin, ICFD is set but data is not transferred to ICRD. In buffer operation, ICFD can be used as an external interrupt signal by setting the ICIDE bit to 1. [Setting condition] When an input capture signal is received [Clearing condition] Read ICFD when ICFD = 1, then write 0 to ICFD

Bit	Bit Name	Initial Value	R/W	Description
3	OCFA	0	R/(W)*	Output Compare Flag A This status flag indicates that the FRC value matches the OCRA value. [Setting condition] When FRC = OCRA [Clearing condition] Read OCFA when OCFA = 1, then write 0 to OCFA
2	OCFB	0	R/(W)*	Output Compare Flag B This status flag indicates that the FRC value matches the OCRB value. [Setting condition] When FRC = OCRB [Clearing condition] Read OCFB when OCFB = 1, then write 0 to OCFB
1	OVF	0	R/(W)*	Overflow Flag This status flag indicates that the FRC has overflowed. [Setting condition] When FRC overflows (changes from H'FFFF to H'0000) [Clearing condition] Read OVF when OVF = 1, then write 0 to OVF
0	CCLRA	0	R/W	Counter Clear A This bit selects whether the FRC is to be cleared at compare-match A (when the FRC and OCRA values match). 0: FRC clearing is disabled 1: FRC is cleared at compare-match A

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

11.3.8 Timer Control Register (TCR)

TCR selects the rising or falling edge of the input capture signals, enables the input capture buffer mode, and selects the FRC clock source.

Bit	Bit Name	Initial Value	R/W	Description
7	IEDGA	0	R/W	Input Edge Select A Selects the rising or falling edge of the input capture A signal (FTIA). 0: Capture on the falling edge of FTIA 1: Capture on the rising edge of FTIA
6	IEDGB	0	R/W	Input Edge Select B Selects the rising or falling edge of the input capture B signal (FTIB). 0: Capture on the falling edge of FTIB 1: Capture on the rising edge of FTIB
5	IEDGC	0	R/W	Input Edge Select C Selects the rising or falling edge of the input capture C signal (FTIC). 0: Capture on the falling edge of FTIC 1: Capture on the rising edge of FTIC
4	IEDGD	0	R/W	Input Edge Select D Selects the rising or falling edge of the input capture D signal (FTID). 0: Capture on the falling edge of FTID 1: Capture on the rising edge of FTID
3	BUFEA	0	R/W	Buffer Enable A Selects whether ICRC is to be used as a buffer register for ICRA. 0: ICRC is not used as a buffer register for ICRA 1: ICRC is used as a buffer register for ICRA
2	BUFEB	0	R/W	Buffer Enable B Selects whether ICRD is to be used as a buffer register for ICRB. 0: ICRD is not used as a buffer register for ICRB 1: ICRD is used as a buffer register for ICRB

Bit	Bit Name	Initial Value	R/W	Description
1	CKS1	0	R/W	Clock Select 1, 0
0	CKS0	0		Select clock source for FRC.
				00: ϕ/2 internal clock source
				01: ϕ/8 internal clock source
				10: ϕ/32 internal clock source
				11: External clock source (counting at FTCI rising edge)

11.3.9 Timer Output Compare Control Register (TOCR)

TOCR enables output from the output compare pins, selects the output levels, switches access between output compare registers A and B, controls the ICRD and OCRA operating modes, and switches access to input capture registers A, B, and C.

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

Bit	Bit Name	Initial Value	R/W	Description
4	OCRS	0	R/W	Output Compare Register Select OCRA and OCRB share the same address. The OCRS determines which register is selected when the shared address is read from or written to. The operation of OCRA or OCRB is not affected. 0: OCRA is selected 1: OCRB is selected
3	OEA	0	R/W	Output Enable A Enables or disables output of the output compare A output pin (FTOA). 0: Output compare A output is disabled 1: Output compare A output is enabled
2	OEB	0	R/W	Output Enable B Enables or disables output of the output compare B output pin (FTOB). 0: Output compare B output is disabled 1: Output compare B output is enabled
1	OLVLA	0	R/W	Output Level A Selects the level to be output at the output compare A output pin (FTOA) in response to compare-match A (signal indicating a match between the FRC and OCRA values). When the OCRAMS bit is 1, this bit is ignored. 0: 0 is output at compare-match A 1: 1 is output at compare-match A
0	OLVLB	0	R/W	Output Level B Selects the level to be output at the output compare B output pin (FTOB) in response to compare-match B (signal indicating a match between the FRC and OCRB values). 0: 0 is output at compare-match B 1: 1 is output at compare-match B

11.4 Operation

11.4.1 Pulse Output

Figure 11.2 shows an example of 50%-duty pulses output with an arbitrary phase difference. When a compare match occurs while the CCLRA bit in TCSR is set to 1, the OLVLA and OLVLB bits are inverted by software.

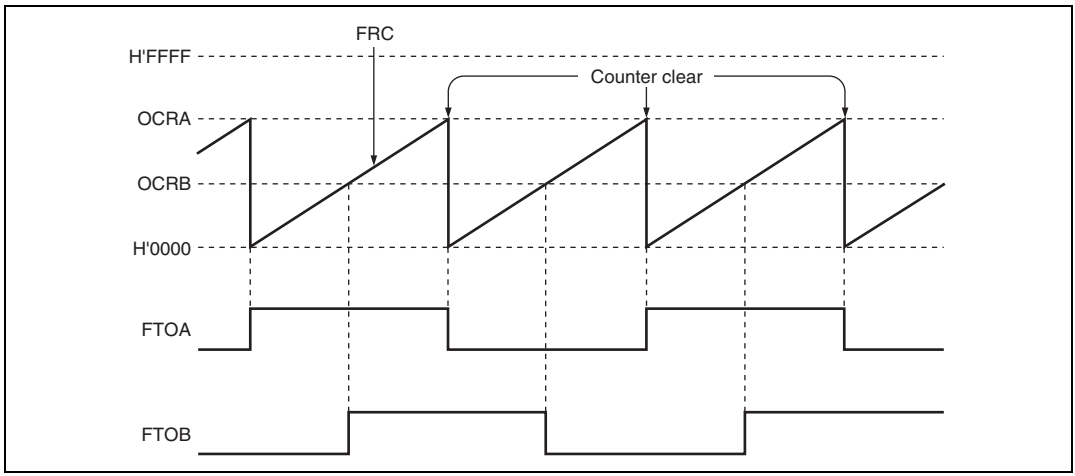


Figure 11.2 Example of Pulse Output

11.5 Operation Timing

11.5.1 FRC Increment Timing

Figure 11.3 shows the FRC increment timing with an internal clock source. Figure 11.4 shows the increment timing with an external clock source. The pulse width of the external clock signal must be at least 1.5 system clocks (ϕ). The counter will not increment correctly if the pulse width is shorter than 1.5 system clocks (ϕ).

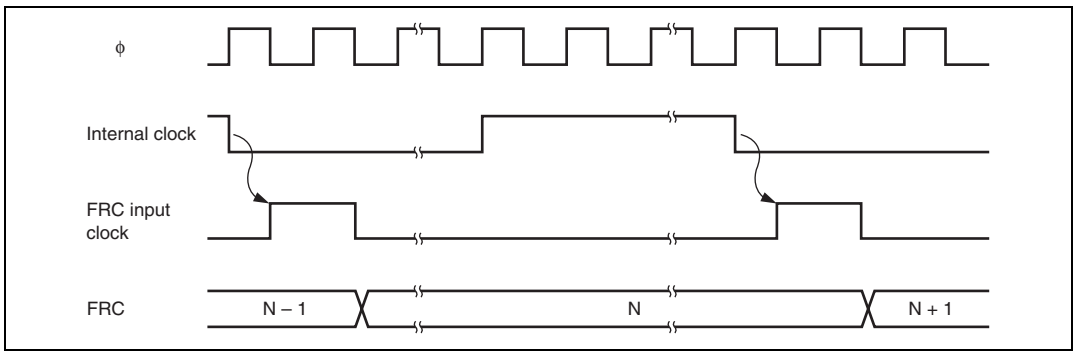


Figure 11.3 Increment Timing with Internal Clock Source

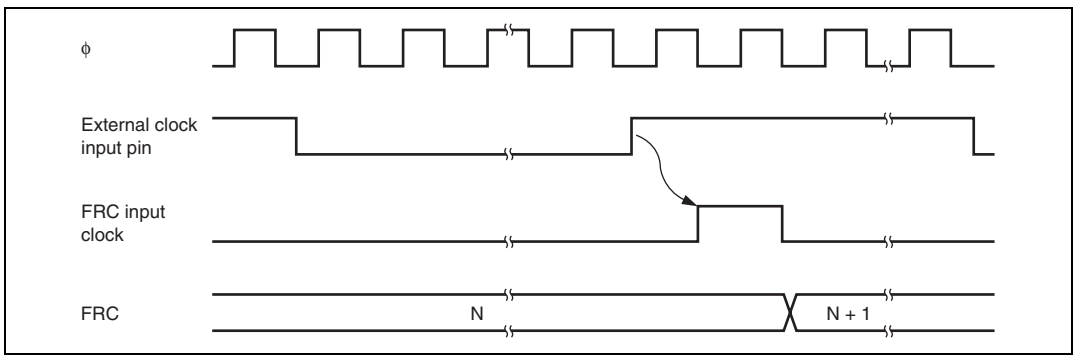


Figure 11.4 Increment Timing with External Clock Source

11.5.2 Output Compare Output Timing

A compare-match signal occurs at the last state when the FRC and OCR values match (at the timing when the FRC updates the counter value). When a compare-match signal occurs, the level selected by the OLVL bit in TOCR is output at the output compare output pin (FTOA or FTOB). Figure 11.5 shows the timing of this operation for compare-match A.

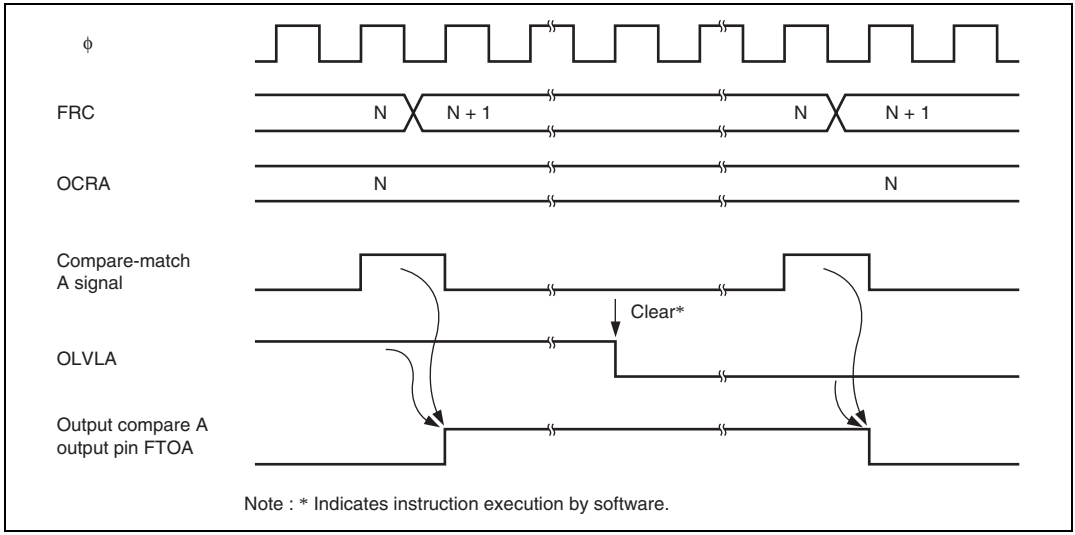


Figure 11.5 Timing of Output Compare A Output

11.5.3 FRC Clear Timing

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

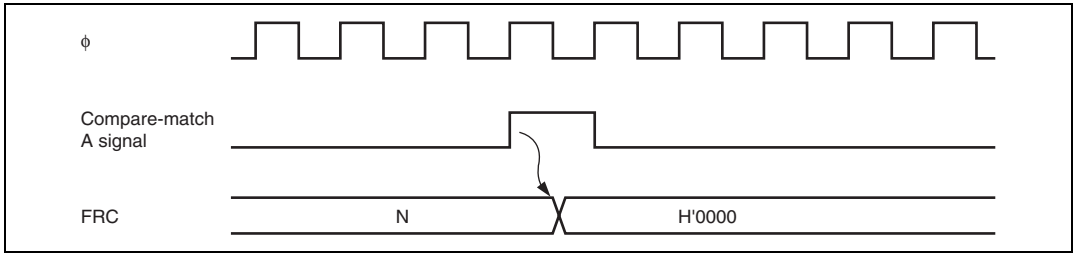


Figure 11.6 Clearing of FRC by Compare-Match A Signal

11.5.4 Input Capture Input Timing

The rising or falling edge can be selected for the input capture input timing by the IEDGA to IEDGD bits in TCR. Figure 11.7 shows the usual input capture timing when the rising edge is selected.

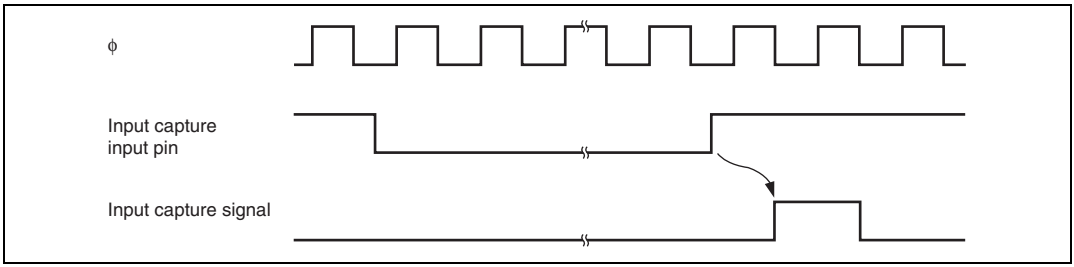


Figure 11.7 Input Capture Input Signal Timing (Usual Case)

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

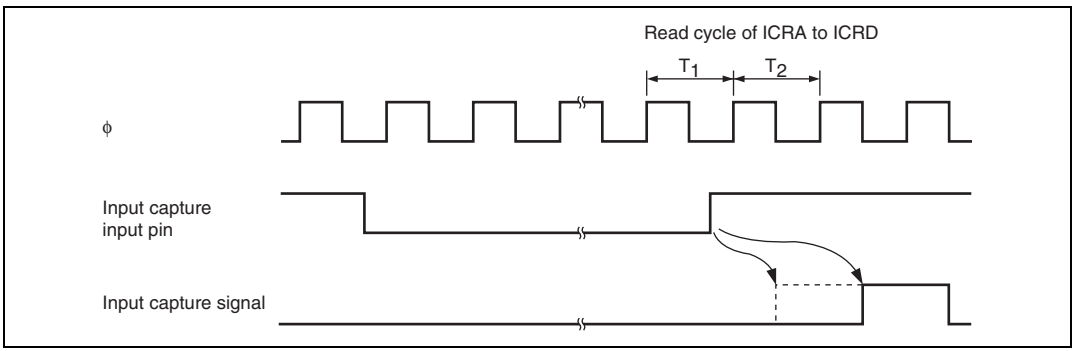


Figure 11.8 Input Capture Input Signal Timing (When ICRA to ICRD is Read)

11.5.5 Buffered Input Capture Input Timing

ICRC and ICRD can operate as buffers for ICRA and ICRB, respectively. Figure 11.9 shows how input capture operates when ICRC is used as ICRA's buffer register (BUFEA = 1) and IEDGA and IEDGC are set to different values (IEDGA = 0 and IEDGC = 1, or IEDGA = 1 and IEDGC = 0), so that input capture is performed on both the rising and falling edges of FTIA.

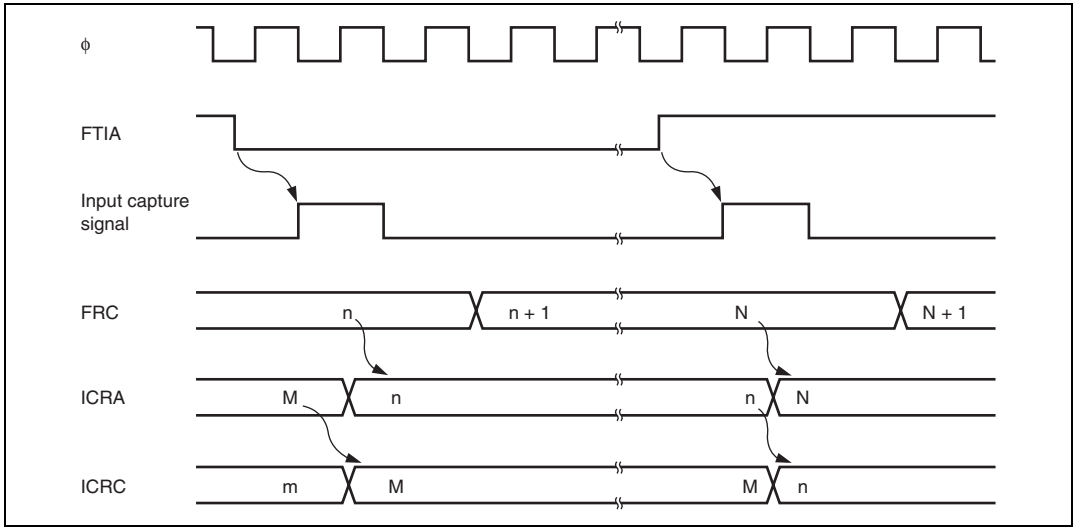


Figure 11.9 Buffered Input Capture Timing

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

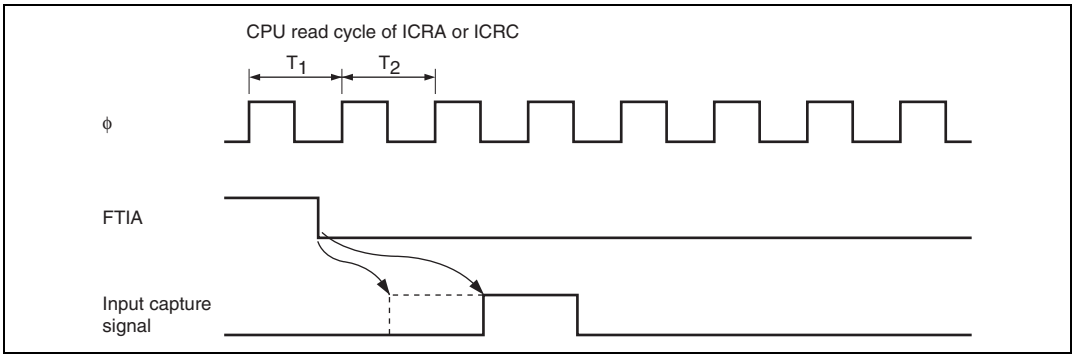


Figure 11.10 Buffered Input Capture Timing (BUFEA = 1)

11.5.6 Timing of Input Capture Flag (ICF) Setting

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

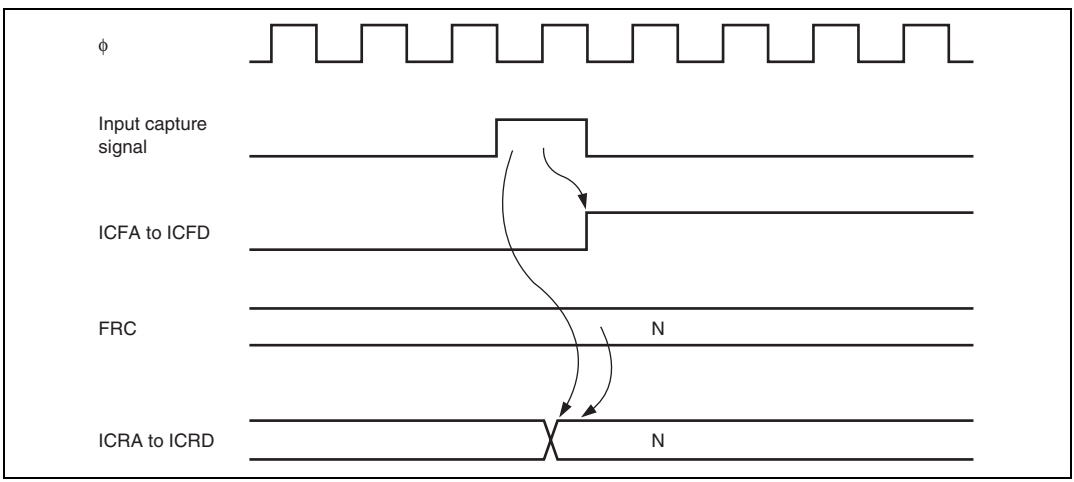


Figure 11.11 Timing of Input Capture Flag (ICFA, ICFB, ICFC, or ICFD) Setting

11.5.7 Timing of Output Compare Flag (OCF) setting

The output compare flag, OCFA or OCFB, is set to 1 by a compare-match signal generated when the FRC value matches the OCRA or OCRB value. This compare-match signal is generated at the last state in which the two values match, just before FRC increments to a new value. When the FRC and OCRA or OCRB value match, the compare-match signal is not generated until the next cycle of the clock source. Figure 11.12 shows the timing of setting the OCFA or OCFB flag.

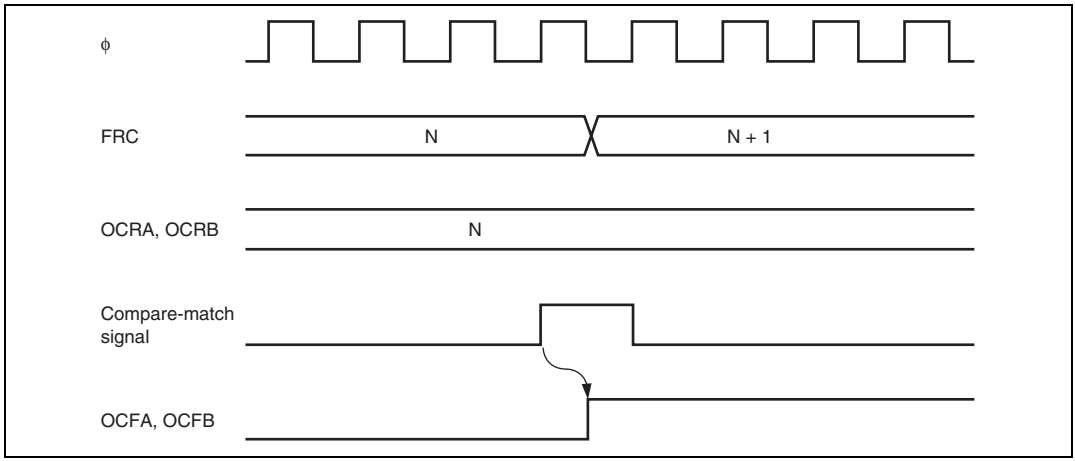


Figure 11.12 Timing of Output Compare Flag (OCFA or OCFB) Setting

11.5.8 Timing of FRC Overflow Flag Setting

The FRC overflow flag (OVF) is set to 1 when FRC overflows (changes from H'FFFF to H'0000). Figure 11.13 shows the timing of setting the OVF flag.

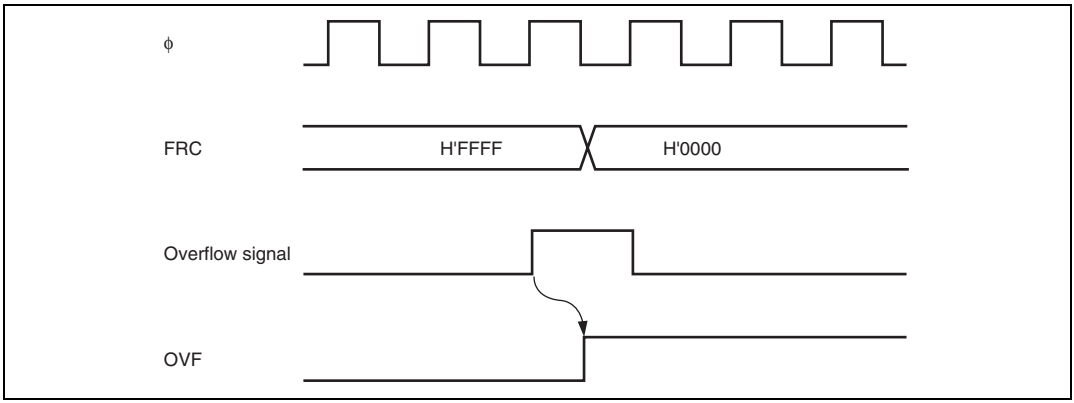


Figure 11.13 Timing of Overflow Flag (OVF) Setting

11.5.9 Automatic Addition Timing

When the OCRAMS bit in TOCR is set to 1, the contents of OCRAR and OCRAF are automatically added to OCRA alternately, and when an OCRA compare-match occurs a write to OCRA is performed. Figure 11.14 shows the OCRA write timing.

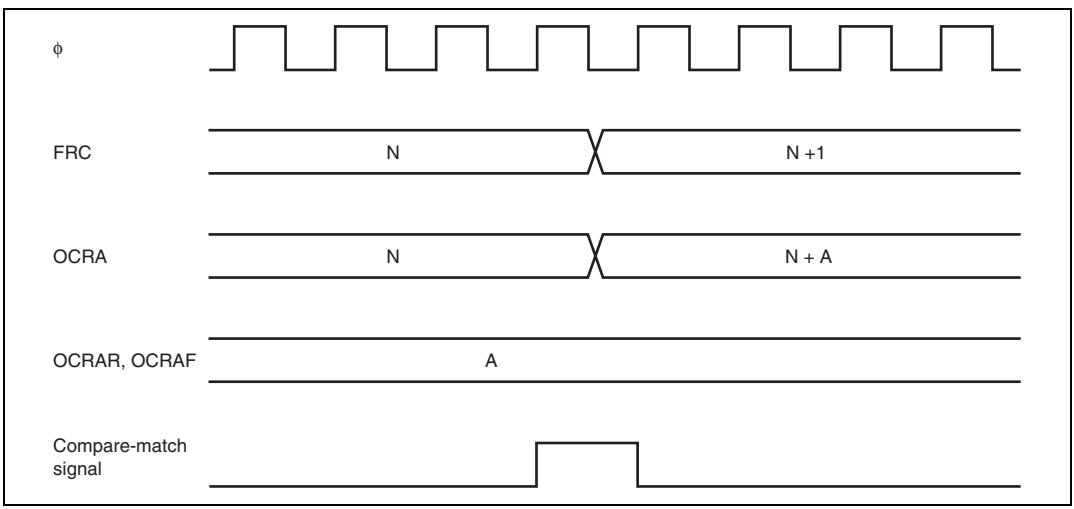


Figure 11.14 OCRA Automatic Addition Timing

11.5.10 Mask Signal Generation Timing

When the ICRDMS bit in TOCR is set to 1 and the contents of OCRDM are other than H'0000, a signal that masks the ICRD input capture signal is generated. The mask signal is set by the input capture signal. The mask signal is cleared by the sum of the ICRD contents and twice the OCRDM contents, and an FRC compare-match. Figure 11.15 shows the timing of setting the mask signal. Figure 11.16 shows the timing of clearing the mask signal.

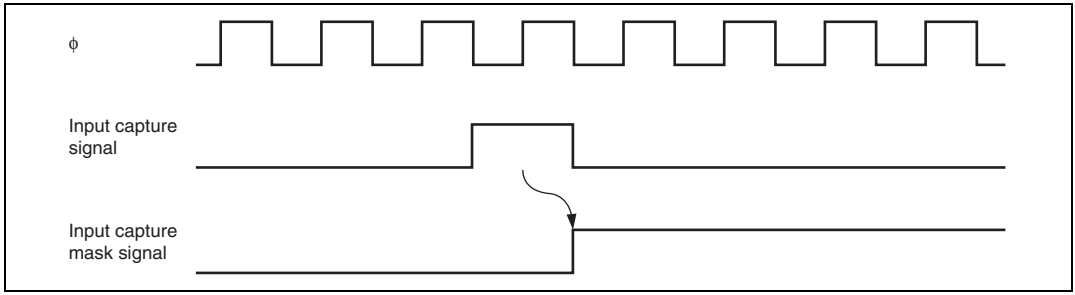


Figure 11.15 Timing of Input Capture Mask Signal Setting

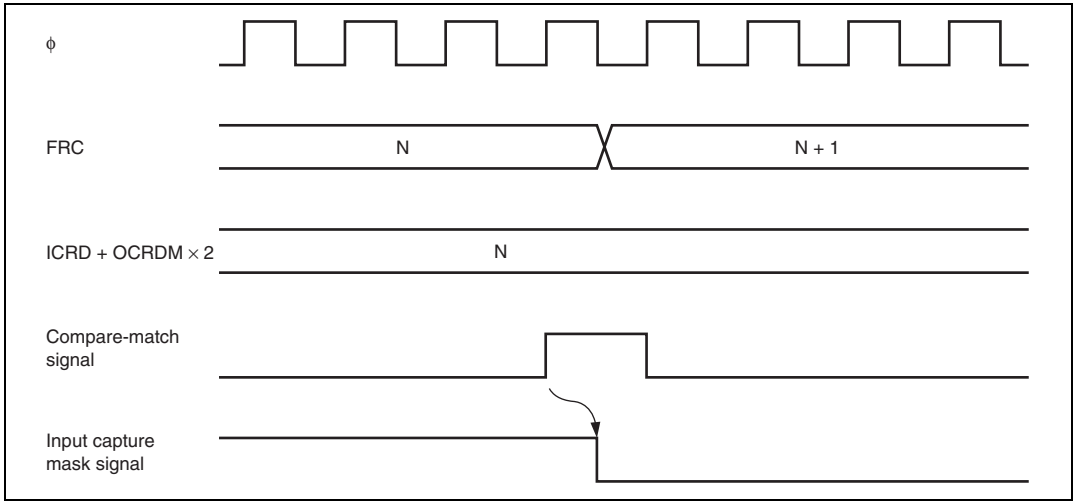


Figure 11.16 Timing of Input Capture Mask Signal Clearing

11.6 Interrupt Sources

The free-running timer can request seven interrupts: ICIA to ICID, OCIA, OCIB, and FOVI. Each interrupt can be enabled or disabled by an enable bit in TIER. Independent signals are sent to the interrupt controller for each interrupt. Table 11.2 lists the sources and priorities of these interrupts.

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

Table 11.2 FRT Interrupt Sources

Interrupt	Interrupt Source	Interrupt Flag	DTC Activation	Priority
ICIA	Input capture of ICRA	ICFA	Enable	High
ICIB	Input capture of ICRB	ICFB	Enable	
ICIC	Input capture of ICRC	ICFC	Disable	
ICID	Input capture of ICRD	ICFD	Disable	
OCIA	Compare match of OCRA	OCFA	Enable	
OCIB	Compare match of OCRB	OCFB	Enable	
FOVI	Overflow of FRC	OVF	Disable	Low

11.7 Usage Notes

11.7.1 Conflict between FRC Write and Clear

If an internal counter clear signal is generated during the state after an FRC write cycle, the clear signal takes priority and the write is not performed. Figure 11.17 shows the timing for this type of conflict.

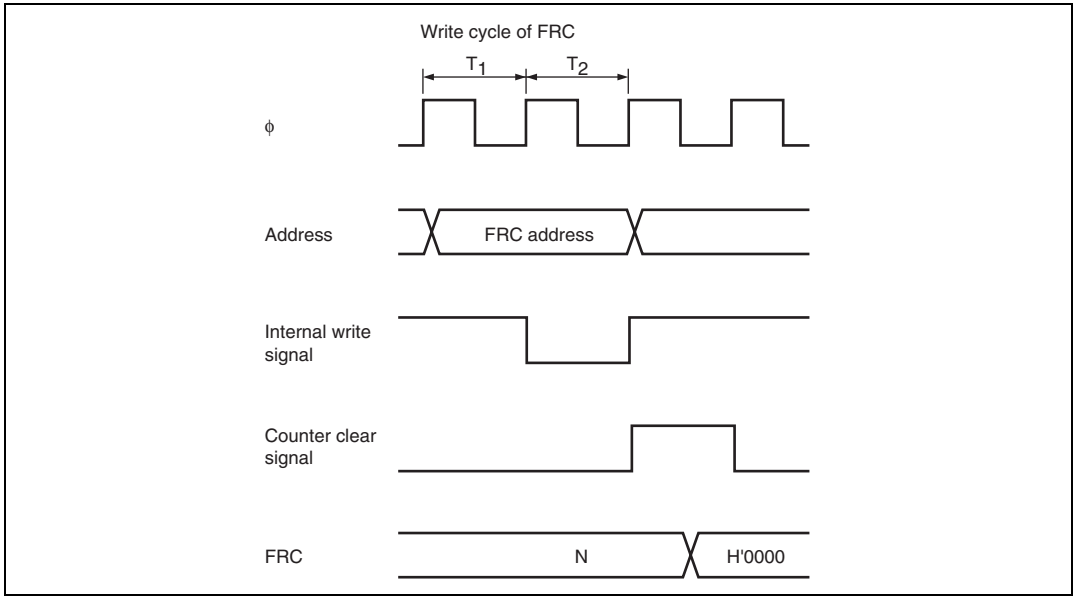


Figure 11.17 Conflict between FRC Write and Clear

11.7.2 Conflict between FRC Write and Increment

If an FRC increment pulse is generated during the state after an FRC write cycle, the write takes priority and FRC is not incremented. Figure 11.18 shows the timing for this type of conflict.

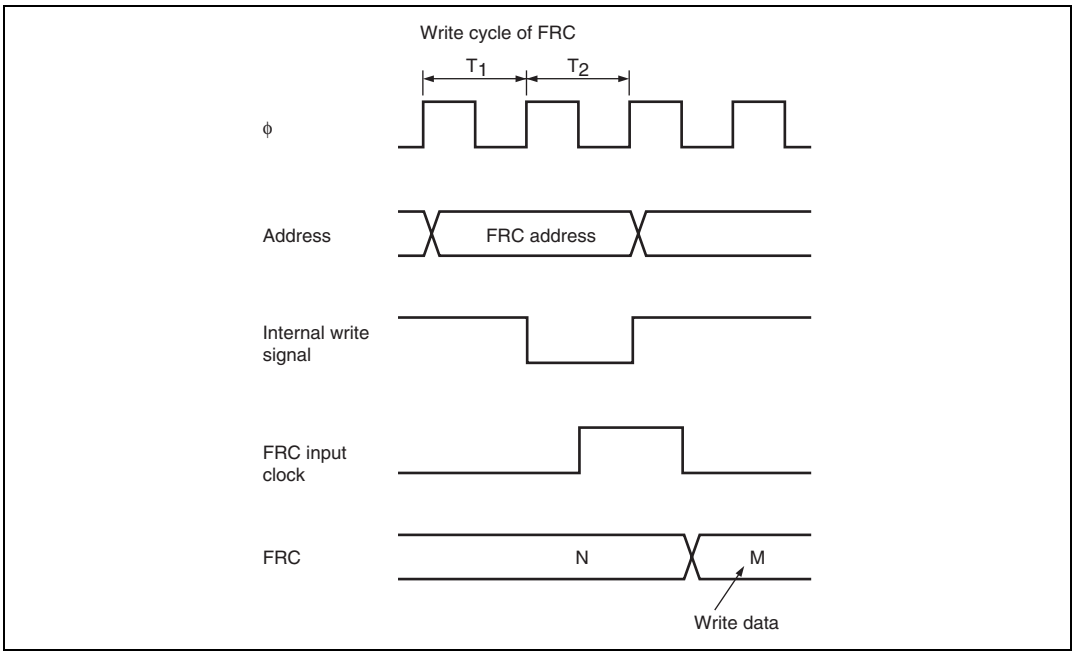


Figure 11.18 Conflict between FRC Write and Increment

11.7.3 Conflict between OCR Write and Compare-Match

If a compare-match occurs during the state after an OCRA or OCRB write cycle, the write takes priority and the compare-match signal is disabled. Figure 11.19 shows the timing for this type of conflict.

If automatic addition of OCRAR and OCRAF to OCRA is selected, and a compare-match occurs in the cycle following the OCRA, OCRAR, and OCRAF write cycle, the OCRA, OCRAR and OCRAF write takes priority and the compare-match signal is disabled. Consequently, the result of the automatic addition is not written to OCRA. Figure 11.20 shows the timing for this type of conflict.

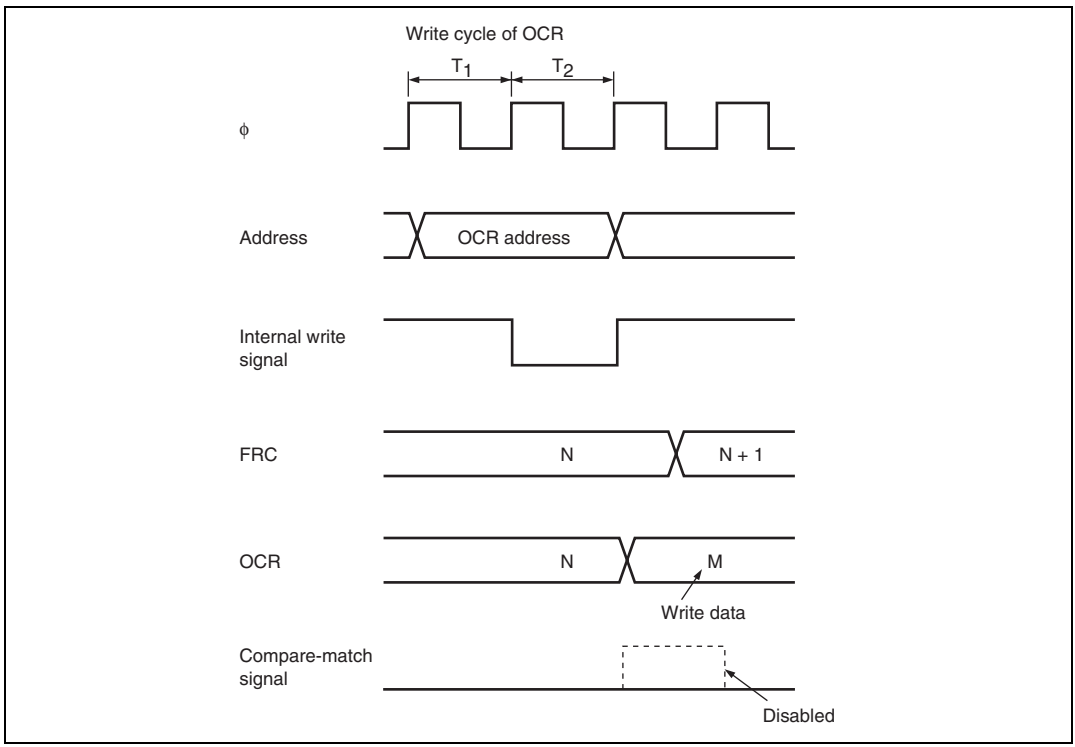


Figure 11.19 Conflict between OCR Write and Compare-Match (When Automatic Addition Function is Not Used)

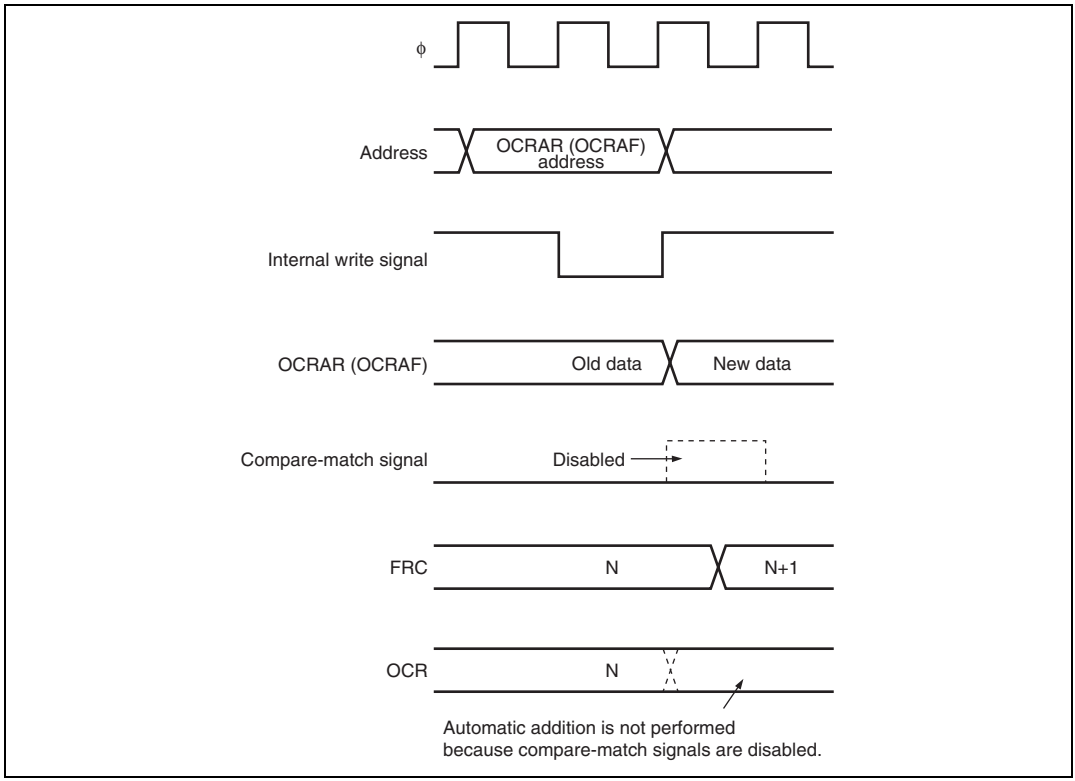


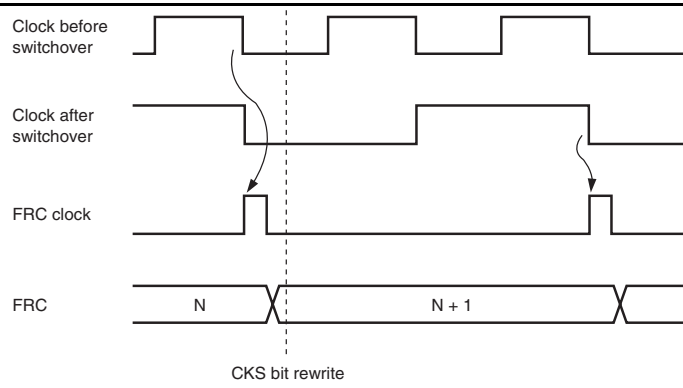
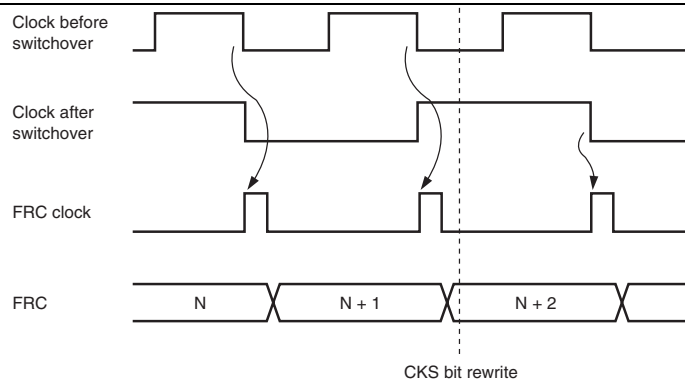
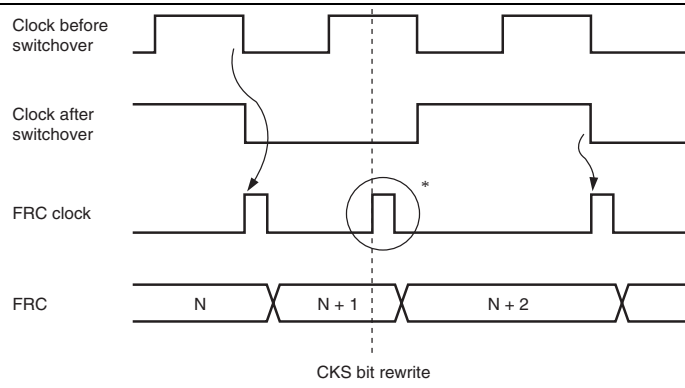
Figure 11.20 Conflict between OCR Write and Compare-Match (When Automatic Addition Function is Used)

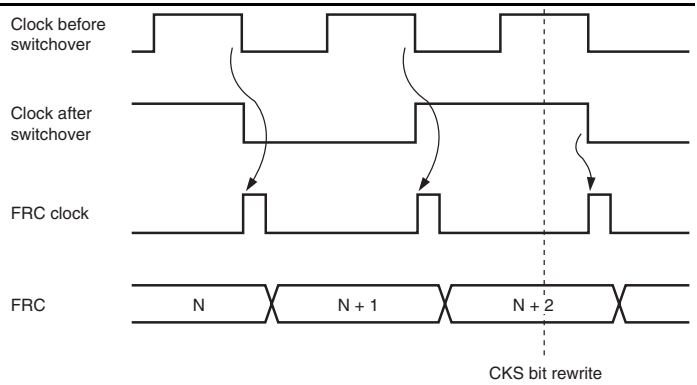
11.7.4 Switching of Internal Clock and FRC Operation

When the internal clock is changed, the changeover may source FRC to increment. This depends on the time at which the clock is switched (bits CKS1 and CKS0 are rewritten), as shown in table 11.3.

When an internal clock is used, the FRC clock is generated on detection of the falling edge of the internal clock scaled from the system clock (φ). If the clock is changed when the old source is high and the new source is low, as in case no. 3 in table 11.3, the changeover is regarded as a falling edge that triggers the FRC clock, and FRC is incremented. Switching between an internal clock and external clock can also source FRC to increment.

Table 11.3 Switching of Internal Clock and FRC Operation

No.	Timing of Switchover by Means of CKS1 and CKS0 Bits	FRC Operation
1	Switching from low to low	
2	Switching from low to high	
3	Switching from high to low	

No.	Timing of Switchover by Means of CKS1 and CKS0 Bits	FRC Operation
4	Switching from high to high	

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

11.7.5 Module Stop Mode Setting

FRT operation can be enabled or disabled by the module stop control register. In the initial state, FRT operation is disabled. Access to FRT registers is enabled when module stop mode is cancelled. For details, see section 24, Power-Down Modes.

Section 12 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 12.1 and figure 12.1, respectively.

12.1 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

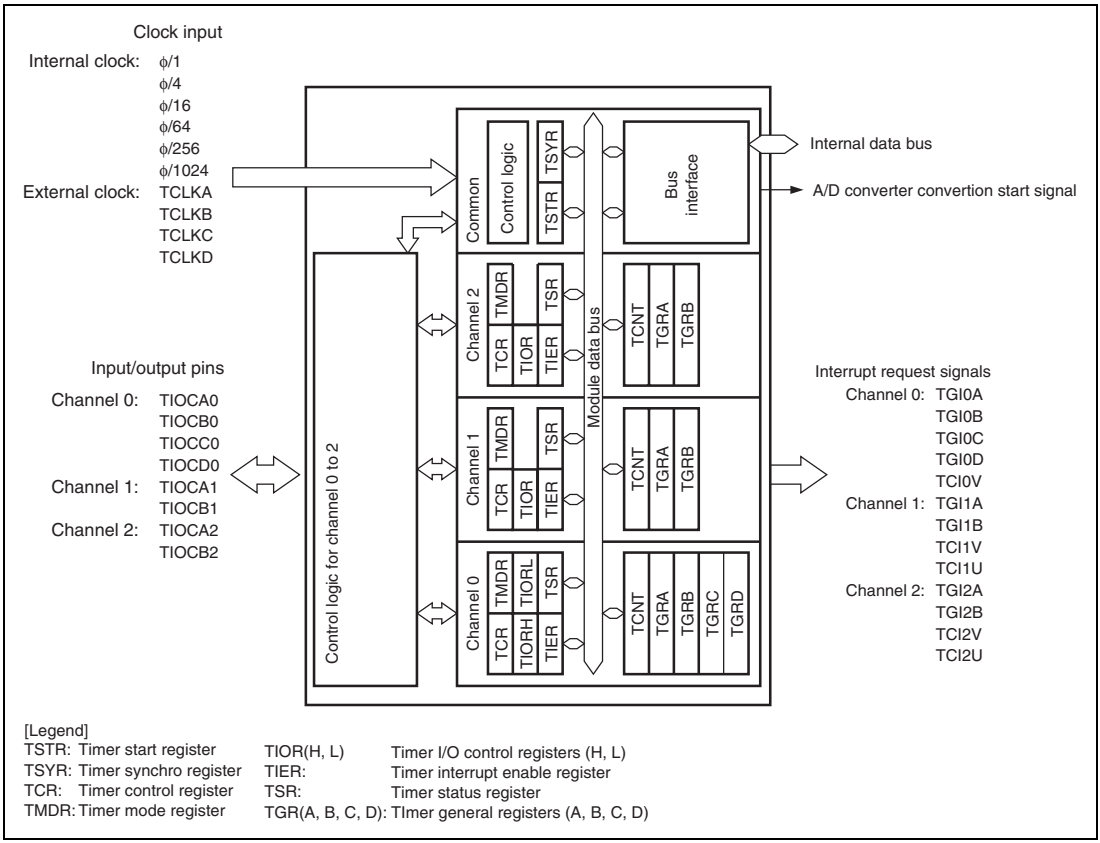


Figure 12.1 Block Diagram of TPU

Table 12.1 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
DTC activation	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture
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

12.2 Input/Output Pins

Table 12.2 Pin Configuration

Channel	Symbol	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

12.3 Register Descriptions

The TPU has the following registers.

- Timer control register_0 (TCR_0)
- Timer mode register_0 (TMDR_0)
- Timer I/O control register H_0 (TIORH_0)
- Timer I/O control register L_0 (TIORL_0)
- Timer interrupt enable register_0 (TIER_0)
- Timer status register_0 (TSR_0)
- Timer counter_0 (TCNT_0)
- Timer general register A_0 (TGRA_0)
- Timer general register B_0 (TGRB_0)
- Timer general register C_0 (TGRC_0)
- Timer general register D_0 (TGRD_0)
- Timer control register_1 (TCR_1)
- Timer mode register_1 (TMDR_1)
- Timer I/O control register _1 (TIOR_1)
- Timer interrupt enable register_1 (TIER_1)
- Timer status register_1 (TSR_1)
- Timer counter_1 (TCNT_1)
- Timer general register A_1 (TGRA_1)
- Timer general register B_1 (TGRB_1)
- Timer control register_2 (TCR_2)
- Timer mode register_2 (TMDR_2)
- Timer I/O control register_2 (TIOR_2)
- Timer interrupt enable register_2 (TIER_2)
- Timer status register_2 (TSR_2)
- Timer counter_2 (TCNT_2)
- Timer general register A_2 (TGRA_2)
- Timer general register B_2 (TGRB_2)

Common Registers

- Timer start register (TSTR)
- Timer synchro register (TSYR)

12.3.1 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 12.3 and 12.4 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 1 and 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, or when overflow/underflow of another channel is selected. 00: Count at rising edge 01: Count at falling edge 1x: Count at both edges [Legend] x: Don't care
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 12.5 to 12.7 for details.
0	TPSC0	0	R/W	

Table 12.3 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 12.4 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 12.5 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 12.6 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 12.7 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 $\phi/4$
		1	0	Internal clock: counts on $\phi/16$
			1	Internal clock: counts on $\phi/64$
	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 $\phi/1024$

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

12.3.2 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. In channels 1 and 2, which 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. In channels 1 and 2, which 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 12.8, MD3 to MD0 for details.
2	MD2	0	R/W	
1	MD1	0	R/W	
0	MD0	0	R/W	

Table 12.8 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.

12.3.3 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 12.9 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 12.10 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 12.11 TIORL_0 (channel 0)

				Description	
Bit 7 IOD3	**Bit 6 IOD2**	**Bit 5 IOD1**	**Bit 4 IOD0**	**TGRA_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 12.12 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 12.13 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 12.14 TIOR_1 (channel 1)

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_1 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 12.15 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 12.16 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

12.3.4 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 (TCU) 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

12.3.5 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 conditions] • When DTC is activated by TGID interrupt while DISEL bit or MRB in DTC is 0 • 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 conditions] • When DTC is activated by TGIC interrupt while DISEL bit or MRB in DTC is 0 • 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 conditions] • When DTC is activated by TGFB interrupt while DISEL bit of MRB in DTC is 0. • 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 conditions] • When DTC is activated by TGIA interrupt while DISEL bit of MRB in DTC is 0 • When 0 is written to TGFA after reading TGFA = 1

Note: * The write value should always be 0 to clear the flag.

12.3.6 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, and in hardware standby mode. The TCNT counters cannot be accessed in 8-bit units; they must always be accessed as a 16-bit unit.

12.3.7 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 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.

12.3.8 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_2 to TCNT_0 count operation is stopped 1: TCNT_2 to TCNT_0 performs count operation

12.3.9 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	SYNC1	0	R/W	These bits select whether operation is independent of or synchronized with other channels.
0	SYNC0	0	R/W	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_2 to TCNT_0 operates independently (TCNT presetting /clearing is unrelated to other channels) 1: TCNT_2 to TCNT_0 performs synchronous operation TCNT synchronous presetting/synchronous clearing is possible

12.4 Interface to Bus Master

12.4.1 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 12.2.

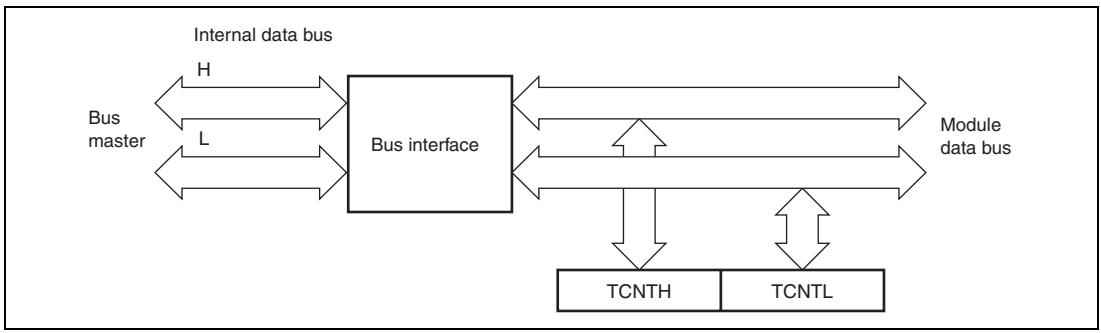


Figure 12.2 16-Bit Register Access Operation [Bus Master ↔ TCNT (16 Bits)]

12.4.2 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 12.3, 12.4, and 12.5.

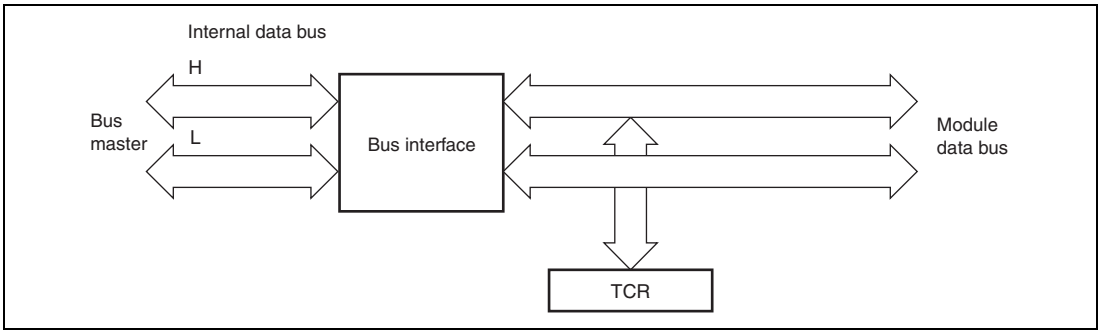


Figure 12.3 8-Bit Register Access Operation [Bus Master ↔ TCR (Upper 8 Bits)]

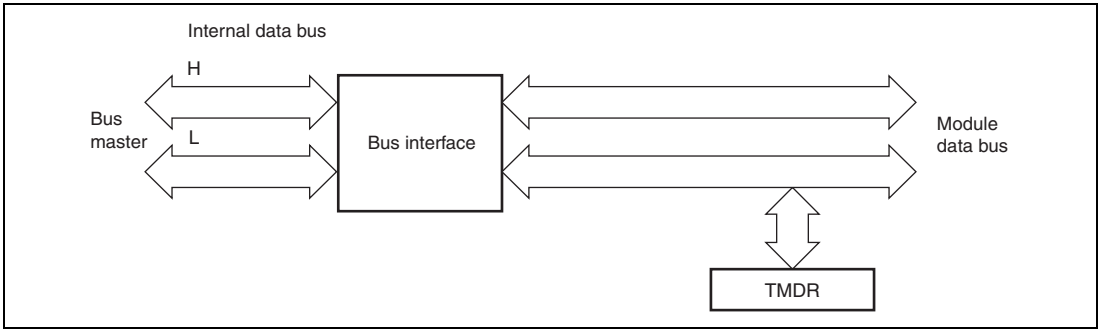


Figure 12.4 8-Bit Register Access Operation [Bus Master ↔ TMDR (Lower 8 Bits)]

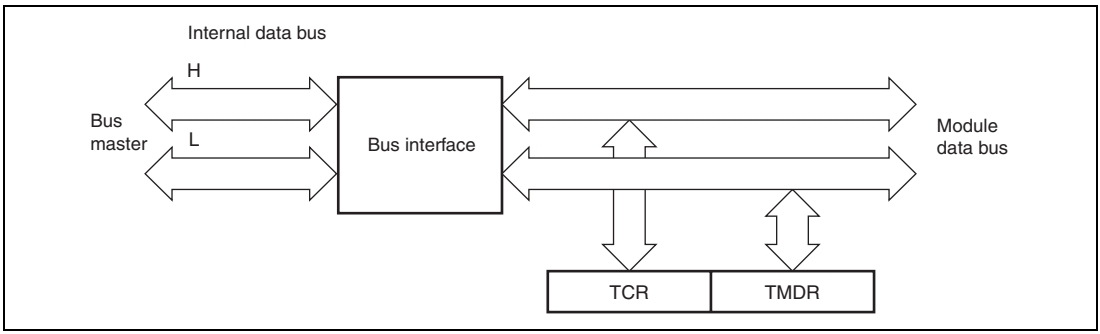


Figure 12.5 8-Bit Register Access Operation [Bus Master ↔ TCR and TMDR (16 Bits)]

12.5 Operation

12.5.1 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.

1. Example of count operation setting procedure Figure 12.6 shows an example of the count operation setting procedure.

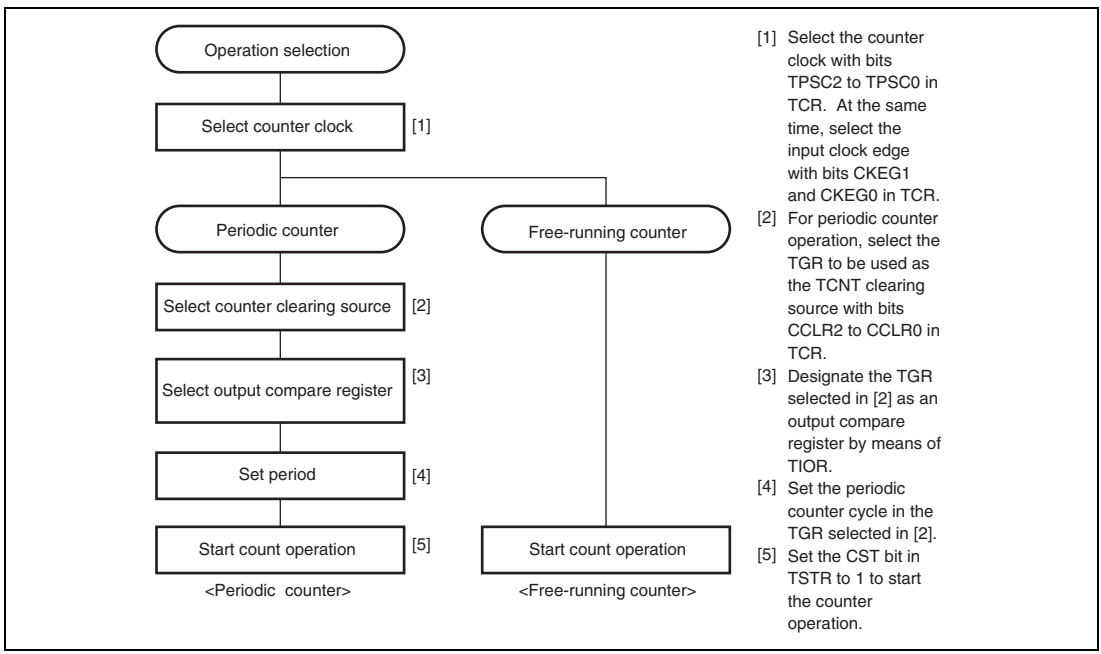


Figure 12.6 Example of Counter Operation Setting Procedure

2. 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 12.7 illustrates free-running counter operation.

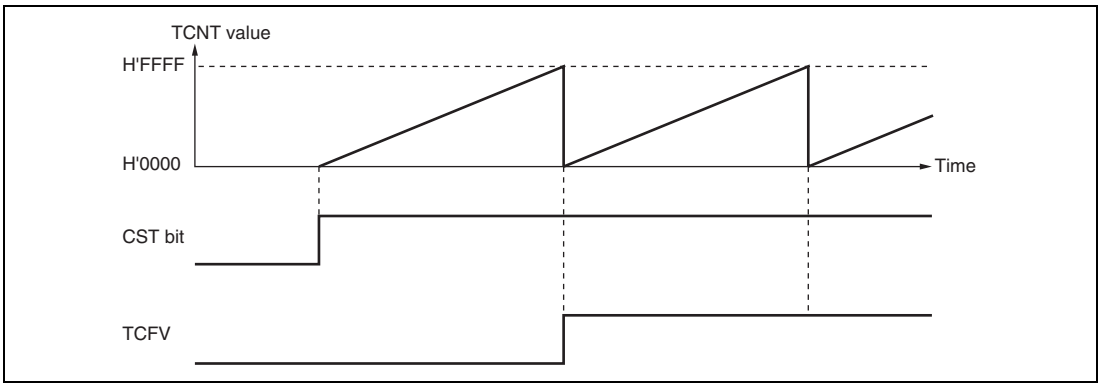


Figure 12.7 Free-Running Counter Operation

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 12.8 illustrates periodic counter operation.

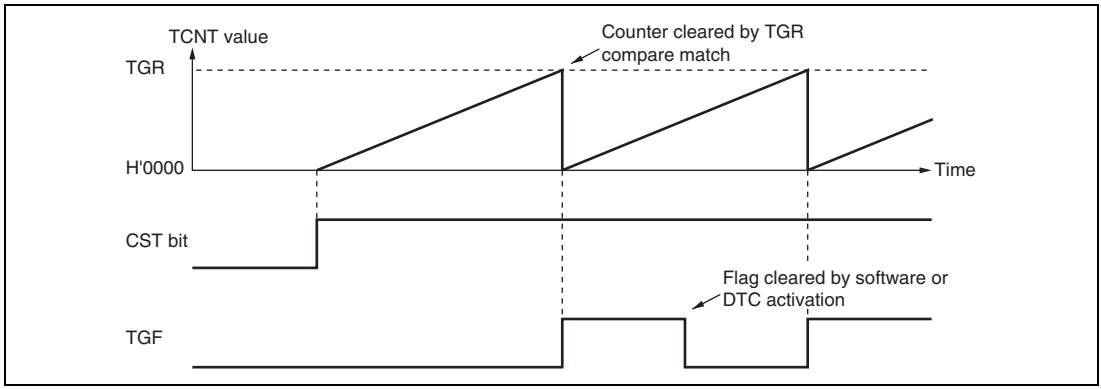


Figure 12.8 Periodic Counter Operation

(2) Waveform Output by Compare Match

The TPU can perform 0, 1, or toggle output from the corresponding output pin using compare match.

1. Example of setting procedure for waveform output by compare match

   Figure 12.9 shows an example of the setting procedure for waveform output by compare match.

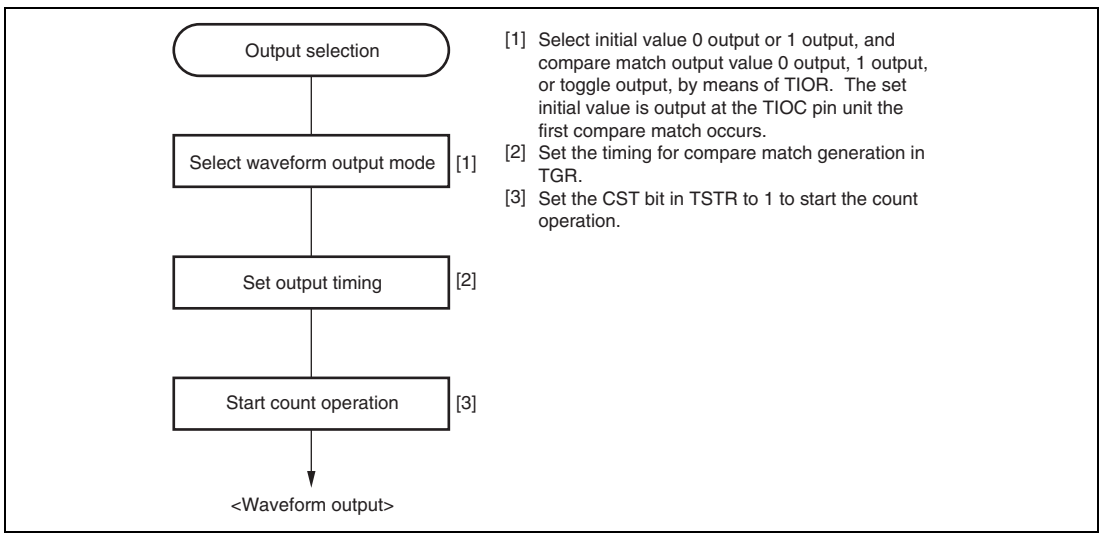


Figure 12.9 Example of Setting Procedure for Waveform Output by Compare Match

2. Examples of waveform output operation

Figure 12.10 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.

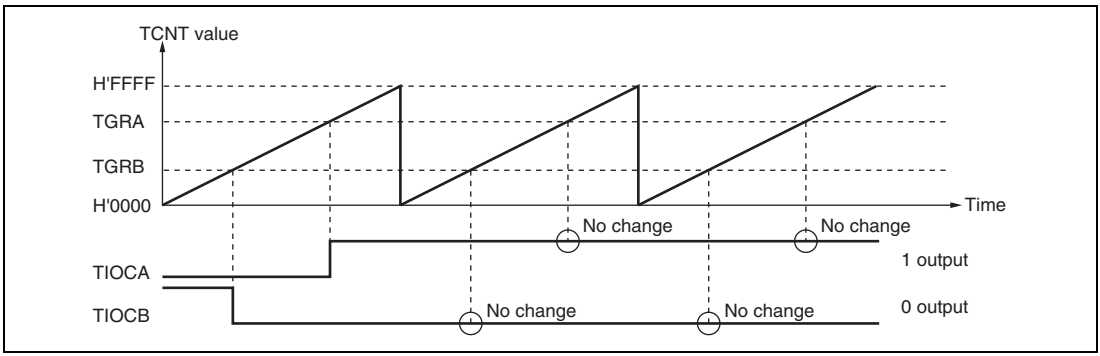


Figure 12.10 Example of 0 Output/1 Output Operation

Figure 12.11 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.

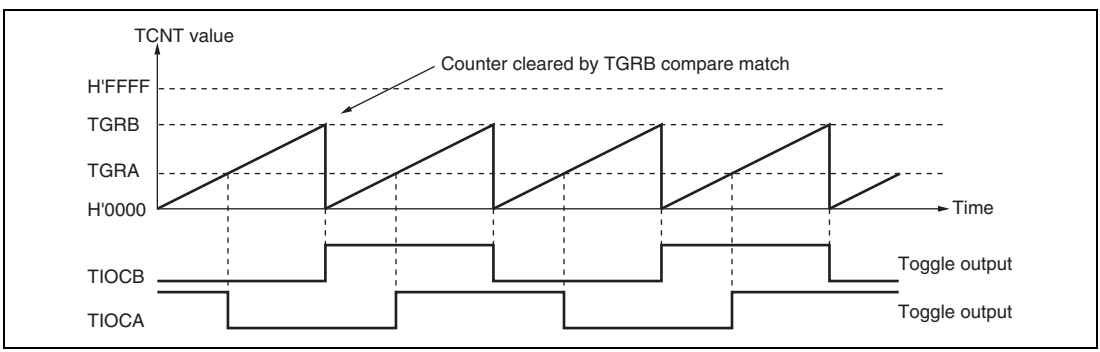


Figure 12.11 Example of Toggle Output Operation

(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.

1. Example of input capture operation setting procedure
   Figure 12.12 shows an example of the input capture operation setting procedure.

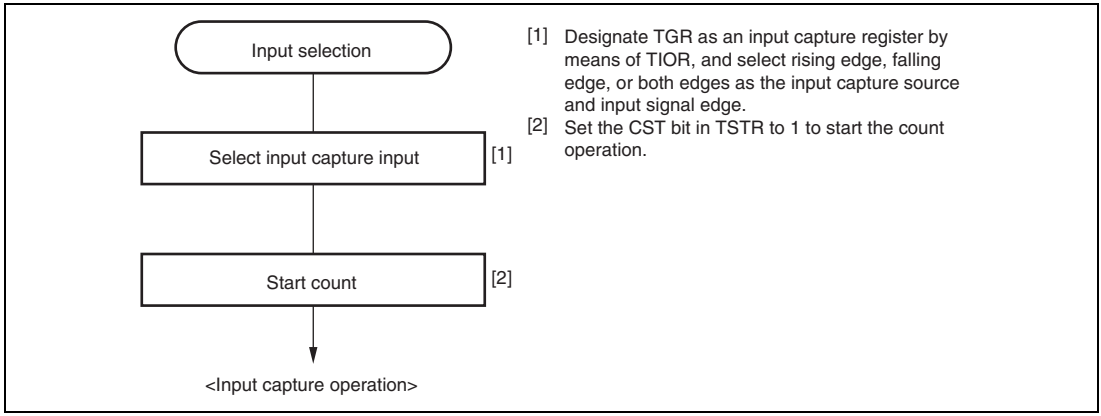


Figure 12.12 Example of Input Capture Operation Setting Procedure

2. Example of input capture operation

Figure 12.13 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.

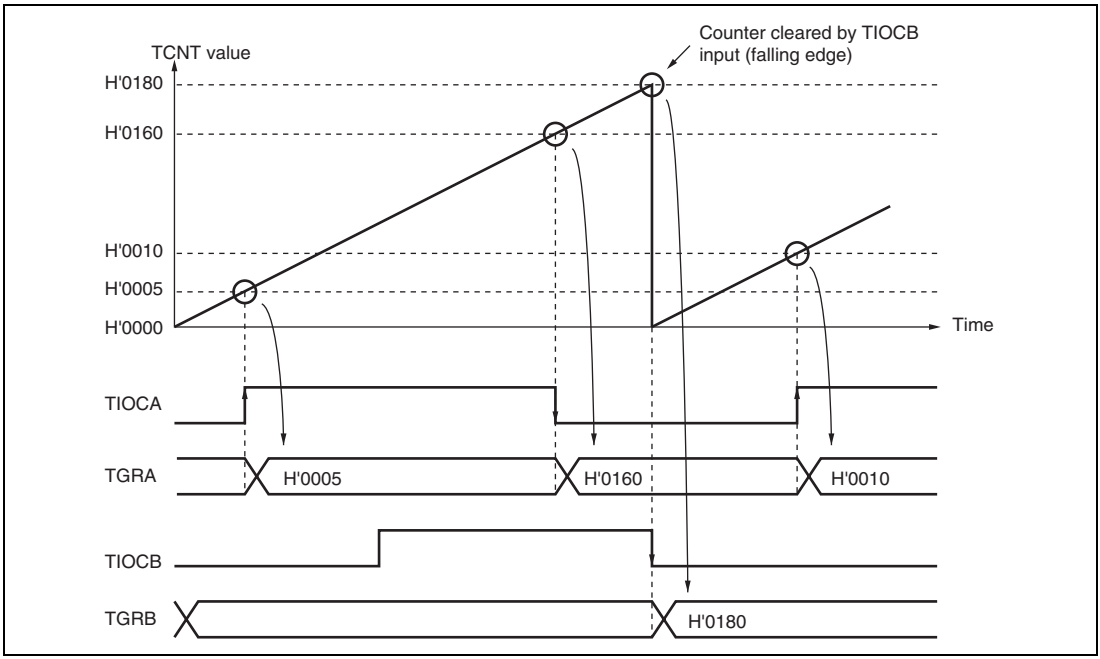


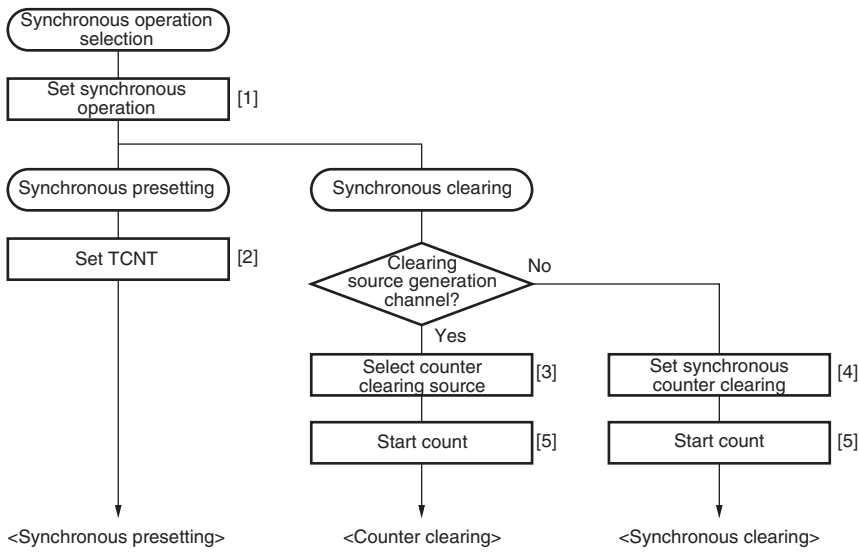
Figure 12.13 Example of Input Capture Operation

12.5.2 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 12.14 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 12.14 Example of Synchronous Operation Setting Procedure

(2) Example of Synchronous Operation

Figure 12.15 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 12.5.4, PWM Modes.

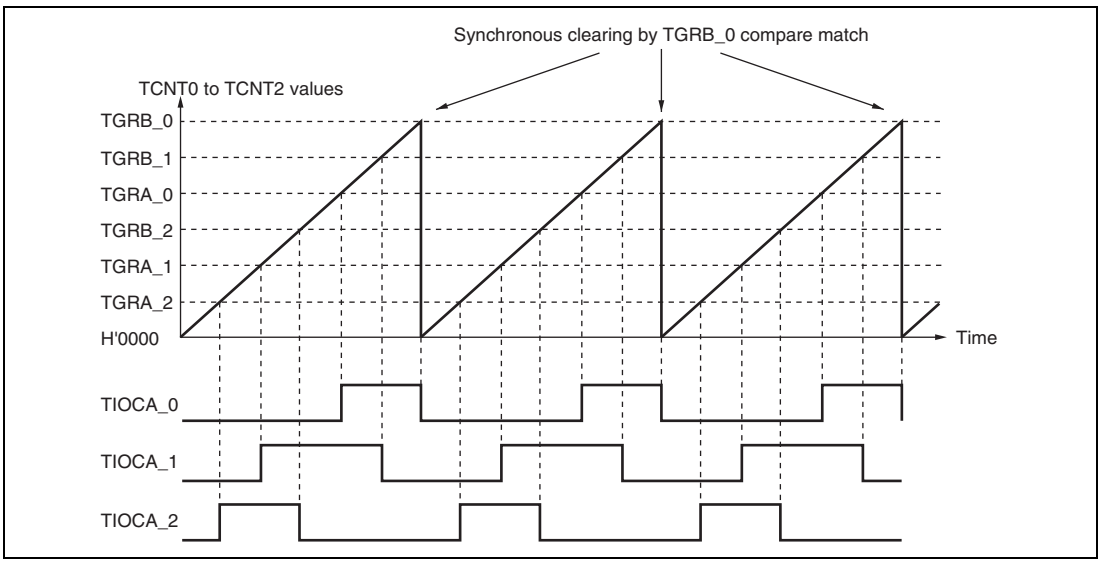


Figure 12.15 Example of Synchronous Operation

12.5.3 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 12.17 shows the register combinations used in buffer operation.

Table 12.17 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 12.16.

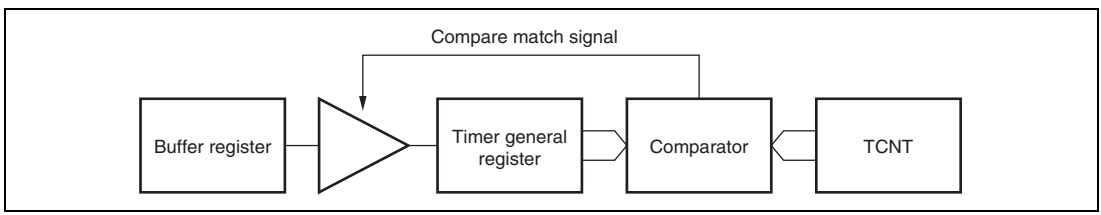


Figure 12.16 Compare Match Buffer Operation

- 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 12.17.

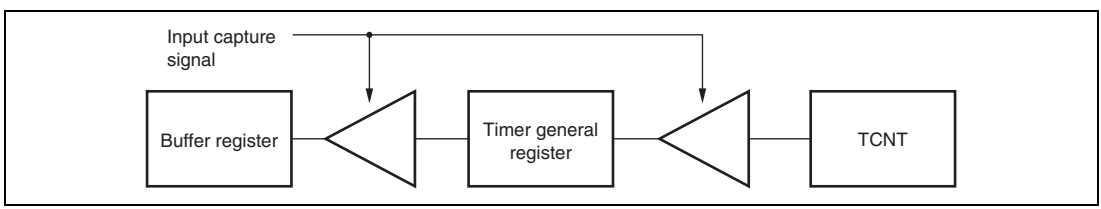


Figure 12.17 Input Capture Buffer Operation

(1) Example of Buffer Operation Setting Procedure

Figure 12.18 shows an example of the buffer operation setting procedure.

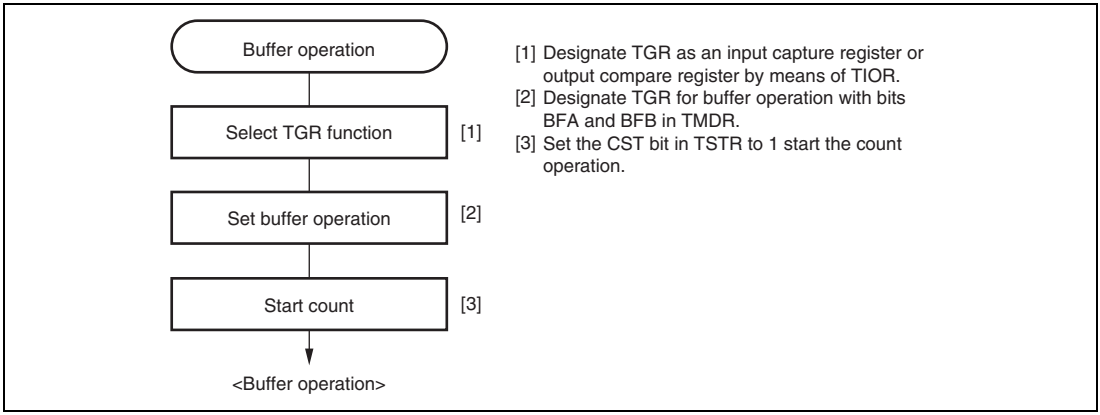


Figure 12.18 Example of Buffer Operation Setting Procedure

(2) Examples of Buffer Operation

1. When TGR is an output compare register

   Figure 12.19 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 12.5.4, PWM Modes.

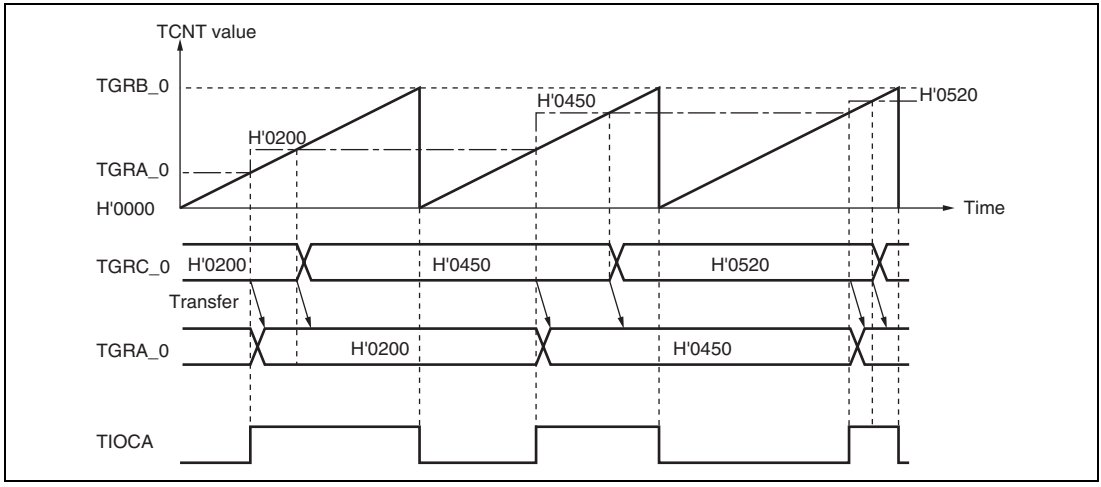


Figure 12.19 Example of Buffer Operation (1)

2. When TGR is an input capture register

   Figure 12.20 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.

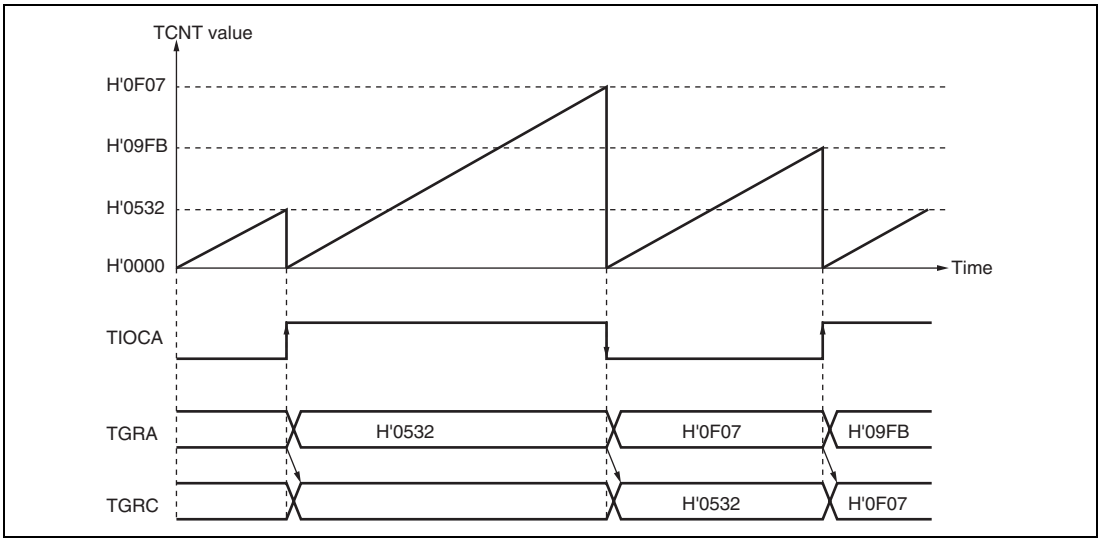


Figure 12.20 Example of Buffer Operation (2)

12.5.4 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 12.18.

Table 12.18 PWM Output Registers and Output Pins

Channel	Registers	Output Pins: PWM Mode 1	Output Pins: 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 12.21 shows an example of the PWM mode setting procedure.

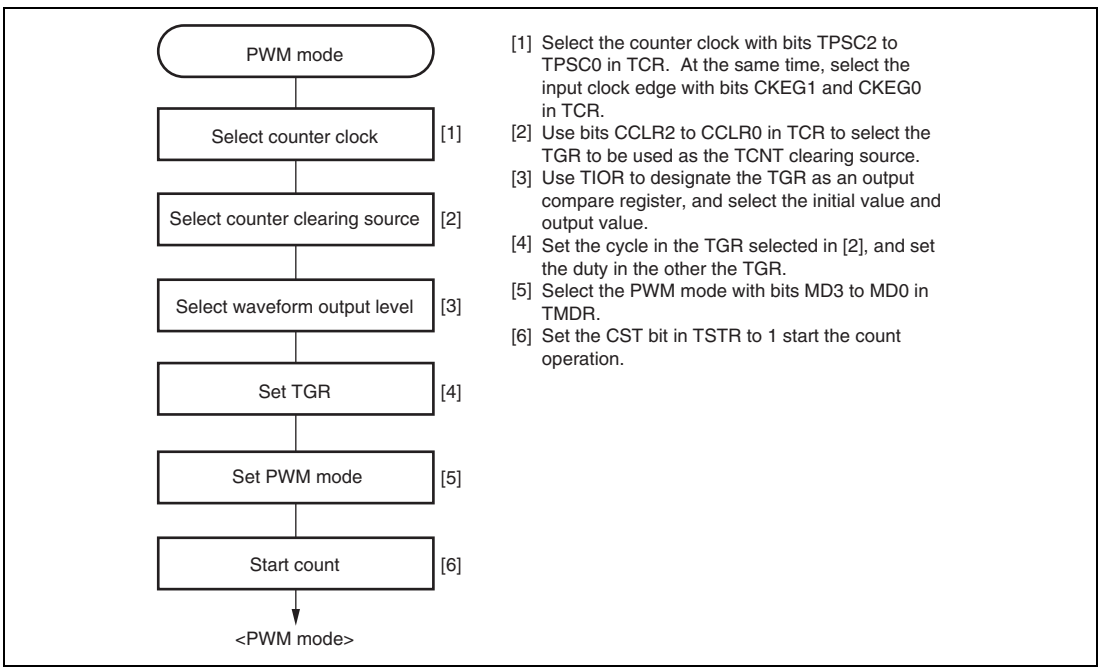


Figure 12.21 Example of PWM Mode Setting Procedure

(2) Examples of PWM Mode Operation

Figure 12.22 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.

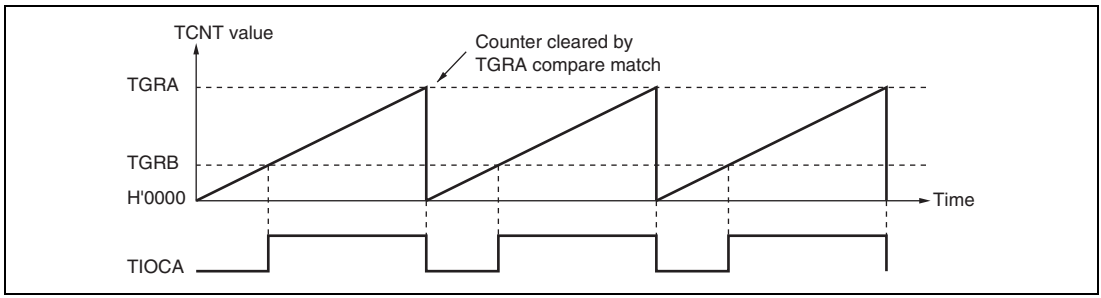


Figure 12.22 Example of PWM Mode Operation (1)

Figure 12.23 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.

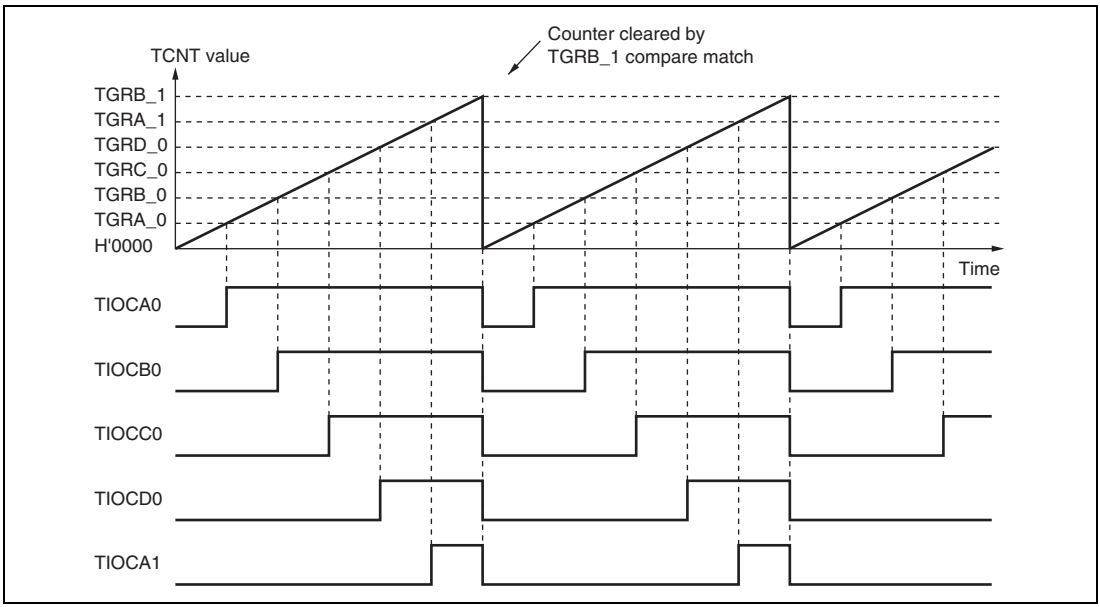


Figure 12.23 Example of PWM Mode Operation (2)

Figure 12.24 shows examples of PWM waveform output with 0% duty and 100% duty in PWM mode.

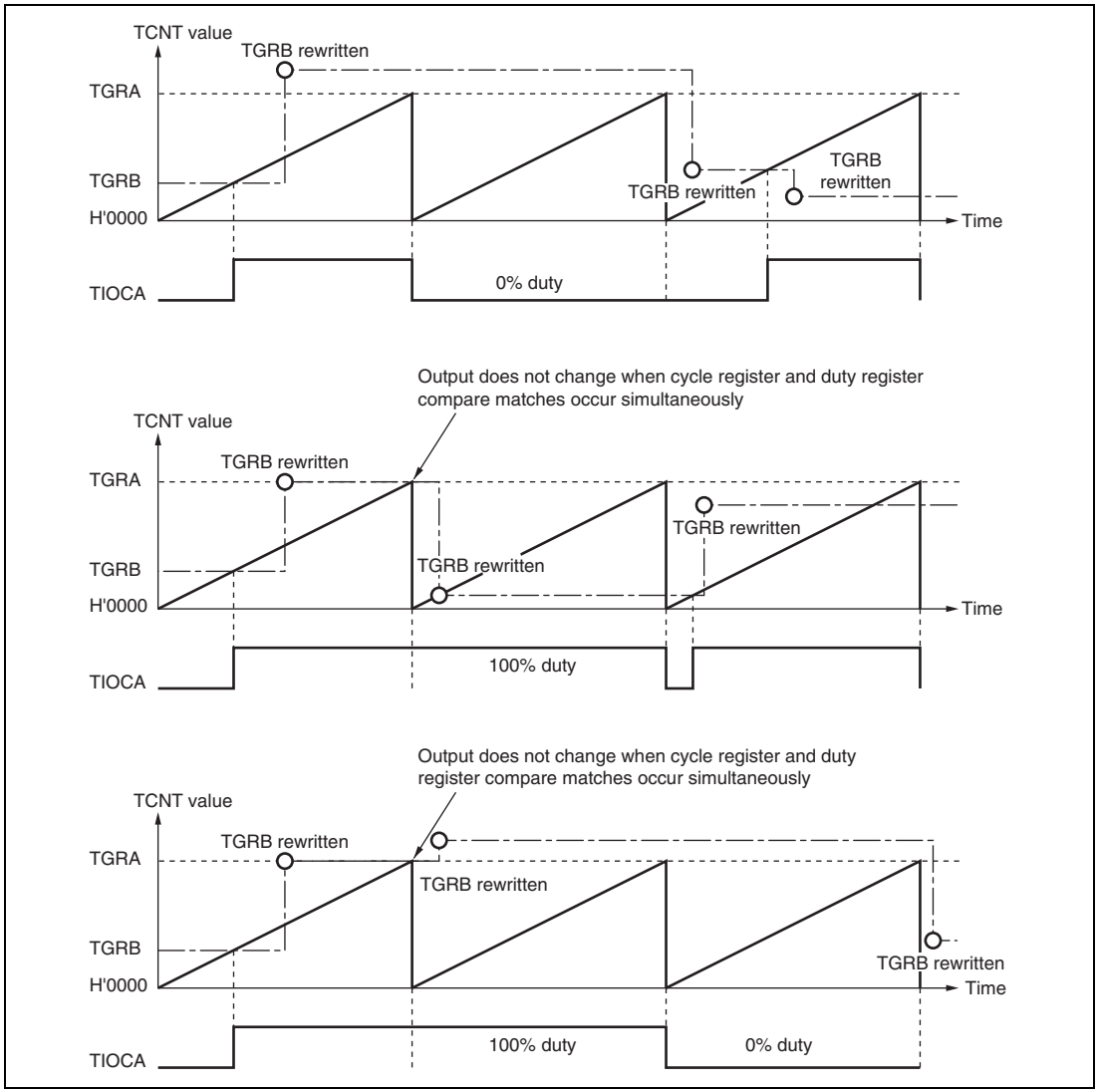


Figure 12.24 Example of PWM Mode Operation (3)

12.5.5 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 12.19 shows the correspondence between external clock pins and channels.

Table 12.19 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 12.25 shows an example of the phase counting mode setting procedure.

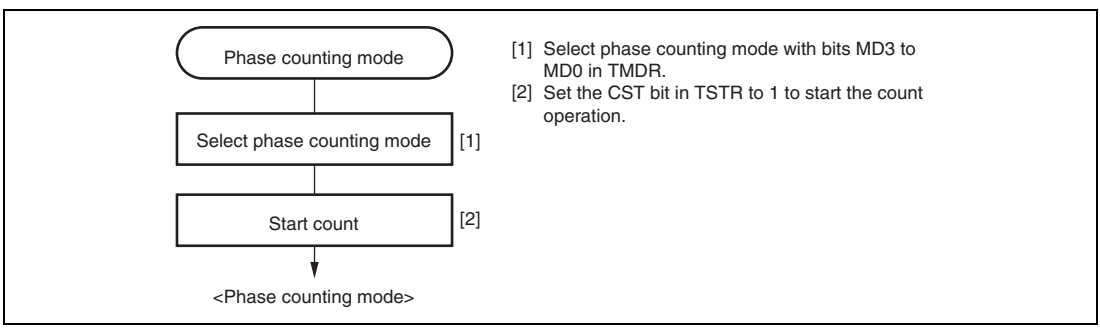


Figure 12.25 Example of Phase Counting Mode Setting Procedure

(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.

1. Phase counting mode 1

   Figure 12.26 shows an example of phase counting mode 1 operation, and table 12.20 summarizes the TCNT up/down-count conditions.

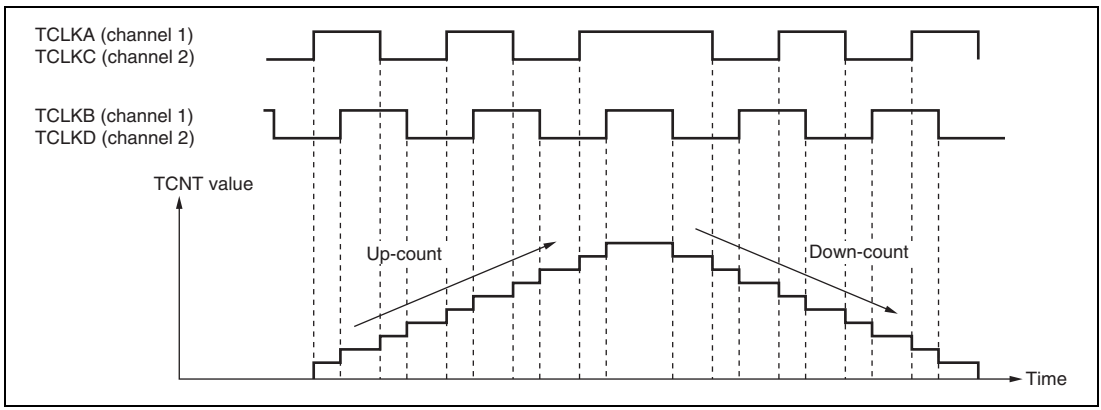


Figure 12.26 Example of Phase Counting Mode 1 Operation

Table 12.20 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

2. Phase counting mode 2

Figure 12.27 shows an example of phase counting mode 2 operation, and table 12.21 summarizes the TCNT up/down-count conditions.

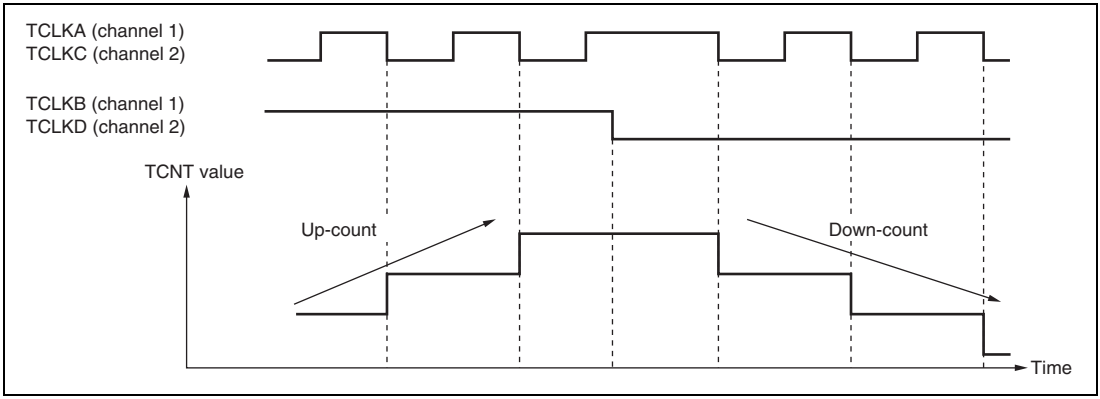


Figure 12.27 Example of Phase Counting Mode 2 Operation

Table 12.21 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

3. Phase counting mode 3

Figure 12.28 shows an example of phase counting mode 3 operation, and table 12.22 summarizes the TCNT up/down-count conditions.

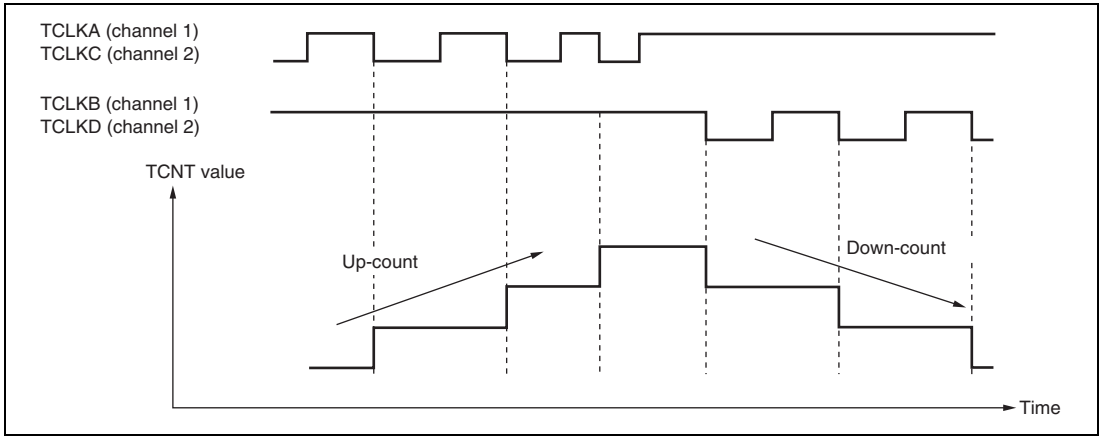


Figure 12.28 Example of Phase Counting Mode 3 Operation

Table 12.22 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

4. Phase counting mode 4

   Figure 12.29 shows an example of phase counting mode 4 operation, and table 12.23 summarizes the TCNT up/down-count conditions.

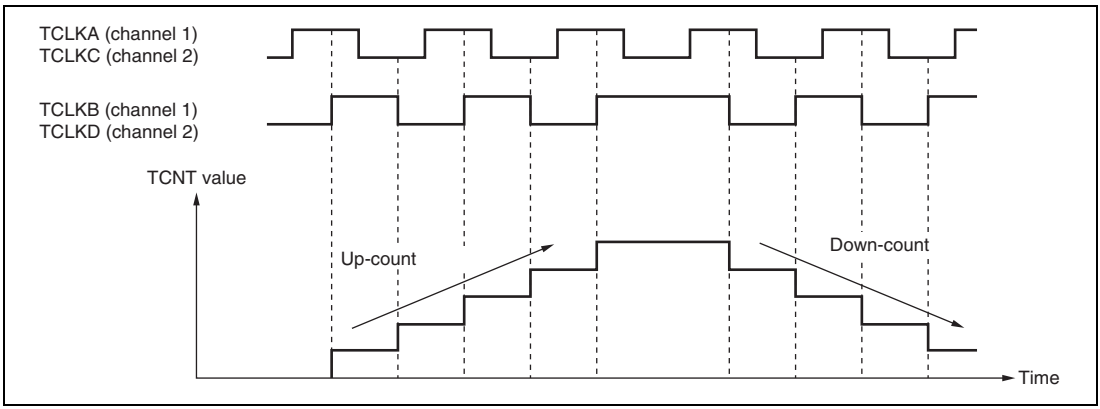


Figure 12.29 Example of Phase Counting Mode 4 Operation

Table 12.23 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

12.6 Interrupts

12.6.1 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 12.24 lists the TPU interrupt sources.

Table 12.24 TPU Interrupts

Channel	Name	Interrupt Source	Interrupt Flag	DTC Activation	Priority
0	TGI0A	TGRA_0 input capture/compare match	TGFA	Enable	High
	TGI0B	TGRB_0 input capture/compare match	TGFB	Enable	
	TGI0C	TGRC_0 input capture/compare match	TGFC	Enable	
	TGI0D	TGRD_0 input capture/compare match	TGFD	Enable	
	TCI0V	TCNT_0 overflow	TCFV	Disable	
1	TGI1A	TGRA_1 input capture/compare match	TGFA	Enable	
	TGI1B	TGRB_1 input capture/compare match	TGFB	Enable	
	TCI1V	TCNT_1 overflow	TCFV	Disable	
	TCI1U	TCNT_1 underflow	TCFU	Disable	
2	TGI2A	TGRA_2 input capture/compare match	TGFA	Enable	
	TGI2B	TGRB_2 input capture/compare match	TGFB	Enable	
	TCI2V	TCNT_2 overflow	TCFV	Disable	
	TCI2U	TCNT_2 underflow	TCFU	Disable	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.

12.6.2 DTC Activation

The DTC can be activated by the TGR input capture/compare match interrupt for a channel. For details, see section 7, Data Transfer Controller (DTC). A total of eight TPU input capture/compare match interrupts can be used as DTC activation sources, four each for channel 0, and two each for channels 1 and 2.

12.6.3 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.

12.7 Operation Timing

12.7.1 Input/Output Timing

(1) TCNT Count Timing

Figure 12.30 shows TCNT count timing in internal clock operation, and figure 12.31 shows TCNT count timing in external clock operation.

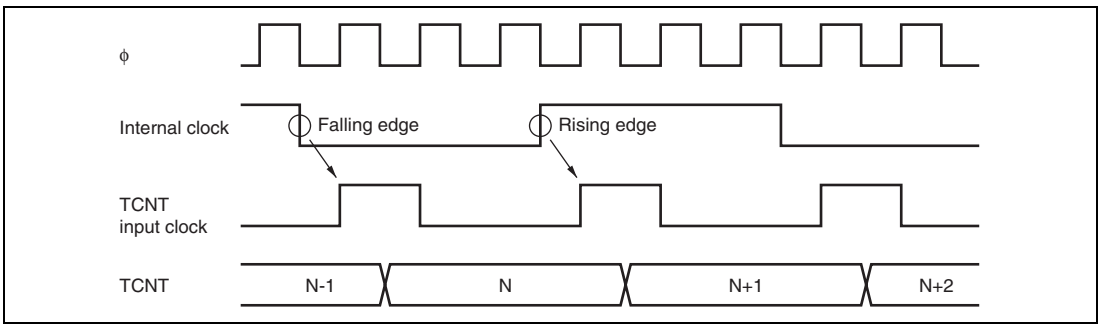


Figure 12.30 Count Timing in Internal Clock Operation

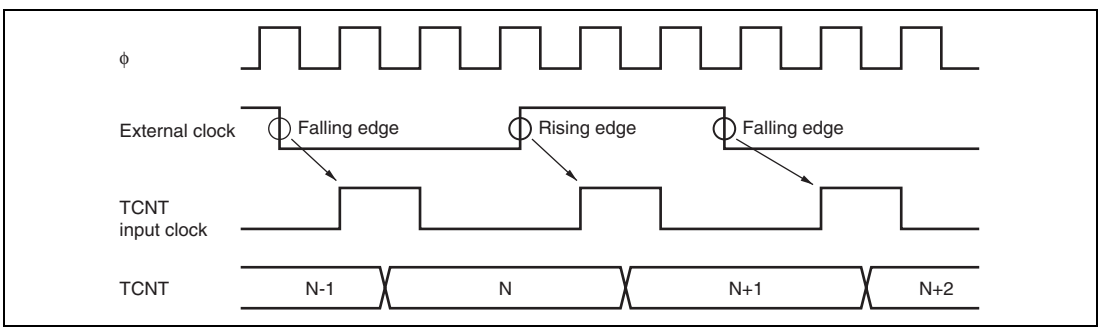


Figure 12.31 Count Timing in External Clock Operation

(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 12.32 shows output compare output timing.

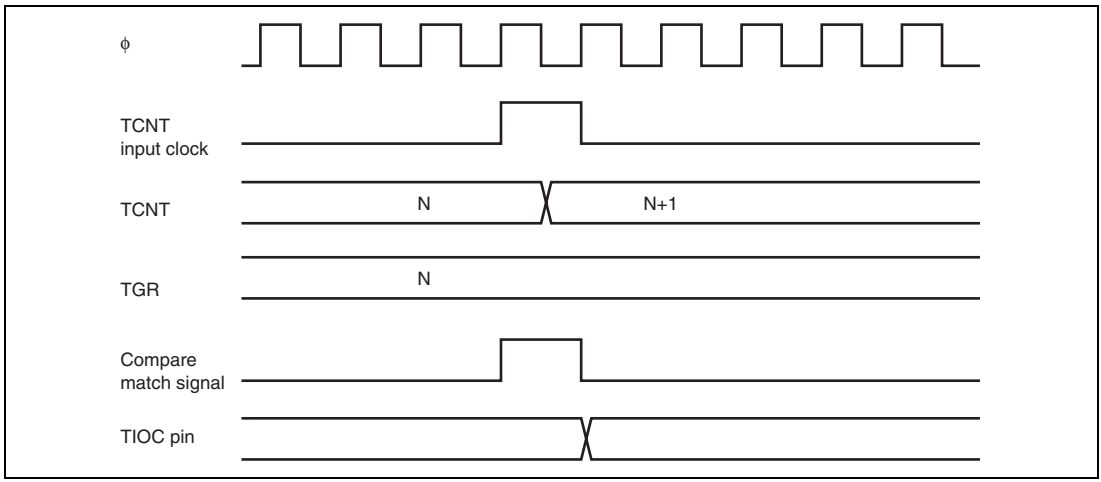


Figure 12.32 Output Compare Output Timing

(3) Input Capture Signal Timing

Figure 12.33 shows input capture signal timing.

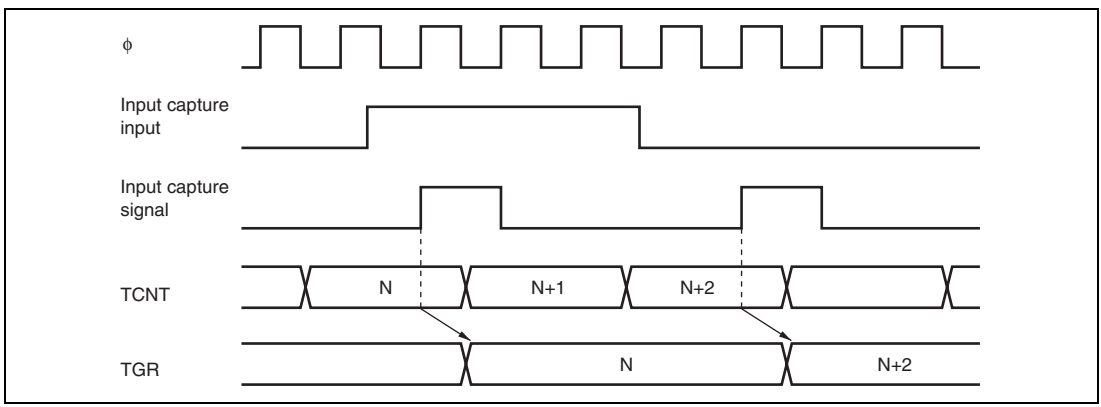


Figure 12.33 Input Capture Input Signal Timing

(4) Timing for Counter Clearing by Compare Match/Input Capture

Figure 12.34 shows the timing when counter clearing by compare match occurrence is specified, and figure 12.35 shows the timing when counter clearing by input capture occurrence is specified.

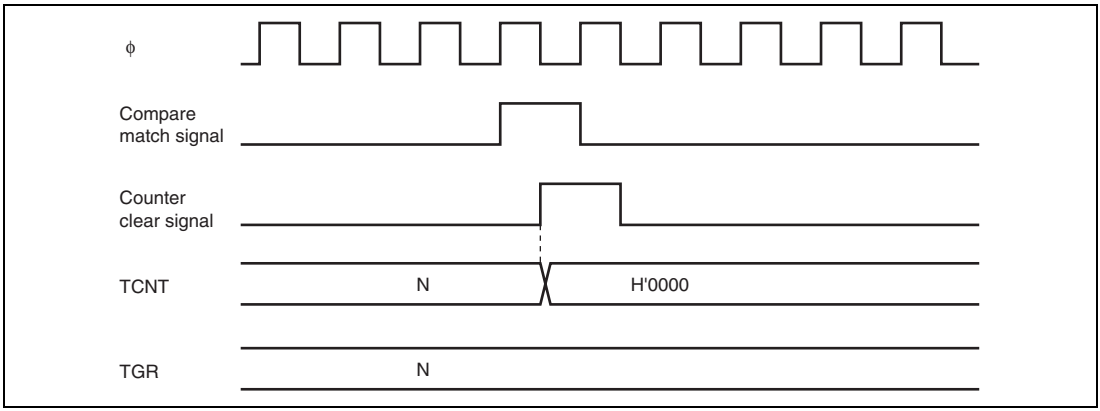


Figure 12.34 Counter Clear Timing (Compare Match)

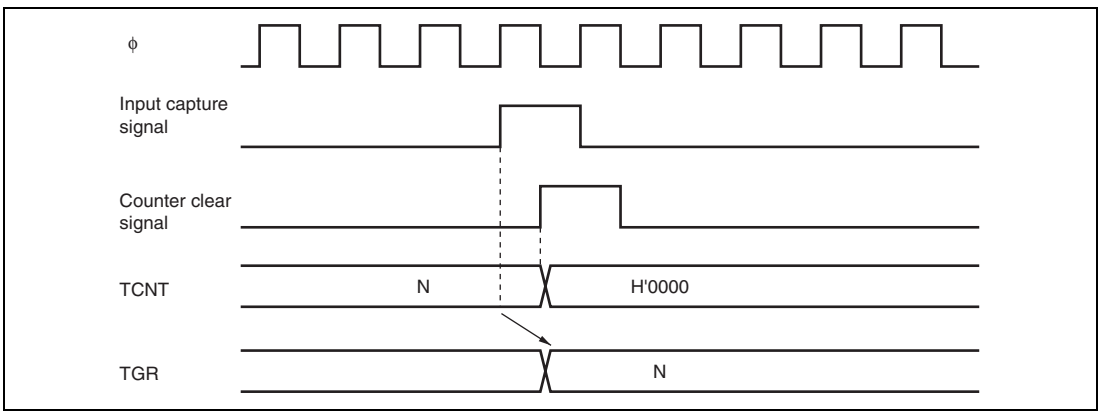


Figure 12.35 Counter Clear Timing (Input Capture)

(5) Buffer Operation Timing

Figures 12.36 and 12.37 show the timing in buffer operation.

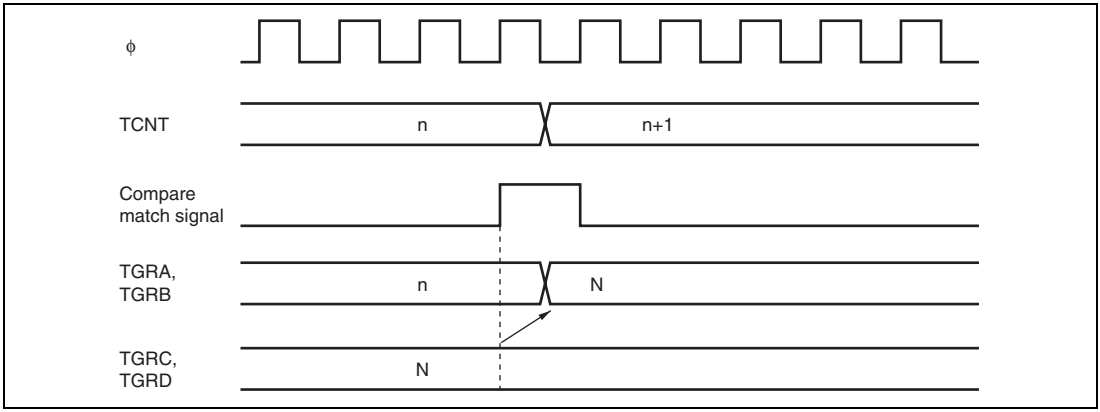


Figure 12.36 Buffer Operation Timing (Compare Match)

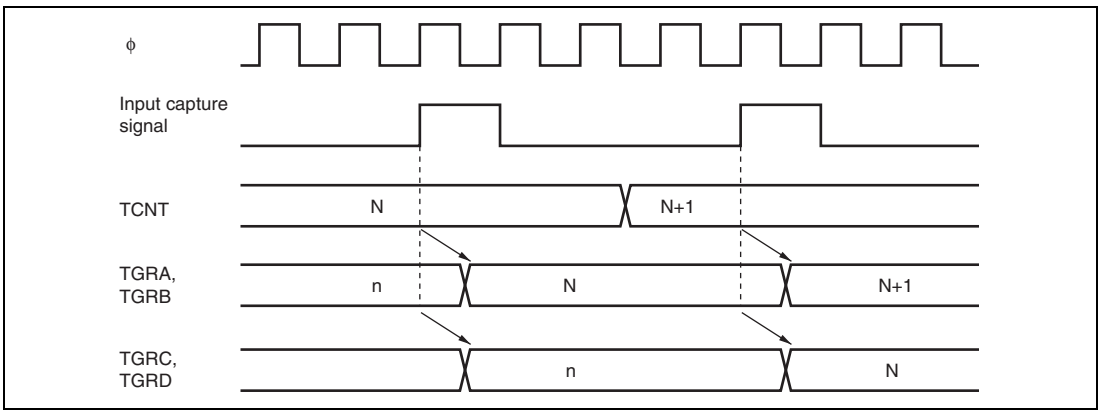


Figure 12.37 Buffer Operation Timing (Input Capture)

12.7.2 Interrupt Signal Timing

(1) TGF Flag Setting Timing in Case of Compare Match

Figure 12.38 shows the timing for setting of the TGF flag in TSR by compare match occurrence, and TGI interrupt request signal timing.

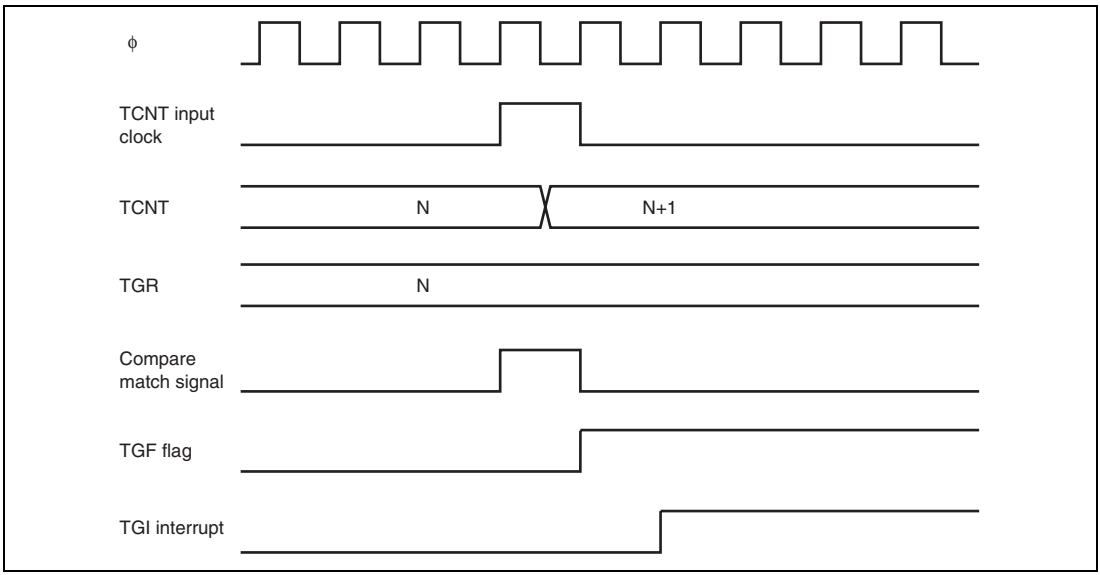


Figure 12.38 TGI Interrupt Timing (Compare Match)

(2) TGF Flag Setting Timing in Case of Input Capture

Figure 12.39 shows the timing for setting of the TGF flag in TSR by input capture occurrence, and TGI interrupt request signal timing.

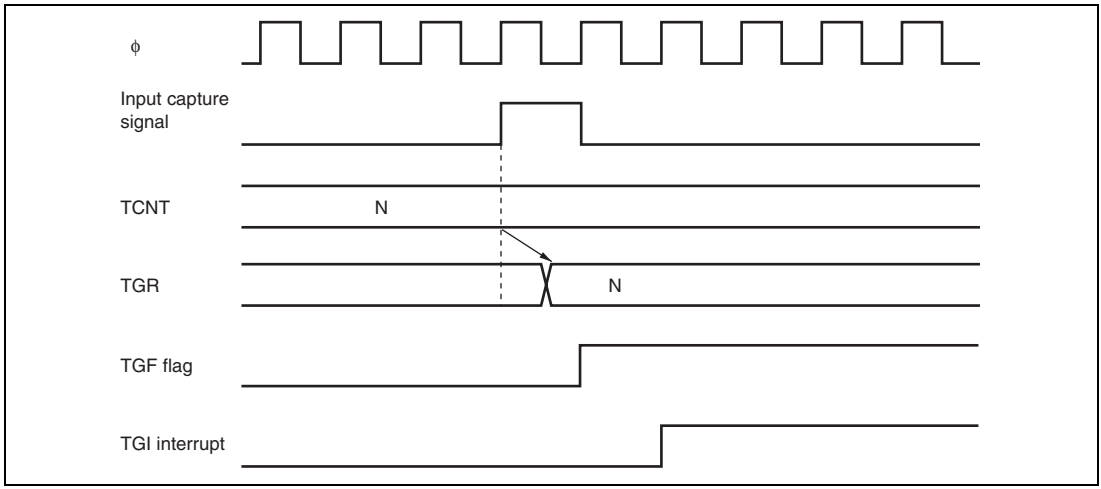


Figure 12.39 TGI Interrupt Timing (Input Capture)

(3) TCFV Flag/TCFU Flag Setting Timing

Figure 12.40 shows the timing for setting of the TCFV flag in TSR by overflow occurrence, and TCIV interrupt request signal timing. Figure 12.41 shows the timing for setting of the TCFU flag in TSR by underflow occurrence, and TCIU interrupt request signal timing.

