To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

Downloaded from Arrow.com.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

Downloaded from Arrow.com.

User's Manual

16

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.

H8/3664 Group

Hardware Manual

Renesas 16-Bit Single-Chip Microcomputer
H8 Family/H8/300H Tiny Series

H8/3664N	HD64N3664
H8/3664F	HD64F3664
H8/3664	HD6433664
H8/3663	HD6433663
H8/3662	HD6433662
H8/3661	HD6433661
H8/3660	HD6433660

Renesas Electronics
www.renesas.com

Rev.6.00 2006.03

Downloaded from Arrow.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)

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

The H8/3664 Group are single-chip microcomputers made up of the high-speed H8/300H CPU employing Renesas Technology original architecture as their cores, and the peripheral functions required to configure a system. The H8/300H CPU has an instruction set that is compatible with the H8/300 CPU.

Target Users: This manual was written for users who will be using the H8/3664 Group in the design of application systems. Target users are expected to understand the fundamentals of electrical circuits, logical circuits, and microcomputers.

Objective: This manual was written to explain the hardware functions and electrical characteristics of the H8/3664 Group to the target users.
Refer to the H8/300H Series Software Manual for a detailed description of the instruction set.

Notes on reading this manual:

- In order to understand the overall functions of the chip
  Read the manual according to the contents. This manual can be roughly categorized into parts on the CPU, system control functions, peripheral functions and electrical characteristics.
- In order to understand the details of the CPU's functions
  Read the H8/300H Series Software Manual.
- In order to understand the details of a register when its name is known
  Read the index that is the final part of the manual to find the page number of the entry on the register. The addresses, bits, and initial values of the registers are summarized in section 19, List of Registers.
  Example: Bit order: The MSB is on the left and the LSB is on the right.

Notes:

When using the on-chip emulator (E7, E8) for H8/3664 program development and debugging, the following restrictions must be noted.

1. The $\overline{\text{NMI}}$ pin is reserved for the E7 or E8, and cannot be used.
2. Pins P85, P86, and P87 cannot be used. In order to use these pins, additional hardware must be provided on the user board.
3. Area H'7000 to H'7FFF is used by the E7 or E8, and is not available to the user.
4. Area H'F780 to H'FB7F must on no account be accessed.

5. When the E7 or E8 is used, address breaks can be set as either available to the user or for use by the E7 or E8. If address breaks are set as being used by the E7 or E8, the address break control registers must not be accessed.
6. When the E7 or E8 is used, $\overline{\text{NMI}}$ is an input/output pin (open-drain in output mode), P85 and P87 are input pins, and P86 is an output pin.

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/

H8/3664 Group manuals:

Document Title	Document No.
H8/3664 Group Hardware Manual	This manual
H8/300H Series Software Manual	REJ09B0213

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
H8S, H8/300 Series Simulator/Debugger User's Manual	REJ10B0211
H8S, H8/300 Series High-Performance Embedded Workshop 3 Tutorial	REJ10B0024
H8S, H8/300 Series High-performance Embedded Workshop 3 User's Manual	REJ10B0026

Application notes:

Document Title	Document No.
H8S, H8/300 Series C/C++ Compiler Package Application Note	REJ05B0464
Single Power Supply F-ZTAT™ On-Board Programming	REJ05B0520

Contents

Section 1 Overview 1
1.1 Features 1
1.2 Internal Block Diagram 3
1.3 Pin Arrangement 5
1.4 Pin Functions 9

Section 2 CPU 13
2.1 Address Space and Memory Map 14
2.2 Register Configuration 17
2.2.1 General Registers 18
2.2.2 Program Counter (PC) 19
2.2.3 Condition-Code Register (CCR) 19
2.3 Data Formats 21
2.3.1 General Register Data Formats 21
2.3.2 Memory Data Formats 23
2.4 Instruction Set 24
2.4.1 Table of Instructions Classified by Function 24
2.4.2 Basic Instruction Formats 34
2.5 Addressing Modes and Effective Address Calculation 35
2.5.1 Addressing Modes 35
2.5.2 Effective Address Calculation 39
2.6 Basic Bus Cycle 41
2.6.1 Access to On-Chip Memory (RAM, ROM) 41
2.6.2 On-Chip Peripheral Modules 42
2.7 CPU States 43
2.8 Usage Notes 44
2.8.1 Notes on Data Access to Empty Areas 44
2.8.2 EEPMOV Instruction 44
2.8.3 Bit Manipulation Instruction 45

Section 3 Exception Handling 51
3.1 Exception Sources and Vector Address 51
3.2 Register Descriptions 53
3.2.1 Interrupt Edge Select Register 1 (IEGR1) 53
3.2.2 Interrupt Edge Select Register 2 (IEGR2) 54
3.2.3 Interrupt Enable Register 1 (IENR1) 55

3.2.4 Interrupt Flag Register 1 (IRR1) ... 56
3.2.5 Wakeup Interrupt Flag Register (IWPR) ... 57
3.3 Reset Exception Handling ... 59
3.4 Interrupt Exception Handling ... 59
3.4.1 External Interrupts ... 59
3.4.2 Internal Interrupts ... 61
3.4.3 Interrupt Handling Sequence ... 61
3.4.4 Interrupt Response Time ... 62
3.5 Usage Notes ... 64
3.5.1 Interrupts after Reset ... 64
3.5.2 Notes on Stack Area Use ... 64
3.5.3 Notes on Rewriting Port Mode Registers ... 64