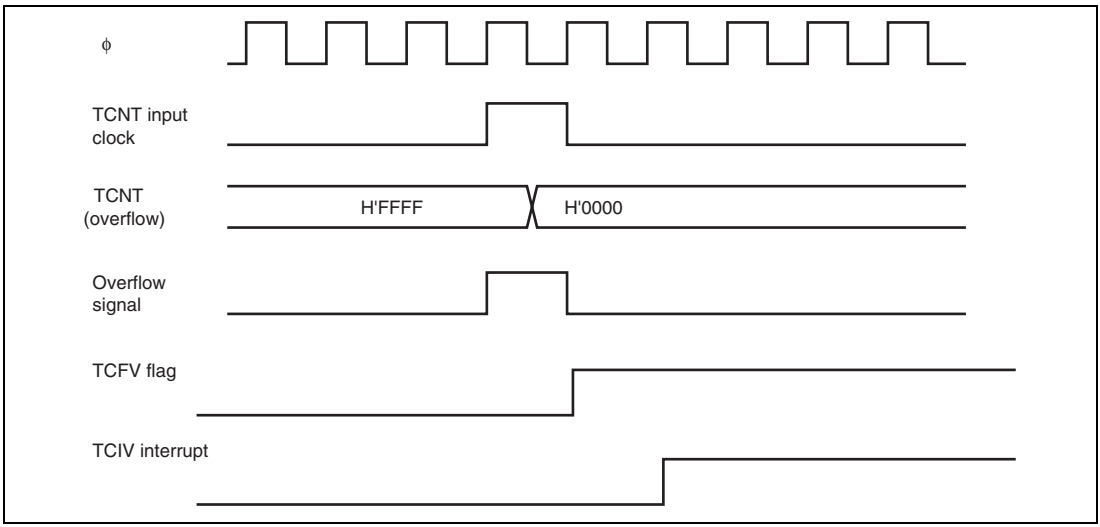


Figure 12.40 TCIV Interrupt Setting Timing

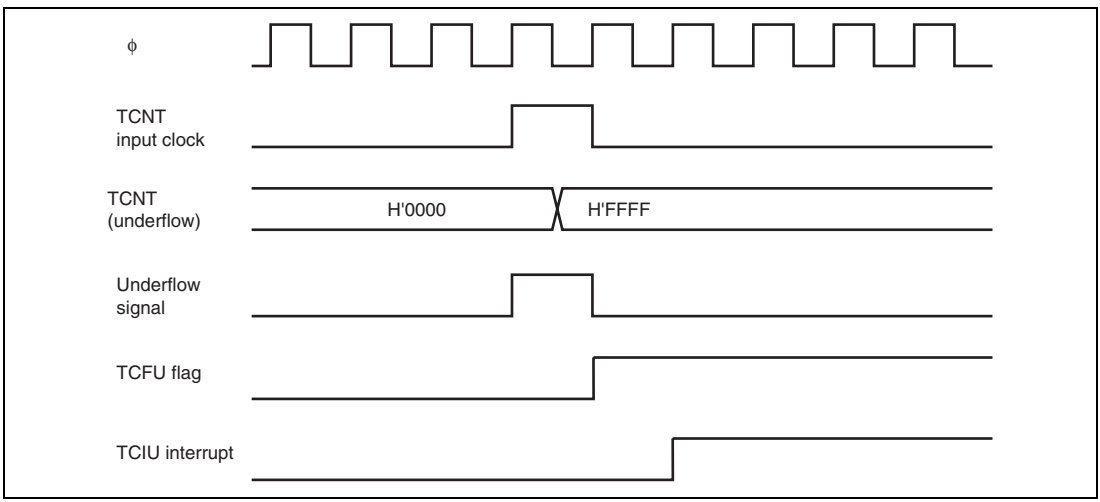


Figure 12.41 TCIU Interrupt Setting Timing

(4) Status Flag Clearing Timing

After a status flag is read as 1 by the CPU, it is cleared by writing 0 to it. When the DTC is activated, the flag is cleared automatically. Figure 12.42 shows the timing for status flag clearing by the CPU, and figure 12.43 shows the timing for status flag clearing by the DTC.

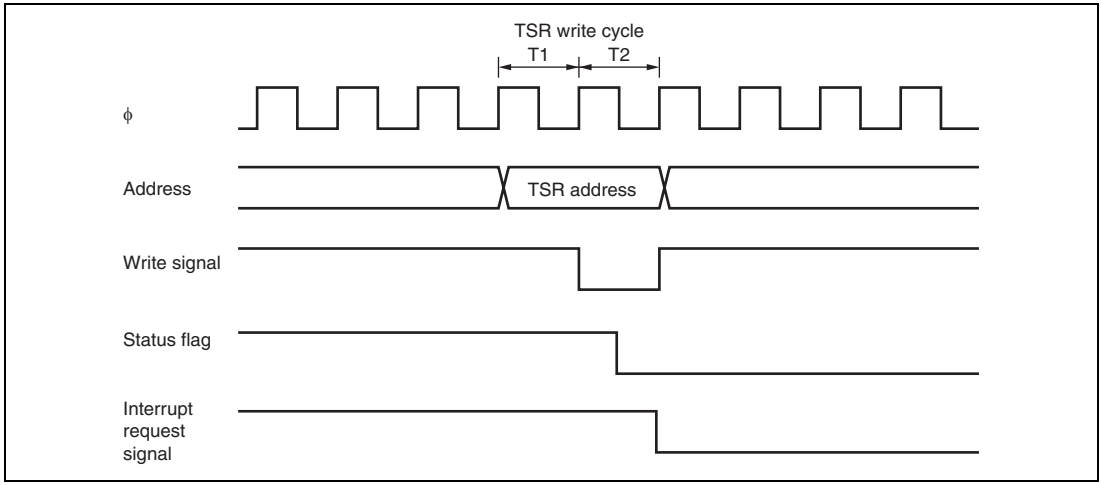


Figure 12.42 Timing for Status Flag Clearing by CPU

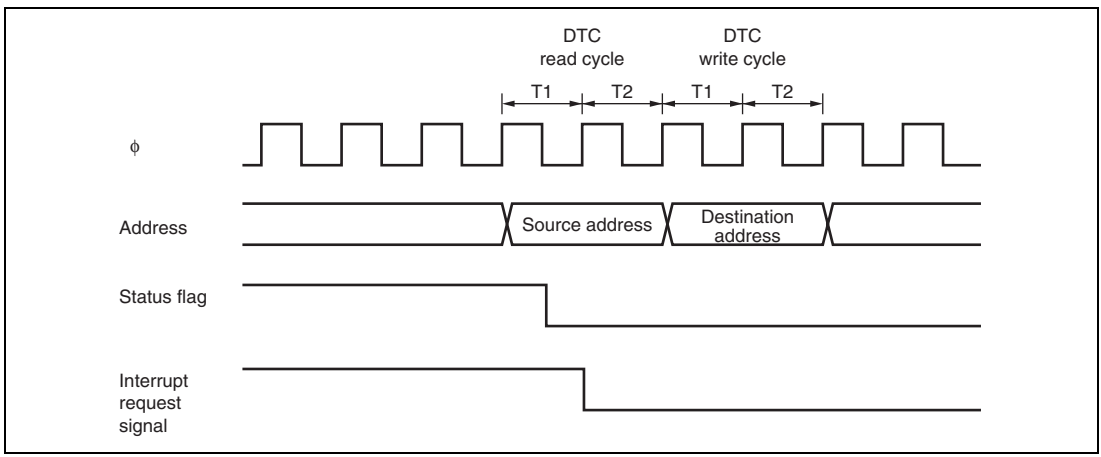


Figure 12.43 Timing for Status Flag Clearing by DTC Activation

12.8 Usage Notes

12.8.1 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 12.44 shows the input clock conditions in phase counting mode.

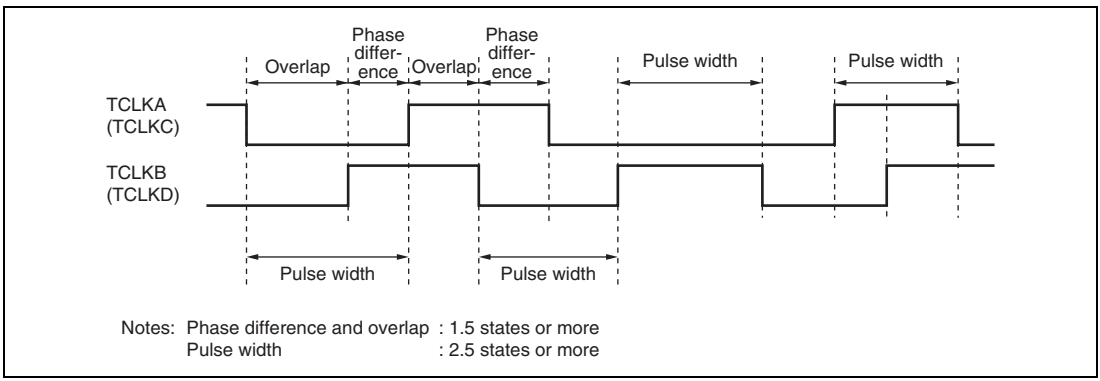


Figure 12.44 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

12.8.2 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

12.8.3 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 12.45 shows the timing in this case.

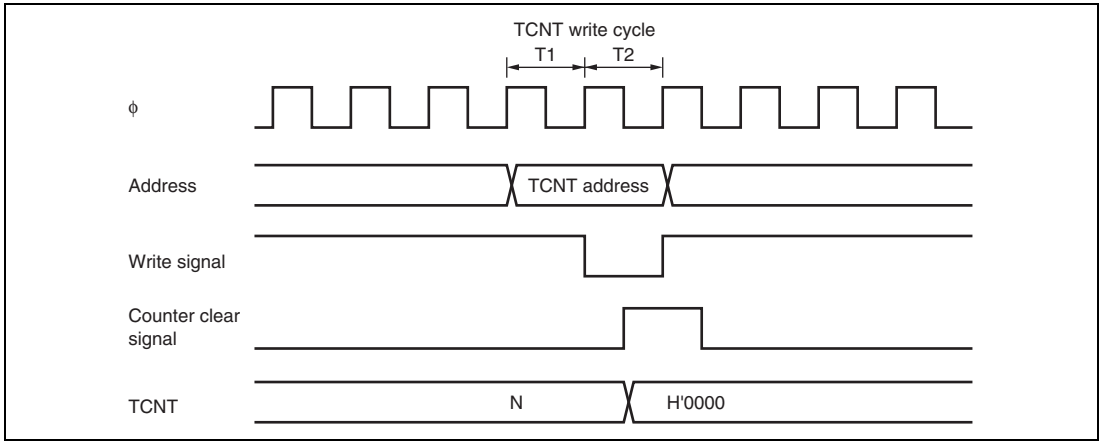


Figure 12.45 Conflict between TCNT Write and Clear Operations

12.8.4 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 12.46 shows the timing in this case.

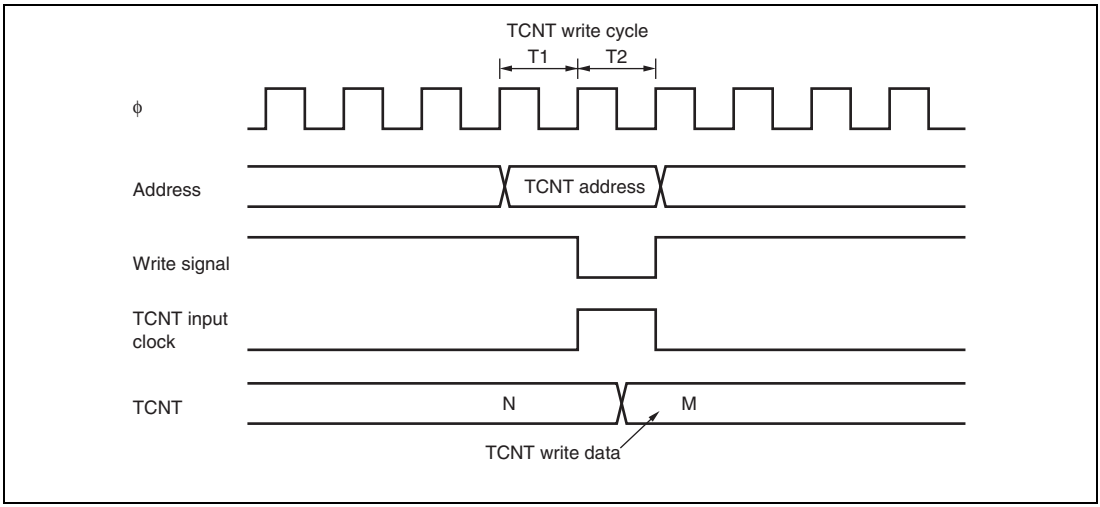


Figure 12.46 Conflict between TCNT Write and Increment Operations

12.8.5 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 12.47 shows the timing in this case.

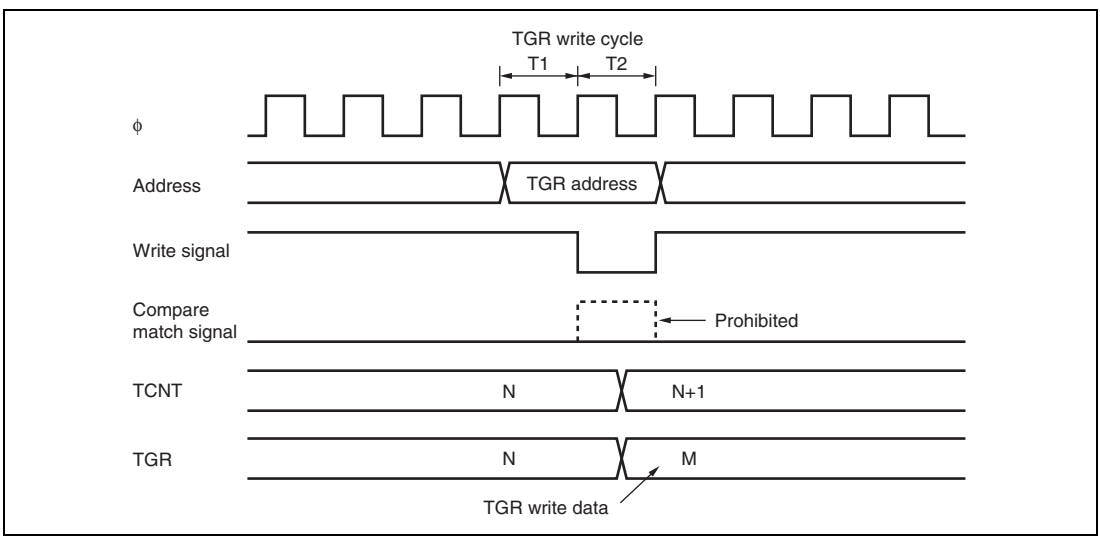


Figure 12.47 Conflict between TGR Write and Compare Match

12.8.6 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 12.48 shows the timing in this case.

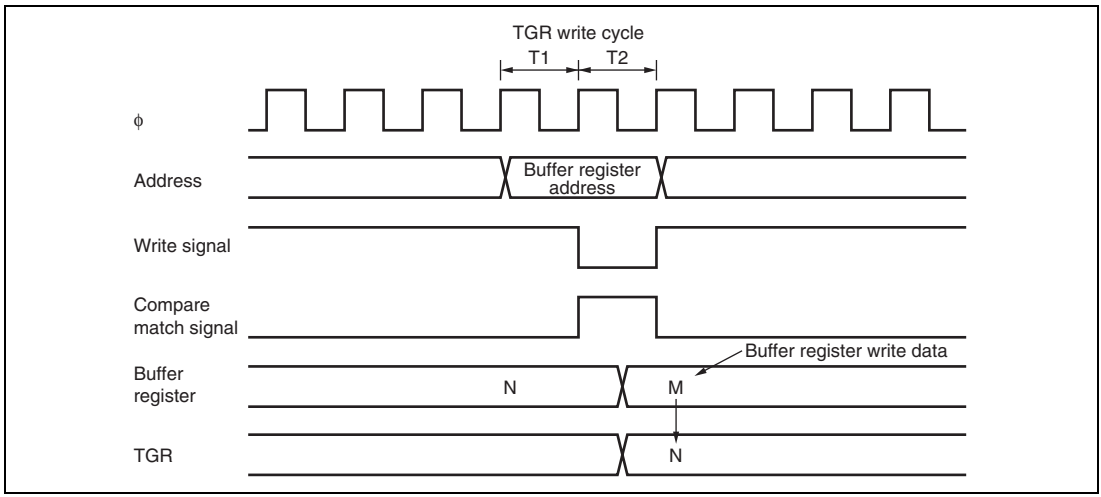


Figure 12.48 Conflict between Buffer Register Write and Compare Match

12.8.7 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 12.49 shows the timing in this case.

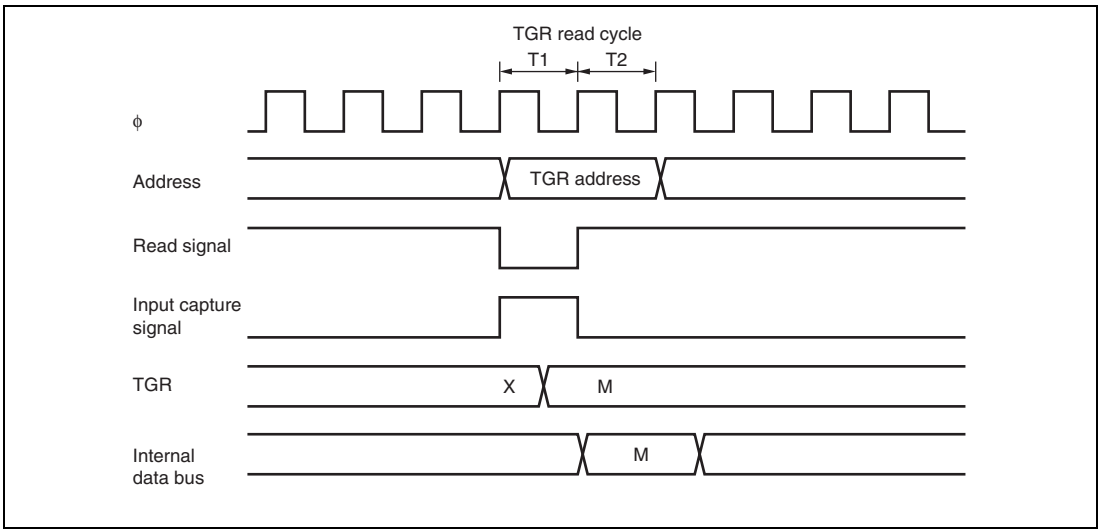


Figure 12.49 Conflict between TGR Read and Input Capture

12.8.8 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 12.50 shows the timing in this case.

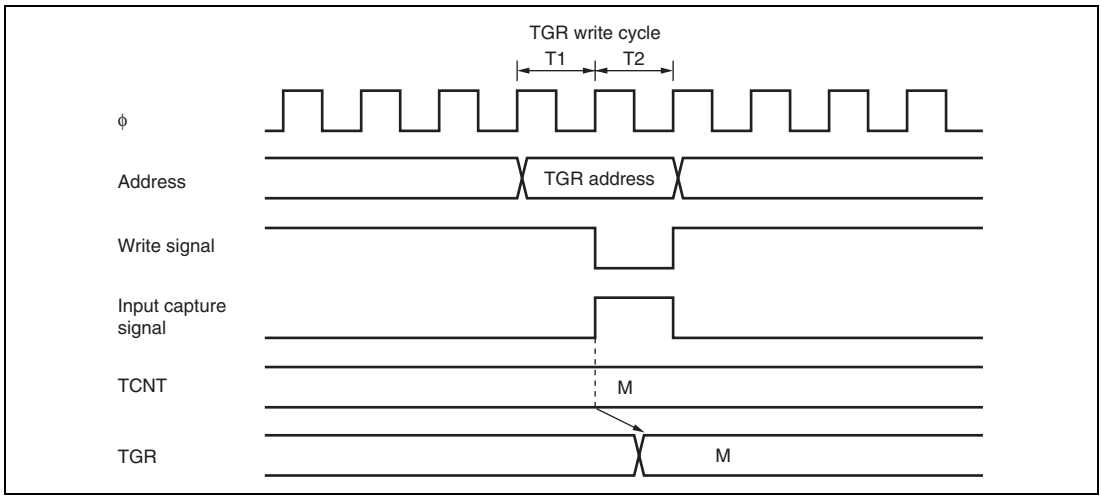


Figure 12.50 Conflict between TGR Write and Input Capture

12.8.9 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 12.51 shows the timing in this case.

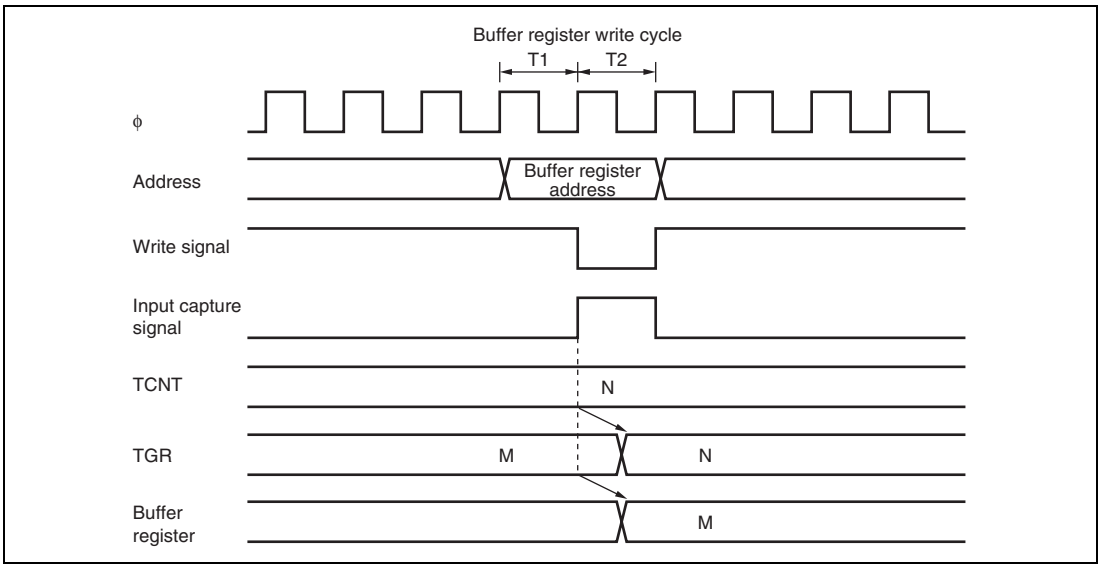


Figure 12.51 Conflict between Buffer Register Write and Input Capture

12.8.10 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 12.52 shows the operation timing when a TGR compare match is specified as the clearing source, and H'FFFF is set in TGR.

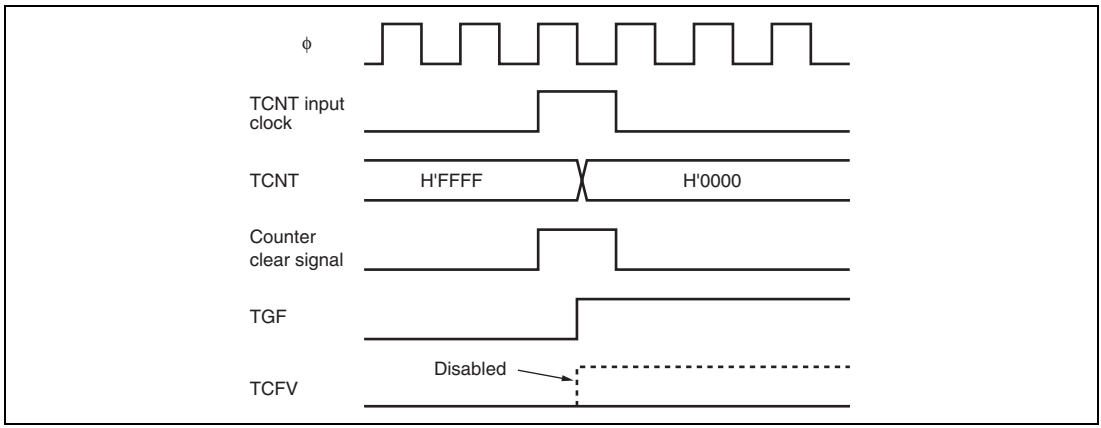


Figure 12.52 Conflict between Overflow and Counter Clearing

12.8.11 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 12.53 shows the operation timing when there is conflict between TCNT write and overflow.

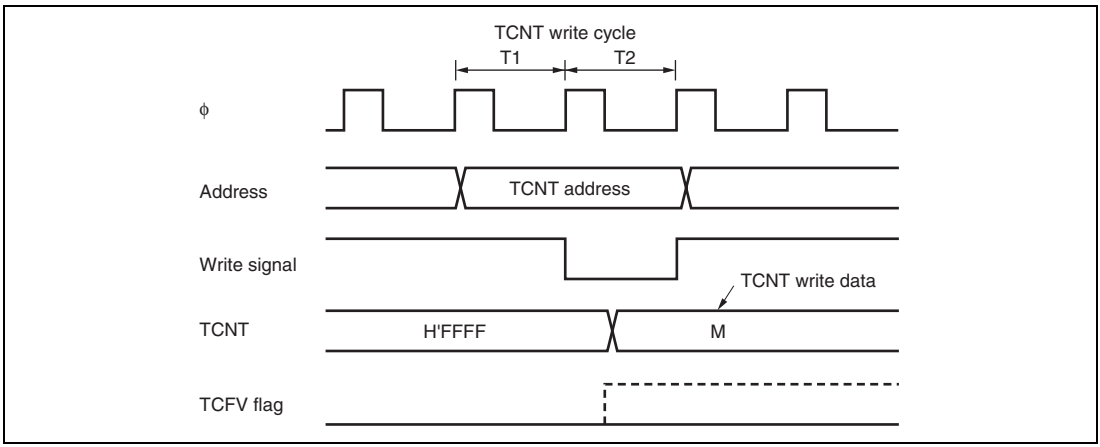


Figure 12.53 Conflict between TCNT Write and Overflow

12.8.12 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.

12.8.13 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 24, 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

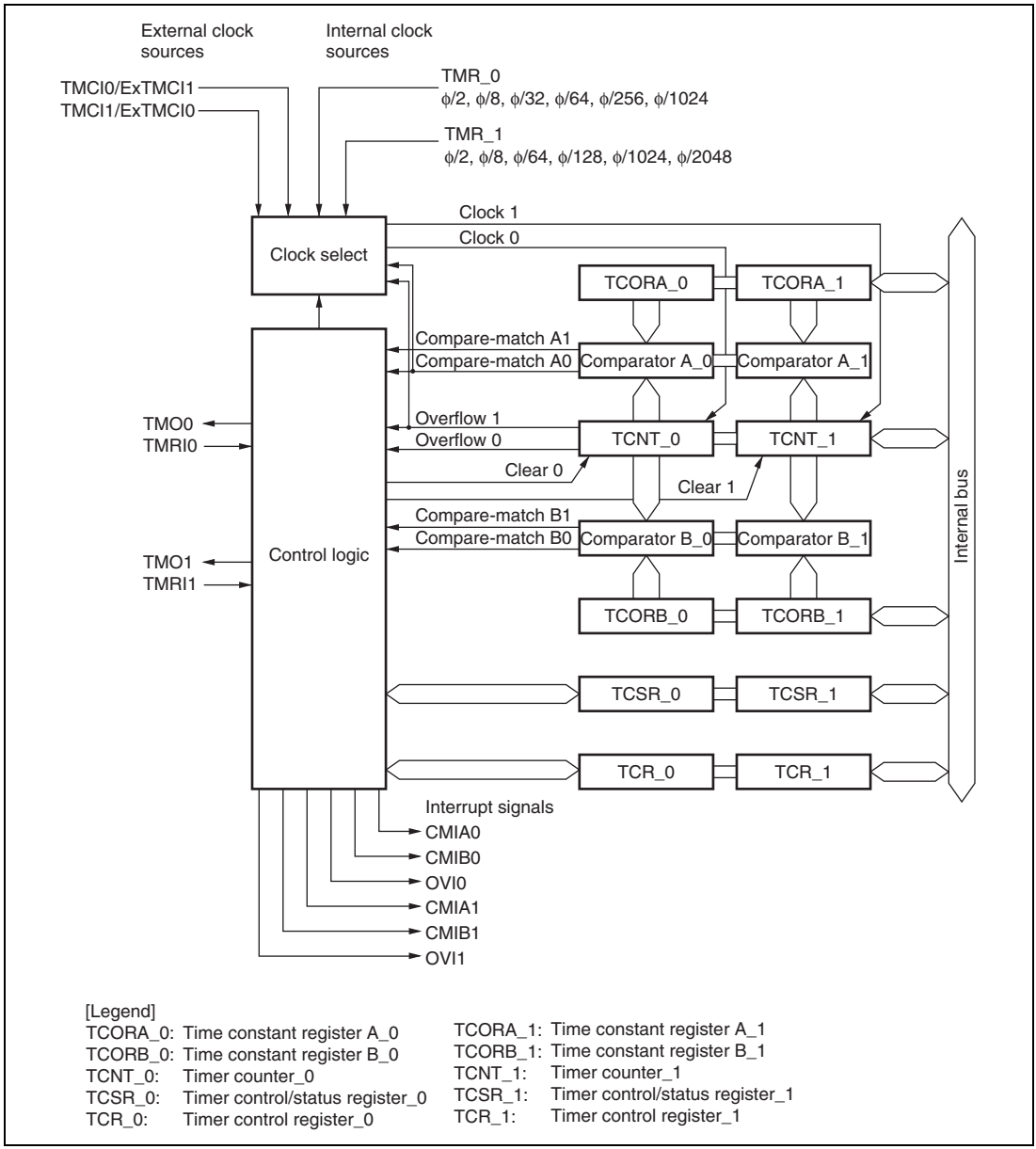
  TMR_0, TMR_1, and TMR_Y: Three types of interrupts: Compare-match A, compare-match B, and overflow

  TMR_X: Four types of interrupts: Compare-match A, compare match B, overflow, and input capture

- Selection of general ports for timer input/output
  - TMCI0/ExTMCI0, TMCI1/ExTMCI1, or TMIX/ExTMIX
  - TMIY/ExTMIY or TMOX/ExTMOX

Figures 13.1 and 13.2 show block diagrams of 8-bit timers.

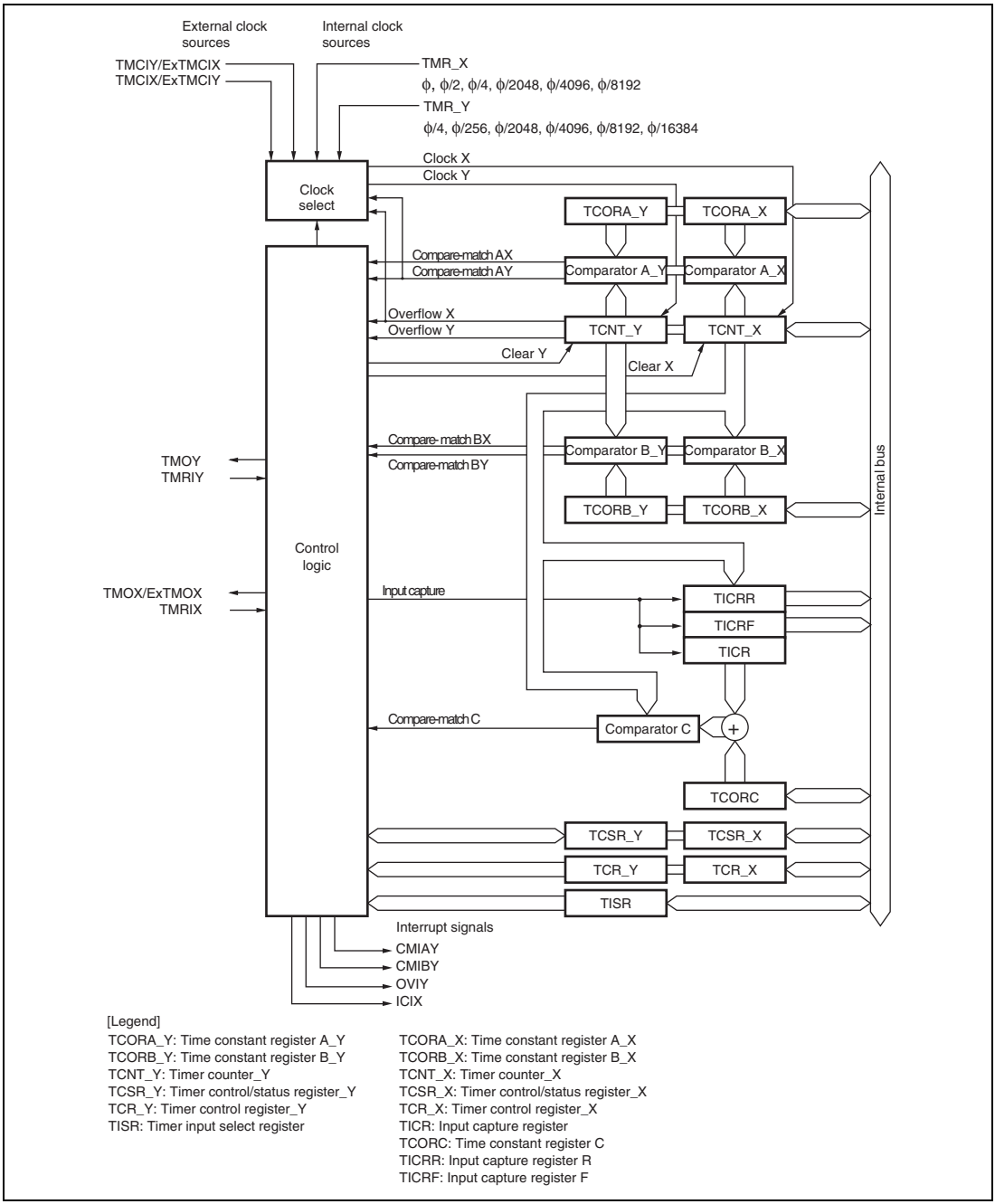
An input capture function is added to TMR_X.



[Legend]

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

Figure 13.1 Block Diagram of 8-Bit Timer (TMR_0 and TMR_1)



[Legend]

TCORA_Y: Time constant register A_Y
TCORB_Y: Time constant register B_Y
TCNT_Y: Timer counter_Y
TCSR_Y: Timer control/status register_Y
TCR_Y: Timer control register_Y
TISR: Timer input select register
TCORA_X: Time constant register A_X
TCORB_X: Time constant register B_X
TCNT_X: Timer counter_X
TCSR_X: Timer control/status register_X
TCR_X: Timer control register_X
TICR: Input capture register
TCORC: Time constant register C
TICRR: Input capture register R
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	Name	Symbol	I/O	Function
TMR_0	Timer output	TMO0	Output	Output controlled by compare-match
	Timer clock input	TMCI0, ExTMCI0	Input	External clock input for the counter TMCI0 or ExTMCI0 is selected for timer input.
	Timer reset input	TMRI0	Input	External reset input for the counter
TMR_1	Timer output	TMO1	Output	Output controlled by compare-match
	Timer clock input	TMCI1, ExTMCI1	Input	External clock input for the counter TMCI1 or ExTMCI1 is selected for timer input.
	Timer reset input	TMRI1	Input	External reset input for the counter
TMR_Y	Timer clock/reset input	TMIY, ExTMIY (TMCIY/TMRIY)	Input	External clock input/external reset input for the counter TMIY or ExTMIY is selected for timer input.
	Timer output	TMOY	Output	Output controlled by compare-match
TMR_X	Timer output	TMOX, ExTMOX	Output	Output controlled by compare-match TMOX or ExTMOX is selected for timer output.
	Timer clock/reset input	TMIX, ExTMIX (TMCIX/TMRIX)	Input	External clock input/external reset input for the counter TMIX or ExTMIX is selected for timer input.

Note: * For details, see section 8.17.1, Port Control Register 0 (PTCNT0).

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).

TMR_0

- Timer counter_0 (TCNT_0)
- Time constant register A_0 (TCORA_0)
- Time constant register B_0 (TCORB_0)
- Timer control register_0 (TCR_0)
- Timer control/status register_0 (TCSR_0)

TMR_1

- Timer counter_1 (TCNT_1)
- Time constant register A_1 (TCORA_1)
- Time constant register B_1 (TCORB_1)
- Timer control register_1 (TCR_1)
- Timer control/status register_1 (TCSR_1)

TMR_Y

- Timer counter_Y (TCNT_Y)
- Time constant register A_Y (TCORA_Y)
- Time constant register B_Y (TCORB_Y)
- Timer control register_Y (TCR_Y)
- Timer control/status register_Y (TCSR_Y)
- Timer input select register (TISR)
- Timer connection register S (TCONRS)

TMR_X

- Timer counter_X (TCNT_X)
- Time constant register A_X (TCORA_X)
- Time constant register B_X (TCORB_X)
- Timer control register_X (TCR_X)
- Timer control/status register_X (TCSR_X)
- Input capture register (TICR)
- Time constant register (TCORC)
- Input capture register R (TICRR)

- Input capture register F (TICRF)
- Timer connection register I (TCONRI)

For both TMR_Y and TMR_X

- Timer XY control register (TCRXY)

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

Bit	Bit Name	Initial Value	R/W	Description
4	CCLR1	0	R/W	Counter Clear 1, 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 13.2.
0	CKS0	0	R/W	

Table 13.2 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*

Channel	TCR CKS2	TCR CKS1	TCR CKS0	STCR ICKS1	STCR 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 13.2 Clock Input to TCNT and Count Condition (2)

Channel	TCR CKS2	TCR CKS1	TCR CKS0	TCRXY CKSX	TCRXY 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

Channel	TCR			TCRXY		Description
	CKS2	CKS1	CKS0	CKSX	CKSY	
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, 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, 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, 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, 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, 2
2	OS2	0	R/W	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)
1	OS1	0	R/W	Output Select 1, 0
0	OS0	0	R/W	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)

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, 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, 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 Input Select Register (TISR)

TISR permits or prohibits a signal source of external clock/reset input for the counter.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 1	R/(W)	Reserved The initial value should not be changed.
0	IS	0	R/W	Input Select Selects a timer clock/reset input pin (TMIY) as the signal source of external clock/reset input for the TMR_Y counter. 0: Input is prohibited 1: TMIY (TMCIY/TMRIY) is permitted for input

13.3.9 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.

13.3.10 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 13.3. 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 13.3 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	TISR		

13.3.11 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 13.2.
4	CKSY	0	R/W	TMR_Y Clock Select For details about selection, see table 13.2.
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.

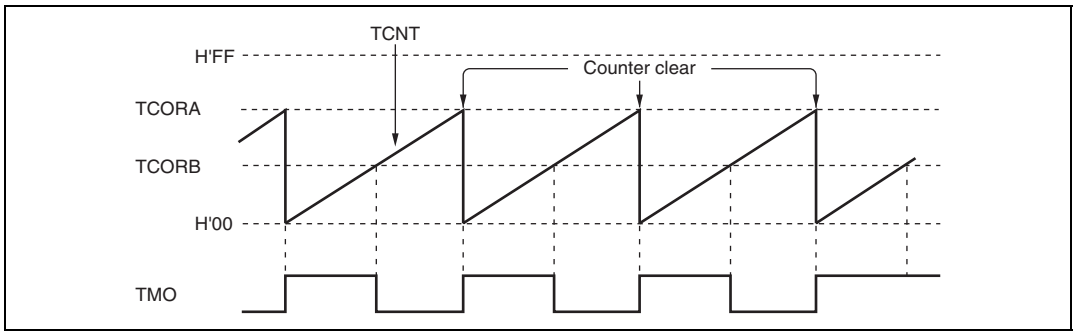


Figure 13.3 Pulse Output Example

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.

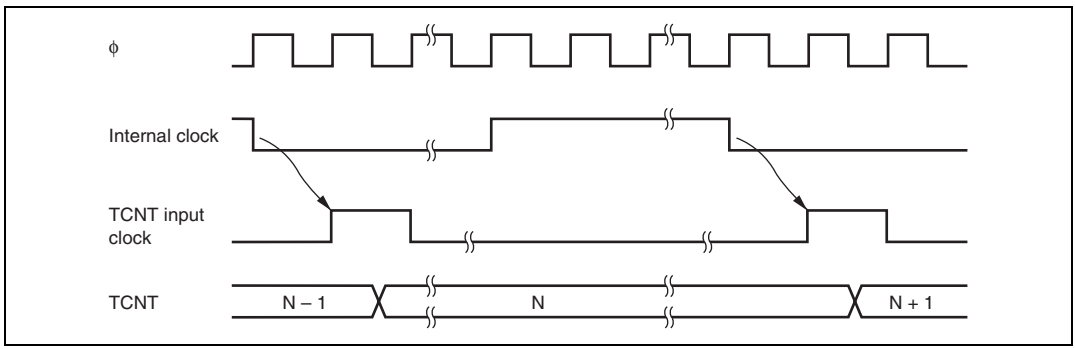


Figure 13.4 Count Timing for Internal Clock Input

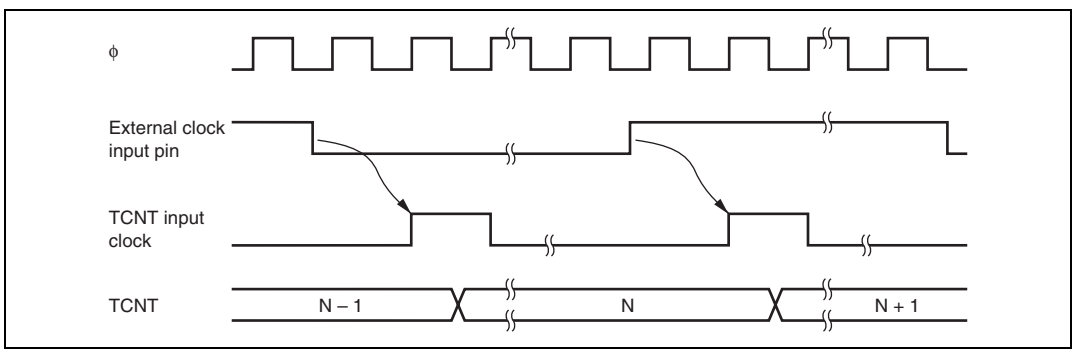


Figure 13.5 Count Timing for External Clock Input (Both Edges)

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.

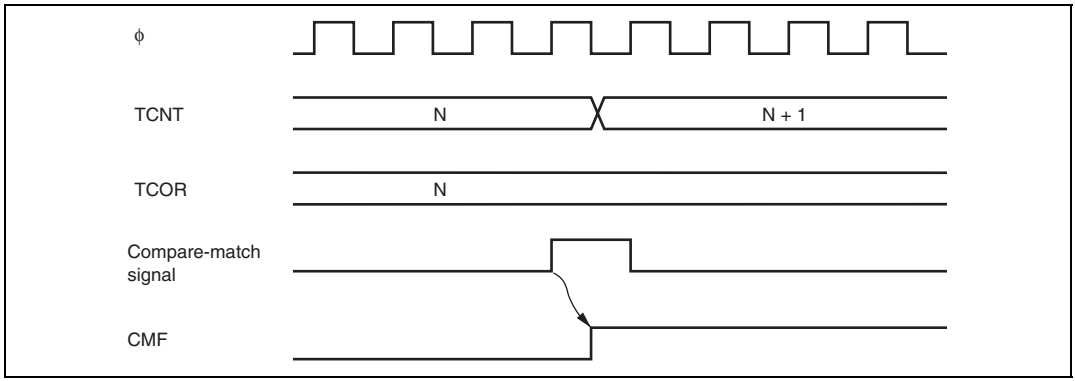


Figure 13.6 Timing of CMF Setting at Compare-Match

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.

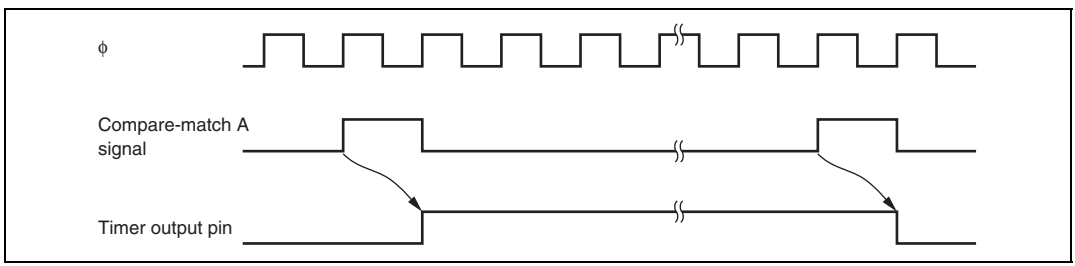


Figure 13.7 Timing of Toggled Timer Output by Compare-Match A Signal

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.

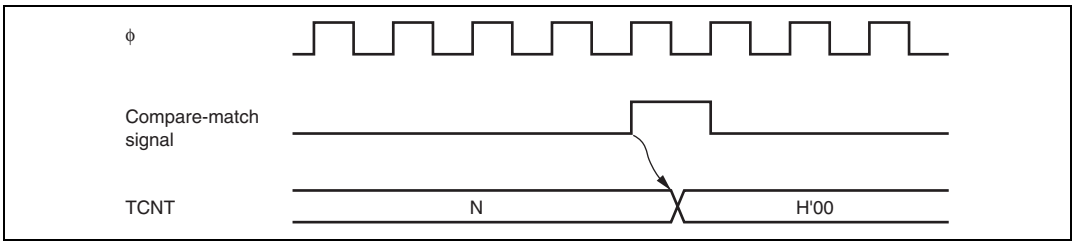


Figure 13.8 Timing of Counter Clear by Compare-Match

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.

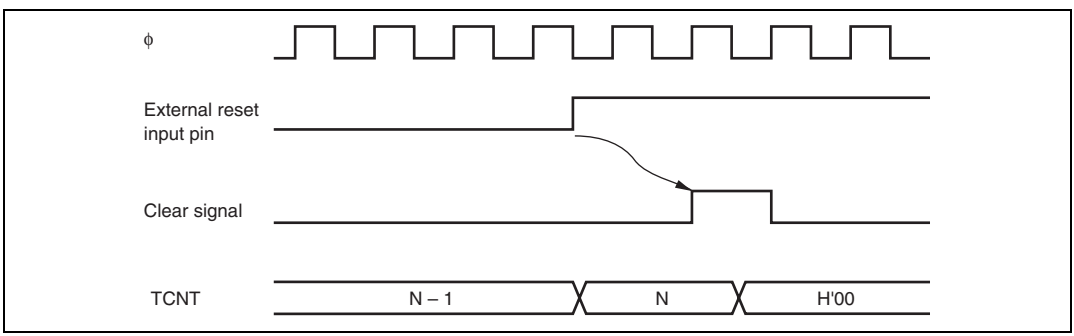


Figure 13.9 Timing of Counter Clear by External Reset Input

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.

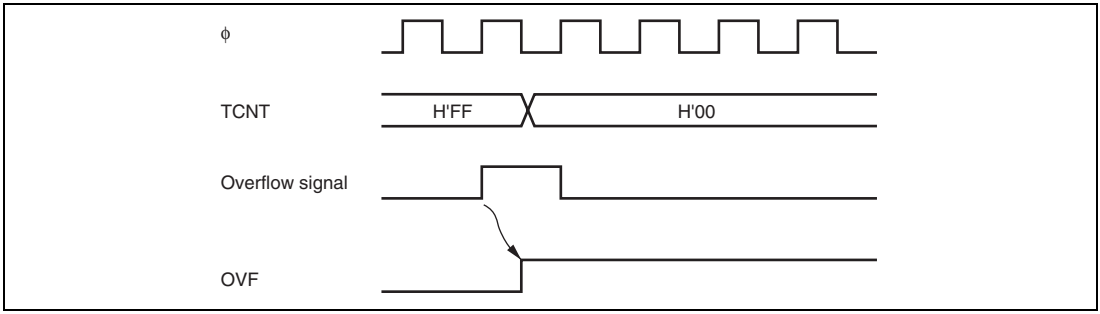


Figure 13.10 Timing of OVF Flag Setting

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.

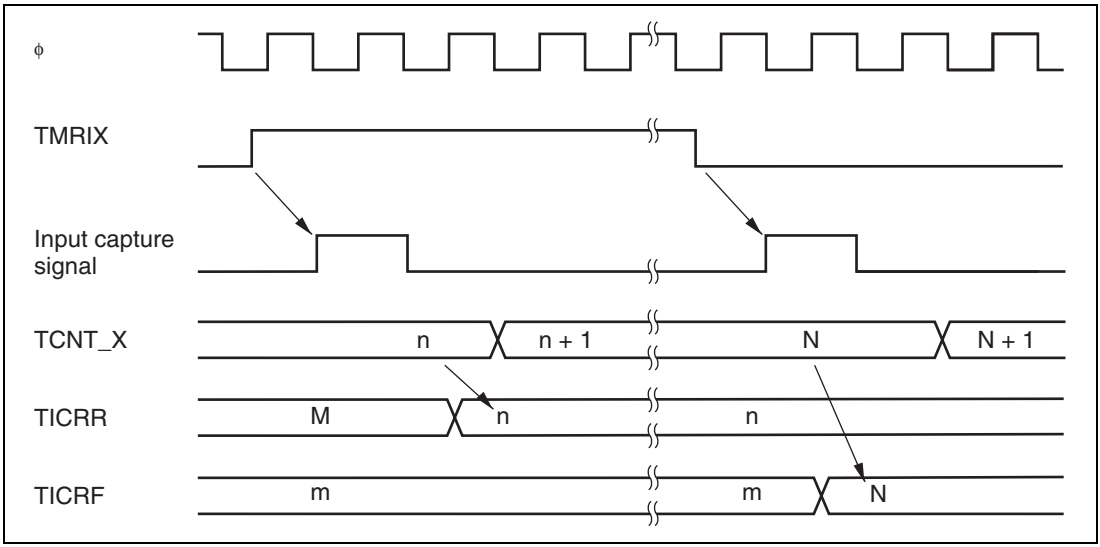


Figure 13.11 Timing of Input Capture Operation

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.

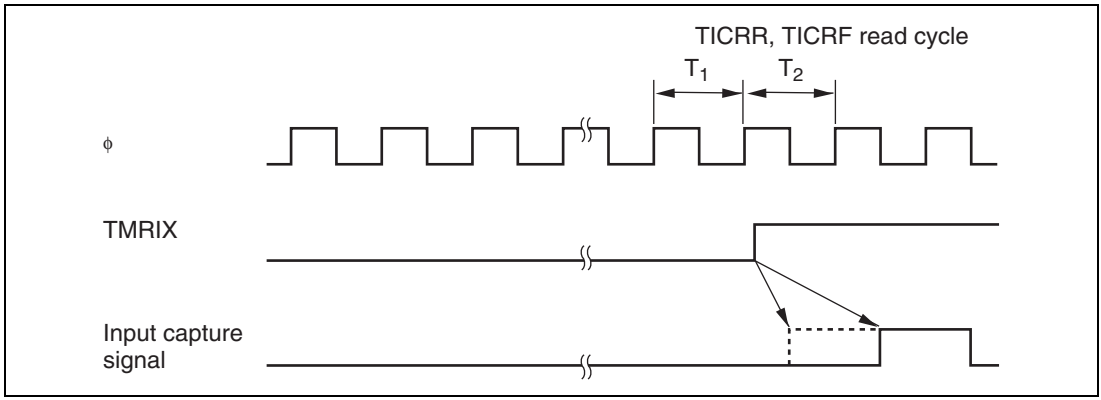


Figure 13.12 Timing of Input Capture Signal (Input capture signal is input during TICRR and TICRF read)

(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 13.4.

Table 13.4 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 13.5 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 13.5 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.

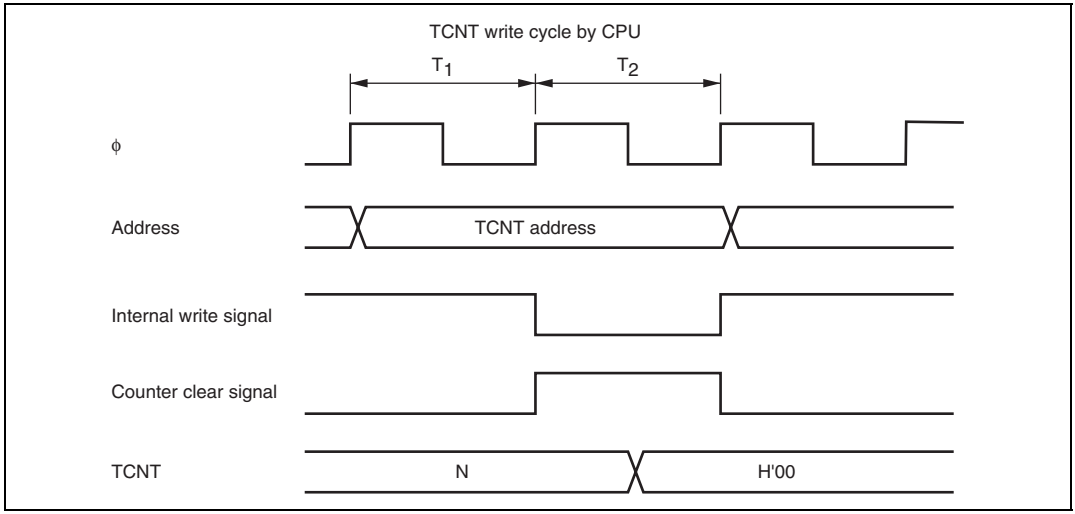


Figure 13.13 Conflict between TCNT Write and Clear

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.

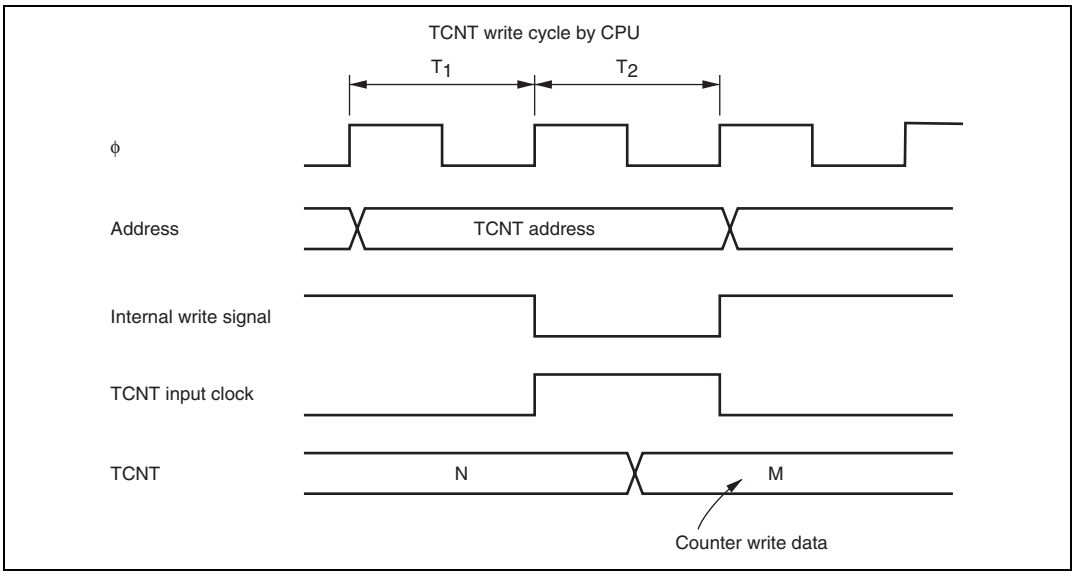


Figure 13.14 Conflict between TCNT Write and Count-Up

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.

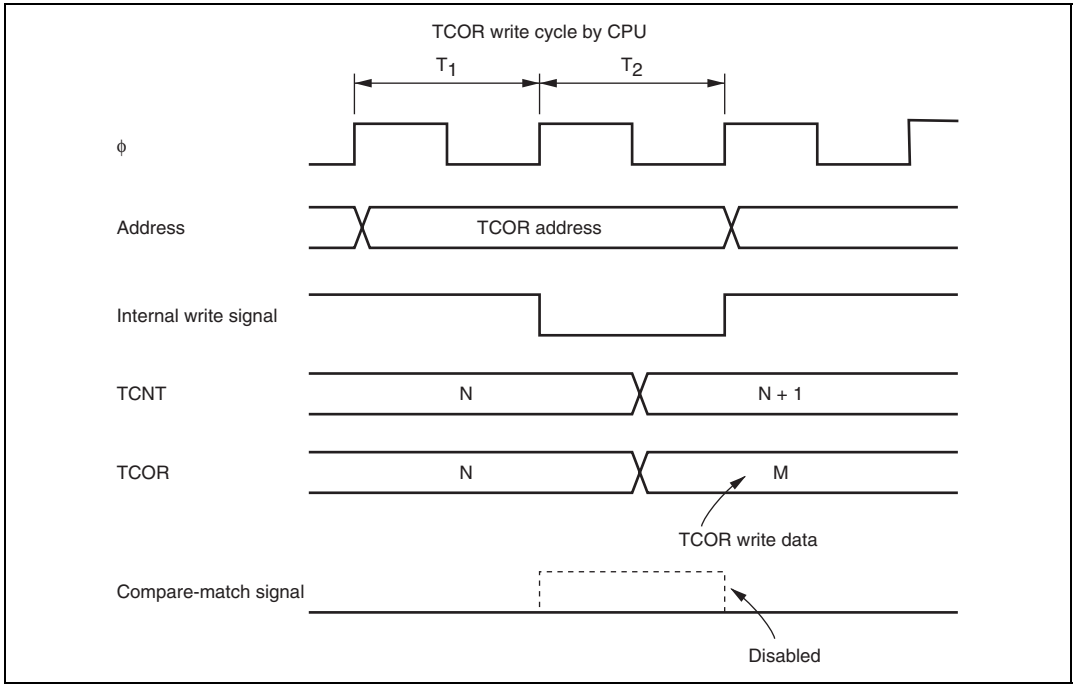


Figure 13.15 Conflict between TCOR Write and Compare-Match

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 13.6.

Table 13.6 Timer Output Priorities

Output Setting	Priority
Toggle output	High
1 output	
0 output	
No change	Low

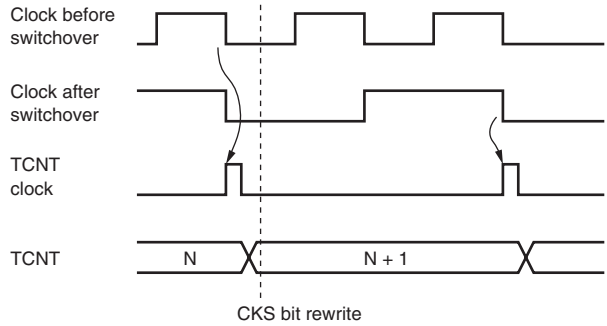
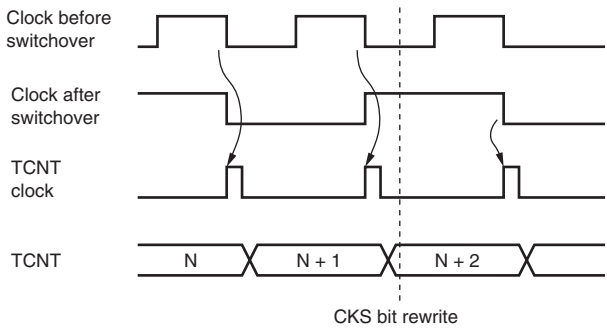
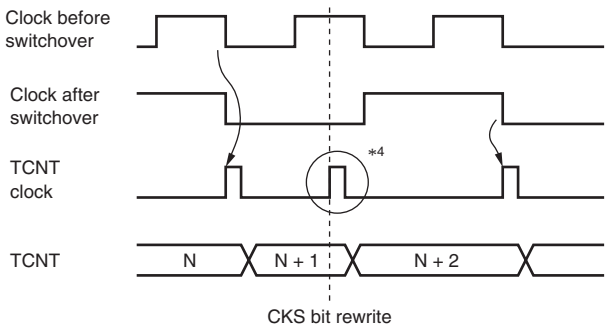
13.9.5 Switching of Internal Clocks and TCNT Operation

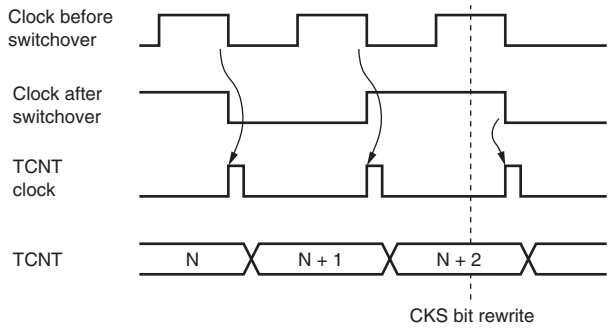
TCNT may increment erroneously when the internal clock is switched over. Table 13.7 shows the relationship between the timing at which the internal clock is switched (by writing to the CKS1 and CKS0 bits) and the TCNT operation.

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 13.7, 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 13.7 Switching of Internal Clocks and TCNT Operation

No.	Timing of Switchover by Means of CKS1 and CKS0 Bits	TCNT Clock Operation
1	Clock switching from low to low level*[1]	
2	Clock switching from low to high level*[2]	
3	Clock switching from high to low level*[3]	

No.	Timing of Switchover by Means of CKS1 and CKS0 Bits	TCNT Clock Operation
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 24, Power-Down Modes.

Section 14 Watchdog Timer (WDT)

This LSI incorporates two watchdog timer channels (WDT_0 and WDT_1). The watchdog timer can output an overflow signal ($\overline{\text{RESO}}$) externally if a system crash prevents the CPU from writing to the timer counter, thus allowing it to overflow. Simultaneously, it can generate an internal reset signal or an internal NMI interrupt signal.

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

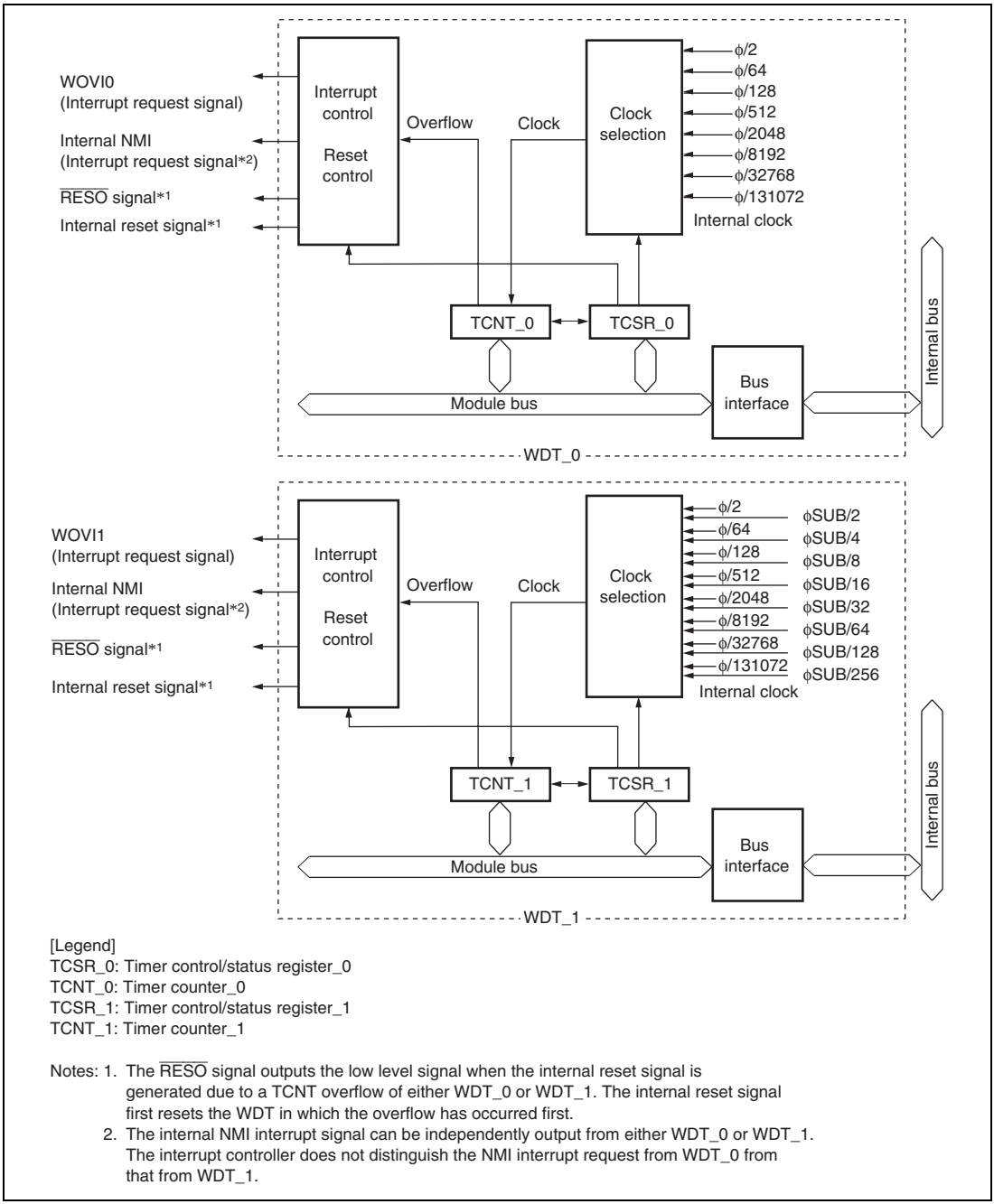
- Selectable from eight (WDT_0) or 16 (WDT_1) counter input clocks.
- Switchable between watchdog timer mode and interval timer mode

Watchdog Timer Mode:

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

Internal Timer Mode:

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



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

Notes: 1. The RESO signal outputs the low level signal when the internal reset signal is generated due to a TCNT overflow of either WDT_0 or WDT_1. The internal reset signal first resets the WDT in which the overflow has occurred first.
2. The internal NMI interrupt signal can be independently output from either WDT_0 or WDT_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	Symbol	I/O	Function
Reset output pin	$\overline{\text{RESO}}$	Output	Outputs the counter overflow signal in watchdog timer mode
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).

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

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.
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 frequency for ϕ = 20 MHz is enclosed in parentheses.
0	CKS0	0	R/W	
				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)

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, and the low level signal is simultaneously output from the $\overline{RESO}$ pin for 132 states, 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. Here, the output from the $\overline{RESO}$ pin remains high.

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.

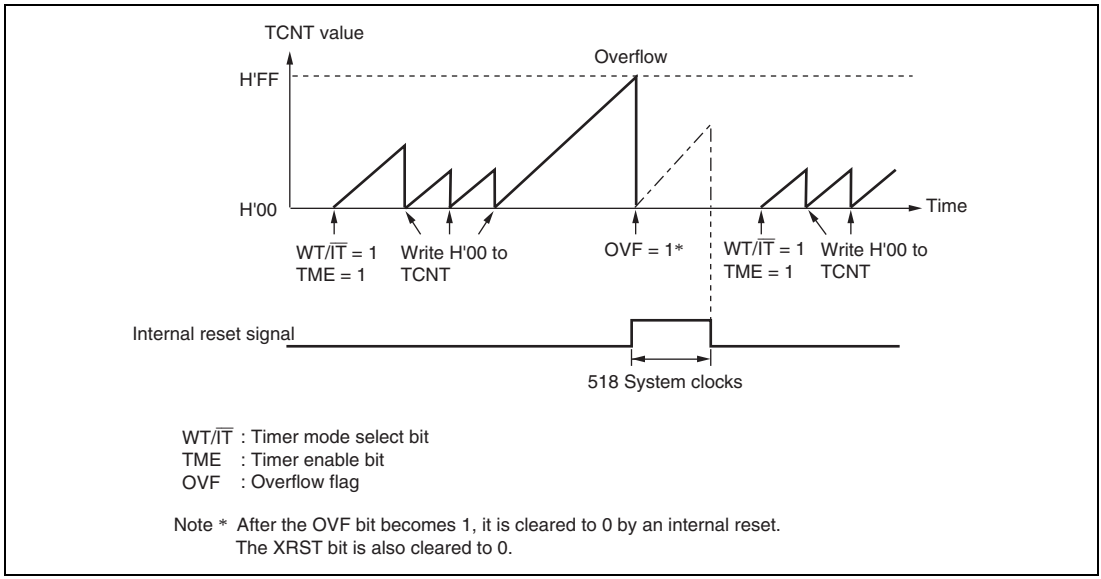


Figure 14.2 Watchdog Timer Mode (RST/$\overline{\text{NMI}}$ = 1) Operation

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.

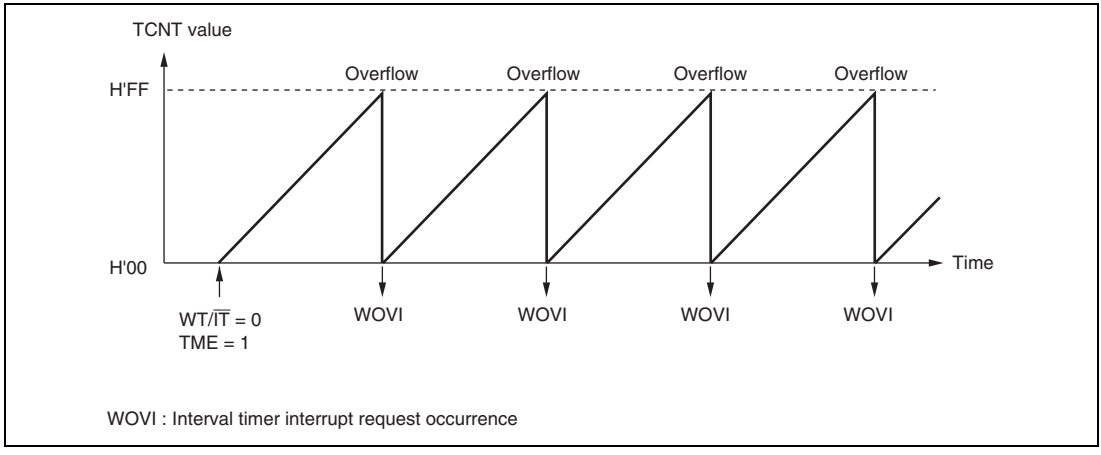


Figure 14.3 Interval Timer Mode Operation

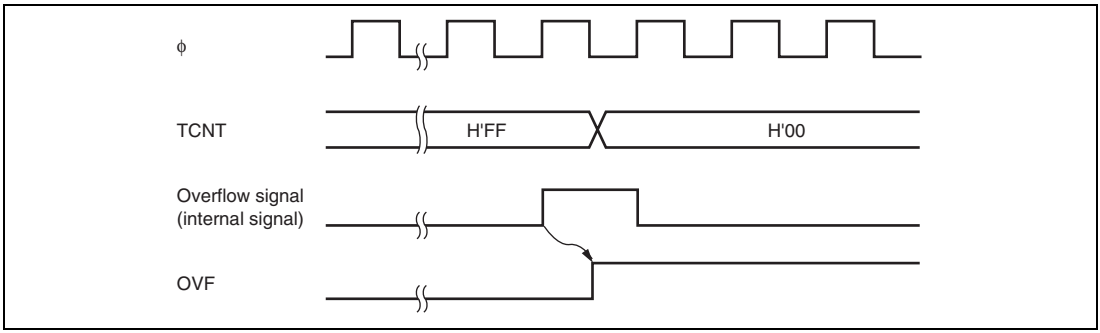


Figure 14.4 OVF Flag Set Timing

14.4.3 $\overline{\text{RESO}}$ Signal Output Timing

When TCNT overflows in watchdog timer mode, the OVF flag in TCSR is set to 1. When the RST/$\overline{\text{NMI}}$ bit is 1 here, the internal reset signal is generated for the entire LSI. At the same time, the low level signal is output from the $\overline{\text{RESO}}$ pin. The timing is shown in figure 14.5.

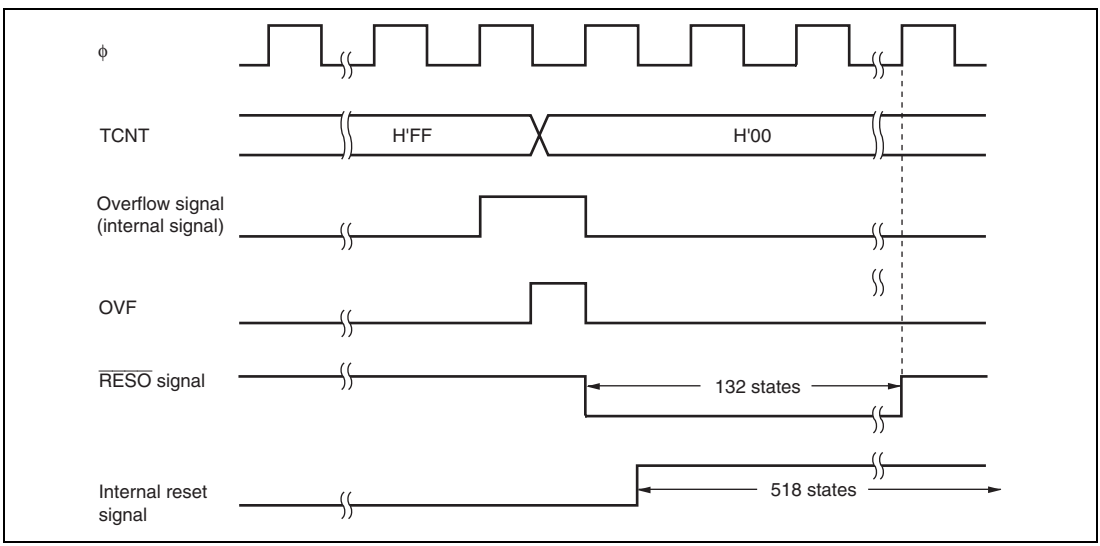


Figure 14.5 Output Timing of $\overline{\text{RESO}}$ signal

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 14.2 WDT Interrupt Source

Name	Interrupt Source	Interrupt Flag	DTC Activation
WOVI	TCNT overflow	OVF	Disable

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 14.6 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.

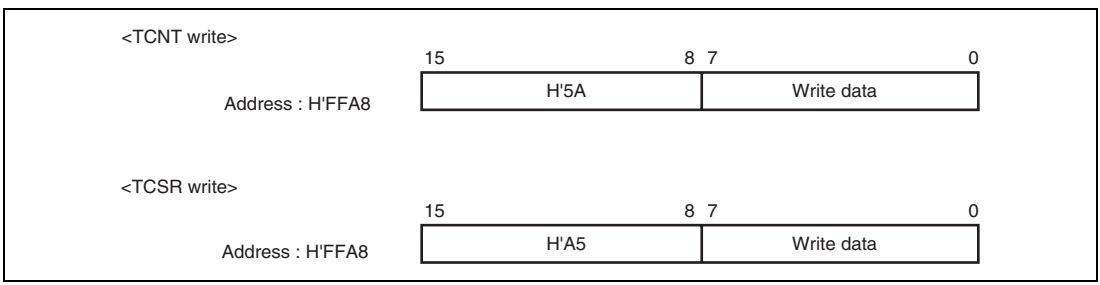


Figure 14.6 Writing to TCNT and TCSR (WDT_0)

(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 14.7 shows this operation.

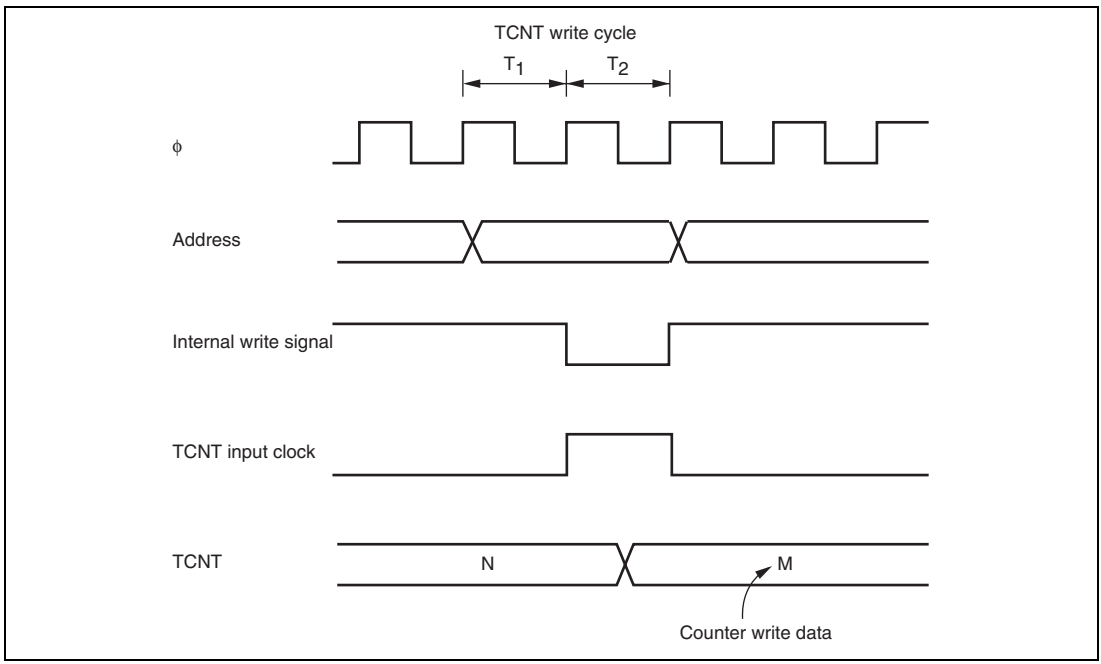


Figure 14.7 Conflict between TCNT Write and Increment

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.

14.6.6 System Reset by $\overline{\text{RESO}}$ Signal

Inputting the $\overline{\text{RESO}}$ output signal to the $\overline{\text{RES}}$ pin of this LSI prevents the LSI from being initialized correctly; the $\overline{\text{RESO}}$ signal must not be logically connected to the $\overline{\text{RES}}$ pin of the LSI. To reset the entire system by the $\overline{\text{RESO}}$ signal, use the circuit as shown in figure 14.8.

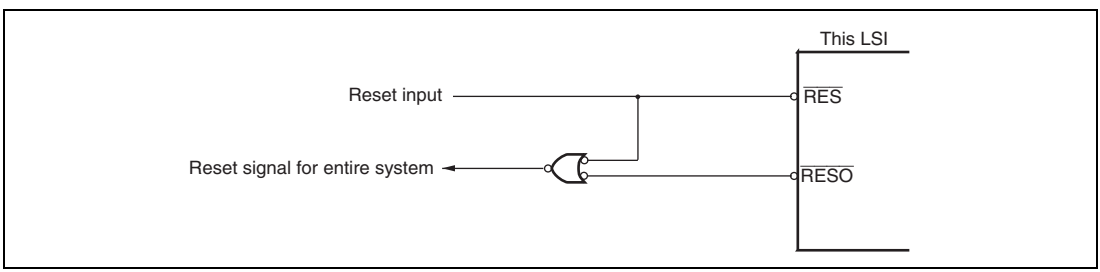


Figure 14.8 Sample Circuit for Resetting the System by the $\overline{\text{RESO}}$ Signal

Section 15 Serial Communication Interface (SCI, IrDA)

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. Communication using the waveform based on the Infrared Data Association (IrDA) standard version 1.0 can also be handled.

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.

  The transmit-data-empty and receive-data-full interrupt sources can activate DTC.

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.

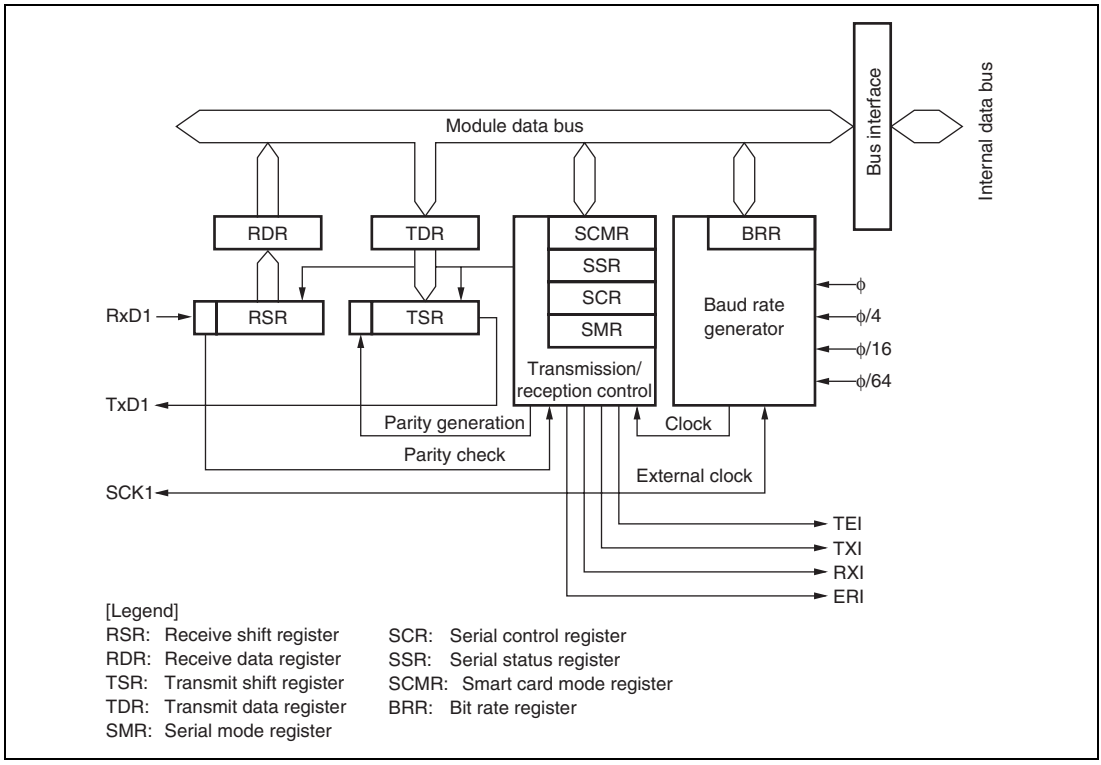


Figure 15.1 Block Diagram of SCI

15.2 Input/Output Pins

Table 15.1 shows the input/output pins for each SCI channel.

Table 15.1 Pin Configuration

Channel	Symbol*	Input/Output	Function
1	SCK1	Input/Output	Channel 1 clock input/output
	RxD1/IrRxD	Input	Channel 1 receive data input (normal/IrDA)
	TxD1/IrTxD	Output	Channel 1 transmit data output (normal/IrDA)
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.

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

Note: * KBCOMP is available in SCI_1.

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.

- 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 1,0
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)).

- 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 1,0
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, 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 15.9, Interrupt Sources. Some bits in SCR 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	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, 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, 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 conditions] • When 0 is written to TDRE after reading TDRE = 1 • When a TXI interrupt request is issued allowing DTC to write data to TDR
6	RDRF	0	R/(W)*	Receive Data Register Full Indicates that receive data is stored in RDR. [Setting condition] • When serial reception ends normally and receive data is transferred from RSR to RDR [Clearing conditions] • When 0 is written to RDRF after reading RDRF = 1 • When an RXI interrupt request is issued allowing DTC to read data from RDR The RDRF flag is not affected and retains its previous value when the RE bit in SCR is cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
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
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 conditions] • When 0 is written to TDRE after reading TDRE = 1 • When a TXI interrupt request is issued allowing DTC to write data to TDR

Bit	Bit Name	Initial Value	R/W	Description
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 conditions] • When 0 is written to TDRE after reading TDRE = 1 • When a TXI interrupt request is issued allowing DTC to write data to TDR

Bit	Bit Name	Initial Value	R/W	Description
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 conditions] • When 0 is written to RDRF after reading RDRF = 1 • When an RXI interrupt request is issued allowing DTC to read data from RDR 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
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

Bit	Bit Name	Initial Value	R/W	Description
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 conditions] • When 0 is written to TDRE after reading TDRE = 1 • When a TXI interrupt request is issued allowing DTC to write the next data to TDR
1	MPB	0	R	Multiprocessor Bit Not used in smart card interface mode.
0	MPBT	0	R/W	Multiprocessor Bit Transfer Write 0 to this bit in smart card interface 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 15.2 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, and it can be read from or written to by the CPU at all times.

Table 15.2 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 15.3 shows sample N settings in BRR in normal asynchronous mode. Table 15.4 shows the maximum bit rate settable for each frequency. Table 15.6 and 15.8 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 15.5 and 15.7 show the maximum bit rates with external clock input.

Table 15.3 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (1)

	Operating Frequency ϕ (MHz)														
	4			4.9152			5			6			6.144		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	70	0.03	2	86	0.31	2	88	–0.25	2	106	–0.44	2	108	0.08
150	1	207	0.16	1	255	0.00	2	64	0.16	2	77	0.16	2	79	0.00
300	1	103	0.16	1	127	0.00	1	129	0.16	1	155	0.16	1	159	0.00
600	0	207	0.16	0	255	0.00	1	64	0.16	1	77	0.16	1	79	0.00
1200	0	103	0.16	0	127	0.00	0	129	0.16	0	155	0.16	0	159	0.00
2400	0	51	0.16	0	63	0.00	0	64	0.16	0	77	0.16	0	79	0.00
4800	0	25	0.16	0	31	0.00	0	32	–1.36	0	38	0.16	0	39	0.00
9600	0	12	0.16	0	15	0.00	0	15	1.73	0	19	–2.34	0	19	0.00
19200	—	—	—	0	7	0.00	0	7	1.73	0	9	–2.34	0	9	0.00
31250	0	3	0.00	0	4	–1.70	0	4	0.00	0	5	0.00	0	5	2.40
38400	—	—	—	0	3	0.00	0	3	1.73	0	4	–2.34	0	4	0.00

	Operating Frequency ϕ (MHz)											
	7.3728			8			9.8304			10		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	130	–0.07	2	141	0.03	2	174	–0.26	2	177	–0.25
150	2	95	0.00	2	103	0.16	2	127	0.00	2	129	0.16
300	1	191	0.00	1	207	0.16	1	255	0.00	2	64	0.16
600	1	95	0.00	1	103	0.16	1	127	0.00	1	129	0.16
1200	0	191	0.00	0	207	0.16	0	255	0.00	1	64	0.16
2400	0	95	0.00	0	103	0.16	0	127	0.00	0	129	0.16
4800	0	47	0.00	0	51	0.16	0	63	0.00	0	64	0.16
9600	0	23	0.00	0	25	0.16	0	31	0.00	0	32	–1.36
19200	0	11	0.00	0	12	0.16	0	15	0.00	0	15	1.73
31250	—	—	—	0	7	0.00	0	9	–1.70	0	9	0.00
38400	0	5	0.00	—	—	—	0	7	0.00	0	7	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 15.3 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (2)

	Operating Frequency ϕ (MHz)											
	12			12.288			14			14.7456		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	212	0.03	2	217	0.08	2	248	–0.17	3	64	0.70
150	2	155	0.16	2	159	0.00	2	181	0.16	2	191	0.00
300	2	77	0.16	2	79	0.00	2	90	0.16	2	95	0.00
600	1	155	0.16	1	159	0.00	1	181	0.16	1	191	0.00
1200	1	77	0.16	1	79	0.00	1	90	0.16	1	95	0.00
2400	0	155	0.16	0	159	0.00	0	181	0.16	0	191	0.00
4800	0	77	0.16	0	79	0.00	0	90	0.16	0	95	0.00
9600	0	38	0.16	0	39	0.00	0	45	–0.93	0	47	0.00
19200	0	19	–2.34	0	19	0.00	0	22	–0.93	0	23	0.00
31250	0	11	0.00	0	11	2.40	0	13	0.00	0	14	–1.70
38400	0	9	–2.34	0	9	0.00	—	—	—	0	11	0.00

	Operating Frequency ϕ (MHz)														
	16			17.2032			18			19.6608			20		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	70	0.03	3	75	0.48	3	79	–0.12	3	86	0.31	3	88	–0.25
150	2	207	0.16	2	223	0.00	2	233	0.16	2	255	0.00	3	64	0.16
300	2	103	0.16	2	111	0.00	2	116	0.16	2	127	0.00	2	129	0.16
600	1	207	0.16	1	223	0.00	1	233	0.16	1	255	0.00	2	64	0.16
1200	1	103	0.16	1	111	0.00	1	116	0.16	1	127	0.00	1	129	0.16
2400	0	207	0.16	0	223	0.00	0	233	0.16	0	255	0.00	1	64	0.16
4800	0	103	0.16	0	111	0.00	0	116	0.16	0	127	0.00	0	129	0.16
9600	0	51	0.16	0	55	0.00	0	58	–0.69	0	63	0.00	0	64	0.16
19200	0	25	0.16	0	27	0.00	0	28	1.02	0	31	0.00	0	32	–1.36
31250	0	15	0.00	0	16	1.20	0	17	0.00	0	19	–1.70	0	19	0.00
38400	0	12	0.16	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 15.4 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
4	125000	0	0	12	375000	0	0
44.9152	153600	0	0	12.288	384000	0	0
5	156250	0	0	14	437500	0	0
6	187500	0	0	14.7456	460800	0	0
6.144	192000	0	0	16	500000	0	0
7.3728	230400	0	0	17.2032	537600	0	0
8	250000	0	0	18	562500	0	0
9.8304	307200	0	0	19.6608	614400	0	0
10	312500	0	0	20	625000	0	0

Table 15.5 Maximum Bit Rate with External Clock Input (Asynchronous Mode)

ϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bit/s)	ϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bit/s)
4	1.0000	62500	12	3.0000	187500
4.9152	1.2288	76800	12.288	3.0720	192000
5	1.2500	78125	14	3.5000	218750
6	15.000	93750	14.7456	3.6864	230400
6.144	1.5360	96000	16	4.0000	250000
7.3728	1.8432	115200	17.2032	4.3008	268800
8	2.0000	125000	18	4.5000	281250
9.8304	2.4576	153600	19.6608	4.9152	307200
10	2.5000	156250	20	5.0000	312500

Table 15.6 BRR Settings for Various Bit Rates (Clocked Synchronous Mode)

Bit Rate (bit/s)	Operating Frequency φ (MHz)									
	4		8		10		16		20	
	n	N	n	N	n	N	n	N	n	N
110	—	—								
250	2	29	3	124	—	—	3	249		
500	2	124	2	249	—	—	3	124	—	—
1k	1	249	2	124	—	—	2	249	—	—
2.5k	1	99	1	199	1	249	2	99	2	124
5k	0	199	1	99	1	124	1	199	1	249
10k	0	99	0	199	0	249	1	99	1	124
25k	0	39	0	79	0	99	0	159	0	199
50k	0	19	0	39	0	49	0	79	0	99
100k	0	9	0	19	0	24	0	39	0	49
250k	0	3	0	7	0	9	0	15	0	19
500k	0	1*	0	3	0	4	0	7	0	9
1M	0	0	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 15.7 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)
4	0.6667	666666.7	14	2.3333	2333333.3
6	1.0000	1000000.0	16	2.6667	2666666.7
8	1.3333	1333333.3	18	3.0000	3000000.0
10	1.6667	1666666.7	20	3.3333	3333333.3
12	2.0000	2000000.0			

Table 15.8 BRR Settings for Various Bit Rates (Smart Card Interface Mode, n = 0, s = 372)

	Operating Frequency φ (MHz)														
Bit Rate	7.1424			10.00			13.00			14.2848			16.00		
(bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	0	0.00	0	1	30	0	1	-8.99	0	1	0.00	0	1	12.01

	Operating Frequency φ (MHz)					
Bit Rate	18.00			20.00		
(bit/s)	n	N	Error (%)	n	N	Error (%)
9600	0	2	-15.99	0	2	-6.65

Table 15.9 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
7.1424	9600	0	0	16.00	21505	0	0
10.00	13441	0	0	18.00	24194	0	0
13.00	17473	0	0	20.00	26882	0	0
14.2848	19200	0	0				

15.3.10 Keyboard Comparator Control Register (KBCOMP)

KBCOMP controls IrDA operation of SCI_1.

Bit	Bit Name	Initial Value	R/W	Description
7	IrE	0	R/W	IrDA Enable Specifies SCI_1 I/O pins for either normal SCI or IrDA. 0: TxD1/IrTxD and RxD1/IrRxD pins function as TxD1 and RxD1 pins, respectively 1: TxD1/IrTxD and RxD1/IrRxD pins function as IrTxD and IrRxD pins, respectively
6	IrCKS2	0	R/W	IrDA Clock Select 2 to 0 Specifies the high-level width of the clock pulse during IrTxD output pulse encoding when the IrDA function is enabled. 000: B x 3/16 (three sixteenths of the bit rate) 001: ϕ/2 010: ϕ/4 011: ϕ/8 100: ϕ/16 101: ϕ/32 110: ϕ/64 111: ϕ/128
5	IrCKS1	0	R/W	
4	IrCKS0	0	R/W	
3	IrTxINV	0	R/W	IrTx Data Invert Specifies the inversion of the logic level of the output from IrTxD. When the inversion is specified, IrCKS2 to IrCKS0 specify the low-level width, not the high-level width. 0: Transmit data is output from IrTxD as it is 1: Transmit data is inverted before being output from IrTxD

Bit	Bit Name	Initial Value	R/W	Description
2	IrRxINV	0	R/W	IrRx Data Invert Specifies the inversion of the logic level of the input to IrRxD. When the inversion is specified, IrCKS2 to IrCKS0 specify the low-level width, not the high-level width. 0: Input to IrRxD is used as receive data as it is 1: Input to IrRxD is inverted before being used as receive data
1, 0	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.

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.

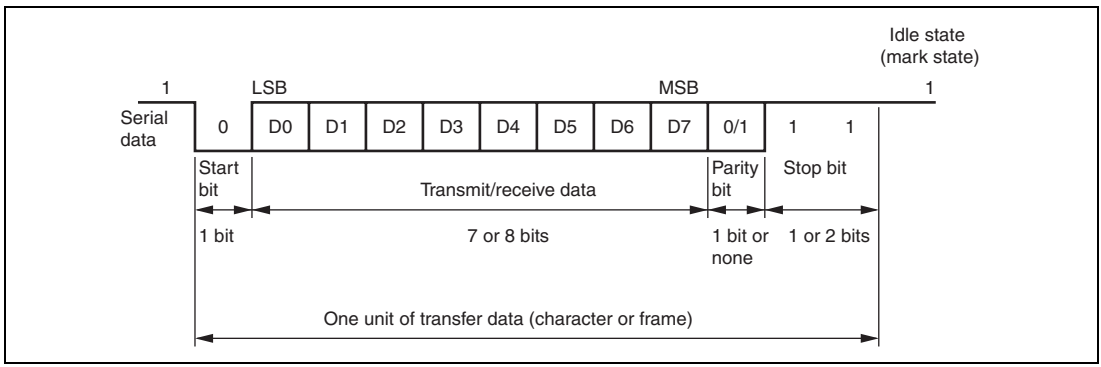


Figure 15.2 Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, Two Stop Bits)

15.4.1 Data Transfer Format

Table 15.10 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 15.10 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.

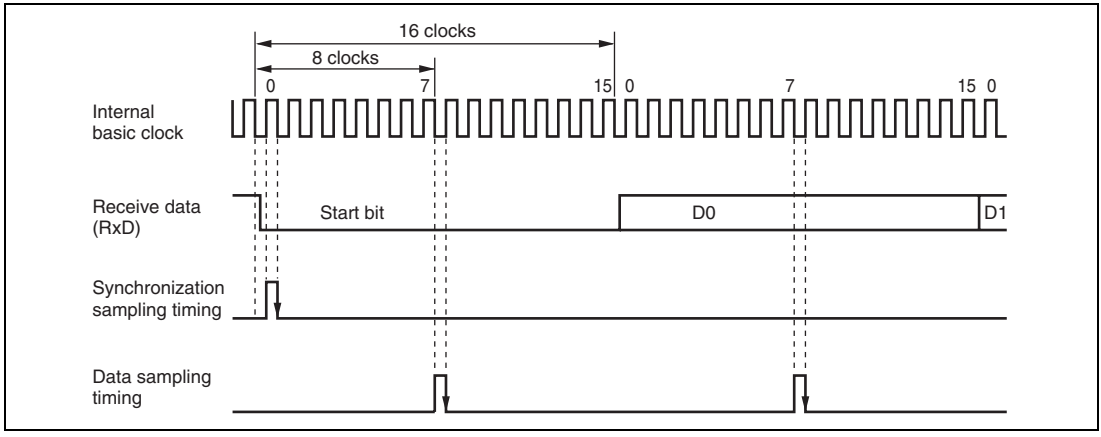


Figure 15.3 Receive Data Sampling Timing in Asynchronous Mode

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.

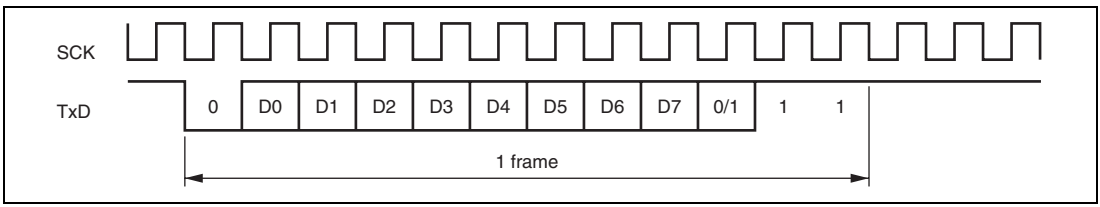
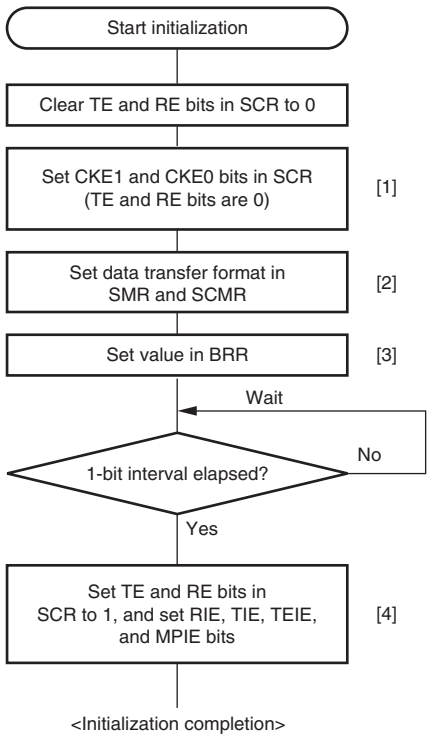


Figure 15.4 Relation between Output Clock and Transmit Data Phase (Asynchronous Mode)

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.

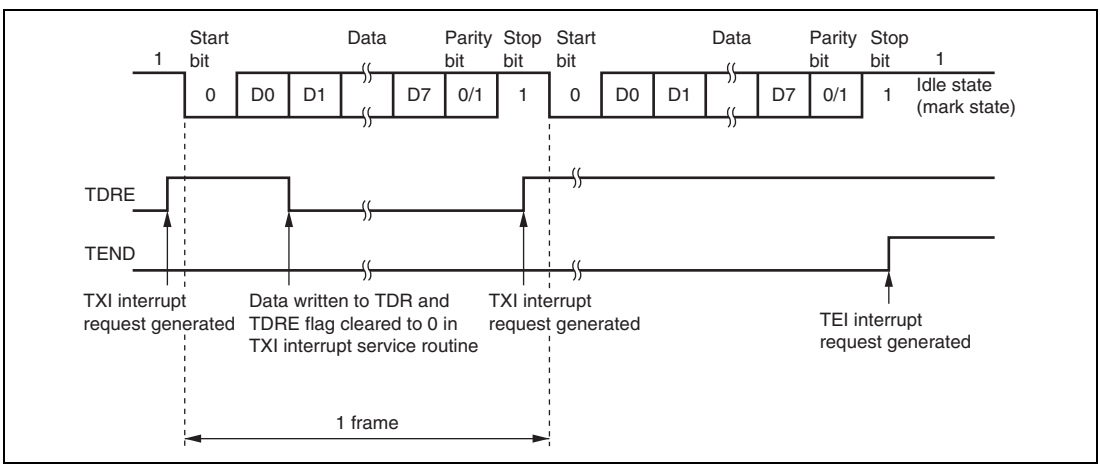
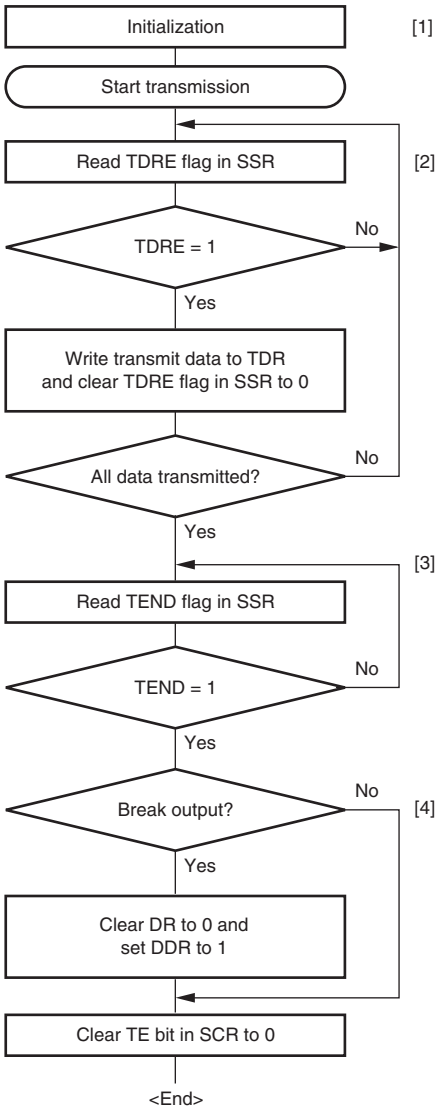


Figure 15.6 Example of Operation in Transmission in Asynchronous Mode (Example with 8-Bit Data, Parity, One Stop Bit)



[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. However, the TDRE flag is checked and cleared automatically when the DTC is initiated by a transmit data empty interrupt (TXI) request and writes data to TDR.

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

Figure 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.

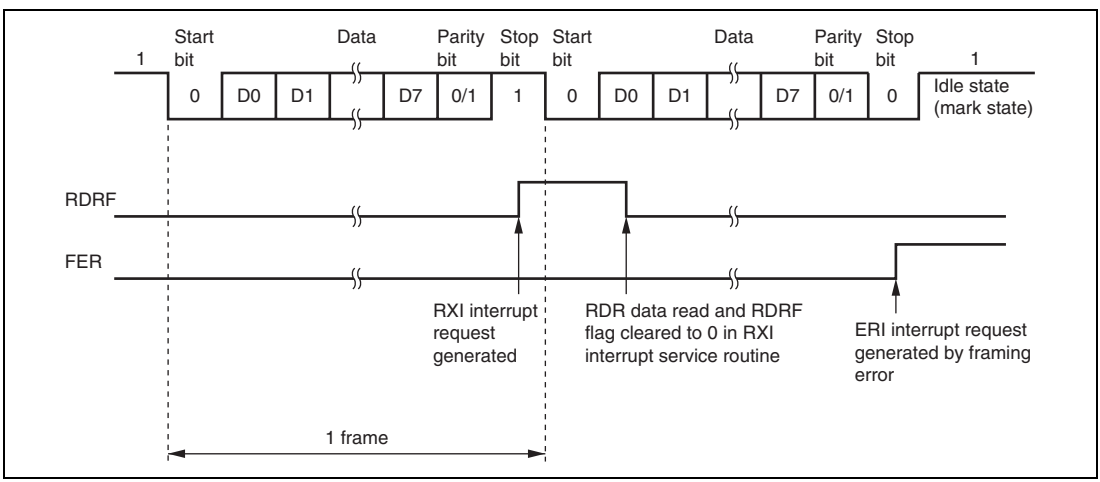


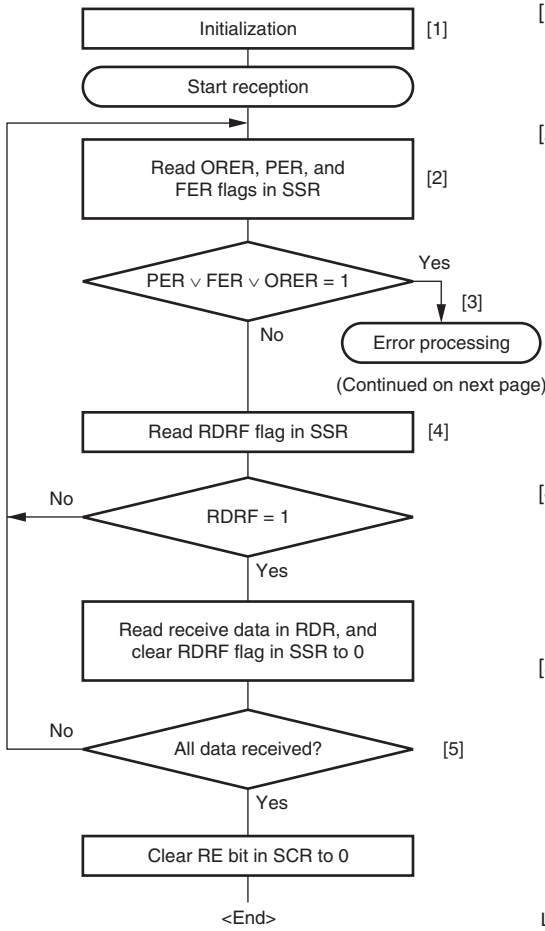
Figure 15.8 Example of SCI Operation in Reception (Example with 8-Bit Data, Parity, One Stop Bit)

Table 15.11 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 15.11 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. However, the RDRF flag is cleared automatically when the DTC is initiated by an RXI interrupt and reads data from RDR.

Legend
∨ : Logical add (OR)

Figure 15.9 Sample Serial Reception Flowchart (1)

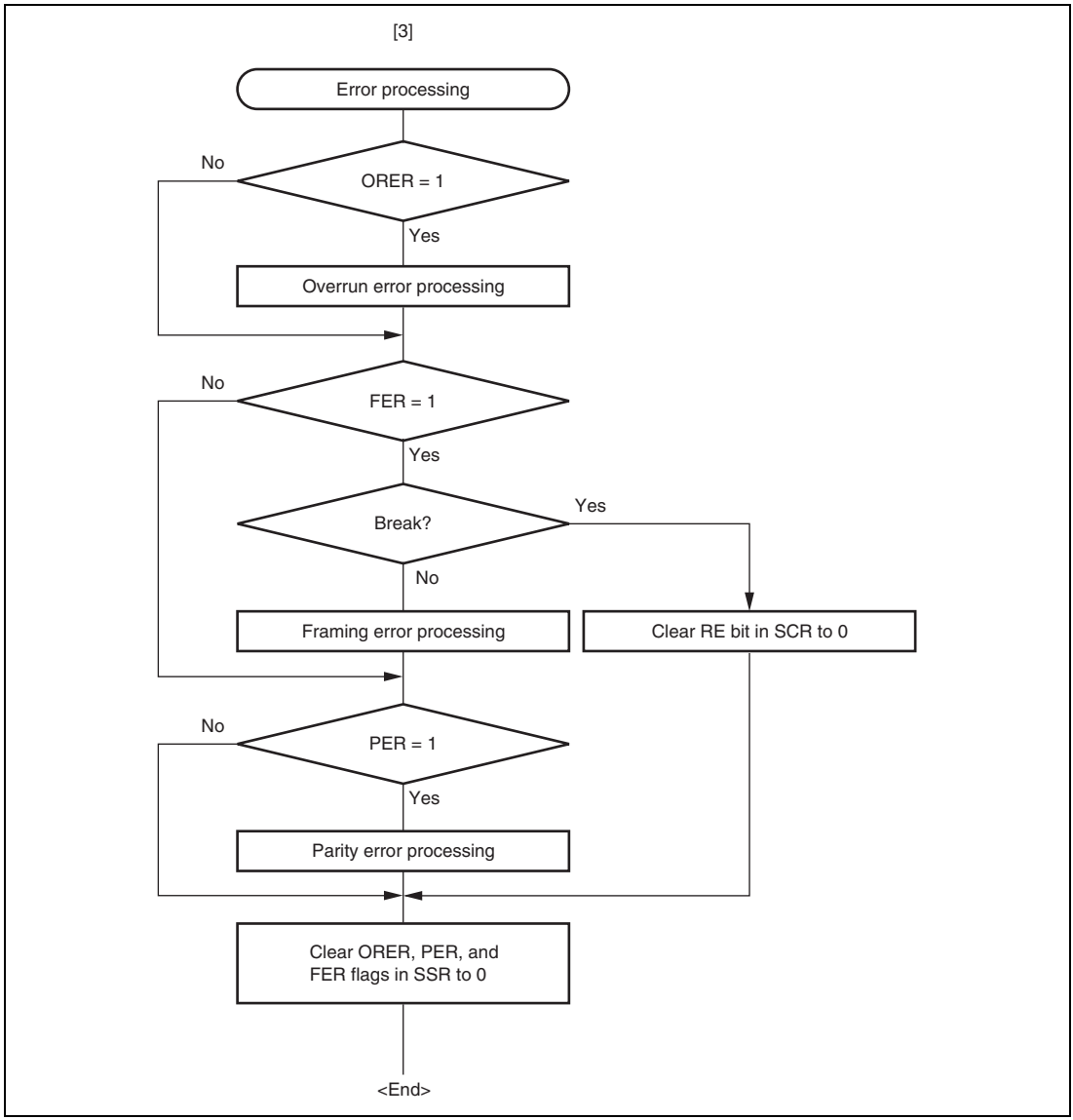


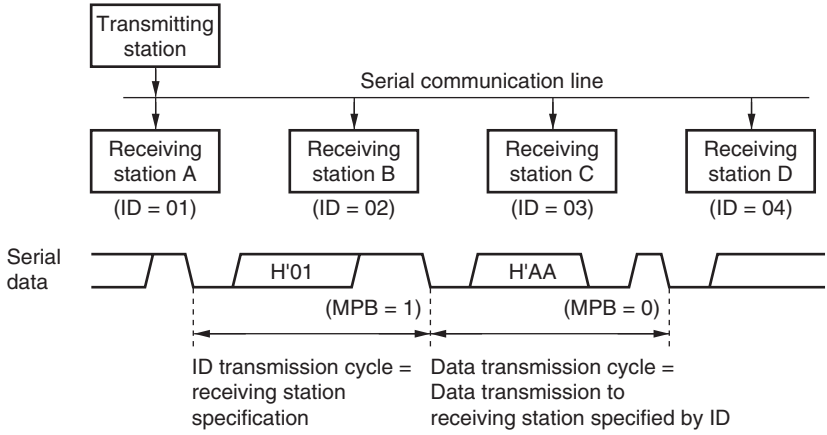
Figure 15.9 Sample Serial Reception Flowchart (2)

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.

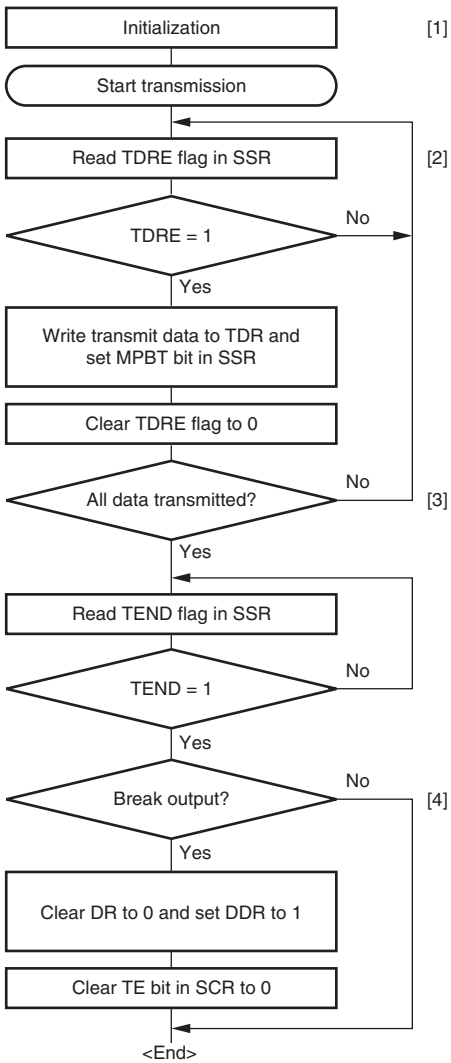


Legend
MPB: Multiprocessor bit

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. However, the TDRE flag is checked and cleared automatically when the DTC is initiated by a transmit data empty interrupt (TXI) request and writes data to TDR.

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

Figure 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.

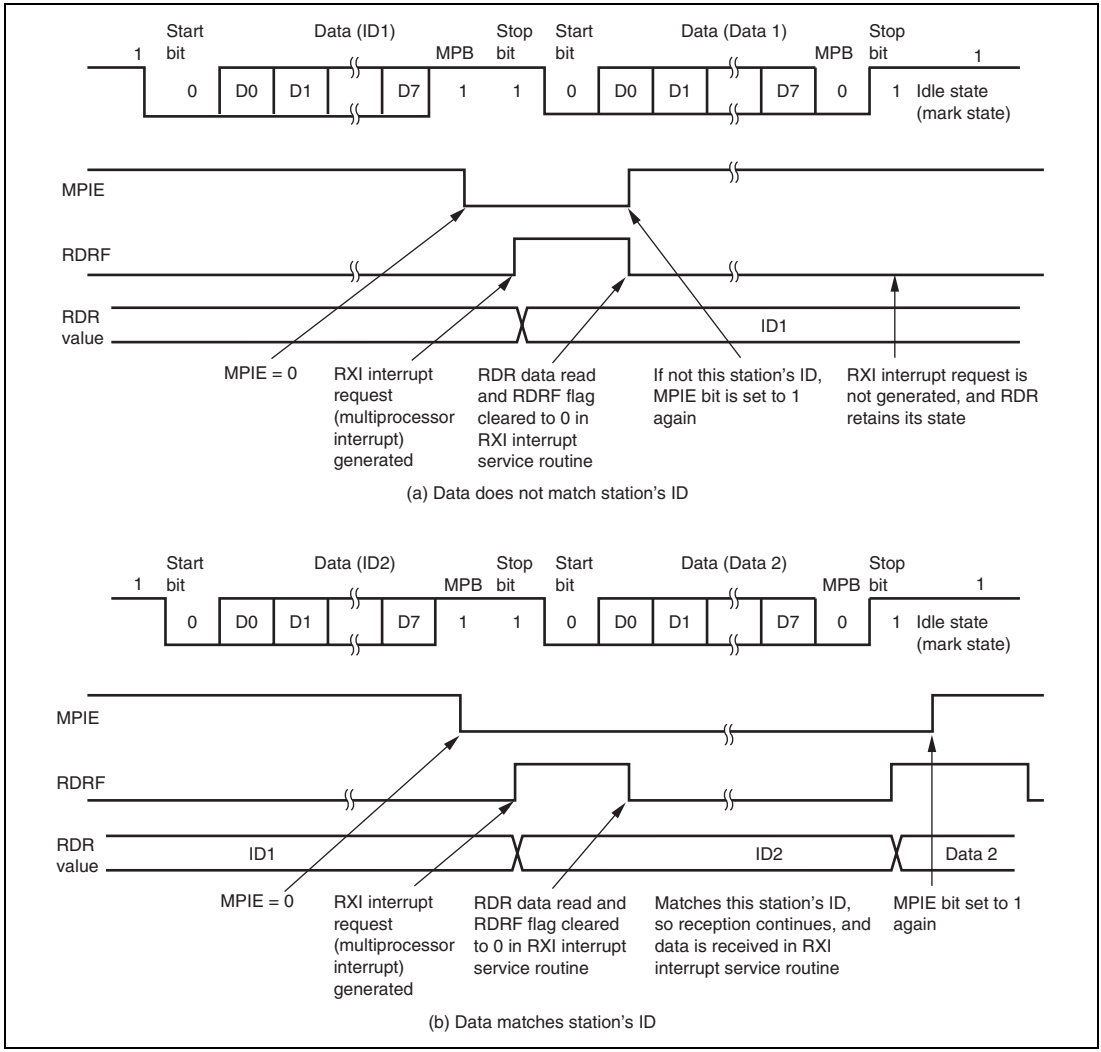


Figure 15.12 Example of SCI Operation in Reception (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)

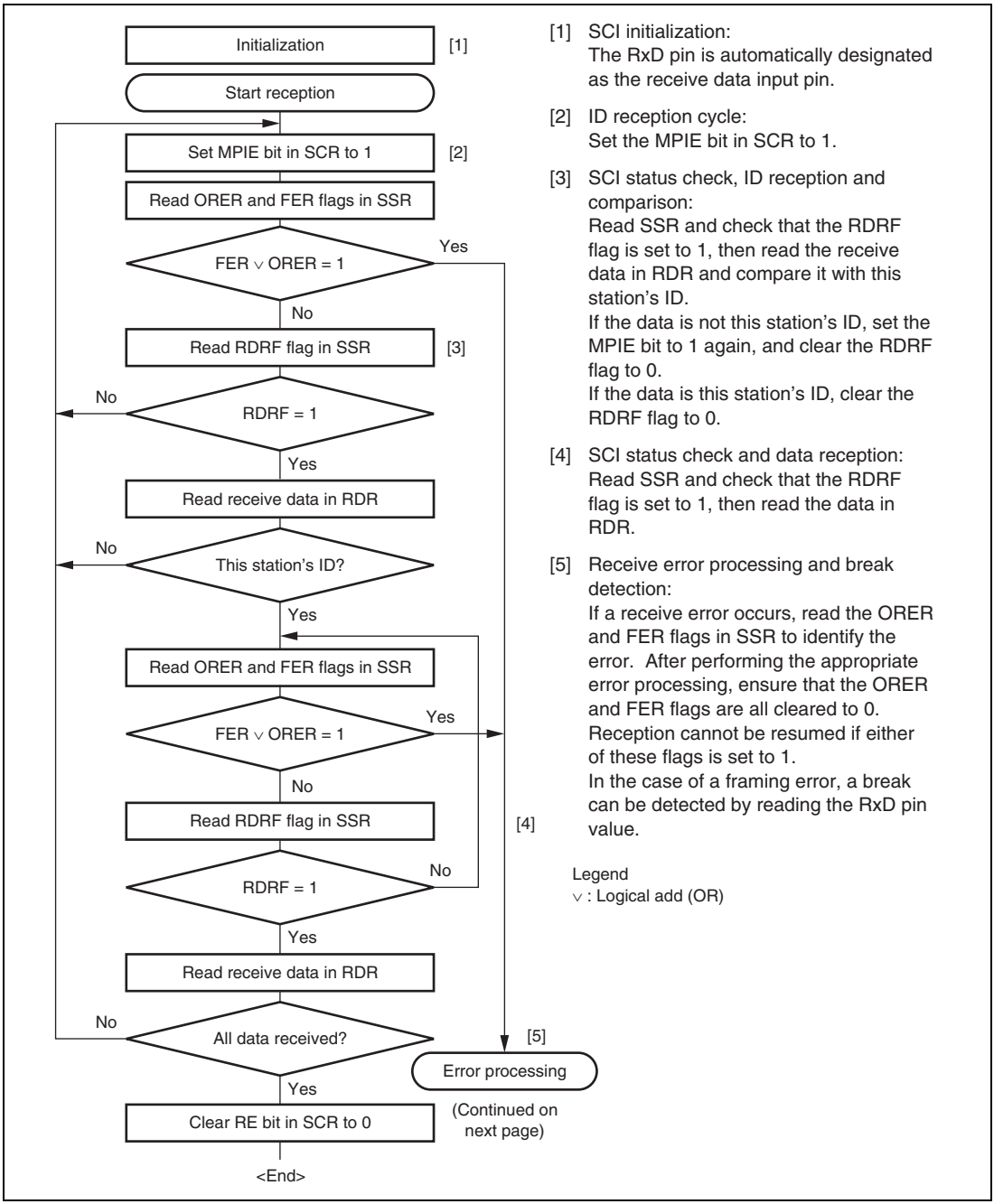


Figure 15.13 Sample Multiprocessor Serial Reception Flowchart (1)

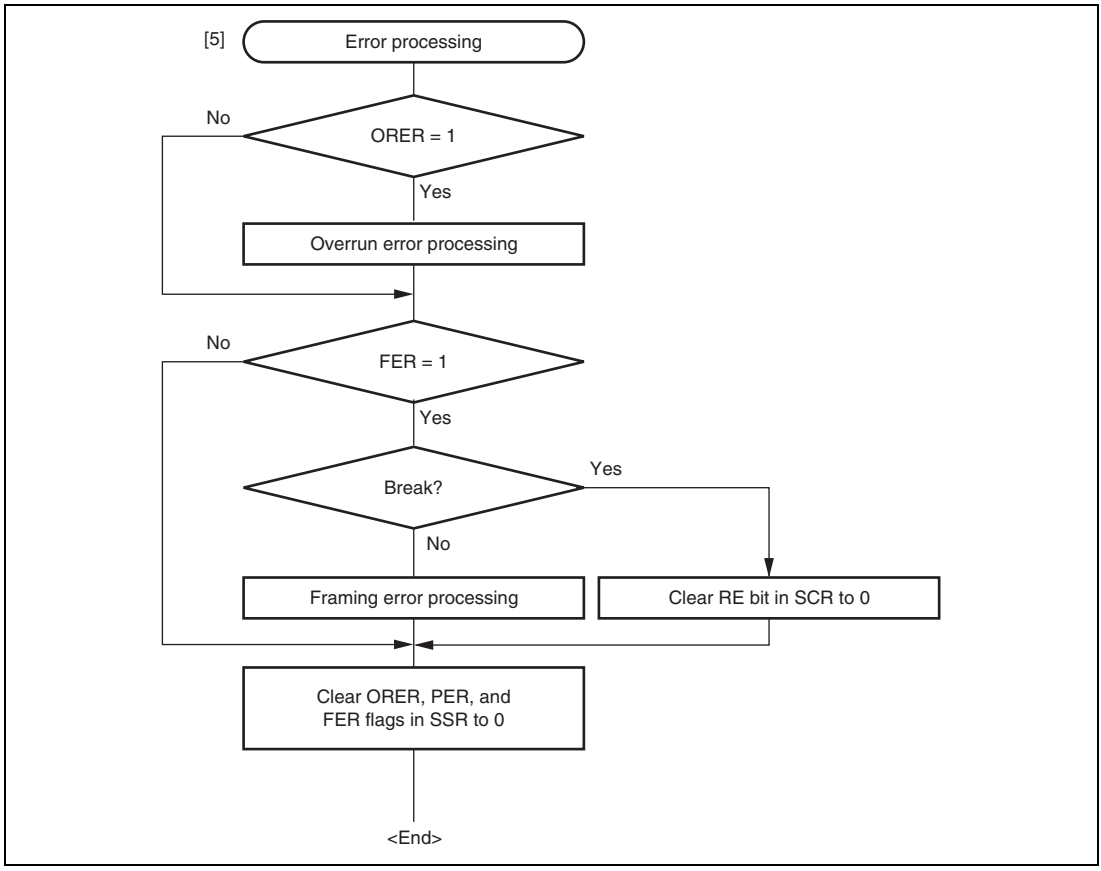


Figure 15.13 Sample Multiprocessor Serial Reception Flowchart (2)

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.

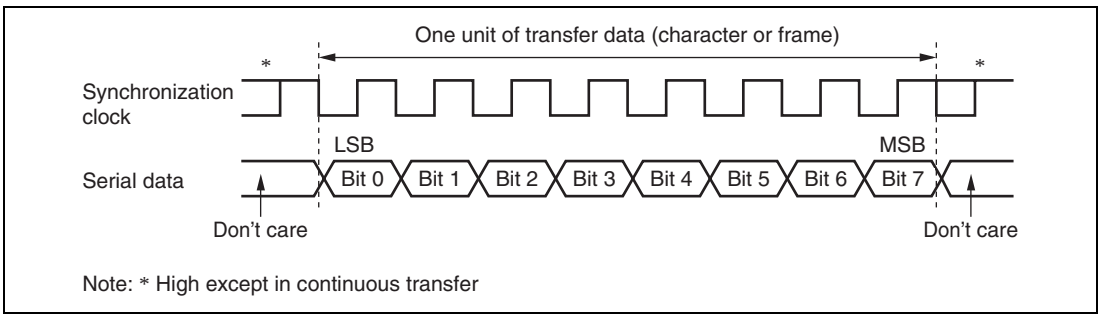


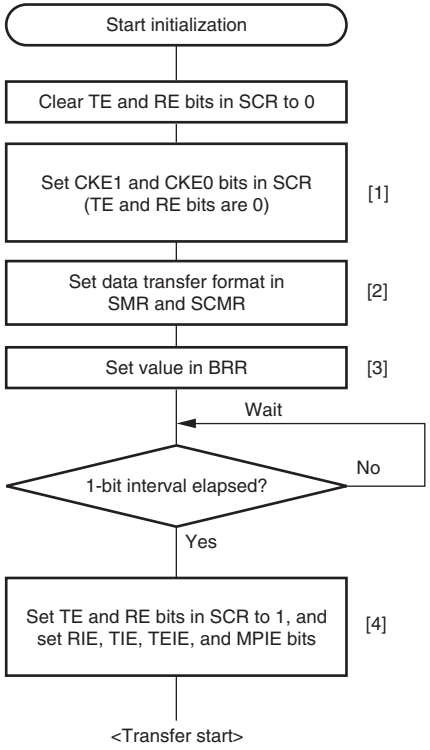
Figure 15.14 Data Format in Synchronous Communication (LSB-First)

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 clock selection in SCR. Be sure to clear bits RIE, TIE, TEIE, MPIE, TE, and RE to 0.

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

[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.

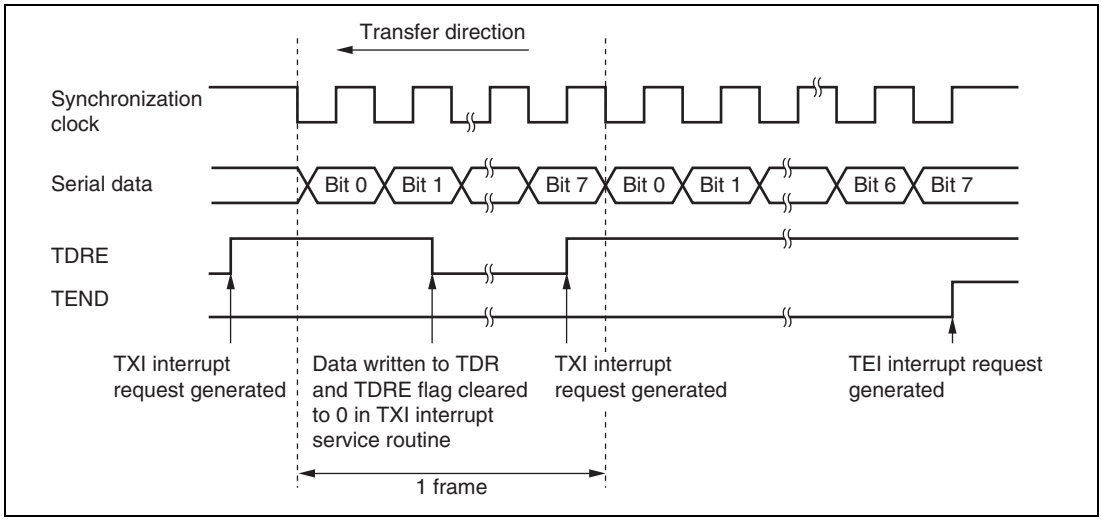
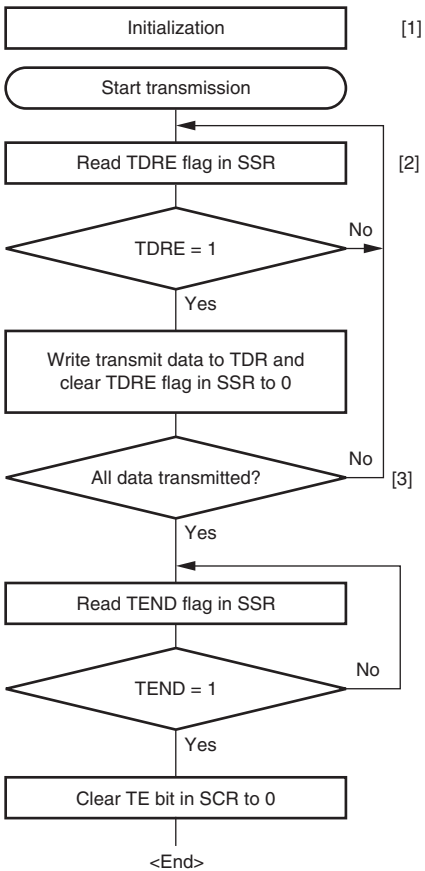


Figure 15.16 Sample SCI Transmission Operation in Clocked Synchronous Mode



[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.
However, the TDRE flag is checked and cleared automatically when the DTC is initiated by a transmit data empty interrupt (TXI) request and writes data to TDR.

Figure 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.

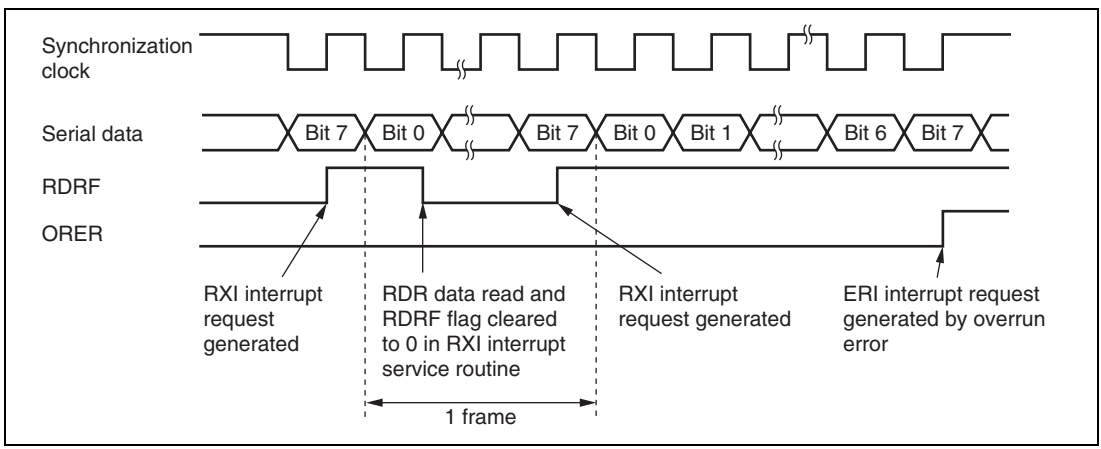
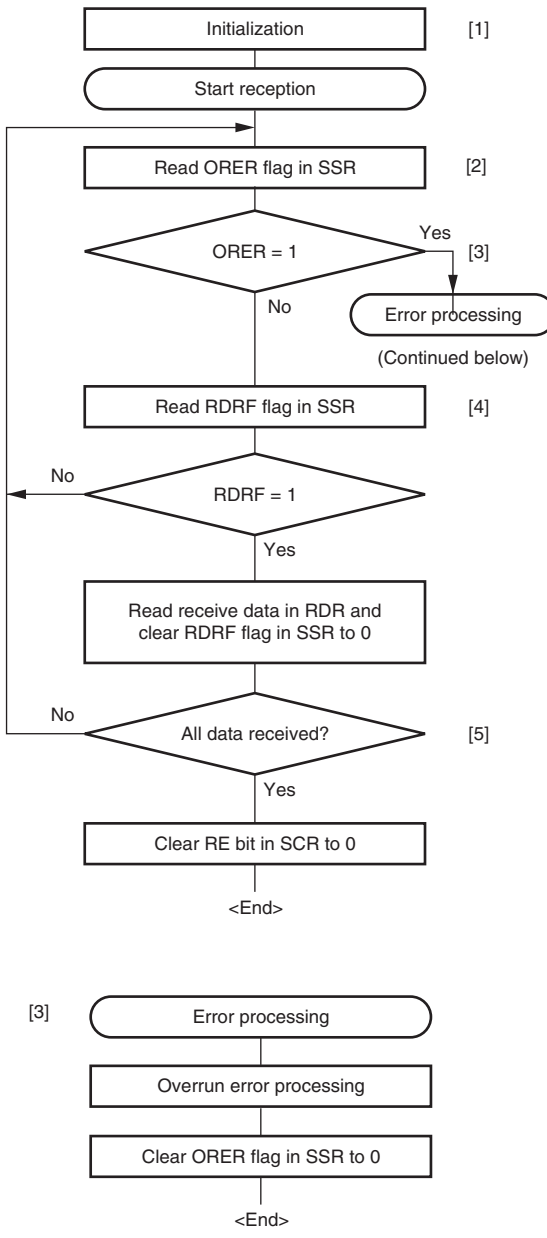


Figure 15.18 Example of SCI Receive Operation in Clocked Synchronous Mode

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.

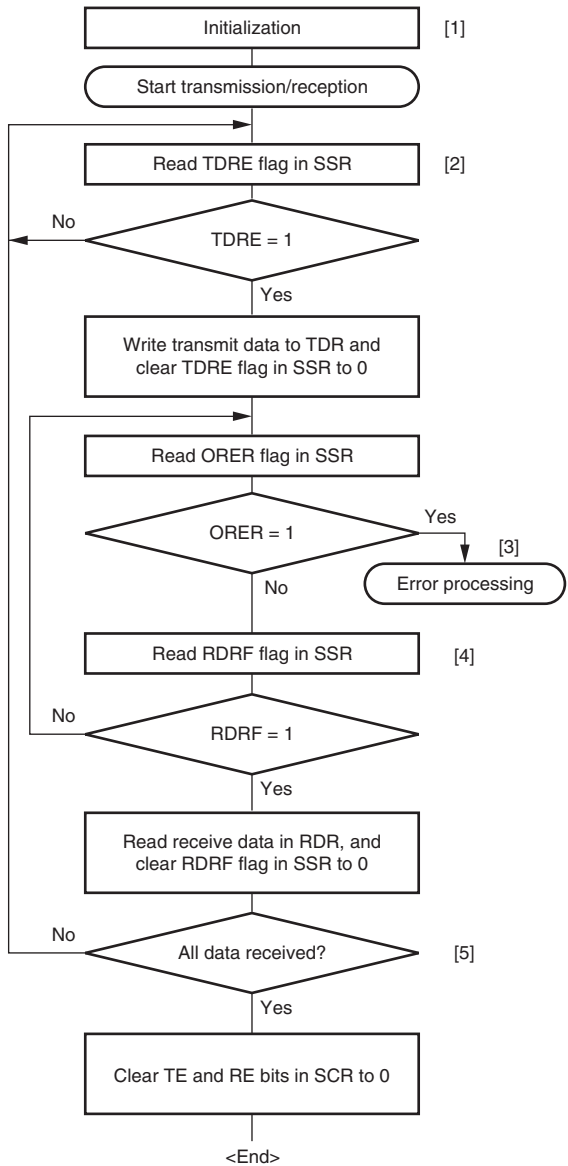
[4] SCI status check and receive data read:
Read SSR and check that the RDRF flag is set to 1, then read the receive data in RDR and clear the RDRF flag to 0.
Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[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.
However, the RDRF flag is cleared automatically when the DTC is initiated by a receive data full interrupt (RXI) and reads data from RDR.

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.



Note: When switching from transmit or receive operation to simultaneous transmit and receive operations, first clear the TE bit and RE bit to 0, then set both these bits to 1 simultaneously.

[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.
However, the TDRE flag is checked and cleared automatically when the DTC is initiated by a transmit data empty interrupt (TXI) request and writes data to TDR. Similarly, the RDRF flag is cleared automatically when the DTC is initiated by a receive data full interrupt (RXI) and reads data from RDR.

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.

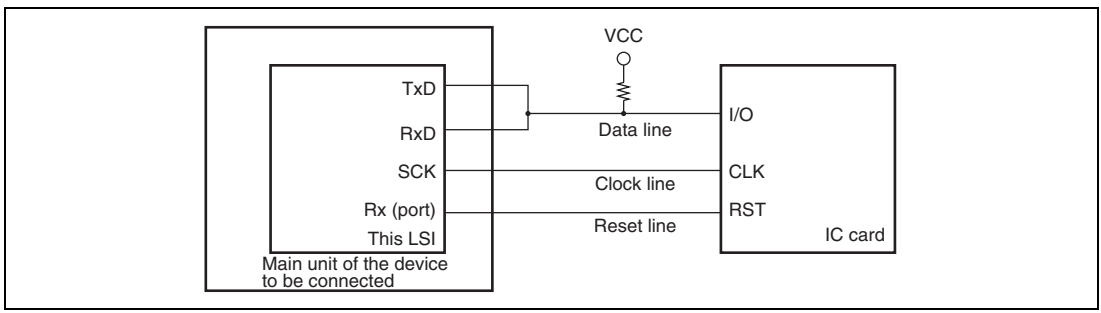


Figure 15.21 Pin Connection for Smart Card Interface

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.

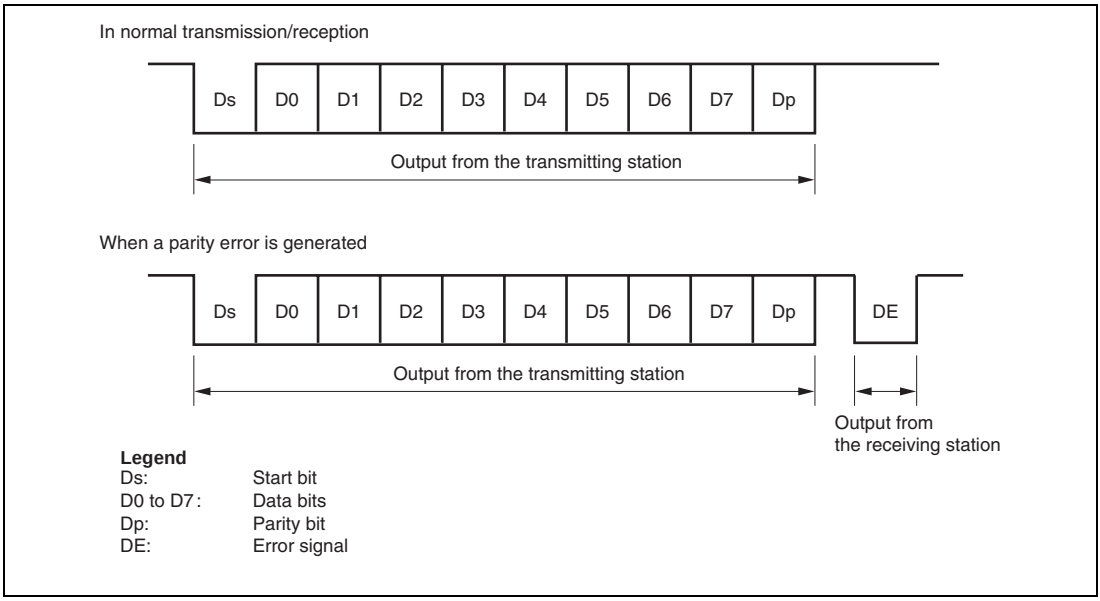


Figure 15.22 Data Formats in Normal Smart Card Interface Mode

For communication with the IC cards of the direct convention and inverse convention types, follow the procedure below.

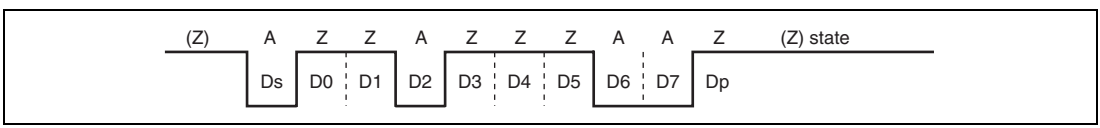


Figure 15.23 Direct Convention (SDIR = SINV = O/$\overline{E}$ = 0)

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.

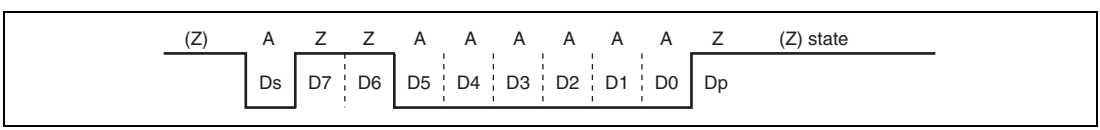


Figure 15.24 Inverse Convention (SDIR = SINV = O/$\overline{E}$ = 1)

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.

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

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.

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

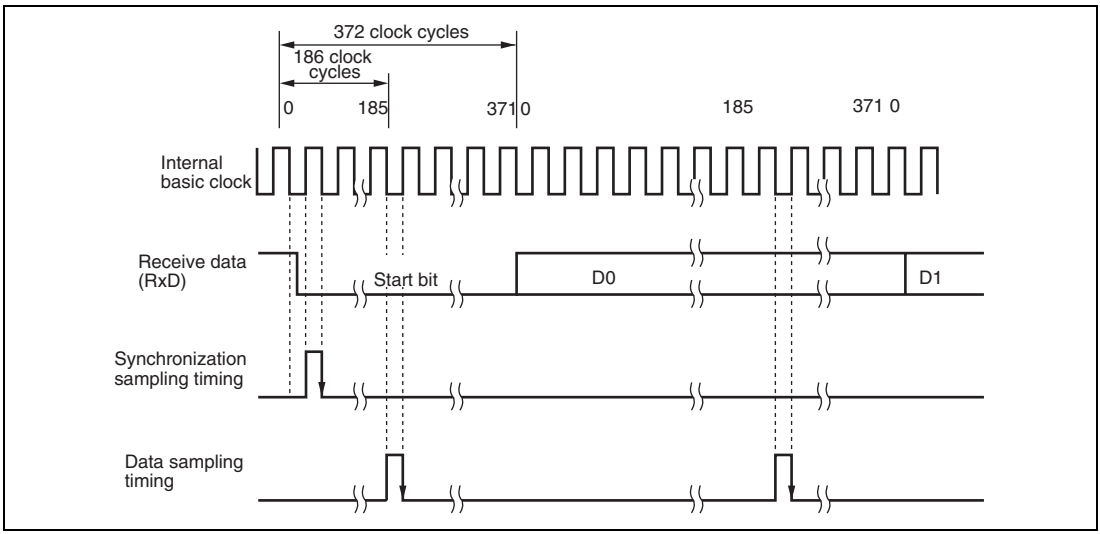


Figure 15.25 Receive Data Sampling Timing in Smart Card Interface Mode (When Clock Frequency is 372 Times the Bit Rate)

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. All the processing steps are automatically performed using a TXI interrupt request to activate the DTC. 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. This activates the DTC by a TXI request thus allowing transfer of transmit data if the TXI interrupt request is specified as a source of DTC activation beforehand. The TDRE and TEND flags are automatically cleared to 0 at data transfer by the DTC. If an error occurs, the SCI automatically re-transmits the same data. During re-transmission, TEND remains as 0, thus not activating the DTC. Therefore, the SCI and DTC automatically transmit the specified number of bytes, including re-transmission in the case of error occurrence. 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.

When transmitting/receiving data using the DTC, be sure to set and enable it prior to making SCI settings. For DTC settings, see section 7, Data Transfer Controller (DTC).

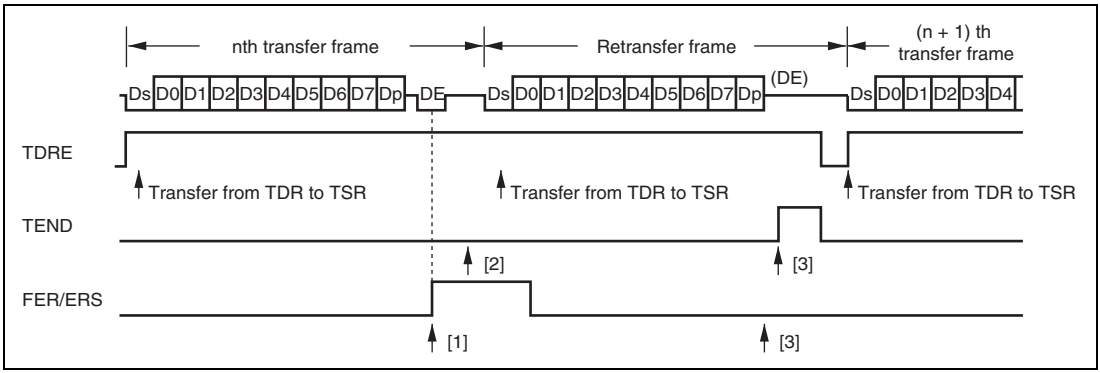


Figure 15.26 Data Re-transfer Operation in SCI Transmission Mode

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.

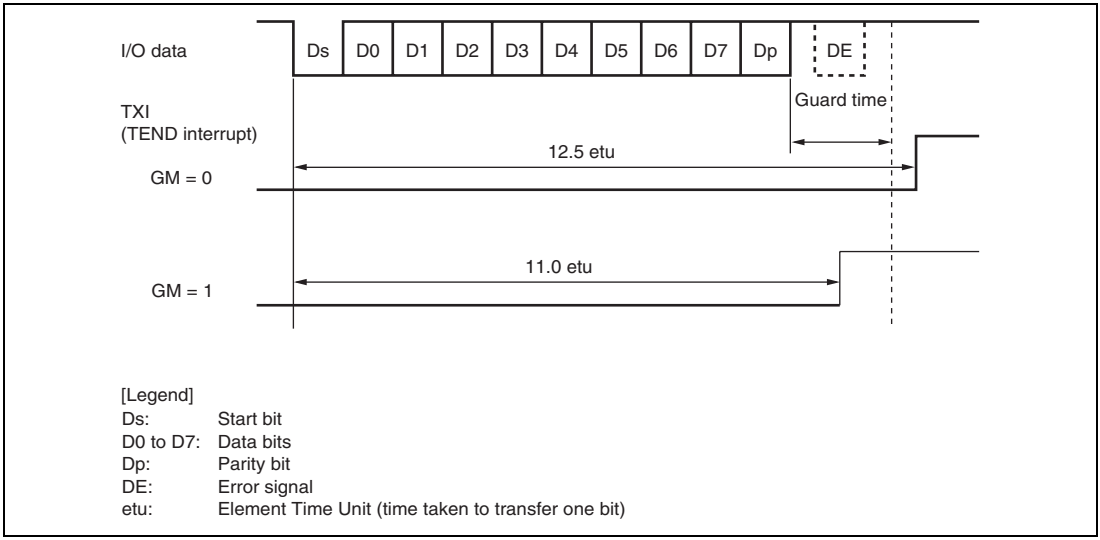


Figure 15.27 TEND Flag Set Timings during Transmission

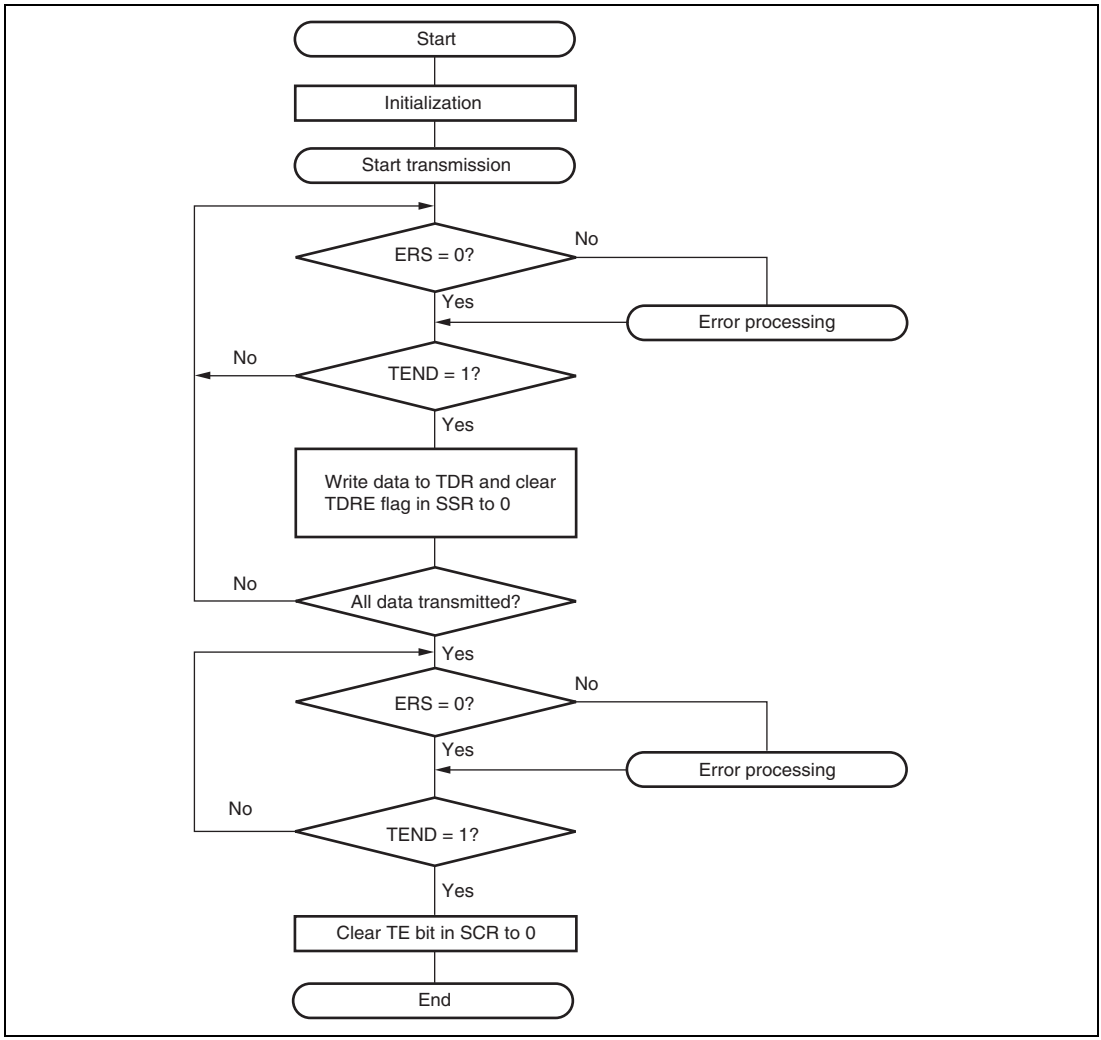


Figure 15.28 Sample Transmission Flowchart

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. All the processing steps are automatically performed using an RXI interrupt request to activate the DTC. In reception, setting the RIE bit to 1 allows an RXI interrupt request to be generated when the RDRF flag is set to 1. This activates DTC by an RXI request thus allowing transfer of receive data if the RXI interrupt request is specified as a source of DTC activate beforehand. The RDRF flag is automatically cleared to 0 at data transfer by DTC. 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. If an error occurs, DTC is not activated and receive data is skipped, therefore, the number of bytes of receive data specified in DTC are transferred. 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.

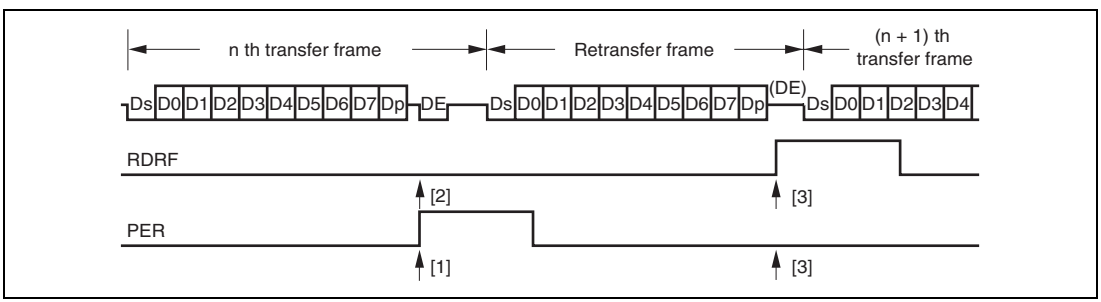


Figure 15.29 Data Re-transfer Operation in SCI Reception Mode

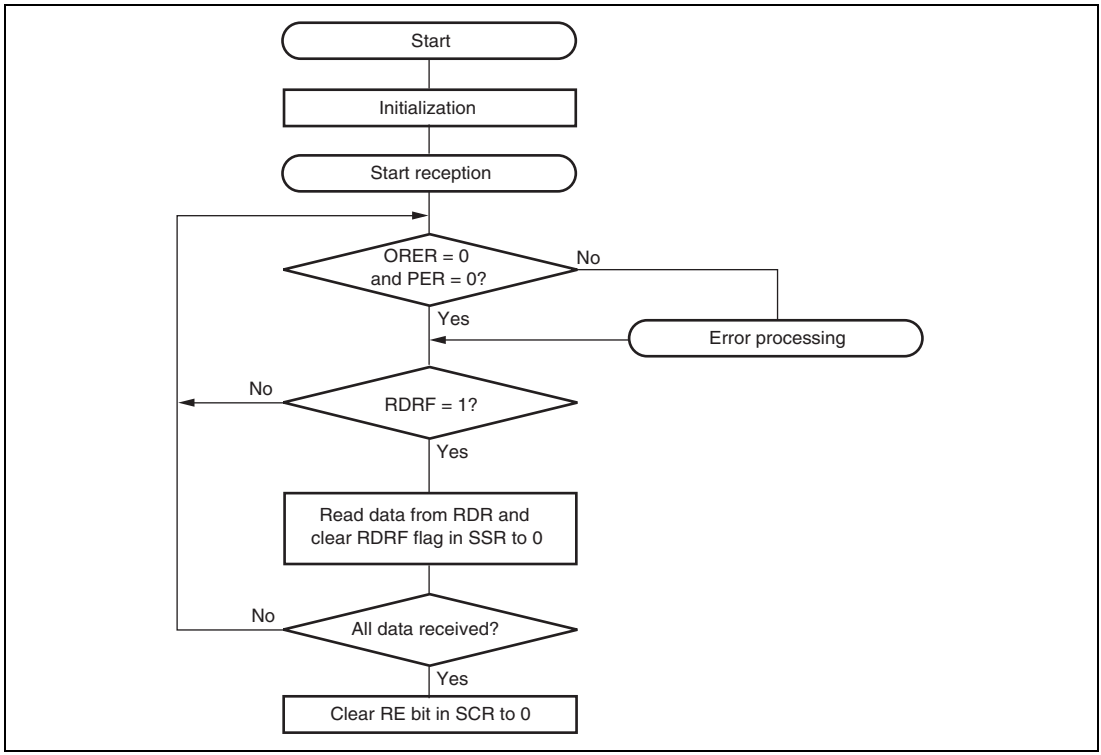


Figure 15.30 Sample Reception Flowchart

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.

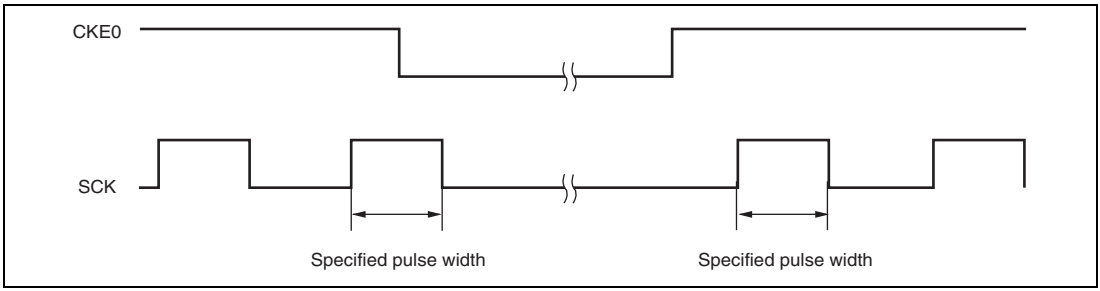


Figure 15.31 Clock Output Fixing Timing

At power-on and transitions to/from software standby mode, use the following procedure to secure the appropriate clock duty ratio.

- At Power-On:

  To secure the appropriate clock duty ratio simultaneously with power-on, use the following procedure.

  A. Initially, port input is enabled in the high-impedance state. To fix the potential level, use a pull-up or pull-down resistor.
  B. Fix the SCK pin to the specified output using the CKE1 bit in SCR.
  C. Set SMR and SCMR to enable smart card interface mode.
  D. Set the CKE0 bit in SCR to 1 to start clock output.

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

  A. 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.
  B. 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.
  C. Write 0 to the CKE0 bit in SCR to stop the clock.
  D. 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.

E. Make the transition to software standby mode.

- At Transition from Software Standby Mode to Smart Card Interface Mode:
  1. Cancel software standby mode.
  2. Write 1 to the CKE0 bit in SCR to start clock output. A clock signal with the appropriate duty ratio is then generated.

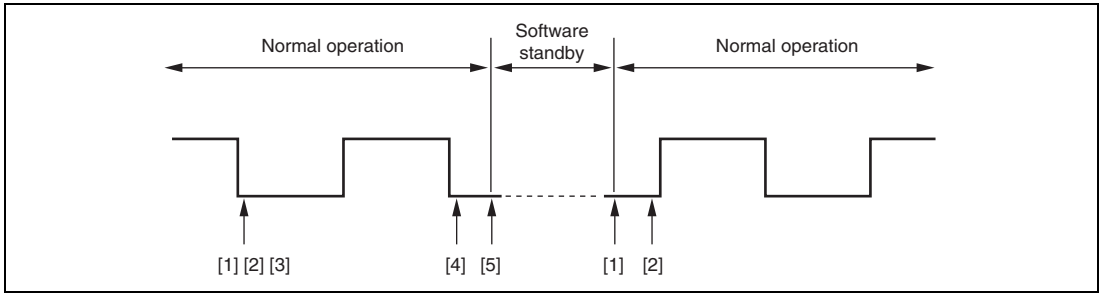


Figure 15.32 Clock Stop and Restart Procedure

15.8 IrDA Operation

IrDA operation can be used with SCI_1. Figure 15.33 shows an IrDA block diagram.

If the IrDA function is enabled using the IrE bit in SCICR, the TxD1 and RxD1 signals for SCI_1 are allowed to encode and decode the waveform based on the IrDA standard version 1.0 (function as the IrTxD and IrRxD pins). Connecting these pins to the infrared data transceiver achieves infrared data communication based on the system defined by the IrDA standard version 1.0.

In the system defined by the IrDA standard version 1.0, communication is started at a transfer rate of 9600 bps, which can be modified as required. The IrDA interface provided by this LSI does not incorporate the capability of automatic modification of the transfer rate; the transfer rate must be modified through programming.

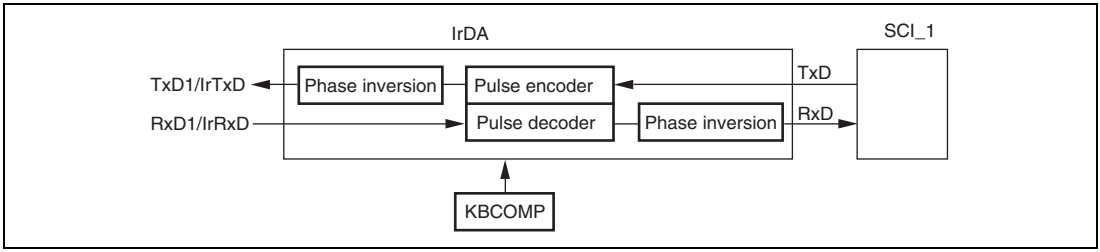


Figure 15.33 IrDA Block Diagram

(1) Transmission

During transmission, the output signals from the SCI (UART frames) are converted to IR frames using the IrDA interface (see figure 15.34).

For serial data of level 0, a high-level pulse having a width of 3/16 of the bit rate (1-bit interval) is output (initial setting). The high-level pulse can be selected using the IrCKS2 to IrCKS0 bits in KBCOMP. The output waveform can also be inverted using the IrTxINV bit in KBCOMP.

The high-level pulse width is defined to be 1.41 µs at the minimum and (3/16 + 2.5%) × bit rate or (3/16 × bit rate) +1.08 µs at the maximum. For example, when the frequency of system clock ϕ is 20 MHz, a high-level pulse width of at least 1.41 µs to 1.6 µs can be specified.

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

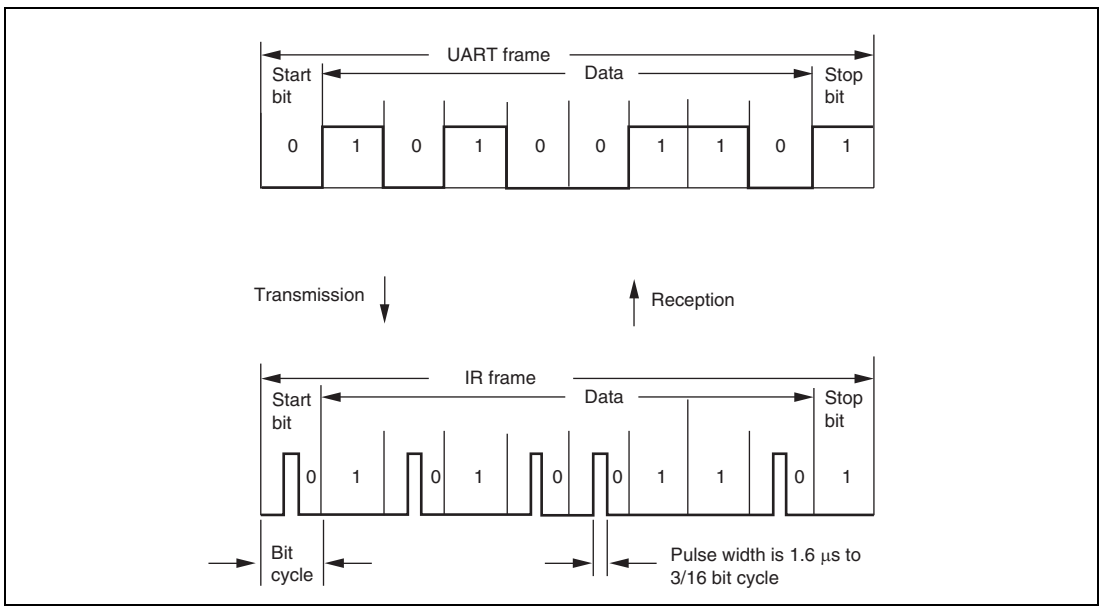


Figure 15.34 IrDA Transmission and Reception

(2) Reception

During reception, IR frames are converted to UART frames using the IrDA interface before inputting to SCI_1. Here, the input waveform can also be inverted using the IrRxINV bit in SCICR.

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

(3) High-Level Pulse Width Selection

Table 15.12 shows possible settings for bits IrCKS2 to IrCKS0 (minimum pulse width), and this LSI's operating frequencies and bit rates, for making the pulse width shorter than 3/16 times the bit rate in transmission.

Table 15.12 IrCKS2 to IrCKS0 Bit Settings

Operating Frequency	Bit Rate (bps) (Upper Row) / Bit Interval × 3/16 (µs) (Lower Row)					
	2400	9600	19200	38400	57600	115200
ϕ (MHz)	78.13	19.53	9.77	4.88	3.26	1.63
4.9152	011	011	011	011	011	011
5	011	011	011	011	011	011
6	100	100	100	100	100	100
6.144	100	100	100	100	100	100
7.3728	100	100	100	100	100	100
8	100	100	100	100	100	100
9.8304	100	100	100	100	100	100
10	100	100	100	100	100	100
12	101	101	101	101	101	101
12.288	101	101	101	101	101	101
14	101	101	101	101	101	101
14.7456	101	101	101	101	101	101
16	101	101	101	101	101	101
16.9344	101	101	101	101	101	101
17.2032	101	101	101	101	101	101
18	101	101	101	101	101	101
19.6608	101	101	101	101	101	101
20	101	101	101	101	101	101

15.9 Interrupt Sources

15.9.1 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. A TXI interrupt can activate the DTC to allow data transfer. The TDRE flag is automatically cleared to 0 at data transfer by the DTC.

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. An RXI interrupt can activate the DTC to allow data transfer. The RDRF flag is automatically cleared to 0 at data transfer by the DTC.

A TEI interrupt is requested when the TEND flag is set to 1 while the TEIE bit is set to 1. 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	DTC Activation	Priority
1	ERI1	Receive error	ORER, FER, PER	Disable	High
	RXI1	Receive data full	RDRF	Enable	
	TXI1	Transmit data empty	TDRE	Enable	
	TEI1	Transmit end	TEND	Disable	
2	ERI2	Receive error	ORER, FER, PER	Disable	
	RXI2	Receive data full	RDRF	Enable	
	TXI2	Transmit data empty	TDRE	Enable	
	TEI2	Transmit end	TEND	Disable	Low

15.9.2 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	DTC Activation	Priority
1	ERI1	Receive error, error signal detection	ORER, PER, ERS	Disable	High
	RXI1	Receive data full	RDRF	Enable	
	TXI1	Transmit data empty	TEND	Enable	
2	ERI2	Receive error, error signal detection	ORER, PER, ERS	Disable	
	RXI2	Receive data full	RDRF	Enable	
	TXI2	Transmit data empty	TEND	Enable	Low

Data transmission/reception using the DTC is also possible in smart card interface mode, similar to in the normal SCI mode. In transmission, the TEND and TDRE flags in SSR are simultaneously set to 1, thus generating a TXI interrupt request. This activates the DTC by a TXI interrupt request thus allowing transfer of transmit data if the TXI interrupt request is specified as a source of DTC activation beforehand. The TDRE and TEND flags are automatically cleared to 0 at data transfer by the DTC. If an error occurs, the SCI automatically re-transmits the same data. During re-transmission, the TEND flag remains as 0, thus not activating the DTC. Therefore, the SCI and DTC automatically transmit the specified number of bytes, including re-transmission in the case of error occurrence. 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.

When transmitting/receiving data using the DTC, be sure to set and enable the DTC prior to making SCI settings. For DTC settings, see section 7, Data Transfer Controller (DTC).

In reception, an RXI interrupt request is generated when the RDRF flag in SSR is set to 1. This activates the DTC by an RXI interrupt request thus allowing transfer of receive data if the RXI interrupt request is specified as a source of DTC activation beforehand. The RDRF flag is automatically cleared to 0 at data transfer by the DTC. If an error occurs, the RDRF flag is not set but the error flag is set. Therefore, the DTC is not activated and an ERI interrupt request is issued to the CPU instead; the error flag must be cleared.

15.10 Usage Notes

15.10.1 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 24, Power-Down Modes.

15.10.2 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.

15.10.3 Mark State and Break Sending

When the TE bit in SCR is 0, the TxD pin is used as an I/O port whose direction (input or 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.

15.10.4 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.

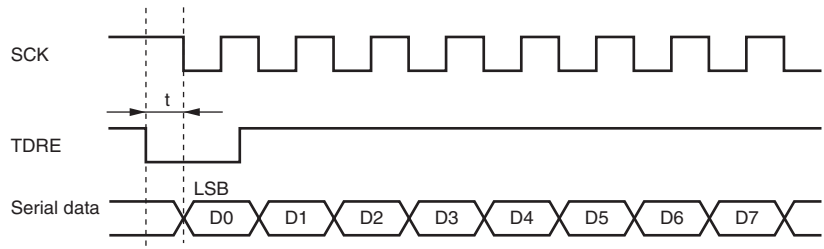
15.10.5 Relation between Writing to TDR and TDRE Flag

Data can be written to TDR irrespective of the TDRE flag status in SSR. However, if the 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.

15.10.6 Restrictions on Using DTC

When the external clock source is used as a synchronization clock, update TDR by the DTC and wait for at least five ϕ clock cycles before allowing the transmit clock to be input. If the transmit clock is input within four clock cycles after TDR modification, the SCI may malfunction (figure 15.35).

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



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

Figure 15.35 Sample Transmission using DTC in Clocked Synchronous Mode

15.10.7 SCI Operations during Mode Transitions

(1) Transmission

Before making the transition to module stop, software standby, or sub-sleep mode, 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 15.36 shows a sample flowchart for mode transition during transmission. Figures 15.37 and 15.38 show the pin states during transmission.

Before making the transition from the transmission mode using DTC transfer to module stop, software standby, or sub-sleep mode, stop all transmit operations (TE = TIE = TEIE = 0). Setting

TE and TIE to 1 after mode cancellation generates a TXI interrupt request to start transmission using the DTC.

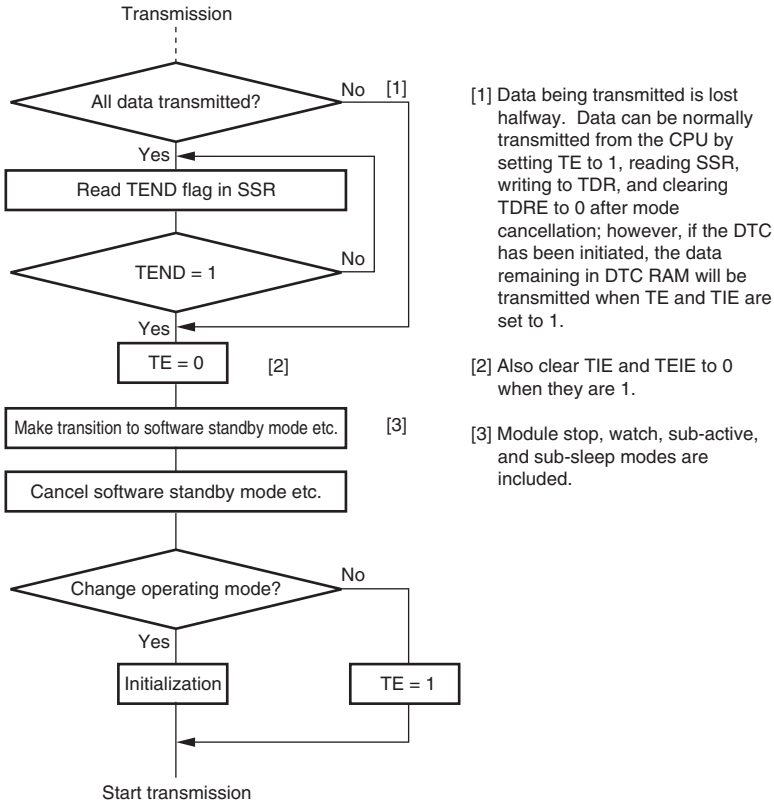


Figure 15.36 Sample Flowchart for Mode Transition during Transmission

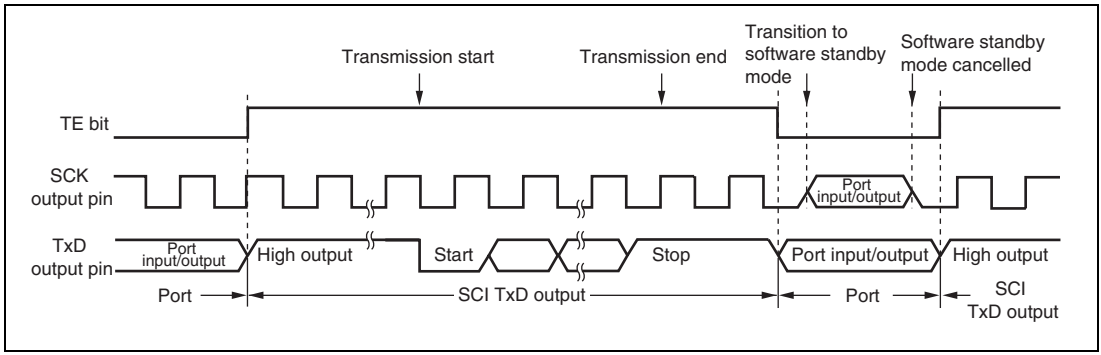


Figure 15.37 Pin States during Transmission in Asynchronous Mode (Internal Clock)

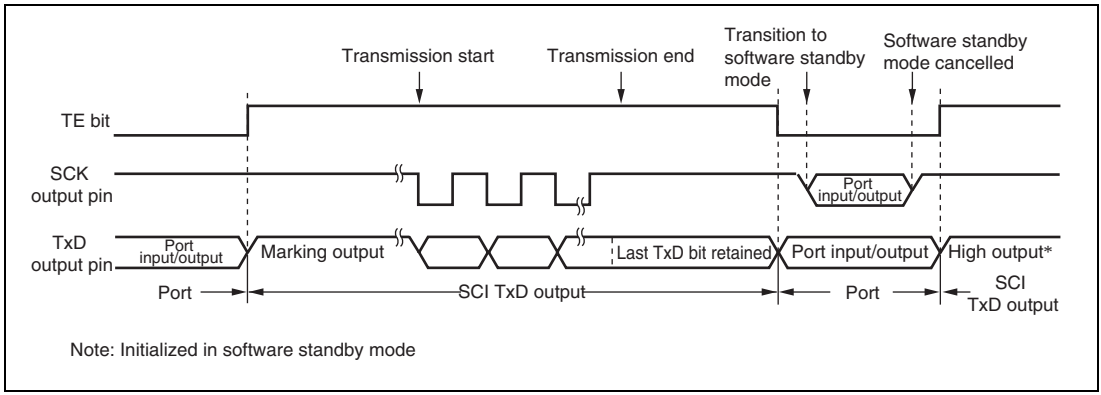


Figure 15.38 Pin States during Transmission in Clocked Synchronous Mode (Internal Clock)

(2) Reception

Before making the transition to module stop, software standby, watch, sub-active, or sub-sleep 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 15.39 shows a sample flowchart for mode transition during reception.

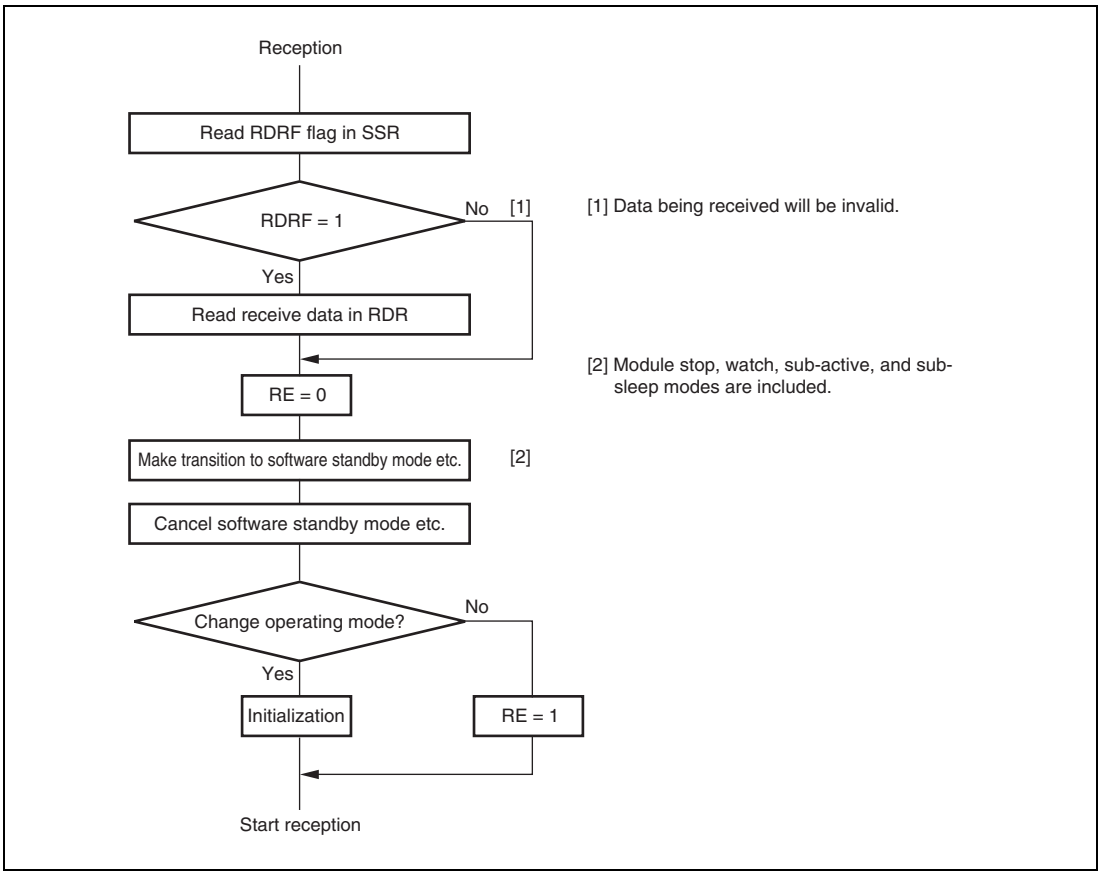


Figure 15.39 Sample Flowchart for Mode Transition during Reception

15.10.8 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.

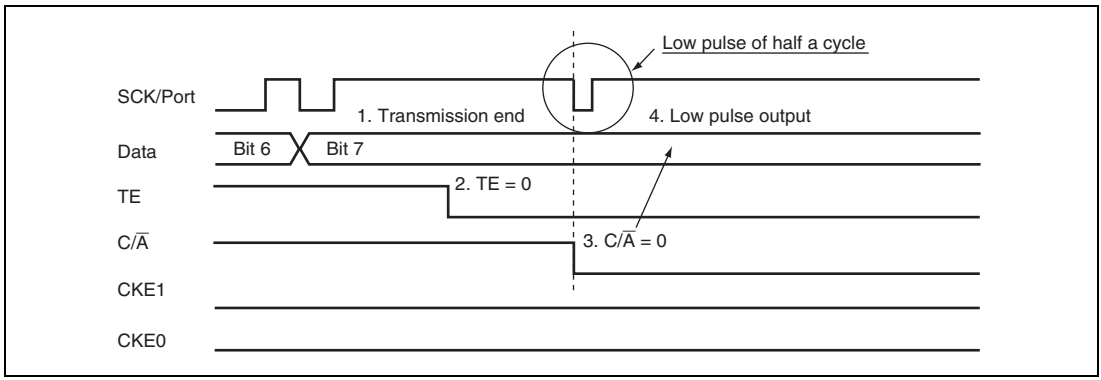


Figure 15.40 Switching from SCK Pins to Port Pins

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

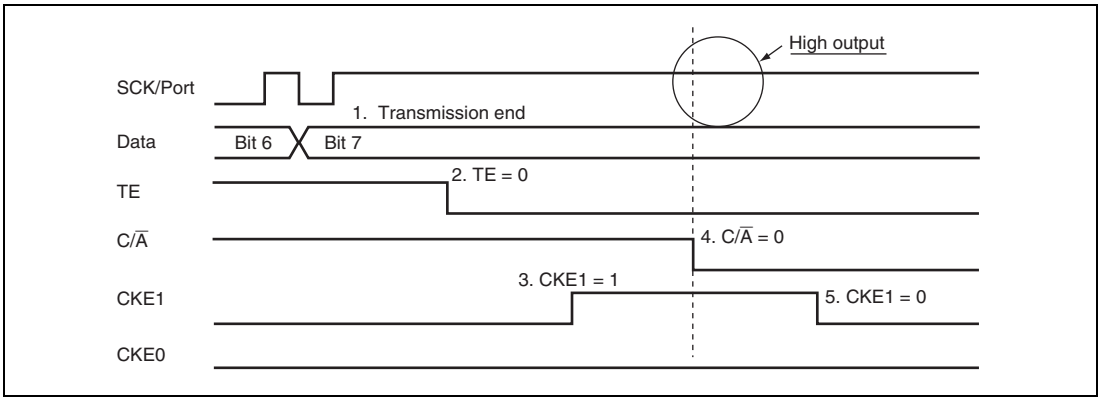


Figure 15.41 Prevention of Low Pulse Output at Switching from SCK Pins to Port Pins

Section 16 I²C Bus Interface (IIC)

This LSI has a two-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.

16.1 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)
  - Eight pins—P52/SCL0, P97/SDA0, P86/SCL1, P42/SDA1, 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.

Figure 16.1 shows a block diagram of the I²C bus interface. Figure 16.2 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 26, Electrical Characteristics.

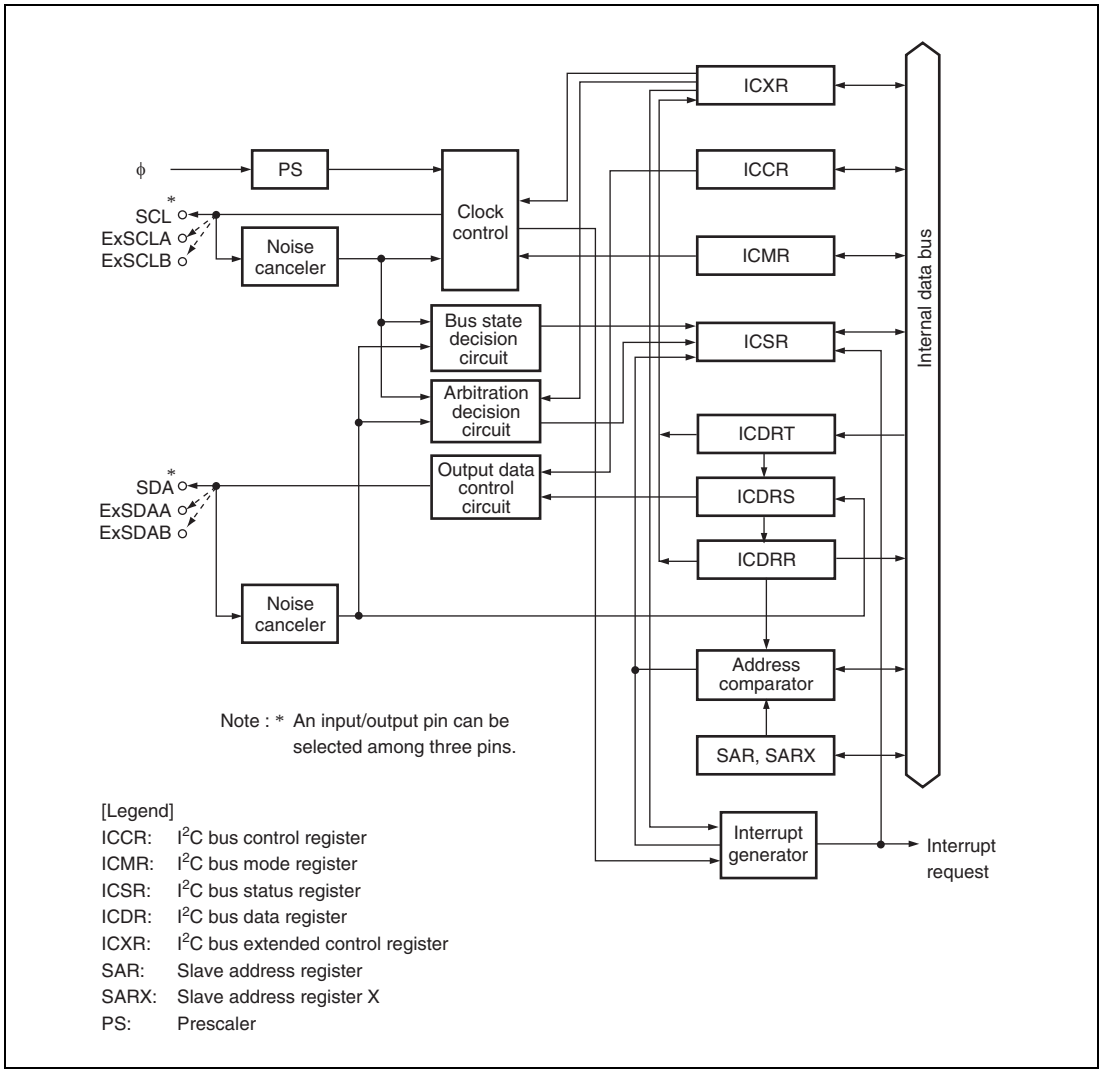


Figure 16.1 Block Diagram of I²C Bus Interface

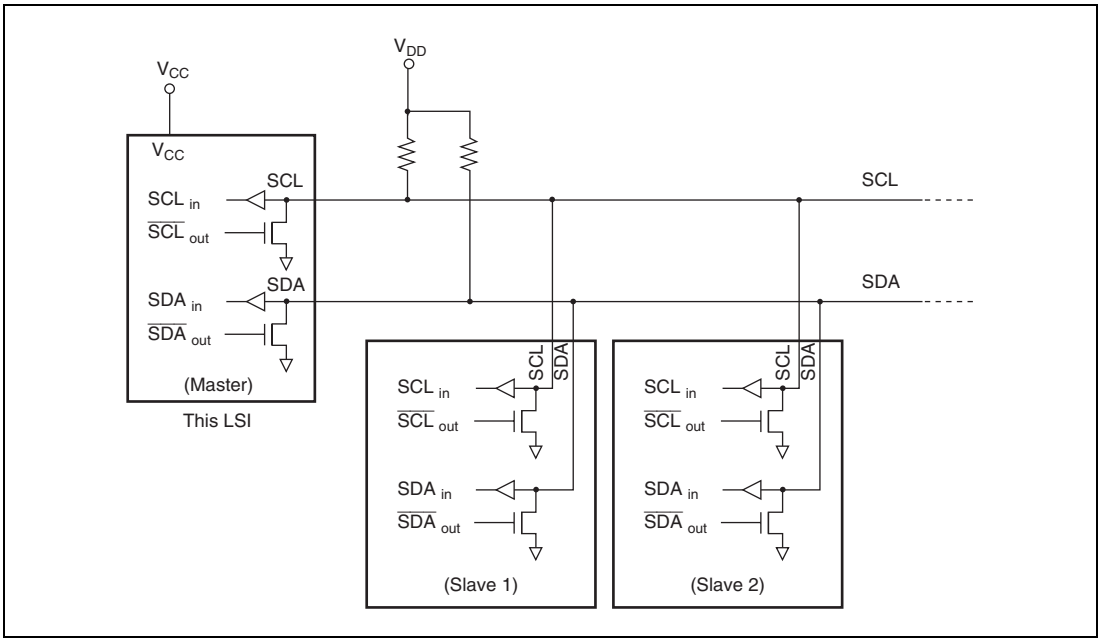


Figure 16.2 I²C Bus Interface Connections (Example: This LSI as Master)

16.2 Input/Output Pins

Table 16.1 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 of each channel. Two or more input/output pins should not be specified for one channel.

For the method of setting pins, see section 8.17.2, Port Control Register 1 (PTCNT1).

Table 16.1 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
—	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.

16.3 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).

- I²C bus control register (ICCR)
- I²C bus status register (ICSR)
- I²C bus data register (ICDR)
- I²C bus mode register (ICMR)
- Slave address register (SAR)
- Second slave address register (SARX)
- I²C bus extended control register (ICXR)
- DDC switch register (DDCSWR)*

Note: DDCSWR is available in IIC_0.

16.3.1 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.

If the IIC is in transmit mode (TRS = 1) and ICDRT has the next data (the ICDRE flag is 0), data is transferred automatically from ICDRT to ICDRS, following transmission of one frame of data using ICDRS. When the ICDRE flag is 1 and the next transmit data writing is waited, data is transferred automatically from ICDRT to ICDRS by writing to ICDR. If I²C is in receive mode (TRS = 0), no data is transferred from ICDRT to ICDRS. Note that data should not be written to ICDR in receive mode.

Reading receive data from ICDR is performed after data is transferred from ICDRS to ICDRR.

If I²C is in receive mode and no previous data remains in ICDRR (the ICDRF flag is 0), data is transferred automatically from ICDRS to ICDRR, following reception of one frame of data using ICDRS. If additional data is received while the ICDRF flag is 1, data is transferred automatically from ICDRS to ICDRR by reading from ICDR. In transmit mode, no data is transferred from ICDRS to ICDRR. Always set I²C to receive mode before reading from ICDR.

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.

16.3.2 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 16.2. This bit should be set to 0 when general call address recognition is performed.

16.3.3 Second Slave Address Register (SARX)

SARX sets the second slave address and selects the communication format. If the LSI is in slave mode, when received address matches the second slave address, transmission/reception using the DTC is enabled. 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 16.2.

Table 16.2 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

16.3.4 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 16.4.7, IRIC Setting Timing and SCL Control.
5	CKS2	0	R/W	Transfer Clock Select 2 to 0
4	CKS1	0	R/W	These bits are used only in master mode.
3	CKS0	0	R/W	These bits select the required transfer rate, together with the IICX1 (IIC_1) and IICX0 (IIC_0) bits in STCR. See table 16.3.

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 16.3 I²C Transfer Rate

STCR	ICMR								
Bits 5 and 6	Bit 5	Bit 4	Bit 3		Transfer Rate				
IICX	CKS2	CKS1	CKS0	Clock	φ = 5 MHz	φ = 8 MHz	φ = 10 MHz	φ = 16 MHz	φ = 20 MHz
0	0	0	0	φ/28	179 kHz	286 kHz	357 kHz	571 kHz*	714 kHz*
0	0	0	1	φ/40	125 kHz	200 kHz	250 kHz	400 kHz	500 kHz*
0	0	1	0	φ/48	104 kHz	167 kHz	208 kHz	333 kHz	417 kHz*
0	0	1	1	φ/64	78.1 kHz	125 kHz	156 kHz	250 kHz	3136 kHz
0	1	0	0	φ/80	62.5 kHz	100 kHz	125 kHz	200 kHz	250 kHz
0	1	0	1	φ/100	50.0 kHz	80.0 kHz	100 kHz	160 kHz	200 kHz
0	1	1	0	φ/112	44.6 kHz	71.4 kHz	89.3 kHz	143 kHz	179 kHz
0	1	1	1	φ/128	39.1 kHz	62.5 kHz	78.1 kHz	125 kHz	156 kHz
1	0	0	0	φ/56	89.3 kHz	143 kHz	179 kHz	286 kHz	357 kHz
1	0	0	1	φ/80	62.5 kHz	100 kHz	125 kHz	200 kHz	250 kHz
1	0	1	0	φ/96	52.1 kHz	83.3 kHz	104 kHz	167 kHz	208 kHz
1	0	1	1	φ/128	39.1 kHz	62.5 kHz	78.1 kHz	125 kHz	156 kHz
1	1	0	0	φ/160	31.3 kHz	50.0 kHz	62.5 kHz	100 kHz	125 kHz
1	1	0	1	φ/200	25.0 kHz	40.0 kHz	50.0 kHz	80.0 kHz	100 kHz
1	1	1	0	φ/224	22.3 kHz	35.7 kHz	44.6 kHz	71.4 kHz	89.3 kHz
1	1	1	1	φ/256	19.5 kHz	31.3 kHz	39.1 kHz	62.5 kHz	78.1 kHz

Note: * 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).

16.3.5 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.

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*[1]	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.

Bit	Bit Name	Initial Value	R/W	Description
1	IRIC	0	R/(W)*2	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 16.4.7, 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] I²C bus format master mode: • When a start condition is detected in the bus line state after a start condition is issued (when the ICDRE flag is set to 1 because of first frame transmission) • When a wait is inserted between the data and acknowledge bit when the WAIT bit is 1 (fall of the 8th transmit/receive clock) • At the end of data transfer (rise of the 9th transmit/receive clock while no wait is inserted) • When a slave address is received after bus arbitration is lost (the first frame after the start condition) • If 1 is received as the acknowledge bit (when the ACKB bit in ICSR is set to 1) when the ACKE bit is 1 • When the AL flag is set to 1 after bus arbitration is lost while the ALIE bit is 1 I²C bus format slave mode: • When the slave address (SVA or SVAX) matches (when the AAS or AASX flag in ICSR is set to 1) and at the end of data transfer up to the subsequent retransmission start condition or stop condition detection (rise of the 9th transmit/receive clock) • When the general call address is detected (when 0 is received as the R/$\overline{W}$ bit and the ADZ flag in ICSR is set to 1) and at the end of data reception up to the subsequent retransmission start condition or stop condition detection (rise of the 9th receive clock) • If 1 is received as the acknowledge bit (when the ACKB bit in ICSR is set to 1) while the ACKE bit is 1 • When a stop condition is detected (when the STOP or ESTP flag in ICSR is set to 1) while the STOPIM bit is 0

Bit	Bit Name	Initial Value	R/W	Description
1	IRIC	0	R/(W) *2	Clocked synchronous serial format mode: • At the end of data transfer (rise of the 8th transmit/receive) • When a start condition is detected When the ICDRE or ICDRF flag is set to 1 in any operating mode: • When a start condition is detected in transmit mode (when a start condition is detected in transmit mode and the ICDRE flag is set to 1) • When data is transferred among the ICDR register and buffer (when data is transferred from ICDRT to ICDRS in transmit mode and the ICDRE flag is set to 1, or when data is transferred from ICDRS to ICDRR in receive mode and the ICDRF flag is set to 1) [Clearing conditions] • When 0 is written in IRIC after reading IRIC = 1 • When ICDR is read/written by the DTC (in some cases, this condition does not work as clearing condition, therefore, for details see following explanation on the operation of DTC)

Notes: 1. The value of the BBSY flag is not changed even though it is written to.
2. Only 0 can be written, to clear the flag.

Using DTC clears the IRIC flag automatically and enables consecutive transfer without CPU.

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 which is a DTC activation source 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.

Even if the IRIC and IRTR flags are set, the ICDRE flag or ICDRF flag may not be set. In the case of continuous transfer by using the DTC, the IRIC and IRTR flags are not cleared after the specified number of transfers is completed. While, as the specified number of reading/writing ICDR has been completed, reading/writing of the ICDRE or ICDRF flag is cleared.

Tables 16.4 and 16.5 show the relationship between the flags and the transfer states.

Table 16.4 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 16.5 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

MST	TRS	BBSY	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB	ICDRF	ICDRE	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.

16.3.6 I²C Bus Status Register (ICSR)

ICSR consists of status flags. Also see tables 16.4 and 16.5.

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 in continuous transmission/reception. 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. Transmit mode: [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: 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. If the ICSR register bit is written using bit-manipulation instructions, the acknowledge data should be re-set since the acknowledge data setting is rewritten by the ACKB bit reading value. Write the ACKE bit to 0 to clear the ACKB flag to 0, before transmission is ended and a stop condition is issued in master mode, or before transmission is ended and SDA is released to issue a stop condition by a master device.

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

16.3.7 DDC Switch Register (DDCSWR)

DDCSWR 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 Controls initialization of the internal state of IIC_0 and IIC_1. 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 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.
2	CLR2	1	W*	
1	CLR1	1	W*	
0	CLR0	1	W*	

Note: * This bit is always read as 1.

16.3.8 I²C Bus Extended Control Register (ICXR)

ICXR enables or disables the I²C bus interface interrupt generation and continuous receive operation, 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	Handshake Receive Operation Select Enables or disables continuous receive operation in receive mode. 0: Enables continuous receive operation 1: Disables continuous receive operation When the HNDS bit is cleared to 0, receive operation is performed continuously after data has been received successfully while ICDRF flag is 0. When the HNDS bit is set to 1, SCL is fixed to the low level and the next data transfer is disabled after data has been received successfully while the ICDRF flag is 0. The bus line is released and next receive operation is enabled by reading the receive data in 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 Bit
0	FNC0	0	R/W	Cancels some restrictions on usage. For details, see section 16.6, Usage Notes. 00: Restrictions on operation remaining in effect 01: Setting prohibited 10: Setting prohibited 11: Restrictions on operation canceled

16.4 Operation

The I²C bus interface has an I²C bus format and a serial format.

16.4.1 I²C Bus Data Format

The I²C bus format is an addressing format with an acknowledge bit. This is shown in figure 16.3. 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 16.4.

Figure 16.5 shows the I²C bus timing.

The symbols used in figures 16.3 to 16.5 are explained in table 16.6.

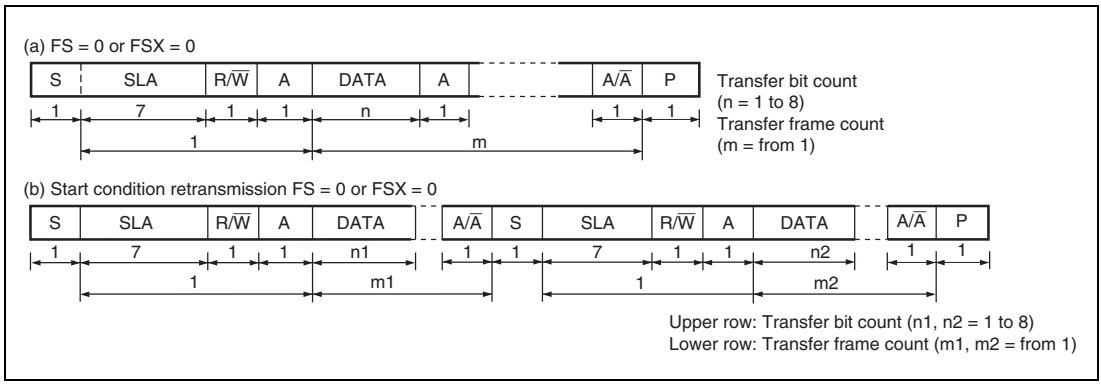


Figure 16.3 I²C Bus Data Format (I²C Bus Format)

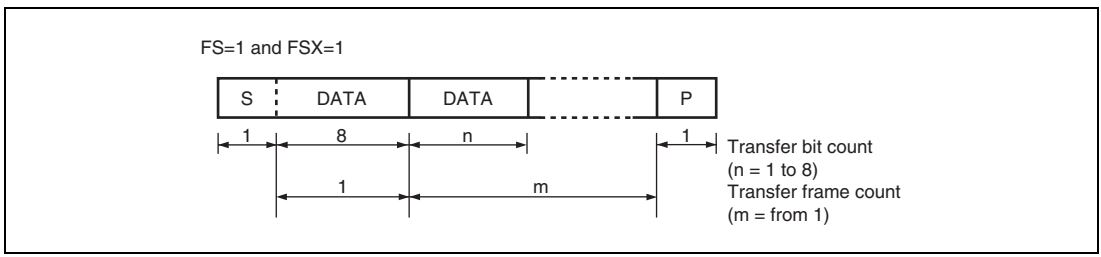


Figure 16.4 I²C Bus Data Format (Serial Format)

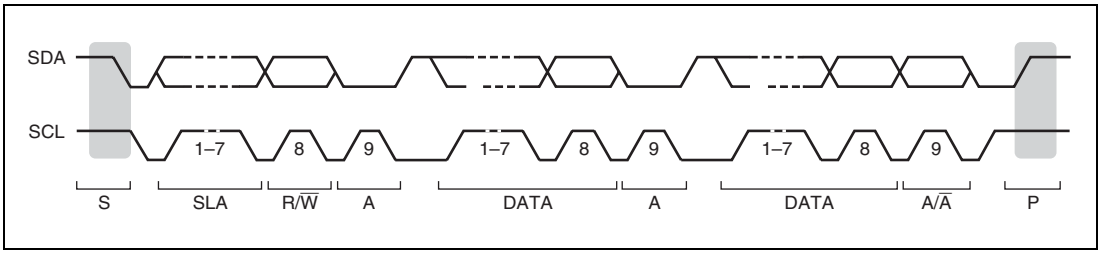


Figure 16.5 I²C Bus Timing

Table 16.6 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

16.4.2 Initialization

Initialize the IIC by the procedure shown in figure 16.6 before starting transmission/reception of data.

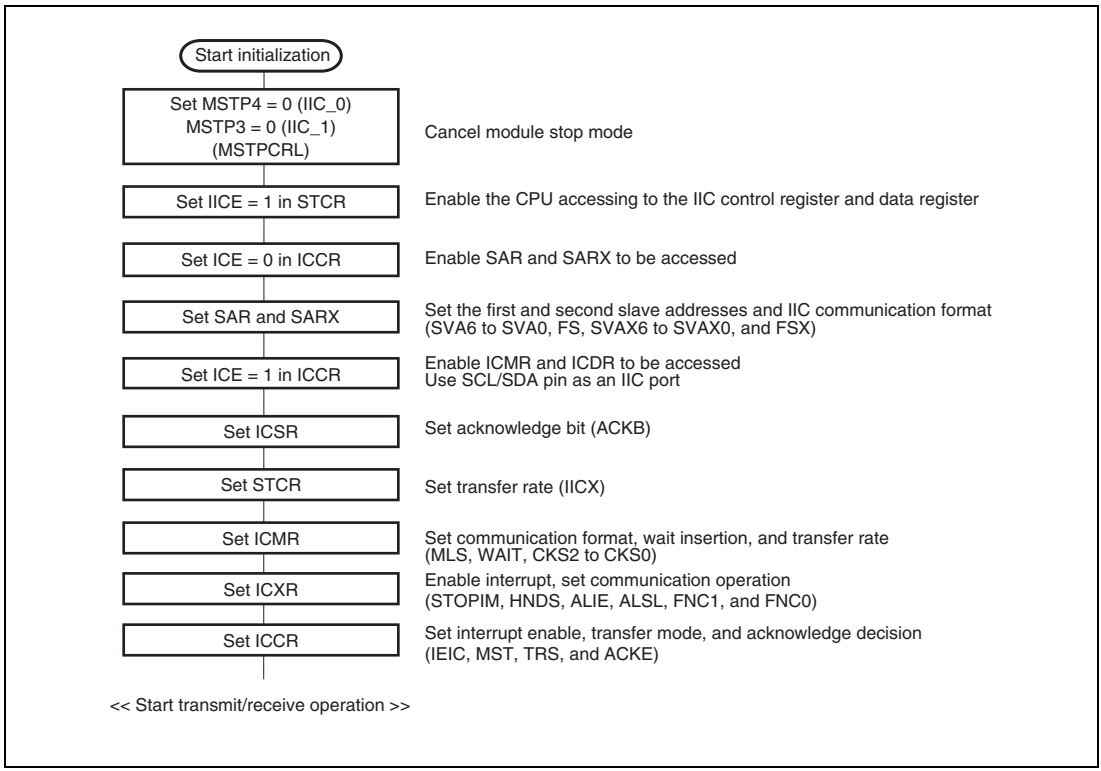


Figure 16.6 Sample Flowchart for IIC Initialization

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.

16.4.3 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 16.7 shows the sample flowchart for the operations in master transmit mode.

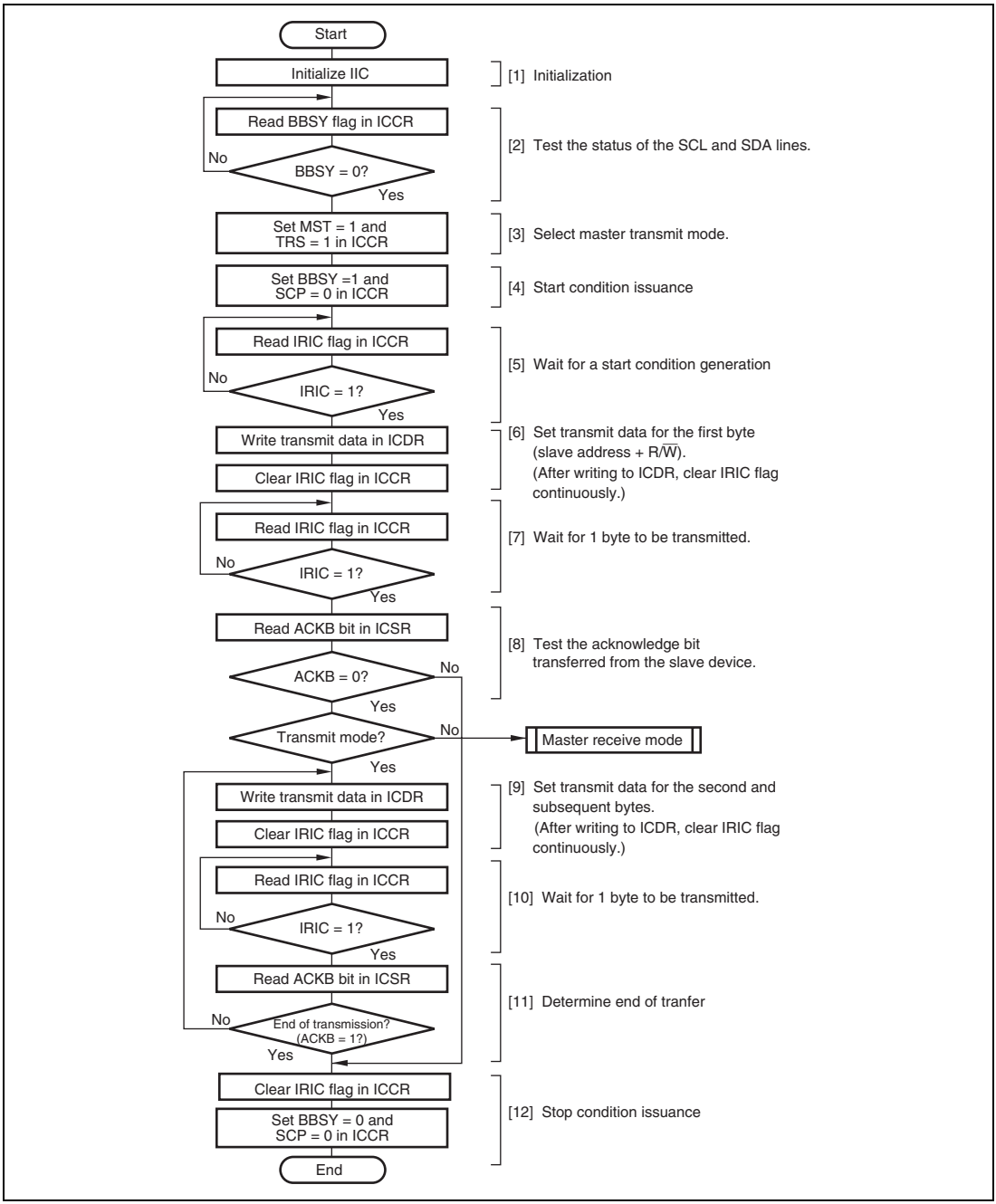


Figure 16.7 Sample Flowchart for Operations in Master Transmit Mode

The transmission procedure and operations by which data is sequentially transmitted in synchronization with ICDR (ICDRT) write operations, are described below.

1. Initialize the IIC as described in section 16.4.2, 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.

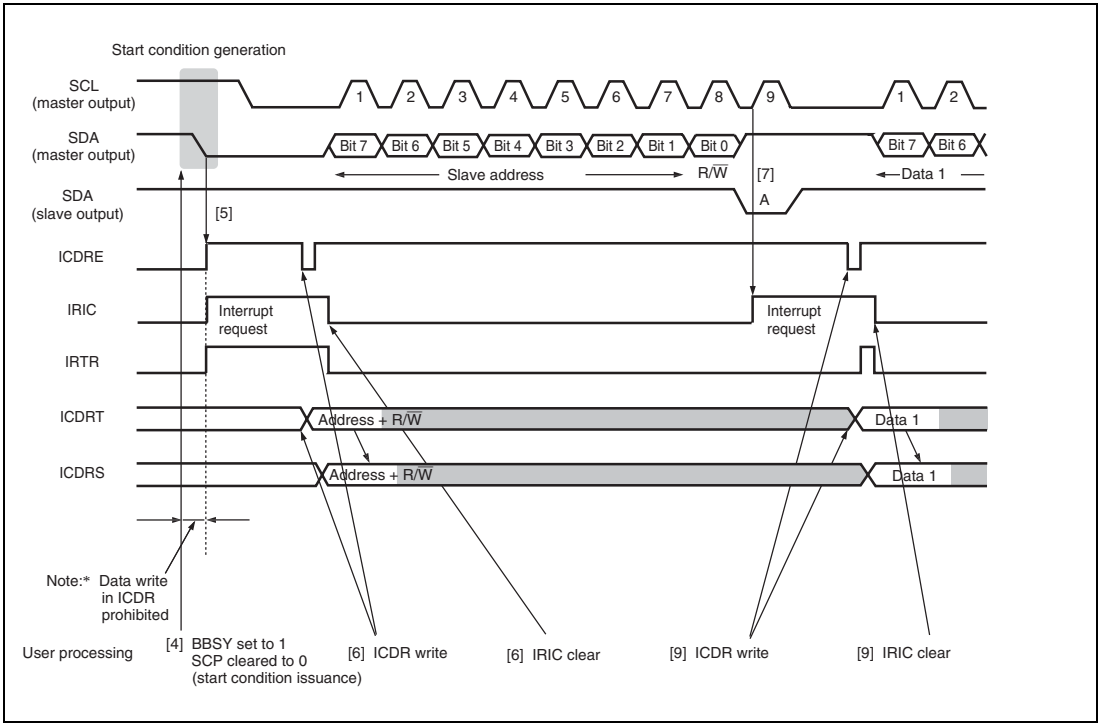


Figure 16.8 Example of Operation Timing in Master Transmit Mode (MLS = WAIT = 0)

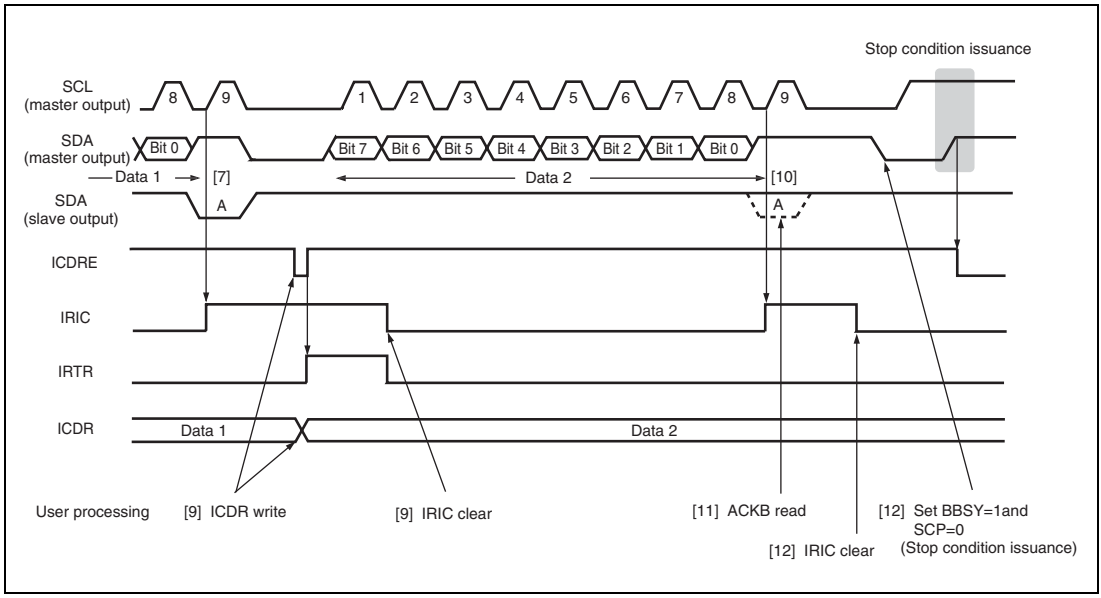


Figure 16.9 Example of Stop Condition Issuance Operation Timing in Master Transmit Mode (MLS = WAIT = 0)

16.4.4 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.

(1) Receive Operation Using the HNDS Function (HNDS = 1)

Figure 16.10 shows the sample flowchart for the operations in master receive mode (HNDS = 1).

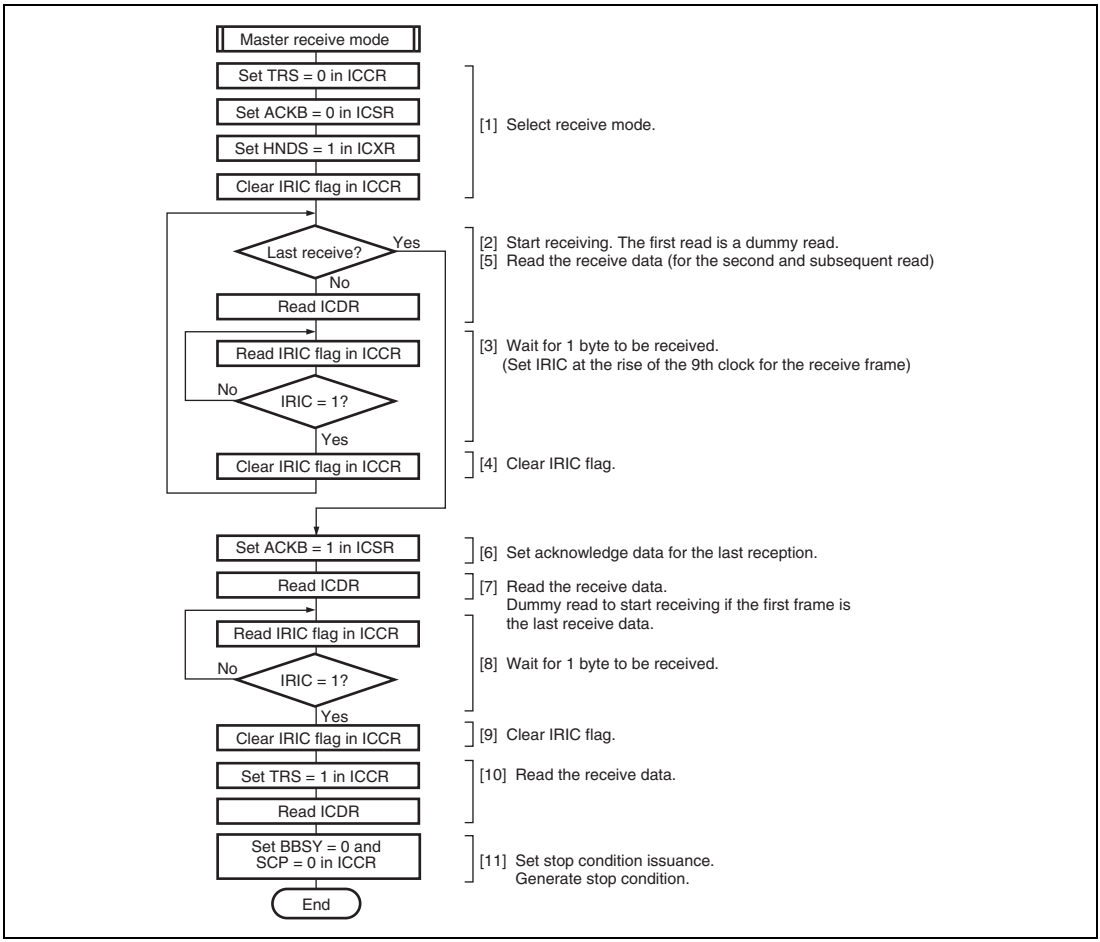


Figure 16.10 Sample Flowchart for Operations in Master Receive Mode (HNDS = 1)

The reception procedure and operations using the HNDS function, by which the data reception process is provided in 1-byte units with SCL fixed low at each data reception, 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).
   Set the HNDS bit in ICXR to 1.
   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.

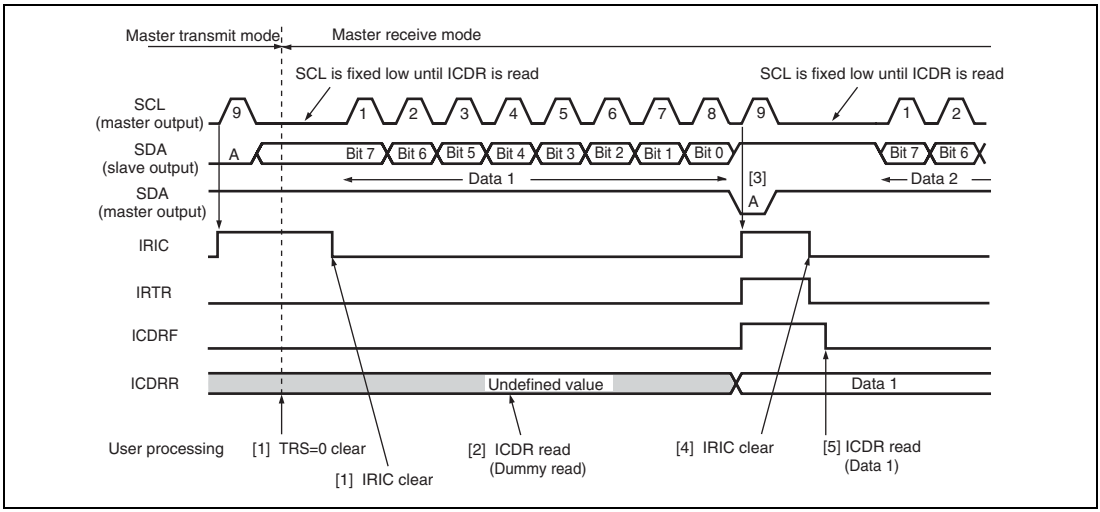


Figure 16.11 Example of Operation Timing in Master Receive Mode (MLS = WAIT = 0, HNDS = 1)

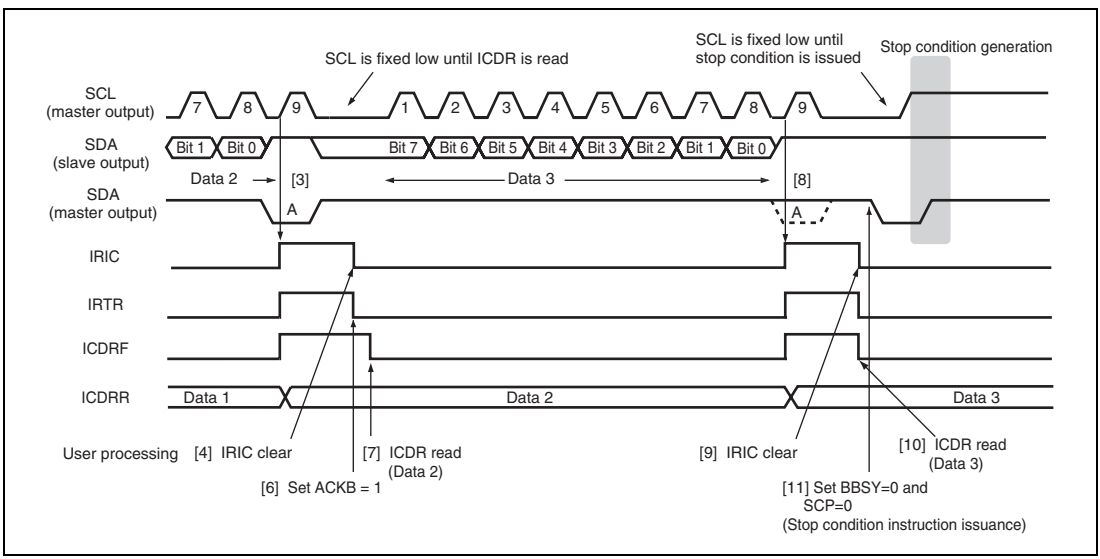


Figure 16.12 Example of Stop Condition Issuance Operation Timing in Master Receive Mode (MLS = WAIT = 0, HNDS = 1)

(2) Receive Operation Using the Wait Function

Figures 16.13 and 16.14 show the sample flowcharts for the operations in master receive mode (WAIT = 1).

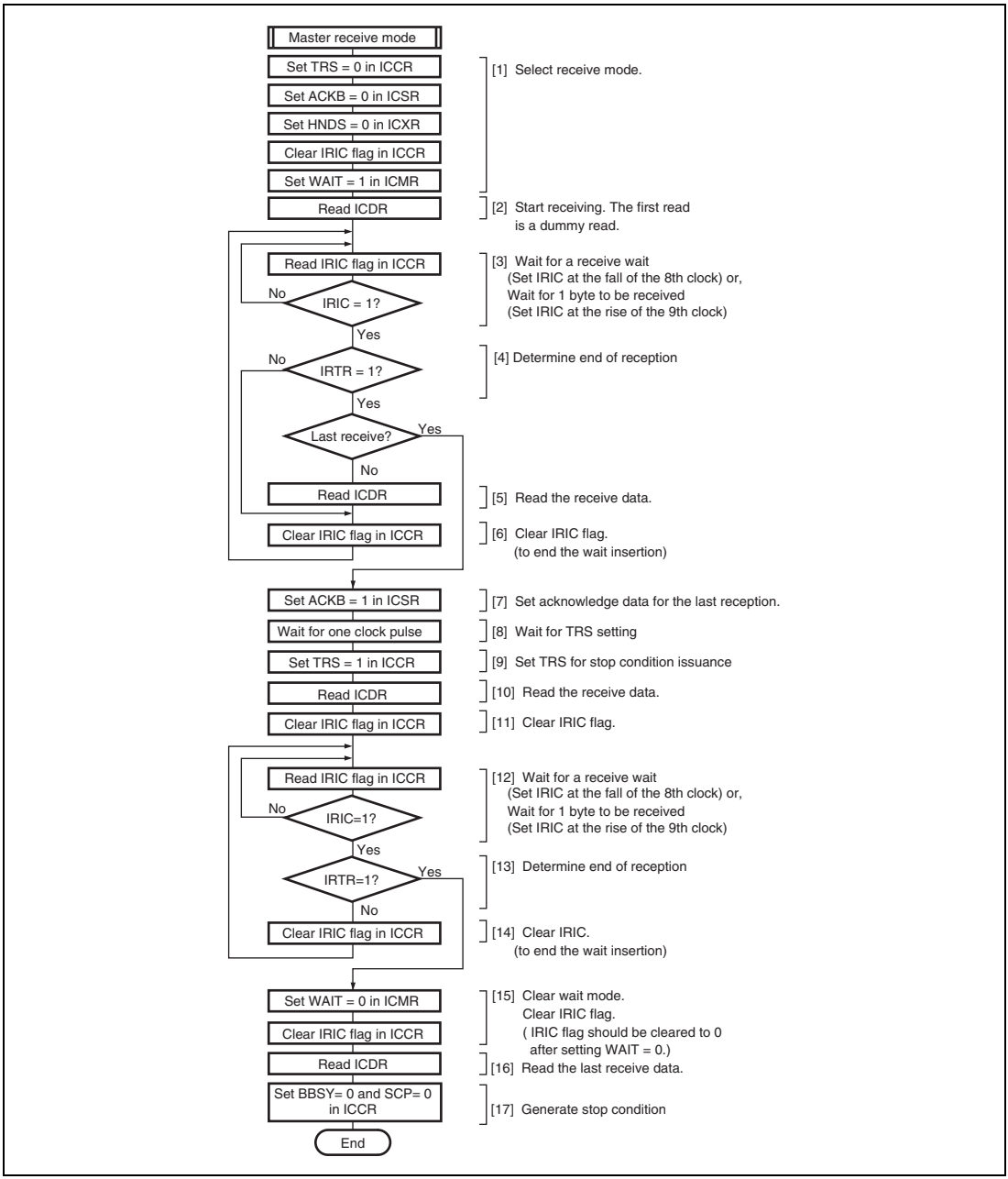


Figure 16.13 Sample Flowchart for Operations in Master Receive Mode (receiving multiple bytes) (WAIT = 1)

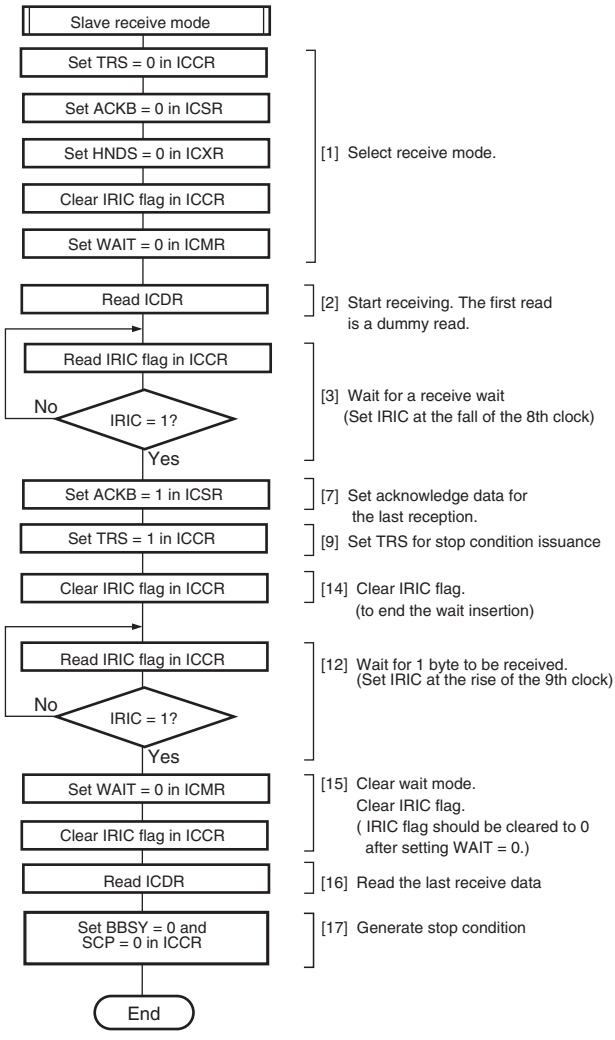


Figure 16.14 Sample Flowchart for Operations in Master Receive Mode (receiving a single byte) (WAIT = 1)

The reception procedure and operations using the wait function (WAIT bit), by which data is sequentially received in synchronization with ICDR (ICDRR) read operations, are described below.

The following describes the multiple-byte reception procedure. In single-byte reception, some steps of the following procedure are omitted. At this time, follow the procedure shown in figure 16.14.

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 to set the acknowledge data.
   Clear the HNDS bit in ICXR to 0 to cancel the handshake function.
   Clear the IRIC flag to 0, and then set the WAIT bit in ICMR to 1.
2. When ICDR is read (dummy data is read), reception is started, the receive clock is output in synchronization with the internal clock, and data is received.
3. The IRIC flag is set to 1 in either of the following cases. If the IEIC bit in ICCR has been set to 1, an interrupt request is sent to the CPU.
   — At the fall of the 8th receive clock pulse for one frame
     SCL is automatically fixed low in synchronization with the internal clock until the IRIC flag clearing.
   — At the rise of the 9th receive clock pulse for one frame
     The IRTR and ICDRF flags are set to 1, indicating that one frame of data has been received. The master device outputs the receive clock continuously to receive the next data.
4. Read the IRTR flag in ICSR.
   If the IRTR flag is 0, execute step [6] to clear the IRIC flag to 0 to release the wait state.
   If the IRTR flag is 1 and the next data is the last receive data, execute step [7] to halt reception.
5. If IRTR flag is 1, read ICDR receive data.
6. Clear the IRIC flag. When the flag is set as the first case in step [3], the master device outputs the 9th clock and drives SDA low at the 9th receive clock pulse to return an acknowledge signal.

Data can be received continuously by repeating steps [3] to [6].

7. Set the ACKB bit in ICSR to 1 so as to return the acknowledge data for the last reception.
8. After the IRIC flag is set to 1, wait for at least one clock pulse until the rise of the first clock pulse for the next receive data.
9. Set the TRS bit in ICCR to 1 to switch from receive mode to transmit mode. The TRS bit value becomes valid when the rising edge of the next 9th clock pulse is input.
10. Read the ICDR receive data.

11. Clear the IRIC flag to 0.
12. The IRIC flag is set to 1 in either of the following cases.
    - At the fall of the 8th receive clock pulse for one frame

      SCL is automatically fixed low in synchronization with the internal clock until the IRIC flag is cleared.
    - At the rise of the 9th receive clock pulse for one frame

      The IRTR and ICDRF flags are set to 1, indicating that one frame of data has been received. The master device outputs the receive clock continuously to receive the next data.
13. Read the IRTR flag in ICSR.

    If the IRTR flag is 0, execute step [14] to clear the IRIC flag to 0 to release the wait state.

    If the IRTR flag is 1 and data reception is complete, execute step [15] to issue the stop condition.
14. If IRTR flag is 0, clear the IRIC flag to 0 to release the wait state.

    Execute step [12] to read the IRIC flag to detect the end of reception.
15. Clear the WAIT bit in ICMR to cancel the wait mode.

    Then, clear the IRIC flag. Clearing of the IRIC flag should be done while WAIT = 0. (If the WAIT bit is cleared to 0 after clearing the IRIC flag and then an instruction to issue a stop condition is executed, the stop condition may not be issued correctly.)
16. Read the last ICDR receive data.
17. Clear the BBSY bit and SCP bit to 0 in ICCR. This changes SDA from low to high when SCL is high, and generates the stop condition.

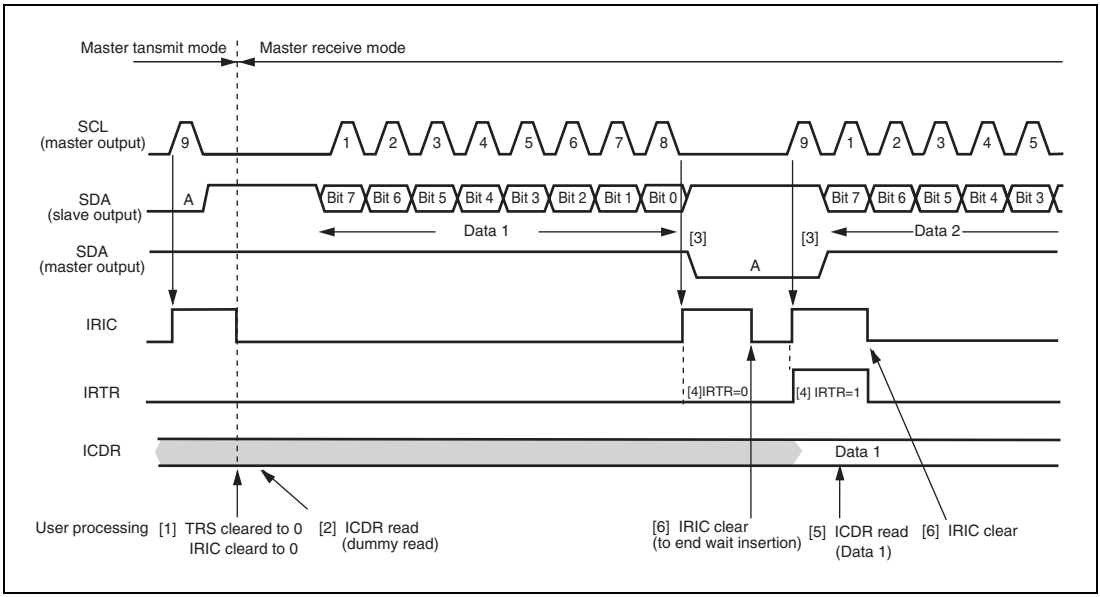


Figure 16.15 Example of Master Receive Mode Operation Timing (MLS = ACKB = 0, WAIT = 1)

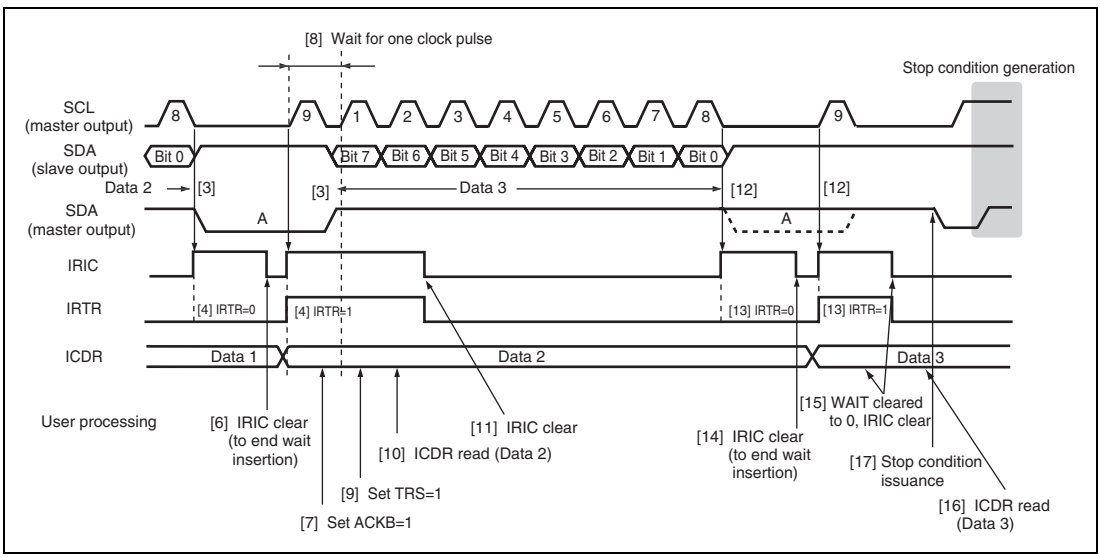


Figure 16.16 Example of Stop Condition Issuance Timing in Master Receive Mode (MLS = ACKB = 0, WAIT = 1)

16.4.5 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.

(1) Receive Operation Using the HNDS Function (HNDS = 1)

Figure 16.17 shows the sample flowchart for the operations in slave receive mode (HNDS = 1).

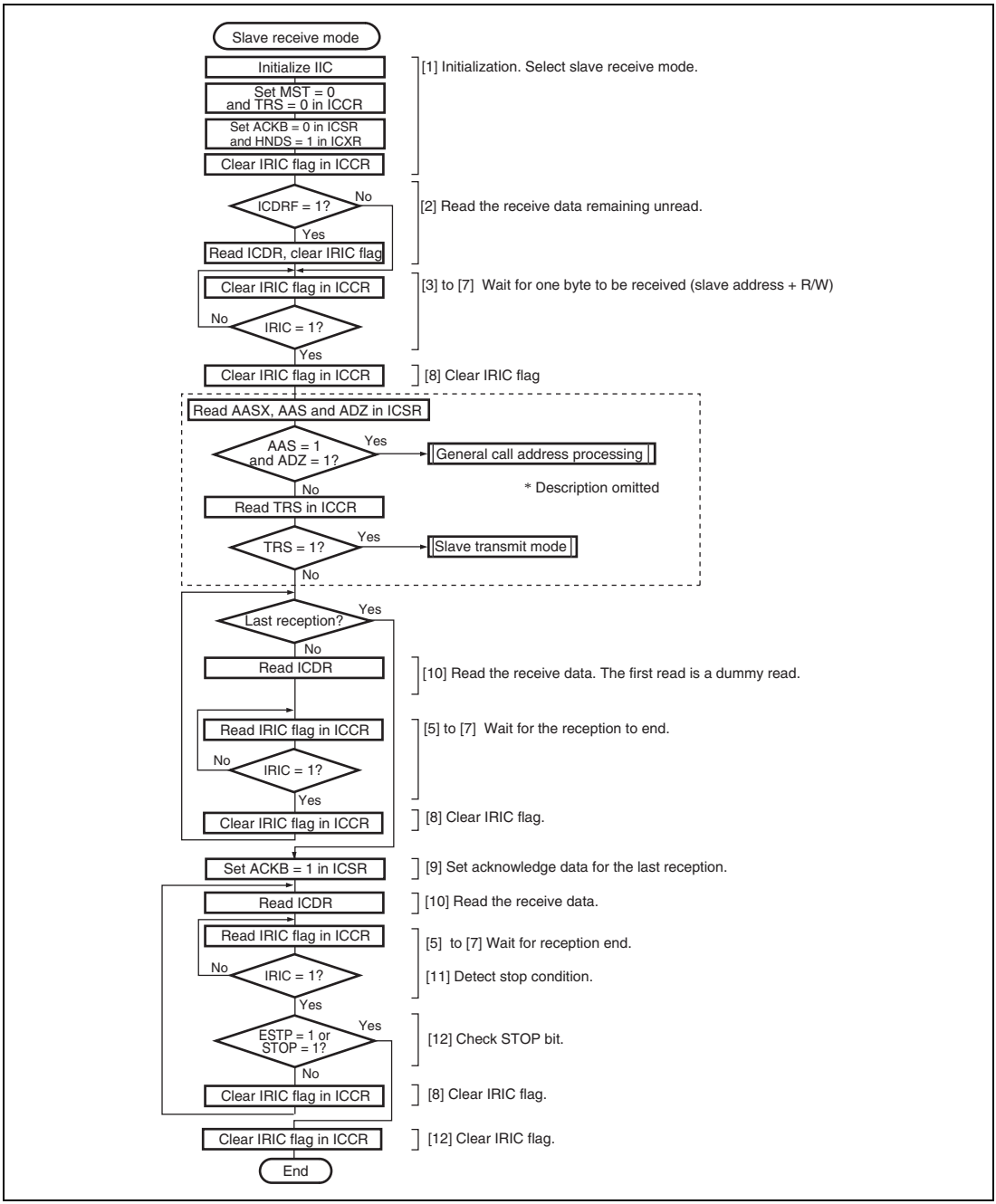


Figure 16.17 Sample Flowchart for Operations in Slave Receive Mode (HNDS = 1)

The reception procedure and operations using the HNDS bit function, by which data reception process is provided in 1-byte unit with SCL being fixed low at every data reception, are described below.

1. Initialize the IIC as described in section 16.4.2, Initialization.

   Clear the MST and TRS bits to 0 to set slave receive mode, and set the HNDS bit to 1 and 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.

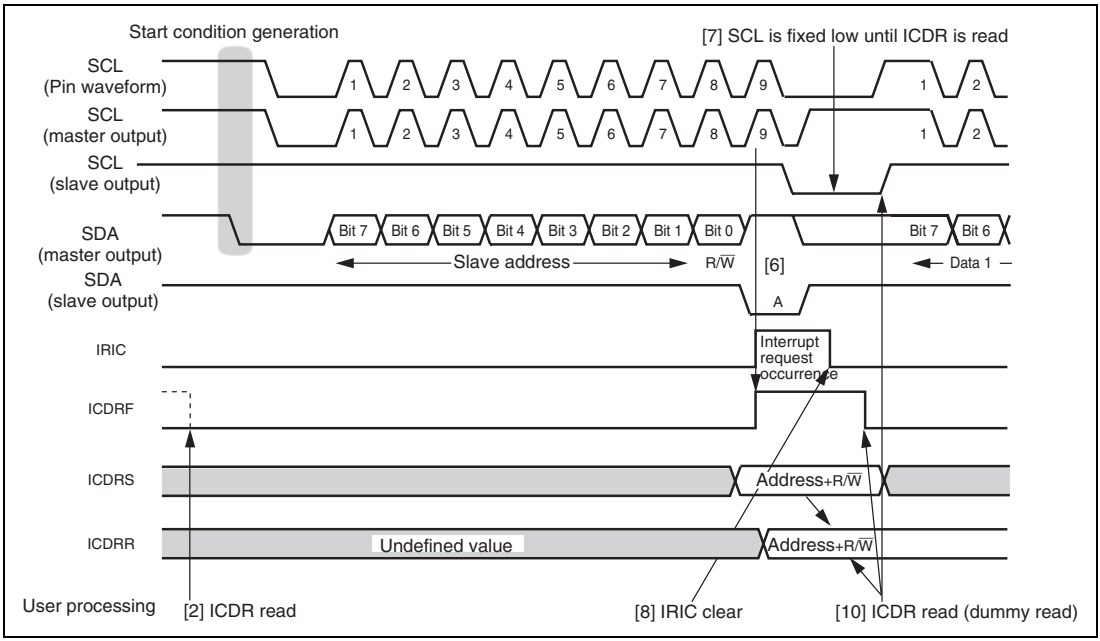


Figure 16.18 Example of Slave Receive Mode Operation Timing (1)
(MLS = 0, HNDS= 1)

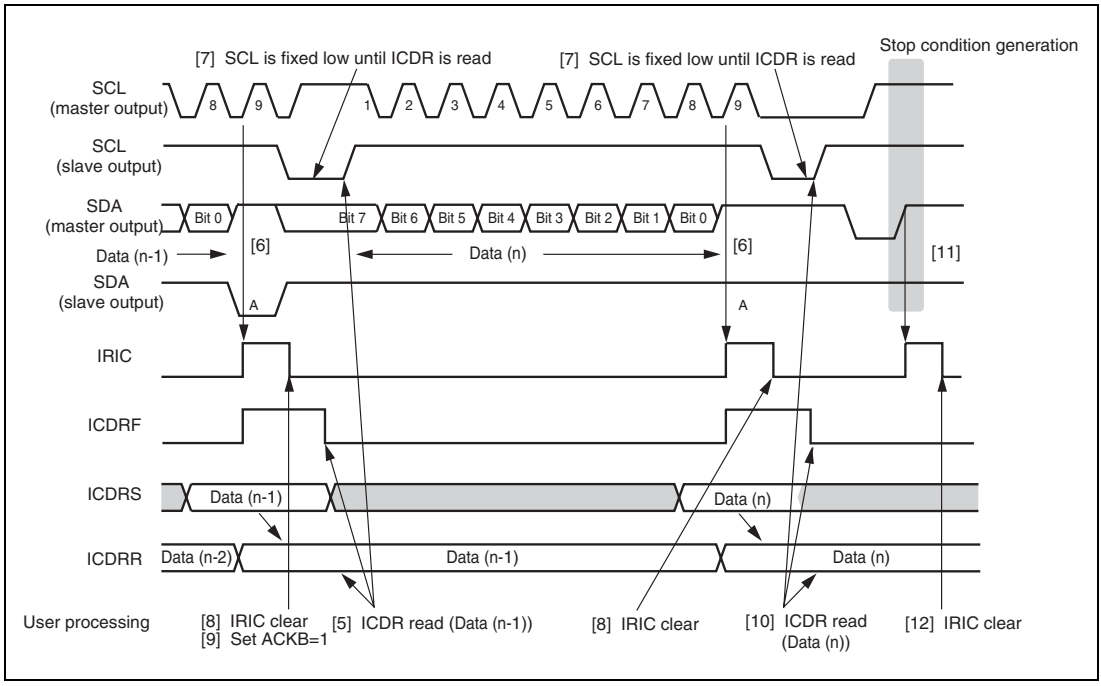


Figure 16.19 Example of Slave Receive Mode Operation Timing (2) (MLS = 0, HNDS= 1)

(2) Continuous Receive Operation

Figure 16.20 shows the sample flowchart for the operations in slave receive mode (HNDS = 0).