Section 4 Address Break ... 67
4.1 Register Descriptions ... 68
4.1.1 Address Break Control Register (ABRKCR) ... 68
4.1.2 Address Break Status Register (ABRKSR) ... 70
4.1.3 Break Address Registers (BARH, BARL) ... 70
4.1.4 Break Data Registers (BDRH, BDRL) ... 70
4.2 Operation ... 71
4.3 Usage Notes ... 73

Section 5 Clock Pulse Generators ... 77
5.1 System Clock Generator ... 78
5.1.1 Connecting Crystal Resonator ... 78
5.1.2 Connecting Ceramic Resonator ... 79
5.1.3 External Clock Input Method ... 79
5.2 Subclock Generator ... 80
5.2.1 Connecting 32.768-kHz Crystal Resonator ... 80
5.2.2 Pin Connection when Not Using Subclock ... 81
5.3 Prescalers ... 81
5.3.1 Prescaler S ... 81
5.3.2 Prescaler W ... 81
5.4 Usage Notes ... 82
5.4.1 Note on Resonators ... 82
5.4.2 Notes on Board Design ... 82

Section 6 Power-Down Modes ... 83
6.1 Register Descriptions ... 84
6.1.1 System Control Register 1 (SYSCR1) ... 84

6.1.2 System Control Register 2 (SYSCR2) ... 86
6.1.3 Module Standby Control Register 1 (MSTCR1) ... 87
6.2 Mode Transitions and States of LSI ... 88
6.2.1 Sleep Mode ... 91
6.2.2 Standby Mode ... 91
6.2.3 Subsleep Mode ... 92
6.2.4 Subactive Mode ... 92
6.3 Operating Frequency in Active Mode ... 93
6.4 Direct Transition ... 93
6.4.1 Direct Transition from Active Mode to Subactive Mode ... 93
6.4.2 Direct Transition from Subactive Mode to Active Mode ... 94
6.5 Module Standby Function ... 94
6.6 Usage Note ... 94

Section 7 ROM ... 95
7.1 Block Configuration ... 96
7.2 Register Descriptions ... 97
7.2.1 Flash Memory Control Register 1 (FLMCR1) ... 97
7.2.2 Flash Memory Control Register 2 (FLMCR2) ... 98
7.2.3 Erase Block Register 1 (EBR1) ... 99
7.2.4 Flash Memory Power Control Register (FLPWCR) ... 99
7.2.5 Flash Memory Enable Register (FENR) ... 100
7.3 On-Board Programming Modes ... 100
7.3.1 Boot Mode ... 101
7.3.2 Programming/Erasing in User Program Mode ... 103
7.4 Flash Memory Programming/Erasing ... 104
7.4.1 Program/Program-Verify ... 104
7.4.2 Erase/Erase-Verify ... 107
7.4.3 Interrupt Handling when Programming/Erasing Flash Memory ... 107
7.5 Program/Erase Protection ... 109
7.5.1 Hardware Protection ... 109
7.5.2 Software Protection ... 109
7.5.3 Error Protection ... 109
7.6 Programmer Mode ... 110
7.7 Power-Down States for Flash Memory ... 110

Section 8 RAM ... 113

Section 9 I/O Ports ... 115
9.1 Port 1 ... 115

9.1.1 Port Mode Register 1 (PMR1) 116
9.1.2 Port Control Register 1 (PCR1) 117
9.1.3 Port Data Register 1 (PDR1) 118
9.1.4 Port Pull-Up Control Register 1 (PUCR1) 118
9.1.5 Pin Functions 119
9.2 Port 2 121
9.2.1 Port Control Register 2 (PCR2) 121
9.2.2 Port Data Register 2 (PDR2) 122
9.2.3 Pin Functions 122
9.3 Port 5 124
9.3.1 Port Mode Register 5 (PMR5) 125
9.3.2 Port Control Register 5 (PCR5) 126
9.3.3 Port Data Register 5 (PDR5) 127
9.3.4 Port Pull-Up Control Register 5 (PUCR5) 127
9.3.5 Pin Functions 128
9.4 Port 7 130
9.4.1 Port Control Register 7 (PCR7) 131
9.4.2 Port Data Register 7 (PDR7) 131
9.4.3 Pin Functions 132
9.5 Port 8 133
9.5.1 Port Control Register 8 (PCR8) 134
9.5.2 Port Data Register 8 (PDR8) 134
9.5.3 Pin Functions 135
9.6 Port B 138
9.6.1 Port Data Register B (PDRB) 138

Section 10 Timer A 139
10.1 Features 139
10.2 Input/Output Pins 140
10.3 Register Descriptions 141
10.3.1 Timer Mode Register A (TMA) 141
10.3.2 Timer Counter A (TCA) 142
10.4 Operation 143
10.4.1 Interval Timer Operation 143
10.4.2 Clock Time Base Operation 143
10.4.3 Clock Output 143
10.5 Usage Note 144

Section 11 Timer V 145
11.1 Features 145

11.2 Input/Output Pins 147
11.3 Register Descriptions 147
11.3.1 Timer Counter V (TCNTV) 147
11.3.2 Time Constant Registers A and B (TCORA, TCORB) 148