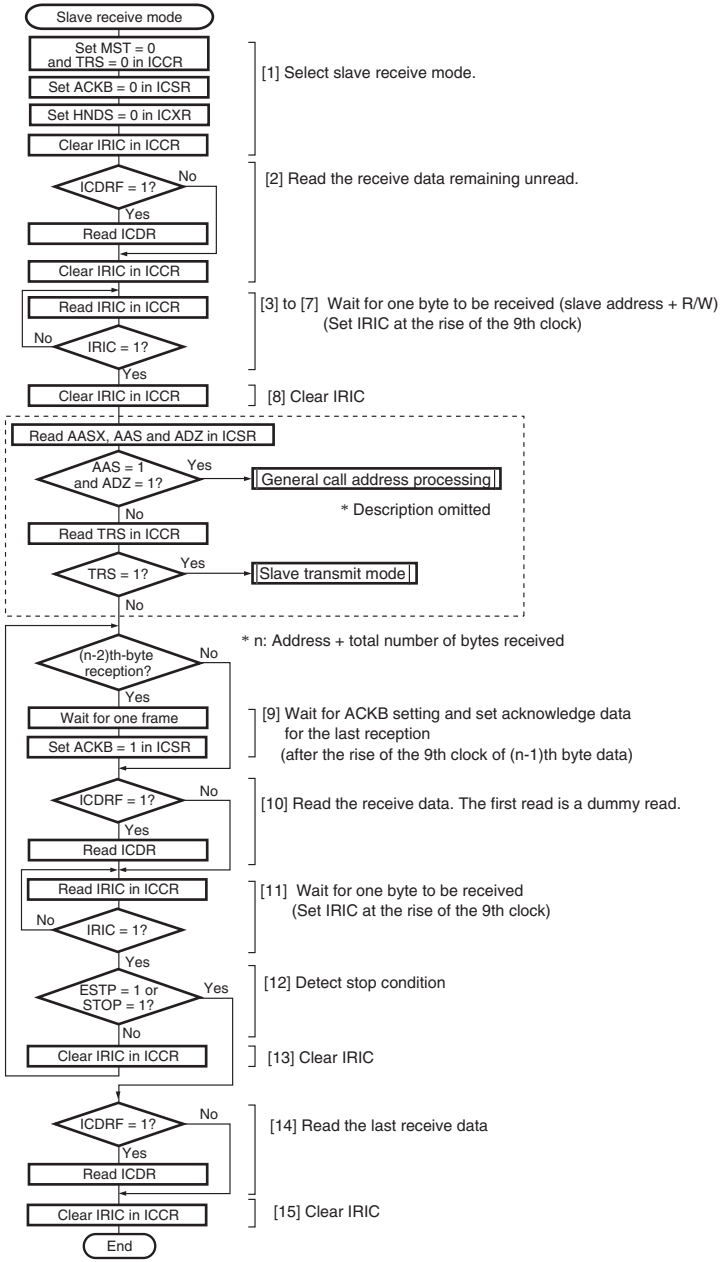


Figure 16.20 Sample Flowchart for Operations in Slave Receive Mode (HNDS = 0)

The reception procedure and operations in slave receive are described below.

1. Initialize the IIC as described in section 16.4.2, Initialization.

   Clear the MST and TRS bits to 0 to set slave receive mode, and set the HNDS and ACKB bits 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 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, the IRTR flag is also set to 1.
7. At the rise of the 9th clock pulse, the receive data is transferred from ICDRS to ICDRR, setting the ICDRF flag to 1.
8. Confirm that the STOP bit is cleared to 0 and clear the IRIC flag to 0.
9. If the next read data is the third last receive frame, wait for at least one frame time to set the ACKB bit. Set the ACKB bit after the rise of the 9th clock pulse of the second last receive frame.
10. Confirm that the ICDRF flag is set to 1 and read ICDR. This clears the ICDRF flag to 0.
11. At the rise of the 9th clock pulse or when the receive data is transferred from IRDRS to ICDRR due to ICDR read operation, the IRIC and ICDRF flags are set to 1.
12. When the stop condition is detected (SDA is changed from low to high when SCL is high), the BBSY flag is cleared to 0 and the STOP or ESTP flag is set to 1. If the STOPIM bit has been cleared to 0, the IRIC flag is set to 1. In this case, execute step [14] to read the last receive data.
13. Clear the IRIC flag to 0.

Receive operations can be performed continuously by repeating steps [9] to [13].

14. Confirm that the ICDRF flag is set to 1, and read ICDR.
15. Clear the IRIC flag.

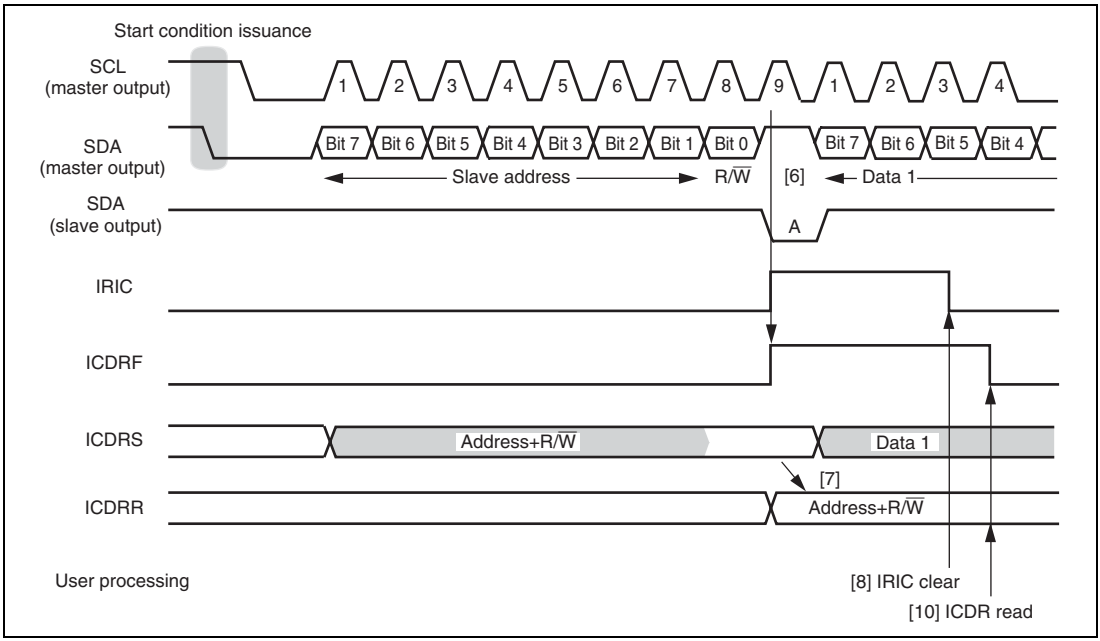


Figure 16.21 Example of Slave Receive Mode Operation Timing (1)
(MLS = ACKB = 0, HNDS = 0)

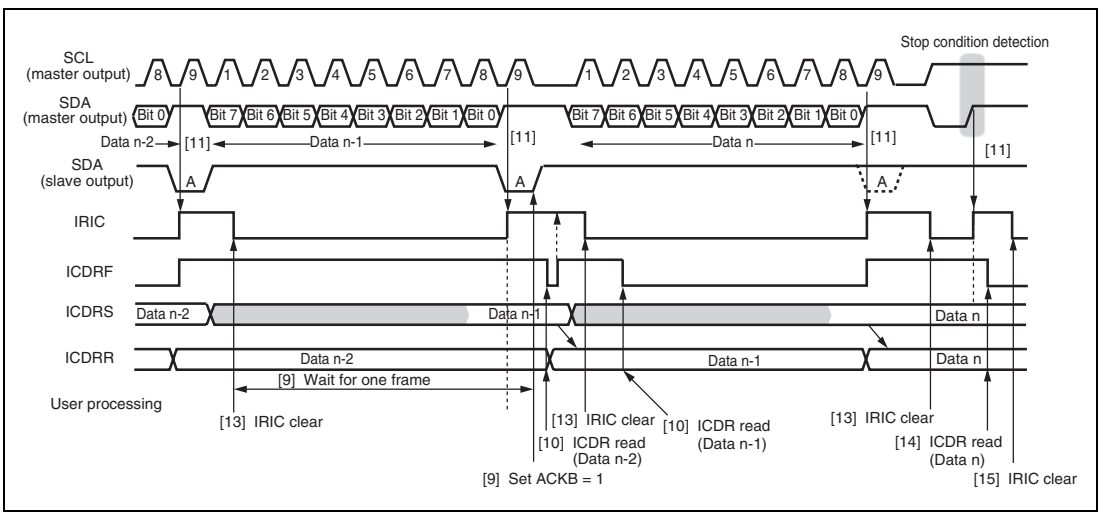


Figure 16.22 Example of Slave Receive Mode Operation Timing (2)
(MLS = ACKB = 0, HNDS = 0)

16.4.6 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 16.23 shows the sample flowchart for the operations in slave transmit mode.

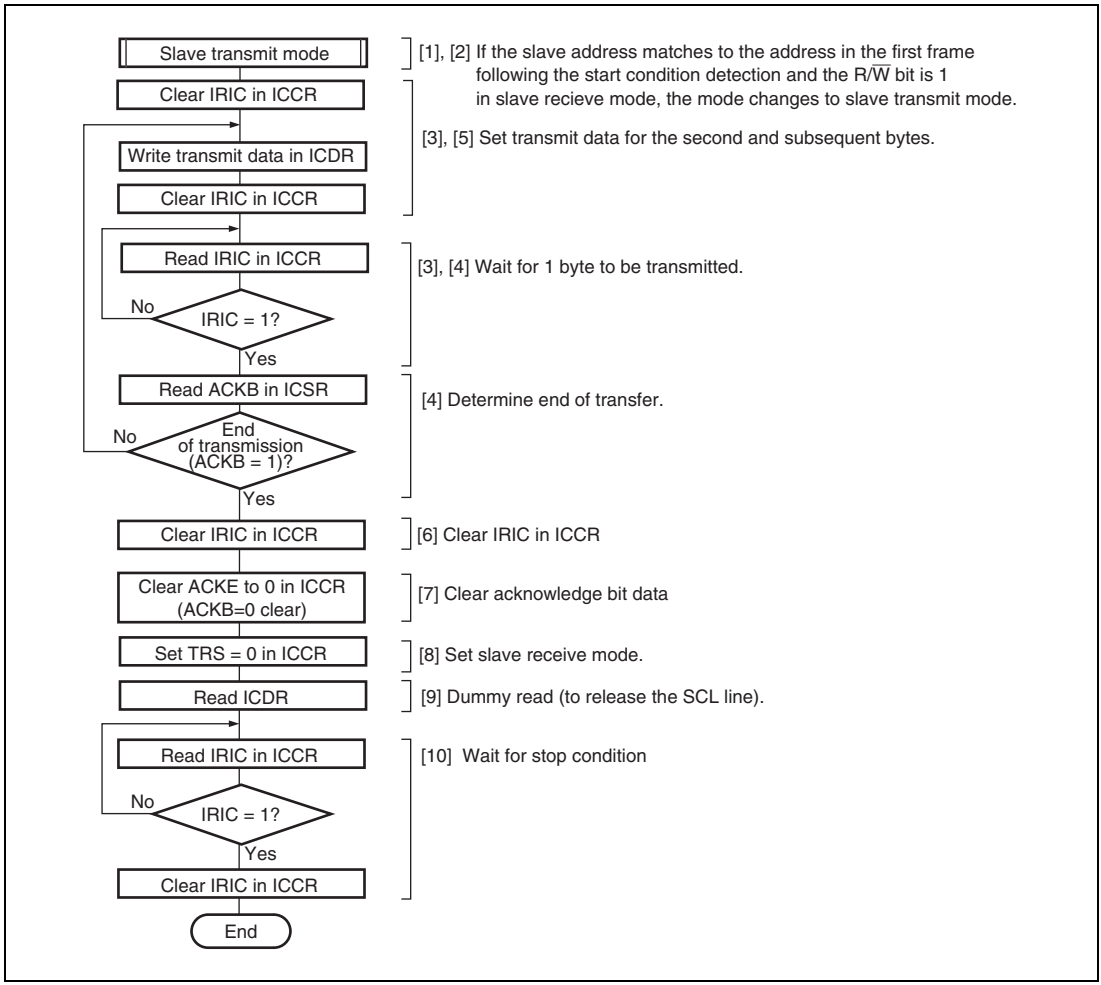


Figure 16.23 Sample Flowchart for Slave Transmit Mode

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.

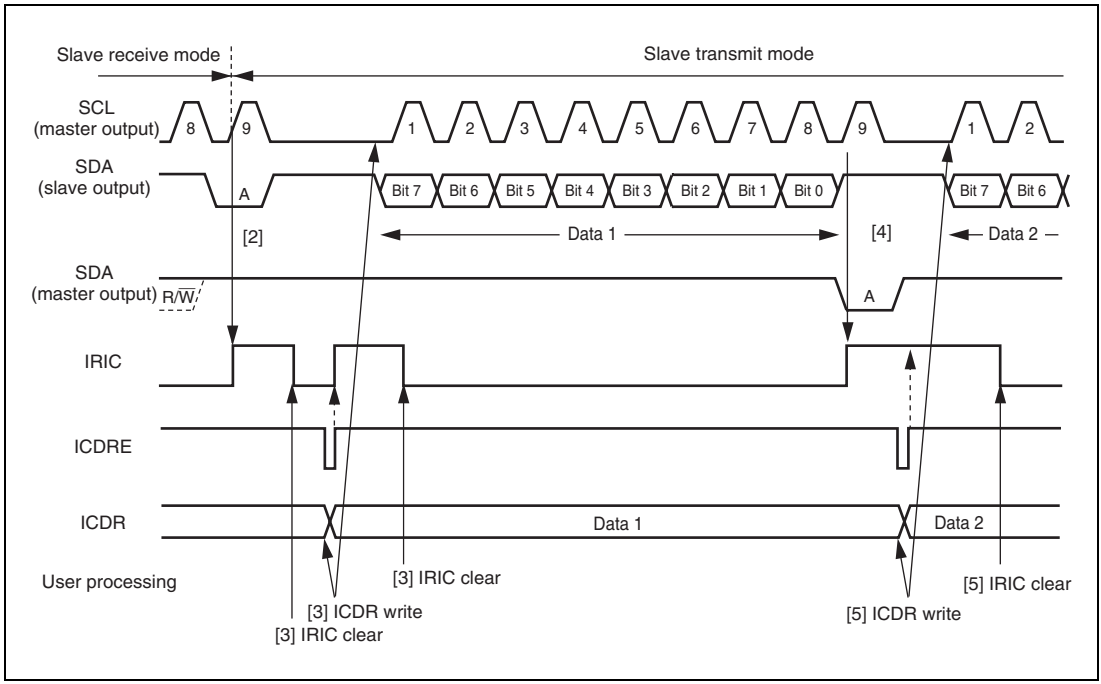


Figure 16.24 Example of Slave Transmit Mode Operation Timing (MLS = 0)

16.4.7 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 16.25 to 16.27 show the IRIC set timing and SCL control.

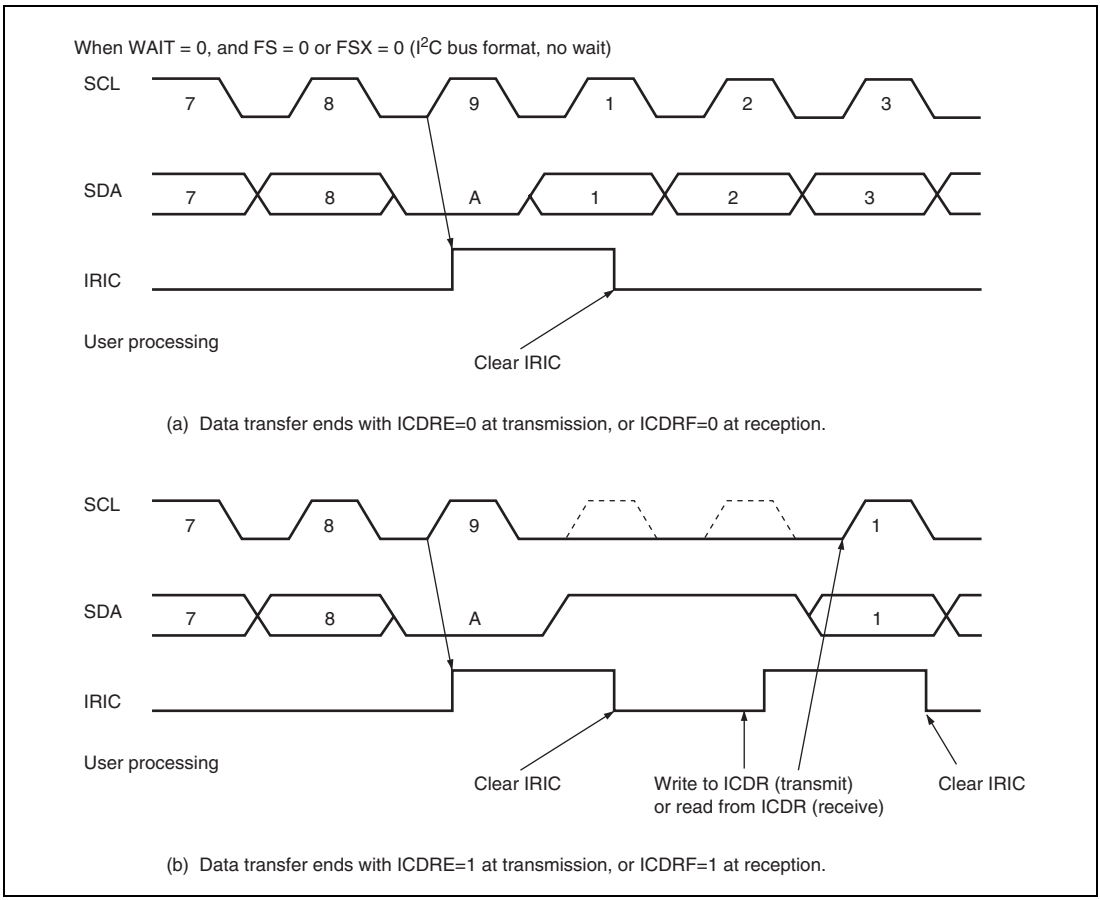
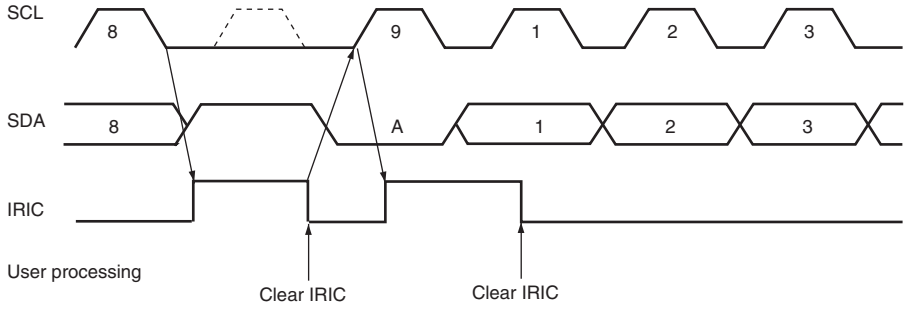
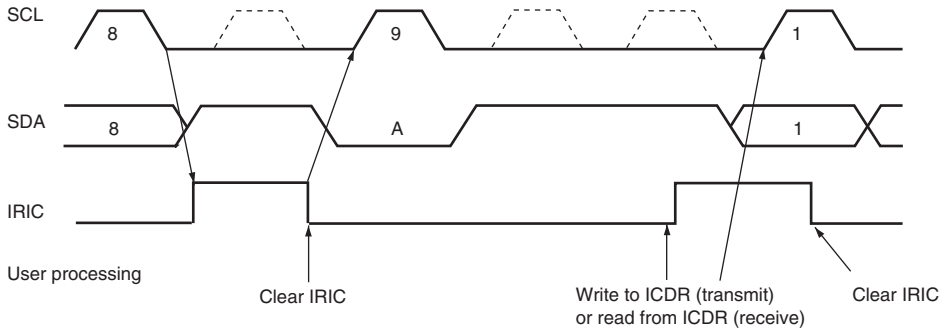


Figure 16.25 IRIC Setting Timing and SCL Control (1)

When WAIT = 1, and FS = 0 or FSX = 0 (I²C bus format, wait inserted)



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



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

Figure 16.26 IRIC Setting Timing and SCL Control (2)

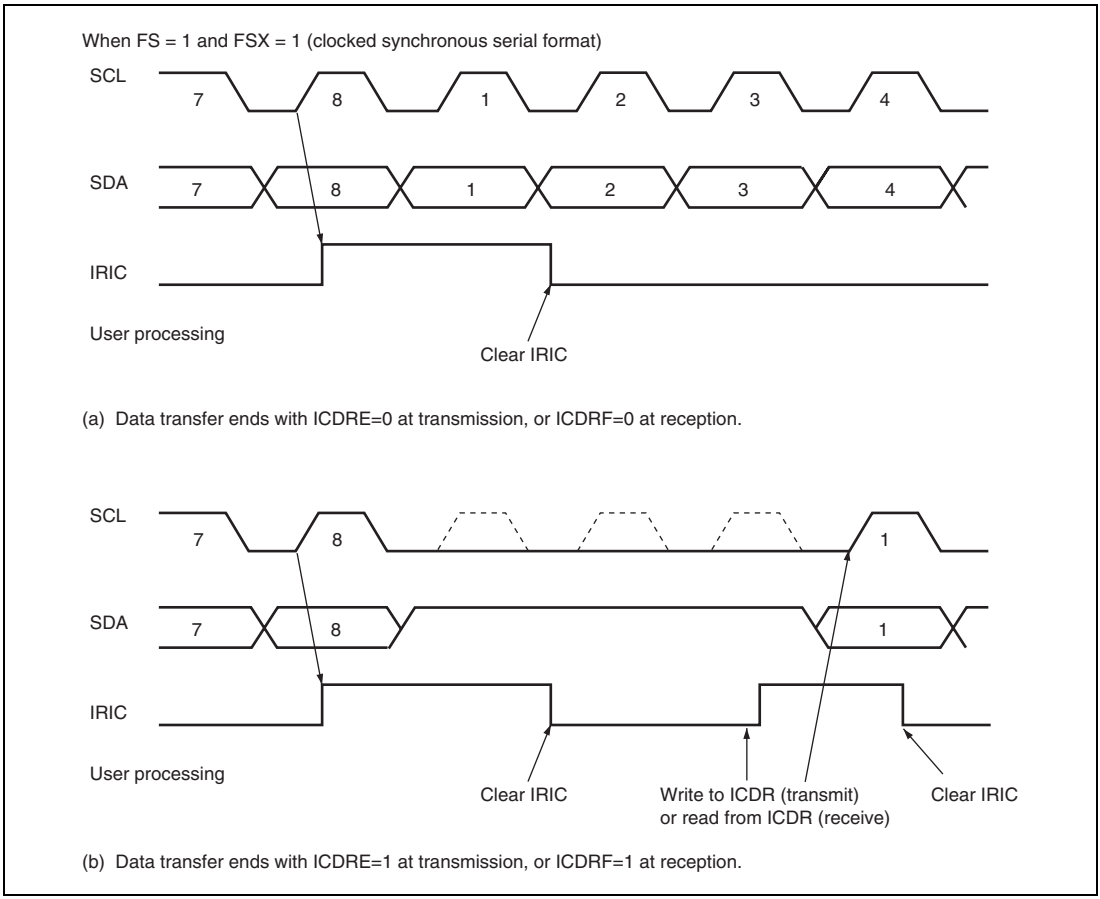


Figure 16.27 IRIC Setting Timing and SCL Control (3)

16.4.8 Operation by Using DTC

This LSI provides the DTC to allow consecutive transfer. The DTC is activated when the IRTR flag which is one of two interrupt flags (IRIC and IRTR) is set to 1. When the ACKE bit is cleared to 0, regardless of the acknowledge bit, the ICDRE, IRIC, and IRTR flags are set at the completion of the data transfer. When the ACKE bit is set to 1, if data transmission has been completed with the acknowledge bit of 0, the ICDRE, IRIC, and IRTR flags are set. When the ACKE bit is set to 1, if data transmission has been completed with the acknowledge bit of 1, only the IRIC flag is set.

When the DTC is activated, the ICDRE, IRIC, and IRTR flags are cleared to 0 after required transfers has been performed. Therefore, any interrupt is occurred while data is transferred continuously. However, when the ACKE bit is set to 1, if the data transmission has been completed with the acknowledge bit of 1, the DTC is not be activated and an interrupt is occurred if enabled.

According to the reception device, the acknowledge bit indicates the completion of receive data processing or is fixed to 1 without any indication.

In the I²C bus format, the selection of a slave device and transfer direction by the slave address and R/$\overline{W}$ bit, and confirming reception and indicating the last frame by the acknowledge bit are performed. Therefore, the consecutive data transfer by the DTC should be executed with the processing of CPU by interrupts.

Table 16.7 shows the sample processing by using the DTC. It is supposed that the number of transfer data has been known in the slave mode.

Table 16.7 Operation by Using DTC

Item	Master Transmit Mode	Master Receive Mode	Slave Transmit Mode	Slave Receive Mode
Slave address + R/$\overline{W}$ bit transmission/ reception	Transmission by DTC (ICDR write)	Transmission by CPU (ICDR write)	Reception by CPU (ICDR read)	Reception by CPU (ICDR read)
Dummy data read	—	Processing by CPU (ICDR read)	—	—
I²C data transmission/ reception	Transmission by DTC (ICDR write)	Transmission by DTC (ICDR read)	Transmission by DTC (ICDR write)	Reception by DTC (ICDR read)
Dummy data (H'FF) write	—	—	Processing by DTC (ICDR write)	—
Last frame processing	Not required	Reception by CPU (ICDR read)	Not required	Reception by CPU (ICDR read)
Transfer request processing after last frame processing completed	1st time: Clearing by CPU 2nd time: Stop condition issue by CPU	Not required	Dummy data (H'FF) Stop condition detection and automatic clear during transmission	Not required
Set the number of frames of DTC transfer data	Transmission: The number of actual data + 1 (+ 1 = slave address + R/$\overline{W}$ bit)	Reception: The number of actual data	Transmission: The number of actual data + 1 (+ 1 = dummy data (H'FF))	Reception: The number of actual data

16.4.9 Noise Canceller

The logic levels at the SCL and SDA pins are routed through noise cancellers before being latched internally. Figure 16.28 shows a block diagram of the noise canceller.

The noise canceller 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.

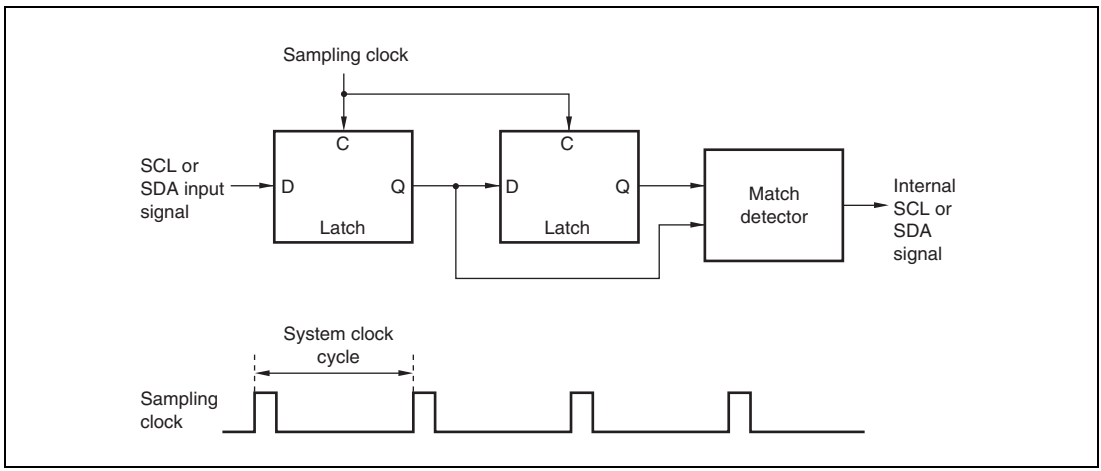


Figure 16.28 Block Diagram of Noise Canceller

16.4.10 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 DDCSWR or clearing ICE bit. For details on the setting of bits CLR3 to CLR0, see section 16.3.7, DDC Switch Register (DDCSWR).

(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 DDCSWR, 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.

16.5 Interrupt Sources

The IIC has interrupt source IICI. Table 16.8 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 16.8 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	Low

16.6 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. To output the stop condition followed by the start condition*, after issuing the instruction that generates the start condition, read DR in each I²C bus output pin, and check that SCL and SDA are both low. The pin states can be monitored by reading DR even if the ICE bit is set to 1. Then issue the instruction that generates the stop condition. Note that SCL may not yet have gone low when BBSY is cleared to 0.

Note: * An illegal procedure in the I²C bus specification.

2. Either of the following two conditions will start the next transfer. Pay attention to these conditions when accessing to 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 16.9 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 16.9 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 26.23
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. SCL and SDA inputs are sampled in synchronization with the internal clock. The AC timing therefore depends on the system clock cycle t_{cyc}, as shown in section 26, Electrical Characteristics. Note that the I²C bus interface AC timing specifications will not be met with a system clock frequency of less than 5 MHz.
5. 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 16.10.

Table 16.10 Permissible SCL Rise Time (t_{sr}) Values

			Time Indication [ns]					
IICX	t_{cyc} Indication		I²C Bus Specification (Max.)	φ = 5 MHz	φ = 8 MHz	φ = 10 MHz	φ = 16 MHz	φ = 20 MHz
0	7.5 t_{cyc}	Standard mode	1000	1000	937	750	468	375
		High-speed mode	300	300	300	300	300	300
1	17.5 t_{cyc}	Standard mode	1000	1000	1000	1000	1000	875
		High-speed mode	300	300	300	300	300	300

6. 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 16.9. However, because of the rise and fall times, the I²C bus interface specifications may not be satisfied at the maximum transfer rate. Table 16.11 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 16.11 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 = 5 MHz	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	4000
		High-speed mode	–300	600	950	950	950	950	950
t_{SCLLO}	0.5 t_{SCLO} ($-t_{Sf}$)	Standard mode	–250	4700	4750	4750	4750	4750	4750
		High-speed mode	–250	1300	1000*[1]	1000*[1]	1000*[1]	1000*[1]	1000*[1]
t_{BUFO}	0.5 t_{SCLO} –1 t_{cyc} ($-t_{Sr}$)	Standard mode	–1000	4700	3800*[1]	3875*[1]	3900*[1]	3939*[1]	3950*[1]
		High-speed mode	–300	1300	750*[1]	825*[1]	850*[1]	888*[1]	900*[1]
t_{STAHO}	0.5 t_{SCLO} –1 t_{cyc} ($-t_{Sf}$)	Standard mode	–250	4000	4550	4625	4650	4688	4700
		High-speed mode	–250	600	800	875	900	938	900
t_{STASO}	1 t_{SCLO} ($-t_{Sr}$)	Standard mode	–1000	4700	9000	9000	9000	9000	9000
		High-speed mode	–300	600	2200	2200	2200	2200	2200
t_{STOSO}	0.5 t_{SCLO} + 2 t_{cyc} ($-t_{Sr}$)	Standard mode	–1000	4000	4400	4250	4200	4125	4100
		High-speed mode	–300	600	1350	1200	1150	1075	1050
t_{SDASO} (master)	1 t_{SCLLO}*[3] –3 t_{cyc} ($-t_{Sr}$)	Standard mode	–1000	250	3100	3325	3400	3513	3550
		High-speed mode	–300	100	400	625	700	813	850
t_{SDASO} (slave)	1 t_{SCLL}*[3] –12 t_{cyc}*[2] ($-t_{Sr}$)	Standard mode	–1000	250	1300	2200	2500	2950	3100
		High-speed mode	–300	100	–1400*[1]	–500*[1]	–200*[1]	250	400
t_{SDAHO}	3 t_{cyc}	Standard mode	0	0	600	375	300	188	150
		High-speed mode	0	0	600	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.).

7. Notes on ICDR read at end of master reception

   To halt reception at the end of a receive operation in master receive mode, set the TRS bit to 1 and write 0 to BBSY and SCP in ICCR. This changes SDA from low to high when SCL is high, and generates the stop condition. After this, receive data can be read by means of an ICDR read, but if data remains in the buffer the ICDRS receive data will not be transferred to ICDR (ICDRR), and so it will not be possible to read the second byte of data.

   If it is necessary to read the second byte of data, issue the stop condition in master receive mode (i.e. with the TRS bit cleared to 0). When reading the receive data, first confirm that the BBSY bit in ICCR is cleared to 0, the stop condition has been generated, and the bus has been released, then read ICDR with TRS cleared to 0.

   Note that if the receive data (ICDR data) is read in the interval between execution of the instruction for issuance of the stop condition (writing of 0 to BBSY and SCP in ICCR) and the actual generation of the stop condition, the clock may not be output correctly in subsequent master transmission.

Clearing of the MST bit after completion of master transmission/reception, or other modifications of IIC control bits to change the transmit/receive operating mode or settings, must be carried out during interval (a) in figure 16.29 (after confirming that the BBSY bit in ICCR has been cleared to 0).

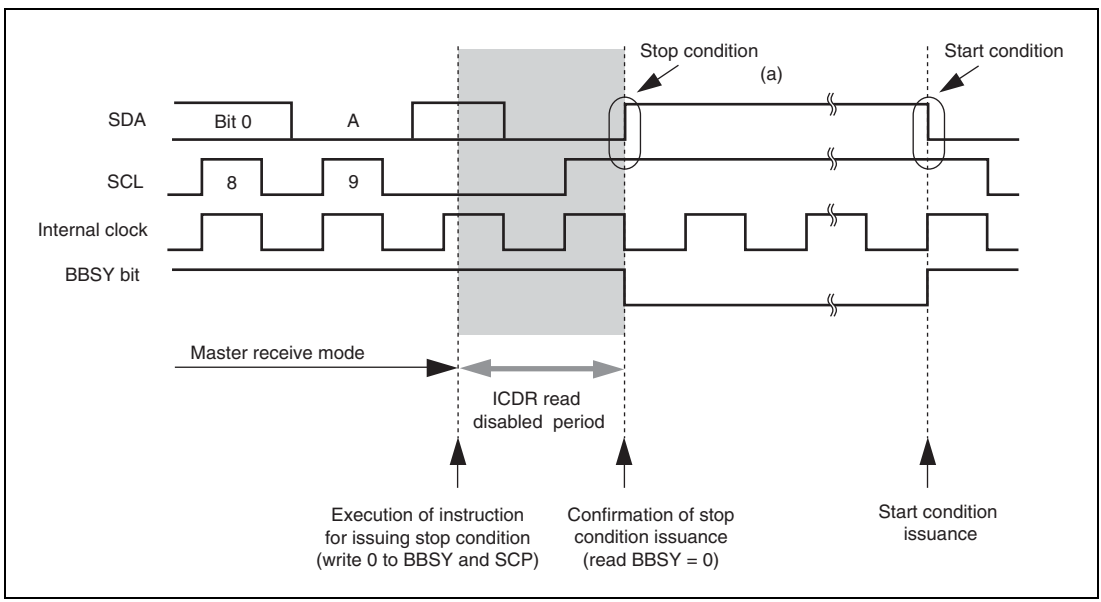


Figure 16.29 Notes on Reading Master Receive Data

Note: This restriction on usage can be canceled by setting the FNC1 and FNC0 bits to B'11 in ICXR.

8. Notes on start condition issuance for retransmission

Figure 16.30 shows the timing of start condition issuance for retransmission, and the timing for subsequently writing data to ICDR, together with the corresponding flowchart. Write the transmit data to ICDR after the start condition for retransmission is issued and then the start condition is actually generated.

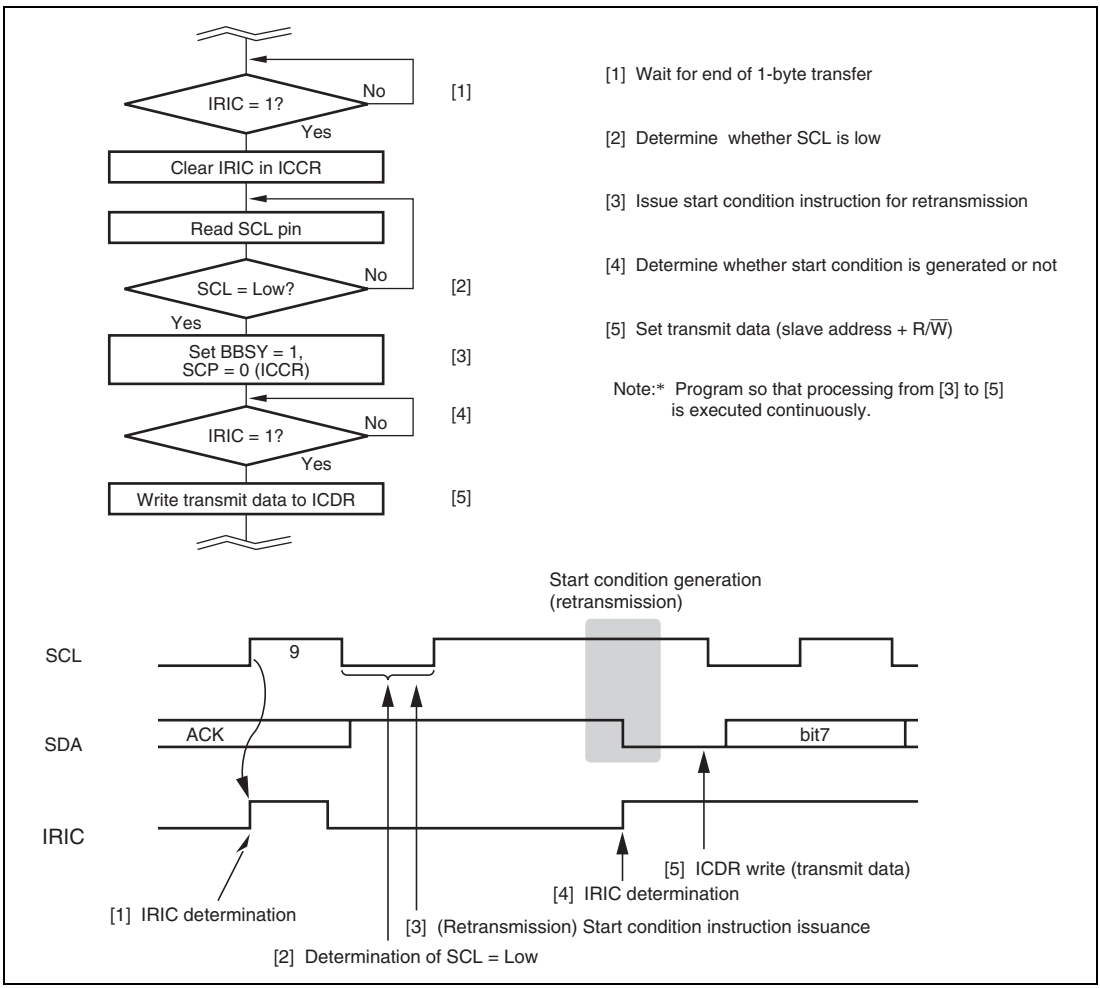


Figure 16.30 Flowchart for Start Condition Issuance Instruction for Retransmission and Timing

Note: This restriction on usage can be canceled by setting the FNC1 and FNC0 bits to B'11 in ICXR.

9. Note on when I²C bus interface stop condition instruction is issued

   In cases where the rise time of the 9th clock of SCL exceeds the stipulated value because of a large bus load capacity or where a slave device in which a wait can be inserted by driving the SCL pin low is used, the stop condition instruction should be issued after reading SCL after the rise of the 9th clock pulse and determining that it is low.

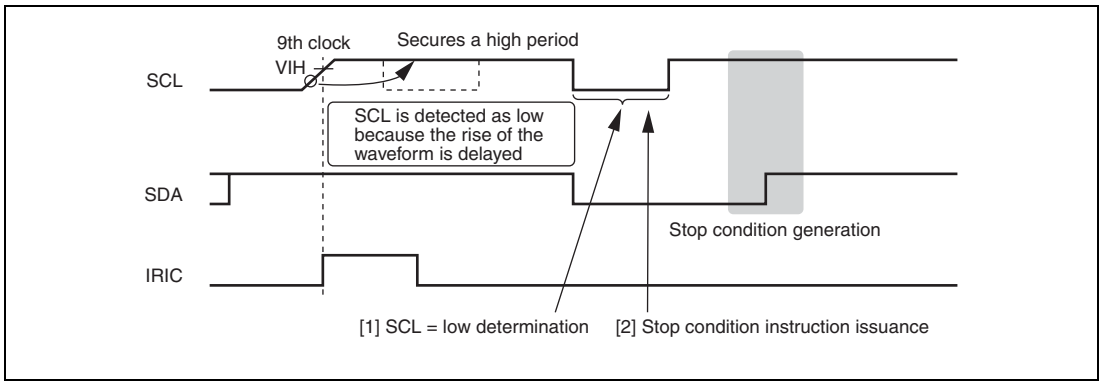


Figure 16.31 Stop Condition Issuance Timing

Note: This restriction on usage can be canceled by setting the FNC1 and FNC0 bits to B'11 in ICXR.

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

    If the rise time of SCL exceeds the stipulated value or a slave device in which a wait can be inserted by driving the SCL pin low is used when the wait function is used in I²C bust interface master mode, the IRIC flag should be cleared after determining that the SCL is low, as described below.

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

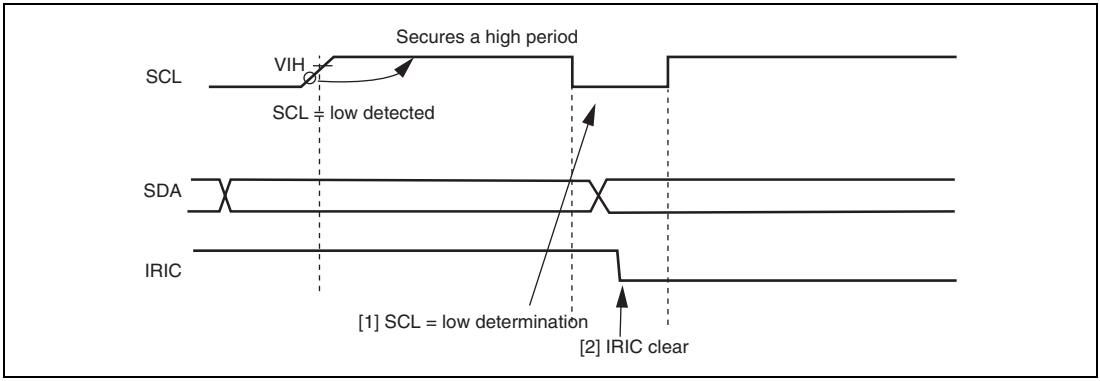


Figure 16.32 IRIC Flag Clearing Timing when WAIT = 1

Note: This restriction on usage can be canceled by setting the FNC1 and FNC0 bits to B'11 in ICXR.

11. Note on ICDR read and ICCR access in slave transmit mode

In I²C bus interface slave transmit mode, do not read ICDR or do not read/write from/to ICCR during the time shaded in figure 16.33. However, such read and write operations cause no problem in interrupt handling processing that is generated in synchronization with the rising edge of the 9th clock pulse because the shaded time has passed before making the transition to interrupt handling.

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

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

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

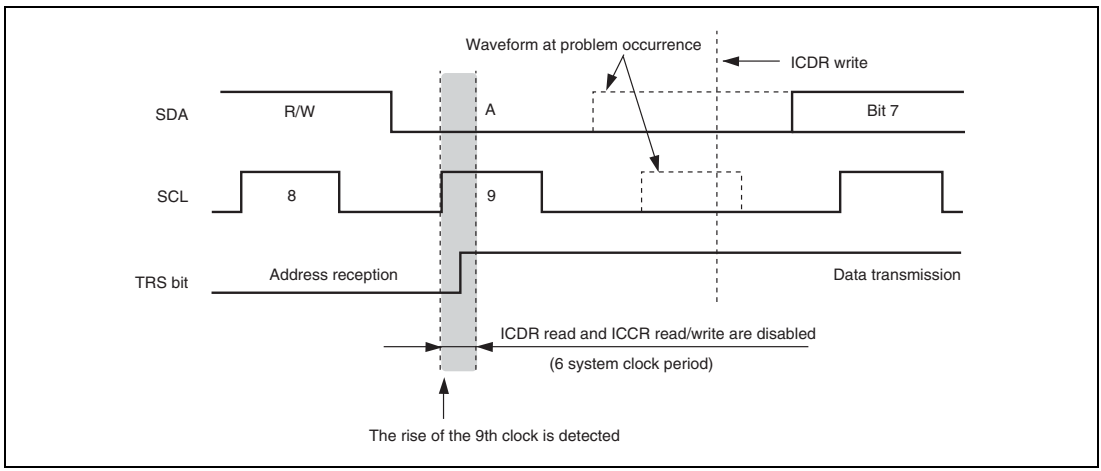


Figure 16.33 ICDR Read and ICCR Access Timing in Slave Transmit Mode

Note: This restriction on usage can be canceled by setting the FNC1 and FNC0 bits to B'11 in ICXR.

12. Note on TRS bit setting in slave mode

In I²C bus interface slave mode, if the TRS bit value in ICCR is set after detecting the rising edge of the 9th clock pulse or the stop condition before detecting the next rising edge on the SCL pin (the time indicated as (a) in figure 16.34), the bit value becomes valid immediately when it is set. However, if the TRS bit is set during the other time (the time indicated as (b) in figure 16.34), the bit value is suspended and remains invalid until the rising edge of the 9th clock pulse or the stop condition is detected. Therefore, when the address is received after the restart condition is input without the stop condition, the effective TRS bit value remains 1 (transmit mode) internally and thus the acknowledge bit is not transmitted after the address has been received at the 9th clock pulse.

To receive the address in slave mode, clear the TRS bit to 0 during the time indicated as (a) in figure 16.34. To release the SCL low level that is held by means of the wait function in slave mode, clear the TRS bit to and then dummy-read ICDR.

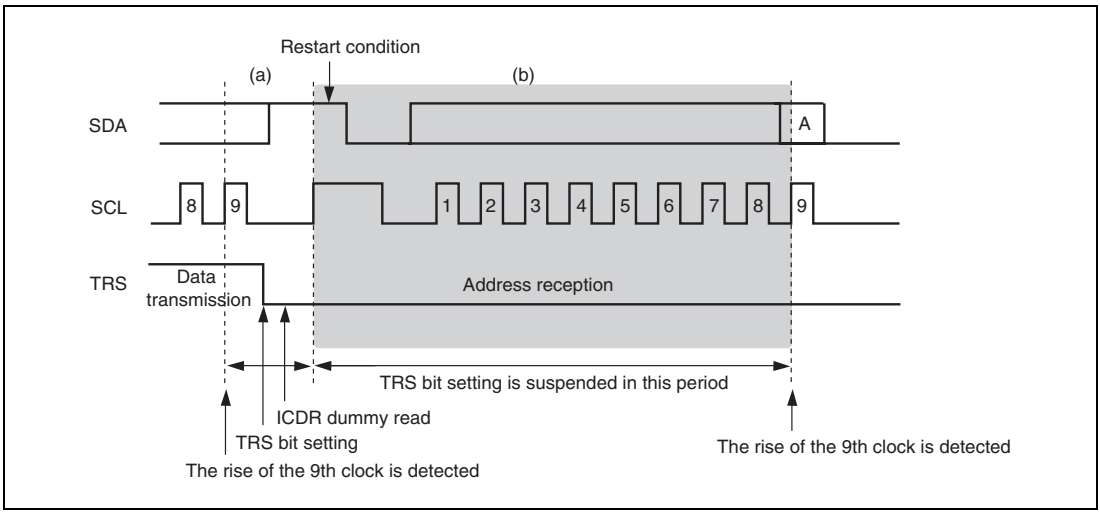


Figure 16.34 TRS Bit Set Timing in Slave Mode

Note: This restriction on usage can be canceled by setting the FNC1 and FNC0 bits to B'11 in ICXR.

13. 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.

14. 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 16.23, in order to switch from slave transmit mode to slave receive mode.

15. Note on Arbitration Lost in Master Mode

    The I²C bus interface recognizes the data in transmit/receive frame as an address when arbitration is lost in master mode and a transition to slave receive mode is automatically carried out.

    When arbitration is lost not in the first frame but in the second frame or subsequent frame, transmit/receive data that is not an address is compared with the value set in the SAR or SARX register as an address. If the receive data matches with the address in the SAR or SARX register, the I²C bus interface erroneously recognizes that the address call has occurred. (See figure 16.35.)

    In multi-master mode, a bus conflict could happen. When the I²C bus interface is operated in master mode, check the state of the AL bit in the ICSR register every time after one frame of data has been transmitted or received.

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

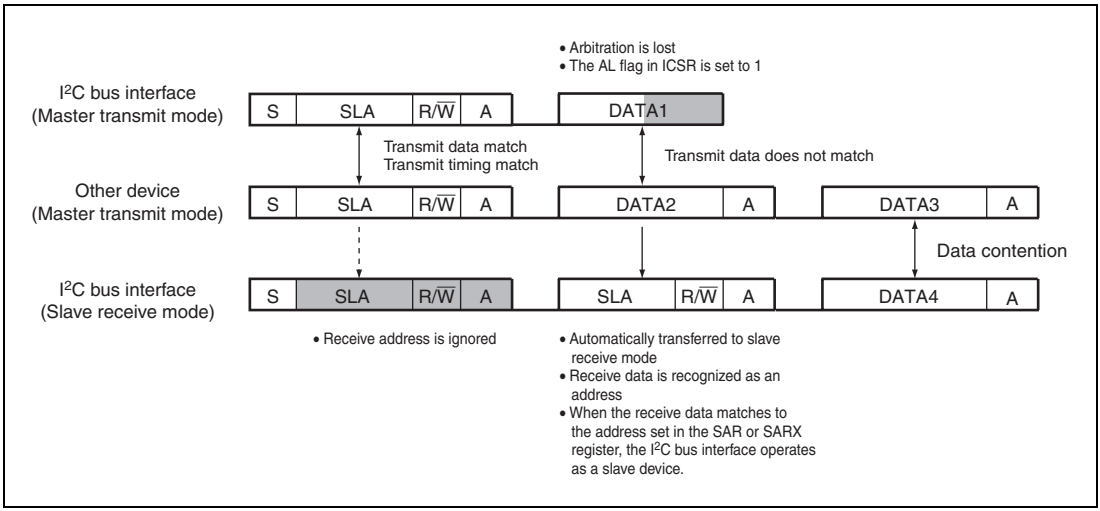


Figure 16.35 Diagram of Erroneous Operation when Arbitration is Lost

Though it is prohibited in the normal I²C protocol, the same problem may occur when the MST bit is erroneously set to 1 and a transition to master mode is occurred during data transmission or reception in slave mode. In multi-master mode, pay attention to the setting of the MST bit when a bus conflict may occur. In this case, the MST bit in the ICCR register should be set to 1 according to the order below.

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

Note: Above restriction can be cleared by setting bits FNC1 and FNC0 in the ICXR register.

16.6.1 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 24, Power-Down Modes.

Section 17 Keyboard Buffer Control Unit (KBU)

This LSI has three on-chip keyboard buffer control unit (KBU) channels. The KBU is provided with functions conforming to the PS/2 interface specifications.

Data transfer using the KBU employs a data line (KD) and a clock line (KCLK), providing economical use of connectors, board surface area, etc. Figure 17.1 shows a block diagram of the KBU.

17.1 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

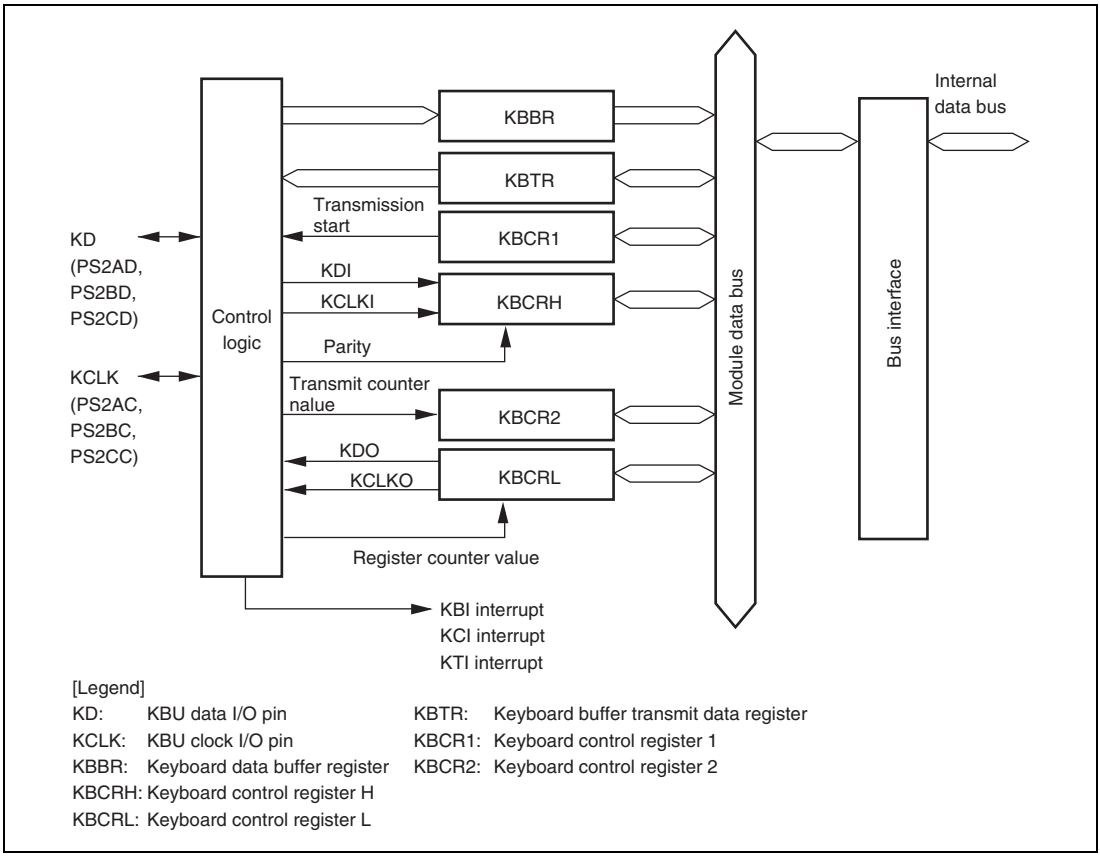


Figure 17.1 Block Diagram of KBU

Figure 17.2 shows how the KBU is connected.

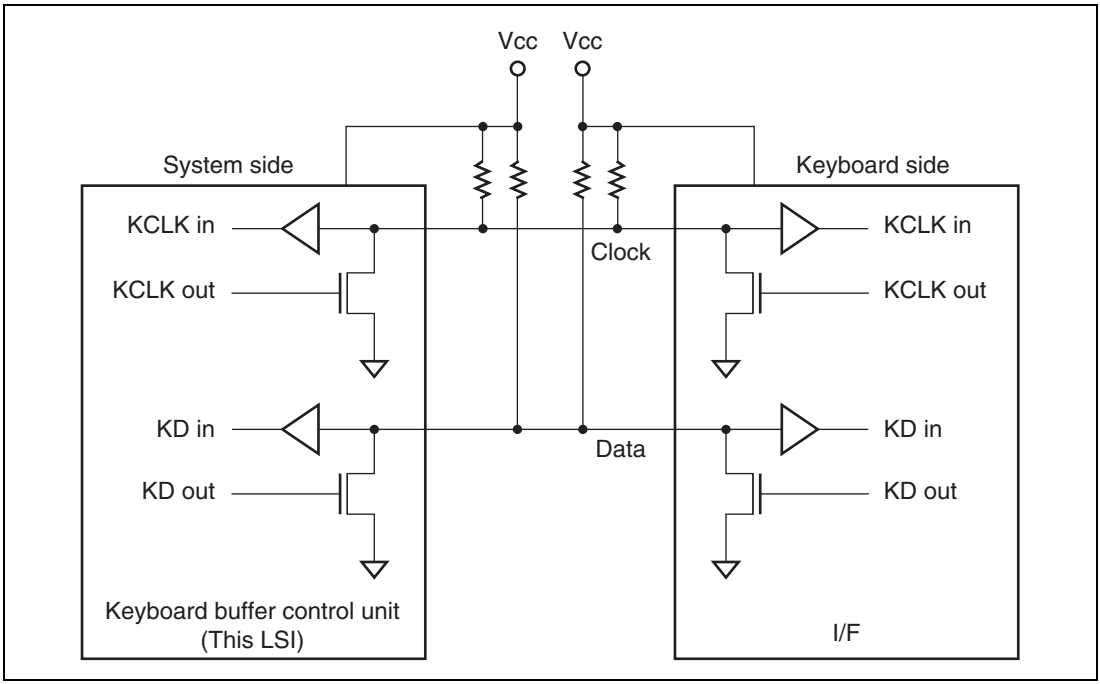


Figure 17.2 KBU Connection

17.2 Input/Output Pins

Table 17.1 lists the input/output pins used by the keyboard buffer control unit.

Table 17.1 Pin Configuration

Channel	Name	Abbreviation*	I/O	Function
0	KBU clock I/O pin (KCLK0)	PS2AC	I/O	KBU clock input/output
	KBU data I/O pin (KD0)	PS2AD	I/O	KBU data input/output
1	KBU clock I/O pin (KCLK1)	PS2BC	I/O	KBU clock input/output
	KBU data I/O pin (KD1)	PS2BD	I/O	KBU data input/output
2	KBU clock I/O pin (KCLK2)	PS2CC	I/O	KBU clock input/output
	KBU data I/O pin (KD2)	PS2CD	I/O	KBU 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.

17.3 Register Descriptions

The KBU has the following registers for each channel.

- Keyboard control register 1 (KBCR1)
- Keyboard control register 2 (KBCR2)
- Keyboard control register H (KBCRH)
- Keyboard control register L (KBCRL)
- Keyboard data buffer register (KBBR)
- Keyboard buffer transmit data register (KBTR)

17.3.1 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

Bit	Bit Name	Initial Value	R/W	Description
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.
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, watch mode, or subsleep 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.

17.3.2 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
2	TXCR2	0	R	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.
1	TXCR1	0	R	
0	TXCR0	0	R	
				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

17.3.3 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

Bit	Bit Name	Initial Value	R/W	Description
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
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.

17.3.4 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 KBU clock I/O pin output. 0: KBU clock I/O pin is low 1: KBU clock I/O pin is high
5	KDO	1	R/W	Keyboard Data Out Controls KBU data I/O pin output. 0: KBU data I/O pin is low 1: KBU 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- -: —

17.3.5 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 8-bit read only data. Initialized to H'00 by a reset, in hardware standby mode or when KBIOE is cleared to 0.
6	KB6	0	R	
5	KB5	0	R	
4	KB4	0	R	
3	KB3	0	R	
2	KB2	0	R	
1	KB1	0	R	
0	KB0	0	R	

17.3.6 Keyboard Buffer Transmit Data Register (KBTR)

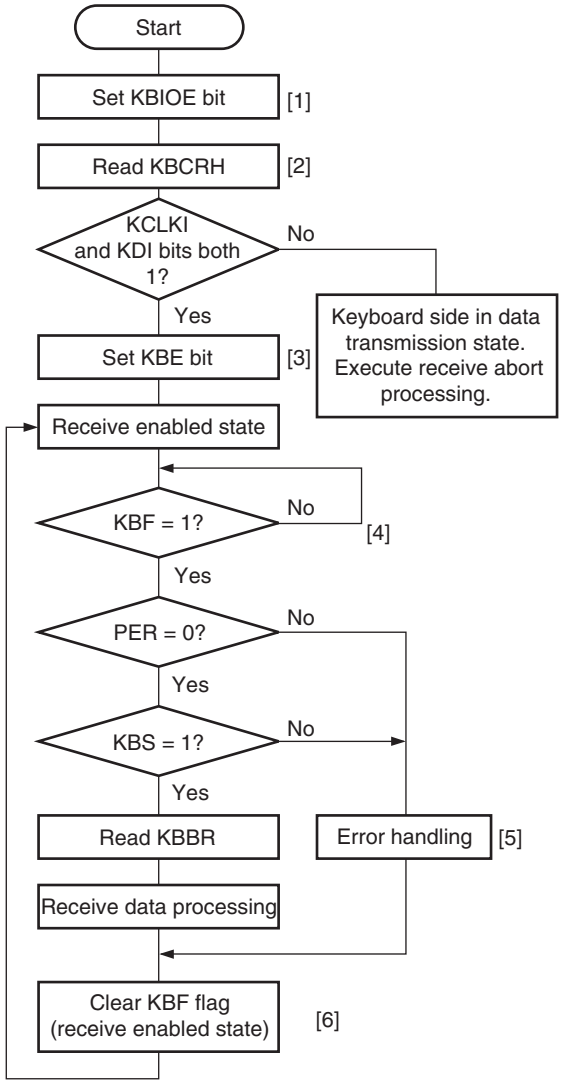
KBTR stores transmit data.

Bit	Bit Name	Initial Value	R/W	Description
7	KBT7	1	R/W	Keyboard Buffer Transmit Data Register 7 to 0 Initialized to H'00 at reset, in hardware standby mode.
6	KBT6	1	R/W	
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	

17.4 Operation

17.4.1 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 17.3, and the receive timing in figure 17.4.



[1] Set the KBIOE bit to 1 in KBCRL.

[2] Read KBCRH, and if the KCLKI and KDI bits are both 1, set the KBE bit (receive enabled state).

[3] Detect the start bit output on the keyboard side and receive data in synchronization with the fall of KCLK.

[4] When a stop bit is received, the keyboard buffer controller drives KCLK low to disable keyboard transmission (automatic I/O inhibit).
If the KBIE bit is set to 1 in KBCRH, an interrupt request is sent to the CPU at the same time.

[5] Perform receive data processing.

[6] Clear the KBF flag to 0 in KBCRL. At the same time, the system automatically drives KCLK high, setting the receive enabled state.

The receive operation can be continued by repeating steps [3] to [6].

Figure 17.3 Sample Receive Processing Flowchart

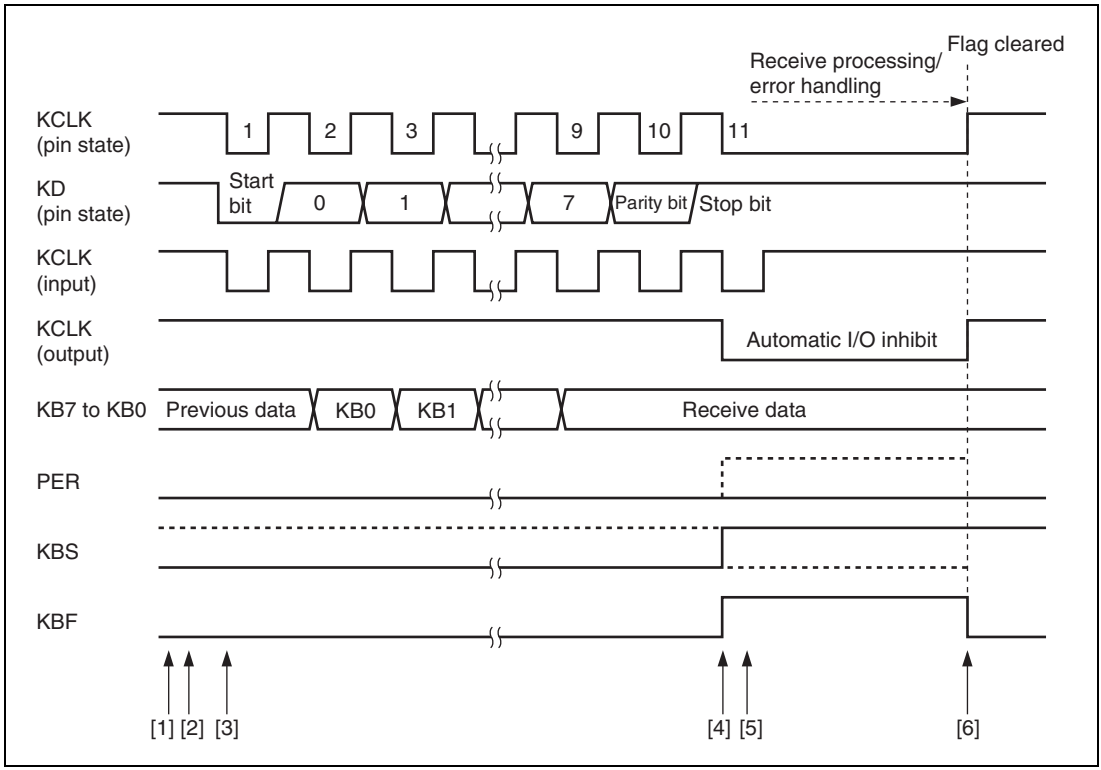
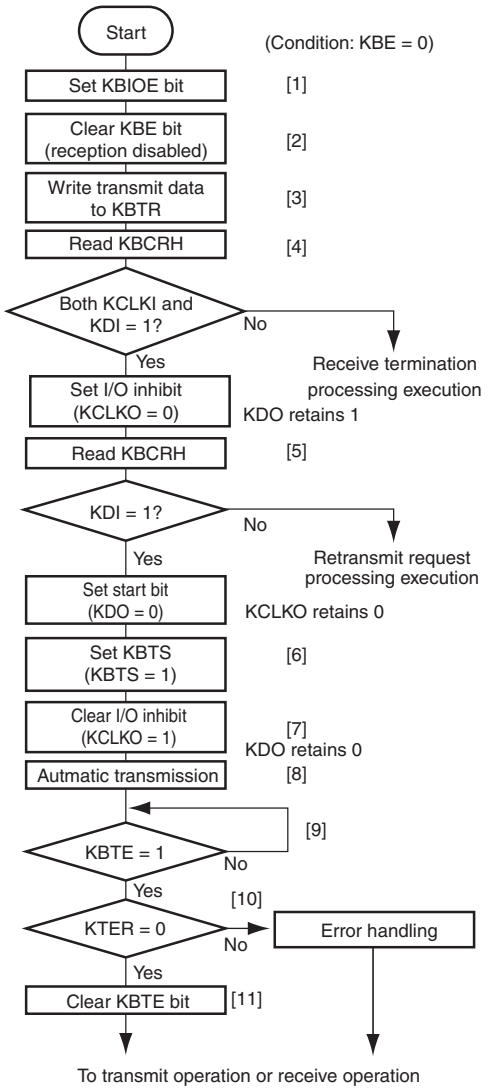


Figure 17.4 Receive Timing

17.4.2 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 17.5, and the transmit timing in figure 17.6.



[1] Write 1 to the KBIOE bit to enable transmission/ reception.

[2] Clear the KBE bit (reception disabled).

[3] Write transmit data to KBTR.

[4] Read KBCRH, and when both the KCLKI and KDI bits are 1, write 0 to the KCLKO bit to set the I/O inhibit. 60 μs or more is required for I/O inhibit.

[5] Read KBCRH, and when the KDI bit is 1, write 0 to the KDO (set start bit).

[6] Write 1 to the KBTS bit to enter the transmit enabled state.

[7] Write 1 to the KCLKO bit to clear the I/O inhibit.

[8] Check D0 to D7, the parity bit, the stop bit, and receive completion notification (send data at the falling edge of the KCLK signal).

[9] The KBTE bit is set to 1 at the eleventh rising edge of the KCLK signal. When KTIE = 1, a CPU interrupt occurs.

[10] When KTER = 0, transmission is successfully completed.

[11] Clear the KBTE bit to 0.

Note: * The start bit (KDO = 0) is automatically initialized (KDO = 1) when automatic transmission is started. After initialization, to write 0 to KDO, read 1 before writing 0 to it.

Figure 17.5 Sample Transmit Processing Flowchart

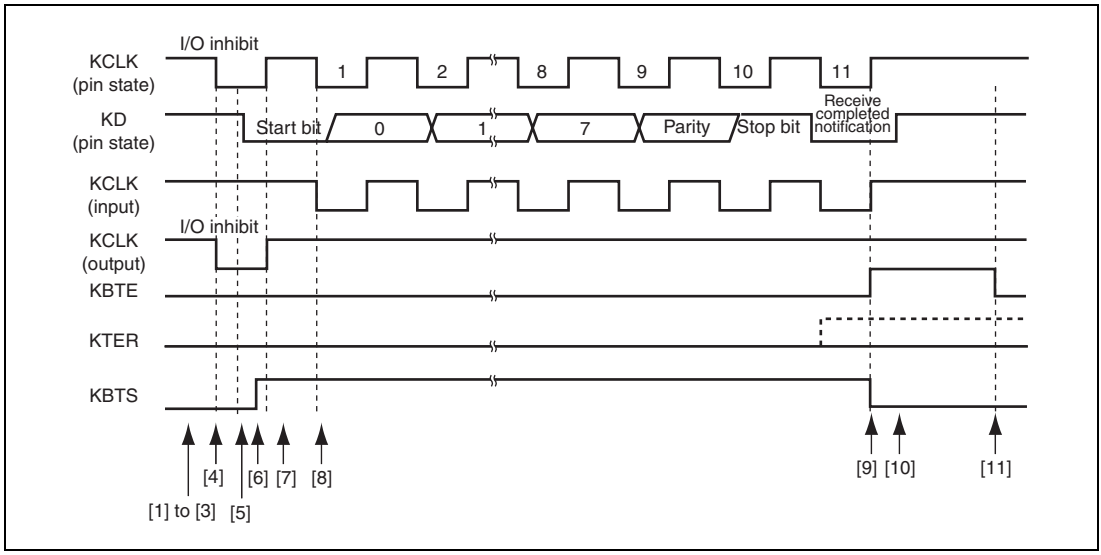


Figure 17.6 Transmit Timing

17.4.3 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 17.7, and the receive abort timing in figure 17.8.

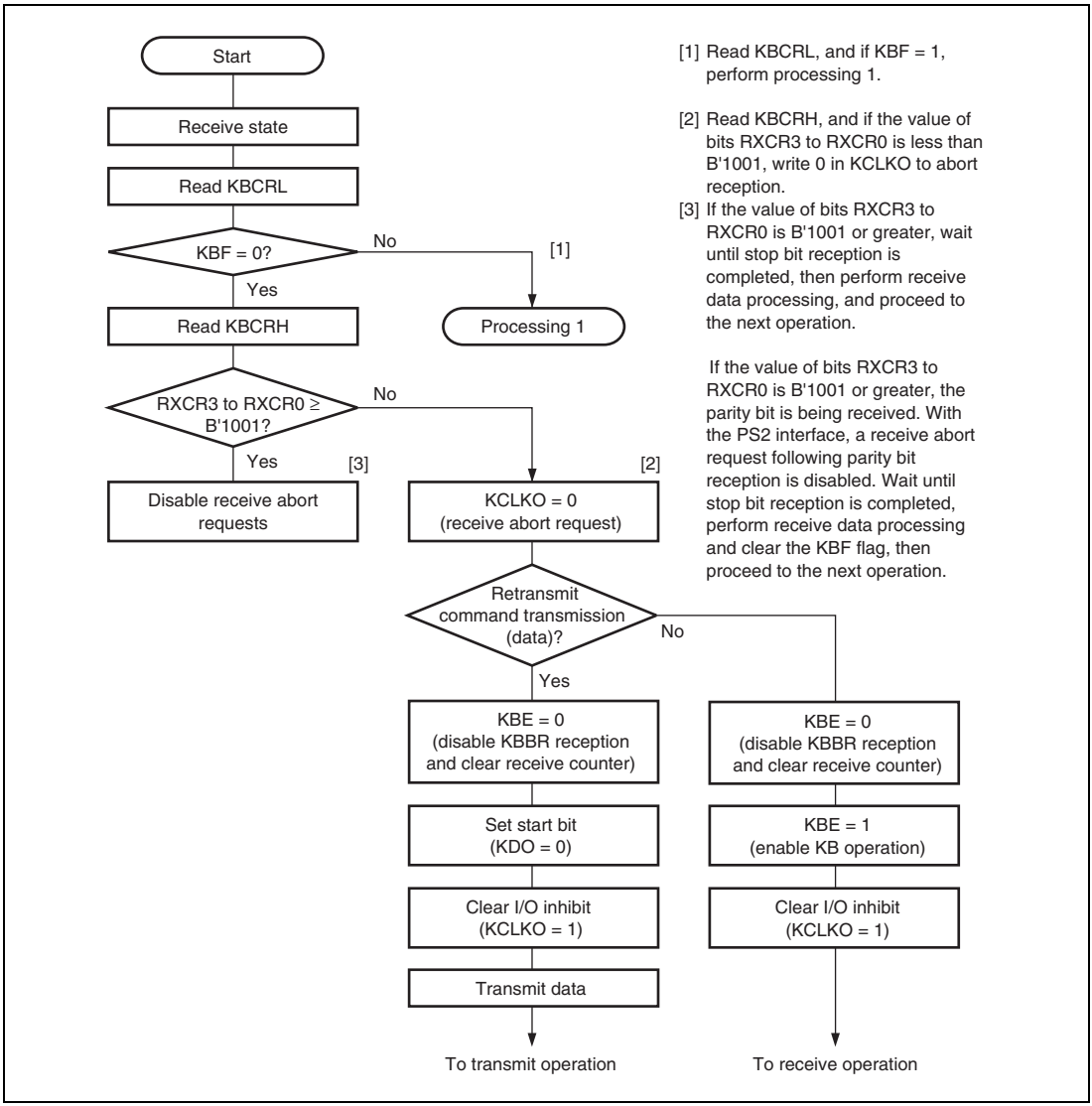


Figure 17.7 (1) Sample Receive Abort Processing Flowchart

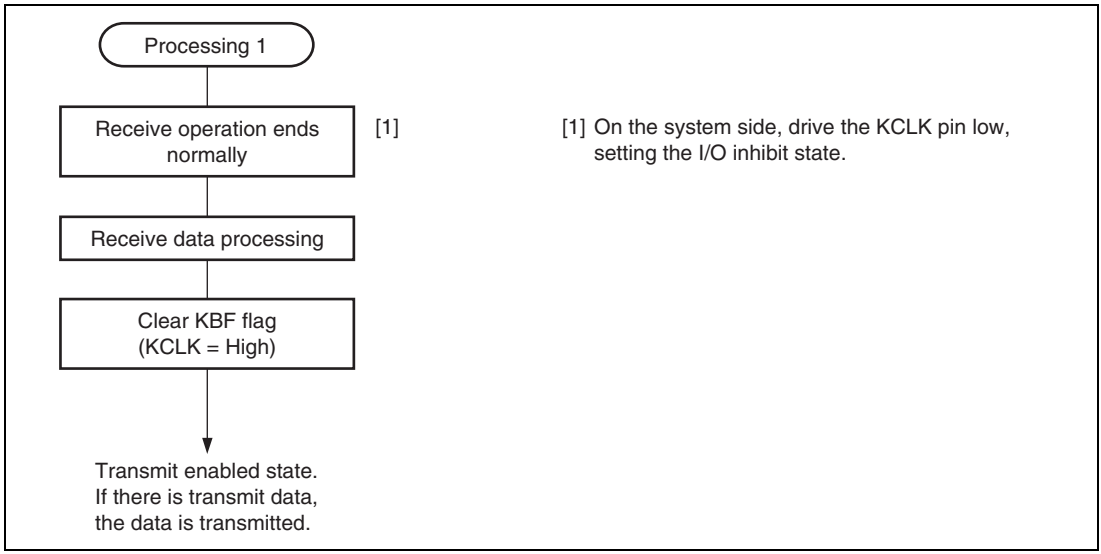


Figure 17.7 (2) Sample Receive Abort Processing Flowchart

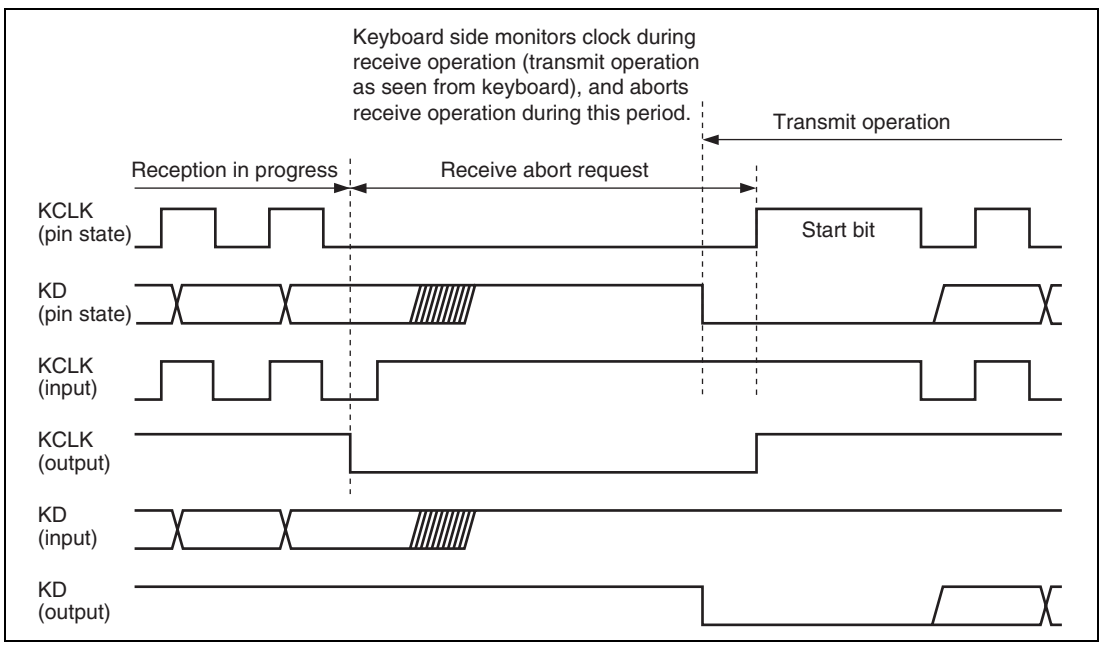
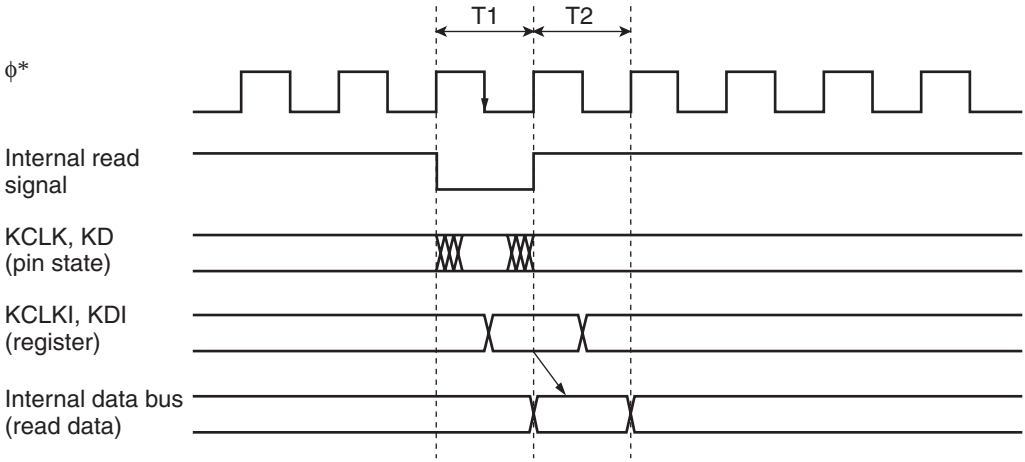


Figure 17.8 Receive Abort and Transmit Start (Transmission/Reception Switchover) Timing

17.4.4 KCLKI and KDI Read Timing

Figure 17.9 shows the KCLKI and KDI read timing.

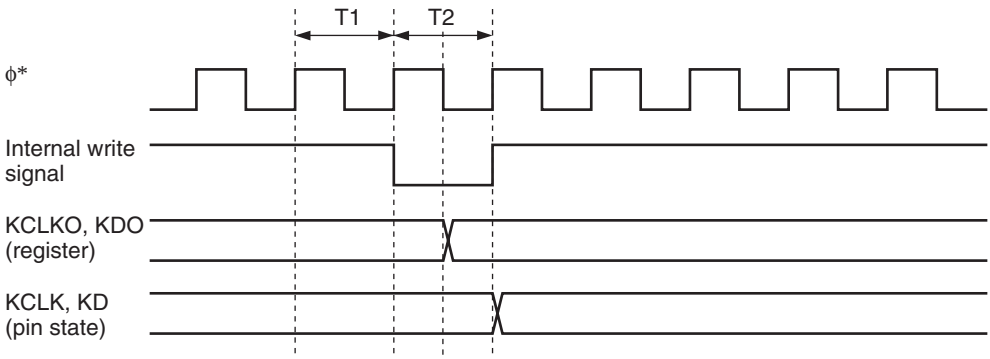


Note:* The φ clock shown here is scaled by 1/N in medium-speed mode when the operating mode is active mode.

Figure 17.9 KCLKI and KDI Read Timing

17.4.5 KCLKO and KDO Write Timing

Figure 17.10 shows the KLCKO and KDO write timing and the KCLK and KD pin states.



Note:* The ϕ clock shown here is scaled by 1/N in medium-speed mode.

Figure 17.10 KCLKO and KDO Write Timing

17.4.6 KBF Setting Timing and KCLK Control

Figure 17.11 shows the KBF setting timing and the KCLK pin states.

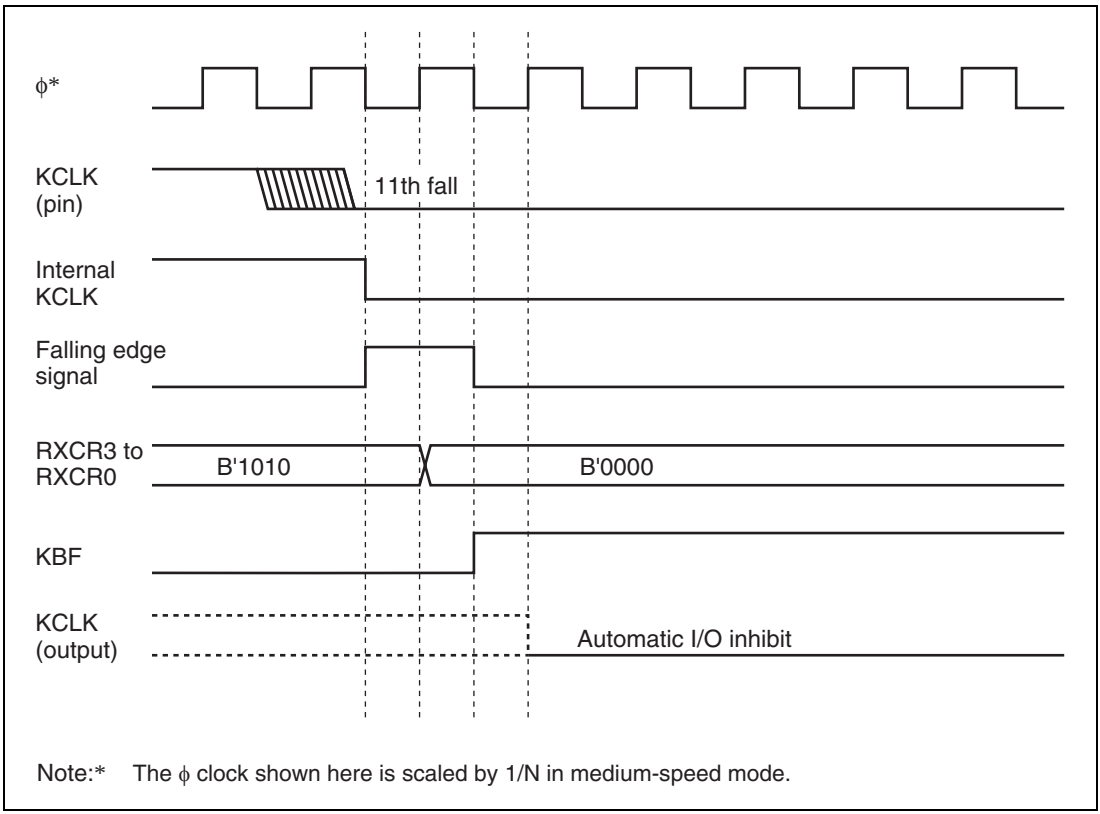
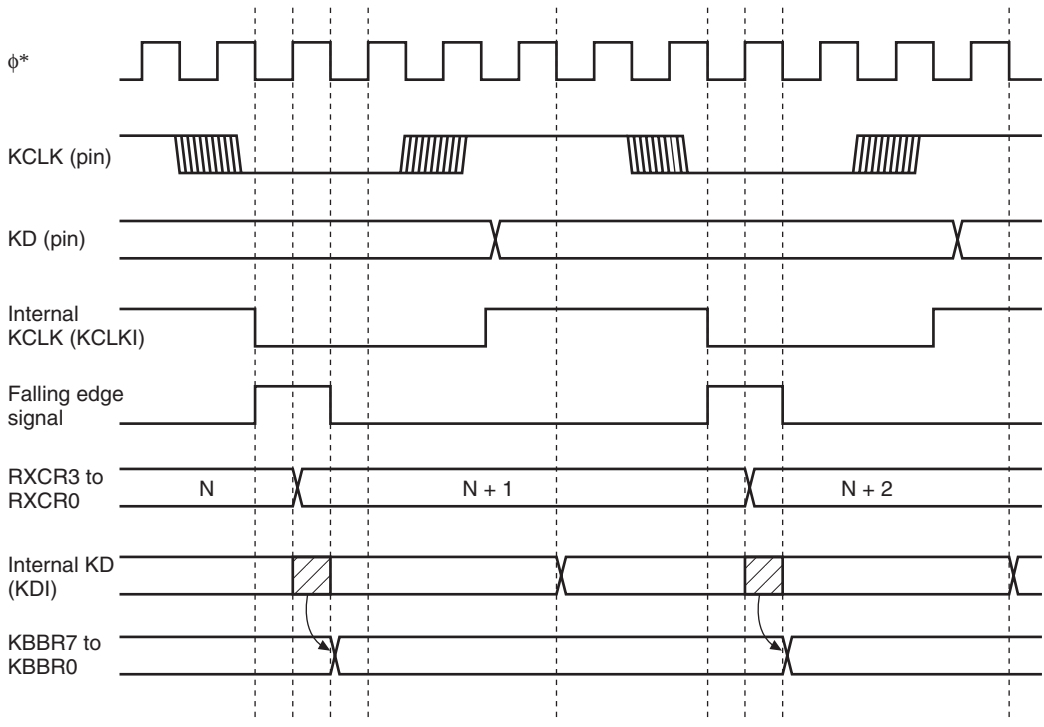


Figure 17.11 KBF Setting and KCLK Automatic I/O Inhibit Generation Timing

17.4.7 Receive Timing

Figure 17.12 shows the receive timing.



Note:* The ϕ clock shown here is scaled by 1/N in medium-speed mode.

Figure 17.12 Receive Counter and KBBR Data Load Timing

17.4.8 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 17.13 shows receive timing and the KCLK.

Note: * Period from the first falling edge of KCLK to completion of reception (KBF = 1).

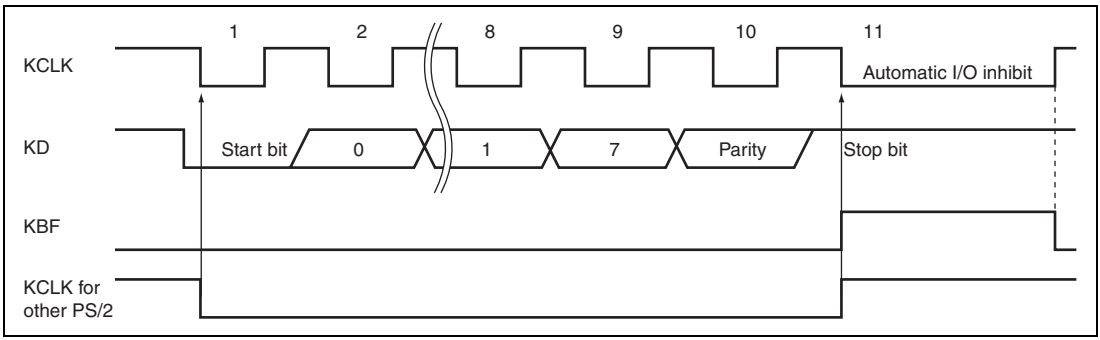
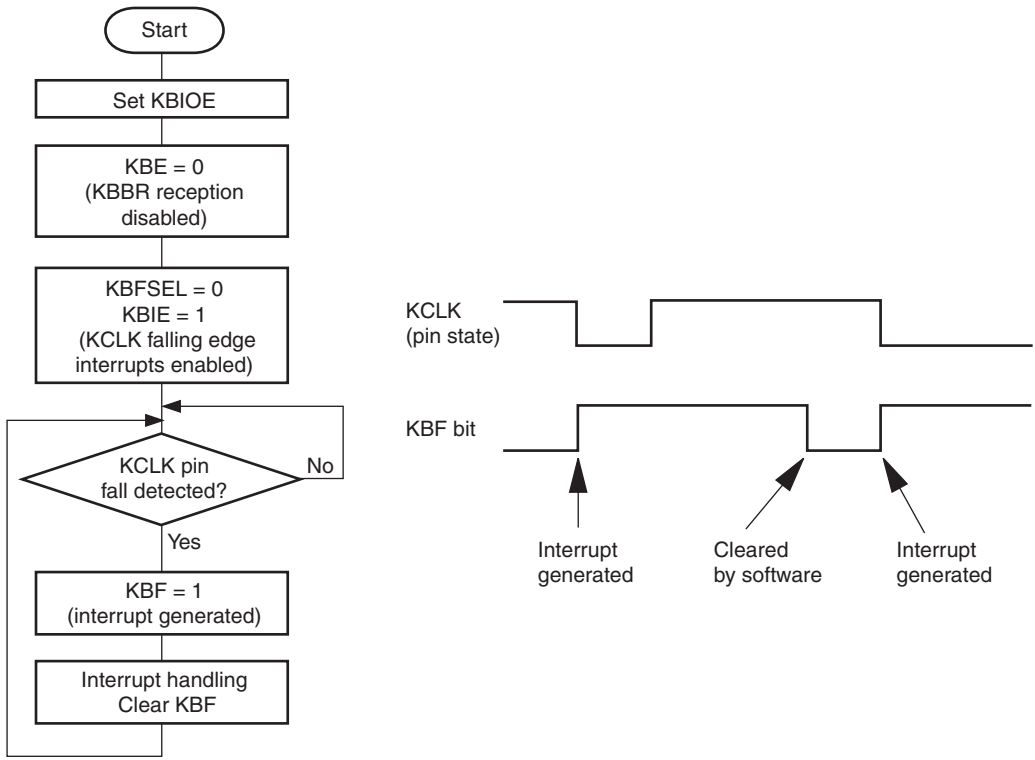


Figure 17.13 Receive Timing and KCLK

17.4.9 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 17.14 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 17.14 Example of KCLK Input Fall Interrupt Operation

17.4.10 First KCLK Falling Interrupt

An interrupt can be generated by detecting the first falling edge of KCLK on reception and transmission. Software standby, watch, and subsleep modes 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.)

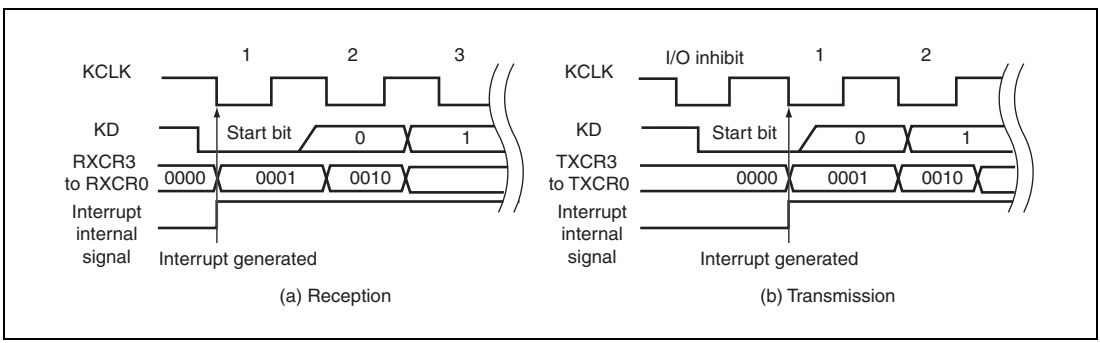


Figure 17.15 Timing of First KCLK Interrupt

- Canceling software standby mode, watch mode, and subsleep mode

  Software standby, watch, and subsleep modes are cancelled by a first KCLK falling interrupt. In this case, an interrupt is generated at the first KCLK since software standby mode, watch mode, or subsleep mode has been shifted (figure 17.17).

  Notes on canceling operation are explained below.

  — When a transition to software standby mode, watch mode, or subsleep mode is performed while both KBIOE and KCIE are set to 1, canceling the current mode is enabled by an first KCLK falling interrupt (the KBE and KBTS are not affected).

  — When software standby mode, watch mode, and subsleep 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, watch mode, and subsleep mode have been cancelled.

  — When software standby mode, watch mode, or subsllep 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, watch mode, or subsleep mode and canceling the mode by a first KCLK falling interrupt are performed 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 17.16 should be applied on interrupt generation.

  — Priority of canceling software standby mode, watch mode, and subsleep mode are 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, watch mode, and subsleep mode. Figure 17.6 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, watch made, and subsleep mode

  — KCLK is input directly to the interrupt control block, not through the KBU, in software standby mode, watch mode, and subsleep 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 17.18 shows setting and clearing timing.

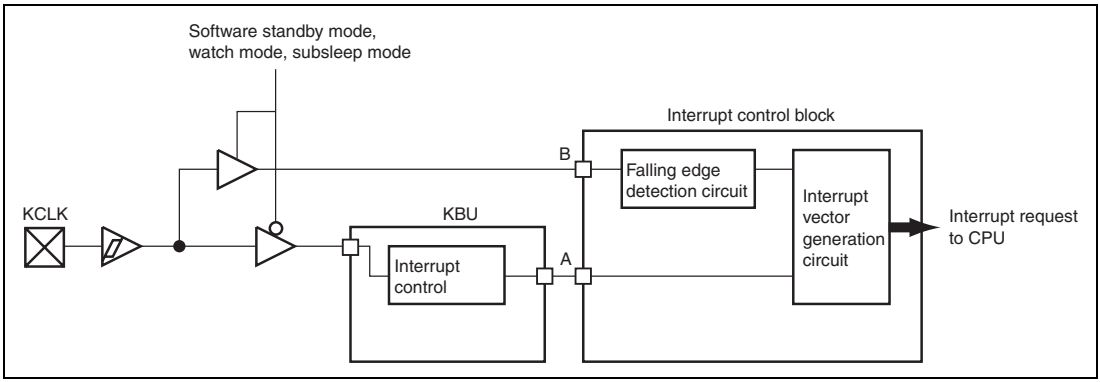


Figure 17.16 First KCLK Interrupt Path

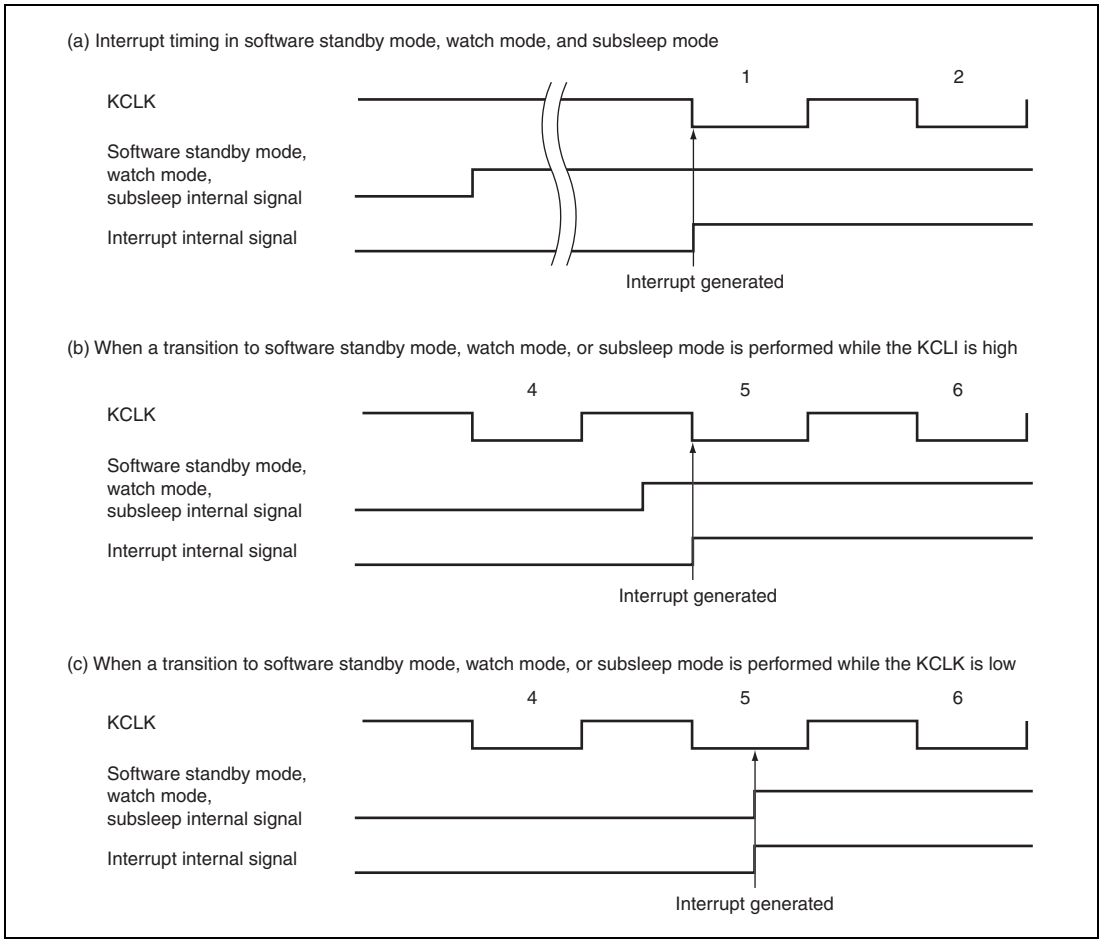


Figure 17.17 Interrupt Timing in Software Standby Mode, Watch Mode, and Subsleep Mode

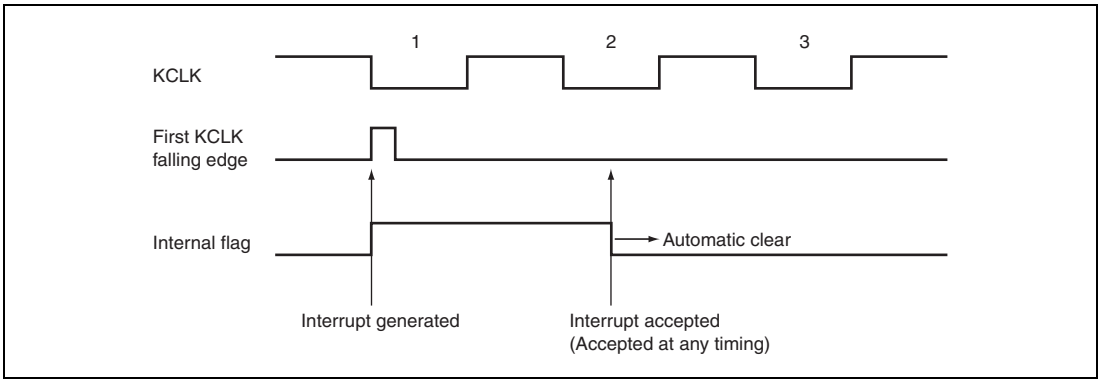


Figure 17.18 Internal Flag of First KCLK Falling Interrupt in Software Standby mode, Watch mode, and Subsleep mode

17.5 Usage Notes

17.5.1 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 17.19 shows the timing of KBIOE setting and KCLK falling edge detection.

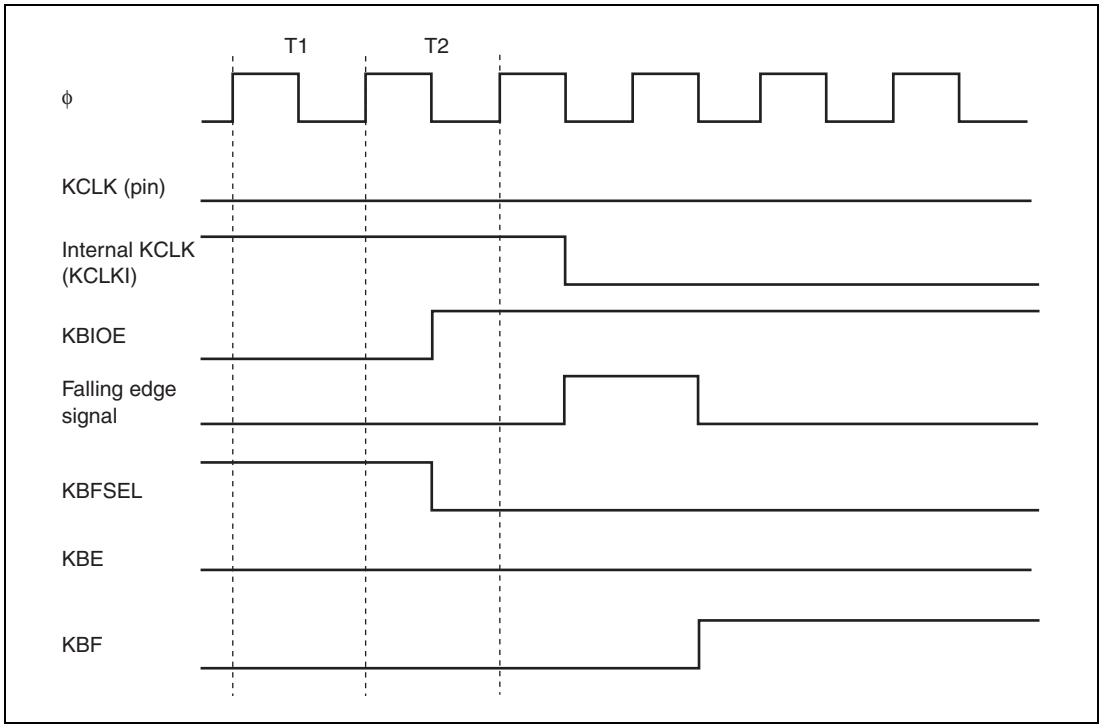


Figure 17.19 KBIOE Setting and KCLK Falling Edge Detection Timing

17.5.2 KD Output by KDO bit (KBCRL) and by Automatic Transmission

Figure 17.20 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.

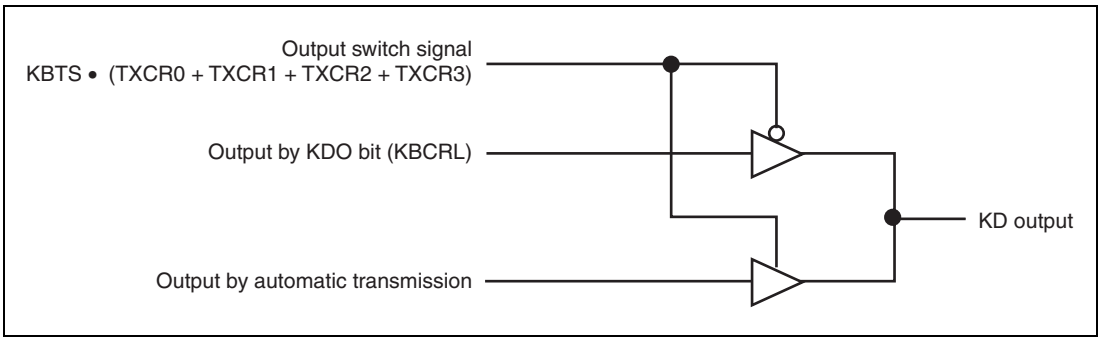


Figure 17.20 KDO Output

17.5.3 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 24, Power-Down Modes.

17.5.4 Medium Speed Mode

The KBU operates with a medium speed clock in medium speed mode. To operate the KBU normally, use at least 300-kHz medium speed clock.

17.5.5 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 18 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.

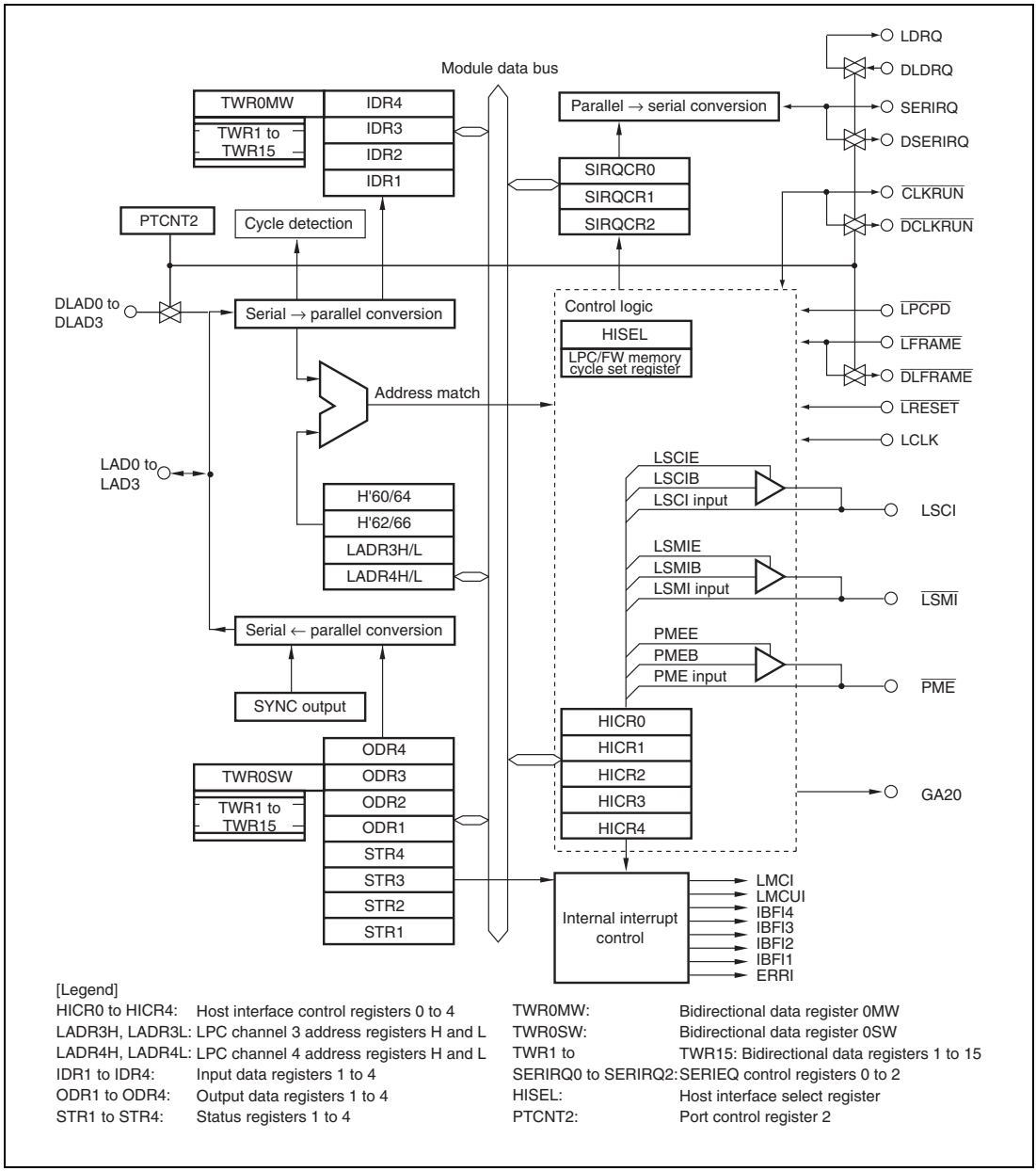
The LPC performs serial transfer of cycle type, address, and data, synchronized with the 33 MHz PCI clock. It uses four signal lines for address/data, and one for host interrupt requests. This LPC module supports I/O read, I/O write, LPC memory read, LPC memory write, firmware (FW) memory read, and firmware (FW) memory write cycle transfers. It is also provided with power-down functions that can control the PCI clock and shut down the LPC interface.

18.1 Features

- Supports LPC interface I/O read and I/O write cycles
  - Uses four signal lines (LAD3 to LAD0) to transfer the cycle type, address, and 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).
  - Fixed I/O addresses of H'60/H'64 are set for channel 1. A fast Gate A20 function is also provided.
  - Fixed I/O addresses of H'62/H'66 are set for channel 2.
  - I/O addresses from H'0000 to H'FFFF is selected for channel 3. Sixteen bidirectional data register bytes can be manipulated in addition to the basic register set.
  - I/O addresses from H'0000 to H'FFFF is selected for channel 4.
- 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.
  - 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.

- Supports LPC/FW memory cycles
  - Supports LPC memory read, LPC memory write, FW memory read, and FW memory write cycle transfer
  - FW memory read and FW memory write cycles can be transferred in bytes/words/longwords
  - LPC and FW memory cycles support the flash memory programming, flash memory erasing, and user commands
- Supports docking LPC
  - LAD3 toLAD0, $\overline{\text{LFRAME}}$, $\overline{\text{LRESET}}$, SERIRQ, $\overline{\text{CLKRUN}}$, and LDRQ can be connected to DLAD3 to DLAD0, $\overline{\text{DLFRAME}}$, $\overline{\text{DLRESET}}$, DSERIRQ, $\overline{\text{DCLKRUN}}$, and DLDRQ, respectively.
  - Resistance is 40 Ω (typ.).

Figure 18.1 shows a block diagram of the LPC.



[Legend]

HICR0 to HICR4:	Host interface control registers 0 to 4	TWR0MW:	Bidirectional data register 0MW
LADR3H, LADR3L:	LPC channel 3 address registers H and L	TWR0SW:	Bidirectional data register 0SW
LADR4H, LADR4L:	LPC channel 4 address registers H and L	TWR1 to	TWR15: Bidirectional data registers 1 to 15
IDR1 to IDR4:	Input data registers 1 to 4	SERIRQ0 to SERIRQ2:	SERIEQ control registers 0 to 2
ODR1 to ODR4:	Output data registers 1 to 4	HISEL:	Host interface select register
STR1 to STR4:	Status registers 1 to 4	PTCNT2:	Port control register 2

Figure 18.1 Block Diagram of LPC

18.2 Input/Output Pins

Table 18.1 lists the LPC pin configuration.

Table 18.1 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 (SMI, HIRQ1, HIRQ6, HIRQ9 to HIRQ12) 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
Docking LPC address/ data 3 to 0	DLAD3 to DLAD0	PB4 to PB7	I/O*[3]	Cycle type/address/data signals serially (4-signal-line) transferred in synchronization with LCLK
Docking LPC frame	$\overline{\text{DLFRAME}}$	PB3	I/O*[3]	Transfer cycle start and forced termination signal
Docking serialized interrupt request	DSERIRQ	P40	I/O*[3]	Serialized host interrupt request signal in synchronization with LCLK
Docking LPC clock run	$\overline{\text{DCLKRUN}}$	P41	I/O*[3]	LCLK restart request signal when serial host interrupt is requested

Name	Abbreviation	Port	I/O	Function
LPC Encoded DMA request	LDRQ	PC6	Output*[4]	DMA request signal
Docking LPC Encoded DMA request	DLDRQ	PC7	Input*[4]	DMA request 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.
3. This function becomes available by setting 1 to LPCS in PTCNT2 and one of LPC3E to LPC1E in HICR0 and LPC4E in HICR4. For details, see section 8.17.3, Port Control Register 2 (PTCNT2).
4. This function becomes available by setting 1 to LDRQS in PTCNT2 and one of LPC3E to LPC1E in HICR0 and LPC4E in HICR4. For details, see section 8.17.3, Port Control Register 2 (PTCNT2).

18.3 Register Descriptions

The LPC has the following registers.

- Host interface control registers 0 to 4 (HICR0 to HICR4)
- LPC channel 3 address registers H and L (LADR3H, LADR3L)
- LPC channel 4 address registers H and L (LADR4H, LADR4L)
- Input data registers 1 to 4 (IDR1 to IDR4)
- Output data registers 1 to 4 (ODR1 to ODR4)
- Bidirectional data registers 0 to 15 (TWR0 to TWR15)
- Status registers 1 to 4 (STR1 to STR4)
- SERIRQ control registers 0 to 2 (SIRQCR0 to SIRQCR2)
- Host interface select register (HISEL)

The following registers are needed to use LPC/FW memory cycles.

- RAM buffer address register (RBUFAR)
- Flash memory programming address registers H and L (FLWARH, FLWARL)
- Manufacture and device ID code registers (LMCMIDCR, LMCDIDCR)
- Erase block register (EBLKR)
- LMC status registers 1 and 2 (LMCST1, LMCST2)
- LMC control registers 1 and 2 (LMCCR1, LMCCR2)

- Host base address registers 1H and 1L (HBAR1H, HBAR1L)
- Host base address registers 2H and 2L (HBAR2H, HBAR2L)
- On-chip RAM host base address registers H and L (RAMBARH, RAMBARL)
- Address space set register (ASSR)
- On-chip RAM address space set register (RAMASSR)
- Slave address register 1 (SAR1)
- Slave address register 2 (SAR2)
- On-chip RAM slave address register (RAMAR)
- Flash memory write protect registers H, M, and L (FWPRH, FWPRM, FWPRL)
- Flash memory read protect registers H, M, and L (FRPRH, FRPRM, FRPRL)
- On-chip RAM protect control register (MPCR)
- User command data register (UCMDTR)

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.

18.3.1 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

			R/W		
Bit	**Bit Name**	**Initial Value**	**Slave**	**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 (H'0062, 66) matches for IDR2, ODR2, or STR2 1: LPC channel 2 operation is enabled • LPC1E 0: LPC channel 1 operation is disabled No address (H'0060, 64) 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 18.4.4, 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

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
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
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

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
4	LRSTB	0	R/W	—	LPC Software Reset Bit Resets the LPC interface. For the scope of initialization by an LPC reset, see section 18.4.4, 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
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 18.4.4, 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.

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
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. For details, refer to description on the LSCIE bit in HICR0.

18.3.2 Host Interface Control Registers 2 and 3 (HICR2 and HICR3)

HICR2 controls interrupts to an LPC interface slave (this LSI). HICR3 monitors the states of the LPC interface pins. Bits 6 to 0 in HICR2 are initialized to H'00 by a reset or in hardware standby mode. 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

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
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
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

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
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
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

18.3.3 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 Slave	R/W Host	Description
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 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.

18.3.4 LPC Channel 3 Address Registers H and L (LADR3H and LADR3L)

LADR3 stores 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	—	Store 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.

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
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 Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Transfer 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, and 4 are different.

18.3.5 LPC Channel 4 Address Registers H and L (LADR4H and LADR4L)

LADR4 stores 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	—	Store 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, and 3 are different.

18.3.6 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. For information on IDR3 and IDR4 selection, see the section of the corresponding LADR. 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 undefined.

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

Note: In bits 15 to 0, channel 1 corresponds to H'0060/H'0064, channel 2 corresponds to H'0062/H'0066.

18.3.7 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. For information on ODR3 and ODR4 selection, see the section of the corresponding LADR. 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 undefined.

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

Note: In bits 15 to 0, channel 1 and channel 2 corresponds to H'0060 and H'0062, respectively.

18.3.8 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 18.3.4, 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 undefined.

18.3.9 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. For information on STR3 and STR4 selection, see the section of the corresponding LADR. 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

Note: In bits 15 to 0, channel 1 and channel 2 corresponds to H'0064 and H'0066, respectively.

- STR1

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

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
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 18.4. 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.

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
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

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
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
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

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
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 The user can use these bits as necessary.
6	DBU36	0	R/W	R	
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

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
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

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
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.

18.3.10 SERIRQ Control Register 0 (SIRQCR0)

SIRQCR0 contains status bits that indicate the SERIRQ operating mode and bits that specify SERIRQ interrupt sources.

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

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
6	SELREQ	0	R/W	—	Start Frame Initiation Request Select Selects the condition of a start frame initiation request when a host interrupt request is cleared in quiet mode. 0: Start frame initiation is requested when all interrupt requests are cleared 1: Start frame initiation is requested when one or more interrupt requests are cleared
5	IEDIR2	0	R/W	—	Interrupt Enable Direct Mode Specifies whether LPC channel 2 and channel 3 SERIRQ interrupt source (SMI, IRQ6, IRQ9 to IRQ11) generation is conditional upon OBF, or is controlled only by the host interrupt enable bit. 0: Host interrupt is requested when host interrupt enable and corresponding OBF bits are both set to 1 1: Host interrupt is requested when host interrupt enable bit is set to 1
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

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
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
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

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
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
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

18.3.11 SERIRQ Control Register 1 (SIRQCR1)

SIRQCR1 contains status bits that indicate the SERIRQ operating mode and bits that specify SERIRQ interrupt sources.

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
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

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
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
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

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
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
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

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
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
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

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
0	IRQ6E2	0	R/W	—	Host IRQ6 Interrupt Enable 3 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

18.3.12 SERIRQ Control Register 2 (SIRQCR2)

SIRQCR2 contains bits that enable or disable SERIRQ interrupt requests and select the host interrupt request outputs.

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
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.

18.3.13 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 18.3.9, 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	—	SERIRQ Output Select
5	SELIRQ10	0	R/W	—	Select the pin output status of SERIRQ.
4	SELIRQ9	0	R/W	—	0: [When host interrupt request is cleared]
3	SELIRQ6	0	R/W	—	SERIRQ pin output is in the Hi-Z state
2	SELSMI	0	R/W	—	[When host interrupt request is set]
1	SELIRQ12	1	R/W	—	SERIRQ pin output is low
0	SELIRQ1	1	R/W	—	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.

18.3.14 RAM Buffer Address Register (RBUFAR)

RBUFAR stores the start address of 256-byte buffer in the on-chip RAM used for the flash memory programming in an LPC/FW memory cycle. The flash memory is programmed in units of 128 bytes. The 128-byte data is stored in the area whose lower address ranges from H'00 to H'7F. Bits 23 to 16 of the RAM buffer start address are fixed H'FF and bits 7 to 0 are fixed H'00. In the case of the initial value, the area to be used as a RAM buffer is from H'FFEF00 to H'FFEFFF (256 bytes). The contents of this register must not be changed in an LPC/FW memory cycle (the LMCE bit is set to 1).

			R/W		
Bit	**Bit Name**	**Initial Value**	**Slave**	**Host**	**Description**
7	RBA15	1	R/W	—	RAM Buffer Addresses 15 to 8
6	RBA14	1	R/W	—	Though H'D0 to H'EF can be set, setting only H'D1 to H'EF is valid. Setting other than these values is invalid. When setting is invalid, the previous value is retained.
5	RBA13	1	R/W	—	
4	RBA12	0	R/W	—	
3	RBA11	1	R/W	—	
2	RBA10	1	R/W	—	
1	RBA9	1	R/W	—	
0	RBA8	1	R/W	—	

Note: * The value must be selected so that the program area for flash memory programming/erasure is not overlapped.

18.3.15 Flash Memory Programming Address Registers H and L (FLWARH and FLARL)

FLWAR stores the start address of the flash memory programming in an LPC/FW memory cycle. Bits 19 to 7 of the start address set by the FLWAR set command are stored in this register. Bits 23 to 20 of the address are fixed H'0, and 6 to 0 are fixed H'00.

- FLWARH

			R/W		
Bit	**Bit Name**	**Initial Value**	**Slave**	**Host**	**Description**
7 to 5	—	All 0	R	—	Reserved These bits are read as 0 and cannot be modified.
4	FWA19	0	R	W*	Flash Memory Programming Start Address 19 to 15
3	FWA18	0	R	W*	
2	FWA17	0	R	W*	
1	FWA16	0	R	W*	
0	FWA15	0	R	W*	

Note: * Can be written to by the FLWAR set command.

- FLWARL

			R/W		
Bit	**Bit Name**	**Initial Value**	**Slave**	**Host**	**Description**
7	FWA14	1	R	W*	Flash Memory Programming Start Addresses 14 to 7
6	FWA13	1	R	W*	
5	FWA12	1	R	W*	
4	FWA11	0	R	W*	
3	FWA10	1	R	W*	
2	FWA9	1	R	W*	
1	FWA8	1	R	W*	
0	FWA7	1	R	W*	

Note: * Can be written to by the FLWAR set command.

18.3.16 Manufacture ID Code Register (LMCMIDCR) and Device ID Code Register (LMCDIDCR)

LMCMIDCR and LMCDIDCR store the manufacture ID code and device ID code, respectively. The contents of start address of LMCMIDCR and LMCDIDCR are output in response to the ID read command. For details of the ID read command, see descriptions of a list of LMC commands and command addresses. The contents of this register must not be changed in an LPC/FW memory cycle (the LMCE bit is set to 1).

- LMCMIDCR

			R/W		
Bit	**Bit Name**	**Initial Value**	**Slave**	**Host**	**Description**
7	Bit 7	0	R/W	R	Indicate the manufacture ID.
6	Bit 6	0	R/W	R	
5	Bit 5	0	R/W	R	
4	Bit 4	0	R/W	R	
3	Bit 3	0	R/W	R	
2	Bit 2	0	R/W	R	
1	Bit 1	0	R/W	R	
0	Bit 0	0	R/W	R	

- LMCDIDCR

			R/W		
Bit	**Bit Name**	**Initial Value**	**Slave**	**Host**	**Description**
7	Bit 7	0	R/W	R	Indicate the device ID.
6	Bit 6	0	R/W	R	
5	Bit 5	0	R/W	R	
4	Bit 4	0	R/W	R	
3	Bit 3	0	R/W	R	
2	Bit 2	0	R/W	R	
1	Bit 1	0	R/W	R	
0	Bit 0	0	R/W	R	

18.3.17 Erase Block Register (EBLKR)

EBLKR stores a block number set by the block erasure command.

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
7	Bit 7	0	R	W*	Store a block number ranging from 0 to 23. The block number is specified in BCD code (binary coded decimal code). EB0 H'00 EB1 H'01 : EB22 H'22 EB23 H'23 H'0A to H'0F, H'1A to H'1F, H'24 to H'FF must not be selected.
6	Bit 6	0	R	W*	
5	Bit 5	0	R	W*	
4	Bit 4	0	R	W*	
3	Bit 3	0	R	W*	
2	Bit 2	0	R	W*	
1	Bit 1	0	R	W*	
0	Bit 0	0	R	W*	

Note: * Can be written to by the block erasure command.

18.3.18 LMC Status Registers 1 and 2 (LMCST1 and LMCST2)

LMCST1 and LMCST2 indicate the processing status of the LMC. The contents of LMCST1 and LMCST2 are output in response to the status read command. For details of the status read command, see section 18.4.8, LPC/FW Memory Access Command.

- LMCST1

			R/W		
Bit	**Bit Name**	**Initial Value**	**Slave**	**Host**	**Description**
7	FLPI	0	R/(W)*[1]	R	Flash Memory Programming Interrupt/End Flag Setting this bit by the flash memory programming command generates an FLPI interrupt (LMCI). 0: Flash memory programming command wait Flash memory programming end [Clearing condition] When writing 0 after reading FLPI = 1 1: Flash memory programming is in progress [Setting condition] When receiving by the flash memory programming command (BUFTRAN = 1)
6	FLEI	0	R/(W)*[1]	R	Flash Memory Erasing Interrupt/End Flag Setting this bit by the flash memory erasing command generates an FLEI interrupt (LMCI). 0: Flash memory erasing command wait Flash memory erasing end [Clearing condition] When writing 0 after reading FLEI = 1 1: Flash memory erasing is in progress [Setting condition] When receiving by the flash memory erasing command (ERASEE = 1)

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
5	BUFINII	0	R/(W)*[1]	R	128-Byte Buffer Initialization Interrupt/End Flag Setting this bit by the buffer initialization command generates a BUFINII interrupt (LMCI). 0: Buffer initialization command wait Buffer initialization end [Clearing condition] When writing 0 after reading BUFINII = 1 1: Buffer initialization is in progress [Setting condition] When receiving by the buffer initialization command (BUFINIIE = 1 or BUFFINIE = 1 and HDININE = 0)
4	USERI	0	R/(W)*[1]	R	User Command Interrupt Flag Setting this bit by the user command generates an USERI interrupt. 0: User command wait User command processing end [Clearing condition] When writing 0 after reading USERI = 1 1: User command processing is in progress [Setting condition] When receiving by the user command
3	FLPERR	0	R/(W)*[2]	R	Flash Memory Programming Error 0: Flash memory has been completed programming [Clearing condition] Clearing by the clear status command 1: Flash memory programming error has been occurred.

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
2	FLEERR	0	R/(W)*[2]	R	Flash Memory Erasing Error 0: Flash memory has been completed erasure [Clearing condition] Clearing by the clear status command 1: Flash memory erasing error has been occurred
1, 0	—	All 0	R	R	Reserved These bits are read as 0 and cannot be modified.

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

- LMCST2

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
7	PROTECT	0	R	R	Indicates protect information in an LPC/FW memory cycle. [Updating conditions] • Data read command (flash memory/on-chip RAM) • FLWAR set command • Data write command (flash memory) (except for the case where an receive address does not match the FLWAR) • Data write command (on-chip RAM) 0: Not protected 1: Protected
6	LMCBUSY	0	R	R	Indicates whether the on-chip RAM or RAM buffer is being written. 0: Wait for write access has been completed or write access has already been completed 1: Write access is in progress

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
5	ERASEE	0	R	R	Enables/disables the block erasure command 0: Disables the block erasure command [Clearing conditions] • Clearing by the block erasure command • Clearing by the clear status command 1: Enables the block erasure command [Setting condition] • Setting by the erasing enable command
4	WRITEE	0	R	R	Enables/disables the on-chip RAM data write command 0: Disables the on-chip RAM data write command [Clearing condition] • Clearing by the WRITEE clear command 1: Enables the on-chip RAM data write command [Setting condition] • Setting by the write enable command
3	BUFTRAN	0	R	R	Indicates the transfer state of the 128-byte buffer. 0: 128-byte buffer transfer has been completed [Clearing conditions] • Clearing by the flash memory programming command • Clearing by the BUFTRAN clear command 1: 128-byte buffer transfer is in progress [Setting condition] • Setting by the FLWAR set command when BUFTRAN = 0
2 to 0	—	All 0	R	R	Reserved These bits are read as 0 and cannot be modified.

18.3.19 LMC Control Registers 1 and 2 (LMCCR1 and LMCCR2)

LMCCR1 enables/disables the LMC host interface function. LMCCR2 enables/disables interrupts requested from the host by the interrupt commands and selects wait-state type.

- LMCCR1

			R/W		
Bit	**Bit Name**	**Initial Value**	**Slave**	**Host**	**Description**
7	LMCE	0	R/W	—	LPC/FW Memory Cycle Enable Enables/disables the LPC/FW memory cycles (LPC memory cycle and FW memory cycle). 0: LPC/FW memory cycles are disabled 1: LPC/FW memory cycles are enabled
6	LPCME	0	R/W	—	LPC Memory Cycle Enable Enables/disables the LPC memory read/write interface function. When enabled, data transfer between the slave (this LSI) and host is performed via the LAD3 to LAD0, $\overline{\text{LFRAME}}$, $\overline{\text{LRESET}}$, and LCLK pins. 0: LPC memory cycles are disabled 1: LPC memory cycles are enabled
5	FWME	0	R/W	—	Firmware Memory Cycle Enable Enables/disables the FW memory read/write interface function. When enabled, data transfer between the slave (this LSI) and host is performed via the LAD3 to LAD0, $\overline{\text{LFRAME}}$, $\overline{\text{LRESET}}$, and LCLK pins. 0: FW memory cycles are disabled 1: FW memory cycles are enabled
4	—	0	R/W	—	Reserved This bit cannot be modified.

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
3	FLASHE	0	R/W	—	Flash Memory Programming/Erasing Enable Enables/disables to program/erase the flash memory by the LPC/FW memory cycle. Programming/erasing the flash memory is controlled in combination with the FLPIE and FLEIE bits by the flash memory programming/erasing command. 0: Disables to program/erase the flash memory 1: Enables to program/erase the flash memory
2	HDINIE	0	R/W	—	RAM Buffer Initialization Enable Enables/disables the function which automatically initializes the 128-byte transfer buffer contents to H'FF. This bit is valid when the BUFINIIE bit is cleared to 0. 0: Disables to initialize RAM buffer automatically 1: Enables to initialize RAM buffer automatically
1, 0	—	All 0	R/W	—	Reserved These bits are read as 0 and cannot be modified.

- LMCCR2

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
7	FLPIE*	0	R/W	—	Flash Memory Programming Interrupt Enable (LMCI) 0: Disables the flash memory programming command receive complete interrupt 1: Enables the flash memory programming command receive complete interrupt
6	FLEIE*	0	R/W	—	Flash Memory Erasing Interrupt Enable (LMCI) 0: Disables the flash memory erasing command receive complete interrupt 1: Enables the flash memory erasing command receive complete interrupt

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
5	BUFINIIE	0	R/W	—	RAM Buffer Initialization Interrupt Enable (LMCI) 0: Disables the RAM buffer initialization command receive complete interrupt 1: Enables the RAM buffer initialization command receive complete interrupt
4	USERIE	0	R/W	—	User Command Interrupt Enable (LMCUI) 0: Disables the user command receive complete interrupt 1: Enables the user command receive complete interrupt
3	WAITSEL	0	R/W	—	Wait Select Bit Selects the wait-state type in an LPC/FW memory cycle. 0: Short wait (4b'0101) 1: Long wait (4b'0110)
2 to 0	—	All 0	R/W	—	Reserved These bits are read as 0 and cannot be modified.

Note: * The FLPIE and FLEIE bits are valid when the FLASHE bit is set to 1.

18.3.20 Host Base Address Registers 1H and 1L (HBAR1H and HBAR1L)

HBAR1 stores the upper 16 bits of a host start address when a host address is translated into a flash memory address. The inverted signal level of pin $\overline{\text{LID3}}$ is reflected in the MSB of HBAR1. The lower 16 bits of the host start address are fixed H'0000. The host address space to be translated is decided in combination with bits AS13 to AS10 in ASSR which select the size of the host address space. When the FW memory cycle is used, bits 7 to 4 in HBAR1H (HB1A31 to HB1A28) are used as IDSEL. The contents of this register must not be changed in LPC/FW memory cycles (while LMCE is set to 1).

- HBAR1H

			R/W		
Bit	**Bit Name**	**Initial Value**	**Slave**	**Host**	**Description**
7	HB1A31	—	R	—	Host Base Address Bits 31 to 24
6	HB1A30	0	R/W	—	Store the host base address 31 to 24. Bit HB1A31 reflects the inverted signal level of pin $\overline{\text{LID3}}$.
5	HB1A29	0	R/W	—	
4	HB1A28	0	R/W	—	
3	HB1A27	0	R/W	—	
2	HB1A26	0	R/W	—	
1	HB1A25	0	R/W	—	
0	HB1A24	0	R/W	—	

- HBAR1L

			R/W		
Bit	**Bit Name**	**Initial Value**	**Slave**	**Host**	**Description**
7	HB1A23	0	R/W	—	Host Base Address Bits 23 to 16
6	HB1A22	0	R/W	—	Store the host base address 23 to 16.
5	HB1A22	0	R/W	—	
4	HB1A20	0	R/W	—	
3	HB1A19	0	R/W	—	
2	HB1A18	0	R/W	—	
1	HB1A17	0	R/W	—	
0	HB1A16	0	R/W	—	

18.3.21 Host Base Address Registers 2H and 2L (HBAR2H and HBAR2L)

HBAR2 stores the upper 16 bits of a host start address when a host address is translated into a flash memory address. The lower 16 bits of the host start address are fixed H'0000. The host address space to be translated is decided in combination with bits AS23 to AS20 in ASSR which select the size of the host address space. When the FW memory cycle is used, bits 7 to 4 in HBAR2H (HB2A31 to HB2A28) are used as IDSEL. The contents of this register must not be changed in LPC/FW memory cycles (while LMCE is set to 1).

- HBAR2H

			R/W		
Bit	Bit Name	Initial Value	Slave	Host	Description
7	HB2A31	0	R/W	—	Host Base Address Bits 31 to 24
6	HB2A30	0	R/W	—	Store the host base address 31 to 24.
5	HB2A29	0	R/W	—	
4	HB2A28	0	R/W	—	
3	HB2A27	0	R/W	—	
2	HB2A26	0	R/W	—	
1	HB2A25	0	R/W	—	
0	HB2A24	0	R/W	—	

- HBAR2L

			R/W		
Bit	Bit Name	Initial Value	Slave	Host	Description
7	HB2A23	0	R/W	—	Host Base Address Bits 23 to 16
6	HB2A22	0	R/W	—	Store the host base address 23 to 16.
5	HB2A22	0	R/W	—	
4	HB2A20	0	R/W	—	
3	HB2A19	0	R/W	—	
2	HB2A18	0	R/W	—	
1	HB2A17	0	R/W	—	
0	HB2A16	0	R/W	—	

18.3.22 On-Chip RAM Host Base Address Registers H and L (RAMBARH and RAMBARL)

RAMBAR stores the upper 16 bits of the host start address when a host address is translated into an on-chip RAM address. The lower 16 bits of the host start address are fixed H'0000. The host address space to be translated is decided in combination with the RAMASSR contents which select the size of the host address space. When the FW memory cycle is used, bits 7 to 4 in RABAHR (MRA31 to MRA28) are used as IDSEL. The contents of this register must not be changed in LPC/FW memory cycles (while LMCE is set to 1).

The host address space of which lower 16 bits are H'FFF0 to H'FFFF is used as command space.

- RAMBARH

			R/W		
Bit	**Bit Name**	**Initial Value**	**Slave**	**Host**	**Description**
7	MRA31	0	R/W	—	On-Chip RAM Host Base Address Bits 31 to 24
6	MRA30	0	R/W	—	Store the host base address 31 to 24.
5	MRA29	0	R/W	—	
4	MRA28	0	R/W	—	
3	MRA27	0	R/W	—	
2	MRA26	0	R/W	—	
1	MRA25	0	R/W	—	
0	MRA24	0	R/W	—	

- RAMBARL

			R/W		
Bit	**Bit Name**	**Initial Value**	**Slave**	**Host**	**Description**
7	MRA23	0	R/W	—	On-Chip RAM Host Base Address Bits 23 to 16
6	MRA22	0	R/W	—	Store the host base address 23 to 16.
5	MRA22	0	R/W	—	
4	MRA20	0	R/W	—	
3	MRA19	0	R/W	—	
2	MRA18	0	R/W	—	
1	MRA17	0	R/W	—	
0	RA16	0	R/W	—	

18.3.23 Address Space Set Register (ASSR)

ASSR selects the flash memory address space to be used by the host and slave. The contents of this register must not be changed in LPC/FW memory cycles (while LMCE is set to 1).

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
7	AS13	0	R/W	—	Select flash memory address space 1.
6	AS12	0	R/W	—	AS13 AS12 AS11 AS10
5	AS11	0	R/W	—	0 0 0 0 : 64 kbytes
4	AS10	0	R/W	—	0 0 0 1 : 128 kbytes
					0 0 1 0 : 256 kbytes
					0 0 1 1 : 384 kbytes
					0 1 0 0 : 512 kbytes
					0 1 0 1 : 640 kbytes
					0 1 1 0 : 768 kbytes
					0 1 1 1 : 1 Mbyte
					B'1000 to B'1111 must not be selected.
3	AS23	0	R/W	—	Select flash memory address space 2.
2	AS22	0	R/W	—	AS23 AS22 AS21 AS20
1	AS21	0	R/W	—	0 0 0 0 : 64 kbytes
0	AS20	0	R/W	—	0 0 0 1 : 128 kbytes
					0 0 1 0 : 256 kbytes
					0 0 1 1 : 384 kbytes
					0 1 0 0 : 512 kbytes
					0 1 0 1 : 640 kbytes
					0 1 1 0 : 768 kbytes
					0 1 1 1 : 1 Mbyte
					B'1000 to B'1111 must not be selected.

18.3.24 On-Chip RAM Address Space Set Register (RAMASSR)

RAMASSR selects the on-chip RAM address space to be used by the host and slave. The bits 7 to 5 do not affect operations. The contents of this register must not be changed in LPC/FW memory cycles (while LMCE is set to 1).

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
7	—	0	R/W	—	On-Chip RAM Address Space Selection
6	—	0	R/W	—	Select the on-chip RAM address space to be used by the host.
5	—	0	R/W	—	
4	RAMAS4	0	R/W	—	
3	RAMAS3	0	R/W	—	
2	RAMAS2	0	R/W	—	
1	RAMAS1	0	R/W	—	
0	RAMAS0	0	R/W	—	

RAM AS4	RAM AS3	RAM AS2	RAM AS1	RAM AS0		
0	0	0	0	0	:	Setting prohibited
0	0	0	0	1	:	256 bytes
0	0	0	1	0	:	512 bytes
0	0	0	1	1	:	768 bytes
0	0	1	0	0	:	1 kbyte
		:				
1	1	1	1	1	:	8 kbytes – 256 bytes

RAM address space = 256 bytes × RAMAS

18.3.25 Slave Address Register 1 (SAR1)

SAR1 selects the slave start address of the flash memory obtained by translating the host address in HBAR1. The bits 23 to 16 are selected by this register. The lower 16 bits are fixed H'0000. The contents of this register must not be changed in LPC/FW memory cycles (while LMCE is set to 1).

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
7	SA1R23	0	R/W	—	Slave Address Bits 23 to 16
6	SA1R22	0	R/W	—	Select bits 23 to 16 of the flash memory address obtained by translating the host address. The value H'10 to H'FF should not be selected.
5	SA1R21	0	R/W	—	
4	SA1R20	0	R/W	—	
3	SA1R19	0	R/W	—	
2	SA1R18	0	R/W	—	
1	SA1R17	0	R/W	—	
0	SA1R16	0	R/W	—	

18.3.26 Slave Address Register 2 (SAR2)

SAR2 selects the upper eight bits of slave start address of the flash memory obtained by translating the host address in HBAR2. The lower 16 bits are fixed H'0000. The contents of this register must not be changed in LPC/FW memory cycles (while LMCE is set to 1).

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
7	SA2R23	0	R/W	—	Slave Address Bits 23 to 16
6	SA2R22	0	R/W	—	Select bits 23 to 16 of the flash memory address obtained by translating the host address. The value H'10 to H'FF should not be selected.
5	SA2R21	0	R/W	—	
4	SA2R20	0	R/W	—	
3	SA2R19	0	R/W	—	
2	SA2R18	0	R/W	—	
1	SA2R17	0	R/W	—	
0	SA2R16	0	R/W	—	

18.3.27 On-Chip RAM Slave Address Register (RAMAR)

RAMAR selects the slave start address (bits 15 to 8) of the on-chip RAM obtained by translating the host address. Bits 23 to 16 are fixed H'FF and bits 7 to 0 are fixed H'00. The contents of this register must not be changed in LPC/FW memory cycles (while LMCE is set to 1).

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
7	RMR15	1	R/W	—	On-Chip RAM Slave Address Bits 15 to 8
6	RMR14	1	R/W	—	Select bits 15 to 8 of the on-chip RAM address obtained by translating the host address. Though H'D0 to H'EF can be set, setting only H'D1 to H'EF is valid. Setting other than these values is invalid. When setting is invalid, the previous value is retained.
5	RMR13	0	R/W	—	
4	RMR12	1	R/W	—	
3	RMR11	0	R/W	—	
2	RMR10	0	R/W	—	
1	RMR9	0	R/W	—	
0	RMR8	0	R/W	—	

Note: * The value must be selected so that the area selected by RBUFAR is not overlapped. In this case, data in the on-chip RAM may be changed.

18.3.28 Flash Memory Write Protect Registers H, M, and L (FWPRH, FWPRM, and FWPRL)

FWPR controls the protect blocks of the flash memory to be accessed in LPC/FW memory write cycles. The contents of this register must not be changed in LPC/FW memory cycles (while LMCE is set to 1).

- FWPRH

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
7	WPB23	1	R/W	—	Set to/clear the write protect blocks of the flash memory. WPB23 H'0F0000 to H'0FFFFF WPB22 H'0E0000 to H'0EFFFF WPB21 H'0D0000 to H'0DFFFF WPB20 H'0C0000 to H'0CFFFF WPB19 H'0B0000 to H'0BFFFF WPB18 H'0A0000 to H'0AFFFF WPB17 H'090000 to H'09FFFF WPB16 H'080000 to H'08FFFF 0: Clears the write protect (0 can be written to only once.) 1: Sets to the write protect
6	WPB22	1	R/W	—	
5	WPB21	1	R/W	—	
4	WPB20	1	R/W	—	
3	WPB19	1	R/W	—	
2	WPB18	1	R/W	—	
1	WPB17	1	R/W	—	
0	WPB16	1	R/W	—	

- FWPRM

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
7	WPB15	1	R/W	—	Set to/clear the write protect blocks of the flash memory. WPB15 H'070000 to H'07FFFF WPB14 H'060000 to H'06FFFF WPB13 H'050000 to H'05FFFF WPB12 H'040000 to H'04FFFF WPB11 H'030000 to H'03FFFF WPB10 H'020000 to H'02FFFF WPB9 H'010000 to H'01FFFF WPB8 H'00F000 to H'00FFFF 0: Clears the write protect (0 can be written to only once.) 1: Sets to the write protect
6	WPB14	1	R/W	—	
5	WPB13	1	R/W	—	
4	WPB12	1	R/W	—	
3	WPB11	1	R/W	—	
2	WPB10	1	R/W	—	
1	WPB9	1	R/W	—	
0	WPB8	1	R/W	—	

- FWPRL

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
7	WPB7	1	R/W	—	Set to/clear the write protect blocks of the flash memory. WPB7 H'00E000 to H'00EFFF WPB6 H'00D000 to H'00DFFF WPB5 H'00C000 to H'00CFFF WPB4 H'004000 to H'00BFFF WPB3 H'003000 to H'003FFF WPB2 H'002000 to H'002FFF WPB1 H'001000 to H'001FFF WPB0 H'000000 to H'000FFF 0: Clears the write protect (0 can be written to only once.) 1: Sets to the write protect
6	WPB6	1	R/W	—	
5	WPB5	1	R/W	—	
4	WPB4	1	R/W	—	
3	WPB3	1	R/W	—	
2	WPB2	1	R/W	—	
1	WPB2	1	R/W	—	
0	WPB0	1	R/W	—	

18.3.29 Flash Memory Read Protect Registers H, M, and L (FRPRH, FRPRM, and FRPRL)

FRPR controls the protect blocks of the flash memory to be accessed in LPC/FW memory read cycles. The contents of this register must not be changed in LPC/FW memory cycles (while LMCE is set to 1).

- FRPRH

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
7	RPB23	1	R/W	—	Set to/clear the read protect blocks of the flash memory.
6	RPB22	1	R/W	—	
5	RPB21	1	R/W	—	RPB23 H'0F0000 to H'0FFFFF
4	RPB20	1	R/W	—	RPB22 H'0E0000 to H'0EFFFF
3	RPB19	1	R/W	—	RPB21 H'0D0000 to H'0DFFFF
2	RPB18	1	R/W	—	RPB20 H'0C0000 to H'0CFFFF
1	RPB17	1	R/W	—	RPB19 H'0B0000 to H'0BFFFF
0	RPB16	1	R/W	—	RPB18 H'0A0000 to H'0AFFFF
					RPB17 H'090000 to H'09FFFF
					RPB16 H'080000 to H'08FFFF
					0: Clears to the read protect (0 can be written to only once.)
					1: Sets the read protect

- FRPRM

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
7	RPB15	1	R/W	—	Set to/clear the read protect blocks of the flash memory.
6	RPB14	1	R/W	—	
5	RPB13	1	R/W	—	RPB15 H'070000 to H'07FFFF
4	RPB12	1	R/W	—	RPB14 H'060000 to H'06FFFF
3	RPB11	1	R/W	—	RPB13 H'050000 to H'05FFFF
2	RPB10	1	R/W	—	RPB12 H'040000 to H'04FFFF
1	RPB9	1	R/W	—	RPB11 H'030000 to H'03FFFF
0	RPB8	1	R/W	—	RPB10 H'020000 to H'02FFFF
					RPB9 H'010000 to H'01FFFF
					RPB8 H'00F000 to H'00FFFF
					0: Clears to the read protect (0 can be written to only once.)
					1: Sets the read protect

- FRPRL

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
7	RPB7	1	R/W	—	Set to/clear the read protect blocks of the flash memory.
6	RPB6	1	R/W	—	
5	RPB5	1	R/W	—	RPB7 H'00E000 to H'00EFFF
4	RPB4	1	R/W	—	RPB6 H'00D000 to H'00DFFF
3	RPB3	1	R/W	—	RPB5 H'00C000 to H'00CFFF
2	RPB2	1	R/W	—	RPB4 H'004000 to H'00BFFF
1	RPB2	1	R/W	—	RPB3 H'003000 to H'003FFF
0	RPB0	1	R/W	—	RPB2 H'002000 to H'002FFF
					RPB1 H'001000 to H'001FFF
					RPB0 H'000000 to H'000FFF
					0: Clears to the read protect (0 can be written to only once.)
					1: Sets the read protect

18.3.30 On-Chip RAM Protect Control Register (MPCR)

MPCR controls the access to the on-chip RAM in LPC/FW memory RW cycles. The contents of this register must not be changed in LPC/FW memory cycles (while LMCE is set to 1).

			R/W		
Bit	**Bit Name**	**Initial Value**	**Slave**	**Host**	**Description**
7 to 2	—	All 0	R/W	—	Reserved
1	RAMWE	0	R/W	—	On-Chip RAM Write Access Enable Enables/disables the access to the on-chip RAM in LPC/FW memory write cycles. 0: Access inhibited 1: Access enabled
0	RAMRE	0	R/W	—	On-Chip RAM Read Access Enable Enables/disables the access to the on-chip RAM in LPC/FW memory read cycles. 0: Access inhibited 1: Access enabled

18.3.31 User Command Register (UCMDTR)

UCMDTR stores the user command data on user command reception.

			R/W		
Bit	**Bit Name**	**Initial Value**	**Slave**	**Host**	**Description**
7	bit7	0	R	W	User Command Data
6	bit6	0	R	W	Writing operation with the user command
5	bit5	0	R	W	
4	bit4	0	R	W	
3	bit3	0	R	W	
2	bit2	0	R	W	
1	bit1	0	R	W	
0	bit0	0	R	W	

18.4 Operation

18.4.1 LPC interface Activation

The LPC interface is activated by setting one of the following bits to 1: LPC3E to LPC1E in HICR0, LPC4E in HICR4, or LMCCR1 in LMCE. 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 channel 4, set 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.

   Set the relevant registers when the LPC/FW memory cycle is used.
4. Set the enable bit (LPC4E to LPC1E, and LMCE) 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, FLPI, FLEI, BUFINI and USERI). Read IDR or TWR15 to clear IBF.
8. Set receive complete interrupt enable bits (IBFIE4 to IBFIE1, ERRIE, FLPIE, FLEIE, BUFINIE, and USERIE) as necessary.

18.4.2 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, I/O write, LPC memory read, LPC memory write, FW memory read, and FW memory write cycles.

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 18.2 and 18.3.

Table 18.2 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

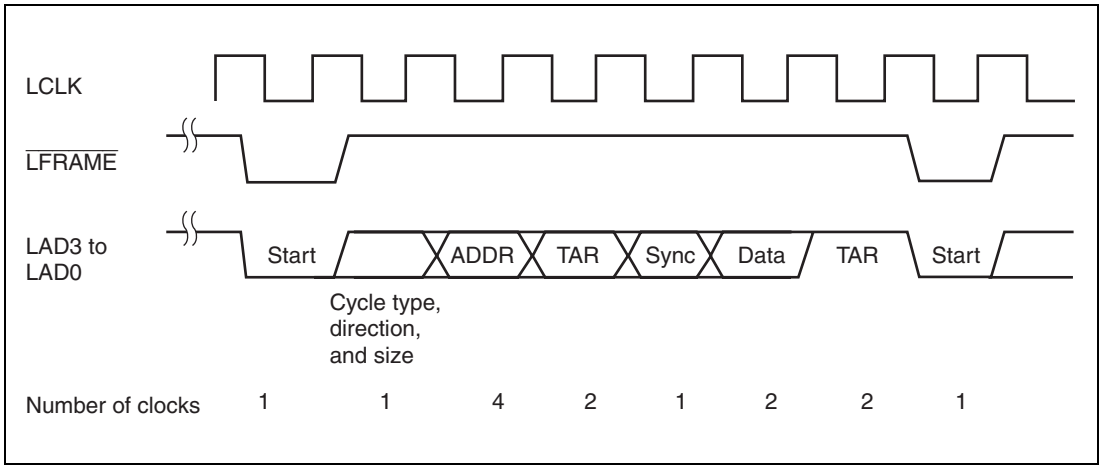


Figure 18.2 Typical LFRAME Timing

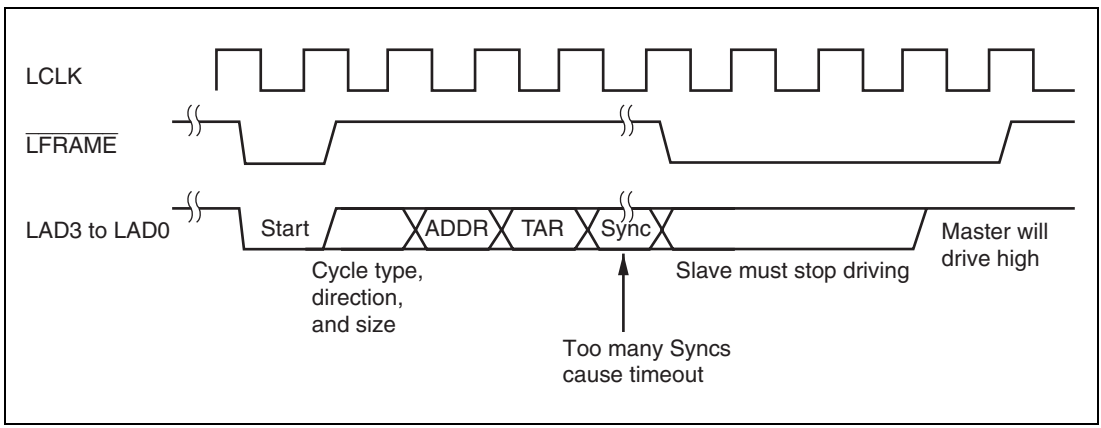


Figure 18.3 Abort Mechanism

18.4.3 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 18.3 shows the conditions that set and clear pin GA20. Figure 18.4 shows the GA20 output flow. Table 18.4 indicates the GA20 output signal values.

Table 18.3 GA20 (P81) 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

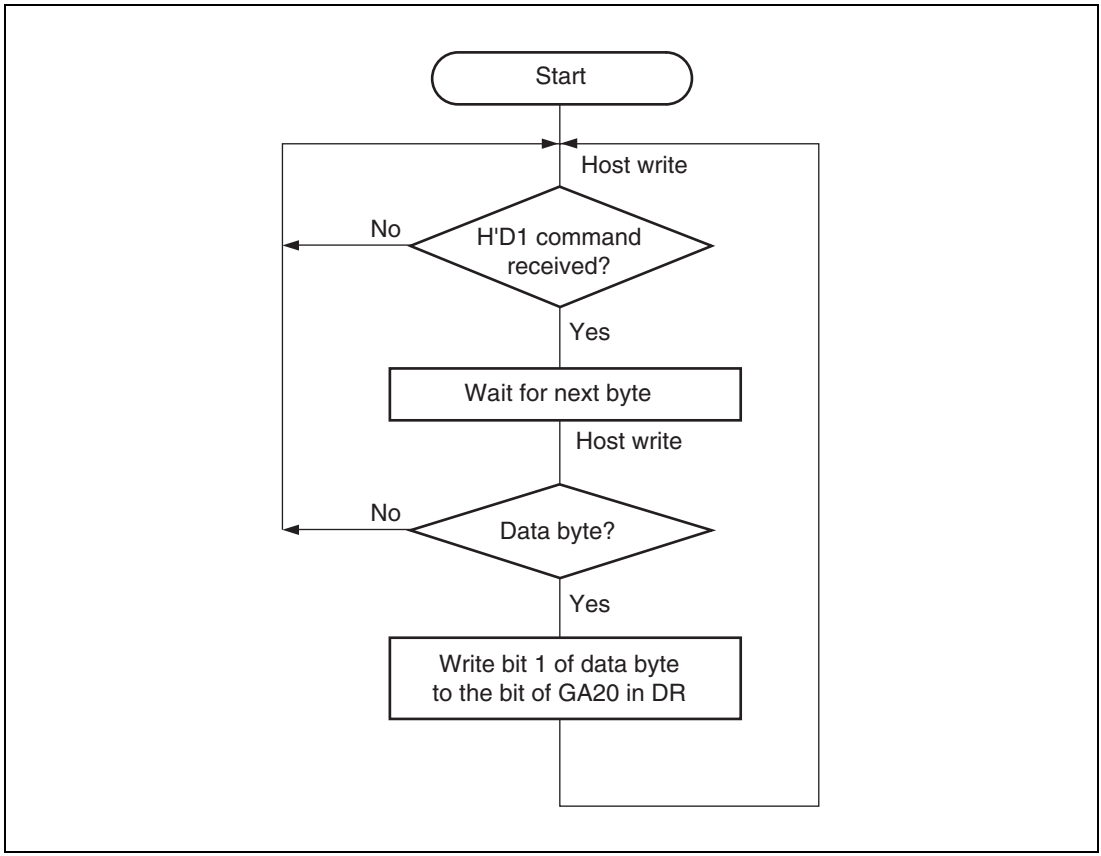


Figure 18.4 GA20 Output

Table 18.4 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.

18.4.4 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. Place the slave in sleep mode or software standby mode after confirming the LMCE bit in LMCCR1 cleared to 0, as necessary.
8. If software standby mode has been set, exit software standby mode by some means independent of the LPC.
9. When a rising edge is detected in the $\overline{\text{LPCPD}}$ signal, the SDWNE bit is 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 18.5 shows the scope of the LPC interface pin shutdown.

Table 18.5 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 $\overline{\text{STBY}}$ or $\overline{\text{RES}}$ pin input, or WDT0 overflow)
   All register bits, including bits LPC4E to LPC1E, are initialized.
2. LPC hardware reset (reset by $\overline{\text{LRESET}}$ pin input)
   LRSTB, SDWNE, and SDWNB bits are cleared to 0.
3. LPC software reset (reset by LRSTB)
   SDWNE and SDWNB bits are cleared to 0.
4. LPC hardware shutdown
   SDWNB bit is cleared to 0.
5. LPC software shutdown

The scope of the initialization in each mode is shown in table 18.6.

Table 18.6 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, LADR4 to LADR1, IBFIE1 to IBFIE4, PMEE, PMEB, LSMIE, LSMIB, LSCIE, LSCIB, TWRE, SELSTR3, SELIRQ1, SELSMI, SELIRQ6, SELIRQ9 to SELIRQ12, HBAR1, HBAR2, RAMBAR, SAR1, SAR2, RAMAR, ASSR, RAMASSR, FWPRH, FWPRM, FWPRL, FRPRH, FRPRM, FRPRL, FLPI, FLEI, BUFINII, USERI, FLPERR, FLEERR, PROTECT, LMCBUSY, LMCE, LPCME, FWE, FLASHE, HDINIE, FLPIE, FLEIE, BUFINIIE, USERIE, WAITSEL, MPCR, RBUFAR, FLWARH, FLWARL, LMCMIDCR, LMCDIDCR, EBLKR, UCMDTR)	Initialized	Retained	Retained
LPC interface operation control bits ERASEE, WRITEE, RUFTRAN	Initialized	Initialized	Retained

Items Initialized	System Reset	LPC Reset	LPC Shutdown
$\overline{\text{LRESET}}$ 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 STBY input, RES input, or WDT overflow
LPC reset: Reset by LPC hardware reset (HR) or LPC software reset (SR)
LPC shutdown: Reset by LPC hardware shutdown (HS) or LPC software shutdown (SS)

Figure 18.5 shows the timing of the $\overline{\text{LPCPD}}$ and $\overline{\text{LRESET}}$ signals.

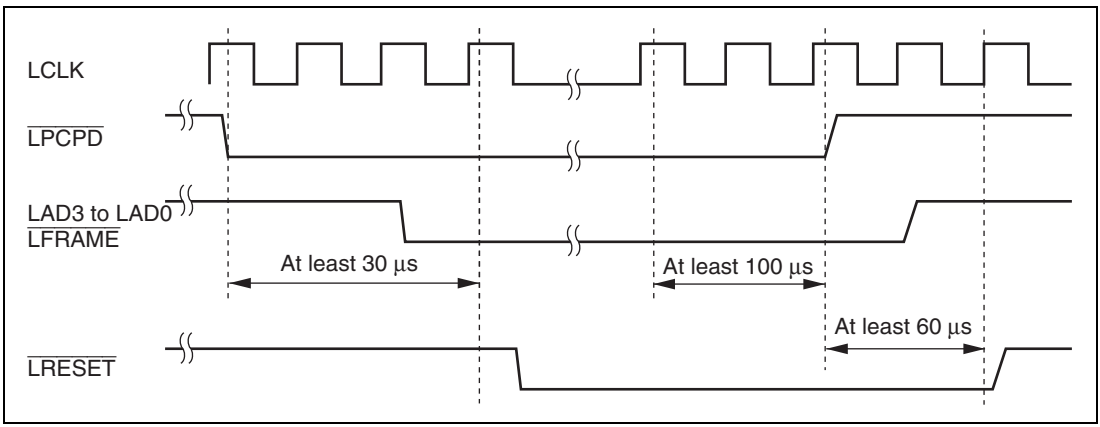


Figure 18.5 Power-Down State Termination Timing

18.4.5 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 18.6.

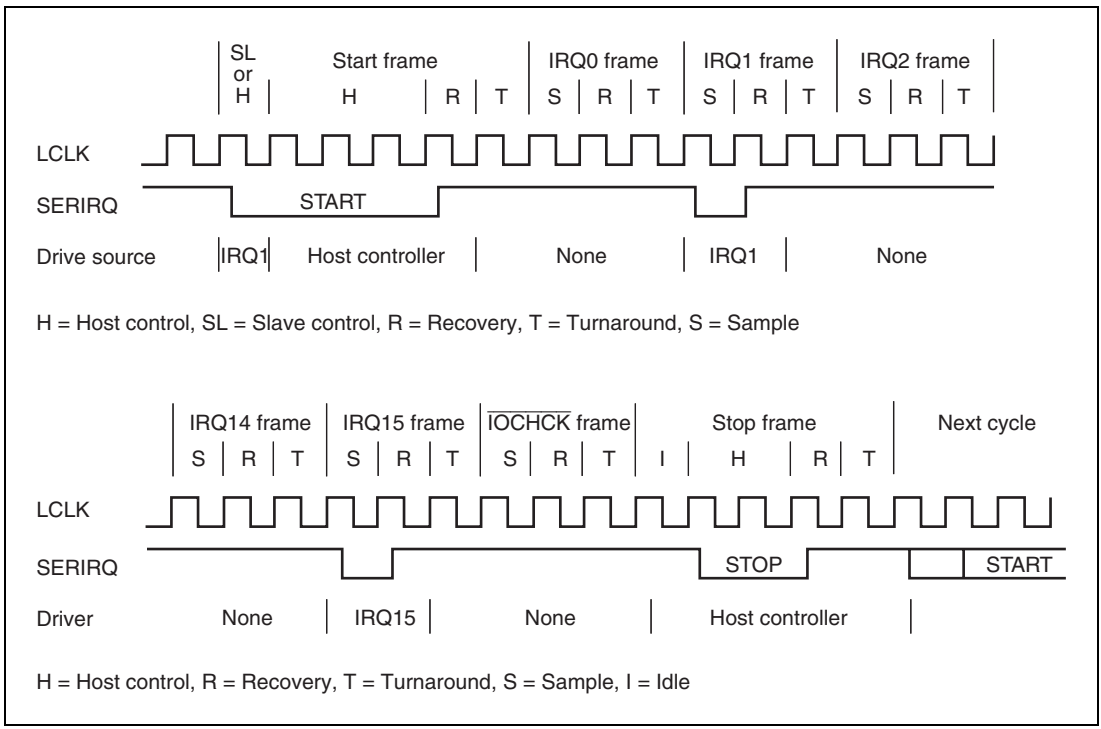


Figure 18.6 SERIRQ Timing

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 18.7 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
3	SMI	Slave	3	Drive possible in LPC channels 2, 3, and 4
4	IRQ3	Slave	3	
5	IRQ4	Slave	3	
6	IRQ5	Slave	3	
7	IRQ6	Slave	3	Drive possible in LPC channels 2, 3, and 4
8	IRQ7	Slave	3	
9	IRQ8	Slave	3	
10	IRQ9	Slave	3	Drive possible in LPC channels 2, 3, and 4
11	IRQ10	Slave	3	Drive possible in LPC channels 2, 3, and 4
12	IRQ11	Slave	3	Drive possible in LPC channels 2, 3, and 4
13	IRQ12	Slave	3	Drive possible in LPC channel 1
14	IRQ13	Slave	3	
15	IRQ14	Slave	3	
16	IRQ15	Slave	3	
17	IOCHCK	Slave	3	
18	Stop	Host	Undefined	First, 1 or more idle states, then 2 or 3 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 18.4.6, LPC Interface Clock Start Request.

18.4.6 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 18.7.

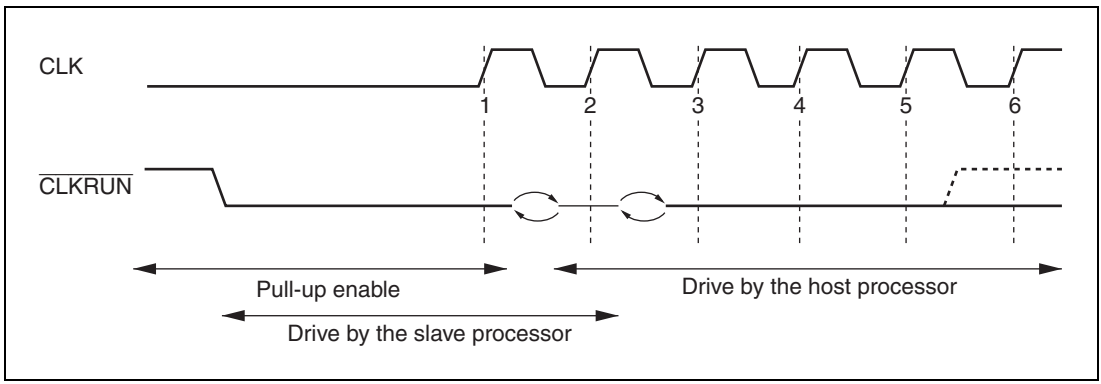


Figure 18.7 Clock Start Request Timing

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.

18.4.7 LPC/FW Memory Cycle

In LPC/FW memory read or LPC/FW memory write cycles, data is transferred via pins LAD3 to LAD0 in the following order, in synchronization with the LCLK. The slave can report an error to the host by sending back a value of B'1010 in the synchronization return cycle of the slave. The LPC of this LSI however a value of B'0000 (ready), B'0101 (short wait), or B'0110(long wait) always returns.

If the received address matches an address of the area the host can access (the area selected by the LPC registers: HBAR1, HBAR2, ASSR, RAMBAR, and RAMASSR), the LPC interface enters the busy state; it returns to the idle state by a turnaround output by the slave. 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. An on-chip memory, however, is read after receiving an

address, or an on-chip memory is written after receiving an address and data. So, if a transfer cycle forced termination (abort) after receiving an address and data, an on-chip memory may be accessed.

Table 18.8 LPC Memory Cycle

	LPC Memory Read Cycle			LPC Memory 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 31 to 28	Address 1	Host	Bits 31 to 28
4	Address 2	Host	Bits 27 to 24	Address 2	Host	Bits 27 to 24
5	Address 3	Host	Bits 23 to 20	Address 3	Host	Bits 23 to 20
6	Address 4	Host	Bits 19 to 16	Address 4	Host	Bits 19 to 16
7	Address 5	Host	Bits 15 to 12	Address 5	Host	Bits 15 to 12
8	Address 6	Host	Bits 11 to 8	Address 6	Host	Bits 11 to 8
9	Address 7	Host	Bits 7 to 4	Address 7	Host	Bits 7 to 4
10	Address 8	Host	Bits 3 to 0	Address 8	Host	Bits 3 to 0
11	Turnaround (recovery)	Host	1111	Data 1	Host	Bits 3 to 0
12	Turnaround	None	ZZZZ	Data 2	Host	Bits 7 to 4
13	Wait*	Slave	0101/0110	Turnaround (recovery)	Host	1111
14	Synchronization	Slave	0000	Turnaround	None	ZZZZ
15	Data 1	Slave	Bits 3 to 0	Synchronization	Slave	0000
16	Data 2	Slave	Bits 7 to 4	Turnaround (recovery)	Slave	1111
17	Turnaround (recovery)	Slave	1111	Turnaround	None	ZZZZ
18	Turnaround	None	ZZZZ	—	—	—

Note: * The number of wait states differs depending on the length of the time to turn the bus control over and the system clock frequency.

Table 18.9 FW Memory Cycle (Byte Transfer)

	FW Memory Read Cycle			FW Memory Write Cycle		
State Count	Contents	Drive Source	Value (3 to 0)	Contents	Drive Source	Value (3 to 0)
1	Start	Host	1101	Start	Host	1110
2	Device selection	Host	ID3 to ID0	Device selection	Host	ID3 to ID0
3	Address 1	Host	Bits 27 to 24	Address 1	Host	Bits 27 to 24
4	Address 2	Host	Bits 23 to 20	Address 2	Host	Bits 23 to 20
5	Address 3	Host	Bits 19 to 16	Address 3	Host	Bits 19 to 16
6	Address 4	Host	Bits 15 to 12	Address 4	Host	Bits 15 to 12
7	Address 5	Host	Bits 11 to 8	Address 5	Host	Bits 11 to 8
8	Address 6	Host	Bits 7 to 4	Address 6	Host	Bits 7 to 4
9	Address 7	Host	Bits 3 to 0	Address 7	Host	Bits 3 to 0
10	Size	Host	0000	Size	Host	0000
11	Turnaround (recovery)	Host	1111	Data 1	Host	Bits 3 to 0
12	Turnaround	None	ZZZZ	Data 2	Host	Bits 7 to 4
13	Wait*	Slave	0101/0110	Turnaround (recovery)	Host	1111
14	Synchronization	Slave	0000	Turnaround	None	ZZZZ
15	Data 1	Slave	Bits 3 to 0	Synchronization	Slave	0000
16	Data 2	Slave	Bits 7 to 4	Turnaround (recovery)	Slave	1111
17	Turnaround (recovery)	Slave	1111	Turnaround	None	ZZZZ
18	Turnaround	None	ZZZZ	—	—	—

Note: * The number of wait states differs depending on the length of the time to turn the bus control over and the system clock frequency.

The LPC supports byte, word, and longword transfer of the FW memory read and write cycle. When transferring in words, the LSB is fixed B'0 and when transferring longwords, the lower two bits are fixed B'00.

18.4.8 LPC/FW Memory Access Command

An LPC/FW memory cycle with a special address can be used as a command. It allows to control flash memory erasure, programming and to read the status register of the flash memory. The host address space of which lower 16 bits are H'FFF0 to H'FFFF can be used as command space according to the RAMBAR setting. Figure 18.8 shows an example of command space setting.

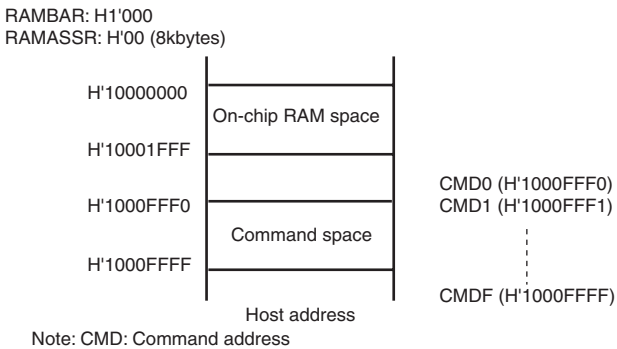


Figure 18.8 Example of Command Space Setting

Table 18.10 lists the LPC/FW memory access commands.

Table 18.10 List of LPC/FW Memory Access Commands

Command	Operation	Address	Size	Data	Wait State	Memory Access	Interrupt
Data read	R	FL/RM	B/W/L	Read data	O	O	X
ID read	R	CMD0 CMD1	B/W	MID DID	X	X	X
Status read	R	CMD2 CMD3	B/W	ST1 ST2	X	X	X
Clear status	W	CMD4	B	—	X	X	X
Erasing enable	W	CMD5	B	—	X	X	X
Block erasure	W	CMD6	B	Block number	X	X	O
Write enable	W	CMD7	B	—	X	X	X
Data write (on-chip RAM)	W	RM	B/W/L	Write data	X	O	X
WRITEE clear	W	CMD8	B	—	X	X	X
FLWAR set	W	FL	B	H'80	X	X	X
Data write (flash memory)	W	FL	B/W/L	Write data	X	O	X
Flash memory programming	W	CMD9	B	—	X	X	O
BUFTRAN clear	W	CMDA	B	—	X	X	X
Buffer initialization	W	CMDB	B	—	X	X/O*	O/X*
User command	W	CMDC	B	—	X	X	O

[Legend]

O: Available
X: Not available
FL: Flash memory address
MID/DID: ID codes (LMCMIDCR/LMCDIDCR)
RM: On-chip RAM address
ST1/2: LMCST (LMCST1/LMCST2)
CMDx: Command address
Block number: Erase block number
—: Any data

Note: * The HDINIE and BUFINIIE bits select whether or not memory is accessed and whether or not an interrupt is generated with the buffer initialization command.

1. Data read command

   When receiving an FL/RM address in an LPC/FW memory read cycle, the LPC sends back data of the address in the memory. Byte, word, and longword transfers are supported in FW memory read cycles.

2. ID read command

   When receiving the CMD0 or CMD1 address in an LPC/FW memory read cycle, the LPC sends back an MID/DID. Byte and word transfers are supported by the ID read command. In word transfer, an MID is sent back before a DID. In longword transfer, the SYNC field is not sent back.

3. Status read command

   When receiving the CMD2 or CMD3 address in an LPC/FW memory read cycle, the slave sends back an LMCST1/LMCST2. Byte and word transfers are supported by the status read command. In word transfer, an LMCST1 is sent back before an LMCST2. In longword transfer, the SYNC field is not sent back.

4. Clear Status command

   When receiving the CMD4 address in an LPC/FW memory write cycle, the slave clears the FLPERR, FLEERR, and ERASEE bits.

5. Erasing enable command

   When receiving the CMD5 address in an LPC/FW memory write cycle, the slave sets the ERASEE bit. Setting the ERASEE bit allows to receive the block erasure command.

6. Block erasure command

   When receiving the CMD6 address and a block number data with ERASEE = 1 in an LPC/FW memory write cycle, the LPC performs a block erasure. The LPC stores the erase block number in EBLKR, sets the FLEI interrupt flag (one of the LMCI interrupt sources) and clears the ERASEE bit when receiving the block erasure command. The LPC must clear the FLEI flag to 0 after reading FLEI = 1. The host reads LMCST1, waiting for the FLEI bit cleared. After reading FLEI = 0 and checking the FLERR bit, the host starts the next command. During block erasure, the commands for a memory access or an interrupt generation are prohibited.

7. Write enable command

   When receiving the CMD7 address in an LPC/FW memory write cycle, the LPC sets the WRITEE bit. Setting the WRITEE bit allows to receive the data write (on-chip RAM) command.

8. Data write (on-chip RAM) command

   When receiving an RM address and the data write command with WRITEE = 1 in an LPC/FW memory write cycle, the LPC writes data of the address in the on-chip RAM. Byte, word, and longword transfers are supported in FW memory write cycles. Since receiving the data write command to the on-chip RAM generates a memory access, confirm the LMCBUSY bit cleared to 0 after completion of an LPC/FW memory write cycle. It is needed to decide the internal memory access status since a wait state is not inserted in a write cycle. When the LMCBUSY is set, the commands for a memory access or an interrupt generation are prohibited. The WRITEE bit is not cleared after the data write command is completed. To clear the WRITEE bit, the WRITEE clear command must be executed.

9. WRITEE clear command

   When receiving the CMD8 address in an LPC/FW memory write cycle, the LPC clears the WRITEE bit. Clearing the WRITEE bit avoids unintentional write accesses to the on-chip RAM.

10. FLWAR set command

    When receiving an FL address and the data of H'80 in an LPC/FW memory write cycle, the LPC stores the FL address in FLWAR and set the BUFTRAN bit to 1 for the data write command (flash memory). The BUFTRAN bit must be cleared to 0 to set data in FLWAR. If the data of H'80 is written to the FL address, the data write command (flash memory) is executed. The BUFTRAN bit must be cleared to 0 to set data in FLWAR.

    Bits 4 to 0 in FLWARH store bits 19 to 15 of the transfer address and bits 7 to 0 in FLWARL store the bits 14 to 7 of the transfer address. Bits 7 to 5 in FLWARH are fixed B'000.

11. Data write (flash memory)

    When receiving an address which matches the FLWAR contents and data to be written to with the BUFTRAN bit set to 1 in an LPC/FW memory write cycle, the LPC stores the data to be written to the address selected by the RBUFAR contents. Byte, word, and longword transfers are supported in FW memory write cycles. Since receiving the data write command to the flash memory generates a memory access, confirm the LMCBUSY bit cleared to 0 after completion of an LPC/FW memory write cycle. It is needed to decide the internal memory access status since a wait state is not inserted in a write cycle. When the LMCBUSY is set, the commands for an interrupt generation are prohibited. The BUFTRAN bit is not cleared after the data write command is completed. To clear the BUFTRAN bit, the BUFTRAN clear command must be executed. The addresses are compared between bits 7 to 0 in FLWARH and bits 22 to 15 of the transfer address, or between bits 7 to 0 in FLWARH and bits 14 to 7 of the transfer address. The lower six bits are not compared and are used as a buffer address without any change.

12. Flash Programming memory

    When receiving the CMD9 address with the BUFTRAN set to 1 in an LPC/FW memory write cycle, the LPC programs the flash memory. When receiving the flash memory programming command, the LPC set the FLPI interrupt flag (one of the LMCI interrupt sources) and clears the BUFTRAN bit at the same time. The slave must clear the FLPI bit to 0 after reading FLPI = 1 on completion of programming the flash memory. The host reads LMCST1, waiting for the FLPI bit cleared. After reading FLPI = 0 and checking the FLPERR bit, the host starts the next command. During programming, the commands for a memory access or an interrupt generation are prohibited.

13. BUFTRAN clear

    When receiving the CMDA address in an LPC/FW memory write cycle, the LPC clears the BUFTRAN bit. FLWAR can store another data after clearing the BUFTRAN bit.

14. Buffer initialization command

    When receiving the CMDB address in an LPC/FW memory write cycle, the LPC initializes the buffer. When BUFINIIE = 1, or BUFINIIE = 0 and HDINIE = 0, the LPC sets the BUFINII interrupt flag (one of the LMCI interrupt sources) to 1 on reception of the buffer initialization command. The slave must clear the BUFINIIE bit to 0 after reading BUFINII = 1 on completion of the buffer initialization. The host reads LMCST1, waiting for the BUFINIIE bit cleared. During initialization, the commands for a memory access or an interrupt generation are prohibited. When BUFINIIE = 0 and HDINIE = 1, the LPC initializes the buffer. In this case, the buffer contents of 128 bytes are initialized to H'FF by generation of memory cycles. To decide whether or not to complete the initialization, check the LMCBUSY bit. During LMCBUSY = 1, the commands for a memory access or an interrupt generation are prohibited.

15. User command

    When receiving the CMDC address in an LPC/FW memory write cycle, the LPC executes the user command. When receiving the user command, the LPC sets the USERI interrupt flag. The slave must clear the USERI bit to 0 after reading USERI = 1. The host reads LMCST1, waiting for the USERI bit cleared. During execution, the commands for an interrupt generation are prohibited.

Table 18.11 lists the factors that prevent the SYNC from being sent back in an LPC/FW memory read cycle.

Table 18.11 List of Factors that Prevents SYNC Field being Sent Back

Command	Factor	Remarks
Common to all commands	Start not match	
	Device selection not match	In FW memory access cycle
	Address not match	
	Size error	In FW memory access cycle (other than byte, word, longword)
Data read	Reading address which has already been accessed disabled	According to the FRPR/RAMRE register settings
	Accessing on-chip memory is in progress	LMCBUSY = 1
	Interrupt processing is in progress	While following processes are in progress: • Programming flash memory (FLPI = 1) • Erasing flash memory (FLEI = 1) • Initializing buffer (BUFINII = 1) • Processing user command (USERI = 1)
Block erasure	Erasure command disabled	ERASEE = 0
	Erasure protect blocks	According to the FWPR register setting
	Erase block number error	Other than 0 to 23
	On-chip memory access is in progress	LMCBUSY = 1
	Interrupt processing is in progress	While following processes are in progress: • Programming flash memory (FLPI = 1) • Erasing flash memory (FLEI = 1) • Initializing buffer (BUFINII = 1) • Processing user command (USERI = 1)

Command	Factor	Remarks
Data write (on-chip RAM)	On-chip RAM data write command disabled	WRITEE = 0
	On-chip RAM write access disabled	RAMWE = 0
	O-chip memory access is in progress	LMCBUSY = 1
	Interrupt processing is in progress	While following processes are in progress: • Programming flash memory (FLPI = 1) • Erasing flash memory (FLEI = 1) • Initializing buffer (BUFINII = 1) • Processing user command (USERI = 1)
FLWAR set	Flash memory write protect	According to the FWPR register
	Data error	Other than H'80
	On-chip memory access is in progress	LMCBUSY = 1
	Interrupt processing is in progress	While following processes are in progress: • Programming flash memory (FLPI = 1) • Erasing flash memory (FLEI = 1) • Initializing buffer (BUFINII = 1) • Processing user command (USERI = 1)
Data write (flash memory)	Flash memory write protect	According to the FWPR register
	Address not match FLWAR (128 bytes)	128 bytes
	On-chip memory access is in progress	LMCBUSY = 1
	Interrupt processing is in progress	While following processes are in progress: • Programming flash memory (FLPI = 1) • Erasing flash memory (FLEI = 1) • Initializing buffer (BUFINII = 1) • Processing user command (USERI = 1)

Command	Factor	Remarks
Flash memory programming	FLWAR setting command not issued	BUFTRAN = 0
	On-chip memory access is in progress	LMCBUSY = 1
	Interrupt processing is in progress	While following processes are in progress: • Programming flash memory (FLPI = 1) • Erasing flash memory (FLEI = 1) • Initializing buffer (BUFINII = 1) • Processing user command (USERI = 1)
Buffer initialization	On-chip memory access is in progress	LMCBUSY = 1
	Interrupt processing is in progress	While following processes are in progress: • Programming flash memory (FLPI = 1) • Erasing flash memory (FLEI = 1) • Initializing buffer (BUFINII = 1) • Processing user command (USERI = 1)
User command	On-chip memory access is in progress	LMCBUSY = 1
	Interrupt processing is in progress	While following processes are in progress: • Programming flash memory (FLPI = 1) • Erasing flash memory (FLEI = 1) • Initializing buffer (BUFINII = 1) Processing user command (USERI = 1)

18.4.9 Flash Memory Address Translation (Host → Slave)

A host address is translated into a flash memory address by the settings of HBAR1, HBAR2, ASSR, SAR1, and SAR2. The slave address which exceeds H'0FFFFF must not be specified. The host addresses within the range of H'00000000 to H'FFFFFFFF are available for translation and the flash memory addresses within the range of H'000000 to H'0FFFFF are available. Figure 18.9 shows an example of the flash memory address translation.

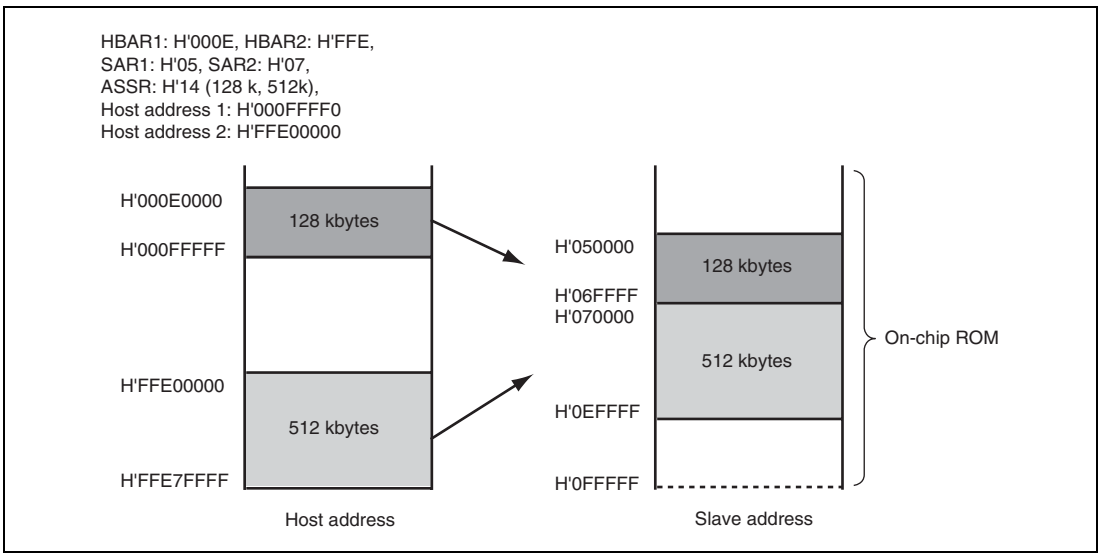


Figure 18.9 Example of Flash Memory Address Translation

The host address space is specified by the settings of HBAR1, HBAR2, and ASSR as shown in figure 18.9. The host must use addresses within this range. Addresses which are not within this range will cause an address exception and any memory access is not performed.

The slave address space is specified by the settings of SAR1, SAR2, and ASSR as shown in figure 18.9.

The host and slave address spaces are specified by the settings of HBAR1, bits 7 to 4 in ASSR, and SAR1, or HBAR2, bits 3 to 0 in ASSR, and SAR2. Each of those two areas must be specified so that they are not overlapped. If two areas are overlapped, address space 1 has priority.

The host address of H'000FFF0 is translated shown below.

1. Host address – HBAR1 = H'000FFFF0 – H'000E0000 = H'0001FFF0
2. H'0001FFF0 + SAR1 = H'0001FFF0 + H'050000 = H'06FFF0 (slave address)

18.4.10 On-Chip RAM Address Translation (Host → Slave)

A host address is translated into an on-chip RAM address by the settings of RAMBAR, RAMSSR, and RAMAR. The slave address which exceeds H'FFEFFF must not be specified. The host addresses within the range of H'00000000 to H'FFFFFFFF are available for translation and the on-chip RAM addresses within the range of H'FFD100 to H'FFEFFF are available. Figure 18.10 shows an example of the on-chip RAM address translation.

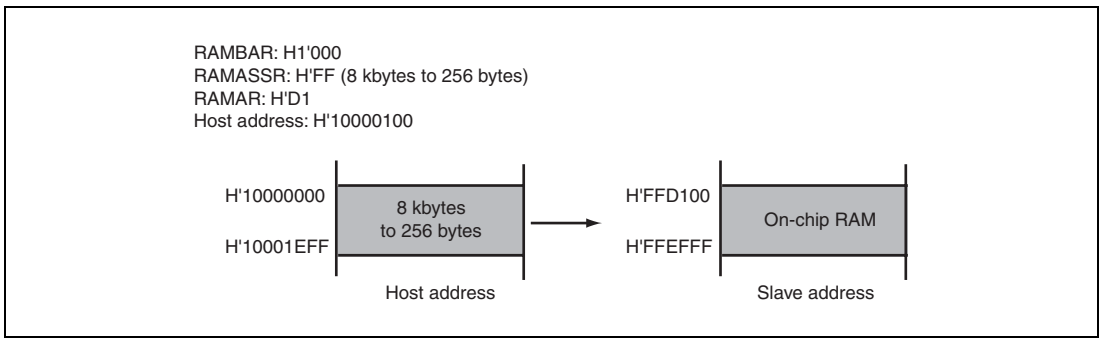


Figure 18.10 Example of On-Chip RAM Address Translation

The host address space is specified by the settings of RAMBAR and RAMASSR as shown in figure 18.10. The host must use addresses within this range. An access to the addresses which are not within this range will cause an address exception and any memory access is not performed.

The slave address space is specified by the settings of RAMAR and RAMASSR as shown in figure 18.10.

The host address of H'10000100 is translated shown below.

1. Host address – RAMBAR = H'10000100 – H'10000000 = H'00000100
2. H'0000100 + RAMAR = H'00000100 + H'FFD100 = H'FFD200 (slave address)

18.4.11 Address Space Priority

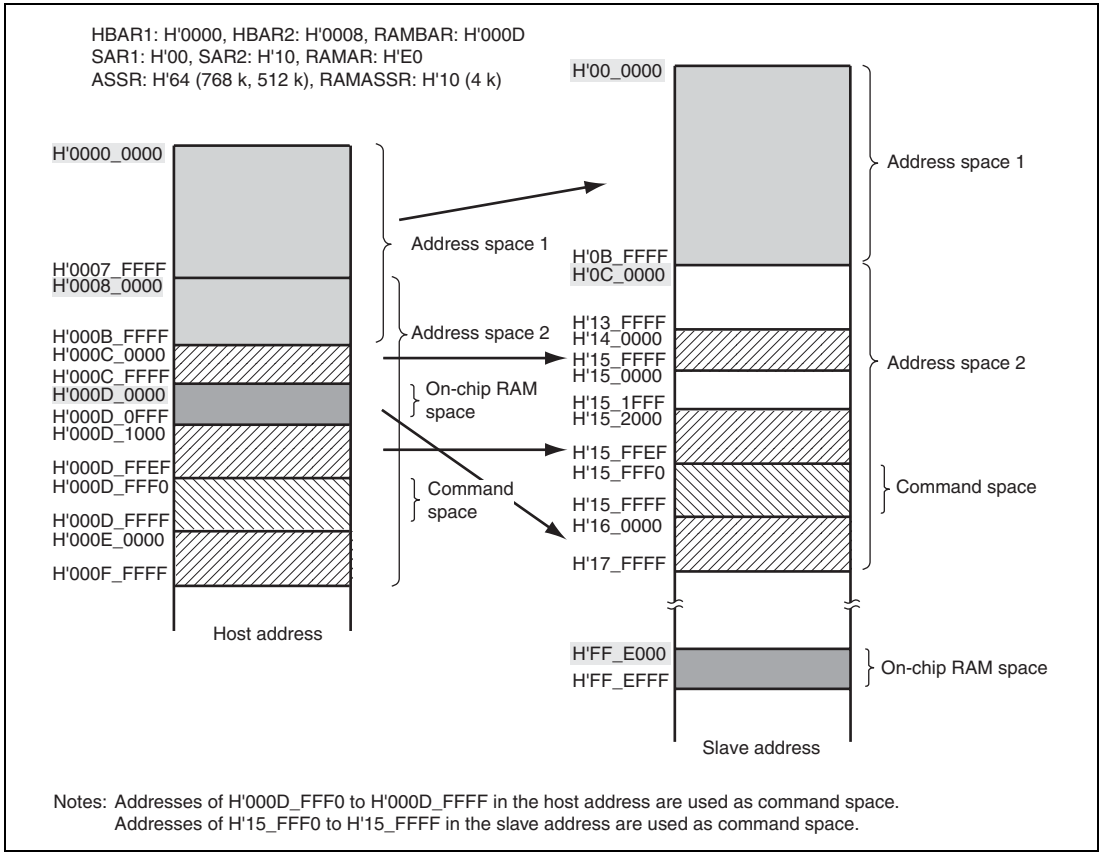
The host addresses can be specified from H'00000000 to H'FFFF0000 by the settings of HBAR1, HBAR2, and RAMBAR. The host address spaces, however, may be overlapped depending on the ASSR setting. Moreover, the slave address spaces may be overlapped depending on the settings of SAR1, SAR2, and ASSR. For this, individual address spaces are given priority. The priority is as follows:

Command space > address space 1 > on-chip RAM space > address space 2

Notes: 1. Please keep in mind that the slave on-chip RAM space may be overlapped with the RBUFAR space.
2. Only the flash memory can be accessed in address spaces 1 and 2.

18.4.12 Example 1 of Address Space Priority

Figure 18.11 shows an example of an address translation when the host address spaces are overlapped. When an address is translated with the settings shown in figure 18.11, host address spaces 1 and 2 are overlapped and host address space 2 and the on-chip RAM space are overlapped. In this case, since command space has priority over others, it is translated into slave command space. Then address space 1 is translated into slave address space 1. Next, the on-chip RAM which has priority over address space 2 and is not overlapped with address space 1 is translated into the slave on-chip RAM space.



Notes: Addresses of H'000D_FFF0 to H'000D_FFFF in the host address are used as command space.
Addresses of H'15_FFF0 to H'15_FFFF in the slave address are used as command space.

Figure 18.11 Example 1 of Address Space Priority

18.4.13 Example 2 of Address Space Priority

Figure 18.12 shows another example of an address translation when the slave address spaces are overlapped. In the case of the settings shown in figure 18.12, slave address spaces 1 and 2 are overlapped. Since host address spaces 1 and 2, however, are not overlapped, those addresses are always translated into the corresponding slave addresses. As long as the host address spaces are not overlapped, host addresses are translated into the corresponding slave addresses even if the slave address spaces are overlapped.

In this example, addresses of H'040000 to H'0BFFFF in the slave address space are accessed from both host address spaces 1 and 2.

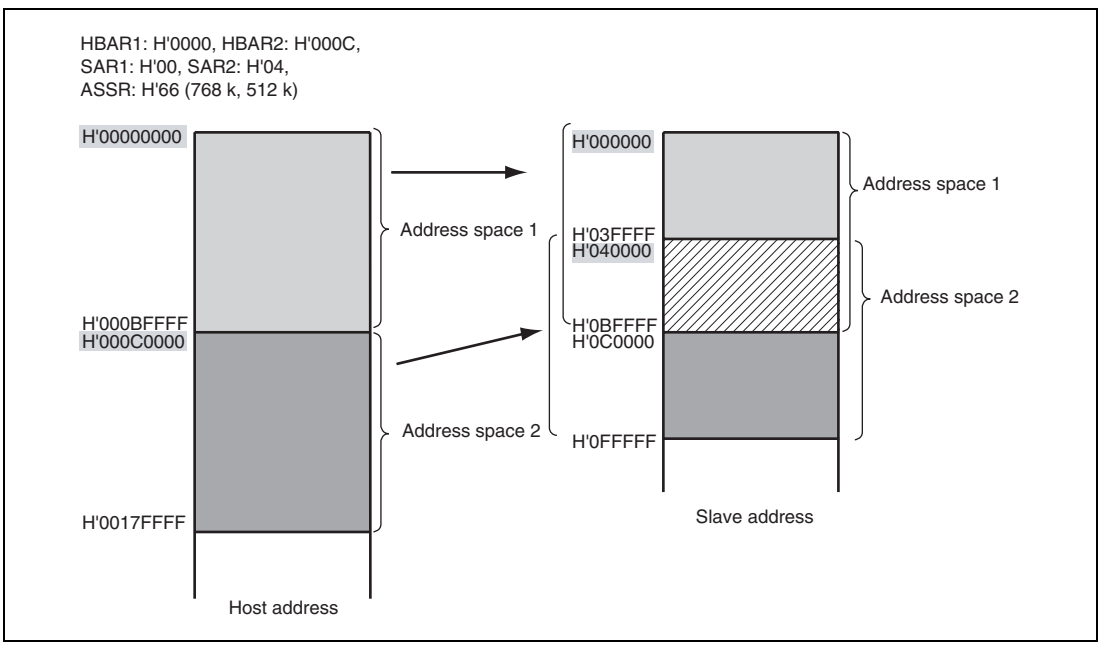


Figure 18.12 Example 2 of Address Space Priority

18.4.14 Flash Memory Protection

To protect the flash memory contents, the flash memory is divided into blocks given in figure 18.13. Protection for each block can be enabled or disabled by the setting of the WPB or RPB bits.

The write blocks are used in LPC/FW memory write cycles and the read blocks are used in LPC/FW memory read cycles. Access to the flash memory with other than LPC/FW memory cycles are not supported. The write block protection by the WPB bit is used to set/clear erase block protection in the block erasure command.

All the blocks are protected by the initial values. When accessing to the flash memory, the protection must be disabled. When the protection is disabled, 0 can be written to only once after system reset. If the protection is enabled again after it has been disabled, the protection can be disabled only by the system reset.

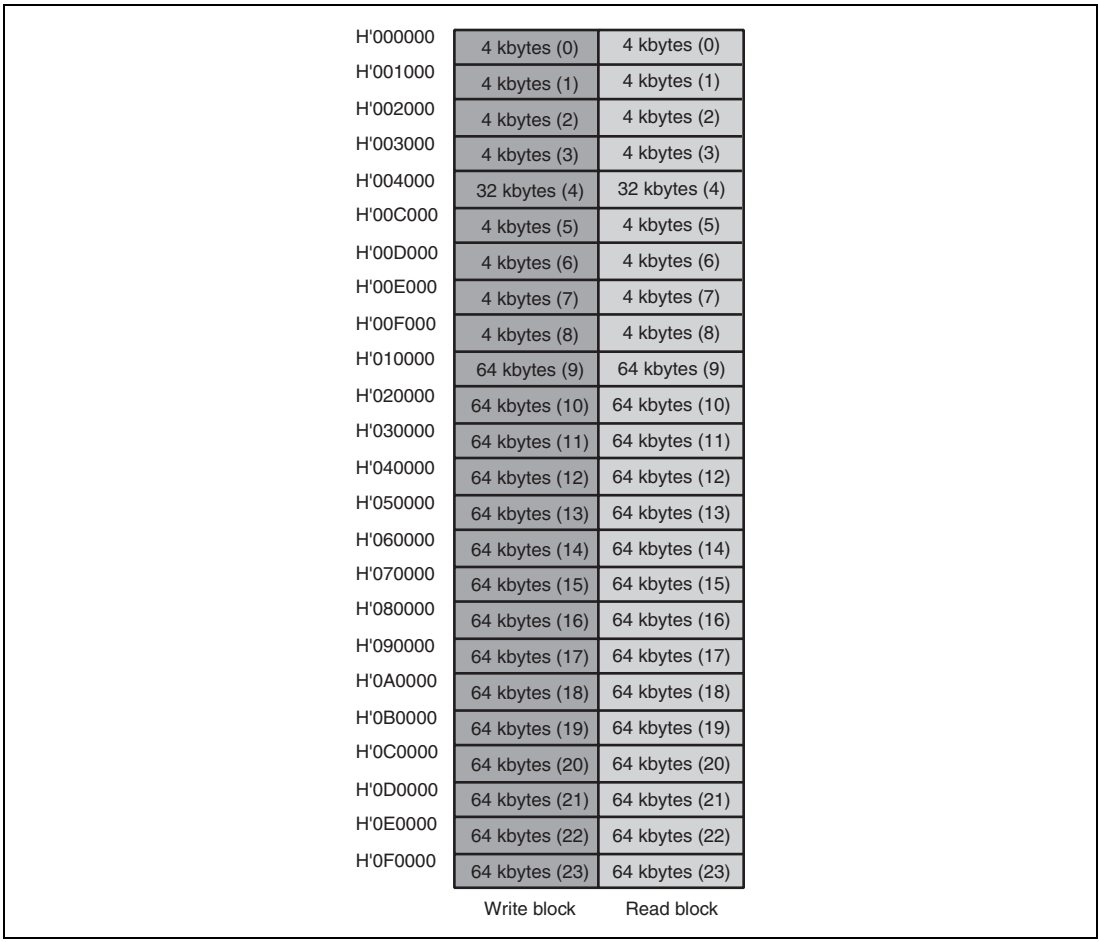


Figure 18.13 Flash Memory Protection

18.4.15 On-Chip RAM Protection

The on-chip RAM protection to the host access can be enabled or disabled by the setting of the RAMWE or RAMRE bits. The on-chip RAM is protected by the initial value. When accessing to the on-chip RAM, the protection must be disabled.

Figure 18.14 shows the protected address space in the on-chip RAM.

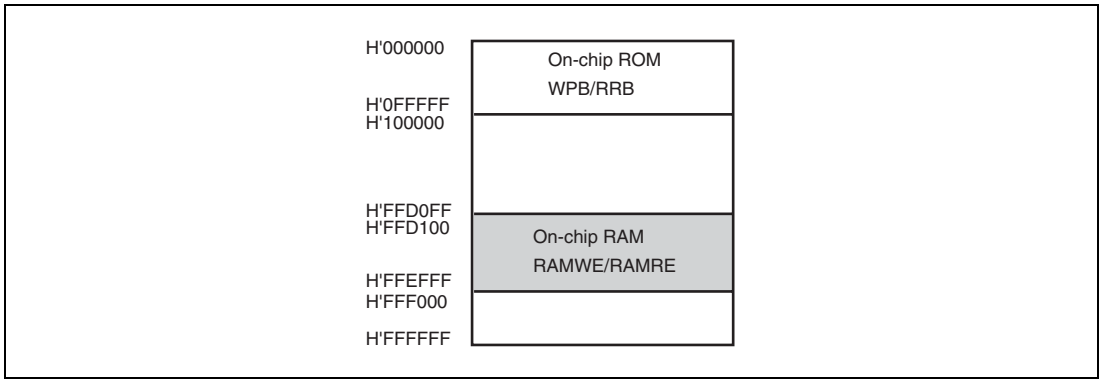


Figure 18.14 Protected Address Space in On-Chip RAM

18.4.16 Flash Memory Programming

Figure 18.15 shows an example of flowchart for programming the flash memory in the LPC/FW memory write cycle.

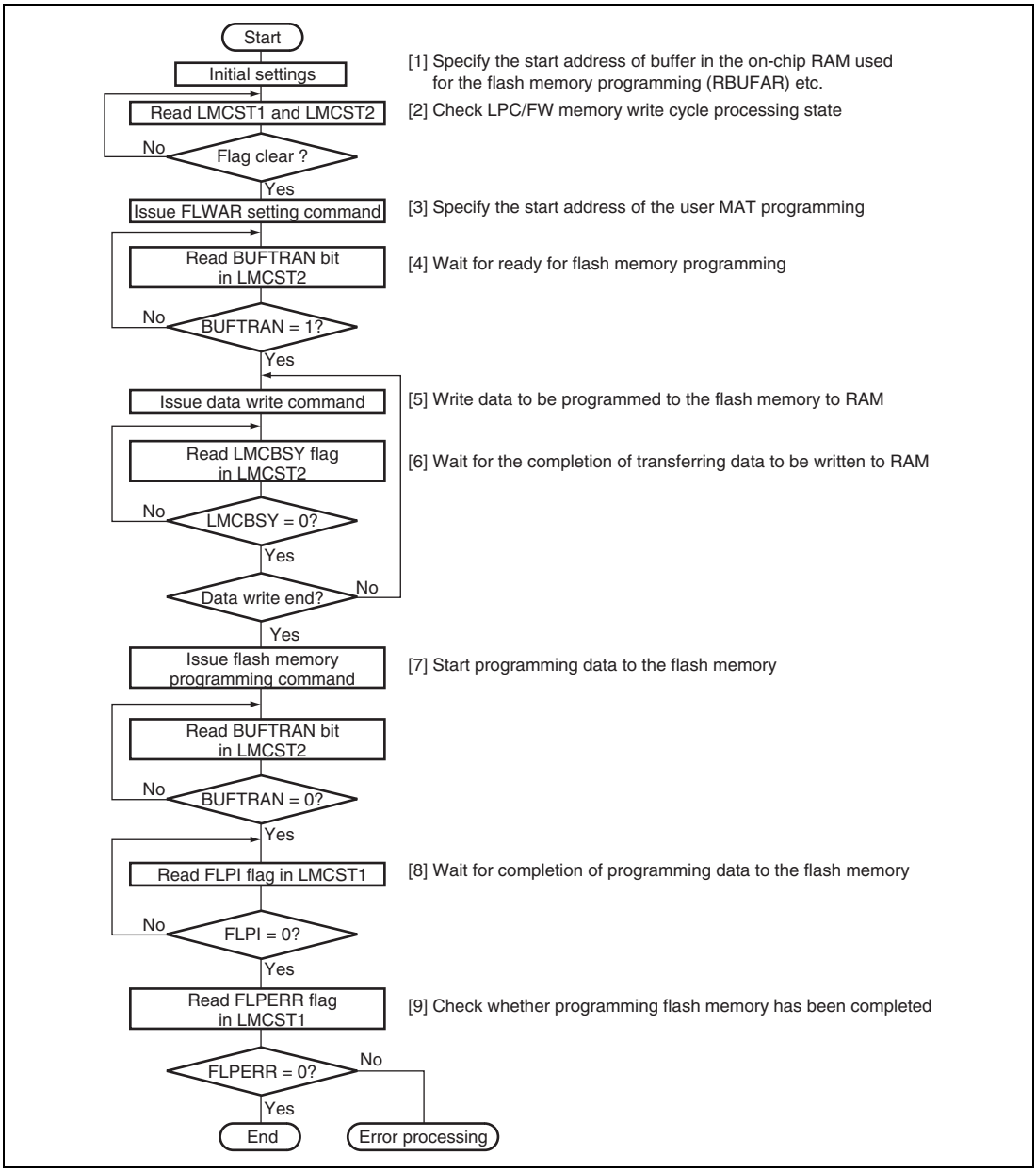


Figure 18.15 Example of Programming Flash Memory

18.4.17 Flash Memory Erasing

Figure 18.16 shows a flowchart example of erasing the flash memory in the LPC/FW memory write cycle.

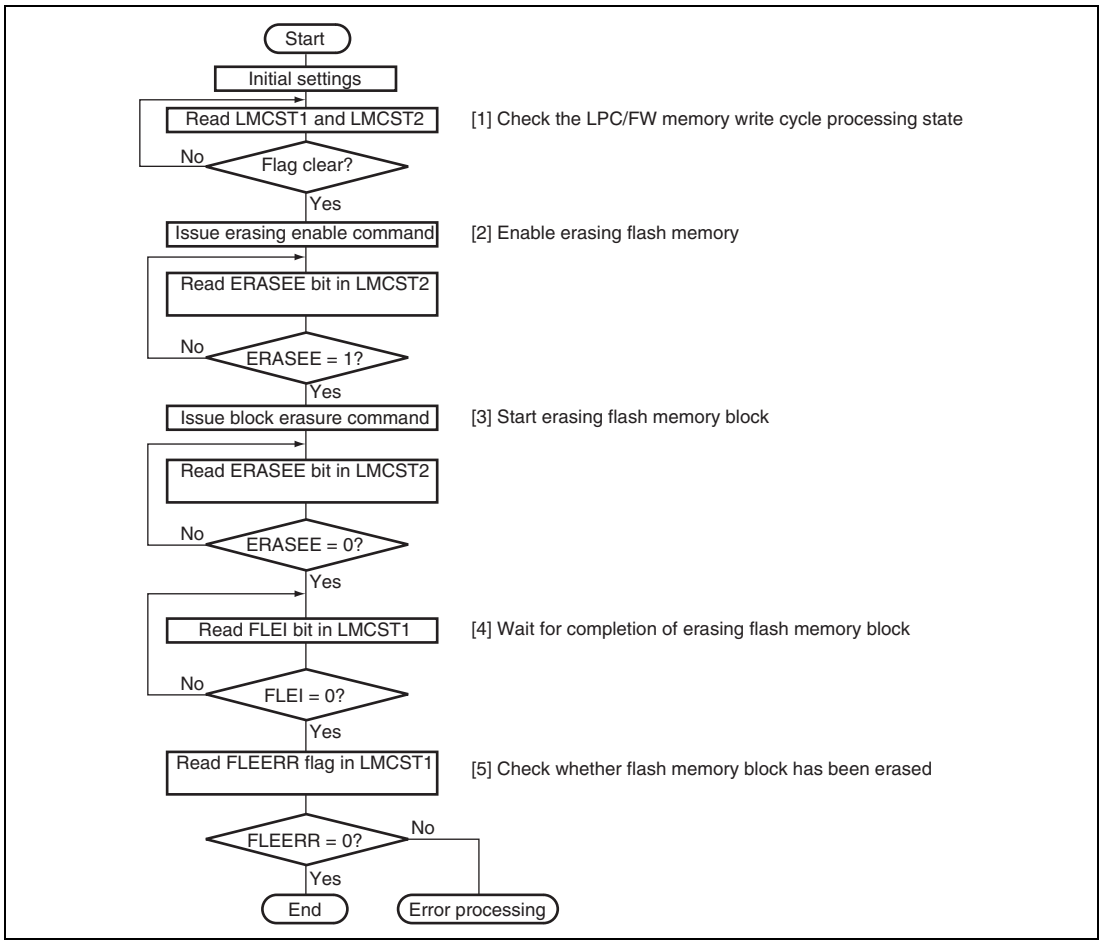


Figure 18.16 Example of Erasing Flash Memory

18.5 Interrupt Sources

18.5.1 IBFI1, IBFI2, IBFI3, IBFI4, LMC, LMCUI, and ERRI

The host has seven interrupt requests for the slave (this LSI): IBF1, IBF2, IBF3, IBF4, LMC, LMCUI, 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. An interrupt request is enabled by setting the corresponding enable bit.

Table 18.12 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
LMCI	• When the FLPI bit is set to 1 with the FLPIE and FLASHE bits set to 1 • When the FLEI bit is set to 1 with the FLEIE and FLASHE bits set to 1 • When the BUFINI bit is set to 1 with the BUFINIE bit set to 1
LMCUI	When the USERI bit is set to 1 with the USERIE bit set to 1.
ERRI	When ERRIE is set to 1 and one of LRST, SDWN and ABRT is set to 1

18.5.2 SMI, HIRQ1, HIRQ6, HIRQ9, HIRQ10, HIRQ11, and HIRQ12

The LPC interface can request seven kinds of host interrupt by means of SERIRQ. HIRQ1 and HIRQ12 are used on LPC channel 1 only, while SMI, HIRQ6, HIRQ9, HIRQ10, and HIRQ11 can be requested from LPC channel 2, 3, or 4.

There are two ways of clearing a host interrupt request.

When the IEDIR bit in SIRQCR0is 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 SIRQCR0, 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, SMIE, 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.)

Table 18.13 summarizes the methods of setting and clearing these bits, and figure 18.17 shows the processing flowchart.

Table 18.13 HIRQ Setting and Clearing Conditions

Host Interrupt	Setting Condition	Clearing Condition
HIRQ1 (independent from IEDIR)	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 (independent from IEDIR)	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 = 1, IEDIR3 = 1, or IEDIR4 = 1)	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 = 1, IEDIR3 = 1, or IEDIR4 = 1)	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

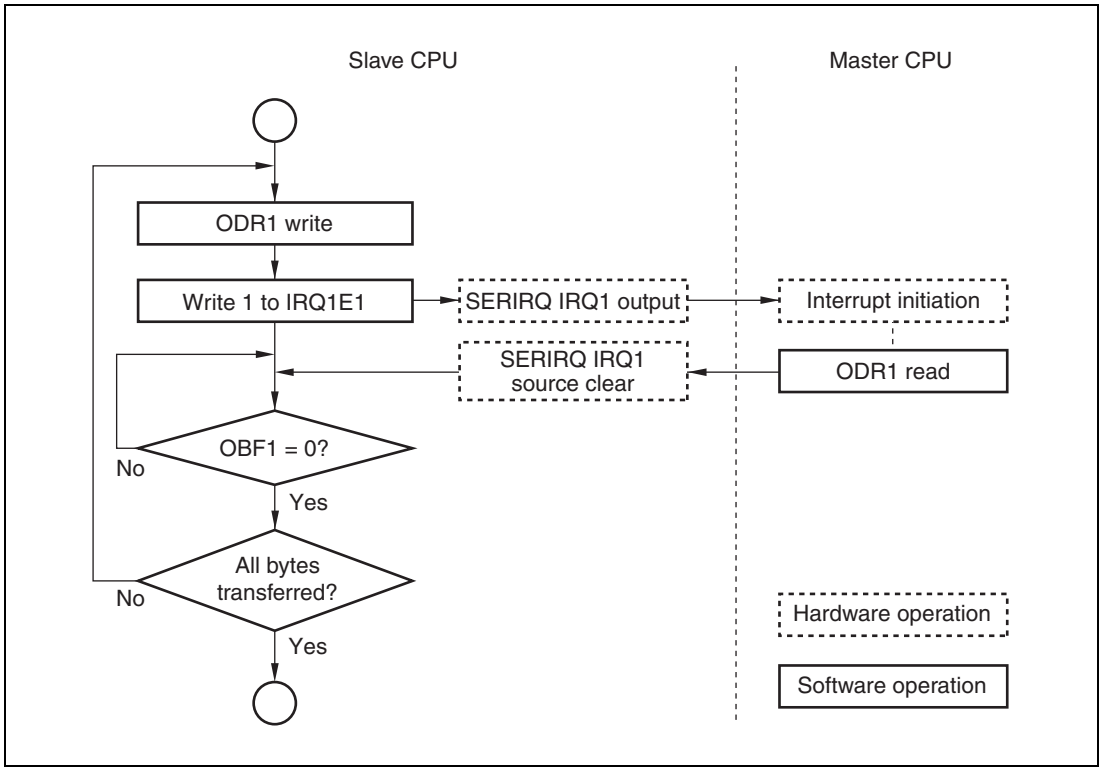


Figure 18.17 HIRQ Flowchart (Example of Channel 1)

18.6 Usage Note

18.6.1 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 18.14 shows host address examples for LADR3 and registers, IDR3, ODR3, STR3, TWR0MW, TWR0SW, and TWR1 to TWR15.

Table 18.14 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

18.6.2 Module Stop Mode Setting

Module stop mode should not be set to the LPC while the LPC/FW memory write cycle is enabled. Specify module stop mode after confirming that the LMCE bit in LMCCR1 is cleared to 0.

18.6.3 Operating Mode in LPC/FW Memory Write Cycle

The LPC/FW memory cycle should be enabled in advanced mode; disabled in normal mode.

Section 19 A/D Converter

This LSI includes a successive-approximation-type 10-bit A/D converter that allows up to eight analog input channels to be selected.

19.1 Features

- 10-bit resolution
- Input channels: Eight analog input channels
- Analog conversion voltage range can be specified using the reference power supply voltage pin (AVref) as an analog reference voltage.
- Conversion time: 13.4 µs per channel (at 20-MHz operation)
- Two kinds of operating modes
  Single mode: Single-channel A/D conversion
  Scan mode: Continuous A/D conversion on one to four channels
- Four 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
  Timer (TPU or 8-bit timer) conversion start trigger
  External trigger signal
- Interrupt source
  A/D conversion end interrupt (ADI) request can be generated
- Module stop mode can be set

A block diagram of the A/D converter is shown in figure 19.1.

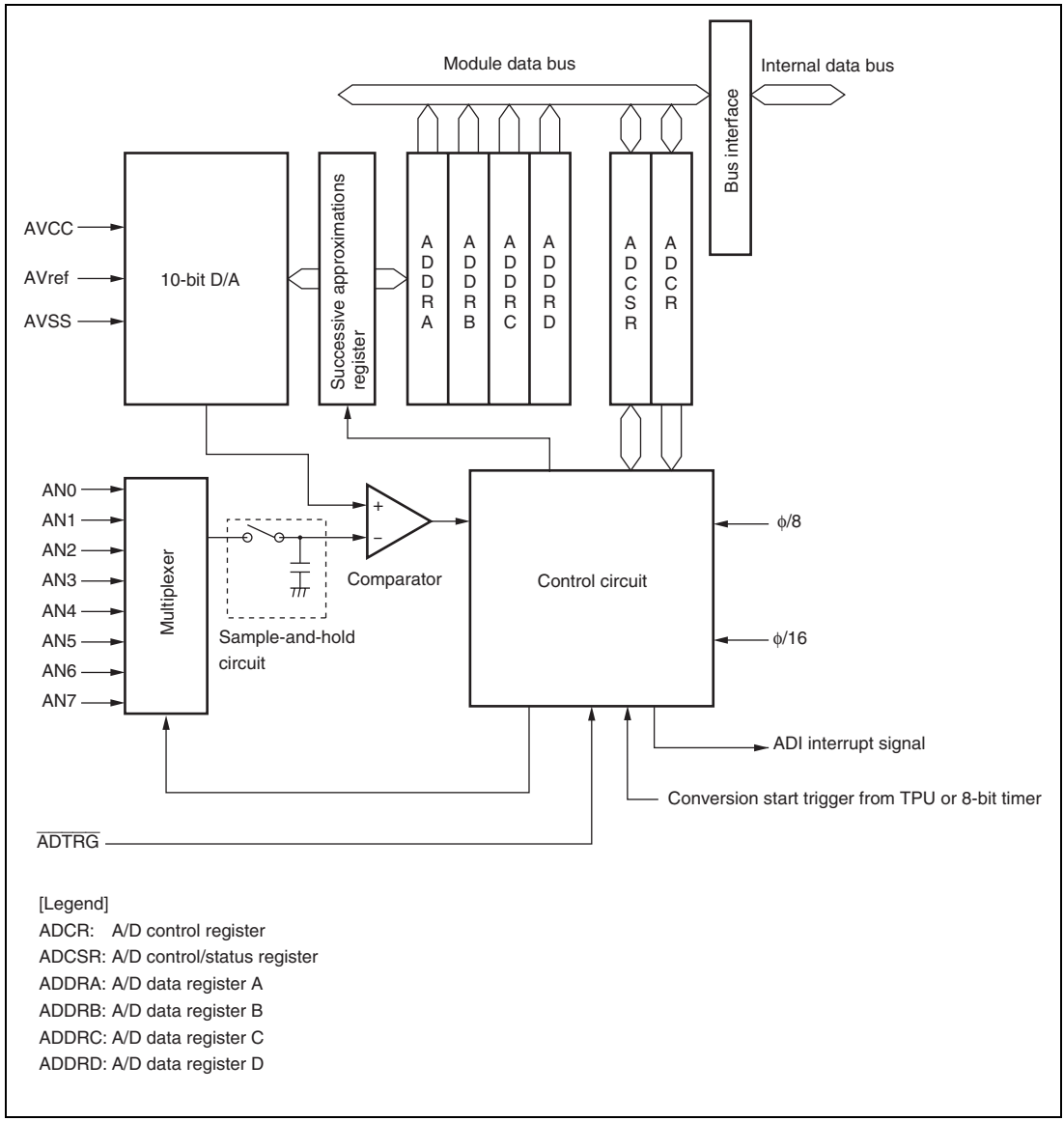


Figure 19.1 Block Diagram of A/D Converter

19.2 Input/Output Pins

Table 19.1 summarizes the pins used by the A/D converter. The eight analog input pins are divided into two groups consisting of four channels. Analog input pins 0 to 3 (AN0 to AN3) comprising group 0 and analog input pins 4 to 7 (AN4 to AN7) comprising group1. The AVCC and AVSS pins are the power supply pins for the analog block in the A/D converter.

Table 19.1 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 and reference voltage
Reference power supply pin	AVref	Input	Analog block reference voltage
Analog input pin 0	AN0	Input	Group 0 analog input pins
Analog input pin 1	AN1	Input	
Analog input pin 2	AN2	Input	
Analog input pin 3	AN3	Input	
Analog input pin 4	AN4	Input	Group 1 analog input pins
Analog input pin 5	AN5	Input	
Analog input pin 6	AN6	Input	
Analog input pin 7	AN7	Input	
A/D external trigger input pin	$\overline{\text{ADTRG}}$	Input	External trigger input pin for starting A/D conversion

19.3 Register Descriptions

The A/D converter has the following registers.

- A/D data register A (ADDRA)
- A/D data register B (ADDRB)
- A/D data register C (ADDRC)
- A/D data register D (ADDRD)
- A/D control/status register (ADCSR)
- A/D control register (ADCR)

19.3.1 A/D Data Registers A to D (ADDRA to ADDRD)

There are four 16-bit read-only ADDR registers, ADDRA to ADDRD, used to store the results of A/D conversion. The ADDR registers which store a conversion result for each channel are shown in table 19.2.

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 A/D converter is eight bits wide. The upper byte can be read directly from the CPU. However, when the lower byte is read from, data that was transferred to a temporary register at reading of the upper byte is read. Accordingly, when reading from ADDR, access in word units or access upper byte first, and then lower byte.

Table 19.2 Analog Input Channels and Corresponding ADDR

Analog Input Channel		A/D Data Register to Store A/D Conversion Results
Group 0	Group 1	
AN0	AN4	ADDRA
AN1	AN5	ADDRB
AN2	AN6	ADDRC
AN3	AN7	ADDRD

19.3.2 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 conditions] • When 0 is written after reading ADF = 1 • When DTC is activated by an ADI interrupt and ADDR is read
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 Setting this bit to 1 starts A/D conversion. In single mode, this bit is cleared to 0 automatically 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 software, a reset, or a transition to standby mode or module stop mode.
4	SCAN	0	R/W	Scan Mode Selects the A/D converter operating mode. 0: Single mode 1: Scan mode Switch the operating mode when ADST = 0.
3	CKS	0	R/W	Clock Select Sets A/D conversion time. 0: Conversion time is 266 states (max) 1: Conversion time is 134 states (max) (when the system clock (ϕ) is 16 MHz or lower) Switch conversion time while the ADST bit is cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description	
2	CH2	0	R/W	Channel Select 2 to 0	
1	CH1	0	R/W	Select analog input channels.	
0	CH0	0	R/W	When SCAN = 0	When SCAN = 1
				000: AN0	000: AN0
				001: AN1	001: AN0 and AN1
				010: AN2	010: AN0 to AN2
				011: AN3	011: AN0 to AN3
				100: AN4	100: AN4
				101: AN5	101: AN4 and AN5
				110: AN6	110: AN4 to AN6
				111: AN7	111: AN4 to AN7
				Switch input channels when ADST = 0.	

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

19.3.3 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. Set these bits only while A/D conversion is stopped (ADST = 0). 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: A/D conversion start by $\overline{\text{ADTRG}}$ pin
5 to 0	—	All 1	R/W	Reserved The initial value should not be changed.

19.4 Operation

The A/D converter operates by successive approximation with 10-bit resolution. It has two operating modes: single mode and scan mode. 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.

19.4.1 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 or an external trigger input.
2. When A/D conversion is completed, the result is transferred to the A/D data register corresponding to the channel.
3. On completion of A/D conversion, the ADF bit in ADCSR is set to 1. If the ADIE 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.

19.4.2 Scan Mode

In scan mode, A/D conversion is to be performed sequentially on the specified channels (max. four channels). Operations are as follows.

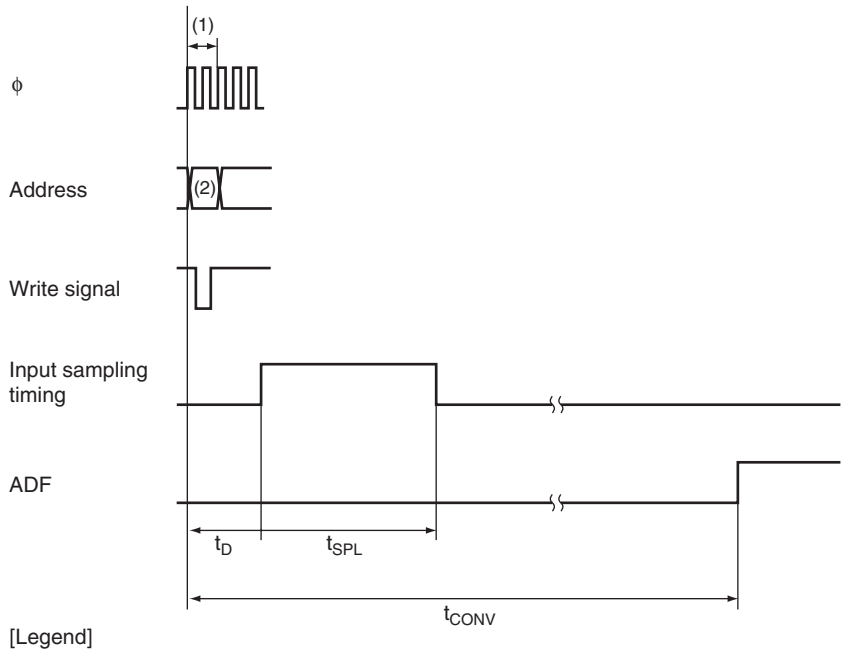
1. When the ADST bit in ADCSR is set to 1 by software or an external trigger input, A/D conversion starts on the first channel in the group (AN0 when the CH2 bit in ADCSR is 0, or AN4 when the CH2 bit in ADCSR is 1).
2. When A/D conversion for each channel is completed, the result is sequentially transferred to the A/D data register corresponding to each channel.
3. When conversion of all the selected channels is completed, the ADF bit in ADCSR is set to 1. If the ADIE bit is set to 1 at this time, an ADI interrupt is requested after A/D conversion ends. Conversion from the first channel in the group starts again.
4. 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.

19.4.3 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 19.2 shows the A/D conversion timing. Table 19.3 indicates the A/D conversion time.

As indicated in figure 19.2, 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 19.3.

In scan mode, the values shown in table 19.3 become those for the first conversion time. For the second and subsequent conversions, the conversion time is 266 states (fixed) when CKS = 0 and 134 states (fixed) when CKS = 1. Use the conversion time of 134 states only when the system clock (ϕ) is 16 MHz or lower.



[Legend]
(1) : ADCSR write cycle
(2) : ADCSR address
t_D : A/D conversion start delay
t_{SPL} : Input sampling time
t_{CONV} : A/D conversion time

Figure 19.2 A/D Conversion Timing

Table 19.3 A/D Conversion Time (Single Mode)

		CKS = 0			CKS = 1*		
Item	**Symbol**	**Min.**	**Typ.**	**Max.**	**Min.**	**Typ.**	**Max.**
A/D conversion start delay time	t_D	10	—	17	6	—	9
Input sampling time	t_{SPL}	—	63	—	—	31	—
A/D conversion time	t_{CONV}	259	—	266	131	—	134

Notes: Values in the table indicate the number of states.
* in the table indicates that the system clock (ϕ) is 16 MHz or lower.

19.4.4 External Trigger Input Timing

A/D conversion can be externally triggered. When the TRGS1 and TRGS0 bits are set to B′11 in ADCR, an external trigger is input to the $\overline{\text{ADTRG}}$ pin. The ADST bit in ADCSR is set to 1 at the falling edge of the $\overline{\text{ADTRG}}$ pin, thus starting A/D conversion. Other operations, in both single and scan modes, are the same as when the ADST bit has been set to 1 by software. Figure 19.3 shows the timing.

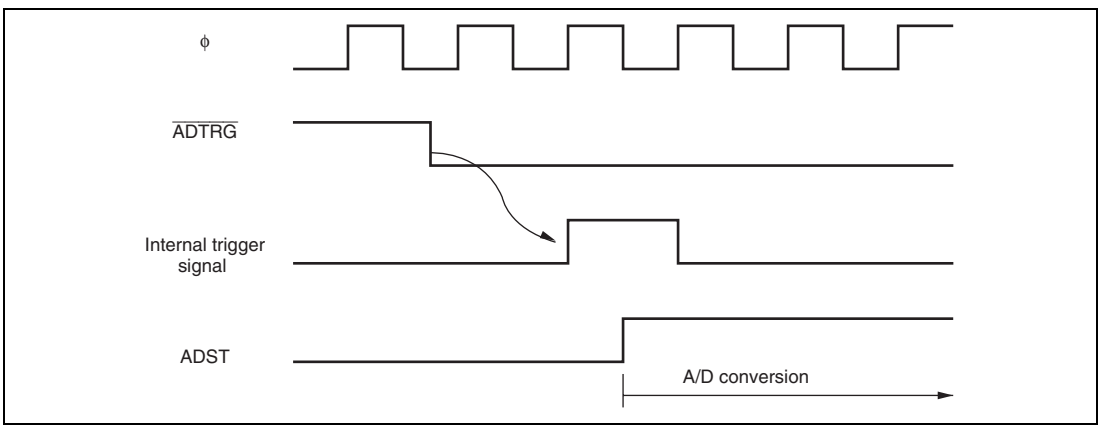


Figure 19.3 External Trigger Input Timing

19.5 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.

The ADI interrupt can be used to activate the on-chip DTC.

Table 19.4 A/D Converter Interrupt Source

Name	Interrupt Source	Interrupt Flag	DTC Activation
ADI	A/D conversion end	ADF	Enable

19.6 A/D Conversion Accuracy Definitions

This LSI's A/D conversion accuracy definitions are given below.

- Resolution

  The number of A/D converter digital output codes
- Quantization error

  The deviation inherent in the A/D converter, given by 1/2 LSB (see figure 19.4).
- 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 19.5).
- 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 19.5).
- 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 19.5).
- 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.

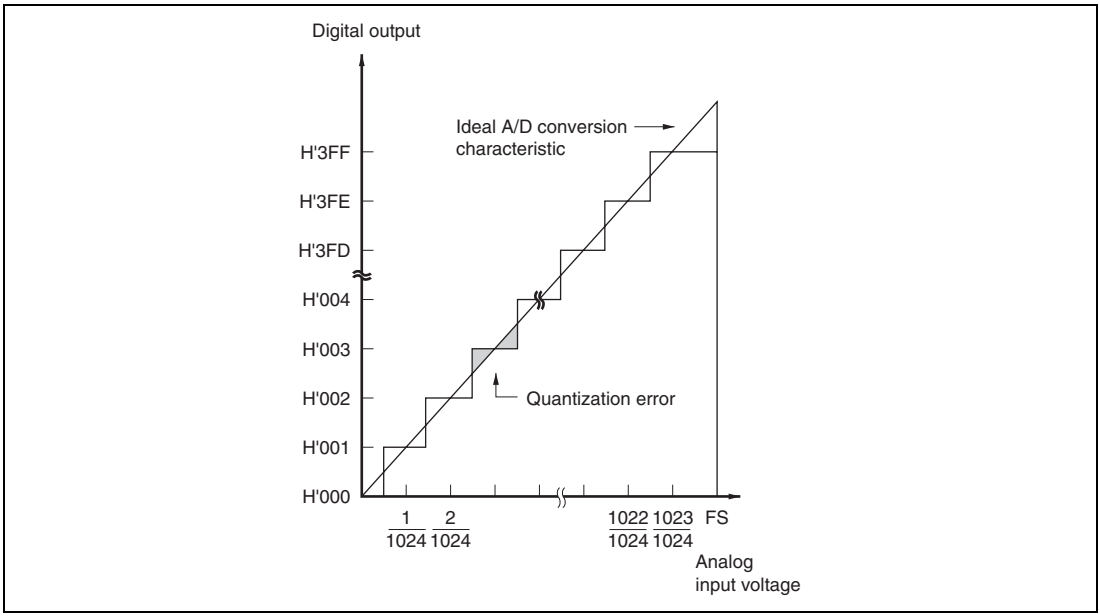


Figure 19.4 A/D Conversion Accuracy Definitions

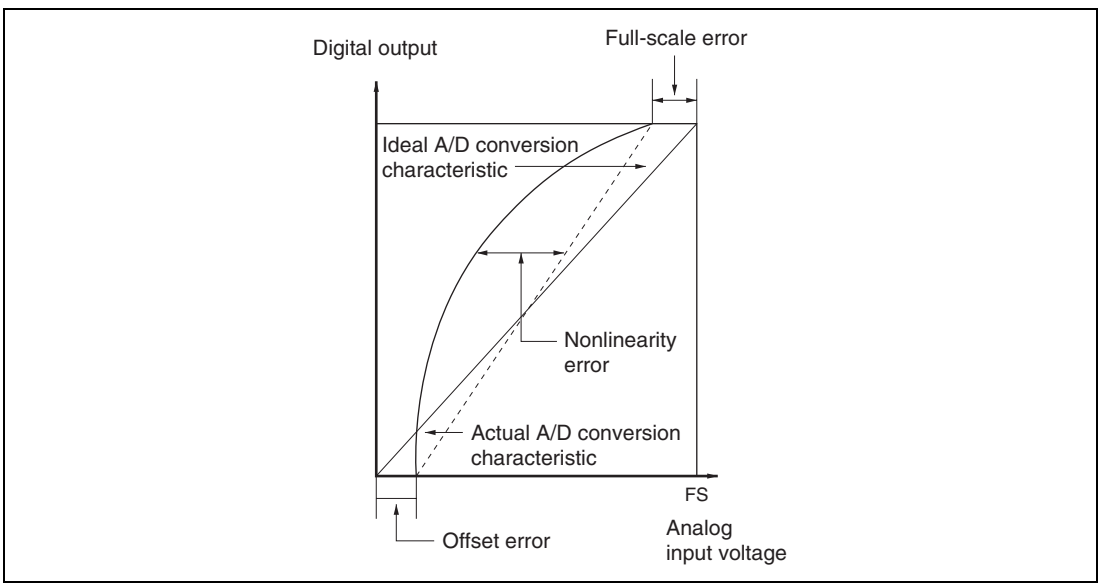


Figure 19.5 A/D Conversion Accuracy Definitions

19.7 Usage Notes

19.7.1 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 19.6). When converting a high-speed analog signal or converting in scan mode, a low-impedance buffer should be inserted.

19.7.2 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.

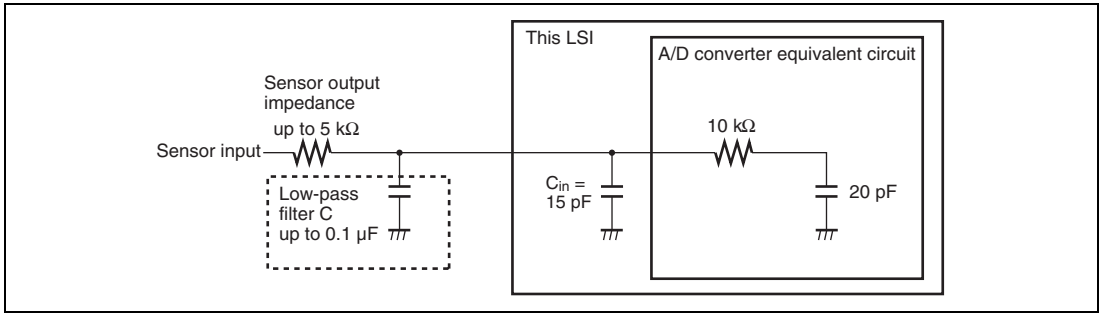


Figure 19.6 Example of Analog Input Circuit

19.7.3 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 pin ANn during A/D conversion should be in the range AVss ≤ ANn ≤ AVref (n = 0 to 7).
- Relation between AVcc, AVss and Vcc, Vss
  For the relationship between AVcc, AVss and Vcc, Vss, set AVss = Vss, but AVcc = Vcc is not necessary and which one is greater does not matter. Even when the A/D converter is not used, the AVcc and AVss pins must on no account be left open.
- AVref pin range
  The reference voltage of the AVref pin should be in the range AVref ≤ AVcc.

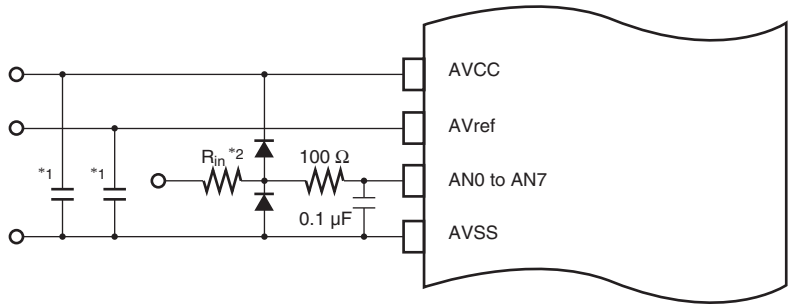
19.7.4 Notes on Board Design

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 AN7), 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.

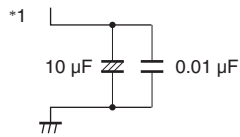
19.7.5 Notes on Noise Countermeasures

A protection circuit connected to prevent damage of the analog input pins (AN0 to AN7) 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 19.7. Also, the bypass capacitors connected to AVcc and AVref, and the filter capacitors connected to AN0 to AN7 must be connected to AVss.

If a filter capacitor is connected, the input currents at the analog input pins (AN0 to AN7) 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.

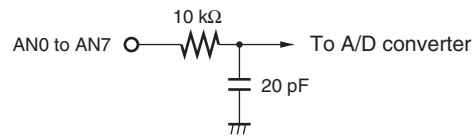


Notes: Values are reference values.



*2 R_{in}: Input impedance

Figure 19.7 Example of Analog Input Protection Circuit



Note: Values are reference values.

Figure 19.8 Analog Input Pin Equivalent Circuit

19.7.6 Module Stop Mode Setting

A/D converter operation can be enabled or disabled by the module stop control register. In the initial state, A/D converter operation is disabled. Access to A/D converter registers is enabled when module stop mode is cancelled. For details, see section 24, Power-Down Modes.

Section 20 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 21 Flash Memory (0.18-µm F-ZTAT Version)

The flash memory has the following features. Figure 21.1 shows a block diagram of the flash memory.

21.1 Features

- Size

Product Classification		ROM Size	ROM Addresses
H8S/2114R	R4F2114R	1 Mbyte	H'000000 to H'0FFFFF (mode 2) H'0000 to H'DFFF (mode 3)

- Two flash-memory MATs according to LSI initiation mode

  The on-chip flash memory has two memory spaces in the same address space (hereafter referred to as memory MATs). The mode setting at initiation determines which memory MAT is initiated first. The MAT can be switched by using the bank-switching method after initiation.

  — The user MAT is initiated at a power-on reset in user mode: 1 Mbyte

  — The user boot memory MAT is initiated at a power-on reset in user boot mode: 8 kbytes
- Programming/erasing interface by the download of on-chip program

  This LSI has a dedicated programming/erasing program. After downloading this program to the on-chip RAM, programming/erasing can be performed by setting the argument parameter.
- Programming/erasing time

  The flash memory programming time is 3 ms (typ) in 128-byte simultaneous programming, and approximately 25 µs per byte. The erasing time is 1000 ms (typ) per 64-kbyte block.
- Number of programming

  The number of flash memory programming can be up to 100 times at the minimum. (The value ranged from 1 to 100 is guaranteed.)
- Three on-board programming modes

  — Boot mode

  This mode is a program mode that uses an on-chip SCI interface. The user MAT and user boot MAT can be programmed. In this mode, the bit rate between the host and this LSI can be automatically adjusted.

  — User program mode

  The user MAT can be programmed by using the optional interface.

— User boot mode

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

- Programming/erasing protection

  Sets protection against flash memory programming/erasing via hardware, software, or error protection.

- Programmer mode

  This mode uses the PROM programmer. The user MAT and user boot MAT can be programmed.

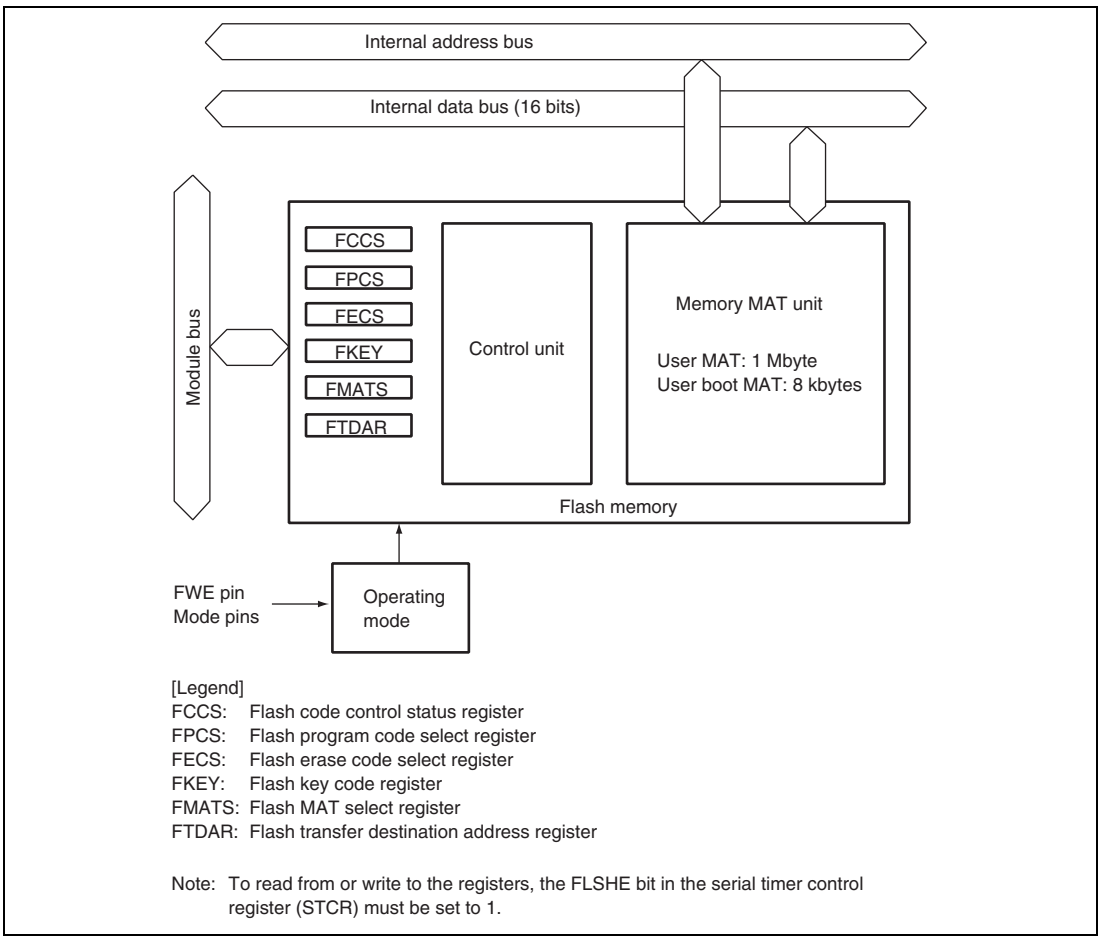


Figure 21.1 Block Diagram of Flash Memory

21.1.1 Mode Transitions

When each mode pin and the FWE pin are set in the reset state and the reset is started, this LSI enters each operating mode as shown in figure 21.2.

- Flash memory can be read in user mode, but cannot be programmed or erased.
- Flash memory can be read, programmed, or erased on the board only in user program mode, user boot mode, and boot mode.
- Flash memory can be read, programmed, or erased by means of the PROM programmer in programmer mode.

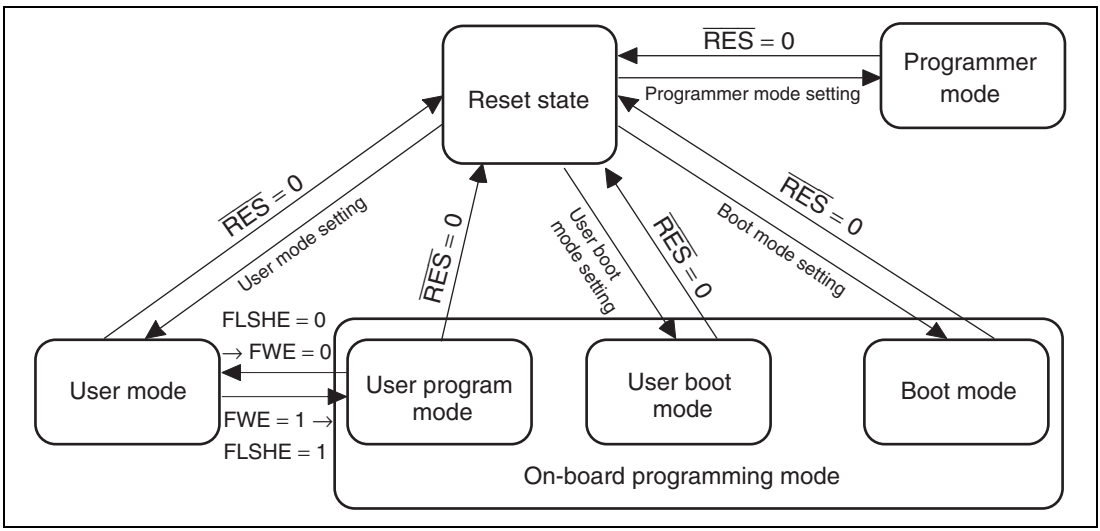


Figure 21.2 Mode Transition for Flash Memory

21.1.2 Mode Comparison

The comparison table of programming and erasing related items about boot mode, user program mode, user boot mode, and programmer mode is shown in table 21.1.

Table 21.1 Comparison of Programming Modes

	Boot Mode	User Program Mode	User Boot Mode	Programmer Mode
Programming/ erasing environment	On-board	On-board	On-board	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 optional device	Via optional device	Via programmer
Reset initiation MAT	Embedded program storage MAT	User MAT	User boot MAT*[2]	—
Transition to user mode	Changing mode setting and reset	Changing FLSHE bit and FWE pin	Changing mode setting and reset	—

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

- The user boot MAT can be programmed or erased only in boot mode and programmer mode.
- In boot mode, the user MAT and user boot MAT are totally erased. Then, the user MAT or user boot MAT can be programmed by means of commands. Note that the contents of the MAT cannot be read until this state.

  Boot mode can be used for programming only the user boot MAT and then programming the user MAT in user boot mode. Another way is to program only the user MAT since user boot mode is not used.
- In user boot mode, boot operation of the optional interface can be performed with mode pin settings different from those in user program mode.

21.1.3 Flash Memory MAT Configuration

This LSI's flash memory is configured by the 1-Mbyte 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.

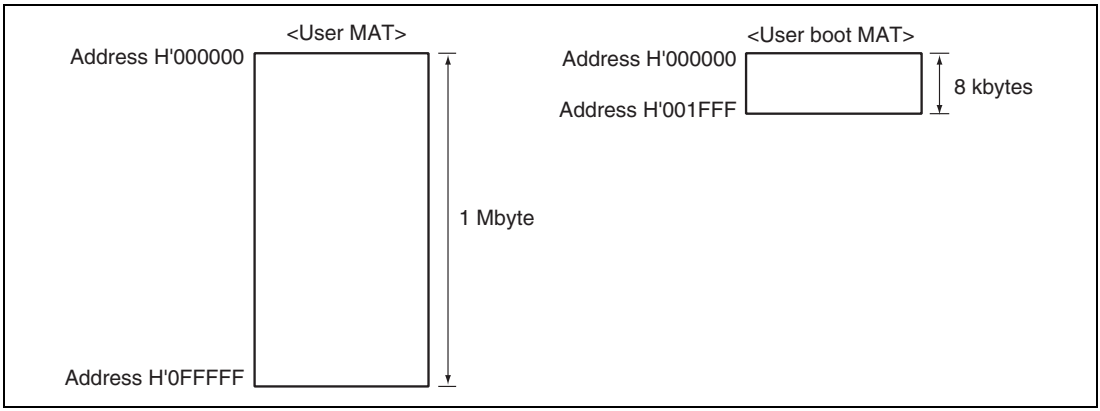


Figure 21.3 Flash Memory Configuration

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.

21.1.4 Block Division

The user MAT is divided into 64 kbytes (15 blocks), 32 kbytes (one block), and 4 kbytes (eight blocks) as shown in figure 21.4. The user MAT can be erased in this divided-block units by specifying the erase-block number of EB0 to EB23 when erasing.

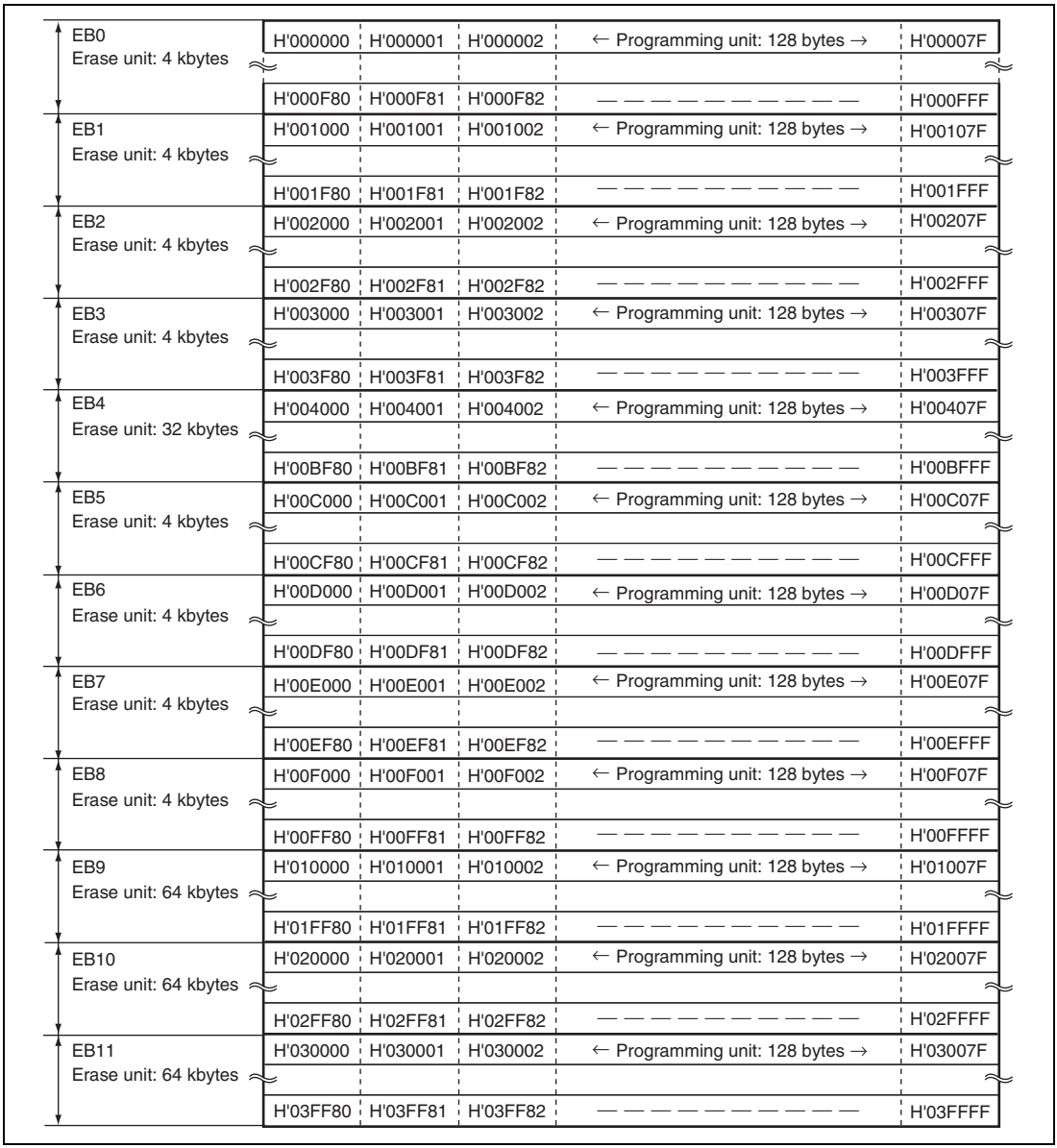


Figure 21.4 Block Division of User MAT (1)

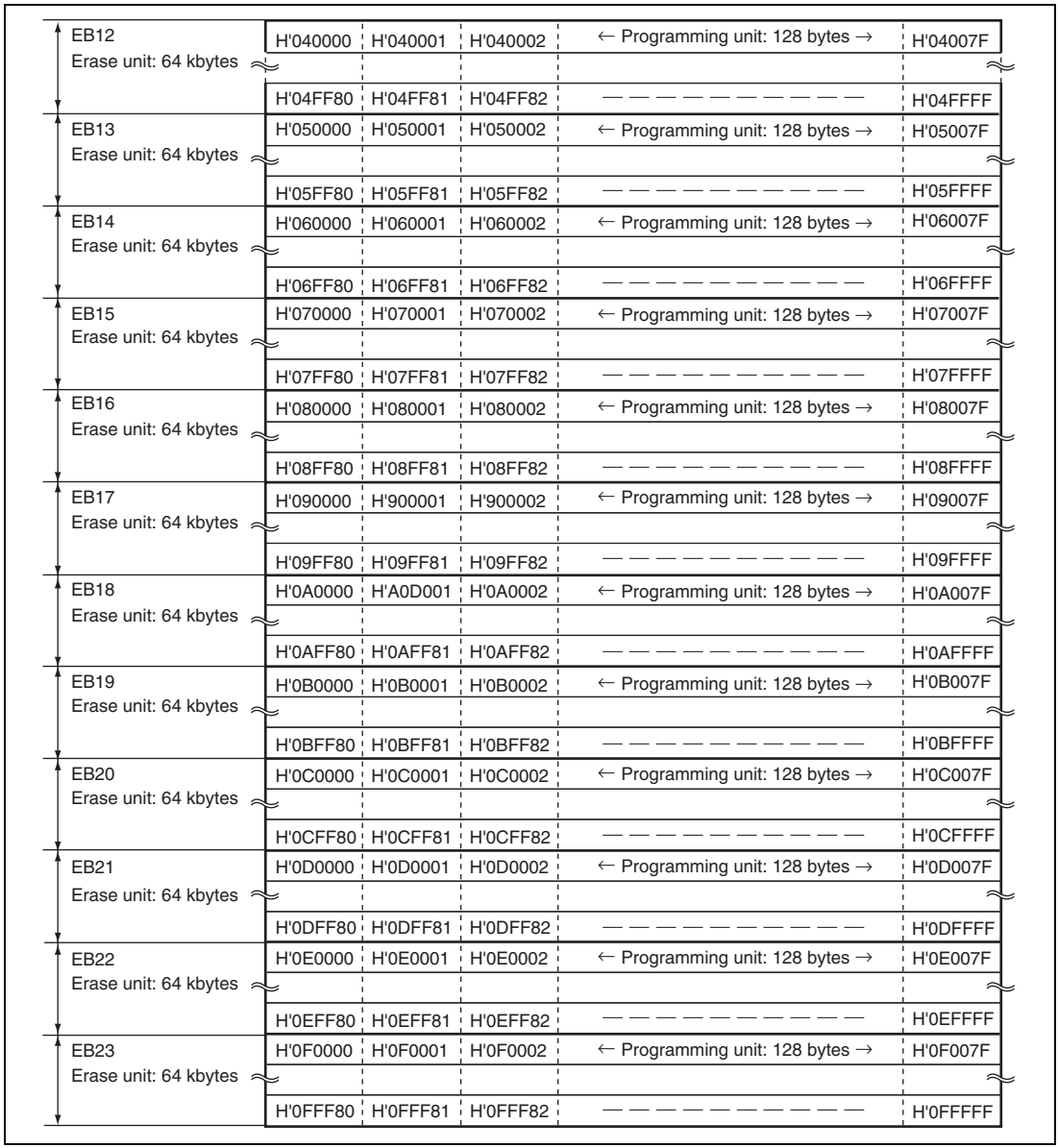


Figure 21.4 Block Division of User MAT (2)

21.1.5 Programming/Erasing Interface

Programming/erasing is executed by downloading the on-chip program to the on-chip RAM and specifying the program address/data and erase block by using the interface register/parameter.

The procedure program is made by the user in user program mode and user boot mode. An overview of the procedure is given as follows. For details, see section 21.4.2, User Program Mode.

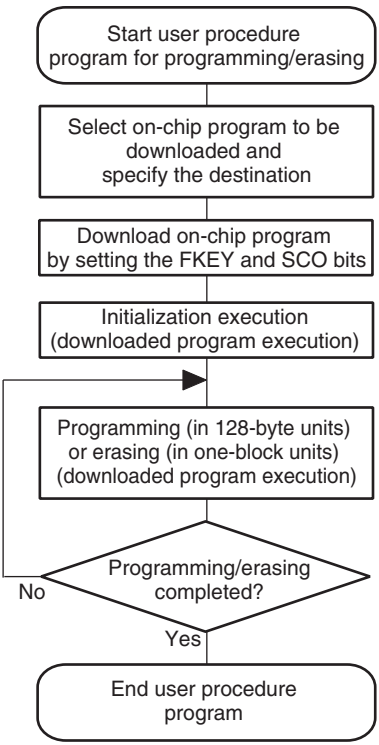


Figure 21.5 Overview of User Procedure Program

1. Selection of on-chip program to be downloaded

   For programming/erasing execution, set the FLSHE bit in STCR to 1 to make a transition to user program mode.

   This LSI has programming/erasing programs that can be downloaded to the on-chip RAM. The on-chip program to be downloaded is selected by setting the corresponding bits in the programming/erasing interface register. The address of the download destination 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), which are programming/erasing interface registers.

   The flash memory MAT is replaced with the embedded program storage MAT during downloading. Since the flash memory cannot be read during programming/erasing, the procedure program that executes download to completion of programming/erasing must be executed in a space other than flash memory (for example, on-chip RAM).

   Since the result of download is returned to the programming/erasing interface parameter, whether download has succeeded or not can be confirmed.

3. Initialization of programming/erasing

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

4. Execution of programming/erasing

   For programming/erasing execution, set the FLSHE bit in STCR and the FWE pin to 1 to make a transition to user program mode.

   The program data/programming destination address is specified in 128-byte units for programming. The block to be erased is specified in erase-block units for erasing.

   Make these specifications by using the programming/erasing interface parameter, and then initiate the on-chip program. The on-chip program is executed by using the JSR or BSR instruction to execute 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 must be disabled during programming and erasing. Interrupts must be masked within the user system.

5. Consecutive execution of programming/erasing

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

   Since the downloaded on-chip program remains in the on-chip RAM even after the processing ends, download and initialization are not required when the same processing is executed consecutively.

21.2 Input/Output Pins

Flash memory is controlled by the pins listed in table 21.2.

Table 21.2 Pin Configuration

Pin Name	Input/Output	Function
$\overline{\text{RES}}$	Input	Reset
FWE	Input	Flash memory programming/erasing enable pin
MD2	Input	Sets operating mode of this LSI
MD1	Input	Sets operating mode of this LSI
MD0	Input	Sets operating mode of this LSI
TxD1	Output	Serial transmit data output (used in boot mode)
RxD1	Input	Serial receive data input (used in boot mode)

21.3 Register Descriptions

The registers/parameters that control flash memory are shown below. To read from or write to these registers/parameters, the FLSHE bit in STCR must be set to 1. For details on STCR, see section 3.2.3, Serial Timer Control Register (STCR).

- Flash code control status register (FCCS)
- Flash program code select register (FPCS)
- Flash erase code select register (FECS)
- Flash key code register (FKEY)
- Flash MAT select register (FMATS)
- Flash transfer destination address register (FTDAR)
- Download pass/fail result (DPFR)
- Flash pass/fail result (FPFR)

- Flash multipurpose address area (FMPAR)
- Flash multipurpose data destination area (FMPDR)
- Flash erase block select (FEBS)
- Flash programming/erasing frequency control (FPEFEQ)

There are several operating modes for accessing flash memory, for example, read mode/program mode.

There are two memory MATs: user MAT and user boot MAT. The dedicated registers/parameters are allocated for each operating mode and MAT selection. The correspondence between operating modes and registers/parameters for use is shown in table 21.3.

Table 21.3 Register/Parameter and Target Mode

		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. The setting is required when programming or erasing the user MAT in user boot mode.
2. The setting may be required according to the combination of initiation mode and read target MAT.

21.3.1 Programming/Erasing Interface Registers

The programming/erasing interface registers are all 8-bit registers that can be accessed in bytes. These registers are initialized at a reset or in hardware standby mode.

- Flash Code Control Status Register (FCCS)

  FCCS is configured by bits which request monitoring of the FWE pin state and error occurrence during programming or erasing flash memory, and the download of an on-chip program.

Bit	Bit Name	Initial Value	R/W	Description
7	FWE	1/0	R	Flash Program Enable Monitors the signal level input to the FWE pin. 0: A low level signal is input to the FWE pin. (Hardware protection state) 1: A high level signal is input to the FWE pin.
6, 5	—	All 0	R/W	Reserved The initial value should not be changed.

Bit	Bit Name	Initial Value	R/W	Description
4	FLER	0	R	Flash Memory Error Indicates an error has occurred during programming or erasing flash memory. When this bit is set to 1, flash memory enters the error-protection state. In case this bit is set to 1, high voltage is applied to the internal flash memory. To reduce the damage to flash memory, the reset must be released after a reset period of 100 µs which is longer than normal. 0: Flash memory operates normally. Programming/erasing protection (error protection) for flash memory is invalid. [Clearing condition] • At a reset or in hardware standby mode 1: An error occurs during programming/erasing flash memory. Programming/erasing protection (error protection) for flash memory is valid. [Setting conditions] • When an interrupt, such as NMI, occurs during programming/erasing flash memory. • When flash memory is read during programming/erasing flash memory (including a vector read or an instruction fetch). • When the SLEEP instruction is executed during programming/erasing flash memory (including software standby mode) • When a bus master other than the CPU, such as the DTC or LPC, gets bus mastership during programming/erasing flash memory.
3 to 1	—	All 0	R/W	Reserved The initial value should not be changed.

Bit	Bit Name	Initial Value	R/W	Description
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/FECS is automatically downloaded in the on-chip RAM 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. Immediately after setting this bit to 1, four NOP instructions must be executed. Since this bit is cleared to 0 when download is completed, this bit cannot be read as 1. All interrupts must be disabled during downloading. Interrupts must be masked within the user system. 0: Download of the on-chip programming/erasing program to the on-chip RAM is not executed. [Clearing condition] When download is completed 1: Request to download the on-chip programming/erasing program to the on-chip RAM has occurred. [Setting conditions] When all of the following conditions are satisfied and this bit is set to 1 • H'A5 is written to FKEY • During execution in the on-chip RAM

Note: * This bit is a write only bit. This bit is always read as 0.

- Flash Program Code Select Register (FPCS)

  FPCS selects the on-chip programming program to be downloaded.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R/W	Reserved The initial value should not be changed.
0	PPVS	0	R/W	Program Pulse Verify Selects the programming program. 0: On-chip programming program is not selected. [Clearing condition] When transfer is completed 1: On-chip programming program is selected.

- Flash Erase Code Select Register (FECS)

  FECS selects the on-chip erasing program to be downloaded.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R/W	Reserved The initial value should not be changed.
0	EPVB	0	R/W	Erase Pulse Verify Block Selects the erasing program. 0: On-chip erasing program is not selected. [Clearing condition] When transfer is completed 1: On-chip erasing program is selected.

- Flash Key Code Register (FKEY)

  FKEY is for software protection that enables download of an on-chip program and programming/erasing of flash memory. Before setting the SCO bit to 1 to download an on-chip program or before executing the downloaded programming/erasing program, the key code must be written, otherwise the processing cannot be executed.

Bit	Bit Name	Initial Value	R/W	Description
7	K7	0	R/W	Key Code
6	K6	0	R/W	Only when H'A5 is written, writing to the SCO bit is valid. When a value other than H'A5 is written to FKEY, 1 cannot be set to the SCO bit. Therefore downloading to the on-chip RAM cannot be executed. Only when H'5A is written, programming/erasing can be executed. Even if the on-chip programming/erasing program is executed, the flash memory cannot be programmed or erased when a value other than H'5A is written to FKEY. H'A5: Writing to the SCO bit is enabled. (The SCO bit cannot be set by a value other than H'A5.) H'5A: Programming/erasing is enabled. (Software protection state is entered for a value other than H'5A.) H'00: Initial value
5	K5	0	R/W	
4	K4	0	R/W	
3	K3	0	R/W	
2	K2	0	R/W	
1	K1	0	R/W	
0	K0	0	R/W	

- 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. When the MAT is switched, follow section 21.6, 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 except for user boot mode (user MAT is selected) [Programmable condition] In the execution state in the on-chip RAM

Note: * Set to 1 in user boot mode, otherwise cleared to 0.

- Flash Transfer Destination Address Register (FTDAR)

  FTDAR specifies the on-chip RAM address where an on-chip program is downloaded. This register must be specified 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 the address specified by bits TDA6 to TDA0, which is the start address where an on-chip program is downloaded, is over the range. Whether or not the address specified by bits TDA6 to TDA0 is within the range of H'00 to H'03 is determined when an on-chip program is downloaded by setting the SCO bit in FCCS to 1. Make sure that this bit is cleared to 0 and the value specified by bits TDA6 to TDA0 is within the range of H'00 to H'03 before setting the SCO bit to 1. 0: The value specified by bits TDA6 to TDA0 is within the range. 1: The value specified by bits TDA6 to TDA0 is over the range (H'04 to H'7F) and download is stopped.
6	TDA6	0	R/W	Transfer Destination Address Specifies the start address where an on-chip program is downloaded. A value of H'00 can be specified as the download start address in the on-chip RAM. H'00: H'FFD080 is specified as the download start address. H'01: H'FFD880 is specified as the download start address. H'02: H'FFE080 is specified as the download start address. H'03 to H'7F: Setting prohibited. Specifying this value sets the TDER bit to 1 during downloading and stops the download.
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	

21.3.2 Programming/Erasing Interface Parameters

The programming/erasing interface parameters specify the operating frequency, storage place for program data, programming destination address, and erase block and exchanges the processing result for the downloaded on-chip program. These parameters use the CPU general registers (ER0 and ER1) or the on-chip RAM area. The initial value is undefined at a reset or in hardware standby mode.

In download, initialization, or execution of the on-chip program, registers of the CPU except for R0L are stored. The return value of the processing result is written in R0L. Since the stack area is used for storing the registers except for R0L, the stack area must be saved at the processing start. (A maximum size of a stack area to be used is 128 bytes.)

The programming/erasing interface parameters is used for the following four functions:

1. Download control
2. Initialization before programming or erasing
3. Programming
4. Erasing

These items use different parameters. The correspondence table is shown in table 21.4.

The meaning of bits in FPFR varies in each processing: initialization, programming, or erasure. For details, see descriptions of FPFR for each processing.

Table 21.4 Parameters and Target Modes

Parameter Name	Abbreviation	Download	Initialization	Programming	Erasure	R/W	Initial Value	Allocation
Download pass/fail result	DPFR	O	—	—	—	R/W	Undefined	On-chip RAM*
Flash pass/fail result	FPFR	—	O	O	O	R/W	Undefined	R0L of CPU
Flash programming/ erasing frequency control	FPEFEQ	—	O	—	—	R/W	Undefined	ER0 of CPU
Flash multipurpose address area	FMPAR	—	—	O	—	R/W	Undefined	ER1 of CPU
Flash multipurpose data destination area	FMPDR	—	—	O	—	R/W	Undefined	ER0 of CPU
Flash erase block select	FEBS	—	—	—	O	R/W	Undefined	R0L of CPU

Note: * A single byte of the download start address specified by FTDAR.

(1) Download Control

The on-chip program is automatically downloaded by setting the SCO bit to 1. The on-chip RAM area where the program is to be downloaded is the 2-kbyte area starting from the address specified by FTDAR.

Download control is set by the programming/erasing interface registers, and the DPFR parameter indicates the return value.

(a) Download pass/fail result parameter (DPFR: single byte of start address specified by FTDAR)

This parameter indicates the return value of the download result. The value of this parameter can be used to determine if downloading was executed or not. Since confirmation whether the SCO bit is set to 1 or not is difficult, certain determination must be gained by setting a value other than the return value of download (for example, H'FF) to the single byte of the start address specified by FTDAR before download starts (before setting the SCO bit to 1).

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	—	—	Unused The return value is 0.
2	SS	—	R/W	Source Select Error Detect Only one type can be specified for the on-chip program that can be downloaded. When more than two types of programs are selected, the program is not selected, or the program is selected without mapping, an error occurs. 0: Download program selection is normal 1: Download error has occurred (multi-selection or program which is not mapped is selected)
1	FK	—	R/W	Flash Key Register Error Detect Returns the check result whether the FKEY value is set to H'A5. 0: FKEY setting is normal (FKEY = H'A5) 1: FKEY setting is abnormal (FKEY = value other than H'A5)
0	SF	—	R/W	Success/Fail Returns the result whether download has ended normally or not. Determines the result whether the program was correctly downloaded to the on-chip RAM by way of the confirming reading of it. 0: Download to on-chip program has ended normally (no error) 1: Download to on-chip program has ended abnormally (error occurred)

(2) Programming/Erasing Initialization

The on-chip programming/erasing program to be downloaded includes the initialization program.

A pulse of the specified width must be applied when programming or erasing. The specified pulse width is made by the method in which a wait loop is configured by CPU instructions. The operating frequency of the CPU must be set too.

The initialization program is used to set the above values as parameters of the programming/erasing program that was downloaded.

(a) Flash programming/erasing frequency control parameter (FPEFEQ: general register ER0 of CPU)

This parameter sets the operating frequency of the CPU. The settable range of the operating frequency in this LSI is 4 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 with the following procedure. 1. The operating frequency shown in MHz units must be rounded off to two decimals. 2. The value multiplied by 100 is converted to the hexadecimal numeral and written to the FPEFEQ parameter (general register ER0). For example, when the operating frequency of the CPU is 20.000 MHz, the setting value is as follows: 1. 20.000 is rounded off to two decimals, thus becoming 20.00. 2. The formula of 20.00 × 100 = 2000 is converted to the hexadecimal numeral and H'07D0 is set to ER0.

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

This parameter indicates the return value of the initialization result.

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	—	—	Unused The return value is 0.
1	FQ	—	R/W	Frequency Error Detect Returns the check result whether the specified CPU operating frequency is in the range of the supported operating frequency. 0: Setting of operating frequency is normal 1: Setting of operating frequency is abnormal
0	SF	—	R/W	Success/Fail Indicates whether initialization has ended normally or not. 0: Initialization has ended normally (no error) 1: Initialization has ended abnormally (error occurred)

(3) Programming Execution

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

1. The start address of the programming destination on the user MAT must be set in general register ER1. This parameter is called the flash multipurpose address area parameter (FMPAR).

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

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

   The start address of the area in which the prepared program data is stored 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 21.4.2, User Program Mode.

(a) Flash multipurpose address area parameter (FMPAR: general register ER1 of CPU)

This parameter stores the start address of the programming destination on the user MAT.

When the address in an area other than the flash memory space is set, an error occurs.

The start address of the programming destination must be at the 128-byte boundary. If this boundary condition is not satisfied, an error occurs. The error occurrence is indicated by the WA bit (bit 1) in the FPFR parameter.

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 programming start address becomes a 128-byte boundary and the MOA6 to MOA0 bits are always 0.

(b) Flash multipurpose data destination area parameter (FMPDR: general register ER0 of CPU)

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

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.

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

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

Bit	Bit Name	Initial Value	R/W	Description
7	—	—	—	Unused The return value is 0.
6	MD	—	R/W	Programming Mode Related Setting Error Detect Returns the check result whether a high level signal is input to the FWE pin or whether the error-protection state is not entered. When a low-level signal is input to the FWE pin or the error-protection state is entered, 1 is written to this bit. These states can be confirmed with the FWE and FLER bits in FCCS. For conditions to enter the error-protection state, see section 21.5.3, Error Protection. 0: FWE and FLER settings are normal (FWE = 1, FLER = 0) 1: Programming cannot be performed because FWE = 0 or FLER = 1
5	EE	—	R/W	Programming Execution Error Detect 1 is returned 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 is partially rewritten. In this case, after removing the error factor, erase the user MAT. If FMATS is set to H'AA and the user boot MAT is selected, an error occurs when programming is performed. In this case, both the user MAT and user boot MAT are not rewritten. Programming of the user boot MAT should be performed in boot mode or programmer mode. 0: Programming has ended normally 1: Programming has ended abnormally and programming result is not guaranteed

Bit	Bit Name	Initial Value	R/W	Description
4	FK	—	R/W	Flash Key Register Error Detect Returns the check result of the FKEY value before the start of the programming processing. 0: FKEY setting is normal (FKEY = H'5A) 1: FKEY setting is abnormal (FKEY = value other than H'5A)
3	—	—	—	Unused The return value is 0.
2	WD	—	R/W	Write Data Address Detect When an address in the flash memory area is specified as the start address of the storage destination of the program data, an error occurs. 0: Setting of program data address is normal 1: Setting of program data address 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. • When the specified programming destination address is in an area other than flash memory • When the specified address is not at a 128-byte boundary (the lower eight bits of the address are other than H'00 or H'80) 0: Setting of programming destination address is normal 1: Setting of programming destination address is abnormal
0	SF	—	R/W	Success/Fail Indicates whether the programming processing has ended normally or not. 0: Programming has ended normally (no error) 1: Programming has ended abnormally (error occurred)

(4) Erasure Execution

When flash memory is erased, the erase-block number on the user MAT must be passed to the erasing program that is downloaded. This is set to the FEBS parameter (general register ER0).

One block is specified from the block numbers 0 to 23.

For details on the erasing procedure, see section 21.4.2, User Program Mode.

(a) Flash erase block select parameter (FEBS: general register ER0 of CPU)

This parameter specifies the erase-block number. Several block numbers cannot be selected at one time.

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	—	—	Unused These bits should be cleared to 0.
7	EB7	—	R/W	Erase Block
6	EB6	—	R/W	These bits set the erase-block number in the range from 0 to 23. 0 corresponds to the EB0 block and 23 corresponds to the EB23 block. An error occurs when a number other than 0 to 23 (H'00 to H'17) is set.
5	EB5	—	R/W	
4	EB4	—	R/W	
3	EB3	—	R/W	
2	EB2	—	R/W	
1	EB1	—	R/W	
0	EB0	—	R/W	

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

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

Bit	Bit Name	Initial Value	R/W	Description
7	—	—	—	Unused The return value is 0.
6	MD	—	R/W	Erasing Mode Related Setting Error Detect Returns the check result whether a high level signal is input to the FWE pin or whether the error-protection state is not entered. When a low-level signal is input to the FWE pin or the error-protection state is entered, 1 is written to this bit. These states can be confirmed with the FWE and FLER bits in FCCS. For conditions to enter the error-protection state, see section 21.5.3, Error Protection. 0: FWE and FLER settings are normal (FWE = 1, FLER = 0) 1: Erasing cannot be performed because FWE = 0 or FLER = 1
5	EE	—	R/W	Erasure Execution Error Detect 1 is returned to this bit when the user MAT could not be erased or when flash-memory related register settings are partially changed. If this bit is set to 1, there is a high possibility that the user MAT is partially erased. In this case, after removing the error factor, erase the user MAT. If FMATS is set to H'AA and the user boot MAT is selected, an error occurs when erasure is performed. In this case, both the user MAT and user boot MAT are not erased. Erasing of the user boot MAT should be performed in boot mode or programmer mode. 0: Erasure has ended normally 1: Erasure has ended abnormally and erasure result is not guaranteed
4	FK	—	R/W	Flash Key Register Error Detect Returns the check result of the FKEY value before the start of the erasing processing. 0: FKEY setting is normal (FKEY = H'5A) 1: FKEY setting is abnormal (FKEY = value other than H'5A)

Bit	Bit Name	Initial Value	R/W	Description
3	EB	—	R/W	Erase Block Select Error Detect Returns the check result whether the specified erase-block number is in the block range of the user MAT. 0: Setting of erase-block number is normal 1: Setting of erase-block number is abnormal
2, 1	—	—	—	Unused The return value is 0.
0	SF	—	R/W	Success/Fail Indicates whether the erasing processing has ended normally or not. 0: Erasure has ended normally (no error) 1: Erasure has ended abnormally (error occurred)

21.4 On-Board Programming

When the pins are set to on-board programming mode and the reset start is executed, a transition is made to an on-board programming state in which the on-chip flash memory can be programmed/erased. On-board programming mode has three operating modes: boot mode, user program mode, and user boot mode.

For details on the pin setting for entering each mode, see table 21.5. For details of the state transition of each mode for flash memory, see figure 21.2.

Table 21.5 On-Board Programming Mode Setting

Mode Setting	FWE	MD2	MD1	MD0	NMI
Boot mode	1	1	0	0	1
User program mode	1*	0	1	0/1	0/1
User boot mode	1	1	0	0	0

Note: * Before downloading a programming/erasing program, the FLSHE bit must be set to 1 to make a transition to user program mode.

21.4.1 Boot Mode

Boot mode executes programming/erasing of the user MAT and user boot MAT by means of the control commands and program data transmitted from the host via the on-chip SCI. The tool for transmitting the control commands, and program data must be prepared in the host. The SCI communication mode is set to asynchronous mode. When reset start is executed after this LSI's pins have been set to boot mode, the boot program built in the microcomputer beforehand is initiated. After the SCI bit rate is automatically adjusted, communication with the host is executed by means of control commands.

A system configuration diagram in boot mode is shown in figure 21.6. For details on the pin settings in boot mode, see table 21.5. The NMI and other interrupts are ignored in boot mode. However, the NMI and other interrupts should be disabled within the user system.

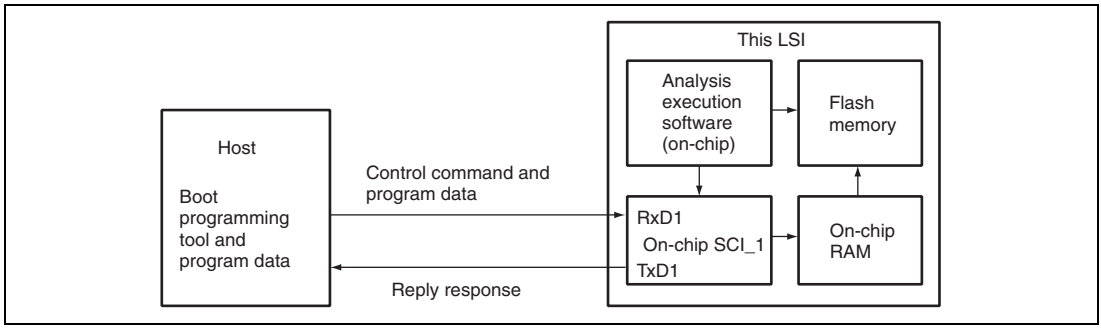


Figure 21.6 System Configuration in Boot Mode

(1) SCI Interface Setting by Host

When boot mode is initiated, this LSI measures the low period of asynchronous SCI communication data (H'00) which is transmitted consecutively from the host. The SCI transmit/receive format is set to 8-bit data, 1 stop bit, and no parity. This LSI calculates the bit rate of transmission by the host by means of the measured low period and transmits the bit adjustment end sign (1 byte of H'00) to the host. The host must confirm that this bit adjustment end sign (H'00) has been received normally and then transmits 1 byte of H'55 to this LSI. When reception has not been executed normally, boot mode is initiated again (reset) and the operation described above must be performed. The bit rates of the host and this LSI do not match due to the bit rate of transmission by the host and the system clock frequency of this LSI. To operate the SCI normally, the transfer bit rate of the host must be set to 4,800 bps, 9,600 bps, or 19,200 bps.

The system clock frequency, which can automatically adjust the transfer bit rate of the host and the bit rate of this LSI, is shown in table 21.6. Boot mode must be initiated in the range of this system clock.

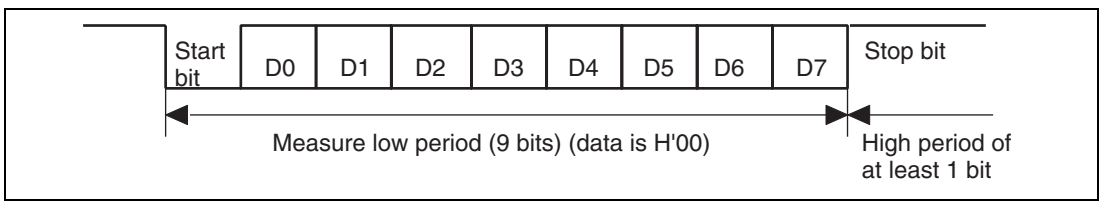


Figure 21.7 Automatic-Bit-Rate Adjustment Operation of SCI

Table 21.6 System Clock Frequency for Automatic-Bit-Rate Adjustment by This LSI

Bit Rate of Host	System Clock Frequency for Automatic-Bit-Rate Adjustment by This LSI
4,800 bps	4 to 20 MHz
9,600 bps	4 to 20 MHz
19,200 bps	8 to 20 MHz

(2) State Transition Diagram

The overview of the state transition diagram after boot mode is initiated is shown in figure 21.8.

1. Bit rate adjustment
   After boot mode is initiated, the bit rate of the SCI interface is adjusted with that of the host.
2. Waiting for inquiry set command
   For inquiries about the user MAT size and configuration, MAT start address, and support state, the required information is transmitted to the host.
3. Automatic erasure of all user MATs and user boot MATs
   After inquiries have finished, all user MATs and user boot MATs are automatically erased.
4. Waiting for programming/erasing command
   - When the program preparation notice is received, the state for waiting for program data is entered. The programming start address and program data must be transmitted following the programming command. When programming is finished, the programming start address must be set to H'FFFFFFFF and transmitted. Then the state of program data wait is returned to the state of programming/erasing command wait.
   - When the erasure preparation notice is received, the state for waiting for erase-block data is entered. The erase-block number must be transmitted following the erasing command. When the erasure is finished, the erase-block number must be set to H'FF and transmitted. Then the state of erase-block data wait is returned to the state of programming/erasing command wait. This erasing operation should be used in a case where after programming has been executed in boot mode, a specific block is to be reprogrammed without a reset start. When programming can be executed by only one operation, since all blocks are erased before entering the state for waiting for a programming/erasing/other command, the erasing operation is not required.
   - There are many commands other than programming/erasing. For example, sum check, blank check (erasure check), and memory read of the user MAT and user boot MAT, and acquisition of current status information.

Note that memory read of the user MAT or user boot MAT can only read out the programmed data after all user MATs or user boot MATs have been automatically erased.

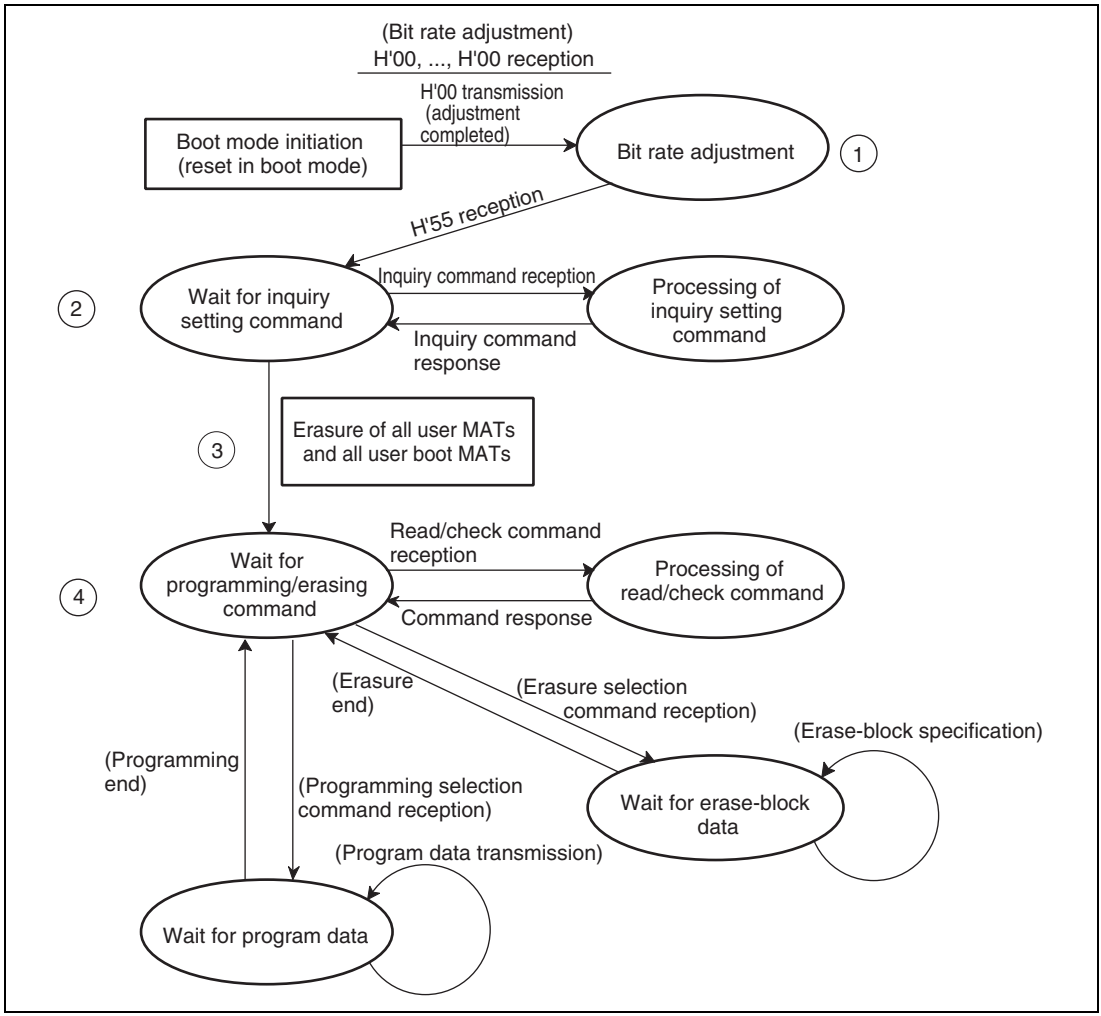


Figure 21.8 Overview of Boot Mode State Transition Diagram

21.4.2 User Program Mode

The user MAT can be programmed/erased in user program mode. (The user boot MAT cannot be programmed/erased.)

Programming/erasing is executed by downloading the program built in the microcomputer beforehand.

The programming/erasing overview flow is shown in figure 21.9.

High voltage is applied to internal flash memory during the programming/erasing processing. Therefore, a transition to the reset state or hardware standby mode must not be made. Doing so may damage and destroy flash memory. If a reset is executed accidentally, the reset must be released after a reset input period of 100 µs which is longer than normal.

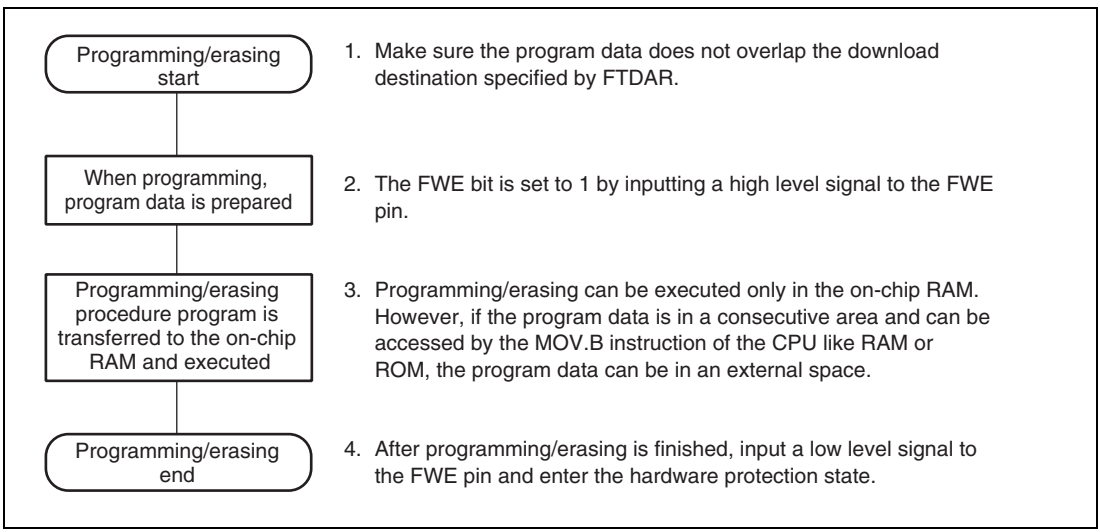


Figure 21.9 Programming/Erasing Overview Flow

(1) On-Chip RAM Address Map when Programming/Erasing is Executed

Part of the procedure program that is made by the user, like the download request, programming/erasing procedure, and determination of the result, must be executed in the on-chip RAM. The on-chip program that is to be downloaded is all in the on-chip RAM. Note that areas in the on-chip RAM must be controlled so that these parts do not overlap.

Figure 21.10 shows the area where a program is downloaded.

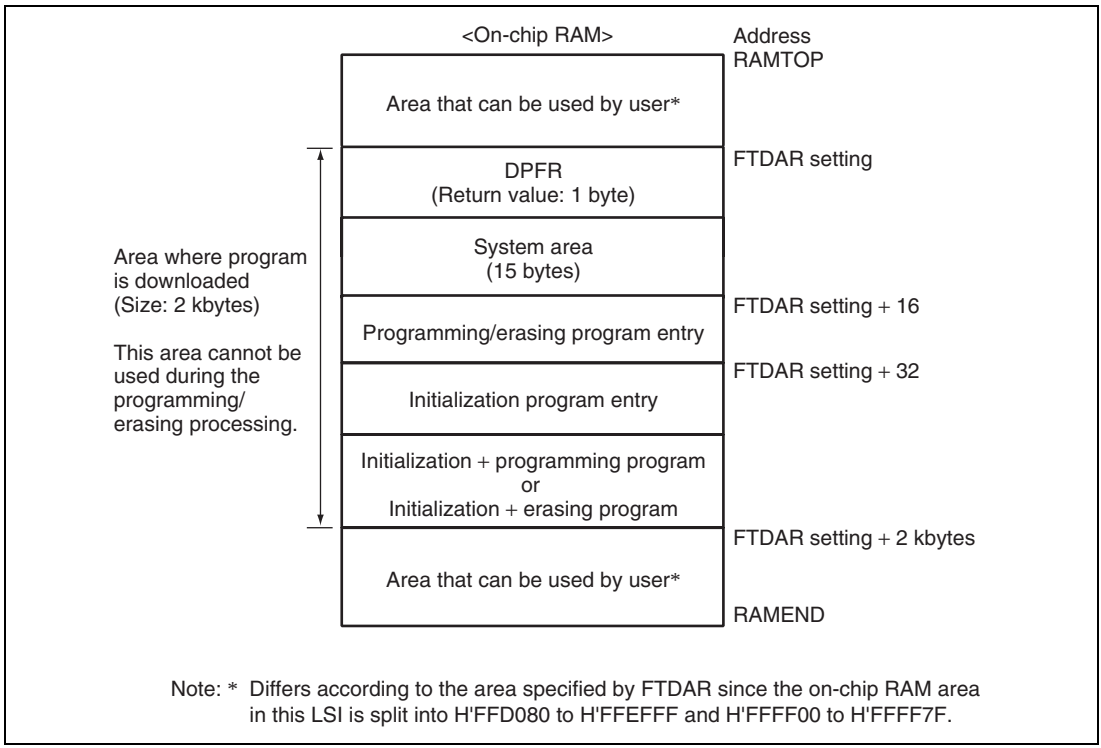


Figure 21.10 RAM Map when Programming/Erasing is Executed

(2) Programming Procedure in User Program Mode

The procedures for download, initialization, and programming are shown in figure 21.11.

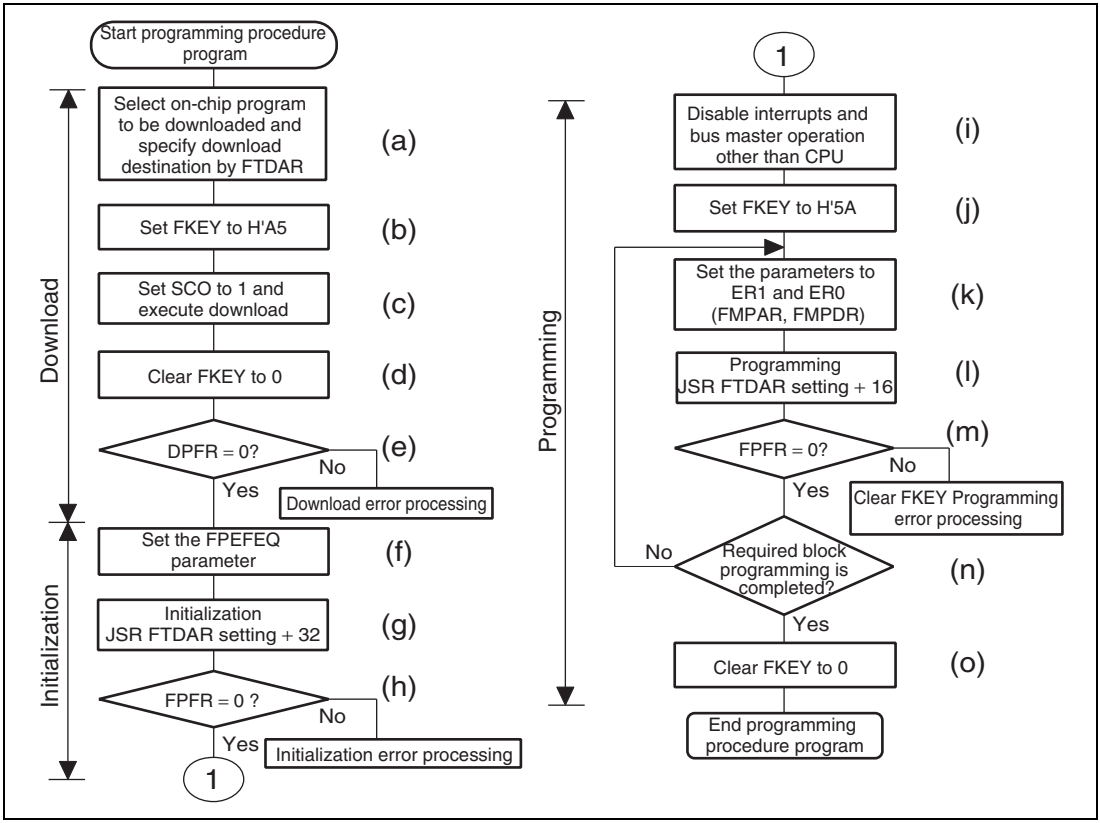


Figure 21.11 Programming Procedure

The procedure program must be executed in an area other than the flash memory to be programmed. Especially the part where the SCO bit in FCCS is set to 1 for downloading must be executed in the on-chip RAM.

The area that can be executed in the steps of the user procedure program (on-chip RAM and user MAT) is shown in section 21.4.4, Storable Areas for Procedure Program and Program Data.

The following description assumes the area to be programmed on the user MAT is erased and program data is prepared in the consecutive area. When erasing has not been done yet, execute erasing before writing.

128-byte programming is performed in one programming processing. To program more than 128 bytes, update the programming destination address/program data parameter in 128-byte units and repeat programming.

When less than 128 bytes of programming is performed, the program data must amount to 128 bytes by filling in invalid data. If the invalid data to be added is H'FF, the programming processing time can be shortened.

(a) Select the on-chip program to be downloaded and specify a 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 set, download is not performed and a download error is returned to the SS bit in DPFR. The start address of the download destination is specified by FTDAR.

(b) Write H'A5 in FKEY

If H'A5 is not written to FKEY for protection, 1 cannot be set to the SCO bit for a download request.

(c) Set the SCO bit in FCCS to 1 to execute download.

To set 1 to the SCO bit, the following conditions must be satisfied.

- H'A5 is written to FKEY.
- The SCO bit writing is executed in the on-chip RAM.

When the SCO bit is set to 1, download is started automatically. When execution returns to the user procedure program, the SCO bit is already cleared to 0. Therefore, the SCO bit cannot be confirmed to be 1 in the user procedure program.

The download result can be confirmed only by the return value of DPFR. To prevent incorrect determination, before the SCO bit is set to 1, set the single byte of the on-chip RAM start address (to be used as the DPFR parameter) specified by FTDAR to a value (e.g. H'FF) other than the return value.

When download is executed, particular interrupt processing, which is accompanied by bank switchover as described below, is performed as a microcomputer internal processing. Execute four NOP instructions immediately after the instruction that sets the SCO bit to 1.

- The user MAT space is switched to the embedded program storage MAT.
- After the selection condition of the download program and the FTDAR address setting are checked, the transfer processing to the on-chip RAM specified by FTDAR is executed.
- The SCO bit in FPCS, FECS, and FCCS is cleared to 0.
- The return value is set to the DPFR parameter.
- After the embedded program storage MAT is returned to the user MAT space, execution returns to the user procedure program.
- In the download processing, the values of CPU general registers are retained.
- In the download processing, all interrupts are not accepted. However, interrupt requests except for NMI are held. Therefore, when execution returns to the user procedure program, the interrupts will occur.
- When the level-detection interrupt requests are to be held, interrupts must be input until the download is ended.
- When hardware standby mode is entered during the download processing, normal download to the on-chip RAM cannot be guaranteed. Therefore, download must be executed again.
- Since a stack area of 128 bytes at the maximum is used, the stack area must be allocated before setting the SCO bit to 1.
- If a flash memory access by the DTC is requested during downloading, the operation cannot be guaranteed. Therefore, access by the DTC must not occur.

(d) Clear FKEY to H'00 for protection.

(e) Check the value of the DPFR parameter to confirm the download result.

- Check the value of the DPFR parameter (single byte of start address of the download destination specified by FTDAR). If the value 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 before downloading (e.g. H'FF), the address setting 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 before downloading, check the SS bit and FK bit in the DPFR parameter to ensure that the download program selection and FKEY setting were normal, respectively.

(f) Set the operating frequency to the FPEFEQ parameter for initialization.

The current frequency of the CPU clock is set to the FPEFEQ parameter (general register ER0).

The settable range of the FPEFEQ parameter is 4 to 20 MHz. When the frequency is set out of this range, an error is returned to the FPFR parameter of the initialization program and initialization is not performed. For details on the frequency setting, see the description in 21.3.2 (2) (a), Flash programming/erasing frequency control parameter (FPEFEQ).

(g) Initialization

When a programming program is downloaded, the initialization program is also downloaded to the on-chip RAM. There is an entry point for the initialization program in the area from the start address of a download destination specified by FTDAR + 32 bytes. The subroutine is called and initialization is executed 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 saved in the initialization program.
- R0L is a return value of the FPFR parameter.
- Since the stack area is used in the initialization program, a 128-byte stack area at the maximum must be allocated in RAM.
- Interrupts can be accepted during the execution of the initialization program. Note however that the program storage area and stack area in the on-chip RAM, and register values must not be rewritten.

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

(i) All interrupts and the use of a bus master other than the CPU are prohibited.

The stipulated voltage is applied for the stipulated time when programming or erasing. If interrupts occur or a bus master other than the CPU gets the bus during this period, a voltage pulse exceeding the regulation may be applied, thus damaging flash memory. Accordingly, interrupts must be disabled and a bus master other than the CPU, such as the DTC or LPC, must not be allowed.

To disable interrupts, bit 7 (I) in the condition code register (CCR) of the CPU should be set to B'1 in interrupt control mode 0, or bits 7 and 6 (I and UI) in the condition code register (CCR) of the

CPU should be set to B'11 in interrupt control mode 1. This enables interrupts other than NMI to be held and not executed.

The NMI interrupt must be masked within the user system.

The interrupts that are held must be executed after all programming processings.

When a bus master other than the CPU, such as the DTC or LPC, acquires the bus, the error-protection state is entered. Therefore, the DTC or LPC acquiring the bus must also be prohibited.

(j) Set H'5A in FKEY and prepare the user MAT for programming.

(k) Set the parameters required for programming.

The start address of the programming destination of the user MAT (FMPAR) is set to general register ER1, and the start address of the program data area (FMPDR) is set to general register ER0.

- Example of FMPAR setting

  FMPAR specifies the programming destination address. When an address other than one in the user MAT area is specified, even if the programming program is executed, programming is not executed and an error is returned to the return value parameter FPFR. Since the programming unit is 128 bytes, the lower eight bits of the address must be at the 128-byte boundary of H'00 or H'80.
- Example of FMPDR setting

  When the storage destination of the program data is flash memory, even if the programming execution 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 before programming is executed.

(l) Programming

There is an entry point for the programming program in the area from the start address of a download destination specified by FTDAR + 16 bytes. The subroutine is called and programming is executed by using the following steps.

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

- The general registers other than R0L are saved in the programming program.
- R0L is a return value of the FPFR parameter.
- Since the stack area is used in the programming program, a 128-byte stack area at the maximum must be allocated in RAM.

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

(n) Determine whether programming of the necessary data has finished.

If more than 128 bytes of data are to be programmed, specify FMPAR and FMPDR in 128-byte units, and repeat steps (l) to (n). Increment the programming destination address by 128 bytes and update the programming data pointer correctly. If an address that has already been programmed is written to again, not only will a programming error occur, but also flash memory will be damaged.

(o) After programming finishes, clear FKEY and specify software protection.

If this LSI is restarted by a reset immediately after user MAT programming has finished, secure a reset period (period of $\overline{RES}$ = 0) of 100 µs which is longer than normal.

(3) Erasing Procedure in User Program Mode

The procedures for download, initialization, and erasing are shown in figure 21.12.

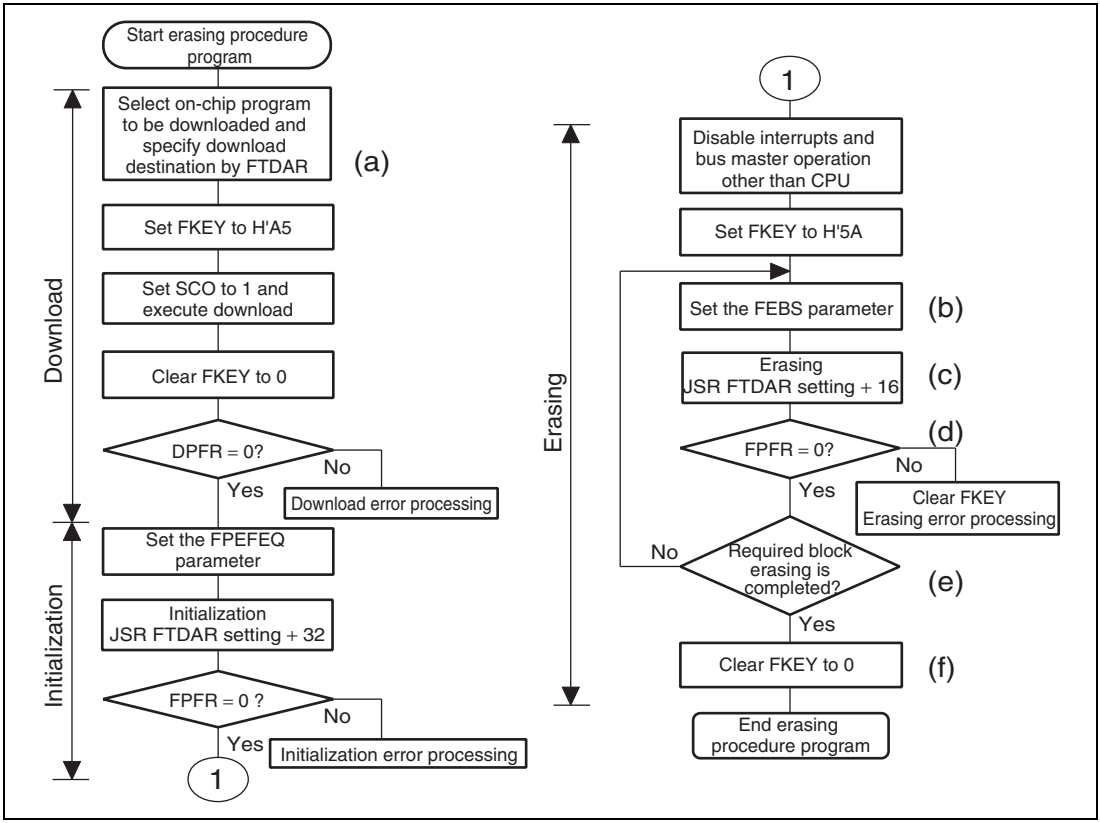


Figure 21.12 Erasing Procedure

The procedure program must be executed in an area other than the user MAT to be erased. Especially the part where the SCO bit in FCCS is set to 1 for downloading must be executed in the on-chip RAM.

The area that can be executed in the steps of the user procedure program (on-chip RAM and user MAT) is shown in section 21.4.4, Storable Areas for Procedure Program and Program Data.

For the downloaded on-chip program area, see the RAM map for programming/erasing in figure 21.10.

A single divided block is erased by one erasing processing. For block divisions, refer to figure 21.4. To erase two or more blocks, update the erase-block number and perform the erasing processing for each block.

(a) Select the on-chip program to be downloaded

Set the EPVB bit in FECS to 1.

Several programming/erasing programs cannot be selected at one time. If several programs are set, download is not performed and a download error is reported to the SS bit in the DPFR parameter.

Specify the start address of the download destination by FTDAR.

The procedures to be carried out after setting FKEY, e.g. download and initialization, are the same as those in the programming procedure. For details, see section 21.4.2 (2), Programming Procedure in User Program Mode.

The procedures after setting parameters for erasing programs are as follows:

(b) Set the FEBS parameter necessary for erasure

Set the erase-block number of the user MAT in the flash erase block select parameter FEBS (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 return value parameter FPFR.

(c) Erasure

Similar to as in programming, there is an entry point for the erasing program in the area from the start address of a download destination specified by FTDAR + 16 bytes. The subroutine is called and erasing is executed by using the following steps.

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

- The general registers other than R0L are saved in the erasing program.
- R0L is a return value of the FPFR parameter.
- Since the stack area is used in the erasing program, a 128-byte stack area at the maximum must be allocated in RAM.

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

(e) Determine whether erasure of the necessary blocks has completed.

If more than one block is to be erased, update the FEBS parameter and repeat steps (b) to (e). Blocks that have already been erased can be erased again.

(f) After erasure completes, clear FKEY and specify software protection.

If this LSI is restarted by a reset immediately after user MAT erasure has completed, secure a reset period (period of $\overline{RES}$ = 0) of 100 µs which is longer than normal.

(4) Erasing and Programming Procedure in User Program Mode

By changing the on-chip RAM address of the download destination in FTDAR, the erasing program and programming program can be downloaded to separate on-chip RAM areas.

Figure 21.13 shows a repeating procedure of erasing and programming.

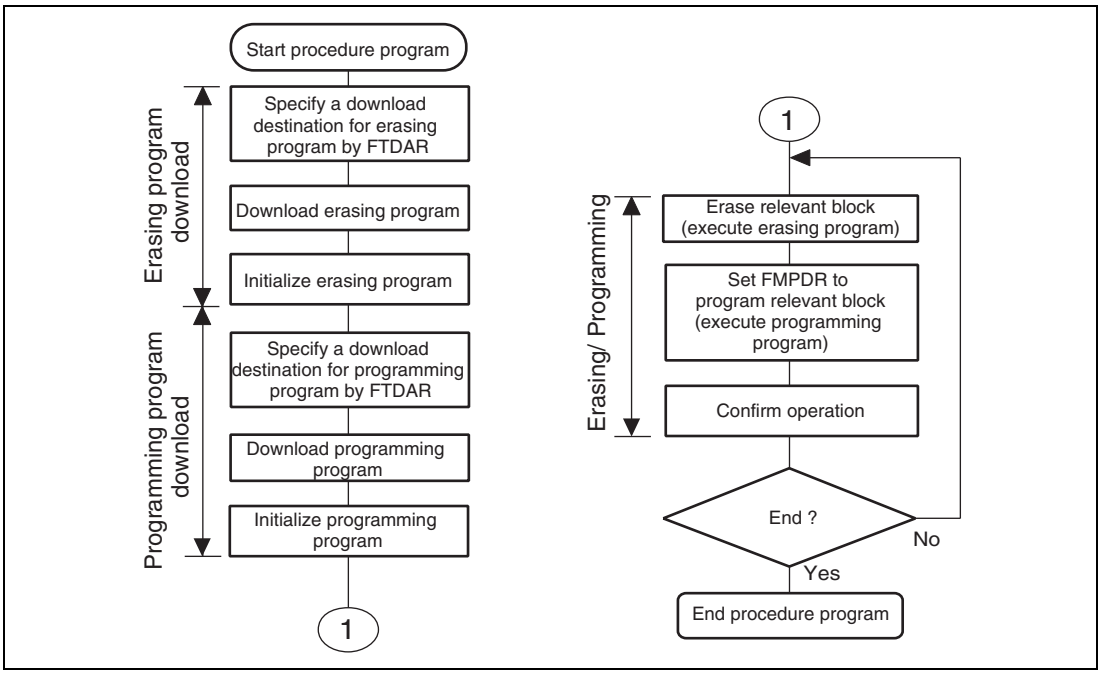


Figure 21.13 Repeating Procedure of Erasing and Programming

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

In this kind of operation, note the following:

- Be careful not to damage on-chip RAM with overlapped settings.
  In addition to the erasing program area and programming program area, areas for the user procedure programs, work area, and stack area are allocated in the on-chip RAM. Do not make settings that will overwrite data in these areas.
- Be sure to initialize both the erasing program and programming program.
  Initialization by setting the FPEFEQ parameter must be performed for both the erasing program and programming program. Initialization must be executed for both entry addresses: (download start address for erasing program) + 32 bytes and (download start address for programming program) + 32 bytes.

21.4.3 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 21.5.

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 21.14 shows the procedure for programming the user MAT in user boot mode.

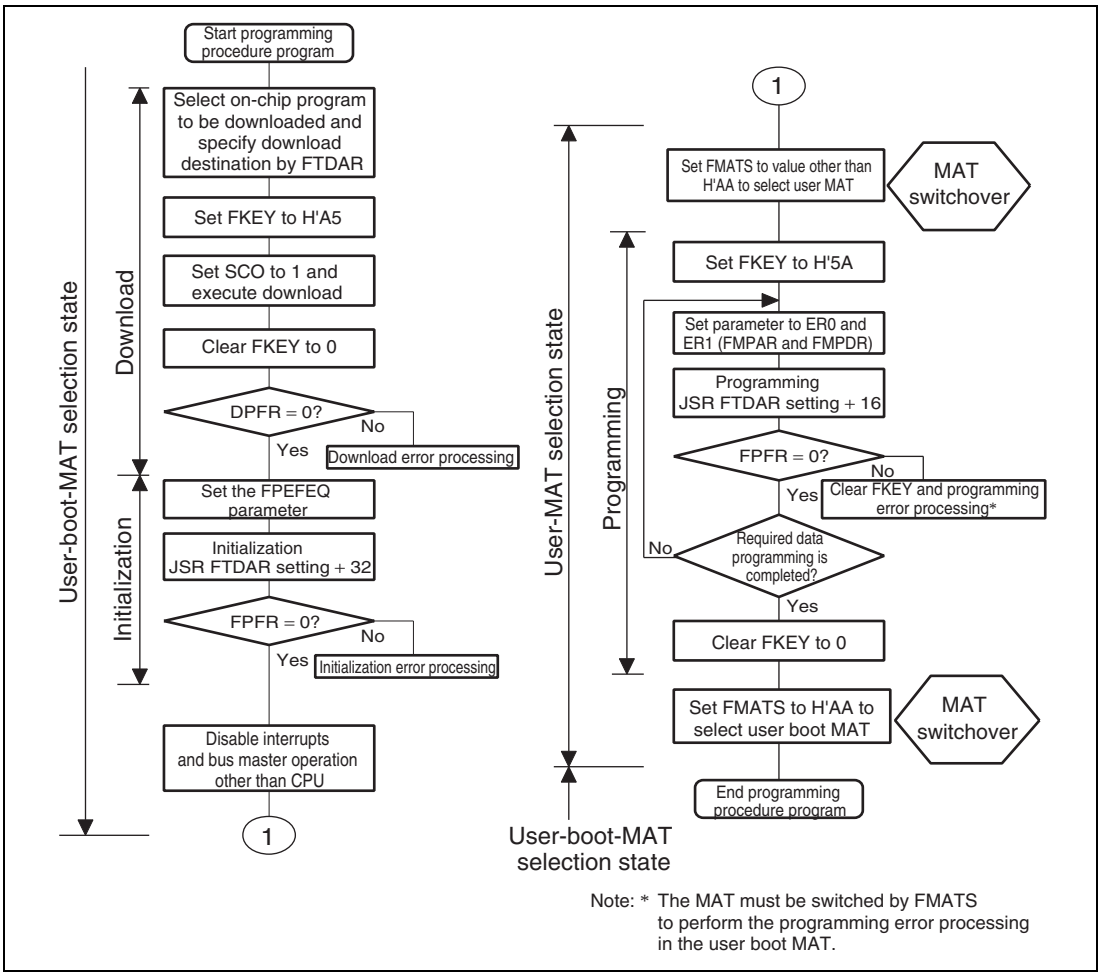


Figure 21.14 Procedure for Programming User MAT in User Boot Mode

The difference between the programming procedures in user program mode and user boot mode is whether the MAT is switched or not as shown in figure 21.14.

In user boot mode, the user boot MAT can be seen in the flash memory space with the 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 21.6, 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 21.4.4, Storable Areas for Procedure 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 21.15 shows the procedure for erasing the user MAT in user boot mode.

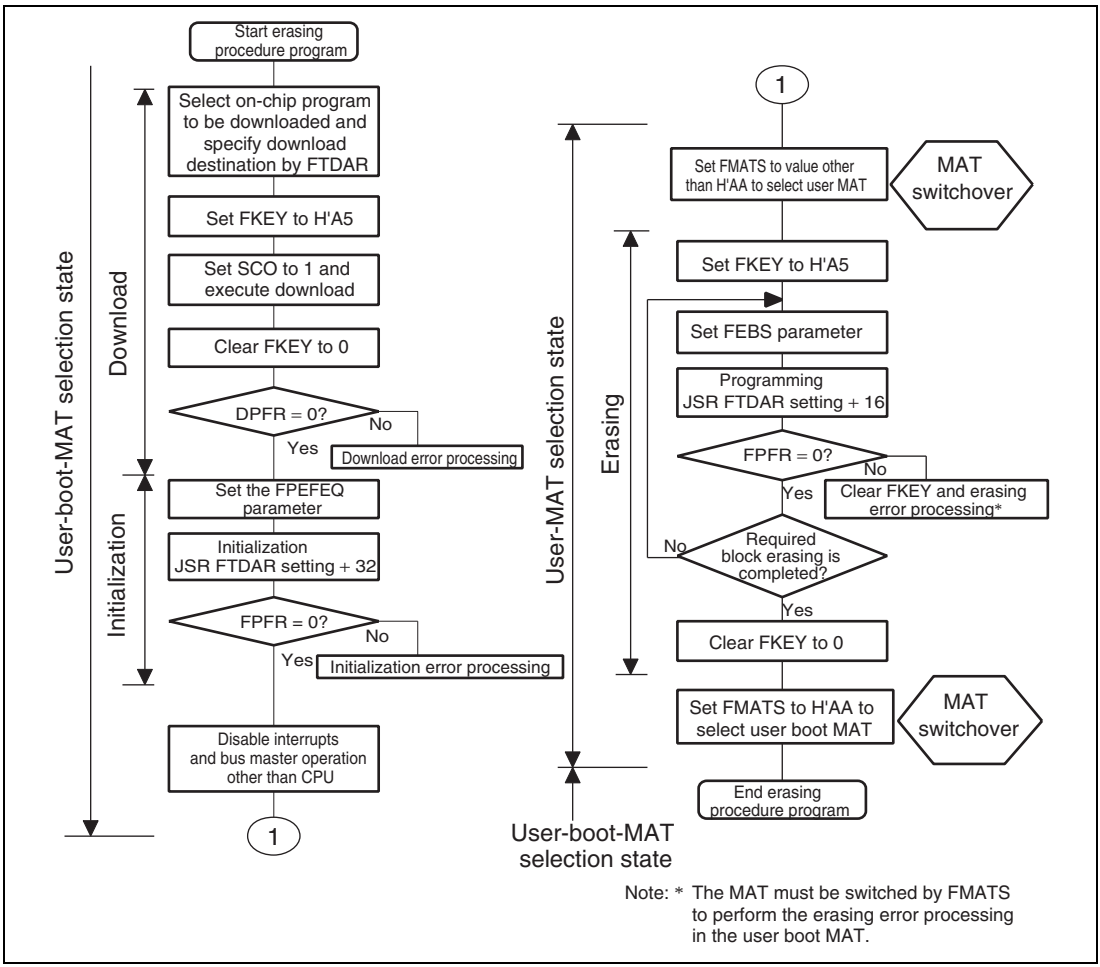


Figure 21.15 Procedure for Erasing User MAT in User Boot Mode

The difference between the erasing procedures in user program mode and user boot mode depends on whether the MAT is switched or not as shown in figure 21.15.

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 21.6, 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 21.4.4, Storable Areas for Procedure Program and Program Data.

21.4.4 Storable Areas for Procedure Program and Program Data

In the descriptions in the previous section, the storable areas for the programming/erasing procedure programs and program data are assumed to be in the on-chip RAM. However, the procedure programs and program data can be stored in and executed from other areas, such as part of flash memory which is not to be programmed or erased.

(1) Conditions that Apply to Programming/Erasing

1. The on-chip programming/erasing program is downloaded from the address in the on-chip RAM specified by FTDAR, therefore, this area is not available for use.
2. The on-chip programming/erasing program will use 128 bytes at the maximum as a stack. So, make sure that this area is allocated.
3. Download by setting the SCO bit to 1 will lead to switching of the MATs. Therefore, if this operation is used, it should be executed from the on-chip RAM.
4. The flash memory is accessible until the start of programming or erasing, that is, until the result of downloading has been determined. The required procedure programs, NMI handling vector, and NMI handling routine should be transferred to the on-chip RAM before programming/erasing of the flash memory starts.
5. Since flash memory is not accessible during programming/erasing processing, programs downloaded to the on-chip RAM are executed. The procedure programs that initiate programming/erasing processing, and execution areas for the NMI interrupt vector table and NMI interrupt handling program must be stored in on-chip RAM.

6. After programming/erasing, access to the flash memory is prohibited until FKEY is cleared.

   In case the LSI mode is changed to generate a reset on completion of a programming/erasing operation, a reset state ($\overline{RES}$ = 0) of 100 µs or more must be secured.

   Transitions to the reset state or hardware standby mode are prohibited during programming/erasing operations. However, when the reset signal is accidentally input to the chip, the reset must be released after a reset period of 100 µs that is longer than normal.
7. Switching of the MATs by FMATS should be required when programming/erasing 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 21.6, Switching between User MAT and User Boot MAT.) Make sure you know which MAT is currently selected when switching them.
8. When the program data storable area indicated by the programming parameter FMPDR is in flash memory, an error will occur even when the program data stored is normal. Therefore, the program data should be temporarily transferred to the on-chip RAM to set an address other than flash memory in FMPDR.

In consideration of these conditions, the following tables show areas where program data can be stored and executed for different combinations of operating mode, user MAT bank configuration, and processing type.

Table 21.7 Executable MAT

	Initiated Mode	
Processing	**User Program Mode**	**User Boot Mode***
Programming	Table 21.8 (1)	Table 21.8 (3)
Erasing	Table 21.8 (2)	Table 21.8 (4)

Note: * Programming/Erasing is possible to the user MAT.

Table 21.8 (1) 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	×*	—	—
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	
Writing H'5A to FKEY	O	O	O	
Setting programming parameter	O	×	O	

	Storable/Executable Area		Selected MAT	
Item	**On-chip RAM**	**User MAT**	**User MAT**	**Embedded Program Storage MAT**
Programming	O	×	O	
Determination of programming result	O	×	O	
Programming error processing	O	×	O	
FKEY clearing	O	×	O	

Note: * Transferring the data to the on-chip RAM in advance enables this area to be used.

Table 21.8 (2) 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**
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	
Writing H'5A to FKEY	O	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	

Table 21.8 (3) 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		

	Storable/Executable Area			Selected MAT	
Item	**On-chip RAM**	**User Boot MAT**	**User MAT**	**User Boot MAT**	**Embedded Program Storage MAT**
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 21.8 (4) 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		

	Storable/Executable Area		Selected MAT		
Item	On-chip RAM	User Boot MAT	User MAT	User Boot MAT	Embedded Program Storage MAT
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.

21.5 Protection

There are two kinds of flash memory programming/erasing protection: hardware and software protection.

21.5.1 Hardware Protection

Programming and erasing of flash memory is forcibly disabled or suspended by hardware protection. In this state, the downloading of an on-chip program and initialization are possible. However, even though a programming/erasing program is initiated, the user MAT cannot be programmed/erased, and a programming/erasing error is reported with the FPFR parameter.

Table 21.9 Hardware Protection

Item	Description	Function to be Protected	
		Download	Programming/ Erasure
FWE pin protection	• When a low-level signal is input to the FWE pin, the FWE bit in FCCS is cleared and the programming/erasing protection state is entered.	—	O
Reset, standby protection	• The programming/erasing interface registers are initialized in the reset state (including a reset by the WDT) and hardware standby mode, 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 stabilized after the power is supplied. In the case of a reset during operation, hold the $\overline{\text{RES}}$ pin low for the RES pulse width that is specified by the AC characteristics. If a reset is input during programming or erasure, values in the flash memory are not guaranteed. In this case, execute erasure and then execute programming again.	O	O

21.5.2 Software Protection

Software protection is set up by disabling download of on-chip programming/erasing programs or by means of a key code.

Table 21.10 Software Protection

		Function to be Protected	
Item	**Description**	**Download**	**Programming/ Erasure**
Protection by SCO bit	• The programming/erasing protection state is entered by clearing the SCO bit in FCCS to 0 to disable downloading of the programming/erasing programs.	O	O
Protection by FKEY	• Downloading and programming/erasing are disabled unless the required key code is written in FKEY. Different key codes are used for downloading and programming/erasing.	O	O

21.5.3 Error Protection

Error protection is a mechanism for aborting programming or erasure when an error occurs, in the form of the microcomputer entering runaway during programming/erasing of the flash memory or operations that are not following the stipulated procedures for programming/erasing. Aborting programming or erasure in such cases prevents damage to the flash memory due to excessive programming or erasing.

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

The FLER bit is set to 1 in the following conditions:

- When an interrupt such as NMI occurs during programming/erasing.
- When the flash memory is read during programming/erasing (including a vector read or an instruction fetch).
- When a SLEEP instruction (including software-standby mode) is executed during programming/erasing.

- When a bus master other than the CPU, such as the DTC or LPC, gets the bus during programming/erasing

Error protection is cancelled only by a reset or a transition to hardware-standby mode.

Note that the reset should be released after a reset period of 100 µs which is longer than normal. 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 damage to the flash memory by extending the reset period so that the charge is released.

The state transition diagram in figure 21.16 shows transitions to and from the error-protection state.

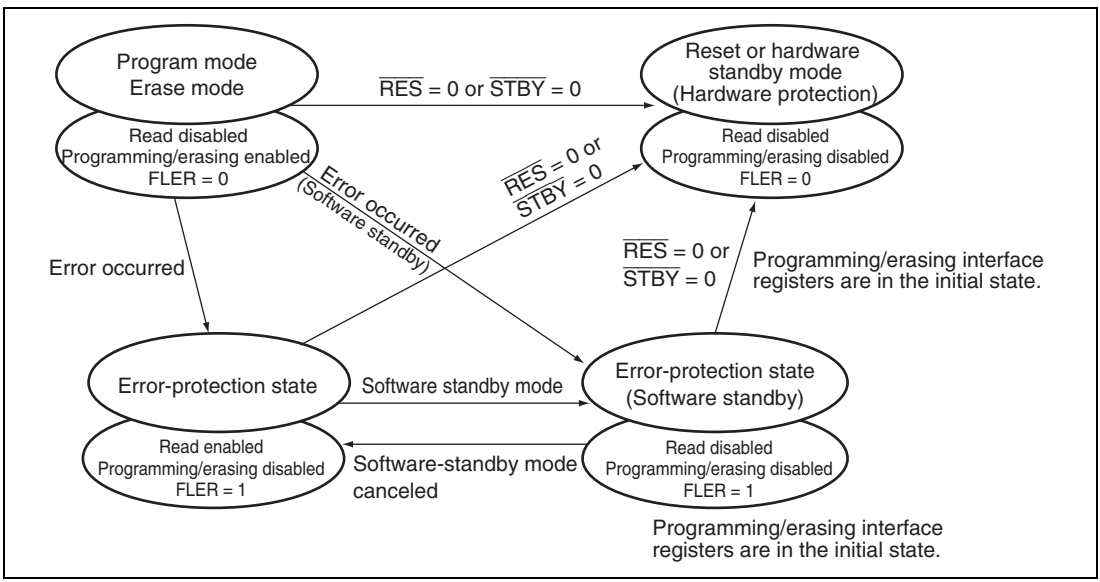


Figure 21.16 Transitions to Error-Protection State

21.6 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.

   If interrupt handling is to be the same before and after MAT switching, transfer the interrupt handling routines to the on-chip RAM and set the WEINTE bit in FCCS to place the interrupt-vector table in the on-chip RAM.
5. Memory sizes of the user MAT and user boot MAT are different. 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.

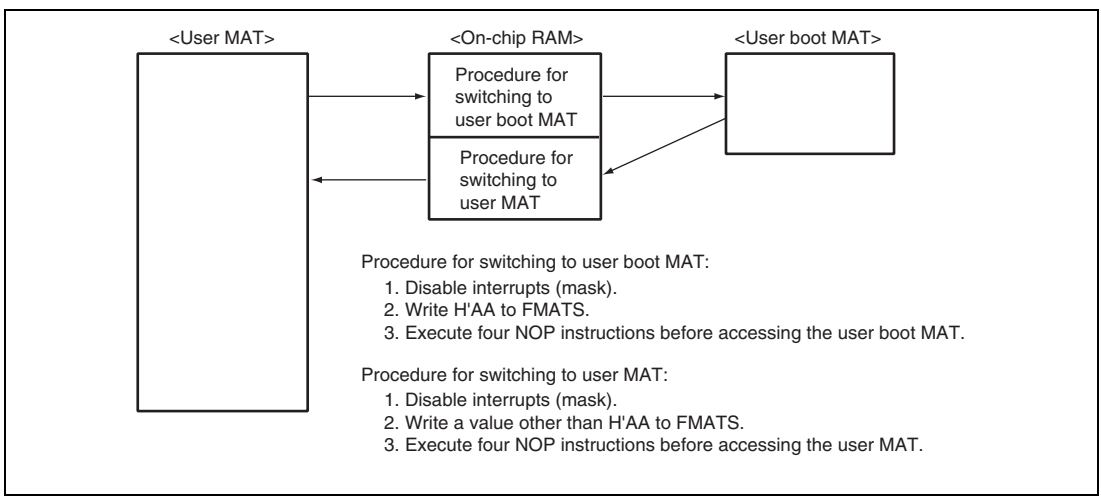


Figure 21.17 Switching between User MAT and User Boot MAT

21.7 Programmer Mode

Along with its on-board programming mode, this LSI also has a programmer mode as another mode for programming/erasing of programs and data. In programmer mode, a general PROM programmer that supports Renesas microcomputers with 1-Mbyte flash memory as a device type*[1] can be used to freely write programs to the on-chip ROM. Programming/erasing is possible on the user MAT and user boot MAT*[2]. Figure 21.18 shows a memory map in programmer mode.

A status-polling system is adopted for operation in automatic programming, automatic erasure, and status-read modes. In status-read mode, details of the internal signals are output after execution of automatic programming or automatic erasure. In programmer mode, a 12-MHz clock signal must be input.

Notes: 1. In this LSI, set the programming voltage of the PROM programmer to 3.3 V.
2. For the PROM programmer and the version of its program, see the instruction manuals for socket adapter.

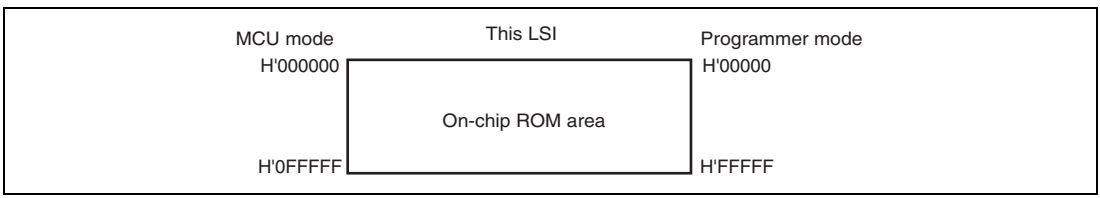


Figure 21.18 Memory Map in Programmer Mode

21.8 Serial Communication Interface Specifications for Boot Mode

The boot program initiated in boot mode performs transmission and reception with the host PC via the on-chip SCI. The serial communication interface specifications for the host and boot program are shown below.

(1) Status

The boot program has three states.

1. Bit-rate-adjustment state

   In this state, the boot program adjusts the bit rate to communicate with the host. Initiating boot mode enables starting of the boot program and transition to the bit-rate-adjustment state. The boot program receives the command from the host to adjust the bit rate. After adjusting the bit rate, the boot 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 in this state. After selection of these settings, the boot program makes a transition to the programming/erasing state by the command for a transition to the programming/erasing state. The boot program transfers the libraries required for erasure to the on-chip RAM and erases the user MATs and user boot MATs before the transition to the programming/erasing state.
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 check and blank check are executed by sending commands from the host.

The boot program states are shown in figure 21.19.

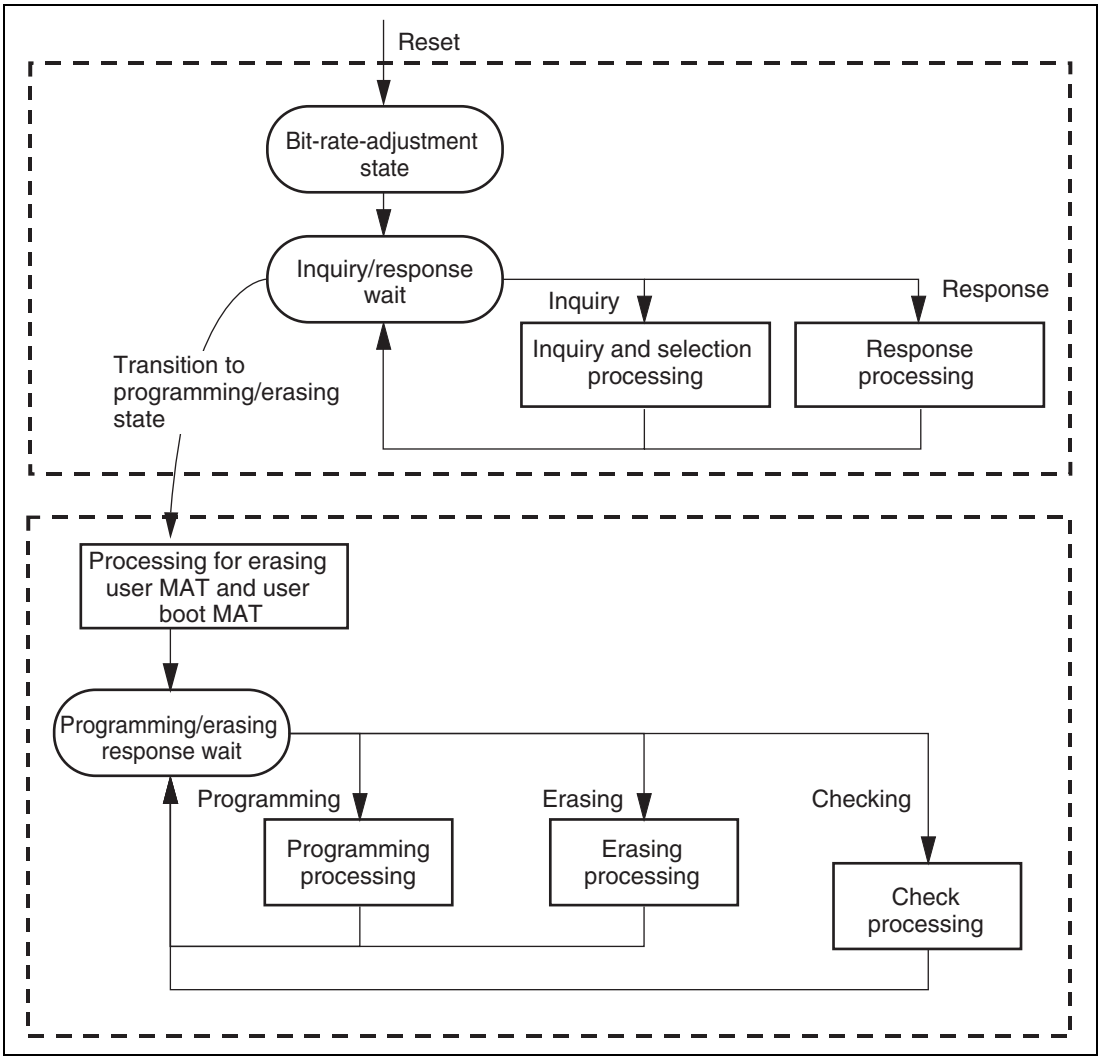


Figure 21.19 Boot Program States

(2) Bit-Rate-Adjustment State

The bit rate is adjusted by measuring the period of a low-level byte (H'00) transmitted 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/selection state. The bit-rate-adjustment sequence is shown in figure 21.20.

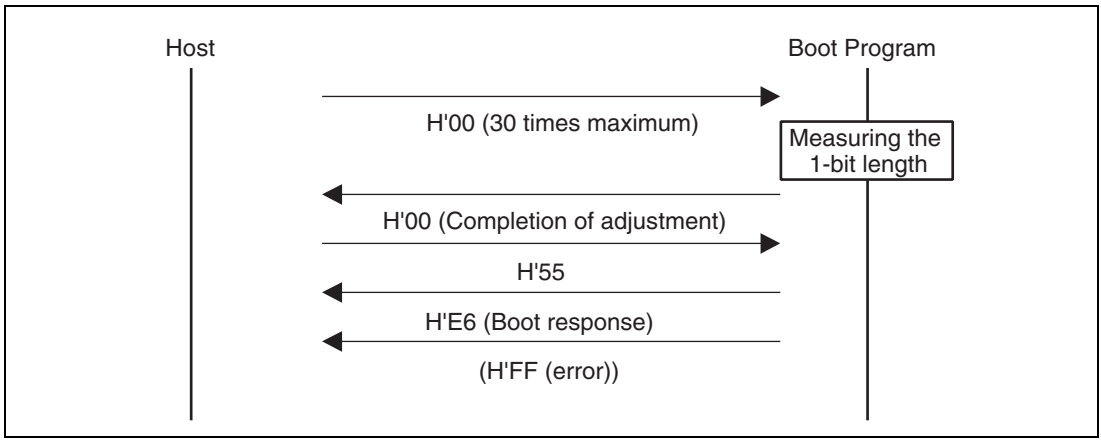


Figure 21.20 Bit-Rate-Adjustment Sequence

(3) Communications Protocol

After adjustment of the bit rate, the protocol for communications between the host and the boot program is as shown below.

1. 1-byte commands and 1-byte responses

   These commands and responses are comprised of a single byte. They are 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. They are selection commands and responses to inquiries.

   The size of program data is not included under this heading because it is determined in another command.
3. Error response

   This response is an error response to the commands. It is two bytes of data, and consists of an error response and an error code.

4. Programming of 128 bytes

   The size is not specified in the commands. The data size is indicated in the response to the programming unit inquiry.

5. Memory read response

   This response consists of r4 bytes of data.

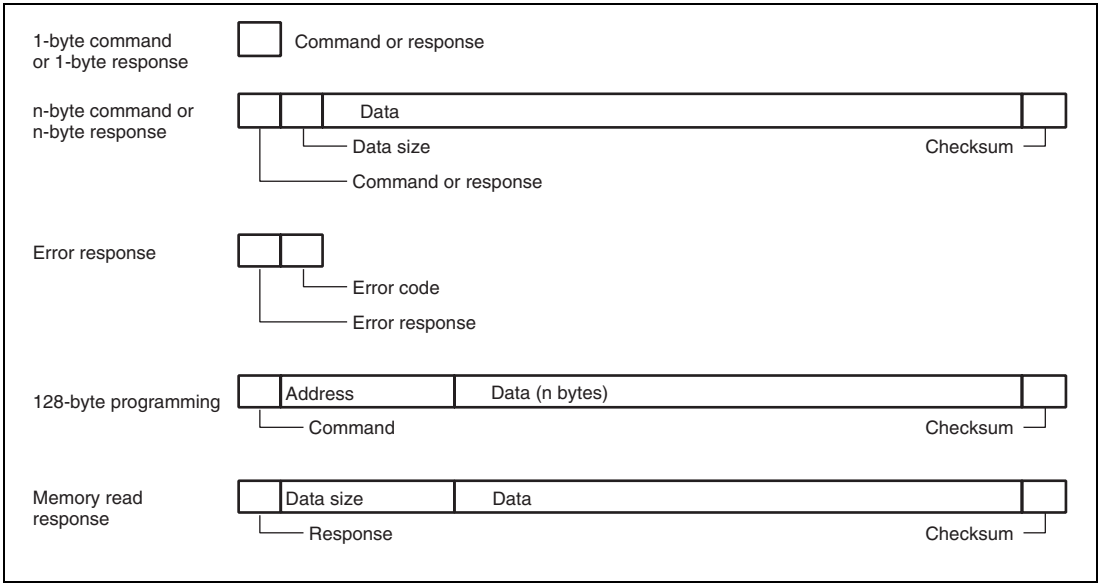


Figure 21.21 Communication Protocol Format

- Command (1 byte): Commands for inquiries, selection, programming, erasing, and checking
- Response (1 byte): Response to an inquiry
- Size (1 byte): The amount of transfer data excluding the command, size, and checksum
- Data (n bytes): Detailed data of a command or response
- Checksum (1 byte): The checksum is calculated so that the total of all values from the command byte to the SUM byte becomes H'00.
- Error response (1 byte): Error response to a command
- Error code (1 byte): Type of the error
- Address (4 bytes): Address for programming
- Data (n bytes): Data to be programmed (n is indicated in the response to the programming unit inquiry.)
- Data size (4 bytes): Four-byte response to a memory read

(4) Inquiry/Selection State

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.

Inquiry and selection commands are listed in table 21.11.

Table 21.11 Inquiry and Selection Commands

Command	Command Name	Description
H'20	Supported Device Inquiry	Inquiry regarding device code and product name
H'10	Device Selection	Selection of device code
H'21	Clock Mode Inquiry	Inquiry regarding number 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 types of division ratios, and the number and values of each ratio type
H'23	Operating Clock Frequency Inquiry	Inquiry regarding the maximum and minimum values of the main clock and peripheral clock
H'24	User Boot MAT Information Inquiry	Inquiry regarding the number of user boot MATs and the start and last addresses of each MAT
H'25	User MAT Information Inquiry	Inquiry regarding the number of user MATs and the start and last addresses of each MAT
H'26	Erased Block 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 size of program data
H'3F	New Bit Rate Selection	Selection of the new bit rate
H'40	Transition to Programming/Erasing State	Erasure of user MAT and user boot MAT, and transition to programming/erasing state
H'4F	Boot Program Status Inquiry	Inquiry into the processing 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 transmitted from the host in that order. These commands are needed in all cases. When two or more selection commands are transmitted at the same time, the last command will be valid.

All of these commands, except for boot program status inquiry (H'4F), will be valid until the boot program receives the programming/erasing state transition command (H'40). The host can choose the needed commands out of the above commands and make inquiries. The boot program status inquiry command (H'4F) remains valid even after the boot program has received the programming/erasing state transition command (H'40).

(a) Supported Device Inquiry

The boot program will return the device codes of the supported devices and the product names in response to the supported device inquiry command.

Command

H'20

- Command, H'20 (1 byte): Inquiry regarding supported devices

Response

H'30	Size	Number of devices	
Number of characters	Device code		Product name
...			
SUM			

- Response, H'30 (1 byte): Response to the supported device inquiry
- Size (1 byte): The number of bytes to be transferred, excluding the command, size, and checksum, that is, the total amount of data consisting the number of devices, the number of characters, device codes, and product names
- Number of devices (1 byte): The number of device types supported by the boot program in the microcomputer
- Number of characters (1 byte): The number of characters in the device codes and boot program's name
- Device code (4 bytes): ASCII code of the supported product name
- Product name (n bytes): ASCII code of the boot program type name
- SUM (1 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 specified supported device in response to the device selection command. The program will return information on the selected device in response to the inquiry after this setting has been made.

Command	H'10	Size	Device code	SUM

- Command, H'10 (1 byte): Device selection
- Size (1 byte): The number of characters in the device code. Fixed at 2.
- Device code (4 bytes): Device code (ASCII code) returned in response to the supported device inquiry
- SUM (1 byte): Checksum

Response	H'06

- Response, H'06 (1 byte): Response to the device selection command.
  The boot program will return ACK when the device code matches.

Error Response	H'90	ERROR

- Error response, H'90 (1 byte): Error response to the device selection command
  ERROR (1 byte): Error code
  H'11: Checksum 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.

Command	H'21

- Command, H'21 (1 byte): Inquiry regarding clock mode

Response	H'31	Size	Number of modes	Mode	···	SUM

- Response, H'31 (1 byte): Response to the clock mode inquiry
- Size (1 byte): Amount of data that represents the number of modes and modes
- Number of clock modes (1 byte): The number of supported clock modes.
  H'00 indicates no clock mode or the device allows the clock mode to be read.
- Mode (1 byte): Values of the supported clock modes (i.e. H'01 means clock mode 1.)
- SUM (1 byte): Checksum

(d) Clock Mode Selection

The boot program will set the specified clock mode in response to the clock mode selection command. The program will return information on the selected clock mode in response to the inquiry after this setting has been made.

The clock mode selection command should be sent after the device selection command.

Command	H'11	Size	Mode	SUM

- Command, H'11 (1 byte): Selection of clock mode
- Size (1 byte): The number of characters that represents the modes. Fixed at 1.
- Mode (1 byte): A clock mode returned in response to the clock mode inquiry.
- SUM (1 byte): Checksum

Response	H'06

- Response, H'06 (1 byte): Response to the clock mode selection command.
  The boot program will return ACK when the clock mode matches.

Error Response	H'91	ERROR

- Error response, H'91 (1 byte) : Error response to the clock mode selection command
- ERROR (1 byte): Error code

  H'11: Checksum error

  H'22: Clock mode error, that is, the clock mode does not match

  Even if the number of clock modes is H'00 or H'01 by a clock mode inquiry, the clock mode must be selected using the respective value.

(e) Division Ratio Inquiry

The boot program will return the supported division ratios in response to the division ratio inquiry command.

Command

H'22

- Command, H'22 (1 byte): Inquiry regarding division ratio

Response

H'32	Size	Number of types					
Number of division ratios	Division ratio	···					
···							
SUM							

- Response, H'32 (1 byte): Response to the division ratio inquiry
- Size (1 byte): The amount of data that represents the number of types, number of division ratios, and division ratios
- Number of types (1 byte): The number of supported division ratio types
  (e.g. H'02 when there are two types: main operating frequency and peripheral module operating frequency)
- Number of division ratios (1 byte): The number of supported division ratios for each operating frequency.
  The number of division ratios supported in the main module and peripheral modules.
- Division ratio (1 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 will be H'FE[–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 (1 byte): Checksum

(f) Operating Clock Frequency Inquiry

The boot program will return the number of operating clock frequencies, and the maximum and minimum values in response to the operating clock frequency inquiry command.

Command	H'23

- Command, H'23 (1 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 (1 byte): Response to operating clock frequency inquiry
- Size (1 byte): The amount of data that represents the number of operating clock frequencies, and the minimum and maximum values of the operating clock frequencies
- Number of operating clock frequencies (1 byte): The number of supported operating clock frequency types
  (e.g. H'02 when there are two types: main operating frequency and peripheral module operating frequency)
- Minimum value of operating clock frequency (2 bytes): Minimum value among the divided clock frequencies.

  The minimum and maximum values of operating clock frequency represent the frequency values (MHz), valid to the hundredths place, and multiplied by 100.

  (e.g. when the value is 20.00 MHz, it will be 2000, which is H'07D0)
- Maximum value of operating clock frequency (2 bytes): Maximum value among the divided clock frequencies.

  There are as many pairs of minimum and maximum values as there are operating clock frequencies.
- SUM (1 byte): Checksum

(g) User Boot MAT Information Inquiry

The boot program will return the number of user boot MATs and their addresses in response to the user boot MAT information inquiry command.

Command	H'24

- Command, H'24 (1 byte): Inquiry regarding user boot MAT information

Response	H'34	Size	Number of areas	
	Area start address			Area last address
	...			
	SUM			

- Response, H'34 (1 byte): Response to user boot MAT information inquiry
- Size (1 byte): The amount of data that represents the number of areas, area start address, and area last address
- Number of areas (1 byte): The number of consecutive user boot MAT areas.
  H'01 when the user boot MAT areas are consecutive.
- Area start address (4 bytes): Start address of the area
- Area last address (4 bytes): Last address of the area.
  There are as many groups of data representing the start and last addresses as there are areas.
- SUM (1 byte): Checksum

(h) User MAT Information Inquiry

The boot program will return the number of user MATs and their addresses in response to the user MAT information inquiry command.

Command	H'25

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

Response	H'35	Size	Number of areas	
	Area start address			Area last address
	...			
	SUM			

- Response, H'35 (1 byte): Response to the user MAT information inquiry
- Size (1 byte): The amount of data that represents the number of areas, area start address, and area last address
- Number of areas (1 byte): The number of consecutive user MAT areas.
  H'01 when the user MAT areas are consecutive.

- Area start address (4 bytes): Start address of the area
- Area last address (4 bytes): Last address of the area.
  There are as many groups of data representing the start and last addresses as there are areas.
- SUM (1 byte): Checksum

(i) Erased Block Information Inquiry

The boot program will return the number of erased blocks and their addresses in response to the erased block information inquiry command.

Command	H'26

- Command, H'26 (1 byte): Inquiry regarding erased block information

Response	H'36	Size	Number of blocks	
	Block start address			Block last address
	...			
	SUM			

- Response, H'36 (1 byte): Response to the erased block information inquiry
- Size (2 bytes): The amount of data that represents the number of blocks, block start address, and block last address.
- Number of blocks (1 byte): The number of erased blocks of flash memory
- Block start address (4 bytes): Start address of a block
- Block last address (4 bytes): Last address of a block
  There are as many groups of data representing the start and last addresses as there are blocks.
- SUM (1 byte): Checksum

(j) Programming Unit Inquiry

The boot program will return the programming unit used to program data in response to the programming unit inquiry command.

Command	H'27

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

Response	H'37	Size	Programming unit	SUM

- Response, H'37 (1 byte): Response to programming unit inquiry
- Size (1 byte): The number of characters that indicate the programming unit. Fixed at 2.
- Programming unit (2 bytes): A unit for programming.
  This is the unit for reception of program data.

- SUM (1 byte): Checksum

(k) New Bit Rate Selection

The boot program will set a new bit rate in response to the new bit rate selection command, and return the new bit rate in response to the confirmation.

This new bit rate selection command 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 (1 byte): Selection of new bit rate
- Size (1 byte): The amount of data that represents the bit rate, input frequency, number of division ratios, and division ratios
- Bit rate (2 bytes): New bit rate
  One hundredth of the value (e.g. when the value is 19200 bps, it will be 192, which is H'00C0.)
- Input frequency (2 bytes): Frequency of the clock input to the boot program.
  This is valid to the hundredths place and represents the frequency value (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 (1 byte): The number of supported division ratios.
  Normally the number is two: one for the main operating frequency and one for peripheral module operating frequency.
- Division ratio 1 (1 byte): The division ratio for the main operating frequency
  - Division ratio: The inverse of the division ratio, i.e. a negative number (e.g. when the clock is divided by two, the value will be H'FE[–2])
- Division ratio 2 (1 byte): The division ratio for the peripheral module operating frequency
  - Division ratio: The inverse of the division ratio, i.e. a negative number (e.g. when the clock is divided by two, the value will be H'FE[–2])
- SUM (1 byte): Checksum

Response

H'06

- Response, H'06 (1 byte): Response to selection of a new bit rate.
  The boot program will return ACK when the new bit rate can be set.

Error Response

H'BF	ERROR

- Error response, H'BF (1 byte): Error response to selection of a new bit rate

- ERROR (1 byte): Error code
  - H'11: Checksum error
  - H'24: Bit rate selection error
    The rate is not available.
  - H'25: Input frequency error
    The input frequency is not within the specified range.
  - H'26: Division ratio error
    The division ratio does not match an available ratio.
  - H'27: Operating frequency error
    The operating frequency is not within the specified range.

(5) Receive Data Check

The methods for checking received data are listed below.

1. Input frequency

   The received value of the input frequency is checked to ensure that it is within the range of the minimum to maximum frequencies which are available with 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 division for the clock modes of the specified device. When the value is out of this range, a division ratio error is generated.
3. Operating frequency

   Operating frequency is calculated from the received value of the input frequency and the division ratio. The input frequency is the frequency input to the LSI, and the operating frequency is the frequency at which the LSI is actually operated. The formula 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 the 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 selection error is generated. The error is calculated using the following formula:

$$\text{Error (\%)} = \{[\frac{\phi \times 10^6}{(N + 1) \times B \times 64 \times 2^{(2 \times n - 1)}}] - 1\} \times 100$$

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 (1 byte): Confirmation of a new bit rate

Response	H'06

- Response, H'06 (1 byte): Response to confirmation of a new bit rate

The sequence of new bit rate selection is shown in figure 21.22.

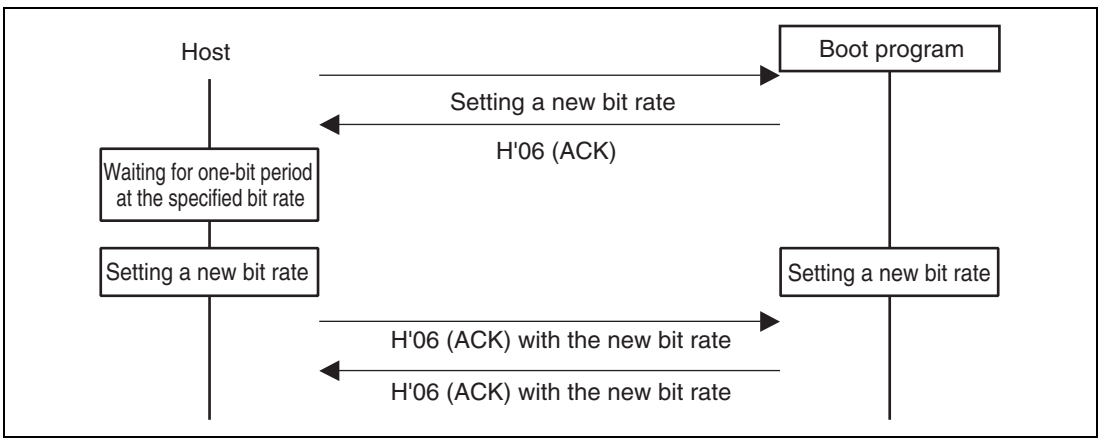


Figure 21.22 Sequence of New Bit Rate Selection

(6) Transition to Programming/Erasing State

The boot program will transfer the erasing program, and erase the user MATs and user boot MATs in that order in response to the transition to the programming/erasing state command. On completion of this erasure, ACK will be returned and a transition made to the programming/erasing state.

Before sending the programming selection command or program data, the host should select the LSI device with the device selection command, the clock mode with the clock mode selection command, and the new bit rate with the new bit rate selection command, and then send the transition to programming/erasing state command.

Command	H'40

- Command, H'40 (1 byte): Transition to programming/erasing state

Response	H'06

- Response, H'06 (1 byte): Response to transition to programming/erasing state.
  The boot program will return ACK when the user MAT and user boot MAT have been erased normally by the transferred erasing program.

Error Response	H'C0	H'51

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

(7) Command Error

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

Error Response	H'80	H'xx

- Error response, H'80 (1 byte): Command error
- Command, H'xx (1 byte): Received command

(8) Command Order

The order for commands in the inquiry/selection state is shown below.

1. A supported device inquiry (H'20) should be made to inquire about the supported 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, programming/erasing information of the user boot MAT and user MAT should be inquired using the user boot MAT information inquiry (H'24), user MAT 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.

(9) Programming/Erasing State

In the 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. The programming/erasing commands are listed in table 21.12.

Table 21.12 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 sum of the user boot MAT
H'4B	User MAT sum check	Checks the sum of the user MAT
H'4C	User boot MAT blank check	Checks whether the contents of the user boot MAT are blank
H'4D	User MAT blank check	Checks whether the contents of the user MAT are blank
H'4F	Boot program status inquiry	Inquires into the boot program's processing status

Programming: Programming is executed by a programming-selection command and a 128-byte programming command.

First, the host should send the programming-selection command, and select the programming method and programming MATs. There are two programming selection commands according to the area and method for programming.

1. User boot MAT programming selection
2. User MAT programming selection

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 program data according to the method specified by the selection command. When more than 128 bytes of data are to be programmed, 128-byte programming commands should be executed repeatedly. Sending from the host 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.

In case of continuing programming with another method or programming 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 21.23.

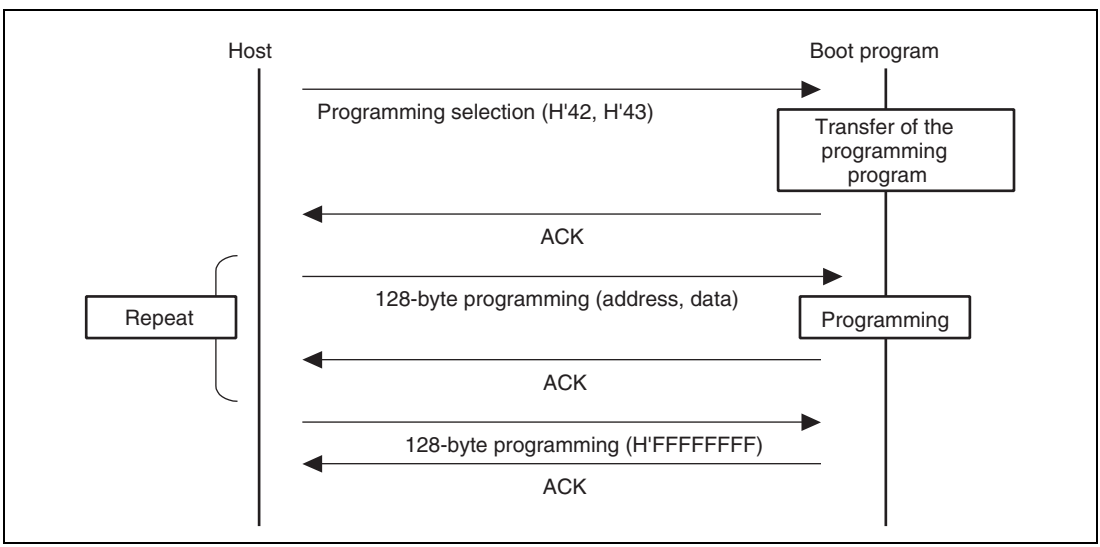


Figure 21.23 Programming Sequence

(a) User Boot MAT Programming Selection

The boot program will transfer a programming program in response to the user boot MAT programming selection command. The data is programmed to the user boot MAT by the transferred programming program.

Command	H'42

- Command, H'42 (1 byte): User boot MAT programming selection

Response	H'06

- Response, H'06 (1 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)

User MAT Programming Selection: The boot program will transfer a programming program in response to the user MAT programming selection command. The data is programmed to the user MAT by the transferred programming program.

Command	H'43

- Command, H'43 (1 byte): User MAT programming selection

Response	H'06

- Response, H'06 (1 byte): Response to user MAT programming selection.
  When the programming program has been transferred, the boot program will return ACK.

Error Response	H'C3	ERROR

- Error response, H'C3 (1 byte): Error response to user MAT programming selection
- ERROR (1 byte): Error code

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

(b) 128-Byte Programming

The boot program will use the programming program transferred by the programming selection command for programming the user boot MAT or user MAT in response to the 128-byte programming command.

Command

H'50	Address						
Data	···						
···							
SUM							

- Command, H'50 (1 byte): 128-byte programming
- Programming address (4 bytes): Start address for programming.
  Multiple of the size specified in response to the programming unit inquiry command.
  (e.g. H'00, H'01, H'00, H'00: H'010000)
- Program data (128 bytes): Data to be programmed.
  The size is specified in response to the programming unit inquiry command.
- SUM (1 byte): Checksum

Response

H'06

- Response, H'06 (1 byte): Response to 128-byte programming.
  On completion of programming, the boot program will return ACK.

Error Response

H'D0	ERROR

- Error response, H'D0 (1 byte): Error response to 128-byte programming
- ERROR (1 byte): Error code

  H'11: Checksum Error
  H'2A: Address error
  H'53: Programming error
  A programming error has occurred and programming cannot be continued.

The specified address should match the boundary of the programming unit. For example, when the programming unit is 128 bytes, the lower eight bits of the address should be H'00 or H'80.
When the program data is less than 128 bytes, 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 programming and wait for selection of programming or erasing.

Command

H'50	Address	SUM

- Command, H'50 (1 byte): 128-byte programming
- Programming address (4 bytes): End code (H'FF, H'FF, H'FF, H'FF)
- SUM (1 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 (1 byte): Error response to 128-byte programming
- ERROR (1 byte): Error code

  H'11: Checksum error
  H'2A: Address error
  H'53: Programming error
  An error has occurred in programming and programming cannot be continued.

(10) Erasure

Erasure is performed with the erasure selection and block erasure commands.

First, 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 processing. On completion of erasing, the boot program will wait for selection of programming or erasing.

The sequence for the erasure selection command and block erasure command is shown in figure 21.24.

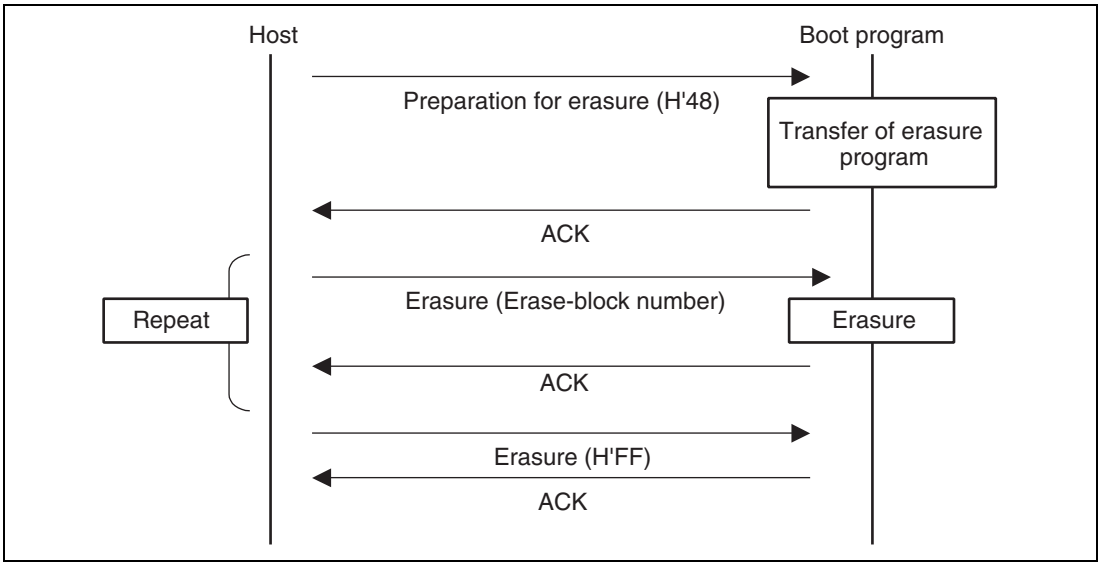


Figure 21.24 Erasure Sequence

(a) Erasure Selection

The boot program will transfer the erasing program in response to the erasure selection command. User MAT data is erased by the transferred erasing program.

Command	H'48

- Command, H'48 (1 byte): Erasure selection

Response	H'06

- Response, H'06 (1 byte): Response to erasure selection.
  After the erasing program has been transferred, the boot program will return ACK.

Error Response	H'C8	ERROR

- Error response, H'C8 (1 byte): Error response to erasure selection
- ERROR (1 byte): Error code

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

(b) Block Erasure

The boot program will erase the contents of the specified block in response to the block erasure command.

Command	H'58	Size	Block number	SUM

- Command, H'58 (1 byte): Erasure
- Size (1 byte): The number of characters that represents the erase-block number.
  Fixed at 1.
- Block number (1 byte): Number of the block to be erased
- SUM (1 byte): Checksum

Response	H'06

- Response, H'06 (1 byte): Response to erasure
  On completion of erasure, the boot program will return ACK.

Error Response	H'D8	ERROR

- Error response, H'D8 (1 byte): Response to erasure
- ERROR (1 byte): Error code

  H'11: Checksum error
  H'29: Block number error
  Block number is incorrect.
  H'51: Erasing 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 (1 byte): Erasure
- Size (1 byte): The number of characters that represents the block number.
  Fixed at 1.
- Block number (1 byte): H'FF
  Stop code for erasure
- SUM (1 byte): Checksum

Response	H'06

- Response, H'06 (1 byte): Response to end of erasure (ACK will be returned)

When erasure is to be performed again after the block number H'FF has been sent, the procedure should be executed from the erasure selection command.

(11) Memory Read

The boot program will return the data in the specified address in response to the memory read command.

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 (one byte)

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

Response								
	H'52	Read size						
	Data	···						
	SUM							

- Response: H'52 (1 byte): Response to memory read
- Read size (4 bytes): Size of data to be read
- Data (n bytes): Data of 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: Checksum error
  H'2A: Address error
  The read address is not in the MAT.
  H'2B: Size error
  The read size exceeds the MAT.

(12) User Boot MAT Sum Check

The boot program will return the total amount of bytes of the user boot MAT contents in response to the user boot MAT sum check command.

Command	H'4A

- Command, H'4A (1 byte): Sum check for user boot MAT

Response	H'5A	Size	Checksum of MAT	SUM

- Response, H'5A (1 byte): Response to the checksum of user boot MAT
- Size (1 byte): The number of characters that represents the checksum. Fixed at 4.
- Checksum of MAT (4 bytes): Checksum of user boot MATs. The total amount of data is obtained in byte units.
- SUM (1 byte): Checksum (for transmit data)

(13) User MAT Sum Check

The boot program will return the total amount of bytes of the user MAT contents in response to the user MAT sum check command.

Command	H'4B

- Command, H'4B (1 byte): Checksum for user MAT

Response	H'5B	Size	Checksum of MAT	SUM

- Response, H'5B (1 byte): Response to the checksum of the user MAT
- Size (1 byte): The number of characters that represents the checksum. Fixed at 4.
- Checksum of MAT (4 bytes): Checksum of user MATs. The total amount of data is obtained in byte units.
- SUM (1 byte): Checksum (for transmit data)

(14) User Boot MAT Blank Check

The boot program will check whether or not all user boot MATs are blank and return the result in response to the user boot MAT blank check command.

Command	H'4C

- Command, H'4C (1 byte): Blank check for user boot MATs

Response	H'06

- Response, H'06 (1 byte): Response to blank check of user boot MATs. If all user boot MATs are blank (H'FF), the boot program will return ACK.

Error Response	H'CC	H'52

- Error response, H'CC (1 byte): Error response to blank check for user boot MATs
- Error code, H'52 (1 byte): Erasure incomplete error

(15) User MAT Blank Check

The boot program will check whether or not all user MATs are blank and return the result in response to the user MAT blank check command.

Command	H'4D

- Command, H'4D (1 byte): Blank check for user MATs

Response	H'06

- Response, H'06 (1 byte): Response to blank check for user MATs.
  If all user MATs are blank (H'FF), the boot program will return ACK.

Error Response	H'CD	H'52

- Error response, H'CD (1 byte): Error response to blank check for user MATs
- Error code, H'52 (1 byte): Erasure incomplete error

(16) Boot Program State Inquiry

The boot program will return indications of its present state and error condition in response to the boot program state inquiry command. This inquiry can be made in either the inquiry/selection state or the programming/erasing state.

Command	H'4F

- Command, H'4F (1 byte): Inquiry regarding boot program's state

Response	H'5F	Size	Status	ERROR	SUM

- Response, H'5F (1 byte): Response to boot program state inquiry
- Size (1 byte): The number of characters. Fixed at 2.
- Status (1 byte): State of the standard boot program
- ERROR (1 byte): Error status

  ERROR = 0 indicates normal operation.
  ERROR = 1 indicates error has occurred.
- SUM (1 byte): Checksum

Table 21.13 Status Code

Code	Description
H'11	Device Selection Wait
H'12	Clock Mode Selection Wait
H'13	Bit Rate Selection Wait
H'1F	Programming/Erasing State Transition Wait (Bit rate selection is completed)
H'31	Programming/Erasing State
H'3F	Programming/Erasing Selection Wait (Erasure is completed)
H'4F	Program Data Receive Wait (Programming is completed)
H'5F	Erase Block Specification Wait (Erasure is completed)

Table 21.14 Error Code

Code	Description
H'00	No Error
H'11	Checksum 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	Erasing 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

21.9 Usage Notes

1. The initial state of a Renesas product at shipment is the erased state. For a product whose history of erasing is undefined, automatic erasure for checking the initial state (erased state) and compensating is recommended.
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 of the PROM programmer does not match the specifications, too much current flows and the product may be damaged.
4. If a voltage higher than the rated voltage is applied, the product may be fatally damaged. Use a PROM programmer that supports a programming voltage of 3.3 V for Renesas microcomputers with 1-Mbyte flash memory. Do not set the programmer to HN28F101 or a programming voltage of 5.0 V. Use only the specified socket adapter. If other adapters are used, the product may be damaged.
5. Do not remove the chip from the PROM programmer nor input a reset signal during programming/erasing. As a high voltage is applied to the flash memory during programming/erasing, doing so may damage flash memory permanently. If a reset is input accidentally, the reset must be released after a reset period of 100 µs which is longer than normal.
6. After programming/erasing, access to the flash memory is prohibited until FKEY is cleared. In case the LSI mode is changed to generate a reset on completion of a programming/erasing operation, a reset state ($\overline{RES}$ = 0) of 100 µs or more must be secured. Transitions to the reset state or hardware standby mode are prohibited during programming/erasing operations. However, when the reset signal is accidentally input to the chip, the reset must be released after a reset period of 100 µs that is longer than normal.
7. At turning on or off the VCC power supply, fix the $\overline{RES}$ pin to low and set the flash memory to the hardware protection state. This power-on or power-off timing must also be satisfied at a power-off or power-on caused by a power failure and other factors.
8. Perform programming to a 128-byte programming-unit block only once in on-board programming or programmer mode.

   Perform programming in the state where the programming-unit block is fully erased.
9. When a chip is to be reprogrammed with the programmer after it has already been programmed or erased in on-board programming mode, automatic programming is recommended to be performed after automatic erasure.
10. To write data or programs to the flash memory, program data and programs must be allocated to addresses higher than that of the external interrupt vector table (in normal mode: H'0020, in advanced mode: H'000040), and H'FF must be written to the areas that are reserved for the system in the exception handling vector table.

11. If data other than H'FF (4 bytes) is written to the key code area (in normal mode: H'001E to H'001F, in advance mode: H'00003C to H'00003F) of 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.

    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.
12. The code size of the programming program that includes the initialization routine or the erasing program that includes the initialization routine is 2 kbytes or less. Accordingly, when the CPU clock frequency is 20 MHz, the download for each program takes approximately 200 µs at the maximum.
13. While an instruction in the on-chip RAM is being executed, the DTC can write to the SCO bit in FCCS that is used for a download request or FMATS that is used for MAT switching. Make sure that these registers are not accidentally written to, otherwise an on-chip program may be downloaded and damage the on-chip RAM or a MAT switchover may occur and the CPU get out of control. Do not use the DTC to write to flash memory related registers.
14. A programming/erasing program for flash memory used in the conventional H8S F-ZTAT microcomputer which does not support download of the on-chip program by a SCO transfer request cannot run in this LSI. Be sure to download the on-chip program to execute programming/erasing of flash memory in this H8S F-ZTAT microcomputer.
15. Unlike the conventional H8S F-ZTAT microcomputer, no countermeasures are available for a runaway by the WDT during programming/erasing. Prepare countermeasures (e.g. use of periodic timer interrupts) for the WDT with taking the programming/erasing time into consideration as required.

Section 22 Boundary Scan (JTAG)

The JTAG (Joint Test Action Group) is standardized as an international standard, IEEE Standard 1149.1, and is open to the public as IEEE Standard Test Access Port and Boundary-Scan Architecture. Although the name of the function is boundary scan and the name of the group who worked on standardization is the JTAG, the JTAG is commonly used as the name of a boundary scan architecture and a serial interface to access the devices having the architecture.

This LSI has a boundary scan function (JTAG). Using this function along with other LSIs facilitates testing a printed-circuit board.

22.1 Features

- Five test pins (ETCK, ETDI, ETDO, ETMS, and $\overline{\text{ETRST}}$)
- TAP controller
- Six instructions
  BYPASS mode
  EXTEST mode
  SAMPLE/PRELOAD mode
  CLAMP mode
  HIGHZ mode
  IDCODE mode
  (These instructions are test modes corresponding to IEEE 1149.1.)

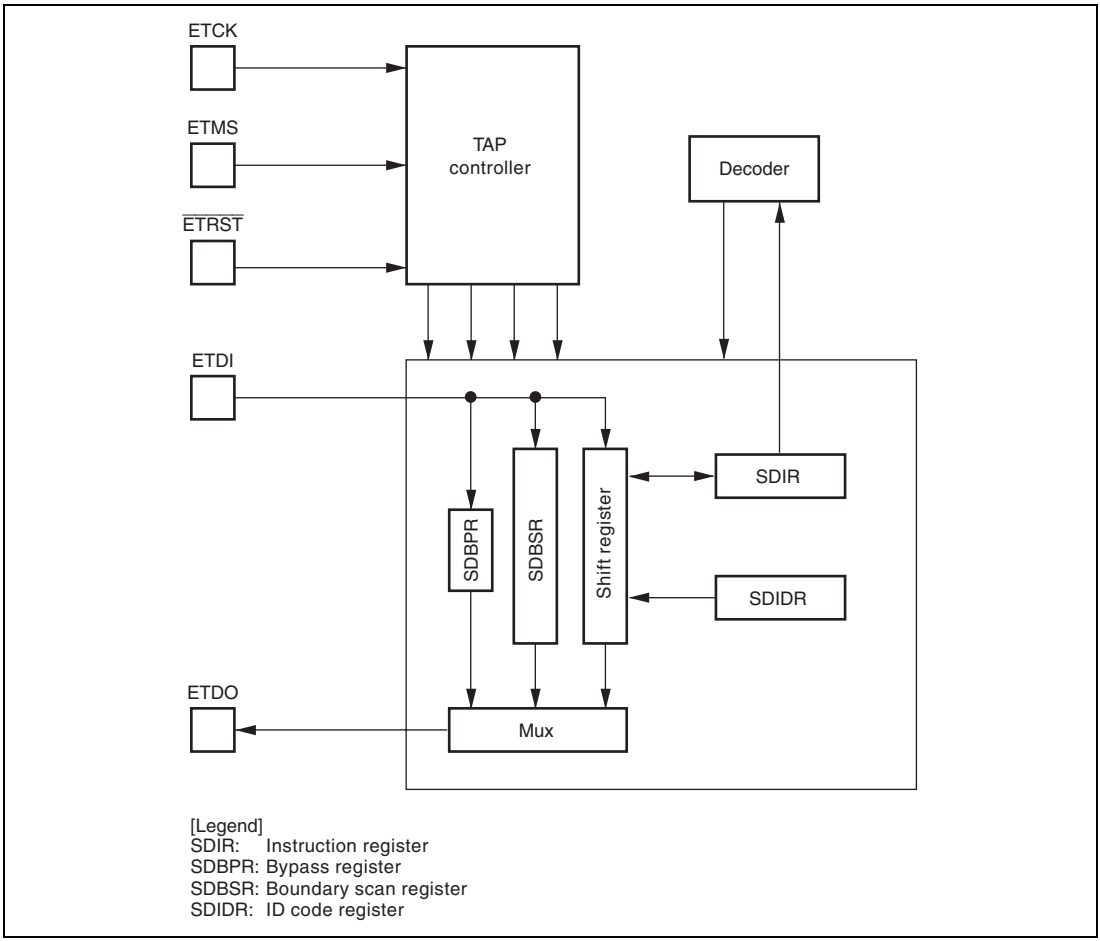


Figure 22.1 JTAG Block Diagram

22.2 Input/Output Pins

Table 22.1 shows the JTAG pin configuration.

Table 22.1 Pin Configuration

Pin Name	Abbreviation	I/O	Function
Test clock	ETCK	Input	Test clock input Provides an independent clock supply to the JTAG. As the clock input to the ETCK pin is supplied directly to the JTAG, a clock waveform with a duty cycle close to 50% should be input. For details, see section 26, Electrical Characteristics. If there is no input, the ETCK pin is fixed to 1 by an internal pull-up.
Test mode select	ETMS	Input	Test mode select input Sampled on the rise of the ETCK pin. The ETMS pin controls the internal state of the TAP controller. If there is no input, the ETMS pin is fixed to 1 by an internal pull-up.
Test data input	ETDI	Input	Serial data input Performs serial input of instructions and data for JTAG registers. ETDI is sampled on the rise of the ETCK pin. If there is no input, the ETDI pin is fixed to 1 by an internal pull-up.
Test data output	ETDO	Output	Serial data output Performs serial output of instructions and data from JTAG registers. Transfer is performed in synchronization with the ETCK pin. If there is no output, the ETDO pin goes to the high-impedance state.
Test reset	$\overline{\text{ETRST}}$	Input	Test reset input signal Initializes the JTAG asynchronously. If there is no input, the $\overline{\text{ETRST}}$ pin is fixed to 1 by an internal pull-up.

22.3 Register Descriptions

The JTAG has the following registers.

- Instruction register (SDIR)
- Bypass register (SDBPR)
- Boundary scan register (SDBSR)
- ID code register (SDIDR)

Instructions can be input to the instruction register (SDIR) by serial transfer from the test data input pin (ETDI). Data from SDIR can be output via the test data output pin (ETDO). The bypass register (SDBPR) is a 1-bit register to which the ETDI and ETDO pins are connected in BYPASS, CLAMP, or HIGHZ mode. The boundary scan register (SDBSR) is a 334-bit register to which the ETDI and ETDO pins are connected in SAMPLE/PRELOAD or EXTEST mode. The ID code register (SDIDR) is a 32-bit register; a fixed code can be output via the ETDO pin in IDCODE mode. All registers cannot be accessed directly by the CPU.

Table 22.2 shows the kinds of serial transfer possible with each JTAG register.

Table 22.2 JTAG Register Serial Transfer

Register	Serial Input	Serial Output
SDIR	Possible	Possible
SDBPR	Possible	Possible
SDBSR	Possible	Possible
SDIDR	Impossible	Possible

22.3.1 Instruction Register (SDIR)

SDIR is a 32-bit read-only register. JTAG instructions can be transferred to SDIR by serial input from the ETDI pin. SDIR can be initialized when the $\overline{\text{ETRST}}$ pin is low or the TAP controller is in the Test-Logic-Reset state, but is not initialized by a reset or in standby mode.

Only 4-bit instructions can be transferred to SDIR. If an instruction exceeding 4 bits is input, the last 4 bits of the serial data will be stored in SDIR.

Bit	Bit Name	Initial Value	R/W	Description
31	TS3	1	R	Test Set Bits
30	TS2	1	R	0000: EXTEST mode
29	TS1	1	R	0001: Setting prohibited
28	TS0	0	R	0010: CLAMP mode
				0011: HIGHZ mode
				0100: SAMPLE/PRELOAD mode
				0101: Setting prohibited
				: :
				1101: Setting prohibited
				1110: IDCODE mode (Initial value)
				1111: BYPASS mode
27 to 14	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
13	—	1	R	Reserved This bit is always read as 1 and cannot be modified.
12	—	0	R	Reserved This bit is always read as 0 and cannot be modified.
11	—	1	R	Reserved This bit is always read as 1 and cannot be modified.
10	—	0	R	Reserved This bit is always read as 0 and cannot be modified.
9, 8	—	All 1	R	Reserved These bits are always read as 1 and cannot be modified.
7 to 1	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
0	—	1	R	Reserved This bit is always read as 1 and cannot be modified.

22.3.2 Bypass Register (SDBPR)

SDBPR is a 1-bit shift register. In BYPASS, CLAMP, or HIGHZ mode, SDBPR is connected between the ETDI and ETDO pins.

22.3.3 Boundary Scan Register (SDBSR)

SDBSR is a shift register provided on the PAD for controlling the I/O terminals of this LSI.

Using EXTEST mode or SAMPLE/PRELOAD mode, a boundary scan test conforming to the IEEE1149.1 standard can be performed.

Table 22.3 shows the relationship between the terminals of this LSI and the boundary scan register.

Table 22.3 Correspondence between Pins and Boundary Scan Register

Pin No.	Pin Name	Input/Output	Bit No.
		from ETDI	
2	P43	Input	333
		Enable	332
		Output	331
3	P44	Input	330
		Enable	329
		Output	328
4	P45	Input	327
		Enable	326
		Output	325
5	P46	Input	324
		Enable	323
		Output	322
6	P47	Input	321
		Enable	320
		Output	319
9	MD1	Input	318
		—	—
		—	—
10	MD0	Input	317
		—	—
		—	—

Pin No.	Pin Name	Input/Output	Bit No.
11	NMI	Input	316
		—	—
		—	—
14	P52	Input	315
		Enable	314
		Output	313
15	P51	Input	312
		Enable	311
		Output	310
16	P50	Input	319
		Enable	308
		Output	307
17	P97	Input	306
		Enable	305
		Output	304
18	P96	Input	303
		Enable	302
		Output	301
19	P95	Input	300
		Enable	299
		Output	298
20	P94	Input	297
		Enable	296
		Output	295
21	P93	Input	294
		Enable	293
		Output	292
22	P92	Input	291
		Enable	290
		Output	289
23	P91	Input	288
		Enable	287
		Output	286
24	P90	Input	285
		Enable	284
		Output	283
25	MD2	Input	282
		—	—
		—	—

Pin No.	Pin Name	Input/Output	Bit No.
26	FWE	Input	281
		—	—
		—	—
32	PE0	Input	280
		Enable	279
		Output	278
33	PA7	Input	277
		Enable	276
		Output	275
34	PA6	Input	274
		Enable	273
		Output	272
35	PA5	Input	271
		Enable	270
		Output	269
37	PA4	Input	268
		Enable	267
		Output	266
38	PA3	Input	265
		Enable	264
		Output	263
39	PA2	Input	262
		Enable	261
		Output	260
40	PA1	Input	259
		Enable	258
		Output	257
41	PA0	Input	256
		Enable	255
		Output	254
43	PF7	Input	253
		Enable	252
		Output	251
44	PF6	Input	250
		Enable	249
		Output	248
45	PF5	Input	247
		Enable	246
		Output	245

Pin No.	Pin Name	Input/Output	Bit No.
46	PF4	Input	244
		Enable	243
		Output	242
47	PF3	Input	241
		Enable	240
		Output	239
48	PF2	Input	238
		Enable	237
		Output	236
49	PF1	Input	235
		Enable	234
		Output	233
50	PF0	Input	232
		Enable	231
		Output	230
51	PG7	Input	229
		Enable	228
		Output	227
52	PG6	Input	226
		Enable	225
		Output	224
53	PG5	Input	223
		Enable	222
		Output	221
54	PG4	Input	220
		Enable	219
		Output	218
55	PG3	Input	217
		Enable	216
		Output	215
56	PG2	Input	214
		Enable	213
		Output	212
57	PG1	Input	211
		Enable	210
		Output	209
58	PG0	Input	208
		Enable	207
		Output	206

Pin No.	Pin Name	Input/Output	Bit No.
59	PD7	Input	205
		Enable	204
		Output	203
60	PD6	Input	202
		Enable	201
		Output	200
61	PD5	Input	199
		Enable	198
		Output	197
62	PD4	Input	196
		Enable	195
		Output	194
63	PD3	Input	193
		Enable	192
		Output	191
64	PD2	Input	190
		Enable	189
		Output	188
65	PD1	Input	187
		Enable	186
		Output	185
66	PD0	Input	184
		Enable	183
		Output	182
68	P70	Input	181
		—	—
		—	—
69	P71	Input	180
		—	—
		—	—
70	P72	Input	179
		—	—
		—	—
71	P73	Input	178
		—	—
		—	—
72	P74	Input	177
		—	—
		—	—

Pin No.	Pin Name	Input/Output	Bit No.
73	P75	Input	176
		—	—
		—	—
74	P76	Input	175
		—	—
		—	—
75	P77	Input	174
		—	—
		—	—
78	P60	Input	173
		Enable	172
		Output	171
79	P61	Input	170
		Enable	169
		Output	168
80	P62	Input	167
		Enable	166
		Output	165
81	P63	Input	164
		Enable	163
		Output	162
82	P64	Input	161
		Enable	160
		Output	159
83	P65	Input	158
		Enable	157
		Output	156
84	P66	Input	155
		Enable	154
		Output	153
85	P67	Input	152
		Enable	151
		Output	150
87	PC7	Input	149
		Enable	148
		Output	147
88	PC6	Input	146
		Enable	145
		Output	144

Pin No.	Pin Name	Input/Output	Bit No.
9+	PC5	Input	143
		Enable	142
		Output	141
90	PC4	Input	140
		Enable	139
		Output	138
91	PC3	Input	137
		Enable	136
		Output	135
92	PC2	Input	134
		Enable	133
		Output	132
93	PC1	Input	131
		Enable	130
		Output	129
94	PC0	Input	128
		Enable	127
		Output	126
96	P27	Input	125
		Enable	124
		Output	123
97	P26	Input	122
		Enable	121
		Output	120
98	P25	Input	119
		Enable	118
		Output	117
99	P24	Input	116
		Enable	115
		Output	114
100	P23	Input	113
		Enable	112
		Output	111
101	P22	Input	110
		Enable	109
		Output	108
102	P21	Input	107
		Enable	106
		Output	105

Pin No.	Pin Name	Input/Output	Bit No.
103	P20	Input	104
		Enable	103
		Output	102
104	P17	Input	101
		Enable	100
		Output	99
105	P16	Input	98
		Enable	97
		Output	96
106	P15	Input	95
		Enable	94
		Output	93
107	P14	Input	92
		Enable	91
		Output	90
108	P13	Input	89
		Enable	88
		Output	87
109	P12	Input	86
		Enable	85
		Output	84
110	P11	Input	83
		Enable	82
		Output	81
112	P10	Input	80
		Enable	79
		Output	78
113	PB7	Input	77
		Enable	76
		Output	75
114	PB6	Input	74
		Enable	73
		Output	72
115	PB5	Input	71
		Enable	70
		Output	69
116	PB4	Input	68
		Enable	67
		Output	66

Pin No.	Pin Name	Input/Output	Bit No.
117	PB3	Input	65
		Enable	64
		Output	63
118	PB2	Input	62
		Enable	61
		Output	60
119	PB1	Input	59
		Enable	58
		Output	57
120	PB0	Input	56
		Enable	55
		Output	54
121	P30	Input	53
		Enable	52
		Output	51
122	P31	Input	50
		Enable	49
		Output	48
123	P32	Input	47
		Enable	46
		Output	45
124	P33	Input	44
		Enable	43
		Output	42
125	P34	Input	41
		Enable	40
		Output	39
126	P35	Input	38
		Enable	37
		Output	36
127	P36	Input	35
		Enable	34
		Output	33
128	P37	Input	32
		Enable	31
		Output	30
129	P80	Input	29
		Enable	28
		Output	27

Pin No.	Pin Name	Input/Output	Bit No.
130	P81	Input	26
		Enable	25
		Output	24
131	P82	Input	23
		Enable	22
		Output	21
132	P83	Input	20
		Enable	19
		Output	18
133	P84	Input	17
		Enable	16
		Output	15
134	P85	Input	14
		Enable	13
		Output	12
135	P86	Input	11
		Enable	10
		Output	9
136	P40	Input	8
		Enable	7
		Output	6
137	P41	Input	5
		Enable	4
		Output	3
138	P42	Input	2
		Enable	1
		Output	0
		To ETDO	

Note: The enable signals are active-high. When an enable signal is driven high, the corresponding pin is driven with the output value.

22.3.4 ID Code Register (SDIDR)

SDIDR is a 32-bit register. In IDCODE mode, SDIDR can output H'0036200F, which is a fixed code, from ETDO. However, no serial data can be written to SDIDR via ETDI.

31 28	27 12	11 1	0
0000	0000 0011 0010 0010	0000 0000 111	1
Version (4 bits)	Part Number (16 bits)	Manufacture Identify (11 bits)	Fixed Code (1 bit)

22.4 Operation

22.4.1 TAP Controller State Transitions

Figure 22.2 shows the internal states of the TAP controller. State transitions basically conform to the IEEE1149.1 standard.

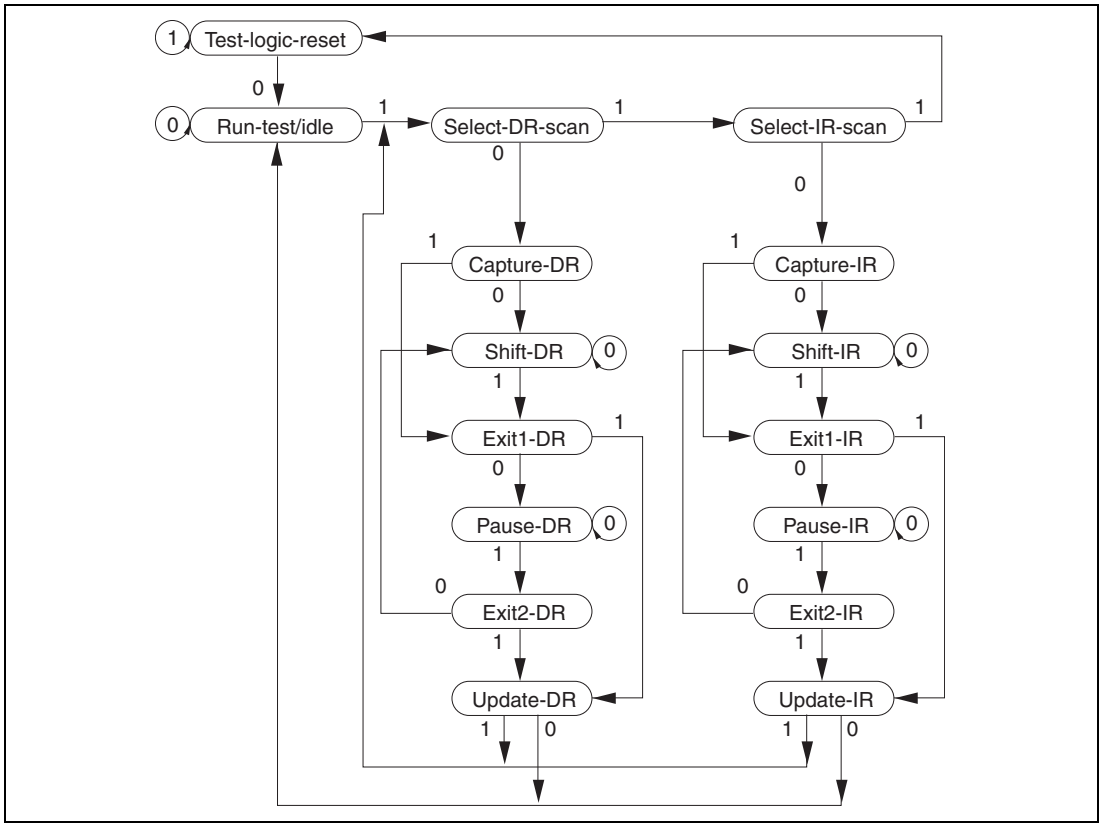


Figure 22.2 TAP Controller State Transitions

22.4.2 JTAG Reset

The JTAG can be reset in two ways.

- The JTAG is reset when the $\overline{\text{ETRST}}$ pin is held at 0.
- When $\overline{\text{ETRST}}$ = 1, the JTAG can be reset by inputting at least five ETCK clock cycles while ETMS = 1.

22.5 Boundary Scan

The JTAG pins can be placed in the boundary scan mode stipulated by the IEEE1149.1 standard by setting a command in SDIR.

22.5.1 Supported Instructions

This LSI supports the three essential instructions defined in the IEEE1149.1 standard (BYPASS, SAMPLE/PRELOAD, and EXTEST) and optional instructions (CLAMP, HIGHZ, and IDCODE).

(1) BYPASS

Instruction code: B'1111

The BYPASS instruction is an instruction that operates the bypass register. This instruction shortens the shift path to speed up serial data transfer involving other chips on the printed circuit board. While this instruction is being executed, the test circuit has no effect on the system circuits.

(2) SAMPLE/PRELOAD

Instruction code: B'0100

The SAMPLE/PRELOAD instruction inputs values from this LSI internal circuitry to the boundary scan register, outputs values from the scan path, and loads data onto the scan path. When this instruction is being executed, this LSI's input pin signals are transmitted directly to the internal circuitry, and internal circuit values are directly output externally from the output pins. This LSI system circuits are not affected by execution of this instruction.

In a SAMPLE operation, a snapshot of a value to be transferred from an input pin to the internal circuitry, or a value to be transferred from the internal circuitry to an output pin, is latched into the boundary scan register and read from the scan path. Snapshot latching does not affect normal operation of this LSI.

In a PRELOAD operation, an initial value is set in the parallel output latch of the boundary scan register from the scan path prior to the EXTEST instruction. Without a PRELOAD operation, when the EXTEST instruction was executed an undefined value would be output from the output pin until completion of the initial scan sequence (transfer to the output latch) (with the EXTEST instruction, the parallel output latch value is constantly output to the output pin).

(3) EXTEST

Instruction code: B'0000

The EXTEST instruction is provided to test external circuitry when this LSI is mounted on a printed circuit board. When this instruction is executed, output pins are used to output test data (previously set by the SAMPLE/PRELOAD instruction) from the boundary scan register to the printed circuit board, and input pins are used to latch test results into the boundary scan register from the printed circuit board. If testing is carried out by using the EXTEST instruction N times, the Nth test data is scanned in when test data (N-1) is scanned out.

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

(4) CLAMP

Instruction code: B'0010

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

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

(5) HIGHZ

Instruction code: B'0011

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

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

(6) IDCODE

Instruction code: B'1110

When the IDCODE instruction is enabled, the value of the ID code register is output from the ETDO pin with LSB first when the TAP controller is in the Shift-DR state. While the IDCODE instruction is being executed, the test circuit does not affect the system circuit.

When the TAP controller is in the Test-Logic-Reset state, the instruction register is initialized to the IDCODE instruction.

22.5.2 Notes

1. Boundary scan mode does not cover power-supply-related pins (VCC, VCL, VSS, AVCC, AVSS, and AVref).
2. Boundary scan mode covers clock-related pins (EXTAL, XTAL, X1, and X2).
3. Boundary scan mode does not cover reset- and standby-related pins ($\overline{\text{RES}}$, $\overline{\text{STBY}}$, and $\overline{\text{RESO}}$).
4. Boundary scan mode does not cover JTAG-related pins (ETCK, ETDI, ETDO, ETMS, and $\overline{\text{ETRST}}$).
5. Fix the MD2 pin low.

22.6 Usage Notes

1. A reset must always be executed by driving the $\overline{\text{ETRST}}$ pin to 0, regardless of whether or not the JTAG is to be activated. The $\overline{\text{ETRST}}$ pin must be held low for 20 ETCK clock cycles. For details, see section 26, Electrical Characteristics. To activate the JTAG after a reset, drive the $\overline{\text{ETRST}}$ pin to 1 and specify the ETCK, ETMS, and ETDI pins to any value. If the JTAG is not to be activated, drive the $\overline{\text{ETRST}}$, ETCK, ETMS, and ETDI pins to 1 or the high-impedance state. These pins are internally pulled up and are noted in standby mode.
2. The following must be considered when the power-on reset signal is applied to the $\overline{\text{ETRST}}$ pin.
   - The reset signal must be applied at power-on.
   - To prevent the LSI system operation from being affected by the $\overline{\text{ETRST}}$ pin of the board tester, circuits must be separated.
   - Alternatively, to prevent the $\overline{\text{ETRST}}$ pin of the board tester from being affected by the LSI system reset, circuits must be separated.

   Figure 22.3 shows a design example of the reset signal circuit wherein no reset signal interference occurs.

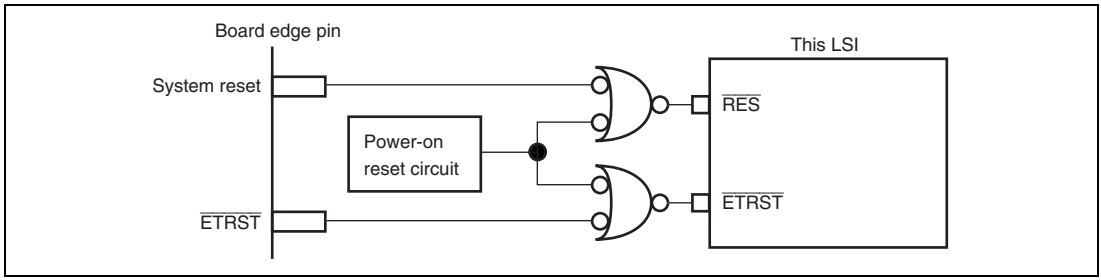


Figure 22.3 Reset Signal Circuit without Reset Signal Interference

3. The registers are not initialized in standby mode. If the $\overline{\text{ETRST}}$ pin is set to 0 in standby mode, IDCODE mode will be entered.
4. The frequency of the ETCK pin must be lower than that of the system clock. For details, see section 26, Electrical Characteristics.
5. Data input/output in serial data transfer starts from the LSB. Figure 22.4 and 22.5 shows examples of serial data input/output.
6. When data that exceeds the number of bits of the register connected between the ETDI and ETDO pins is serially transferred, the serial data that exceeds the number of register bits and output from the ETDO pin is the same as that input from the ETDI pin.
7. If the JTAG serial transfer sequence is disrupted, the $\overline{\text{ETRST}}$ pin must be reset. Transfer should then be retried, regardless of the transfer operation.
8. If a pin with a pull-up function is sampled while its pull-up function is enabled, 1 can be detected at the corresponding input scan register. In this case, the corresponding enable scan register should be cleared to 0.
9. If a pin with an open-drain function is sampled while its open-drain function is enabled and its corresponding output scan register is 1, 0 can be detected at the corresponding enable scan register.

SDIR serial data input/output

SDIR is captured into the shift register in Capture-IR, and bits 0 to 31 of SDIR are output in that order from the ETDO pin in Shift-IR.

Data input from the ETDI pin is written to SDIR in Update-IR.

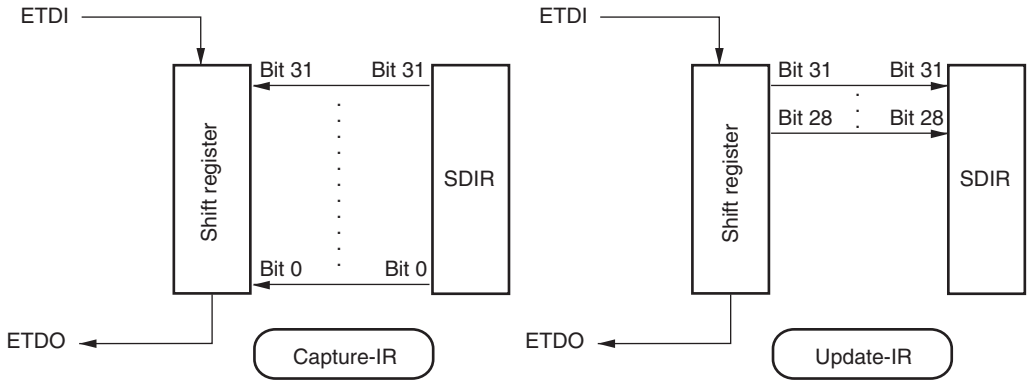


Figure 22.4 Serial Data Input/Output (1)

SDIDR serial data input/output

SDIDR is captured into the shift register in Capture-DR in IDCODE mode, and bits 0 to 31 of SDIDR are output in that order from the ETDO pin in Shift-DR.

Data input from the ETDI pin is not written to any register in Update-DR.

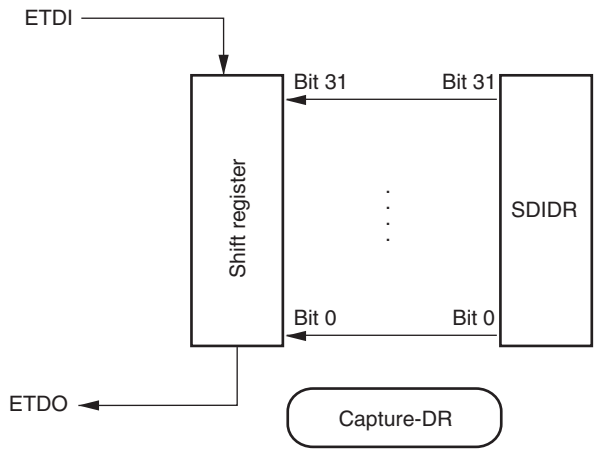


Figure 22.5 Serial Data Input/Output (2)

Section 23 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, medium-speed clock divider, bus master clock select circuit, subclock input circuit, and subclock waveform forming circuit. Figure 23.1 shows a block diagram of the clock pulse generator.

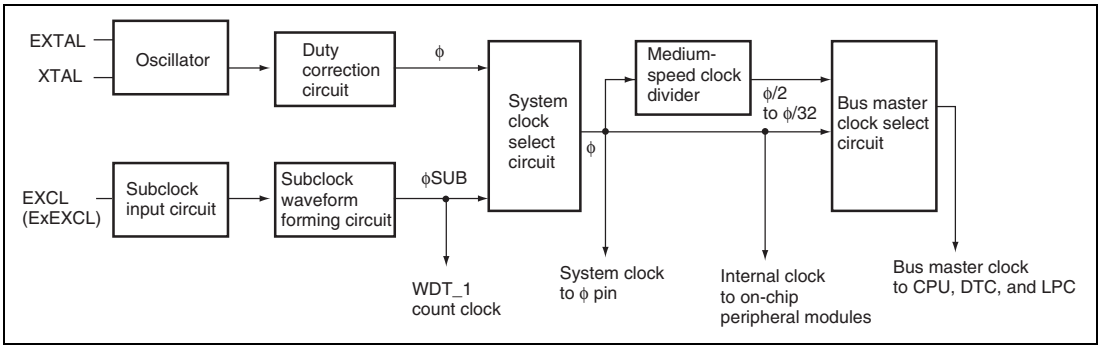


Figure 23.1 Block Diagram of Clock Pulse Generator

In high-speed mode or medium-speed mode, the bus master clock is selected by software according to the settings of the SCK2 to SCK0 bits in the standby control register (SBYCR). For details on SBYCR, see section 24.1.1, Standby Control Register (SBYCR).

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 24.1.2, Low-Power Control Register (LPWRCR). For details on PTCNT0, see section 8.17.1, Port Control Register 0 (PTCNT0).

23.1 Oscillator

Clock pulses can be supplied either by connecting a crystal resonator or by providing external clock input.

23.1.1 Connecting Crystal Resonator

Figure 23.2 shows a typical method for connecting a crystal resonator. An appropriate damping resistance R_d, given in table 23.1 should be used. An AT-cut parallel-resonance crystal resonator should be used.

Figure 23.3 shows an equivalent circuit of a crystal resonator. A crystal resonator having the characteristics given in table 23.2 should be used.

The frequency of the crystal resonator should be the same as that of the system clock (ϕ).

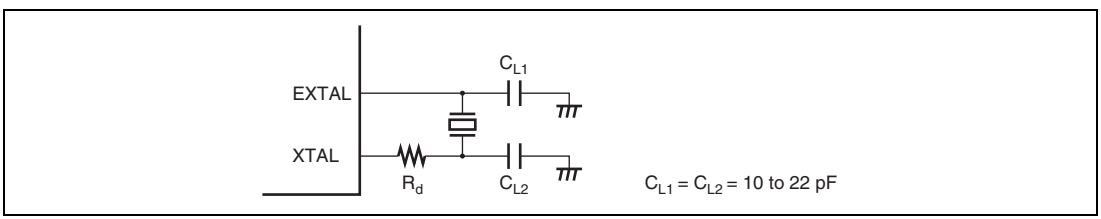


Figure 23.2 Typical Connection to Crystal Resonator

Table 23.1 Damping Resistor Values

Frequency (MHz)	4	8	10	12	16	20
R_d (Ω)	500	200	0	0	0	0

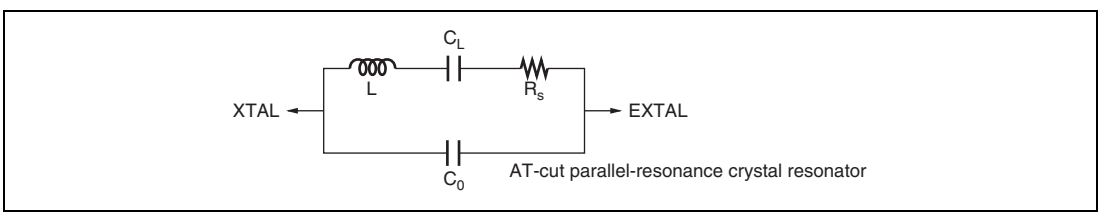


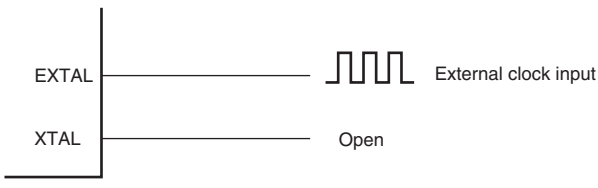
Figure 23.3 Equivalent Circuit of Crystal Resonator

Table 23.2 Crystal Resonator Parameters

Frequency (MHz)	4	8	10	12	16	20
R_S (max) (Ω)	120	80	70	60	50	40
C_0 (max) (pF)	7	7	7	7	7	7

23.1.2 External Clock Input Method

Figure 23.4 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, subactive mode, subsleep mode, and watch mode. External clock input conditions are shown in table 23.3. The frequency of the external clock should be the same as that of the system clock (ϕ).



(a) Example of external clock input when XTAL pin is left open

(b) Example of external clock input when an inverted clock is input to XTAL pin

Figure 23.4 Example of External Clock Input

Table 23.3 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 23.5	
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}	$\phi \geq 5$ MHz	Figure 25.4
		80	—	ns	$\phi < 5$ MHz	
Clock pulse width high level	t_{CH}	0.4	0.6	t_{cyc}	$\phi \geq 5$ MHz	
		80	—	ns	$\phi < 5$ MHz	

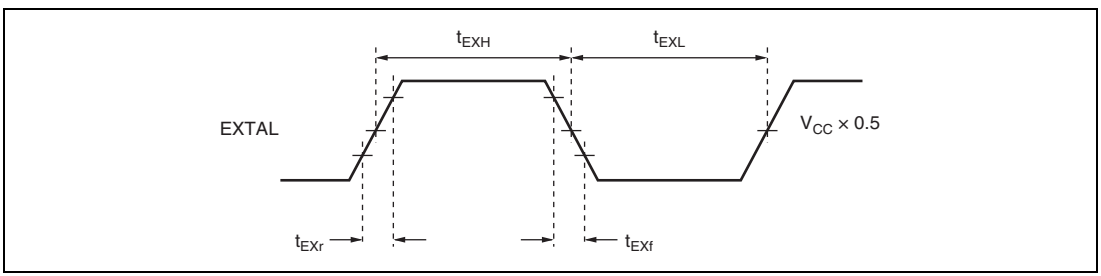


Figure 23.5 External Clock Input Timing

The oscillator and duty correction circuit can adjust the waveform of the external clock input that is input from the EXTAL pin.

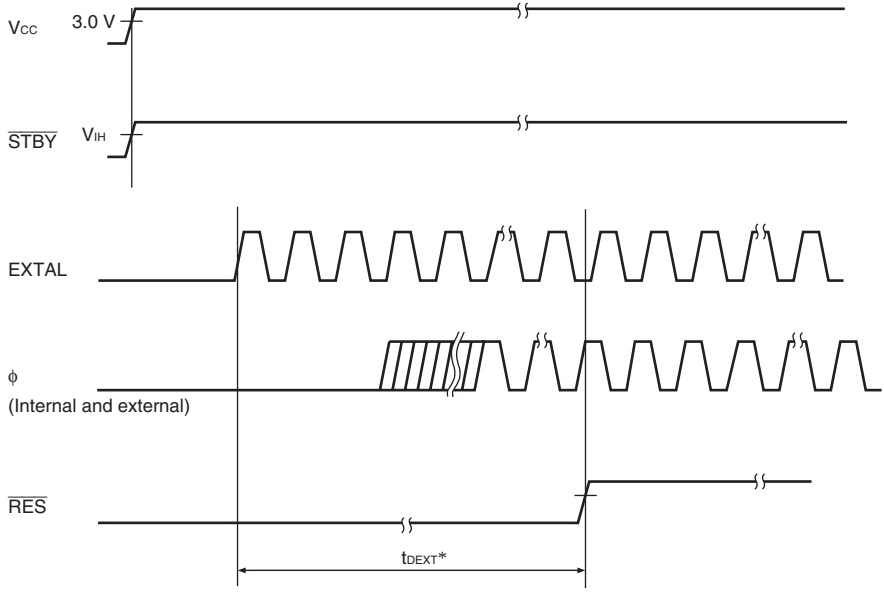
When a specified clock signal is input to the EXTAL pin, internal clock signal output is determined after the external clock output stabilization delay time (t_{DEXT}) has passed. As the clock signal output is not determined during the t_{DEXT} cycle, a reset signal should be set to low to maintain the reset state. Table 23.4 shows the external clock output stabilization delay time. Figure 23.6 shows the timing of the external clock output stabilization delay time.

Table 23.4 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 23.6

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 23.6 Timing of External Clock Output Stabilization Delay Time

23.2 Duty Correction Circuit

The duty correction circuit generates the system clock (ϕ) by correcting the duty of the clock output from the oscillator.

23.3 Medium-Speed Clock Divider

The medium-speed clock divider divides the system clock (ϕ), and generates ϕ/2, ϕ/4, ϕ/8, ϕ/16, and ϕ/32 clocks.

23.4 Bus Master Clock Select Circuit

The bus master clock select circuit selects a clock to supply to the bus master from either the system clock (ϕ) or medium-speed clock (ϕ/2, ϕ/4, ϕ/8, ϕ/16, or ϕ/32) by the SCK2 to SCK0 bits in SBYCR.

23.5 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 23.7 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.

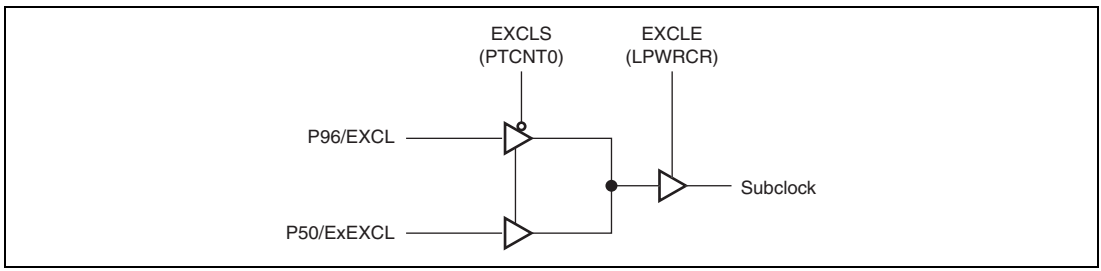


Figure 23.7 Subclock Input from EXCL Pin and ExEXCL Pin

Subclock input conditions are shown in table 23.5. When the subclock is not used, subclock input should not be enabled.

Table 23.5 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 23.8
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	

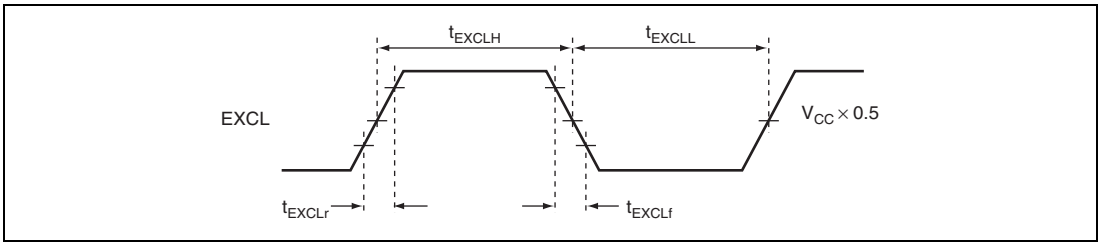


Figure 23.8 Subclock Input Timing

23.6 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 subactive mode, subsleep mode, or watch mode.

23.7 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, medium-speed mode, sleep mode, the reset state, or standby mode.

In subactive mode, subsleep mode, or 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 the CPU, TMR_0, TMR_1, WDT_0, WDT_1, I/O ports, and interrupt controller and their functions operate on the ϕSUB clock. The count clock and sampling clock for each timer are divided ϕSUB clocks.

23.8 Handling of X1 and X2 Pins

The X1 and X2 pins should be open, as shown in figure 23.9.

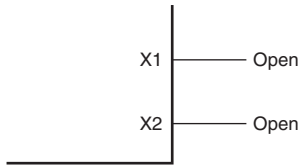


Figure 23.9 Handling of X1 and X2 Pins

23.9 Usage Notes

23.9.1 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.

23.9.2 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 23.10.

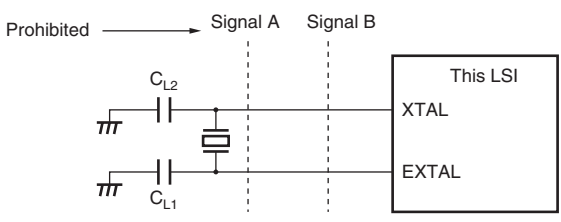


Figure 23.10 Note on Board Design of Oscillator Section

Section 24 Power-Down Modes

For operating modes after the reset state is cancelled, this LSI has not only the normal program execution state but also seven power-down 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 $\phi/32$.
- Subactive mode

  The CPU operates based on the subclock, and on-chip peripheral modules TMR_0, TMR_1, WDT_0, and WDT_1 continue operating.
- Sleep mode

  The CPU stops but on-chip peripheral modules continue operating.
- Subsleep mode

  The CPU stops but on-chip peripheral modules TMR_0, TMR_1, WDT_0, and WDT_1 continue operating.
- Watch mode

  The CPU stops but on-chip peripheral module WDT_1 continue operating.
- Software standby mode

  The clock pulse generator stops, and the CPU and on-chip peripheral modules stop operating.
- Hardware standby mode

  The clock pulse generator stops, and the CPU and on-chip peripheral modules enter the reset state.
- Module stop mode

  Independently of above operating modes, on-chip peripheral modules that are not used can be stopped individually.

24.1 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 14.3.2,Timer Control/Status Register (TCSR).

- Standby control register (SBYCR)
- Low power control register (LPWRCR)
- Module stop control register H (MSTPCRH)
- Module stop control register L (MSTPCRL)
- Module stop control register A (MSTPCRA)

24.1.1 Standby Control Register (SBYCR)

SBYCR controls power-down modes.

Bit	Bit Name	Initial Value	R/W	Description
7	SSBY	0	R/W	Software Standby Specifies the operating mode to be entered after executing the SLEEP instruction. When the SLEEP instruction is executed in high-speed mode or medium-speed mode: 0: Shifts to sleep mode 1: Shifts to software standby mode, subactive mode, or watch mode When the SLEEP instruction is executed in subactive mode: 0: Shifts to subsleep mode 1: Shifts to watch mode or high-speed mode Note that the SSBY bit is not changed even if a mode transition occurs by an interrupt.

Bit	Bit Name	Initial Value	R/W	Description
6	STS2	0	R/W	Standby Timer Select 2 to 0
5	STS1	0	R/W	On canceling software standby mode, watch mode, or subactive 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 24.1 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.
4	STS0	0	R/W	
3	—	0	R/W	Reserved The initial value should not be changed.
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. When making a transition to subactive mode or 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
0	SCK0	0	R/W	

[Legend]

X: Don't care

Table 24.1 Operating Frequency and Wait Time

STS2	STS1	STS0	Wait Time	20 MHz	10 MHz	8 MHz	6 MHz	4 MHz	Unit
0	0	0	8192 states	0.4	0.8	1.0	1.3	2.0	ms
0	0	1	16384 states	0.8	1.6	2.0	2.7	4.1	
0	1	0	32768 states	1.6	3.3	4.1	5.5	8.2	
0	1	1	65536 states	3.3	6.6	8.2	10.9	16.4	
1	0	0	131072 states	6.6	13.1	16.4	21.8	32.8	
1	0	1	262144 states	13.1	26.2	32.8	43.6	65.6	
1	1	0	Reserved	—	—	—	—	—	—
1	1	1	16 states*	0.8	1.6	2.0	2.7	4.0	µs

Recommended specification

Note: * Setting prohibited.

24.1.2 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 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, software standby mode, or watch mode 1: Shifts directly to subactive mode, or shifts to sleep mode or software standby mode When the SLEEP instruction is executed in subactive mode: 0: Shifts to subsleep mode or watch mode 1: Shifts directly to high-speed mode, or shifts to subsleep mode

Bit	Bit Name	Initial Value	R/W	Description
6	LSON	0	R/W	Low-Speed On Flag Specifies the operating mode to be entered after executing the SLEEP instruction. This bit also controls whether to shift to high-speed mode or subactive mode when watch mode is cancelled. When the SLEEP instruction is executed in high-speed mode or medium-speed mode: 0: Shifts to sleep mode, software standby mode, or watch mode 1: Shifts to watch mode or subactive mode When the SLEEP instruction is executed in subactive mode: 0: Shifts directly to watch mode or high-speed mode 1: Shifts to subsleep mode or watch mode When watch mode is cancelled: 0: Shifts to high-speed mode 1: Shifts to subactive mode
5	NESEL	0	R/W	Noise Elimination Sampling Frequency Select Selects the frequency by which the subclock (φSUB) 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. 0: Sampling using φ/32 clock 1: Sampling using φ/4 clock
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.

24.1.3 Module Stop Control Registers H, L, and A (MSTPCRH, MSTPCRL, MSTPCRA)

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	Data transfer controller (DTC)
5	MSTP13	1	R/W	16-bit free-running timer (FRT)
4	MSTP12	1	R/W	8-bit timers (TMR_0 and TMR_1)
3	MSTP11	1	R/W	8-bit PWM timer (PWM), 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	KBU
1	MSTP1	1	R/W	TPU
0	MSTP0	1	R/W	LPC

- MSTPCRA

Bit	Bit Name	Initial Value	R/W	Corresponding Module
7	MSTPA7	0	R/W	Reserved The initial value should not be changed.
6	MSTPA6	0	R/W	Reserved The initial value should not be changed.
5	MSTPA5	0	R/W	Reserved The initial value should not be changed.
4	MSTPA4	0	R/W	Reserved The initial value should not be changed.
3	MSTPA3	0	R/W	Reserved The initial value should not be changed.
2	MSTPA2	0	R/W	Reserved The initial value should not be changed.
1	MSTPA1	0	R/W	14-bit PWM timer (PWMX)
0	MSTPA0	0	R/W	8-bit PWM timer (PWM)

MSTPCRH and MSTPCRA set 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.

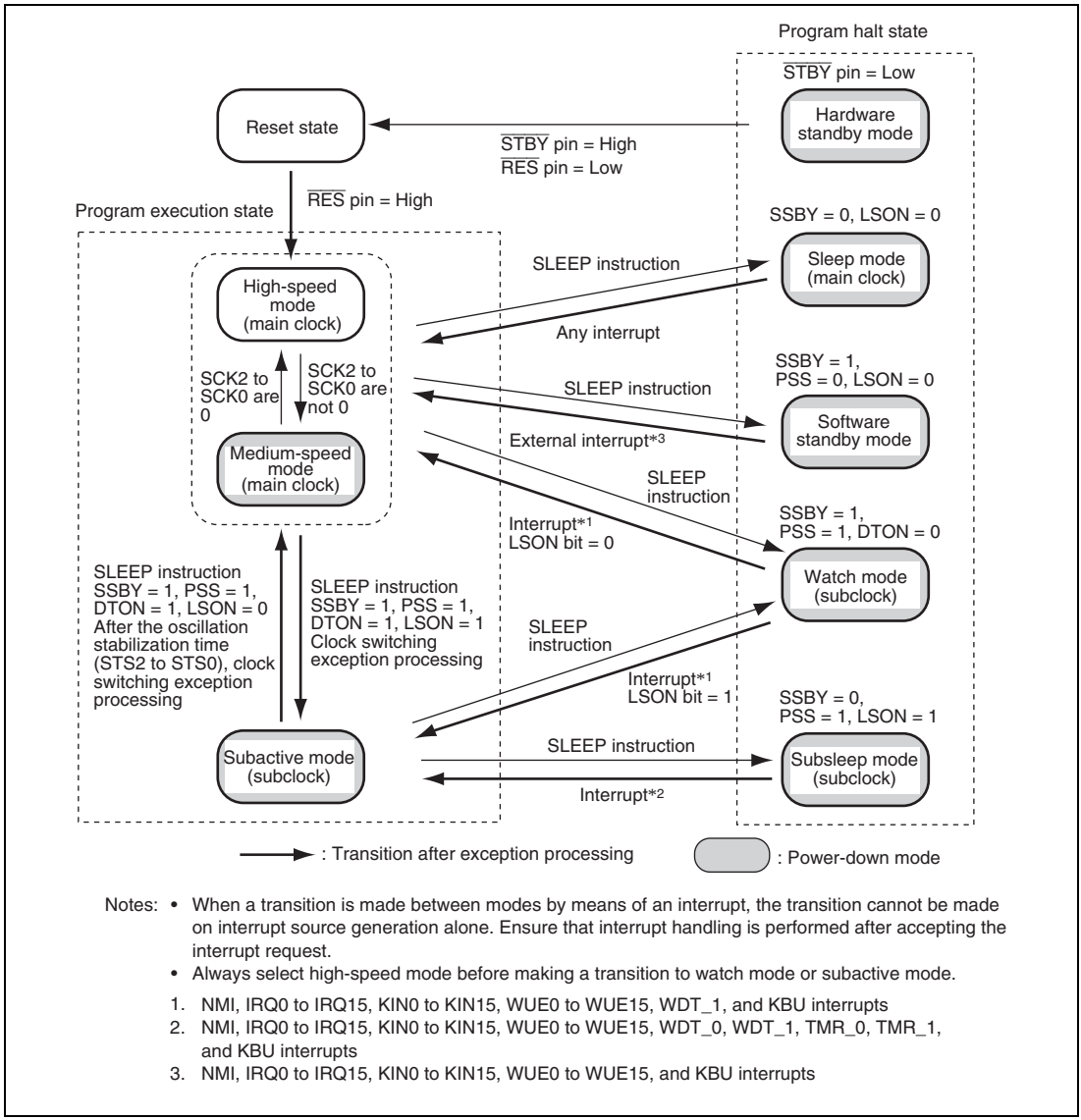
MSTPCRH: MSTP11	MSTPCRA: MSTPA0	Function
0	0	8-bit PWM timer (PWM) operates.
0	1	8-bit PWM timer (PWM) stops.
1	0	8-bit PWM timer (PWM) stops.
1	1	8-bit PWM timer (PWM) stops.

Note: The MSTP11 bit in MSTPCRH is the module stop bit of PWM and PWMX.

MSTPCRB 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.

24.2 Mode Transitions and LSI States

Figure 24.1 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 $\overline{\text{STBY}}$ input causes a mode transition from any state to hardware standby mode. The $\overline{\text{RES}}$ input causes a mode transition from a state other than hardware standby mode to the reset state. Table 24.2 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.
• Always select high-speed mode before making a transition to watch mode or subactive mode.

1. NMI, IRQ0 to IRQ15, KIN0 to KIN15, WUE0 to WUE15, WDT_1, and KBU interrupts
2. NMI, IRQ0 to IRQ15, KIN0 to KIN15, WUE0 to WUE15, WDT_0, WDT_1, TMR_0, TMR_1, and KBU interrupts
3. NMI, IRQ0 to IRQ15, KIN0 to KIN15, WUE0 to WUE15, and KBU interrupts

Figure 24.1 Mode Transition Diagram

Table 24.2 LSI Internal States in Each Operating Mode

Function		High-Speed	Medium-Speed	Sleep	Module Stop	Watch	Sub-active	Sub-sleep	Software Standby	Hardware Standby
System clock pulse generator		Functioning	Functioning	Functioning	Functioning	Halted	Halted	Halted	Halted	Halted
Subclock input		Functioning	Functioning	Functioning	Functioning	Functioning	Functioning	Functioning	Halted	Halted
CPU	Instruction execution	Functioning	Functioning in medium-speed mode	Halted	Functioning	Halted	Subclock operation	Halted	Halted	Halted
	Registers			Retained		Retained		Retained	Retained	Undefined
External interrupts	NMI	Functioning	Functioning	Functioning	Functioning	Functioning	Functioning	Functioning	Functioning	Halted
	IRQ0 to IRQ15									
	KIN0 to KIN15									
	WUE0 to WUE15									
On-chip peripheral modules	DTC	Functioning	Functioning in medium-speed mode/ Functioning	Functioning	Functioning/Halted (retained)	Halted (retained)	Halted (retained)	Halted (retained)	Halted (retained)	Halted (reset)
	WDT_1	Functioning	Functioning	Functioning	Functioning	Subclock operation	Subclock operation	Subclock operation	Halted (retained)	Halted (reset)
	WDT_0					Halted (retained)				
	TMR_0, TMR_1				Functioning/Halted (retained)					
	FRT						Halted (retained)	Halted (retained)		
	TPU									
	TMR_X, TMR_Y									
	IIC_0									
	IIC_1									
	LPC									
	KBU		Functioning in medium-speed mode/ Functioning							

Function		High-Speed	Medium-Speed	Sleep	Module Stop	Watch	Sub-active	Sub-sleep	Software Standby	Hardware Standby
On-chip peripheral modules	PWM	Function-ing	Function-ing	Function-ing	Function-ing/Halted (reset)	Halted (retained)	Halted (retained)	Halted (retained)	Halted (retained)	Halted (reset)
	PWMX									
	SCI_1					Halted (reset)	Halted (reset)	Halted (reset)	Halted (reset)	
	SCI_2									
	A/D converter									
	RAM	Function-ing	Function-ing	Function-ing (DTC)	Function-ing	Retained	Function-ing	Retained	Retained	Retained
	I/O	Function-ing	Function-ing	Function-ing	Function-ing	Retained	Function-ing	Function-ing	Retained	High impedance

Note: Halted (retained) means that the internal register values are retained and the internal state is operation suspended.

Halted (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 halted (reset or retained).

24.3 Medium-Speed Mode

The CPU makes a transition to medium-speed mode as soon as the current bus cycle ends according to the setting of the SCK2 to SCK0 bits in SBYCR. In medium-speed mode, the operating clock can be selected from ϕ/2, ϕ/4, ϕ/8, ϕ/16, or ϕ/32. On-chip peripheral modules other than the bus masters and KBU 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.

By clearing all of bits SCK2 to SCK0 to 0 in medium-speed mode, a transition is made to high-speed mode at the end of the current bus cycle.

When the SLEEP instruction is executed with the SSBY bit in SBYCR cleared to 0 and the LSON bit in LPWRCR cleared to 0, a transition is made to sleep mode. When sleep mode is cleared by an interrupt, medium-speed mode is restored. 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, a transition is made to software standby mode. When software standby mode is cleared by an external interrupt, medium-speed mode is restored.

When the $\overline{\text{RES}}$ pin is driven low, medium-speed mode is cancelled and a transition is made to the reset state. The same applies in the case of a reset caused by overflow of the watchdog timer.

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

Figure 24.2 shows an example of medium-speed mode timing.

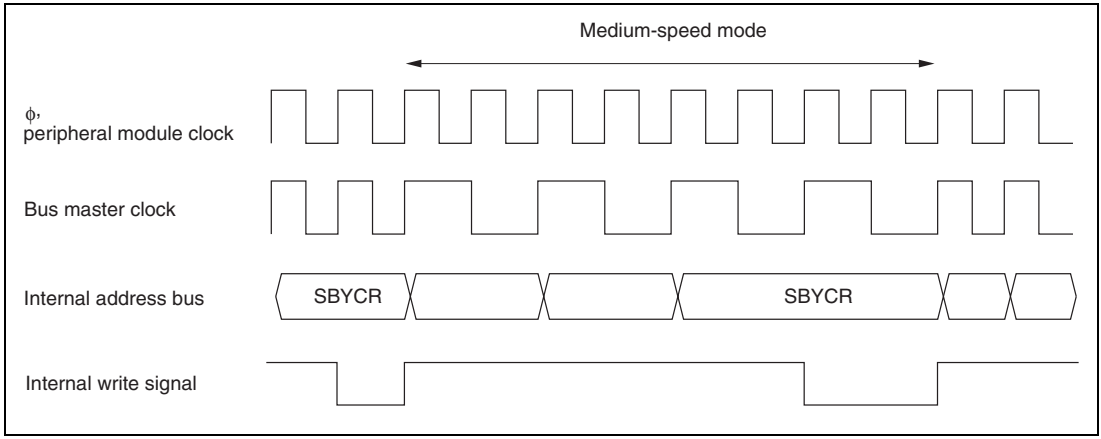


Figure 24.2 Medium-Speed Mode Timing

24.4 Sleep Mode

The CPU makes a transition to sleep mode if the SLEEP instruction is executed when the 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, the $\overline{\text{RES}}$ pin input, or the $\overline{\text{STBY}}$ 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.

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

24.5 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, PWM, 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 WUE0 to WUE15), KBU interrupt, $\overline{\text{RES}}$ pin input, or $\overline{\text{STBY}}$ 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 WUE0 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 WUE0 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.

When the $\overline{\text{STBY}}$ pin is driven low, software standby mode is cleared and a transition is made to hardware standby mode.

Figure 24.3 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.

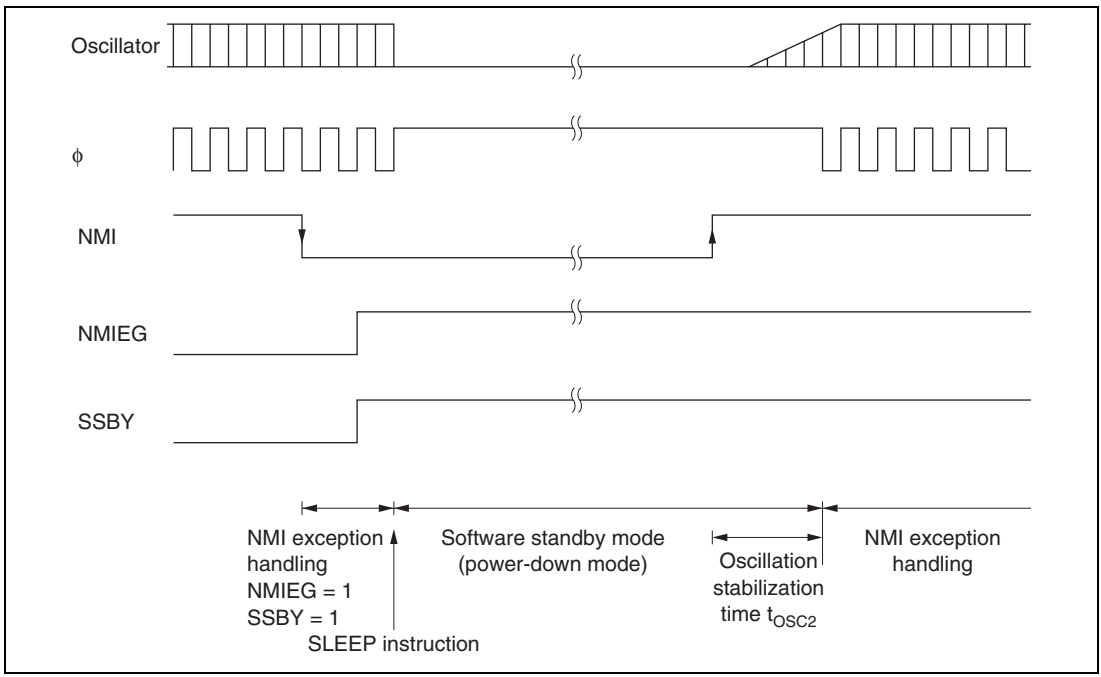


Figure 24.3 Software Standby Mode Application Example

24.6 Hardware Standby Mode

The CPU makes a transition to hardware standby mode from any mode when the $\overline{\text{STBY}}$ pin is driven low.

In hardware standby mode, all functions enter the reset state. As long as the prescribed voltage is supplied, on-chip RAM data is retained. The I/O ports are set to the high-impedance state.

In order to retain on-chip RAM data, the RAME bit in SYSCR should be cleared to 0 before driving the $\overline{\text{STBY}}$ pin low. Do not change the state of the mode pins (MD2, MD1, and MD0) while this LSI is in hardware standby mode.

Hardware standby mode is cleared by the $\overline{\text{STBY}}$ pin input or the $\overline{\text{RES}}$ pin input.

When the $\overline{\text{STBY}}$ pin is driven high while the $\overline{\text{RES}}$ pin is low, the clock pulse generator starts oscillation. Ensure that the $\overline{\text{RES}}$ pin is held low until system clock oscillation stabilizes. When the $\overline{\text{RES}}$ pin is subsequently driven high after the clock oscillation stabilization time has elapsed, reset exception handling starts.

Figure 24.4 shows an example of hardware standby mode timing.

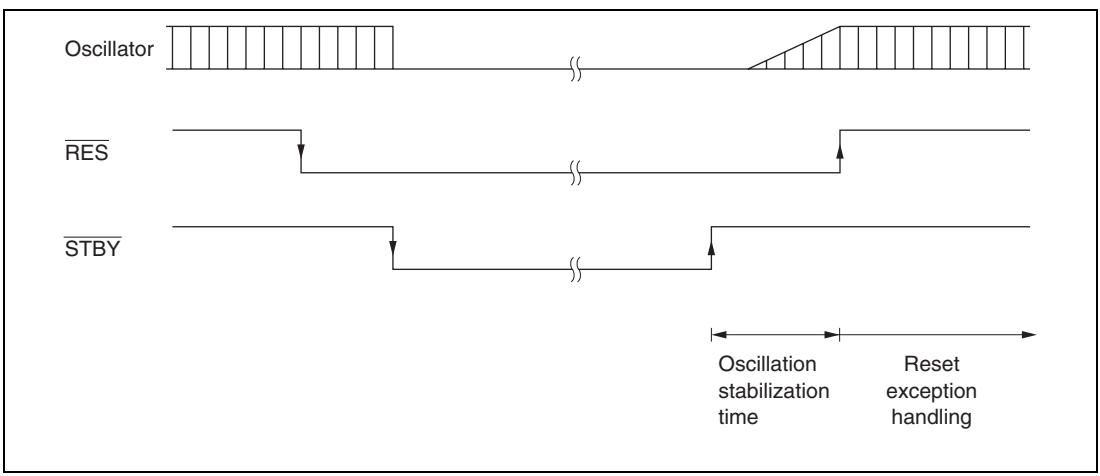


Figure 24.4 Hardware Standby Mode Timing

24.7 Watch Mode

The CPU makes a transition to watch mode when the SLEEP instruction is executed in high-speed mode or subactive 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 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 WUE0 to WUE15), KBU interrupt, $\overline{\text{RES}}$ pin input, or $\overline{\text{STBY}}$ pin input.

When an interrupt occurs, watch mode is cleared and a transition is made to high-speed mode or medium-speed mode when the LSON bit in LPWRCR cleared to 0, or a transition is made to subactive mode when the LSON bit is set to 1. 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 WUE0 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.

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

24.8 Subsleep Mode

The CPU makes a transition to subsleep mode when the SLEEP instruction is executed in subactive mode with the SSBY bit in SBYCR cleared to 0, the LSON bit in LPWRCR set to 1, and the PSS bit in TCSR (WDT_1) set to 1.

In subsleep mode, the CPU is stopped. On-chip peripheral modules other than TMR_0, TMR_1, WDT_0, and WDT_1 are also stopped. The contents of the CPU 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.

Subsleep mode is cleared by an interrupt (interrupts by on-chip peripheral modules, NMI, IRQ0 to IRQ15, KIN0 to KIN15, or WUE0 to WUE15), $\overline{\text{RES}}$ pin input, or $\overline{\text{STBY}}$ pin input.

When an interrupt occurs, subsleep mode is cleared and interrupt exception handling starts.

In the case of an IRQ0 to IRQ15 interrupt, subsleep 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 WUE0 to WUE15 interrupt, subsleep 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, subsleep 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.

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

24.9 Subactive Mode

The CPU makes a transition to subactive mode when the SLEEP instruction is executed in high-speed mode with the SSBY bit in SBYCR set to 1, the DTON bit and LSON bit in LPWRCR both set to 1, and the PSS bit in TCSR (WDT_1) set to 1. When an interrupt occurs in watch mode with the LSON bit in LPWRCR set to 1, a direct transition is made to subactive mode. Similarly, if an interrupt occurs in subsleep mode, a transition is made to subactive mode.

In subactive mode, the CPU operates at a low speed based on the subclock and sequentially executes programs. On-chip peripheral modules other than TMR_0, TMR_1, WDT_0, and WDT_1 are also stopped.

When operating the CPU in subactive mode, the SCK2 to SCK0 bits in SBYCR must all be cleared to 0.

Subactive mode is cleared by the SLEEP instruction, $\overline{\text{RES}}$ pin input, or $\overline{\text{STBY}}$ pin input.

When the SLEEP instruction is executed with the SSBY bit in SBYCR set to 1, the DTON bit in LPWRCR cleared to 0, and the PSS bit in TCSR (WDT_1) set to 1, subactive mode is cleared and a transition is made to watch mode. When the SLEEP instruction is executed with the SSBY bit in SBYCR cleared to 0, the LSON bit in LPWRCR set to 1, and the PSS bit in TCSR (WDT_1) set to 1, a transition is made to subsleep mode. When the SLEEP instruction is executed with the SSBY bit in SBYCR set to 1, the DTON bit in LPWRCR set to 1, the LSON bit in LPWRCR cleared to 0, and the PSS bit in TCSR (WDT_1) set to 1, a direct transition is made to high-speed mode.

For details on direct transitions, see section 24.11, Direct Transitions.

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.

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

24.10 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 other than the DTC 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.

24.11 Direct Transitions

The CPU executes programs in three modes: high-speed, medium-speed, and subactive. When a direct transition is made from high-speed mode to subactive mode and vice versa, there is no interruption of program execution. A direct transition is enabled by executing the SLEEP instruction after setting the DTON bit in LPWRCR to 1. After a transition, direct transition exception handling starts.

When the SLEEP instruction is executed in high-speed mode with the SSBY bit in SBYCR set to 1, the LSON bit and DTON bit in LPWRCR both set to 1, and the PSS bit in TSCR (WDT_1) set to 1, the CPU makes a direct transition to subactive mode.

When the SLEEP instruction is executed in subactive mode with the SSBY bit in SBYCR set to 1, the LSON bit in LPWRCR cleared to 0, the DTON bit in LPWRCR set to 1, and the PSS bit in TSCR (WDT_1) set to 1, after the time set in the STS2 to STS0 bits in SBYCR has elapsed, the CPU makes a direct transition to high-speed mode.

24.12 Usage Notes

24.12.1 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.

24.12.2 Current Consumption when Waiting for Oscillation Stabilization

The current consumption increases during oscillation stabilization.

24.12.3 DTC Module Stop Mode

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

After completing the DTC bus cycle, set the MSTP bit to 1 again.

Section 25 List of Registers

The list of registers gives information on the on-chip I/O 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.
- 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.
- 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.
- Register selection conditions with the RELOCATE bit in the system control register 3 (SYSCR3) cleared to 0 are indicated. For details, see section 3.2.2, System Control Register (SYSCR), section 3.2.3, Serial Timer Control Register (STCR), section 23.1.3, Module Stop Control Register H, L, A (MSTPCRH, MSTPCRL, MSTPCRA), 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.

25.1 Register Addresses (Address Order)

The data bus width indicates the numbers of bits by which the register is accessed.

The number of access states indicates the number of states based on the specified reference clock.

Register Name	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
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 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
RAM buffer address register	RBUFAR	8	H'FDE0	LPC	8	2
Erase block register	EBLKR	8	H'FDE1	LPC	8	2
LMC status register 1	LMCST1	8	H'FDE2	LPC	8	2
LMC status register 2	LMCST2	8	H'FDE3	LPC	8	2
LMC control register 1	LMCCR1	8	H'FDE4	LPC	8	2
LMC control register 2	LMCCR2	8	H'FDE5	LPC	8	2
On-chip RAM protect control register	MPCR	8	H'FDE6	LPC	8	2
Host base address register 1H	HBAR1H	8	H'FDE8	LPC	8	2
Host base address register 1L	HBAR1L	8	H'FDE9	LPC	8	2
Host base address register 2H	HBAR2H	8	H'FDEA	LPC	8	2
Host base address register 2L	HBAR2L	8	H'FDEB	LPC	8	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
On-chip RAM host base address register H	RAMBARH	8	H'FDEC	LPC	8	2
On-chip RAM host base address register L	RAMBARL	8	H'FDED	LPC	8	2
Address space set register	ASSR	8	H'FDEE	LPC	8	2
On-chip RAM address space set register	RAMASSR	8	H'FDEF	LPC	8	2
Slave address register 1	SAR1	8	H'FDF0	LPC	8	2
Slave address register 2	SAR2	8	H'FDF1	LPC	8	2
Flash memory write protect register H	FWPRH	8	H'FDF2	LPC	8	2
Flash memory write protect register M	FWPRM	8	H'FDF3	LPC	8	2
Flash memory write protect register L	FWPRL	8	H'FDF4	LPC	8	2
On-chip RAM address initial value register	RAMAR	8	H'FDF5	LPC	8	2
Flash memory read protect register H	FRPRH	8	H'FDF6	LPC	8	2
Flash memory read protect register M	FRPRM	8	H'FDF7	LPC	8	2
Flash memory read protect register L	FRPRL	8	H'FDF8	LPC	8	2
User command data register	UCMDTR	8	H'FDF9	LPC	8	2
Flash memory programming address register H	FLWARH	8	H'FDFA	LPC	8	2
Flash memory programming address register L	FLWARL	8	H'FDFB	LPC	8	2
Manufacture ID code register	LMCMIDCR	8	H'FDFC	LPC	8	2
Device ID code register	LMCDIDCR	8	H'FDFD	LPC	8	2
Port 6 noise canceller enable register	P6NCE	8	H'FE00	PORT	8	2
Port 6 noise canceller mode control register	P6NCMC	8	H'FE01	PORT	8	2
Port 6 noise cancel cycle setting register	P6NCCS	8	H'FE02	PORT	8	2
Port C noise canceller enable register	PCNCE	8	H'FE03	PORT	8	2
Port C noise canceller mode control register	PCNCMC	8	H'FE04	PORT	8	2
Port C noise cancel cycle setting register	PCNCCS	8	H'FE05	PORT	8	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
Port G noise canceller enable register	PGNCE	8	H'FE06	PORT	8	2
Port G noise canceller mode control register	PGNCMC	8	H'FE07	PORT	8	2
Port G noise cancel cycle setting register	PGNCCS	8	H'FE08	PORT	8	2
Port control register 0	PTCNT0	8	H'FE10	PORT	8	2
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	PORT	8	2
Port G Nch-OD control register	PGNOCR	8	H'FE16	PORT	8	2
Port F Nch-OD control register	PFNOCR	8	H'FE19	PORT	8	2
Port C Nch-OD control register	PCNOCR	8	H'FE1C	PORT	8	2
Port D Nch-OD control register	PDNOCR	8	H'FE1D	PORT	8	2
Bidirectional data register 0 MW	TWR0MW	8	H'FE20	LPC	8	2
Bidirectional data register 0 SW	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

Register Name	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
Status register 3	STR3	8	H'FE32	LPC	8	2
LPC channel address register H	LADR3H	8	H'FE34	LPC	8	2
LPC channel 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
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
Wake-up event interrupt mask register B	WUEMRB	8	H'FE44	INT	8	2
Wake-up event interrupt mask register	WUEMR	8	H'FE45	INT	8	2
Port G output data register	PGODR	8	H'FE46	PORT	8	2
Port G input data register	PGPIN	8	H'FE47 (read)	PORT	8	2
Port G data direction register	PGDDR	8	H'FE47 (write)	PORT	8	2
Port E pull-up MOS control register	PEPCR	8	H'FE48	PORT	8	2
Port F output data register	PFODR	8	H'FE49	PORT	8	2
Port E input data register	PEPIN	8	H'FE4A (read) (writing prohibited)	PORT	8	2
Port F input data register	PFPIN	8	H'FE4B (read)	PORT	8	2
Port F data direction register	PFDDR	8	H'FE4B (write)	PORT	8	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
Port C output data register	PCODR	8	H'FE4C	PORT	8	2
Port D output data register	PDODR	8	H'FE4D	PORT	8	2
Port C input data register	PCPIN	8	H'FE4E (read)	PORT	8	2
Port C data direction register	PCDDR	8	H'FE4E (write)	PORT	8	2
Port D input data register	PDPIN	8	H'FE4F (read)	PORT	8	2
Port D data direction register	PDDDR	8	H'FE4F (write)	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

Register Name	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
Keyboard matrix interrupt mask register	KMIMR	8	H'FE81 (RELOCATE = 1)	INT	8	2
Pull-up MOS control register	KMPCR	8	H'FE82 (RELOCATE = 1)	PORT	8	2
Keyboard matrix interrupt mask register A	KMIMRA	8	H'FE83 (RELOCATE = 1)	INT	8	2
Interrupt control register D	ICRD	8	H'FE87	INT	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 erase 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
Flash transfer destination address register	FTDAR	8	H'FEAE	ROM	8	2
Timer start register	TSTR	8	H'FEB0	TPU	8	2
Timer synchro register	TSYR	8	H'FEB1	TPU	8	2
Keyboard control register 1_0	KBCR1_0	8	H'FEC0	KBU_0	8	2
Keyboard buffer transmit data register_0	KBTR_0	8	H'FEC1	KBU_0	8	2
Keyboard control register 1_1	KBCR1_1	8	H'FEC2	KBU_1	8	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
Keyboard buffer transmit data register_1	KBTR_1	8	H'FEC3	KBU_1	8	2
Keyboard control register 1_2	KBCR1_2	8	H'FEC4	KBU_2	8	2
Keyboard buffer transmit data register_2	KBTR_2	8	H'FEC5	KBU_2	8	2
Timer XY control register	TCR_XY	8	H'FEC6	TMR_XY	8	2
Timer control register_,x	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
Timer input select register	TISR	8	H'FECD (RELOCATE = 1)	TMR_Y	8	2
I2C 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
I2C 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
I2C bus control register_1	ICCR_1	8	H'FED0 (RELOCATE = 1)	IIC_1	8	2
I2C bus status register_1	ICSR_1	8	H'FED1 (RELOCATE = 1)	IIC_1	8	2
I2C bus extended control register_0	ICXR_0	8	H'FED4	IIC_0	8	2
I2C bus extended control register_1	ICXR_1	8	H'FED5	IIC_1	8	2
Keyboard control register H_0	KBCRH_0	8	H'FED8	KBU_0	8	2
Keyboard control register L_0	KBCRL_0	8	H'FED9	KBU_0	8	2
Keyboard data buffer register_0	KBBR_0	8	H'FEDA	KBU_0	8	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
Keyboard control register 2_0	KBCR2_0	8	H'FEDB	KBU_0	8	2
Keyboard control register H_1	KBCRH_1	8	H'FEDC	KBU_1	8	2
Keyboard control register L_1	KBCRL_1	8	H'FEDD	KBU_1	8	2
Keyboard data buffer register_1	KBBR_1	8	H'FEDE	KBU_1	8	2
Keyboard control register 2_1	KBCR2_1	8	H'FEDF	KBU_1	8	2
Keyboard control register H_2	KBCRH_2	8	H'FEE0	KBU_2	8	2
Keyboard control register L_2	KBCRL_2	8	H'FEE1	KBU_2	8	2
Keyboard data buffer register_2	KBBR_2	8	H'FEE2	KBU_2	8	2
Keyboard control register 2_1	KBCR2_2	8	H'FEE3	KBU_2	8	2
Keyboard comparator control register	KBCOMP	8	H'FEE4	IrDA	8	2
DDC switch register	DDCSWR	8	H'FEE6	IIC_0, IIC_1	8	2
Interrupt control register A	ICRA	8	H'FEE8	INT	8	2
Interrupt control register B	ICRB	8	H'FEE9	INT	8	2
Interrupt control register C	ICRC	8	H'FEEA	INT	8	2
IRQ status register	ISR	8	H'FEEB	INT	8	2
IRQ sense control register H	ISCRH	8	H'FEEC	INT	8	2
IRQ sense control register L	ISCRL	8	H'FEED	INT	8	2
DTC enable register A	DTCERA	8	H'FEEE	DTC	8	2
DTC enable register B	DTCERB	8	H'FEEF	DTC	8	2
DTC enable register C	DTCERC	8	H'FEF0	DTC	8	2
DTC enable register D	DTCERD	8	H'FEF1	DTC	8	2
DTC enable register E	DTCERE	8	H'FEF2	DTC	8	2
DTC vector register	DTVECR	8	H'FEF3	DTC	8	2
Address break control register	ABRKCR	8	H'FEF4	INT	8	2
Break address register A	BARA	8	H'FEF5	INT	8	2
Break address register ,a	BARB	8	H'FEF6	INT	8	2
Break address register ,b	BARC	8	H'FEF7	INT	8	2
IRQ enable register 16	IER16	8	H'FEF8	INT	8	2
IRQ status register 16	ISR16	8	H'FEF9	INT	8	2
IRQ sense control register 16 H	ISCR16H	8	H'FEFA	INT	8	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
IRQ sense control register 16 L	ISCR16L	8	H'FEFB	INT	8	2
IRQ sense port select register 16	ISSR16	8	H'FEFC	INT	8	2
IRQ sense port select register	ISSR	8	H'FEFD	INT	8	2
Peripheral clock select register	PCSR	8	H'FF82	PWM, PWMX	8	2
System control register 2	SYSCR2	8	H'FF83	PORT	8	2
Standby control register	SBYCR	8	H'FF84	SYSTEM	8	2
Low power control register	LPWRCR	8	H'FF85	SYSTEM	8	2
Module stop control register H	MSTPCRH	8	H'FF86	SYSTEM	8	2
Module stop control register L	MSTPCRL	8	H'FF87	SYSTEM	8	2
Serial mode register_1	SMR_1	8	H'FF88	SCI_1	8	2
I2C 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
I2C 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
I2C 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
I2C 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
Timer interrupt enable register	TIER	8	H'FF90	FRT	8	2
Timer control/status register	TCSR	8	H'FF91	FRT	8	2
Free-running counter	FRC	16	H'FF92	FRT	16	2
Output control register A	OCRA	16	H'FF94	FRT	16	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
Output control register B	OCRB	16	H'FF94	FRT	16	2
Timer control register	TCR	8	H'FF96	FRT	8	2
Timer output compare control register	TOCR	8	H'FF97	FRT	8	2
Input capture register A	ICRA	16	H'FF98	FRT	16	2
Output control register AR	OCRAR	16	H'FF98	FRT	16	2
Input capture register B	ICRB	16	H'FF9A	FRT	16	2
Output control register AF	OCRAF	16	H'FF9A	FRT	16	2
Input capture register C	ICRC	16	H'FF9C	FRT	16	2
Output compare register DM	OCRDM	16	H'FF9C	FRT	16	2
Input capture register D	ICRD	16	H'FF9E	FRT	16	2
Serial mode register_2	SMR_2	8	H'FFA0	SCI_2	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
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
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

Register Name	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
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	PORT	8	2
Port A input data register	PAPIN	8	H'FFAB	PORT	8	2
Port A data direction register	PADDR	8	H'FFAB	PORT	8	2
Port 1 pull-up MOS control register	P1PCR	8	H'FFAC	PORT	8	2
Port 2 pull-up MOS control register	P2PCR	8	H'FFAD	PORT	8	2
Port 3 pull-up MOS control register	P3PCR	8	H'FFAE	PORT	8	2
Port 1 data direction register	P1DDR	8	H'FFB0	PORT	8	2
Port 2 data direction register	P2DDR	8	H'FFB1	PORT	8	2
Port 1 data register	P1DR	8	H'FFB2	PORT	8	2
Port 2 data register	P2DR	8	H'FFB3	PORT	8	2
Port 3 data direction register	P3DDR	8	H'FFB4	PORT	8	2
Port 4 data direction register	P4DDR	8	H'FFB5	PORT	8	2
Port 3 data register	P3DR	8	H'FFB6	PORT	8	2
Port 4 data register	P4DR	8	H'FFB7	PORT	8	2
Port 5 data direction register	P5DDR	8	H'FFB8	PORT	8	2
Port 6 data direction register	P6DDR	8	H'FFB9	PORT	8	2
Port 5 data register	P5DR	8	H'FFBA	PORT	8	2
Port 6 data register	P6DR	8	H'FFBB	PORT	8	2
Port B output data register	PBODR	8	H'FFBC	PORT	8	2
Port 8 data direction register	P8DDR	8	H'FFBD	PORT	8	2
Port B input data register	PBPIN	8	H'FFBD	PORT	8	2
Port 7 input data register	P7PIN	8	H'FFBE	PORT	8	2
Port B data direction register	PBDDR	8	H'FFBE	PORT	8	2
Port 8 data register	P8DR	8	H'FFBF	PORT	8	2
Port 9 data direction register	P9DDR	8	H'FFC0	PORT	8	2
Port 9 data register	P9DR	8	H'FFC1	PORT	8	2
Interrupt enable register	IER	8	H'FFC2	INT	8	2
Serial timer control register	STCR	8	H'FFC3	SYSTEM	8	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
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
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	16	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
PWM output enable register B	PWOERB	8	H'FFD2	PWM	8	2
PWM data polarity register B	PWDPRB	8	H'FFD4	PWM	8	2
PWM register select	PWSL	8	H'FFD6	PWM	8	2
PWM data register 15 to 8	PWDR 15 to 8	8	H'FFD7	PWM	8	2
I2C bus control register_0	ICCR_0	8	H'FFD8	IIC_0	8	2
I2C bus status register_0	ICSR_0	8	H'FFD9	IIC_0	8	2
I2C 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
I2C 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
A/D data register AH	ADDRAH	8	H'FFE0	A/D Converter	8	2
A/D data register AL	ADDRAL	8	H'FFE1	A/D Converter	8	2
A/D data register BH	ADDRBH	8	H'FFE2	A/D Converter	8	2
A/D data register BL	ADDRBL	8	H'FFE3	A/D Converter	8	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
A/D data register CH	ADDRCH	8	H'FFE4	A/D Converter	8	2
A/D data register CL	ADDRCL	8	H'FFE5	A/D Converter	8	2
A/D data register DH	ADDRDH	8	H'FFE6	A/D Converter	8	2
A/D data register DL	ADDRDL	8	H'FFE7	A/D Converter	8	2
A/D control/status register	ADCSR	8	H'FFE8	A/D Converter	8	2
A/D control register	ADCR	8	H'FFE9	A/D Converter	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_,x	TCR_Y	8	H'FFF0 (RELOCATE = 0)	TMR_Y	8	2
Keyboard matrix interrupt mask 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
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 mask register A	KMIMRA	8	H'FFF3 (RELOCATE = 0)	INT	8	2
Timer counter_X	TCNT_X	8	H'FFF4	TMR_X	8	2

Register Name	Abbreviation	Number of Bits	Address	Module	Data Width	Access States
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
Timer input select register	TISR	8	H'FFF5 (RELOCATE = 0)	TMR_Y	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

25.2 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
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	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TGRA_1	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TGRB_1	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
LADR4H	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	LPC
LADR4L	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
IDR4	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
ODR4	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
STR4	DBU47	DBU46	DBU45	DBU44	C/$\overline{D}$4	DBU42	IBF4	OBF4	
HICR4	—	LPC4E	IBFIE4	—	—	—	—	—	
SIRQCR2	IEDIR3	IEDIR4	IRQ11E4	IRQ10E4	IRQ9E4	IRQ6E4	SMIE4	—	
RBUFAR	RBA15	RBA14	RBA13	RBA12	RBA11	RBA10	RBA9	RBA8	
EBLKR	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
LMCST1	FLPI	FLEI	BUFINII	USERI	FLPERR	FLEERR	—	—	
LMCST2	PROTECT	LMCBUSY	ERASEE	WRITEE	BUFTRAN	—	—	—	
LMCCR1	LMCE	LPCME	FWME	—	FLASHE	HDINIE	—	—	
LMCCR2	FLPIE	FLEIE	BUFINIIE	USERIE	WAITSEL	—	—	—	
MPCR	—	—	—	—	—	—	RAMWE	RAMRE	
HBAR1H	HB1A31	HB1A30	HB1A29	HB1A28	HB1A27	HB1A26	HB1A25	HB1A24	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
HBAR1L	HB1A23	HB1A22	HB1A21	HB1A20	HB1A19	HB1A18	HB1A17	HB1A16	LPC
HBAR2H	HB2A31	HB2A30	HB2A29	HB2A28	HB2A27	HB2A26	HB2A25	HB2A24	
HBAR2L	HB2A23	HB2A22	HB2A21	HB2A20	HB2A19	HB2A18	HB2A17	HB2A16	
RAMBARH	MRA31	MRA30	MRA29	MRA28	MRA27	MRA26	MRA25	MRA24	
RAMBARL	MRA23	MRA22	MRA21	MRA20	MRA19	MRA18	MRA17	MRA16	
ASSR	AS13	AS12	AS11	AS10	AS23	AS22	AS21	AS20	
RAMASSR	—	—	—	RAMAS4	RAMAS3	RAMAS2	RAMAS1	RAMAS0	
SAR1	SA1R23	SA1R22	SA1R21	SA1R20	SA1R19	SA1R18	SA1R17	SA1R16	
SAR2	SA2R23	SA2R22	SA2R21	SA2R20	SA2R19	SA2R18	SA2R17	SA2R16	
FWPRH	WPB23	WPB22	WPB21	WPB20	WPB19	WPB18	WPB17	WPB16	
FWPRM	WPB15	WPB14	WPB13	WPB12	WPB11	WPB10	WPB9	WPB8	
FWPRL	WPB7	WPB6	WPB5	WPB4	WPB3	WPB2	WPB1	WPB0	
RAMAR	RMR7	RMR6	RMR5	RMR4	RMR3	RMR2	RMR1	RMR0	
FRPRH	RPB23	RPB22	RPB21	RPB20	RPB19	RPB18	RPB17	RPB16	
FRPRM	RPB15	RPB14	RPB13	RPB12	RPB11	RPB10	RPB9	RPB8	
FRPRL	RPB7	RPB6	RPB5	RPB4	RPB3	RPB2	RPB1	RPB0	
UCMDTR	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
FLWARH	—	—	—	FWA19	FWA18	FWA17	FWA16	FWA15	
FLWARL	FWA14	FWA13	FWA12	FWA11	FWA10	FWA9	FWA8	FWA7	
LMCMIDCR	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
LMCDIDCR	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
P6NCE	P67NCE	P66NCE	P65NCE	P64NCE	P63NCE	P62NCE	P61NCE	P60NCE	PORT
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	
PGNCMC	PG7NCMC	PG6NCMC	PG5NCMC	PG4NCMC	PG3NCMC	PG2NCMC	PG1NCMC	PG0NCMC	
PGNCCS	—	—	—	—	—	PGNCCK2	PGNCCK1	PGNCCK0	
PTCNT0	TMCI0S	TMCI1S	TMIXS	TMIYS	TMOXS	PWMAS	PWMBS	EXCLS	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
PTCNT1	SCL0AS	SCL1AS	SCL0BS	SCL1BS	SDA0AS	SDA1AS	SDA0BS	SDA1BS	PORT
PTCNT2	LPCS	—	—	—	—	—	—	LDRQS	
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	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	LPC
TWR0SW	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TWR1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TWR2	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TWR3	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TWR4	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TWR5	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TWR6	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TWR7	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TWR8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TWR9	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TWR10	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TWR11	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TWR12	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TWR13	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TWR14	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TWR15	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
IDR3	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
ODR3	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
STR3*[2]	IBF3B	OBF3B	MWMF	SWMF	C/$\overline{D}$3	DBU32	IBF3	OBF3	
STR3*[3]	DBU37	DBU36	DBU35	DBU34	C/$\overline{D}$3	DBU32	IBF3	OBF3	
LADR3H	bit15	bi14	bit13	bit12	bit11	bit10	bit9	bit8	
LADR3L	bit7	bit6	bit5	bit4	bit3	—	bit1	TWRE	
SIRQCR0	Q/C	SELREQ	IEDIR2	SMIE3B	SMIE3A	SMIE2	IRQ12E1	IRQ1E1	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
SIRQCR1	IRQ11E3	IRQ10E3	IRQ9E3	IRQ6E3	IRQ11E2	IRQ10E2	IRQ9E2	IRQ6E2	LPC
IDR1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
ODR1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
STR1	DBU17	DBU16	DBU15	DBU14	C/D1	DBU12	IBF1	OBF1	
IDR2	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
ODR2	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
STR2	DBU27	DBU26	DBU25	DBU24	C/D2	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	
WUEMRB	WUEMR7	WUEMR6	WUEMR5	WUEMR4	WUEMR3	WUEMR2	WUEMR1	WUEMR0	INT
WUEMR	WUEMR15	WUEMR14	WUEMR13	WUEMR12	WUEMR11	WUEMR10	WUEMR9	WUEMR8	
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	
PEPCR	—	—	—	PE4PCR	PE3PCR	PE2PCR	PE1PCR	PE0PCR	
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	
PDPIN	PD7PIN	PD6PIN	PD5PIN	PD4PIN	PD3PIN	PD2PIN	PD1PIN	PD0PIN	
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	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
TIORL_0	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0	TPU_0
TIER_0	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	
TSR_0	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
TCNT_0	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TGRA_0	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TGRB_0	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TGRC_0	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TGRD_0	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
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	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TGRA_2	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TGRB_2	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
SYSCR3	—	EIVS	RELOCATE	—	—	—	—	—	SYSTEM
MSTPCRA	MSTPA7	MSTPA6	MSTPA5	MSTPA4	MSTPA3	MSTPA2	MSTPA1	MSTPA0	
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
ICRD	ICRD7	ICRD6	ICRD5	ICRD4	ICRD3	ICRD2	ICRD1	ICRD0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
DACR	TEST	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	
DACNT	DACNT7	DACNT6	DACNT5	DACNT4	DACNT3	DACNT2	DACNT1	DACNT0	
	DACNT8	DACNT9	DACNT10	DACNT11	DACNT12	DACNT13	—	REGS	
FCCS	FWE	—	—	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	
TSTR	—	—	—	—	—	CST2	CST1	CST0	TPU
TSYR	—	—	—	—	—	SYNC2	SYNC1	SYNC0	
KBCR1_0	KBTS	PS	KCIE	KTIE	—	KCIF	KBTE	KTER	KBU
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	OSX	OEY	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	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TCORB_Y	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TCNT_Y	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TISR	—	—	—	—	—	—	—	IS	
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	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
SAR_1	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS	IIC_1
ICCR_1	ICE	IEIC	MST	TRS	ACKE	BBSY	IRIC	SCP	
ICSR_1	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB	
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	KBU_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	KBU_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	
KBCRH_2	KBIOE	KCLKI	KDI	KBFSEL	KBIE	KBF	PER	KBS	KBU_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	
KBCOMP	IrE	IrCKS2	IrCKS1	IrCKS0	IrTxINV	IrRxINV	—	—	IrDA
DDCSWR	—	—	—	—	CLR3	CLR2	CLR1	CLR0	IIC_0, IIC_1
ICRA	ICRA7	ICRA6	ICRA5	ICRA4	ICRA3	ICRA2	ICRA1	ICRA0	INT
ICRB	ICR7	ICRB6	ICRB5	ICRB4	ICRB3	ICRB2	ICRB1	ICRB0	
ICRC	ICR7	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	
DTCERA	DTCEA7	DTCEA6	DTCEA5	DTCEA4	DTCEA3	DTCEA2	DTCEA1	DTCEA0	DTC
DTCERB	DTCEB7	DTCEB6	DTCEB5	DTCEB4	DTCEB3	DTCEB2	DTCEB1	DTCEB0	
DTCERC	DTCEC7	DTCEC6	DTCEC5	DTCEC4	DTCEC3	DTCEC2	DTCEC1	DTCEC0	
DTCERD	DTCED7	DTCED6	DTCED5	DTCED4	DTCED3	DTCED2	DTCED1	DTCED0	
DTCERE	DTCEE7	DTCEE6	DTCEE5	DTCEE4	DTCEE3	DTCEE2	DTCEE1	DTCEE0	
DTVECR	SWDTE	DTVEC6	DTVEC5	DTVEC4	DTVEC3	DTVEC2	DTVEC1	DTVEC0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
ABRKCR	CMF	—	—	—	—	—	—	BIE	INT
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	—	ISS5	ISS4	ISS3	ISS2	ISS1	ISS0	
PCSR	—	—	PWCKXB	PWCKXA	—	PWCKB	PWCKA	PWCKXC	PWM, PWMX
SYSCR2	KWUL1	KWUL0	P6PUE	—	—	—	—	—	PORT
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/A (GM)	CHR (BLK)	PE (PE)	O/E (O/E)	STOP (BCP1)	MP (BCP0)	CKS1 (CKS1)	CKS0 (CKS0)	SCI_1
BRR_1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
SCR_1	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR_1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
SSR_1*[1]	TDRE (TDRE)	RDRF (RDRF)	ORER (ORER)	FER (ERS)	PER (PER)	TEND (TEND)	MPB (MPB)	MPBT (MPBT)	
RDR_1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
SCMR_1	—	—	—	—	SDIR	SINV	—	SMIF	
TIER	ICIAE	ICIBE	ICICE	ICIDE	OCIAE	OCIBE	OVIE	—	FRT
TCSR	ICFA	ICFB	ICFC	ICFD	OCFA	OCFB	OVF	CCLRA	
FRC	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
OCRA/	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
OCRB	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
TCR	IEDGA	IEDGB	IEDGC	IEDGD	BUFEA	BUFEB	CKS1	CKS0	FRT
TOCR	ICRDMS	OCRAMS	ICRS	OCRS	OEA	OEB	OLVLA	OLVLB	
ICRA/	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
OCRAR	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
ICRB/	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
OCRAF	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
ICRC/	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
OCRDM	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
ICRD	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
SMR_2*[1]	C/A (GM)	CHR (BLK)	PE (PE)	O/E (O/E)	STOP (BCP1)	MP (BCP0)	CKS1 (CKS1)	CKS0 (CKS0)	SCI_2
BRR_2	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
SCR_2	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR_2	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	SCI_2
SSR_2*[1]	TDRE (TDRE)	RDRF (RDRF)	ORER (ORER)	FER (ERS)	PER (PER)	TEND (TEND)	MPB (MPB)	MPBT (MPBT)	
RDR_2	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
SCMR_2	—	—	—	—	SDIR	SINV	—	SMIF	
TCSR_0	OVF	WT/IT	TME	—	RST/NMI	CKS2	CKS1	CKS0	WDT_0
TCNT_0	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
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	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
P4DDR	P47DDR	P46DDR	P45DDR	P44DDR	P43DDR	P42DDR	P41DDR	P40DDR	PORT
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	
PBODR	PB7ODR	PB6ODR	PB5ODR	PB4ODR	PB3ODR	PB2ODR	PB1ODR	PB0ODR	
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	IICS	IICX1	IICX0	IICE	FLSHE	—	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	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TCORA_1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TCORB_0	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TCORB_1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TCNT_0	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TCNT_1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
PWOERB	OE15	OE14	OE13	OE12	OE11	OE10	OE9	OE8	PWM
PWDPRB	OS15	OS14	OS13	OS12	OS11	OS10	OS9	OS8	
PWSL	PWCKE	PWCKS	—	—	RS3	RS2	RS1	RS0	
PWDR 15 to 8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
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	
ADDRAH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D Converter
ADDRAL	AD1	AD0	—	—	—	—	—	—	
ADDRBH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
ADDRBL	AD1	AD0	—	—	—	—	—	—	
ADDRCH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
ADDRCL	AD1	AD0	—	—	—	—	—	—	
ADDRDH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
ADDRDL	AD1	AD0	—	—	—	—	—	—	A/D Converter
ADCSR	ADF	ADIE	ADST	SCAN	CKS	CH2	CH1	CH0	
ADCR	TRGS1	TRGS0	—	—	—	—	—	—	
TCSR_1	OVF	WT/IT	TME	PSS	RST/NMI	CKS2	CKS1	CKS0	WDT_1
TCNT_1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TCR_X	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	TMR_X
TCSR_X	CMFB	CMFA	OVF	ICF	OS3	OS2	OS1	OS0	
TICRR	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TICRF	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TCNT_X	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TCORC	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TCORA_X	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TCORB_X	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
TCONRI	—	—	—	ICST	—	—	—	—	

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
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 SELSTR = 3.
3. When TWRE = 1 and SELSTR = 3.

25.3 Register States in Each Operating Mode

Register Abbreviation	Reset	High-Speed/ Medium-Speed	Watch	Sleep	Sub-Active	Sub-Sleep	Module Stop	Software Standby	Hardware Standby	Module
TCR_1	Initialized	—	—	—	—	—	—	—	Initialized	TPU_1
TMDR_1	Initialized	—	—	—	—	—	—	—	Initialized	
TIOR_1	Initialized	—	—	—	—	—	—	—	Initialized	
TIER_1	Initialized	—	—	—	—	—	—	—	Initialized	
TSR_1	Initialized	—	—	—	—	—	—	—	Initialized	
TCNT_1	Initialized	—	—	—	—	—	—	—	Initialized	
TGRA_1	Initialized	—	—	—	—	—	—	—	Initialized	
TGRB_1	Initialized	—	—	—	—	—	—	—	Initialized	
LADR4H	Initialized	—	—	—	—	—	—	—	Initialized	LPC
LADR4L	Initialized	—	—	—	—	—	—	—	Initialized	
IDR4	—	—	—	—	—	—	—	—	—	
ODR4	—	—	—	—	—	—	—	—	—	
STR4	Initialized	—	—	—	—	—	—	—	Initialized	
HICR4	Initialized	—	—	—	—	—	—	—	Initialized	
SIRQCR2	Initialized	—	—	—	—	—	—	—	Initialized	
RBUFAR	Initialized	—	—	—	—	—	—	—	Initialized	
EBLKR	Initialized	—	—	—	—	—	—	—	Initialized	
LMCST1	Initialized	—	—	—	—	—	—	—	Initialized	
LMCST2	Initialized	—	—	—	—	—	—	—	Initialized	
LMCCR1	Initialized	—	—	—	—	—	—	—	Initialized	
LMCCR2	Initialized	—	—	—	—	—	—	—	Initialized	
MPCR	Initialized	—	—	—	—	—	—	—	Initialized	
HBAR1H	Initialized	—	—	—	—	—	—	—	Initialized	
HBAR1L	Initialized	—	—	—	—	—	—	—	Initialized	
HBAR2H	Initialized	—	—	—	—	—	—	—	Initialized	
HBAR2L	Initialized	—	—	—	—	—	—	—	Initialized	
RAMBARH	Initialized	—	—	—	—	—	—	—	Initialized	
RAMBARL	Initialized	—	—	—	—	—	—	—	Initialized	

Register Abbreviation	Reset	High-Speed/ Medium-Speed	Watch	Sleep	Sub-Active	Sub-Sleep	Module Stop	Software Standby	Hardware Standby	Module
ASSR	Initialized	—	—	—	—	—	—	—	Initialized	LPC
RAMASSR	Initialized	—	—	—	—	—	—	—	Initialized	
SAR1	Initialized	—	—	—	—	—	—	—	Initialized	
SAR2	Initialized	—	—	—	—	—	—	—	Initialized	
FWPRH	Initialized	—	—	—	—	—	—	—	Initialized	
FWPRM	Initialized	—	—	—	—	—	—	—	Initialized	
FWPRL	Initialized	—	—	—	—	—	—	—	Initialized	
RAMAR	Initialized	—	—	—	—	—	—	—	Initialized	
FRPRH	Initialized	—	—	—	—	—	—	—	Initialized	
FRPRM	Initialized	—	—	—	—	—	—	—	Initialized	
FRPRL	Initialized	—	—	—	—	—	—	—	Initialized	
UCMDTR	Initialized	—	—	—	—	—	—	—	Initialized	
FLWARH	Initialized	—	—	—	—	—	—	—	Initialized	
FLWARL	Initialized	—	—	—	—	—	—	—	Initialized	
LMCMIDCR	Initialized	—	—	—	—	—	—	—	Initialized	
LMCDIDCR	Initialized	—	—	—	—	—	—	—	Initialized	
P6NCE	Initialized	—	—	—	—	—	—	—	Initialized	PORT
P6NCMC	Initialized	—	—	—	—	—	—	—	Initialized	
P6NCCS	Initialized	—	—	—	—	—	—	—	Initialized	
PCNCE	Initialized	—	—	—	—	—	—	—	Initialized	
PCNCMC	Initialized	—	—	—	—	—	—	—	Initialized	
PCNCCS	Initialized	—	—	—	—	—	—	—	Initialized	
PGNCE	Initialized	—	—	—	—	—	—	—	Initialized	
PGNCMC	Initialized	—	—	—	—	—	—	—	Initialized	
PGNCCS	Initialized	—	—	—	—	—	—	—	Initialized	
PTCNT0	Initialized	—	—	—	—	—	—	—	Initialized	
PTCNT1	Initialized	—	—	—	—	—	—	—	Initialized	
PTCNT2	Initialized	—	—	—	—	—	—	—	Initialized	
P9PCR	Initialized	—	—	—	—	—	—	—	Initialized	
PGNOCR	Initialized	—	—	—	—	—	—	—	Initialized	

Register Abbreviation	Reset	High-Speed/ Medium-Speed	Watch	Sleep	Sub-Active	Sub-Sleep	Module Stop	Software Standby	Hardware Standby	Module
PFNOCR	Initialized	—	—	—	—	—	—	—	Initialized	LPC
PCNOCR	Initialized	—	—	—	—	—	—	—	Initialized	
PDNOCR	Initialized	—	—	—	—	—	—	—	Initialized	
TWR0MW	—	—	—	—	—	—	—	—	—	
TWR0SW	—	—	—	—	—	—	—	—	—	
TWR1	—	—	—	—	—	—	—	—	—	
TWR2	—	—	—	—	—	—	—	—	—	
TWR3	—	—	—	—	—	—	—	—	—	
TWR4	—	—	—	—	—	—	—	—	—	
TWR5	—	—	—	—	—	—	—	—	—	
TWR6	—	—	—	—	—	—	—	—	—	
TWR7	—	—	—	—	—	—	—	—	—	
TWR8	—	—	—	—	—	—	—	—	—	
TWR9	—	—	—	—	—	—	—	—	—	
TWR10	—	—	—	—	—	—	—	—	—	
TWR11	—	—	—	—	—	—	—	—	—	
TWR12	—	—	—	—	—	—	—	—	—	
TWR13	—	—	—	—	—	—	—	—	—	
TWR14	—	—	—	—	—	—	—	—	—	
TWR15	—	—	—	—	—	—	—	—	—	
IDR3	—	—	—	—	—	—	—	—	—	
ODR3	—	—	—	—	—	—	—	—	—	
STR3	Initialized	—	—	—	—	—	—	—	Initialized	
LADR3H	Initialized	—	—	—	—	—	—	—	Initialized	
LADR3L	Initialized	—	—	—	—	—	—	—	Initialized	
SIRQCR0	Initialized	—	—	—	—	—	—	—	Initialized	
SIRQCR1	Initialized	—	—	—	—	—	—	—	Initialized	
IDR1	—	—	—	—	—	—	—	—	—	
ODR1	—	—	—	—	—	—	—	—	—	
STR1	Initialized	—	—	—	—	—	—	—	Initialized	

Register Abbreviation	Reset	High-Speed/ Medium-Speed	Watch	Sleep	Sub-Active	Sub-Sleep	Module Stop	Software Standby	Hardware Standby	Module
IDR2	—	—	—	—	—	—	—	—	—	LPC
ODR2	—	—	—	—	—	—	—	—	—	
STR2	Initialized	—	—	—	—	—	—	—	Initialized	
HISEL	Initialized	—	—	—	—	—	—	—	Initialized	
HICR0	Initialized	—	—	—	—	—	—	—	Initialized	
HICR1	Initialized	—	—	—	—	—	—	—	Initialized	
HICR2	Initialized	—	—	—	—	—	—	—	Initialized	
HICR3	—	—	—	—	—	—	—	—	—	
WUEMRB	Initialized	—	—	—	—	—	—	—	Initialized	INT
WUEMR	Initialized	—	—	—	—	—	—	—	Initialized	
PGODR	Initialized	—	—	—	—	—	—	—	Initialized	PORT
PGPIN	—	—	—	—	—	—	—	—	—	
PGDDR	Initialized	—	—	—	—	—	—	—	Initialized	
PEPCR	Initialized	—	—	—	—	—	—	—	Initialized	
PFODR	Initialized	—	—	—	—	—	—	—	Initialized	
PEPIN	—	—	—	—	—	—	—	—	—	
PFPIN	—	—	—	—	—	—	—	—	—	
PCODR	Initialized	—	—	—	—	—	—	—	Initialized	
PDODR	Initialized	—	—	—	—	—	—	—	Initialized	
PCPIN	—	—	—	—	—	—	—	—	—	PORT
PFDDR	Initialized	—	—	—	—	—	—	—	Initialized	
PCDDR	Initialized	—	—	—	—	—	—	—	Initialized	
PDPIN	—	—	—	—	—	—	—	—	—	
PDDDR	Initialized	—	—	—	—	—	—	—	Initialized	
TCR_0	Initialized	—	—	—	—	—	—	—	Initialized	TPU_0
TMDR_0	Initialized	—	—	—	—	—	—	—	Initialized	
TIORH_0	Initialized	—	—	—	—	—	—	—	Initialized	
TIORL_0	Initialized	—	—	—	—	—	—	—	Initialized	
TIER_0	Initialized	—	—	—	—	—	—	—	Initialized	
TSR_0	Initialized	—	—	—	—	—	—	—	Initialized	

Register Abbreviation	Reset	High-Speed/ Medium-Speed	Watch	Sleep	Sub-Active	Sub-Sleep	Module Stop	Software Standby	Hardware Standby	Module
TCNT_0	Initialized	—	—	—	—	—	—	—	Initialized	TPU_0
TGRA_0	Initialized	—	—	—	—	—	—	—	Initialized	
TGRB_0	Initialized	—	—	—	—	—	—	—	Initialized	
TGRC_0	Initialized	—	—	—	—	—	—	—	Initialized	
TGRD_0	Initialized	—	—	—	—	—	—	—	Initialized	
TCR_2	Initialized	—	—	—	—	—	—	—	Initialized	TPU_2
TMDR_2	Initialized	—	—	—	—	—	—	—	Initialized	
TIOR_2	Initialized	—	—	—	—	—	—	—	Initialized	
TIER_2	Initialized	—	—	—	—	—	—	—	Initialized	
TSR_2	Initialized	—	—	—	—	—	—	—	Initialized	
TCNT_2	Initialized	—	—	—	—	—	—	—	Initialized	
TGRA_2	Initialized	—	—	—	—	—	—	—	Initialized	
TGRB_2	Initialized	—	—	—	—	—	—	—	Initialized	
SYSCR3	Initialized	—	—	—	—	—	—	—	Initialized	SYSTEM
MSTPCRA	Initialized	—	—	—	—	—	—	—	Initialized	
KMIMR	Initialized	—	—	—	—	—	—	—	Initialized	INT
KMPCR	Initialized	—	—	—	—	—	—	—	Initialized	PORT
KMIMRA	Initialized	—	—	—	—	—	—	—	Initialized	INT
ICRD	Initialized	—	—	—	—	—	—	—	Initialized	
DACR	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialized	PWMX
DADRA	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialized	PWMX
DADRB	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialized	
DACNT	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialized	
FCCS	Initialized	—	—	—	—	—	—	—	Initialized	ROM
FPCS	Initialized	—	—	—	—	—	—	—	Initialized	
FECS	Initialized	—	—	—	—	—	—	—	Initialized	
FKEY	Initialized	—	—	—	—	—	—	—	Initialized	
FMATS	Initialized	—	—	—	—	—	—	—	Initialized	
FTDAR	Initialized	—	—	—	—	—	—	—	Initialized	

Register Abbreviation	Reset	High-Speed/ Medium-Speed	Watch	Sleep	Sub-Active	Sub-Sleep	Module Stop	Software Standby	Hardware Standby	Module
TSTR	Initialized	—	—	—	—	—	—	—	Initialized	TPU
TSYR	Initialized	—	—	—	—	—	—	—	Initialized	
KBCR1_0	Initialized	—	—	—	—	—	—	—	Initialized	KBU
KBTR_0	Initialized	—	—	—	—	—	—	—	Initialized	
KBCR1_1	Initialized	—	—	—	—	—	—	—	Initialized	
KBTR_1	Initialized	—	—	—	—	—	—	—	Initialized	
KBCR1_2	Initialized	—	—	—	—	—	—	—	Initialized	
KBTR_2	Initialized	—	—	—	—	—	—	—	Initialized	
TCRXY	Initialized	—	—	—	—	—	—	—	Initialized	TMR_XY
TCR_Y	Initialized	—	—	—	—	—	—	—	Initialized	TMR_Y
TCSR_Y	Initialized	—	—	—	—	—	—	—	Initialized	
TCORA_Y	Initialized	—	—	—	—	—	—	—	Initialized	
TCORB_Y	Initialized	—	—	—	—	—	—	—	Initialized	
TCNT_Y	Initialized	—	—	—	—	—	—	—	Initialized	
TISR	Initialized	—	—	—	—	—	—	—	Initialized	
ICDR_1	—	—	—	—	—	—	—	—	—	IIC_1
SARX_1	Initialized	—	—	—	—	—	—	—	Initialized	
ICMR_1	Initialized	—	—	—	—	—	—	—	Initialized	
SAR_1	Initialized	—	—	—	—	—	—	—	Initialized	
ICCR_1	Initialized	—	—	—	—	—	—	—	Initialized	
ICSR_1	Initialized	—	—	—	—	—	—	—	Initialized	
ICXR_0	Initialized	—	—	—	—	—	—	—	Initialized	IIC_0
ICXR_1	Initialized	—	—	—	—	—	—	—	Initialized	IIC_1
KBCRH_0	Initialized	—	—	—	—	—	—	—	Initialized	KBU_0
KBCRL_0	Initialized	—	—	—	—	—	—	—	Initialized	
KBBR_0	Initialized	—	—	—	—	—	—	—	Initialized	
KBCR2_0	Initialized	—	—	—	—	—	—	—	Initialized	
KBCRH_1	Initialized	—	—	—	—	—	—	—	Initialized	KBU_1

Register Abbreviation	Reset	High-Speed/ Medium-Speed	Watch	Sleep	Sub-Active	Sub-Sleep	Module Stop	Software Standby	Hardware Standby	Module
KBCRL_1	Initialized	—	—	—	—	—	—	—	Initialized	KBU_1
KBBR_1	Initialized	—	—	—	—	—	—	—	Initialized	
KBCR2_1	Initialized	—	—	—	—	—	—	—	Initialized	
KBCRH_2	Initialized	—	—	—	—	—	—	—	Initialized	KBU_2
KBCRL_2	Initialized	—	—	—	—	—	—	—	Initialized	
KBBR_2	Initialized	—	—	—	—	—	—	—	Initialized	
KBCR2_2	Initialized	—	—	—	—	—	—	—	Initialized	
KBCOMP	Initialized	—	—	—	—	—	—	—	Initialized	IrDA
DDCSWR	Initialized	—	—	—	—	—	—	—	Initialized	IIC_0, IIC_1
ICRA	Initialized	—	—	—	—	—	—	—	Initialized	INT
ICRB	Initialized	—	—	—	—	—	—	—	Initialized	
ICRC	Initialized	—	—	—	—	—	—	—	Initialized	
ISR	Initialized	—	—	—	—	—	—	—	Initialized	
ISCRH	Initialized	—	—	—	—	—	—	—	Initialized	
ISCRL	Initialized	—	—	—	—	—	—	—	Initialized	
DTCERA	Initialized	—	—	—	—	—	—	—	Initialized	DTC
DTCERB	Initialized	—	—	—	—	—	—	—	Initialized	
DTCERC	Initialized	—	—	—	—	—	—	—	Initialized	
DTCERD	Initialized	—	—	—	—	—	—	—	Initialized	
DTCERE	Initialized	—	—	—	—	—	—	—	Initialized	
DTVECR	Initialized	—	—	—	—	—	—	—	Initialized	
ABRKCR	Initialized	—	—	—	—	—	—	—	Initialized	INT
BARA	Initialized	—	—	—	—	—	—	—	Initialized	
BARB	Initialized	—	—	—	—	—	—	—	Initialized	
BARC	Initialized	—	—	—	—	—	—	—	Initialized	
IER16	Initialized	—	—	—	—	—	—	—	Initialized	
ISR16	Initialized	—	—	—	—	—	—	—	Initialized	
ISCR16H	Initialized	—	—	—	—	—	—	—	Initialized	
ISCR16L	Initialized	—	—	—	—	—	—	—	Initialized	

Register Abbreviation	Reset	High-Speed/ Medium-Speed	Watch	Sleep	Sub-Active	Sub-Sleep	Module Stop	Software Standby	Hardware Standby	Module
ISSR16	Initialized	—	—	—	—	—	—	—	Initialized	INT
ISSR	Initialized	—	—	—	—	—	—	—	Initialized	
PCSR	Initialized	—	—	—	—	—	—	—	Initialized	PWM, PWMX
SYSCR2	Initialized	—	—	—	—	—	—	—	Initialized	PORT
SBYCR	Initialized	—	—	—	—	—	—	—	Initialized	SYSTEM
LPWRCR	Initialized	—	—	—	—	—	—	—	Initialized	
MSTPCRH	Initialized	—	—	—	—	—	—	—	Initialized	
MSTPCRL	Initialized	—	—	—	—	—	—	—	Initialized	
SMR_1	Initialized	—	—	—	—	—	—	—	Initialized	SCI_1
BRR_1	Initialized	—	—	—	—	—	—	—	Initialized	
SCR_1	Initialized	—	—	—	—	—	—	—	Initialized	
TDR_1	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialized	
SSR_1	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialized	
RDR_1	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialized	
SCMR_1	Initialized	—	—	—	—	—	—	—	Initialized	
TIER	Initialized	—	—	—	—	—	—	—	Initialized	FRT
TCSR	Initialized	—	—	—	—	—	—	—	Initialized	
FRC	Initialized	—	—	—	—	—	—	—	Initialized	
OCRA/	Initialized	—	—	—	—	—	—	—	Initialized	
OCRB	Initialized	—	—	—	—	—	—	—	Initialized	
TCR	Initialized	—	—	—	—	—	—	—	Initialized	
TOCR	Initialized	—	—	—	—	—	—	—	Initialized	
ICRA/	Initialized	—	—	—	—	—	—	—	Initialized	
OCRAR	Initialized	—	—	—	—	—	—	—	Initialized	
ICRB/	Initialized	—	—	—	—	—	—	—	Initialized	
OCRAF	Initialized	—	—	—	—	—	—	—	Initialized	
ICRC/	Initialized	—	—	—	—	—	—	—	Initialized	
OCRDM	Initialized	—	—	—	—	—	—	—	Initialized	
ICRD	Initialized	—	—	—	—	—	—	—	Initialized	

Register Abbreviation	Reset	High-Speed/ Medium-Speed	Watch	Sleep	Sub-Active	Sub-Sleep	Module Stop	Software Standby	Hardware Standby	Module
SMR_2	Initialized	—	—	—	—	—	—	—	Initialized	SCI_2
BRR_2	Initialized	—	—	—	—	—	—	—	Initialized	
SCR_2	Initialized	—	—	—	—	—	—	—	Initialized	
TDR_2	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialized	
SSR_2	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialized	
RDR_2	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialized	
SCMR_2	Initialized	—	—	—	—	—	—	—	Initialized	
TCSR_0	Initialized	—	—	—	—	—	—	—	Initialized	WDT_0
TCNT_0	Initialized	—	—	—	—	—	—	—	Initialized	
PAODR	Initialized	—	—	—	—	—	—	—	Initialized	PORT
PAPIN	—	—	—	—	—	—	—	—	—	
PADDR	Initialized	—	—	—	—	—	—	—	Initialized	
P1PCR	Initialized	—	—	—	—	—	—	—	Initialized	
P2PCR	Initialized	—	—	—	—	—	—	—	Initialized	
P3PCR	Initialized	—	—	—	—	—	—	—	Initialized	
P1DDR	Initialized	—	—	—	—	—	—	—	Initialized	
P2DDR	Initialized	—	—	—	—	—	—	—	Initialized	
P1DR	Initialized	—	—	—	—	—	—	—	Initialized	
P2DR	Initialized	—	—	—	—	—	—	—	Initialized	
P3DDR	Initialized	—	—	—	—	—	—	—	Initialized	
P4DDR	Initialized	—	—	—	—	—	—	—	Initialized	
P3DR	Initialized	—	—	—	—	—	—	—	Initialized	
P4DR	Initialized	—	—	—	—	—	—	—	Initialized	
P5DDR	Initialized	—	—	—	—	—	—	—	Initialized	
P6DDR	Initialized	—	—	—	—	—	—	—	Initialized	
P5DR	Initialized	—	—	—	—	—	—	—	Initialized	
P6DR	Initialized	—	—	—	—	—	—	—	Initialized	
PBODR	Initialized	—	—	—	—	—	—	—	Initialized	
PBPIN	—	—	—	—	—	—	—	—	—	
P8DDR	Initialized	—	—	—	—	—	—	—	Initialized	

Register Abbreviation	Reset	High-Speed/ Medium-Speed	Watch	Sleep	Sub-Active	Sub-Sleep	Module Stop	Software Standby	Hardware Standby	Module
P7PIN	—	—	—	—	—	—	—	—	—	PORT
PBDDR	Initialized	—	—	—	—	—	—	—	Initialized	
P8DR	Initialized	—	—	—	—	—	—	—	Initialized	
P9DDR	Initialized	—	—	—	—	—	—	—	Initialized	
P9DR	Initialized	—	—	—	—	—	—	—	Initialized	
IER	Initialized	—	—	—	—	—	—	—	Initialized	INT
STCR	Initialized	—	—	—	—	—	—	—	Initialized	SYSTEM
SYSCR	Initialized	—	—	—	—	—	—	—	Initialized	
MDCR	Initialized	—	—	—	—	—	—	—	Initialized	
BCR	Initialized	—	—	—	—	—	—	—	Initialized	BSC
WSCR	Initialized	—	—	—	—	—	—	—	Initialized	
TCR_0	Initialized	—	—	—	—	—	—	—	Initialized	TMR_0•A
TCR_1	Initialized	—	—	—	—	—	—	—	Initialized	TMR_1
TCSR_0	Initialized	—	—	—	—	—	—	—	Initialized	
TCSR_1	Initialized	—	—	—	—	—	—	—	Initialized	
TCORA_0	Initialized	—	—	—	—	—	—	—	Initialized	
TCORA_1	Initialized	—	—	—	—	—	—	—	Initialized	
TCORB_0	Initialized	—	—	—	—	—	—	—	Initialized	
TCORB_1	Initialized	—	—	—	—	—	—	—	Initialized	TMR_1
TCNT_0	Initialized	—	—	—	—	—	—	—	Initialized	
TCNT_1	Initialized	—	—	—	—	—	—	—	Initialized	
PWOERB	Initialized	—	—	—	—	—	—	—	Initialized	PWM
PWDPRB	Initialized	—	—	—	—	—	—	—	Initialized	
PWSL	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialized	
PWDR 15 to 8	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialized	
ICCR_0	Initialized	—	—	—	—	—	—	—	Initialized	IIC_0
ICSR_0	Initialized	—	—	—	—	—	—	—	Initialized	
ICDR_0	—	—	—	—	—	—	—	—	—	
SARX_0	Initialized	—	—	—	—	—	—	—	Initialized	

Register Abbreviation	Reset	High-Speed/ Medium-Speed	Watch	Sleep	Sub-Active	Sub-Sleep	Module Stop	Software Standby	Hardware Standby	Module
ICMR_0	Initialized	—	—	—	—	—	—	—	Initialized	IIC_0
SAR_0	Initialized	—	—	—	—	—	—	—	Initialized	
ADDRAH	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialized	A/D Converter
ADDRAL	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialized	
ADDRBH	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialized	
ADDRBL	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialized	
ADDRCH	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialized	
ADDRCL	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialized	
ADDRDH	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialized	
ADDRDL	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialized	
ADCSR	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialized	
ADCR	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialized	
TCSR_1	Initialized	—	—	—	—	—	—	—	Initialized	WDT_1
TCNT_1	Initialized	—	—	—	—	—	—	—	Initialized	
TCR_X	Initialized	—	—	—	—	—	—	—	Initialized	TMR_X
TCSR_X	Initialized	—	—	—	—	—	—	—	Initialized	
TICRR	Initialized	—	—	—	—	—	—	—	Initialized	
TICRF	Initialized	—	—	—	—	—	—	—	Initialized	
TCNT_X	Initialized	—	—	—	—	—	—	—	Initialized	
TCORC	Initialized	—	—	—	—	—	—	—	Initialized	
TCORA_X	Initialized	—	—	—	—	—	—	—	Initialized	
TCORB_X	Initialized	—	—	—	—	—	—	—	Initialized	
TCONRI	Initialized	—	—	—	—	—	—	—	Initialized	
TCONRS	Initialized	—	—	—	—	—	—	—	Initialized	TMR_X, TMR_Y

25.4 Register Selection Condition

Lower Address	Register Abbreviation	Register selection condition	Module
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		
H'FDD4	LADR4H	MSTP0 = 0	LPC
H'FDD5	LADR4L		
H'FDD6	IDR4		
H'FDD7	ODR4		
H'FDD8	STR4		
H'FDD9	HICR4		
H'FDDA	SIRQCR2		
H'FDE0	RBUFAR		
H'FDE1	EBLKR		
H'FDE2	LMCST1		
H'FDE3	LMCST2		
H'FDE4	LMCCR1		
H'FDE5	LMCCR2		
H'FDE6	MPCR		
H'FDE8	HBAR1H		
H'FDE9	HBAR1L		
H'FDEA	HBAR2H		
H'FDEB	HBAR2L		
H'FDEC	RAMBARH		
H'FDED	RAMBARL		
H'FDEE	ASSR		
H'FDEF	RAMASSR		

Lower Address	Register Abbreviation	Register selection condition	Module
H'FDF0	SAR1	MSTP0 = 0	LPC
H'FDF1	SAR2		
H'FDF2	FWPRH		
H'FDF3	FWPRM		
H'FDF4	FWPRL		
H'FDF5	RAMAR		
H'FDF6	FRPRH		
H'FDF7	FRPRM		
H'FDF8	FRPRL		
H'FDF9	UCMDTR		
H'FDFA	FLWARH		
H'FDFB	FLWARL		
H'FDFC	LMCMIDCR		
H'FDFD	LMCDIDCR		
H'FE00	P6NCE	No condition	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'FE10	PTCNT0		
H'FE11	PTCNT1		
H'FE12	PTCNT2		
H'FE14	P9PCR		
H'FE16	PGNOCR		
H'FE19	PFNOCR		
H'FE1C	PCNOCR		
H'FE1D	PDNOCR		

Lower Address	Register Abbreviation	Register selection condition	Module
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		
H'FE32	STR3		
H'FE34	LADR3H		
H'FE35	LADR3L		
H'FE36	SIRQCR0		
H'FE37	SIRQCR1		
H'FE38	IDR1		
H'FE39	ODR1		
H'FE3A	STR1		
H'FE3C	IDR2		
H'FE3D	ODR2		
H'FE3E	STR2		
H'FE3F	HISEL		
H'FE40	HICR0		

Lower Address	Register Abbreviation	Register selection condition	Module
H'FE41	HICR1	MSTP0 = 0	LPC
H'FE42	HICR2		
H'FE43	HICR3		
H'FE44	WUEMRB	No condition	INT
H'FE45	WUEMR		
H'FE46	PGODR	No condition	PORT
H'FE47	PGPIN (read)		
	PGDDR (write)		
H'FE48	PEPCR		
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)		
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		

Lower Address	Register Abbreviation	Register selection condition		Module
H'FE70	TCR_2	MSTP1 = 0		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'FE81	KMIMR (RELOCATE = 1)	MSTP2 = 0		INT
H'FE82	KMPCR (RELOCATE = 1)			PORT
H'FE83	KMIMRA (RELOCATE = 1)			INT
H'FE87	ICRD	No condition		
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			
H'FEB0	TSTR	MSTP1 = 0		TPU
H'FEB1	TSYR			

Lower Address	Register Abbreviation	Register selection condition		Module
H'FEC0	KBCR1_0	MSTP2 = 0		KBU
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'FECD	TISR (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)			
H'FED4	ICXR_0	MSTP4 = 0		IIC_0
H'FED5	ICXR_1	MSTP3 = 0		IIC_1
H'FED8	KBCRH_0	MSTP2 = 0		KBU
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			

Lower Address	Register Abbreviation	Register selection condition	Module
H'FEE1	KBCRL_2	MSTP2 = 0	KBU
H'FEE2	KBBR_2		
H'FEE3	KBCR2_2		
H'FEE4	KBCOMP	No condition	IrDA
H'FEE6	DDCSWR	MSTP4 = 0, IICE in STCR = 1	IIC_0, IIC_1
H'FEE8	ICRA	No condition	INT
H'FEE9	ICRB		
H'FEEA	ICRC		
H'FEEB	ISR		
H'FEEC	ISCRH		
H'FEED	ISCRL		
H'FEEE	DTCERA	No condition	DTC
H'FEEF	DTCERB		
H'FEF0	DTCERC		
H'FEF1	DTCERD		
H'FEF2	DTCERE		
H'FEF3	DTVECR		
H'FEF4	ABRKCR	No condition	INT
H'FEF5	BARA		
H'FEF6	BARB		
H'FEF7	BARC		
H'FEF8	IER16		
H'FEF9	ISR16		
H'FEFA	ISCR16H		
H'FEFB	ISCR16L		
H'FEFC	ISSR16		
H'FEFD	ISSR		
H'FF82	PCSR	No condition	PWM, PWMX
H'FF83	SYSCR2	FLSHE in STCR = 0	SYSTEM
H'FF84	SBYCR		
H'FF85	LPWRCR		

Lower Address	Register Abbreviation	Register selection condition		Module
H'FF86	MSTPCRH	FLSHE in STCR = 0		SYSTEM
H'FF87	MSTPCRL			
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
H'FF8A	SCR_1	MSTP6 = 0		SCI_1
H'FF8B	TDR1			
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 ICCR1 = 1	IIC_1
	SARX_1 (RELOCATE = 0)		ICE in ICCR1 = 0	
H'FF8F	ICMR_1 (RELOCATE = 0)		ICE in ICCR1 = 1	
	SAR_1 (RELOCATE = 0)		ICE in ICCR1 = 0	
H'FF90	TIER	MSTP13 = 0		FRT
H'FF91	TCSR			
H'FF92	FRC			
H'FF94	OCRA	MSTP13 = 0	OCRS in TOCR = 0	
	OCRB		OCRS in TOCR = 1	
H'FF96	TCR			
H'FF97	TOCR			
H'FF98	ICRA		ICRS in TOCR = 0	
	OCRAR		ICRS in TOCR = 1	
H'FF9A	ICRB		ICRS in TOCR = 0	
	OCRAF		ICRS in TOCR = 1	
H'FF9C	ICRC	MSTP13 = 0	ICRS in TOCR = 0	FRT
	OCRDM		ICRS in TOCR = 1	
H'FF9E	ICRD			

Lower Address	Register Abbreviation	Register selection condition		Module
H'FFA0	SMR_2 (RELOCATE = 1)	MSTP5 = 0		SCI_2
	SMR_2 (RELOCATE = 0)	MSTP5 = 0, IICE in STCR = 0		
	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	
H'FFA1	BRR_2 (RELOCATE = 1)	MSTP5 = 0		SCI_2
	BRR_2 (RELOCATE = 0)	MSTP5 = 0, IICE in STCR = 0		
	DADRAL (RELOCATE = 0)	MSTP11 = 0, MSTPA1 = 0, IICE in STCR = 1	REGS in DACNT/DADRB = 0	PWMX
H'FFA2	SCR_2	MSTP5 = 0		SCI_2
H'FFA3	TDR_2			
H'FFA4	SSR_2			
H'FFA5	RDR_2			
H'FFA6	SCMR_2 (RELOCATE = 1)			
	SCMR_2 (RELOCATE = 0)	MSTP5 = 0, IICE in STCR = 0		
	DADRBH (RELOCATE = 0)	MSTP11 = 0, MSTPA1 = 0, IICE in STCR = 1	REGS in DACNT/DADRB = 0	PWMX
	DACNTH (RELOCATE = 0)		REGS in DACNT/DADRB = 1	
H'FFA7	DADRBL (RELOCATE = 0)		REGS in DACNT/DADRB = 0	
	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	No condition		PORT
H'FFAB	PAPIN (read)			
	PADDR (write)			
H'FFAC	P1PCR			
H'FFAD	P2PCR			
H'FFAE	P3PCR			

Lower Address	Register Abbreviation	Register selection condition	Module
H'FFB0	P1DDR	No condition	PORT
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)		
H'FFBE	P7PIN (read)		
	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,
H'FFC9	TCR_1		TMR_1
H'FFCA	TCSR_0		
H'FFCB	TCSR_1		
H'FFCC	TCORA_0		
H'FFCD	TCORA_1		

Lower Address	Register Abbreviation	Register selection condition		Module
H'FFCE	TCORB	MSTP12 = 0		TMR_0,
H'FFCF	TCORB_1			TMR_1
H'FFD0	TCNT_0			
H'FFD1	TCNT_1			
H'FFD2	PWOERB	No condition		PWM
H'FFD4	PWDPRB			
H'FFD6	PWSL	MSTP11 = 0		
H'FFD7	PWDR 15 to 8	MSTPA0 = 0		
H'FFD8	ICCR_0 (RELOCATE = 0)	MSTP4 = 0, IICE in STCR = 1		IIC_0
H'FFD9	ICSR_0 (RELOCATE = 0)			
H'FFDE	ICDR_0 (RELOCATE = 0)	MSTP4 = 0, IICE in STCR = 1	ICE in ICCR0 = 1	
	SARX_0 (RELOCATE = 0)		ICE in ICCR0 = 0	
H'FFDF	ICMR_0 (RELOCATE = 0)	MSTP4 = 0, IICE in STCR = 1	ICE in ICCR0 = 1	
	SAR_0 (RELOCATE = 0)		ICE in ICCR0 = 0	
H'FFE0	ADDRAH	MSTP9 = 0		A/D Converter
H'FFE1	ADDRAL			
H'FFE2	ADDRBH			
H'FFE3	ADDRBL			
H'FFE4	ADDRCH			
H'FFE5	ADDRCL			
H'FFE6	ADDRDH			
H'FFE7	ADDRDL			
H'FFE8	ADCSR			
H'FFE9	ADCR			
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 STCR = 0	TMRX/Y in TCONRS = 0	
	TCR_Y (RELOCATE = 0)		TMRX/Y in TCONRS = 1	TMR_Y

Lower Address	Register Abbreviation	Register selection condition		Module
H'FFF1	KMIMR (RELOCATE = 0)	MSTP2 = 0, KINWUE in STCR = 1	TMRX/Y in TCONRS = 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
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	
	TISR (RELOCATE = 0)		TMRX/Y in TCONRS = 1	TMR_Y
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 = 1	

Lower Address	Register Abbreviation	Register selection condition	Module
H'FFFC	TCONRI (RELOCATE = 1)	MSTP8 = 0	TMR_Y
	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	

25.5 Register Addresses (Classification by Type of Module)

Module	Register name	Number of bits	Address	Initial value	Data width	Address states
INT	WUEMRB	8	H'FE44	H'FF	8	2
INT	WUEMR	8	H'FE45	H'FF	8	2
INT	KMIMR	8	H'FE81 (RELOCATE = 1)	H'BF	8	2
INT	KMIMR	8	H'FFF1 (RELOCATE = 0)	H'BF	8	2
INT	KMIMRA	8	H'FE83 (RELOCATE = 1)	H'FF	8	2
INT	KMIMRA	8	H'FFF3 (RELOCATE = 0)	H'FF	8	2
INT	ICRD	8	H'FE87	H'00	8	2
INT	ICRA	8	H'FEE8	H'00	8	2
INT	ICRB	8	H'FEE9	H'00	8	2
INT	ICRC	8	H'FEEA	H'00	8	2
INT	ISR	8	H'FEEB	H'00	8	2
INT	ISCRH	8	H'FEEC	H'00	8	2
INT	ISCRL	8	H'FEED	H'00	8	2
INT	ABRKCR	8	H'FEF4	—	8	2
INT	BARA	8	H'FEF5	H'00	8	2
INT	BARB	8	H'FEF6	H'00	8	2
INT	BARC	8	H'FEF7	H'00	8	2
INT	IER16	8	H'FEF8	H'00	8	2
INT	ISR16	8	H'FEF9	H'00	8	2
INT	ISCR16H	8	H'FEFA	H'00	8	2
INT	ISCR16L	8	H'FEFB	H'00	8	2
INT	ISSR16	8	H'FEFC	H'00	8	2
INT	ISSR	8	H'FEFD	H'00	8	2
INT	IER	8	H'FFC2	H'00	8	2
BSC	BCR	8	H'FFC6	H'D3	8	2
BSC	WSCR	8	H'FFC7	H'F3	8	2
DTC	DTCERA	8	H'FEEE	H'00	8	2

Module	Register name	Number of bits	Address	Initial value	Data width	Address states
DTC	DTCERB	8	H'FEEF	H'00	8	2
DTC	DTCERC	8	H'FEF0	H'00	8	2
DTC	DTCERD	8	H'FEF1	H'00	8	2
DTC	DTCERE	8	H'FEF2	H'00	8	2
DTC	DTVECR	8	H'FEF3	H'00	8	2
PORT	P1PCR	8	H'FFAC	H'00	8	2
PORT	P1DDR	8	H'FFB0	H'00	8	2
PORT	P1DR	8	H'FFB2	H'00	8	2
PORT	P2PCR	8	H'FFAD	H'00	8	2
PORT	P2DDR	8	H'FFB1	H'00	8	2
PORT	P2DR	8	H'FFB3	H'00	8	2
PORT	P3PCR	8	H'FFAE	H'00	8	2
PORT	P3DDR	8	H'FFB4	H'00	8	2
PORT	P3DR	8	H'FFB6	H'00	8	2
PORT	P4DDR	8	H'FFB5	H'00	8	2
PORT	P4DR	8	H'FFB7	H'00	8	2
PORT	P5DR	8	H'FFBA	H'F8	8	2
PORT	P5DDR	8	H'FFB8	H'00	8	2
PORT	P6NCE	8	H'FE00	H'00	8	2
PORT	P6NCMC	8	H'FE01	H'00	8	2
PORT	P6NCCS	8	H'FE02	H'00	8	2
PORT	KMPCR	8	H'FE82 (RELOCATE = 1)	H'00	8	2
PORT	KMPCR	8	H'FFF2 (RELOCATE = 0)	H'00	8	2
PORT	SYSCR2	8	H'FF83	H'00	8	2
PORT	P6DR	8	H'FFBB	H'00	8	2
PORT	P6DDR	8	H'FFB9	H'00	8	2
PORT	P7PIN	8	H'FFBE	—	8	2
PORT	P8DDR	8	H'FFBD	H'80	8	2
PORT	P8DR	8	H'FFBF	H'80	8	2
PORT	P9PCR	8	H'FE14	H'00	8	2

Module	Register name	Number of bits	Address	Initial value	Data width	Address states
PORT	P9DDR	8	H'FFC0	H'00	8	2
PORT	P9DR	8	H'FFC1	H'00/H'40	8	2
PORT	PAODR	8	H'FFAA	H'00	8	2
PORT	PAPIN	8	H'FFAB	H'00	8	2
PORT	PADDR	8	H'FFAB	H'00	8	2
PORT	PBODR	8	H'FFBC	H'00	8	2
PORT	PBPIN	8	H'FFBD	—	8	2
PORT	PBDDR	8	H'FFBE	H'00	8	2
PORT	PCNCE	8	H'FE03	H'00	8	2
PORT	PCNCMC	8	H'FE04	H'00	8	2
PORT	PCNCCS	8	H'FE05	H'00	8	2
PORT	PCNOCR	8	H'FE1C	H'00	8	2
PORT	PCODR	8	H'FE4C	H'00	8	2
PORT	PCPIN	8	H'FE4E (Read)	—	8	2
PORT	PCDDR	8	H'FE4E (Write)	H'00	8	2
PORT	PDNOCR	8	H'FE1D	H'00	8	2
PORT	PDODR	8	H'FE4D	H'00	8	2
PORT	PDPIN	8	H'FE4F (Read)	—	8	2
PORT	PDDDR	8	H'FE4F (Write)	H'00	8	2
PORT	PEPCR	8	H'FE48	H'00	8	2
PORT	PEPIN	8	H'FE4A (Read) (Writing prohibited)	—	8	2
PORT	PFNOCR	8	H'FE19	H'00	8	2
PORT	PFDDR	8	H'FE4B (Write)	H'00	8	2
PORT	PFODR	8	H'FE49	H'00	8	2
PORT	PFPIN	8	H'FE4B (Read)	—	8	2
PORT	PGNCE	8	H'FE06	H'00	8	2
PORT	PGNCMC	8	H'FE07	H'00	8	2
PORT	PGNCCS	8	H'FE08	H'00	8	2
PORT	PGNOCR	8	H'FE16	H'00	8	2
PORT	PGODR	8	H'FE46	H'00	8	2

Module	Register name	Number of bits	Address	Initial value	Data width	Address states
PORT	PGPIN	8	H'FE47 (Read)	—	8	2
PORT	PGDDR	8	H'FE47 (Write)	H'00	8	2
PORT	PTCNT0	8	H'FE10	H'00	8	2
PORT	PTCNT1	8	H'FE11	H'00	8	2
PORT	PTCNT2	8	H'FE12	H'00	8	2
PWM	PWOERB	8	H'FFD2	H'00	8	2
PWM	PWDPRB	8	H'FFD4	H'00	8	2
PWM	PWSL	8	H'FFD6	H'20	8	2
PWM	PWDR 15 to 8	8	H'FFD7	H'00	8	2
PWM	PCSR	8	H'FF82	H'00	8	2
PWMX	DACR	8	H'FEA0 (RELOCATE = 1)	H'30	8	2
PWMX	DACR	8	H'FFA0 (RELOCATE = 0)	H'FF	8	2
PWMX	DADRAH	8	H'FEA0 (RELOCATE = 1)	H'00	8	2
PWMX	DADRAH	8	H'FFA0 (RELOCATE = 0)	H'FF	8	2
PWMX	DADRAL	8	H'FEA1 (RELOCATE = 1)	H'FF	8	2
PWMX	DADRAL	8	H'FFA1 (RELOCATE = 0)	H'FF	8	2
PWMX	DACNTH	8	H'FEA6 (RELOCATE = 1)	H'FF	8	2
PWMX	DACNTH	8	H'FFA6 (RELOCATE = 0)	H'00	8	2
PWMX	DADRBH	8	H'FEA6 (RELOCATE = 1)	H'FF	8	2
PWMX	DADRBH	8	H'FFA6 (RELOCATE = 0)	H'FF	8	2
PWMX	DACNTL	8	H'FEA7 (RELOCATE = 1)	H'03	8	2
PWMX	DACNTL	8	H'FFA7 (RELOCATE = 0)	H'03	8	2

Module	Register name	Number of bits	Address	Initial value	Data width	Address states
PWMX	DADRBL	8	H'FEA7 (RELOCATE = 1)	H'FF	8	2
PWMX	DADRBL	8	H'FFA7 (RELOCATE = 0)	H'FF	8	2
PWMX	PCSR	8	H'FF82	H'00	8	2
FRT	TIER	8	H'FF90	H'01	8	2
FRT	TCSR	8	H'FF91	H'00	8	2
FRT	FRC	16	H'FF92	H'0000	16	2
FRT	OCRA	16	H'FF94	H'FFFF	16	2
FRT	OCRB	16	H'FF94	H'FFFF	16	2
FRT	TCR	8	H'FF96	H'00	8	2
FRT	TOCR	8	H'FF97	H'00	8	2
FRT	ICRA	16	H'FF98	H'0000	16	2
FRT	OCRAR	16	H'FF98	H'FFFF	16	2
FRT	ICRB	16	H'FF9A	H'0000	16	2
FRT	OCRAF	16	H'FF9A	H'FFFF	16	2
FRT	ICRC	16	H'FF9C	H'0000	16	2
FRT	OCRDM	16	H'FF9C	H'0000	16	2
FRT	ICRD	16	H'FF9E	H'0000	16	2
TPU_0	TCR_0	8	H'FE50	H'00	8	2
TPU_0	TMDR0	8	H'FE51	H'C0	8	2
TPU_0	TIORH_0	8	H'FE52	H'00	8	2
TPU_0	TIORL_0	8	H'FE53	H'00	8	2
TPU_0	TIER_0	8	H'FE54	H'40	8	2
TPU_0	TSR_0	8	H'FE55	H'C0	8	2
TPU_0	TCNT_0	16	H'FE56	H'0000	16	2
TPU_0	TGRA_0	16	H'FE58	H'FFFF	16	2
TPU_0	TGRB_0	16	H'FE5A	H'FFFF	16	2
TPU_0	TGRC_0	16	H'FE5C	H'FFFF	16	2
TPU_0	TGRD_0	16	H'FE5E	H'FFFF	16	2
TPU_1	TCR_1	8	H'FD40	H'00	8	2
TPU_1	TMDR_1	8	H'FD41	H'C0	8	2

Module	Register name	Number of bits	Address	Initial value	Data width	Address states
TPU_1	TIOR_1	8	H'FD42	H'00	8	2
TPU_1	TIER_1	8	H'FD44	H'40	8	2
TPU_1	TSR_1	8	H'FD45	H'C0	8	2
TPU_1	TCNT_1	16	H'FD46	H'0000	16	2
TPU_1	TGRA_1	16	H'FD48	H'FFFF	16	2
TPU_1	TGRB_1	16	H'FD4A	H'FFFF	16	2
TPU_2	TCR_2	8	H'FE70	H'00	8	2
TPU_2	TMDR_2	8	H'FE71	H'C0	8	2
TPU_2	TIOR_2	8	H'FE72	H'00	8	2
TPU_2	TIER_2	8	H'FE74	H'40	8	2
TPU_2	TSR_2	8	H'FE75	H'C0	8	2
TPU_2	TCNT_2	16	H'FE76	H'0000	16	2
TPU_2	TGRA_2	16	H'FE78	H'FFFF	16	2
TPU_2	TGRB_2	16	H'FE7A	H'FFFF	16	2
TPU	TSTR	8	H'FEB0	H'00	8	2
TPU	TSYR	8	H'FEB1	H'00	8	2
TMR_0	TCR_0	8	H'FFC8	H'00	8	2
TMR_0	TCSR_0	8	H'FFCA	H'00	8	2
TMR_0	TCORA_0	8	H'FFCC	H'FF	16	2
TMR_0	TCORB_0	8	H'FFCE	H'FF	16	2
TMR_0	TCNT_0	8	H'FFD0	H'00	16	2
TMR_1	TCR_1	8	H'FFC9	H'00	8	2
TMR_1	TCSR_1	8	H'FFCB	H'FF	16	2
TMR_1	TCORA_1	8	H'FFCD	H'FF	16	2
TMR_1	TCORB_1	8	H'FFCF	H'FF	16	2
TMR_1	TCNT_1	8	H'FFD1	H'00	16	2
TMR_X	TCR_X	8	H'FFF0	H'00	8	2
TMR_X	TCSR_X	8	H'FFF1	H'00	8	2
TMR_X	TICRR	8	H'FFF2	H'00	8	2
TMR_X	TICRF	8	H'FFF3	H'00	8	2
TMR_X	TCNT_X	8	H'FFF4	H'00	8	2

Module	Register name	Number of bits	Address	Initial value	Data width	Address states
TMR_X	TCORC	8	H'FFF5	H'FF	8	2
TMR_X	TCORA_X	8	H'FFF6	H'FF	8	2
TMR_X	TCORB_X	8	H'FFF7	H'FF	8	2
TMR_X	TCONRI	8	H'FFFC	H'00	8	2
TMR_Y	TCR_Y	8	H'FEC8 (RELOCATE = 1)	H'00	8	2
TMR_Y	TCR_Y	8	H'FFF0 (RELOCATE = 0)	H'00	8	2
TMR_Y	TCSR_Y	8	H'FEC9 (RELOCATE = 1)	H'00	8	2
TMR_Y	TCSR_Y	8	H'FFF1 (RELOCATE = 0)	H'00	8	2
TMR_Y	TCORA_Y	8	H'FECA (RELOCATE = 1)	H'FF	8	2
TMR_Y	TCORA_Y	8	H'FFF2 (RELOCATE = 0)	H'FF	8	2
TMR_Y	TCORB_Y	8	H'FECB (RELOCATE = 1)	H'FF	8	2
TMR_Y	TCORB_Y	8	H'FFF3 (RELOCATE = 0)	H'FF	8	2
TMR_Y	TCNT_Y	8	H'FECC (RELOCATE = 1)	H'00	8	2
TMR_Y	TCNT_Y	8	H'FFF4 (RELOCATE = 0)	H'00	8	2
TMR_Y	TISR	8	H'FECD (RELOCATE = 1)	H'FE	8	2
TMR_Y	TISRF	8	H'FFF5 (RELOCATE = 0)	H'FE	8	2
TMR_X, TMR_Y	TCONRS	8	H'FFFE	H'00	8	2
TMR_XY	TCRXY	8	H'FEC6	H'00	8	2
WDT_0	TCSR_0	8	H'FFA8 (Write)	H'00	16	2
WDT_0	TCSR_0	8	H'FFA8 (Read)	H'00	8	2
WDT_0	TCNT_0	8	H'FFA8 (Write)	H'00	16	2
WDT_0	TCNT_0	8	H'FFA9 (Read)	H'00	8	2

Module	Register name	Number of bits	Address	Initial value	Data width	Address states
WDT_1	TCSR_1	8	H'FFEA (Write)	H'00	16	2
WDT_1	TCSR_1	8	H'FFEA (Read)	H'00	8	2
WDT_1	TCNT_1	8	H'FFEA (Write)	H'00	16	2
WDT_1	TCNT_1	8	H'FFEB (Read)	H'00	8	2
IrDA	KBCOMP	8	H'FEE4	H'00	8	2
SCI_1	SMR_1	8	H'FF88	H'00	8	2
SCI_1	BRR_1	8	H'FF89	H'FF	8	2
SCI_1	SCR_1	8	H'FF8A	H'00	8	2
SCI_1	TDR_1	8	H'FF8B	H'FF	8	2
SCI_1	SSR_1	8	H'FF8C	H'84	8	2
SCI_1	RDR_1	8	H'FF8D	H'00	8	2
SCI_1	SCMR_1	8	H'FF8E	H'F2	8	2
SCI_2	SMR_2	8	H'FFA0	H'00	8	2
SCI_2	BRR_2	8	H'FFA1	H'FF	8	2
SCI_2	SCR_2	8	H'FFA2	H'00	8	2
SCI_2	TDR_2	8	H'FFA3	H'FF	8	2
SCI_2	SSR_2	8	H'FFA4	H'84	8	2
SCI_2	RDR_2	8	H'FFA5	H'00	8	2
SCI_2	SCMR_2	8	H'FFA6	H'F2	8	2
IIC_0	ICXR_0	8	H'FED4	H'00	8	2
IIC_0	ICCR_0	8	H'FFD8	H'01	8	2
IIC_0	ICSR_0	8	H'FFD9	H'00	8	2
IIC_0	ICDR_0	8	H'FFDE	—	8	2
IIC_0	SARX_0	8	H'FFDE	H'01	8	2
IIC_0	ICMR_0	8	H'FFDF	H'00	8	2
IIC_0	SAR_0	8	H'FFDF	H'00	8	2
IIC_1	ICDR_1	8	H'FECE (RELOCATE = 1)	—	8	2
IIC_1	SARX_1	8	H'FECE (RELOCATE = 1)	H'01	8	2
IIC_1	ICMR_1	8	H'FECF (RELOCATE = 1)	H'00	8	2

Module	Register name	Number of bits	Address	Initial value	Data width	Address states
IIC_1	SAR_1	8	H'FECF (RELOCATE = 1)	H'00	8	2
IIC_1	ICCR_1	8	H'FED0 (RELOCATE = 1)	H'01	8	2
IIC_1	ICSR_1	8	H'FED1 (RELOCATE = 1)	H'00	8	2
IIC_1	ICXR_1	8	H'FED5	H'00	8	2
IIC_1	ICCR_1	8	H'FF88 (RELOCATE = 0)	H'01	8	2
IIC_1	ICSR_1	8	H'FF89 (RELOCATE = 0)	H'00	8	2
IIC_1	ICDR_1	8	H'FF8E (RELOCATE = 0)	—	8	2
IIC_1	SARX_1	8	H'FF8E (RELOCATE = 0)	H'01	8	2
IIC_1	ICMR_1	8	H'FF8F (RELOCATE = 0)	H'00	8	2
IIC_1	SAR_1	8	H'FF8F (RELOCATE = 0)	H'00	8	2
IIC_0, IIC_1	DDCSWR	8	H'FEE6	H'0F	8	2
KBU_0	KBCR1_0	8	H'FEC0	H'00	8	2
KBU_0	KBTR_0	8	H'FEC1	H'FF	8	2
KBU_0	KBCRH_0	8	H'FED8	H'70	8	2
KBU_0	KBCRL_0	8	H'FED9	H'70	8	2
KBU_0	KBBR_0	8	H'FEDA	H'00	8	2
KBU_0	KBCR2_0	8	H'FEDB	H'F0	8	2
KBU_1	KBCR1_1	8	H'FEC2	H'00	8	2
KBU_1	KBTR_1	8	H'FEC3	H'FF	8	2
KBU_1	KBCRH_1	8	H'FEDC	H'70	8	2
KBU_1	KBCRL_1	8	H'FEDD	H'70	8	2
KBU_1	KBBR_1	8	H'FEDE	H'00	8	2
KBU_1	KBCR2_1	8	H'FEDF	H'F0	8	2
KBU_2	KBCR1_2	8	H'FEC4	H'00	8	2
KBU_2	KBTR_2	8	H'FEC5	H'FF	8	2

Module	Register name	Number of bits	Address	Initial value	Data width	Address states
KBU_2	KBCRH_2	8	H'FEE0	H'70	8	2
KBU_2	KBCRL_2	8	H'FEE1	H'70	8	2
KBU_2	KBBR_2	8	H'FEE2	H'00	8	2
KBU_2	KBCR2_2	8	H'FEE3	H'F0	8	2
LPC	RBUFAR	8	H'FDE0	H'EF	8	2
LPC	EBLKR	8	H'FDE1	H'00	8	2
LPC	LMCST1	8	H'FDE2	H'00	8	2
LPC	LMCST2	8	H'FDE3	H'00	8	2
LPC	LMCCR1	8	H'FDE4	H'00	8	2
LPC	LMCCR2	8	H'FDE5	H'00	8	2
LPC	MPCR	8	H'FDE6	H'00	8	2
LPC	HBAR1H	8	H'FDE8	—	8	2
LPC	HBAR1L	8	H'FDE9	H'00	8	2
LPC	HBAR2H	8	H'FDEA	H'00	8	2
LPC	HBAR2L	8	H'FDEB	H'00	8	2
LPC	RAMBARH	8	H'FDEC	H'00	8	2
LPC	RAMBARL	8	H'FDED	H'00	8	2
LPC	ASSR	8	H'FDEE	H'00	8	2
LPC	RAMASSR	8	H'FDEF	H'00	8	2
LPC	SAR1	8	H'FDF0	H'00	8	2
LPC	SAR2	8	H'FDF1	H'00	8	2
LPC	FWPRH	8	H'FDF2	H'FF	8	2
LPC	FWPRM	8	H'FDF3	H'FF	8	2
LPC	FWPRL	8	H'FDF4	H'FF	8	2
LPC	RAMAR	8	H'FDF5	H'D0	8	2
LPC	FRPRH	8	H'FDF6	H'FF	8	2
LPC	FRPRM	8	H'FDF7	H'FF	8	2
LPC	FRPRL	8	H'FDF8	H'FF	8	2
LPC	UCMDTR	8	H'FDF9	H'00	8	2
LPC	FLWARH	8	H'FDFA	H'00	8	2
LPC	FLWARL	8	H'FDFB	H'00	8	2

Module	Register name	Number of bits	Address	Initial value	Data width	Address states
LPC	LMCMIDCR	8	H'FDFC	H'00	8	2
LPC	LMCDIDCR	8	H'FDFD	H'00	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	H'00	8	2
LPC	LADR3H	8	H'FE34	H'00	8	2
LPC	LADR3L	8	H'FE35	H'00	8	2
LPC	SIRQCR0	8	H'FE36	H'00	8	2
LPC	SIRQCR1	8	H'FE37	H'00	8	2
LPC	IDR1	8	H'FE38	—	8	2
LPC	ODR1	8	H'FE39	—	8	2
LPC	STR1	8	H'FE3A	H'00	8	2
LPC	IDR2	8	H'FE3C	—	8	2
LPC	ODR2	8	H'FE3D	—	8	2

Module	Register name	Number of bits	Address	Initial value	Data width	Address states
LPC	STR2	8	H'FE3E	H'00	8	2
LPC	HISEL	8	H'FE3F	H'03	8	2
LPC	HICR0	8	H'FE40	H'00	8	2
LPC	HICR1	8	H'FE41	H'00	8	2
LPC	HICR2	8	H'FE42	—	8	2
LPC	HICR3	8	H'FE43	—	8	2
LPC	LADR4H	8	H'FDD4	H'00	8	2
LPC	LADR4L	8	H'FDD5	H'00	8	2
LPC	IDR4	8	H'FDD6	—	8	2
LPC	ODR4	8	H'FDD7	—	8	2
LPC	STR4	8	H'FDD8	H'00	8	2
LPC	HICR4	8	H'FDD9	H'00	8	2
LPC	SIRQCR2	8	H'FDDA	H'00	8	2
A/D Converter	ADDRAH	8	H'FFE0	H'00	8	2
A/D Converter	ADDRAL	8	H'FFE1	H'00	8	2
A/D Converter	ADDRBH	8	H'FFE2	H'00	8	2
A/D Converter	ADDRBL	8	H'FFE3	H'00	8	2
A/D Converter	ADDRCH	8	H'FFE4	H'00	8	2
A/D Converter	ADDRCL	8	H'FFE5	H'00	8	2
A/D Converter	ADDRDH	8	H'FFE6	H'00	8	2
A/D Converter	ADDRDL	8	H'FFE7	H'00	8	2
A/D Converter	ADCSR	8	H'FFE8	H'00	8	2
A/D Converter	ADCR	8	H'FFE9	H'3F	8	2
ROM	FCCS	8	H'FEA8	—	8	2

Module	Register name	Number of bits	Address	Initial value	Data width	Address states
ROM	FPCS	8	H'FEA9	H'00	8	2
ROM	FECS	8	H'FEAA	H'00	8	2
ROM	FKEY	8	H'FEAC	H'00	8	2
ROM	FMATS	8	H'FEAD	—	8	2
ROM	FTDAR	8	H'FEAE	H'00	8	2
SYSTEM	MSTPCRA	8	H'FE7E	H'00	8	2
SYSTEM	SBYCR	8	H'FF84	H'01	8	2
SYSTEM	LPWRCR	8	H'FF85	H'00	8	2
SYSTEM	MSTPCRH	8	H'FF86	H'3F	8	2
SYSTEM	MSTPCRL	8	H'FF87	H'FF	8	2
SYSTEM	SYSCR3	8	H'FE7D	H'00	8	2
SYSTEM	STCR	8	H'FFC3	H'00	8	2
SYSTEM	SYSCR	8	H'FFC4	H'09	8	2
SYSTEM	MDCR	8	H'FFC5	—	8	2
SYSTEM	SYSCR2	8	H'FF83	H'00	8	2

Section 26 Electrical Characteristics

26.1 Absolute Maximum Ratings

Table 26.1 lists the absolute maximum ratings.

Table 26.1 Absolute Maximum Ratings

Item	Symbol	Value	Unit
Power supply voltage*	V_{CC}	–0.3 to +4.3	V
Input voltage (except port 7, A, G, P97, P86, P52, and P42)	V_{in}	–0.3 to V_{CC} +0.3	
Input voltage (port A, G, P97, P86, P52, and P42)	V_{in}	–0.3 to +7.0	
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}	–20 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 4.3V.

Note: * Voltage applied to the VCC pin. The VCL pin should not be applied a voltage.

26.2 DC Characteristics

Table 26.2 lists the DC characteristics. Table 26.3 lists the permissible output currents. Table 26.4 lists the bus drive characteristics.

Table 26.2 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(KWUL = 00)*[2], $\overline{IRQ7}$ to $\overline{IRQ0}$*[3], $\overline{IRQ15}$ to $\overline{IRQ8}$, $\overline{KIN7}$ to $\overline{KIN0}$, $\overline{KIN15}$ to $\overline{KIN8}$, $\overline{WUE15}$ to $\overline{WUE0}$, $\overline{ExIRQ7}$ to $\overline{ExIRQ0}$, and $\overline{ExIRQ15}$ to $\overline{ExIRQ8}$	(1)	V_T^-	V_{CC} × 0.2	—	—	V	
			V_T^+	—	—	V_{CC} × 0.7		
			$V_T^+ - V_T^-$	V_{CC} × 0.05	—	—		
Schmitt trigger input voltage (changing levels)	P67 to P60(KWUL = 01)		V_T^-	V_{CC} × 0.3	—	—		
			V_T^+	—	—	V_{CC} × 0.7		
			$V_T^+ - V_T^-$	V_{CC} × 0.05	—	—		
	P67 to P60(KWUL = 10)		V_T^-	V_{CC} × 0.4	—	—		
			V_T^+	—	—	V_{CC} × 0.8		
			$V_T^+ - V_T^-$	V_{CC} × 0.03	—	—		
	P67 to P60(KWUL = 11)		V_T^-	V_{CC} × 0.45	—	—		
			V_T^+	—	—	V_{CC} × 0.9		
			$V_T^+ - V_T^-$	0.05	—	—		
Input high voltage	$\overline{RES}$, $\overline{STBY}$, NMI, MD2, MD1, MD0, FWE, and $\overline{ETRST}$	(2)	V_{IH}	V_{CC} × 0.9	—	V_{CC} + 0.3		
	EXTAL			V_{CC} × 0.7	—	V_{CC} + 0.3		
	Port 7			V_{CC} × 0.7	—	AV_{CC} + 0.3		
	Port A, G, P97, P86, P52, and P42			V_{CC} × 0.7	—	5.5		
	Input pins other than (1) and (2) above			V_{CC} × 0.7	—	V_{CC} + 0.3		

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input low voltage	$\overline{RES}$, $\overline{STBY}$, MD2, MD1, MD0, FWE, 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 port A, G, P97, P86, P52, and P42)	V_{OH}	V_{CC} – 0.5	—	—		I_{OH} = –200 µA
			V_{CC} – 1.0	—	—		I_{OH} = –1 mA
	Port A, G, P97, P86, P52, and P42*[4]		0.5	—	—		I_{OH} = –200 µA
Output low voltage	All output pins *[5]	V_{OL}	—	—	0.4		I_{OL} = 1.6 mA
	Ports 1, 2, 3, C, and D		—	—	1.0		I_{OL} = 5 mA

Table 26.2 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	V_{in} = 0.5 to V_{CC} – 0.5 V
	$\overline{STBY}$, NMI, MD2, MD1, MD0, and FWE		—	—	1.0		
	Port 7		—	—	1.0		V_{in} = 0.5 to AV_{CC} – 0.5 V
Three-state leakage current (off state)	Ports 1 to 6 Ports 8, 9, and A to G	$\lvert I_{TSI} \rvert$	—	—	1.0		V_{in} = 0.5 to V_{CC} – 0.5 V
Input pull-up MOS current	Ports 1 to 3 and P95 to P90	$-I_P$	5	—	150		V_{in} = 0 V
	Port 6 (P6PUE=0), B to F		30	—	300		
	Port 6 (P6PUE=1)		3	—	100		
Input capacitance	P41, P40, PB7 to PB3, and PC7	C_{in}	—	—	15	pF	V_{in} = 0 V f = 1MHz Ta = 25°C
	Other than above		—	—	10		
Current consumption*[6]	Normal operation	I_{CC}	—	30	45	mA	V_{CC} = 3.0 V to 3.6 V f = 20 MHz, all modules operating, high-speed mode
	Sleep mode		—	22	35		V_{CC} = 3.0 V to 3.6 V f = 20 MHz
	Standby mode*[7]		—	10	40	µA	Ta ≤ 50 °C
			—	—	80		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 connection to the power supply (V_{CC}). The relationship between these two pins should be AVref ≤ AV_{CC}.

2. Includes peripheral module inputs multiplexed on the pin.
3. IRQ2 includes the ADTRG input multiplexed on the pin.
4. Ports A, G, 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, ExSCLA, ExSCLB, ExSDAA, and ExSDAB (ICE bit in ICCR is 1).
Ports A, G, P97, P86/SCK1, P52, and P42/SCK2 (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.
5. Indicates values when ICCS = 0, ICE = 0, and KBIOE = 0. Low level output when the bus drive function is selected is rated separately.
6. 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 26.2 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 P83 to P80, PB1, PB0	V_{IH}	V_{CC} × 0.5	—	V	
Input low voltage	P37 to P30 P83 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 26.3 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, ExSCLA, ExSDAA, ExSCLB, ExSDAB, PS2AC to PS2CC, PS2AD to PS2CD, and PA7 to PA4 (bus drive function selected)	I_{OL}	—	—	10	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	$\sum I_{OL}$	—	—	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	$\sum -I_{OH}$	—	—	30	

Notes: 1. To protect LSI reliability, do not exceed the output current values in table 26.3.

2. When driving a Darlington transistor or LED, always insert a current-limiting resistor in the output line, as show in figures 26.1 and 26.2.

Table 26.4 Bus Drive Characteristics

Conditions: V_{CC} = 3.0 V to 3.6V, V_{SS} = 0 V

Applicable Pins: SCL0, SDA0, SCL1, SDA1, 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 PS2CC, PS2AD to PS2CD, and PA7 to PA4 (bus drive function selected)

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Output low voltage	V_{OL}	—	—	0.8	V	I_{OL} = 16mA
		—	—	0.5		I_{OL} = 8 mA
		—	—	0.4		I_{OL} = 3 mA

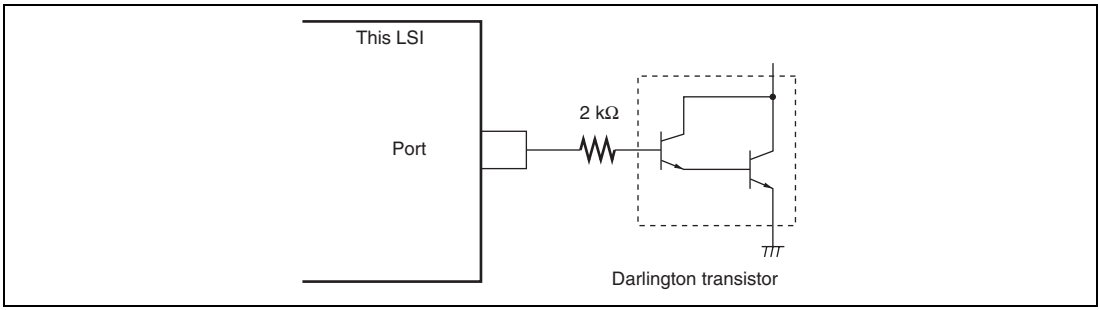


Figure 26.1 Darlington Transistor Drive Circuit (Example)

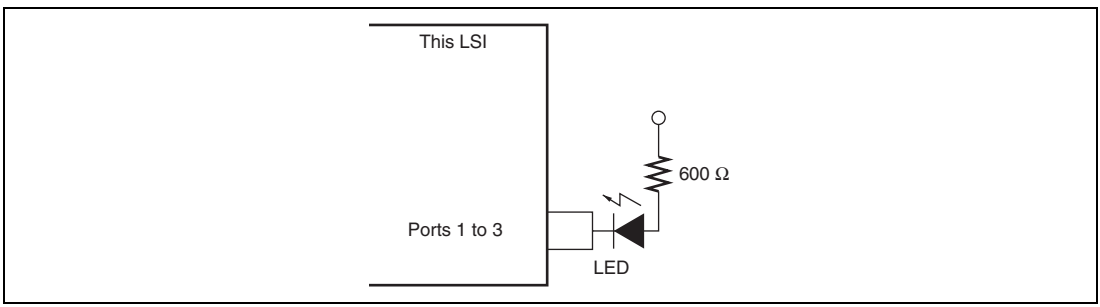


Figure 26.2 LED Drive Circuit (Example)

26.3 AC Characteristics

Figure 26.3 shows the test conditions for the AC characteristics.

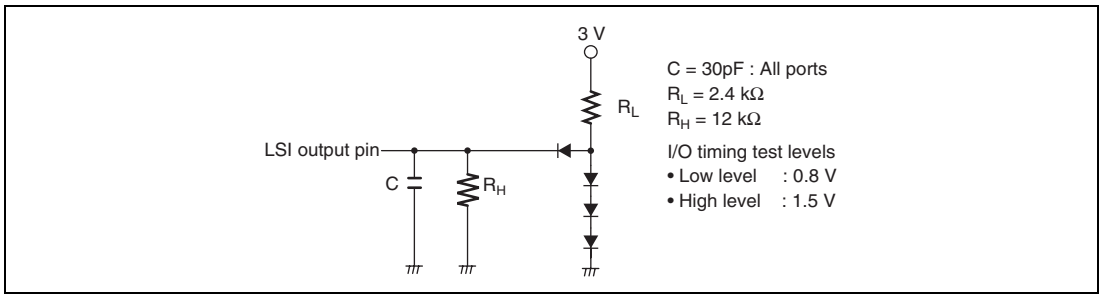


Figure 26.3 Output Load Circuit

26.3.1 Clock Timing

Table 26.5 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 23, Clock Pulse Generator.

Table 26.5 Clock Timing

Condition A: V_{CC} = 3.0 V to 3.6 V, V_{SS} = 0 V, ϕ = 4 MHz to 10 MHz

Condition B: V_{CC} = 3.0 V to 3.6 V, V_{SS} = 0 V, ϕ = 4 MHz to 20 MHz

		Condition A		Condition B			
Item	**Symbol**	**Min.**	**Max.**	**Min.**	**Max.**	**Unit**	**Reference**
Clock cycle time	t_{cyc}	100	250	50	250	ns	Figure 26.4
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	—	10	—	ms	Figure 26.5
Software standby oscillation stabilization time (crystal)	t_{OSC2}	8	—	8	—		Figure 26.6
External clock output stabilization delay time	t_{DEXT}	500	—	500	—	µs	Figure 26.5

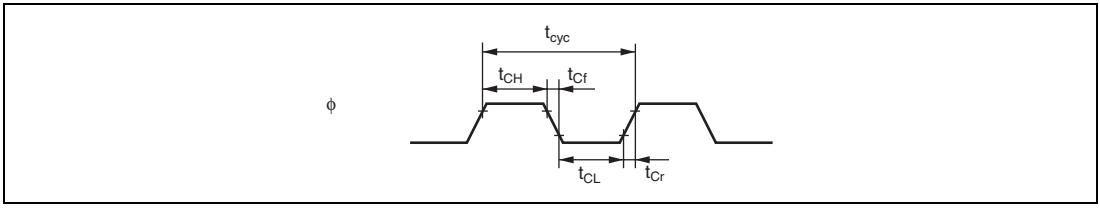


Figure 26.4 System Clock Timing

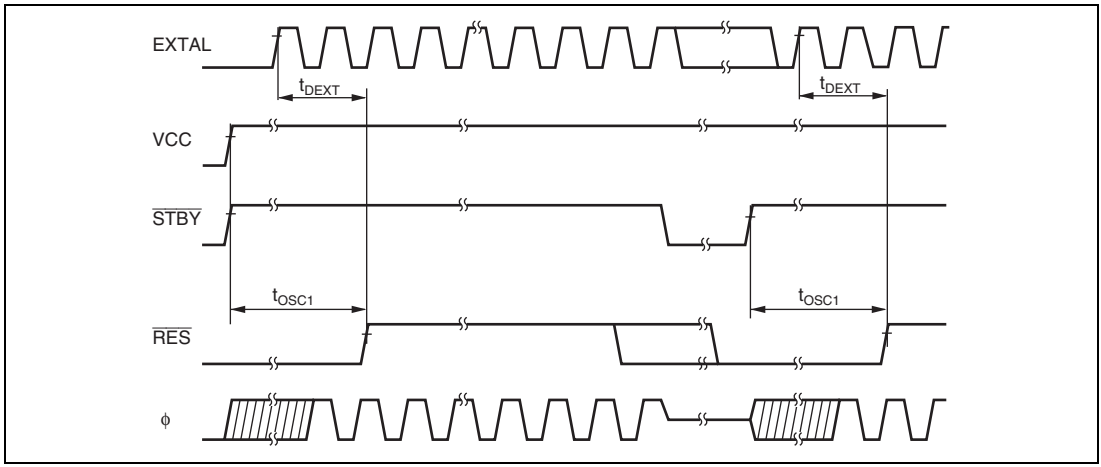


Figure 26.5 Oscillation Stabilization Timing

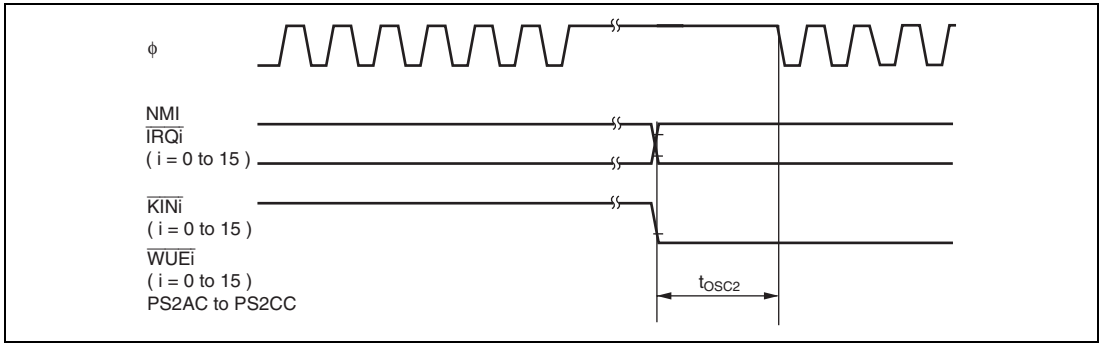


Figure 26.6 Oscillation Stabilization Timing (Exiting Software Standby Mode)

26.3.2 Control Signal Timing

Table 26.6 shows the control signal timing. Only external interrupts NMI, IRQ0 to IRQ15, KIN0 to KIN15, WUE0 to WUE15, and KBCA to KBCC can be operated based on the subclock (ϕ_{SUB} = 32.768 kHz).

Table 26.6 Control Signal Timing

Conditions: V_{CC} = 3.0 V to 3.6 V, V_{SS} = 0 V, ϕ = 32.768 kHz, 4 MHz to 20 MHz

Item	Symbol	Min.	Max.	Unit	Test Conditions
$\overline{RES}$ setup time	t_{RESS}	200	—	ns	Figure 26.7
$\overline{RES}$ pulse width	t_{RESW}	20	—	t_{cyc}	
NMI setup time	t_{NMIS}	150	—	ns	Figure 26.8
NMI hold time	t_{NMIH}	10	—		
NMI pulse width (exiting software standby mode)	t_{NMIW}	200	—		
IRQ setup time ($\overline{IRQ15}$ to $\overline{IRQ0}$, $\overline{KIN15}$ to $\overline{KIN0}$, $\overline{WUE15}$ to $\overline{WUE0}$)	t_{IRQS}	150	—		
IRQ hold time ($\overline{IRQ15}$ to $\overline{IRQ0}$, $\overline{KIN15}$ to $\overline{KIN0}$, $\overline{WUE15}$ to $\overline{WUE0}$)	t_{IRQH}	10	—		
IRQ pulse width ($\overline{IRQ15}$ to $\overline{IRQ0}$, $\overline{KIN15}$ to $\overline{KIN0}$, $\overline{WUE15}$ to $\overline{WUE0}$) (exiting software standby mode)	t_{IRQW}	200	—		

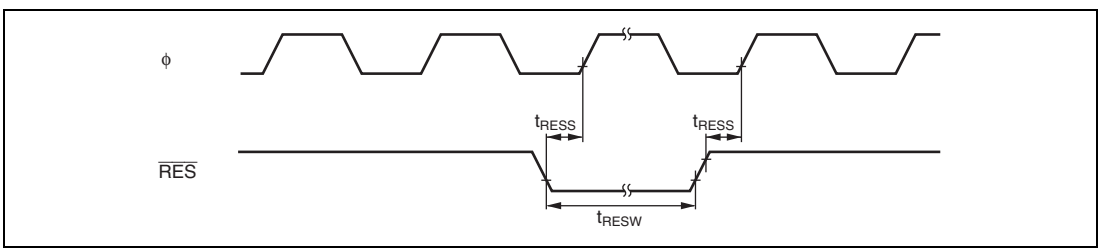


Figure 26.7 Reset Input Timing

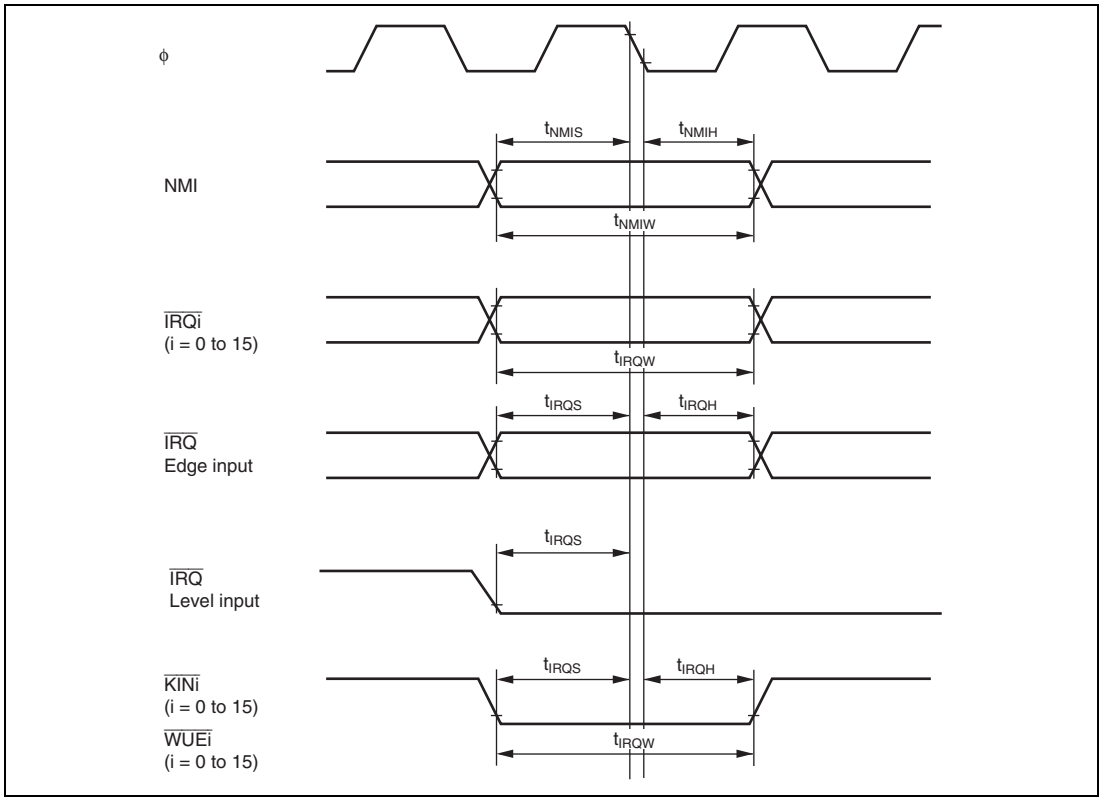


Figure 26.8 Interrupt Input Timing

26.3.3 Timing of On-Chip Peripheral Modules

Tables 26.7 to 26.9 show 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, WUE0 to WUE15, and KBCA to KBCC), watchdog timer, and 8-bit timer (channels 0 and 1) only.

Table 26.7 Timing of On-Chip Peripheral Modules

Conditions: V_{CC} = 3.0 V to 3.6 V, V_{SS} = 0 V, ϕ_{SUB} = 32.768 kHz*, ϕ = 4 MHz to 20 MHz

Item			Symbol	Min.	Max.	Unit	Test Conditions
I/O ports	Output data delay time		t_{PWD}	—	50	ns	Figure 26.9
	Input data setup time		t_{PRS}	30	—		
	Input data hold time		t_{PRH}	30	—		
FRT	Timer output delay time		t_{FTOD}	—	50	ns	Figure 26.10
	Timer input setup time		t_{FTIS}	30	—		
	Timer clock input setup time		t_{FTCS}	30	—		Figure 26.11
	Timer clock pulse width	Single edge	t_{FTCWH}	1.5	—	t_{cyc}	
		Both edges	t_{FTCWL}	2.5	—		
TPU	Timer output delay time		t_{TOCD}	—	50	ns	Figure 26.12
	Timer input setup time		t_{TICS}	30	—		
	Timer clock input setup time		t_{TCKS}	30	—		Figure 26.13
	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 26.14
	Timer reset input setup time		t_{TMRS}	30	—		Figure 26.16
	Timer clock input setup time		t_{TMCS}	30	—		Figure 26.15
	Timer clock pulse width	Single edge	t_{TMCWH}	1.5	—	t_{cyc}	
		Both edges	t_{TMCWL}	2.5	—		
PWM, PWMX	Timer output delay time		t_{PWOD}	—	50	ns	Figure 26.17
SCI	Input clock cycle	Asynchronous	t_{Scyc}	4	—	t_{cyc}	Figure 26.18
		Synchronous		6	—		
	Input clock pulse width		t_{SCKW}	0.4	0.6	t_{Scyc}	
	Input clock rise time		t_{SCKr}	—	1.5	t_{cyc}	
	Input clock fall time		t_{SCKf}	—	1.5		
	Transmit data delay time (synchronous)		t_{TXD}	—	50	ns	Figure 26.19
	Receive data setup time (synchronous)		t_{RXS}	50	—		
	Receive data hold time (synchronous)		t_{RXH}	50	—		
A/D converter	Trigger input setup time		t_{TRGS}	30	—	ns	Figure 26.20
WDT	$\overline{\text{RESO}}$ output delay time		t_{RESD}	—	100	ns	Figure 26.21
	$\overline{\text{RESO}}$ output pulse width		t_{RESOW}	132	—	t_{cyc}	

Note: * Applied only for the peripheral modules that are available during subclock operation.

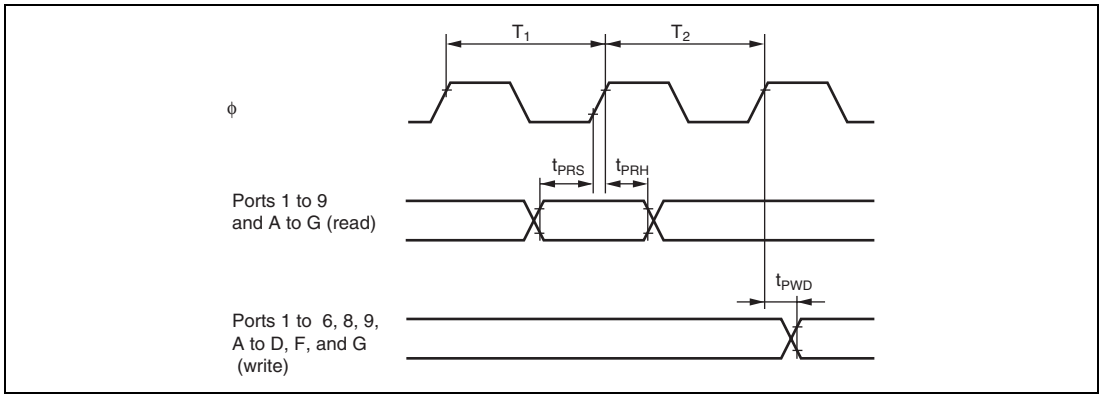


Figure 26.9 I/O Port Input/Output Timing

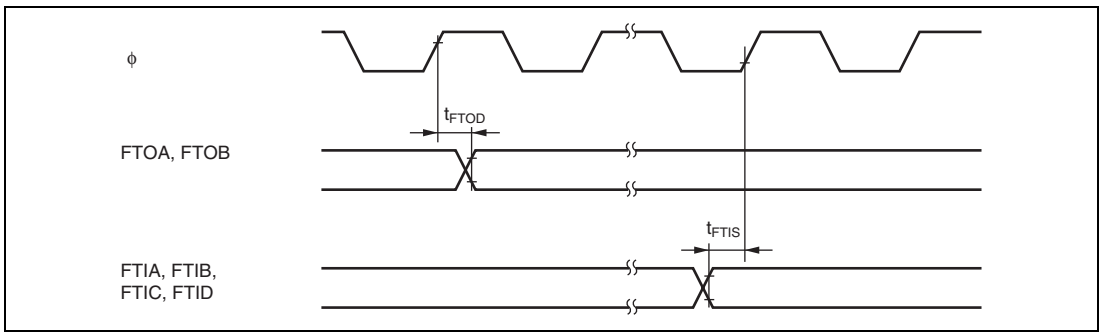


Figure 26.10 FRT Input/Output Timing

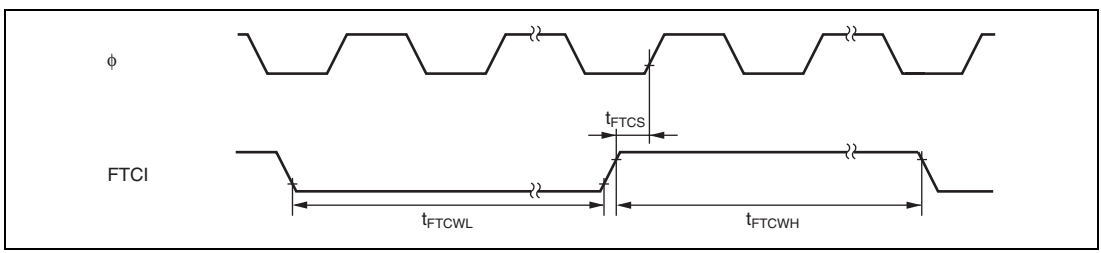


Figure 26.11 FRT Clock Input Timing

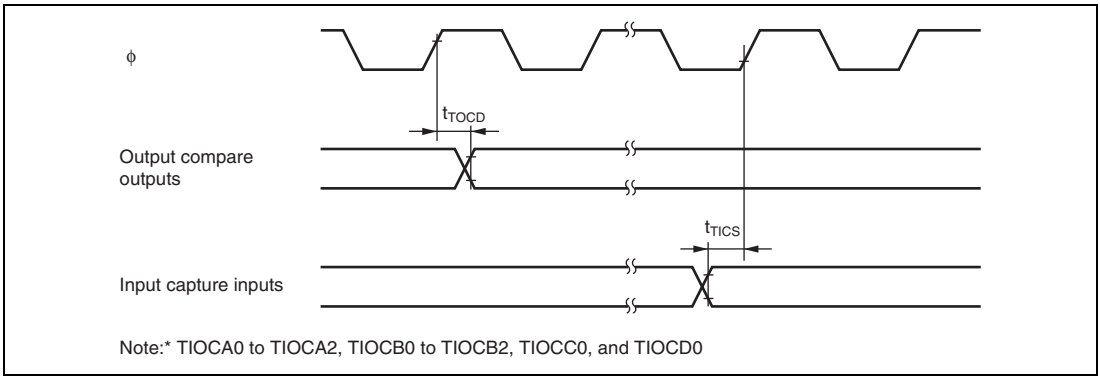


Figure 26.12 TPU Input/Output Timing

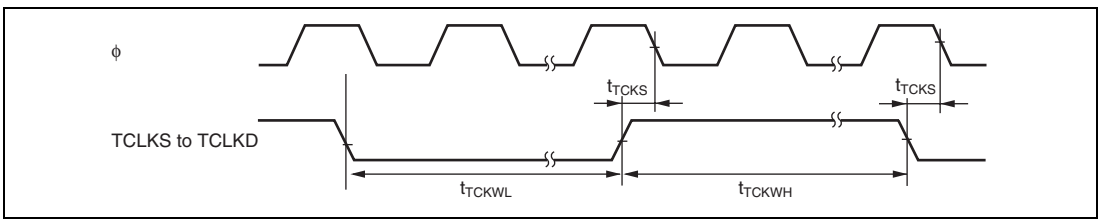


Figure 26.13 TPU Clock Input Timing

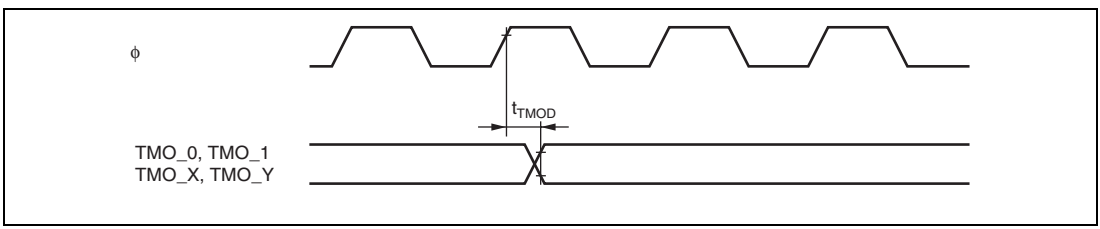


Figure 26.14 8-Bit Timer Output Timing

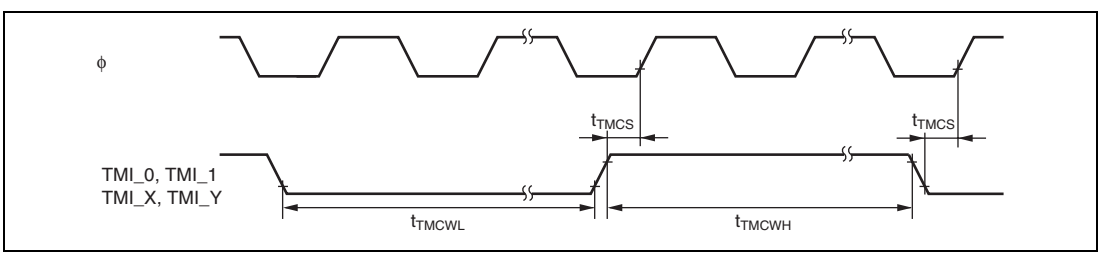


Figure 26.15 8-Bit Timer Clock Input Timing

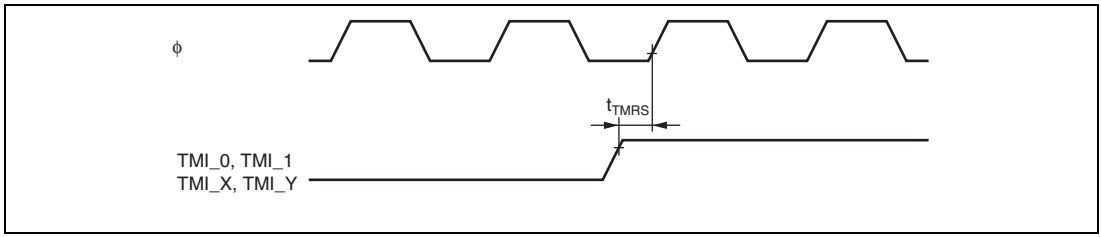


Figure 26.16 8-Bit Timer Reset Input Timing

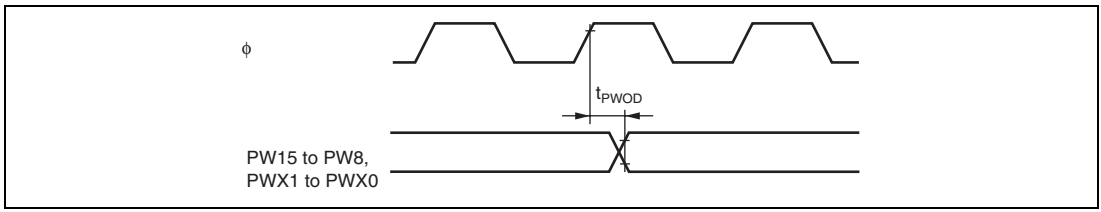


Figure 26.17 PWM, PWMX Output Timing

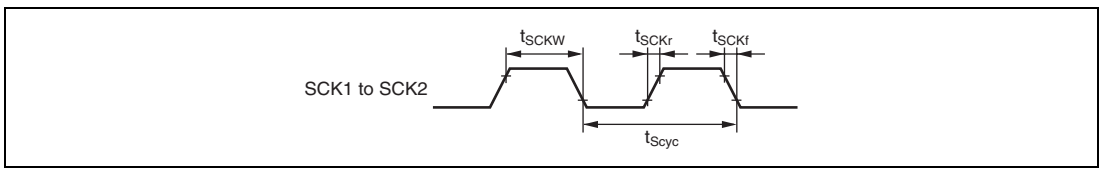


Figure 26.18 SCK Clock Input Timing

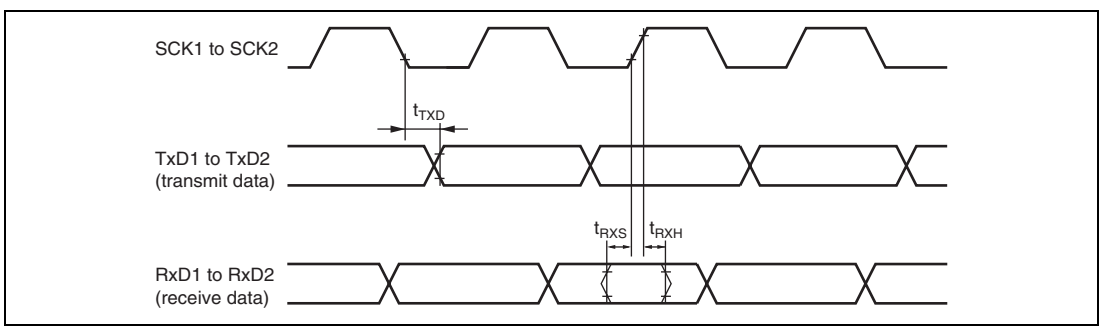


Figure 26.19 SCI Input/Output Timing (Clock Synchronous Mode)

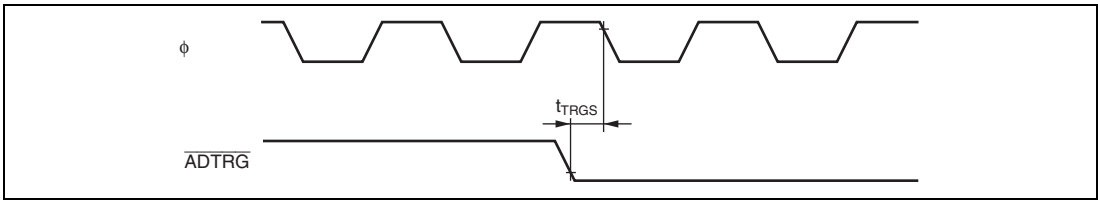


Figure 26.20 A/D Converter External Trigger Input Timing

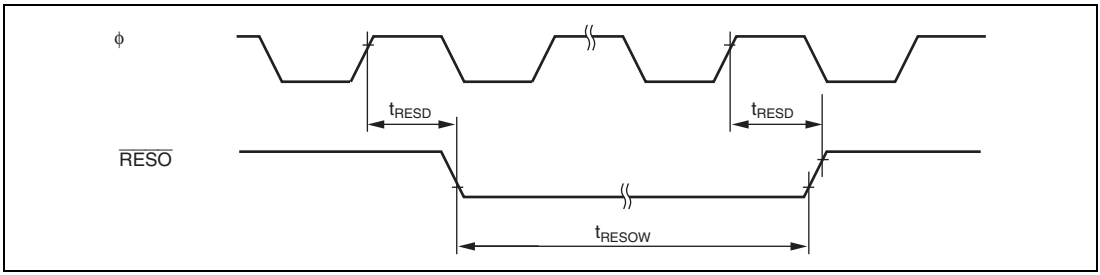


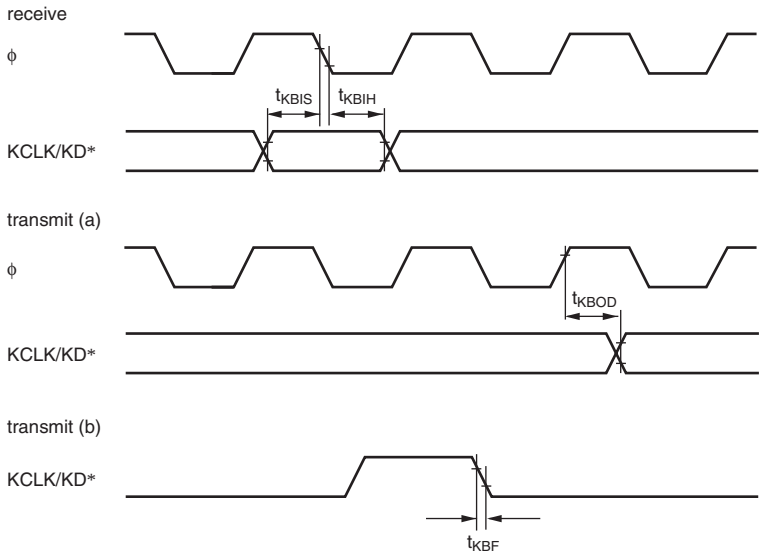
Figure 26.21 WDT Output Timing ($\overline{\text{RESO}}$)

Table 26.8 KBU Bus Timing

Conditions: V_{CC} = 3.0 V to 3.6 V, V_{SS} = 0 V, ϕ = 4 MHz to 20 MHz

Item	Symbol	Standard Value			Unit	Test Conditions	Remarks
		Min.	Typ.	Max.			
KCLK, KD output fall time	t_{KBF}	20 + 0.1 C_b	—	—	ns		Figure 26.22
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 26.22.



Note: ϕ shown in this figure is divided by 1/N when the operating mode is medium-speed mode.

KCLK : PS2AC to PS2CC
KD : PS2AD to PS2CD

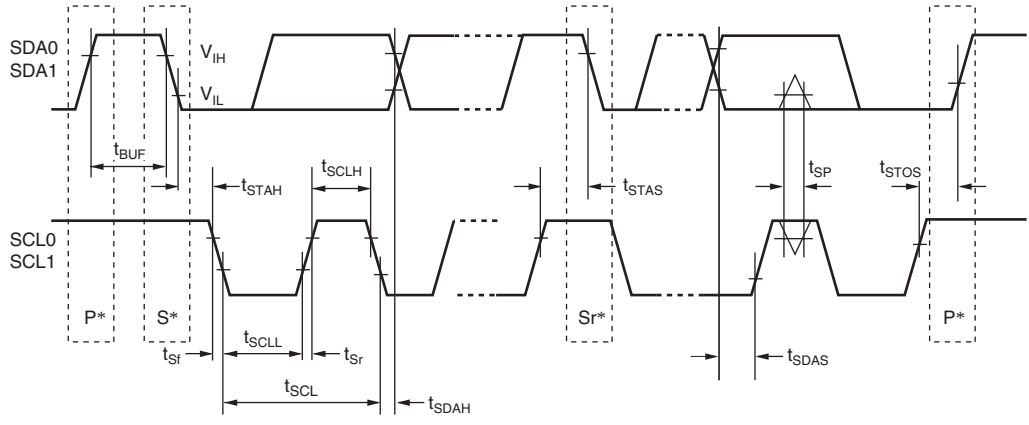
Figure 26.22 Keyboard Buffer Control Unit Timing

Table 26.9 I²C Bus Timing

Conditions: V_{CC} = 3.0 V to 3.6 V, V_{SS} = 0 V, ϕ = 4 MHz to 20 MHz

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
SCL input cycle time	t_{SCL}	12	—	—	t_{cyc}	Figure 26.23
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 output fall time	t_{Of}	20 + 0.1 C_b	—	250		
SCL, SDA input spike pulse elimination time	t_{SP}	—	—	1	t_{cyc}	
SDA input bus free time	t_{BUF}	5	—	—		
Start condition input hold time	t_{STAH}	3	—	—		
Retransmission start condition input setup time	t_{STAS}	3	—	—		
Stop condition input setup time	t_{STOS}	3	—	—		
Data input setup time	t_{SDAS}	0.5	—	—		
Data input hold time	t_{SDAH}	0	—	—	ns	
SCL, SDA capacitive load	C_b	—	—	400	pF	

Note: * 17.5 t_{cyc} can be set according to the clock selected for use by the I²C module.



Note: * S, P, and Sr indicate the following conditions:
S: Start condition
P: Stop condition
Sr: Retransmission start condition

Figure 26.23 I²C Bus Interface Input/Output Timing

Table 26.10 LPC Timing

Conditions: V_{CC} = 3.0 V to 3.6V, V_{SS} = 0 V, ϕ = 4 MHz to 20 MHz

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input clock cycle	t_{Lcyc}	30	—	—	ns	Figure 26.24
Input clock pulse width (H)	t_{LCKH}	11	—	—		
Input clock pulse width (L)	t_{LCKL}	11	—	—		
Transmit signal delay time	t_{TXD}	2	—	11		
Transmit signal floating delay time	t_{OFF}	—	—	28		
Receive signal setup time	t_{RXS}	7	—	—		
Receive signal hold time	t_{RXH}	0	—	—		

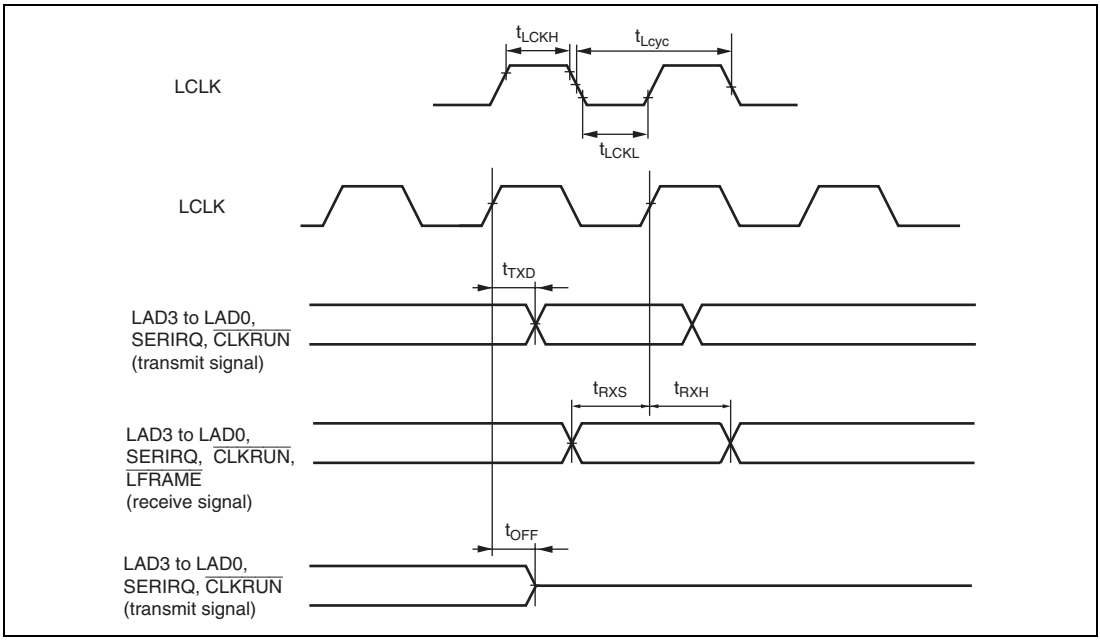


Figure 26.24 LPC Interface Timing

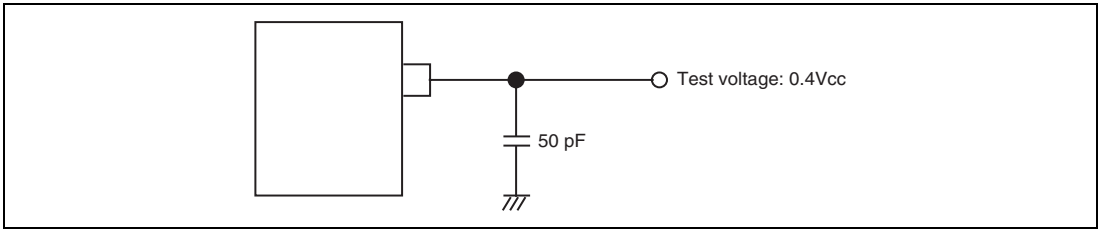


Figure 26.25 Test Conditions for Tester

Table 26.11 JTAG Timing

Conditions: V_{CC} = 3.0 V to 3.6 V, V_{SS} = 0 V, ϕ = 4 MHz to 20 MHz

Item	Symbol	Min.	Max.	Unit	Test Conditions
ETCK clock cycle time	t_{TCKcyc}	50*	250*	ns	Figure 26.26
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 26.27
Reset hold transition pulse width	t_{RSTHW}	3	—		
ETMS setup time	t_{TMSS}	20	—	ns	Figure 26.28
ETMS hold time	t_{TMSH}	20	—		
ETDI setup time	t_{TDIS}	20	—		
ETDI hold time	t_{TDIH}	20	—		
ETDO data delay time	t_{TDOD}	—	20		

Note: * When $t_{cyc} \leq t_{TCKcyc}$

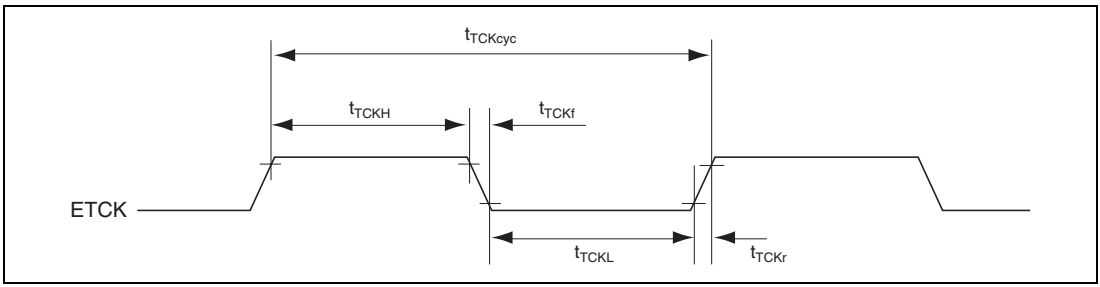


Figure 26.26 JTAG ETCK Timing

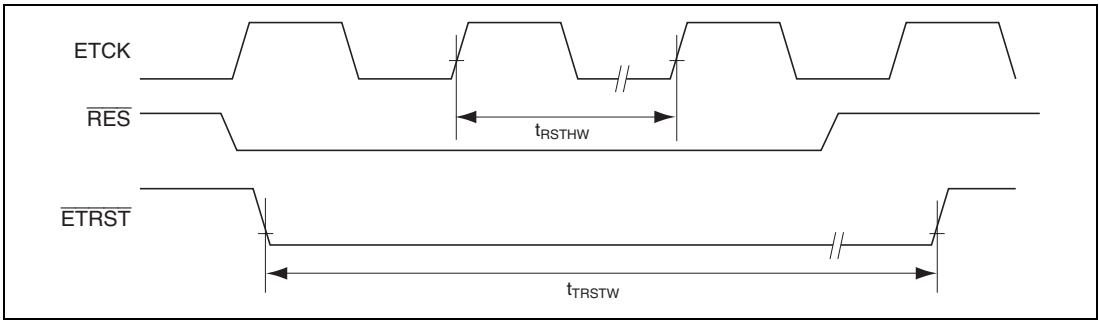


Figure 26.27 Reset Hold Timing

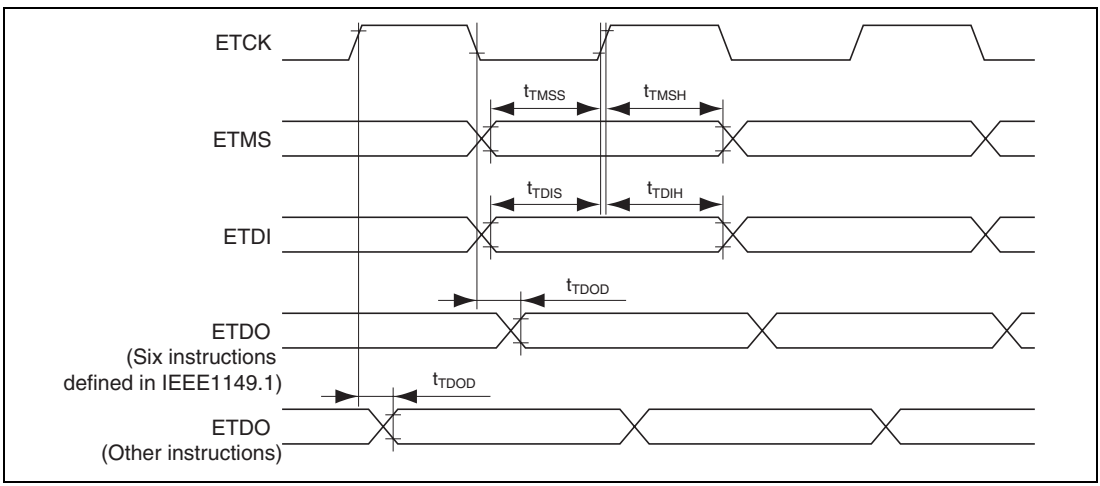


Figure 26.28 JTAG Input/Output Timing

26.3.4 A/D Conversion Characteristics

Table 26.12 lists the A/D conversion characteristics.

Table 26.12 A/D Conversion Characteristics (AN7 to AN0 Input: 134/266-State Conversion)

Condition A: 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 = 4 MHz to 16 MHz

Condition B: 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 = 4 MHz to 20 MHz

	Condition A			Condition B			
Item	**Min.**	**Typ.**	**Max.**	**Min.**	**Typ.**	**Max.**	**Unit**
Resolution		10			10		Bits
Conversion time	—	—	8.38*[1]	—	—	13.4*[2]	µs
Analog input capacitance	—	—	20	—	—	20	pF
Permissible signal-source impedance	—	—	5	—	—	5	kΩ
Nonlinearity error	—	—	±7.0	—	—	±7.0	LSB
Offset error	—	—	±7.5	—	—	±7.5	
Full-scale error	—	—	±7.5	—	—	±7.5	
Quantization error	—	—	±0.5	—	—	±0.5	
Absolute accuracy	—	—	±8.0	—	—	±8.0	

Notes: 1. Value when using the maximum operating frequency in single mode of 134 states.
2. Value when using the maximum operating frequency in single mode of 266 states.

26.4 Flash Memory Characteristics

Table 26.13 lists the flash memory characteristics.

Table 26.13 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	—	3	30	ms/128 bytes	
Erase time*1*2*4	t_E	—	80	800	ms/4-kbyte block	
		—	500	5000	ms/32-kbyte block	
		—	1000	10000	ms/64-kbyte block	
Programming time (total)*1*2*4	Σt_P	—	20	60	s/1 Mbyte	Ta = 25°C
Erase time (total)*1*2*4	Σt_E	—	20	60	s/1 Mbytes	Ta = 25°C
Programming and Erase time (total)*1*2*4	Σt_{PE}	—	40	120	s/1 Mbytes	Ta = 25°C
Reprogramming count	N_{WEC}	100*3	—	—	Times	
Data retention time*4	t_{DRP}	10	—	—	Years	

Notes: 1. Programming and erase time depends on the data.
2. Programming and erase time do not include data transfer time.
3 This value indicates the minimum number of which the flash memory are reprogrammed with all characteristics guaranteed. (The guaranteed value 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).

26.5 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 26.29.

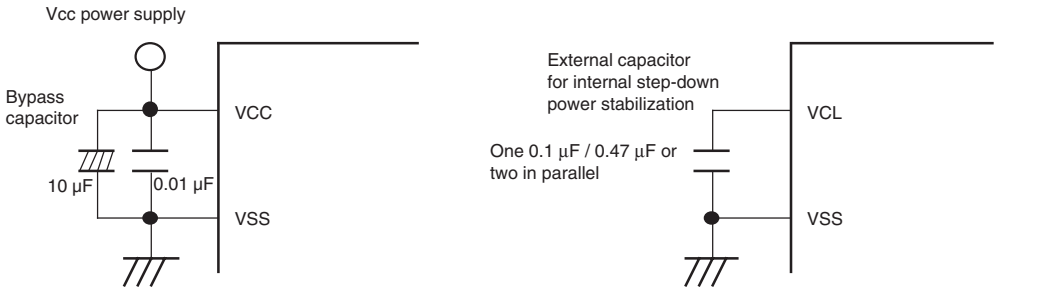


Figure 26.29 Connection of VCL Capacitor

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	MCU Operating Mode	Reset	Hardware Standby Mode	Software Standby Mode	Watch Mode	Sleep Mode	Sub-Sleep Mode	Sub-Active Mode	Program Execution State
Port 1	2, 3	T	T	keep	keep	keep	keep	I/O port	I/O port
Port 2	2, 3	T	T	keep	keep	keep	keep	I/O port	I/O port
Port 3	2, 3	T	T	keep	keep	keep	keep	I/O port	I/O port
Port 4	2, 3	T	T	keep	keep	keep	keep	I/O port	I/O port
Port 50 ExEXCL	2, 3	T	T	keep	ExEXCL input /keep	keep	ExEXCL input/ keep	ExEXCL input/ I/O port	ExEXCL input/ I/O port
Port 51, 52	2, 3	T	T	keep	keep	keep	keep	I/O port	I/O port
Port 6	2, 3	T	T	keep	keep	keep	keep	I/O port	I/O port
Port 7, E	2, 3	T	T	T	T	T	T	Input port	Input port
Port 8	2, 3	T	T	keep	keep	keep	keep	I/O port	I/O port
Port 97	2, 3	T	T	keep	keep	keep	keep	I/O port	I/O port
Port 96 ϕ, EXCL	2, 3	T	T	[DDR = 1]H [DDR = 0]T	EXCL input/ keep	[DDR = 1] Clock output [DDR = 0]T	EXCL input/ keep	EXCL input/ Input port	Clock output/ EXCL input/ input port
Port 95 to 90	2, 3	T	T	keep	keep	keep	keep	I/O port	I/O port
Port A to D, F, G	2, 3	T	T	keep	keep	keep	keep	I/O port	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.

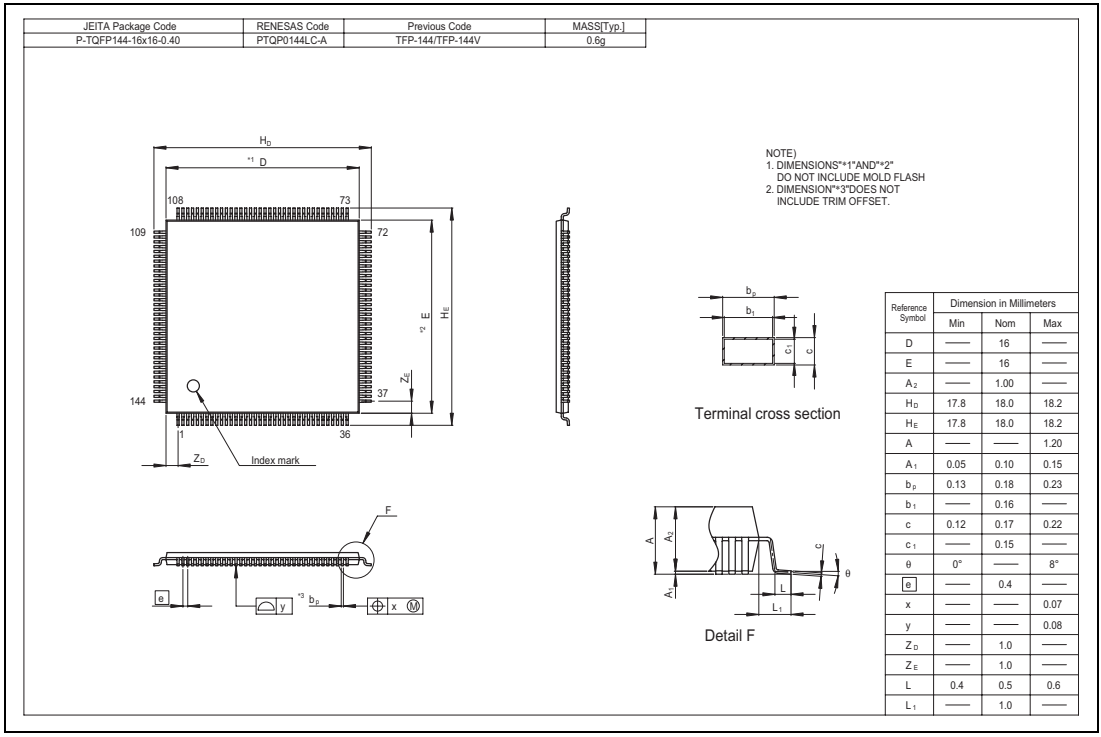
DDR: Data direction register

B. Product Lineup

Product Type		Type Code	Mark Code	Package (Code)
H8S/2114R	F-ZTAT version	R4F2114R	F2114RVTE20	PTQP0144LC-A (TFP-144)

C. Package Dimensions

For package dimensions, dimensions described in Renesas Semiconductor Packages Data Book have priority.



Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	—	16	—
E	—	16	—
A_2	—	1.00	—
H_D	17.8	18.0	18.2
H_E	17.8	18.0	18.2
A	—	—	1.20
A_1	0.05	0.10	0.15
b_p	0.13	0.18	0.23
b_1	—	0.16	—
c	0.12	0.17	0.22
c_1	—	0.15	—
θ	0°	—	8°
e	—	0.4	—
x	—	—	0.07
y	—	—	0.08
Z_D	—	1.0	—
Z_E	—	1.0	—
L	0.4	0.5	0.6
L_1	—	1.0	—

Figure C.1 Package Dimensions (TFP-144)

Index

Numerics

14-bit PWM timer (PWMX) 259
16-bit count mode 402
16-bit free-running timer (FRT) 277
16-bit timer pulse unit 305
8-bit PWM timer (PWM) 247
8-bit timer (TMR) 377

A

A/D conversion time 724
A/D converter 717
A/D converter activation 359
A20 gate 680
ABRKCR 85, 890, 904, 915, 926, 933
Absolute address 49
Activation by interrupt 155
Activation by software 155
ADCR 722, 895, 907, 919, 930, 944
ADCSR 721, 895, 907, 919, 930, 944
Additional pulse 255
ADDR 720, 894, 907, 919, 930, 944
Address map 67
Address space 26
Addressing modes 48
ADI 727
Arithmetic operations instructions 38
ASSR 668, 884, 898, 910, 920, 942

B

BARA 86, 890, 904, 915, 926, 933
BARB 86, 890, 904, 915, 926, 933
BARC 86, 890, 904, 915, 926, 933
Basic pulse 254
BCR 130, 894, 906, 918, 929, 933
Bit manipulation instructions 42
Bit rate 445
Block data transfer instructions 46
Block transfer mode 150
Boot mode 764
Boundary scan 844
Boundary scan (JTAG) 827
Branch instructions 44
BRR 445, 891, 904, 916, 927, 940
Buffer operation 343
Bus arbitration 132
Bus controller (BSC) 129

C

Carrier frequency 250
Cascaded connection 402
Chain transfer 151
Clock pulse generator 849
Clocked synchronous mode 472
CMI 406
CMIA 406
CMIAY 406
CMIB 406
CMIBY 406
Communications protocol 799
Compare-match count mode 402
Condition field 47
Condition-code register 30
Conversion cycle 269
CPU 19
CPU operating modes 22
CRA 140
CRB 140
Crystal resonator 850

D

DACNT 261, 888, 902, 913, 924, 936
DACR.....264, 888, 902, 913, 928, 936
DADRA.....262, 888, 892, 902, 928, 936
DADRB.....263, 888, 892, 902, 928, 936
DAR..... 140
Data direction register 159
Data register..... 159
Data transfer controller (DTC) 135
Data transfer instructions..... 37
DDCSWR.....530, 890, 903, 915, 926, 941
Direct transitions 878
Download pass/fail result parameter..... 754
DTC vector table 144
DTCERA.....141, 890, 903, 915, 926, 933
DTCERB 141, 890, 903, 915, 926, 934
DTCERC 141, 890, 903, 915, 926, 934
DTCERD.....141, 890, 903, 915, 926, 934
DTCERE 141, 890, 903, 915, 926, 934
DTVECR.....142, 890, 903, 915, 926, 934

E

EBLKR.....657, 883, 897, 909, 920, 942
Effective address 52
Effective address extension 47
ERI1..... 496
ERI2..... 496
ERRI..... 710
Error protection 793
Exception handling..... 69
Exception handling vector table 70, 72
Extended control register..... 29
External clock..... 851
External trigger..... 726

F

FCCS 746, 888, 902, 913, 924, 944
FECS 749, 888, 902, 913, 924, 945
FKEY.....750, 888, 902, 913, 924, 945
Flash erase block select parameter 761
Flash MAT configuration 739
Flash multipurpose address area parameter 758
Flash multipurpose data destination area parameter 758
Flash pass/fail result parameter..... 762
Flash programming/erasing frequency control parameter 756
FLWAR 655, 884, 898, 910, 921, 942
FMATS..... 751, 888, 902, 913, 924, 945
FOVI..... 298
FPCS..... 749, 888, 902, 913, 924, 945
Framing error..... 462
FRC..... 280, 891, 904, 916, 927, 937
Free-running count operation..... 336
FRPR 674, 884, 898, 910, 921, 942
FTDAR 752, 888, 902, 913, 924, 945
FWPR 672, 884, 898, 910, 921, 942

G

General registers 28

H

Hardware protection 791
Hardware standby mode 874
HBAR 665, 883, 897, 909, 920, 942
HICR0..... 619, 886, 900, 912, 922, 944
HICR1..... 622, 886, 900, 912, 923, 944
HICR2..... 625, 886, 900, 912, 923, 944
HICR3..... 628, 886, 900, 912, 923, 944
HISEL 653, 886, 900, 912, 922, 944
Host interface LPC interface (LPC)..... 613

I

I/O ports..... 159
I^2C bus data format 535
I^2C bus interface (IIC)..... 505
ICCR..... 517, 894, 914, 918, 930, 940
ICDR..... 510, 894, 907, 918, 930, 940

ICIA 298
ICIB 298
ICIC 298
ICID 298
ICIX 406
ICMR 514, 894, 907, 919, 930, 940
ICRA 83, 890, 903, 915, 926, 933
ICRB 83, 890, 903, 915, 926, 933
ICRC 83, 890, 903, 915, 926, 933
ICRD 83, 888, 901, 913, 924, 933
ICSR 526, 894, 907, 918, 930, 940
ICXR 531, 889, 903, 914, 925, 940
IDR 632, 883, 897, 909, 920, 944
IER 90, 893, 906, 918, 929, 933
IER16 .....90, 890, 904, 915, 926, 933
IICI 569
Immediate 50
Input capture 292
Input capture function 339
Input capture operation 404
Input pull-up MOSs 159
Instruction set 35
Interface 427
Internal block diagram 3
Interrupt control modes 109
Interrupt controller 79
Interrupt exception handling 76
Interrupt exception handling sequence 117
Interrupt exception handling vector table 102
Interrupt mask bit 30
Interval timer mode 421
IrDA 492
IRQ15 to IRQ0 interrupts 99
ISCR16H 87, 890, 904, 915, 926, 933
ISCR16L 88, 891, 904, 915, 926, 933
ISCRH 89, 890, 903, 915, 926, 933
ISCRL 89, 890, 903, 915, 926, 933
ISR 91, 890, 903, 915, 926, 933
ISR16 ..... 91, 890, 904, 915, 926, 933
ISSR 97, 891, 904, 916, 926, 933
ISSR16 ..... 97, 891, 904, 916, 926, 933

K

KBBR 592, 889, 903, 914, 925, 941
KBCOMP 453, 890, 903, 915, 926, 940
KBCR1 ..... 585, 888, 902, 914, 925, 941
KBCR2 ..... 587, 890, 903, 914, 925, 941
KBCRH 588, 889, 903, 914, 925, 941
KBCRL 590, 889, 903, 914, 925, 941
KBTR 592, 888, 902, 914, 925, 941
Keyboard buffer controller 581
KIN15 to KIN0 interrupts 100
KMIMR 93, 888, 895, 901, 913, 924
KMIMRA 93, 888, 895, 901, 913, 924
KMPCR 181, 888, 901, 913, 924, 934

L

LADR3 .... 628, 629, 886, 899, 911, 922, 943
LADR4 ..... 631, 883, 897, 909, 920, 944
List of registers 881
LMCCR 662, 883, 897, 909, 920, 942
LMCST 658, 883, 897, 909, 920, 942
Logic operations instructions 40
LPWRCR 862, 891, 904, 916, 926, 945
LSI internal states in each operating mode 868

M

MDCR 60, 894, 906, 918, 929, 945
Medium-speed mode 870
Memory indirect 51
Mode comparison 738
Mode transition diagram 867
Module stop mode 878

MPCR.....676, 883, 897, 909, 920, 942
MRA 138
MRB 139
MSTPCRA 865, 887, 901, 913, 924, 945
MSTPCRH 864, 891, 904, 916, 927, 945
MSTPCRL.....864, 891, 904, 916, 927, 945
Multiprocessor communication function..... 466

N

NMI interrupt..... 99
Noise canceler 567
Normal mode 22, 148, 156
Number of DTC execution states 154

O

OCIA 298
OCIB 298
OCR..... 927
OCRA 280, 891, 904, 916, 927, 937
OCRAF.....281, 892, 905, 916, 927, 937
OCRAR 281, 892, 905, 916, 927, 937
OCRB 280, 892, 904, 916, 927, 937
OCRDM 281, 892, 905, 916, 927, 937
ODR.....633, 883, 897, 909, 920, 922, 943
On-board programming 764
On-board programming mode 735
Operating modes..... 59
Operation field..... 47
Output compare 291
Overflow..... 420
Overrun error 462
OVI..... 406
OVIY 406

P

P1DDR 164, 893, 905, 917, 929, 934
P1DR..... 165, 893, 905, 917, 929, 934
P1PCR..... 165, 893, 905, 917, 928, 934
P2DDR..... 167, 893, 905, 917, 929, 934
P2DR..... 168, 893, 905, 917, 929, 934
P2PCR..... 168, 893, 905, 917, 928, 934
P3DDR..... 171, 893, 905, 917, 929, 934
P3DR..... 172, 893, 906, 917, 929, 934
P3PCR..... 172, 893, 905, 917, 928, 934
P4DDR..... 174, 893, 906, 917, 929, 934
P4DR..... 175, 893, 906, 917, 929, 934
P5DDR..... 178, 893, 906, 917, 929, 934
P5DR..... 178, 893, 906, 917, 929, 934
P6DDR..... 180, 893, 906, 917, 929, 934
P6DR..... 181, 893, 906, 917, 929, 934
P6NCCS..... 183, 884, 898, 910, 921, 934
P6NCE 182, 884, 898, 910, 921, 934
P6NCMC 182, 884, 898, 910, 921, 934
P7PIN..... 189, 893, 906, 918, 929, 934
P8DDR..... 191, 893, 906, 917, 929, 934
P8DR..... 192, 893, 906, 918, 929, 934
P9DDR..... 196, 893, 906, 918, 929, 935
P9DR..... 197, 893, 906, 918, 929, 935
P9PCR..... 197, 885, 899, 910, 921, 934
PADDR..... 201, 893, 905, 917, 928, 935
PAODR..... 202, 893, 905, 917, 928, 935
PAPIN..... 202, 893, 905, 917, 928, 935
Parity error..... 462
PBDDR..... 204, 893, 906, 918, 929, 935
PBODR..... 205, 893, 906, 917, 929, 935
PBPIN..... 205, 893, 906, 917, 929, 935
PCDDR..... 209, 887, 900, 912, 923, 935
PCNCCS..... 212, 884, 898, 910, 921, 935
PCNCE 211, 884, 898, 910, 921, 935
PCNCMC..... 211, 884, 898, 910, 921, 935
PCNOCR 215, 885, 899, 911, 921, 935
PCODR..... 210, 887, 900, 912, 923, 935
PCPIN..... 210, 887, 900, 912, 923, 935
PCSR 265, 891, 904, 916, 926, 936
PDDDR..... 217, 887, 900, 912, 923, 935
PDNOCR 223, 885, 899, 911, 921, 935

PDODR.....218, 887, 900, 912, 923, 935
PDPIN.....218, 887, 900, 912, 923, 935
PEPCR.....225, 886, 900, 912, 923, 935
PEPIN.....225, 886, 900, 912, 923, 935
Periodic count operation 336
PFNOCR.....231, 885, 899, 911, 921, 935
PFODR 228, 886, 900, 912, 923, 935
PFPIN 228, 886, 900, 912, 923, 935
PGDDR.....233, 886, 900, 912, 923, 936
PGNCCS.....236, 885, 898, 910, 921, 935
PGNCE.....235, 885, 898, 910, 921, 935
PGNCMC 235, 885, 898, 910, 921, 935
PGNOCR.....242, 885, 899, 910, 921, 935
PGODR.....234, 886, 900, 912, 923, 935
PGPIN.....234, 886, 900, 912, 923, 936
Phase counting mode 352
Pin arrangement 4
Pin functions 10
Power-down modes 859
Procedure program 783
Processing states 54
Program counter 29
Program-counter relative 50
Programmer mode 796
Programming/erasing interface parameter 753
Programming/erasing interface register 746
Protection..... 791
PTCNT0 .....243, 885, 898, 910, 921, 936
PTCNT1 .....244, 885, 899, 910, 921, 936
PTCNT2 .....245, 885, 899, 910, 921, 936
PWDPRB.....252, 894, 907, 918, 930, 936
PWDR.....251, 894, 907, 918, 936
PWM conversion period 250
PWM modes 347
PWOERB 252, 894, 907, 918, 930, 936
PWSL 250, 894, 907, 918, 930, 936

R

RAM 733
RAMAR..... 671, 884, 898, 910, 921, 942
RAMASSR 669, 884, 898, 910, 920, 942
RAMBAR 667, 884, 898, 909, 920, 942
RBUFAR 654, 883, 897, 909, 920, 942
RDR 431, 891, 904, 916, 927, 940
Register addresses 883, 933
Register bits 897
Register configuration..... 27
Register direct 48
Register field..... 47
Register indirect 48
Register indirect with displacement 49
Register indirect with post-increment 49
Register indirect with pre-decrement 49
Register selection condition 920
Register states in each operating mode ... 909
Repeat mode 149
Reset 74
Reset exception handling 74
Resolution 250
RSR..... 431
RXI1 496
RXI2 496

S

SAR..... 511, 889, 903, 914, 930, 940
SAR1..... 670, 884, 898, 910, 921, 942
SAR2..... 671, 884, 898, 910, 921, 942
SARX..... 512, 894, 907, 918, 930, 940
SBYCR 860, 891, 904, 916, 926, 945
Scan mode..... 723
SCMR 444, 891, 904, 916, 927, 940
SCR..... 436, 891, 904, 916, 927, 940
SDBPR..... 832
SDBSR..... 832

SDIDR 842
SDIR 830
Serial communication interface (SCI) 427
Serial communication interface specifications 797
Serial data reception 462, 477
Serial data transmission 460, 474
Serial formats 535
Shift instructions 41
Single mode 723
SIRQCR 640, 886, 899, 911, 922, 943
Sleep mode 871
Smart card 427
SMI 712
SMR 432, 891, 904, 916, 927, 940
Software protection 793
Software standby mode 872
SSR 439, 891, 904, 916, 927, 940
Stack pointer 28
Stack status 77
STCR 63, 893, 906, 918, 929, 945
STR 634, 886, 897, 911, 922, 943
Subactive mode 877
Subsleep mode 876
Synchronous operation 341
SYSCR 61, 894, 906, 918, 929, 945
SYSCR2 .....185, 891, 904, 916, 926, 934
SYSCR3 .....66, 887, 901, 913, 924, 945
System control instructions 45

T

TAP controller 842
TCI0V 358
TCI1U 358
TCI1V 358
TCI2U 358
TCI2V 358
TCNT .....331, 383, 415, 923, ...930, 937, 938, 939
TCNT_0 ..... 383, 894, 906, 918, 930, 937
TCONRI 395, 896, 907, 919, 932, 939
TCONRS 396, 896, 908, 919, 932, 939
TCORA 383, 894, 906, 918, 929, 938
TCORB 384, 894, 906, 918, 930, 938
TCORC 394, 896, 907, 919, 931, 939
TCR ..... 286, 311, 384, 923, ... 927, 929, 937, 938
TCRXY 396, 889, 902, 914, 925, 939
TCSR 283, 389, 927, 929, 937, 938
TDR 432, 891, 904, 916, 927, 940
TEI1 496
TEI2 496
TGI0A 358
TGI0B 358
TGI0C 358
TGI0D 358
TGI1A 358
TGI1B 358
TGI2A 358
TGI2B 358
TGR 331, 887, 901, 913, 923, 937
TGRA 883, 897, 909, 920, 923, 938
TICRF 394, 895, 907, 919, 931, 938
TICRR 394, 895, 907, 919, 931, 938
TIER 282, 891, 904, 916, 927, 937
TIOR 317, 883, 897, 909, 920, 937
TISR 395, 889, 902, 914, 925, 939
TMDR 315, 883, 897, 909, 920, 937
TOCR 287, 892, 905, 916, 927, 937
Toggle output 338
Trap instruction exception handling 76
TRAPA instruction 76
TSR 328, 432, 883, 897, 909, 923, 937
TSTR 331, 888, 902, 914, 924, 938
TSYR 332, 888, 902, 914, 924, 938
TWR 633, 885, 899, 911, 922, 943
TXI1 496
TXI2 496

U

UCMDTR 676, 884, 898, 910, 921, 942
User boot MAT 795
User boot memory MAT 735
User boot mode 779
User MAT 735, 795
User program mode 768

V

Vector number for the software activation interrupt 142

W

Watch mode 875
Watchdog timer (WDT) 413
Watchdog timer mode 420
Waveform output by compare match 337
WOVI 423
WSCR 131, 894, 906, 918, 929, 933
WUE15 to WUE0 interrupts 100
WUEMR 94, 886, 900, 912, 923, 933
WUEMRB 94, 886, 900, 912, 923, 933

Renesas 16-Bit Single-Chip Microcomputer
Hardware Manual
H8S/2114R Group

Publication Date:	Rev.1.00, Oct 31, 2003
	Rev.3.00, Jul. 14, 2005
Published by:	Sales Strategic Planning Div.
	Renesas Technology Corp.
Edited by:	Technical Documentation & Information Department
	Renesas Kodaira Semiconductor Co., Ltd.

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, 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, United Kingdom
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

Renesas Technology Hong Kong Ltd.
7th Floor, North Tower, World Finance Centre, Harbour City, 1 Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2730-6071

Renesas Technology Taiwan Co., Ltd.
10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 2713-2999

Renesas Technology (Shanghai) Co., Ltd.
Unit2607 Ruijing Building, No.205 Maoming Road (S), Shanghai 200020, China
Tel: <86> (21) 6472-1001, Fax: <86> (21) 6415-2952

Renesas Technology Singapore Pte. Ltd.
1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

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, Jalan Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: <603> 7955-9390, Fax: <603> 7955-9510

Colophon 3.0

H8S/2114R Group
Hardware Manual

Renesas Technology Corp.
2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan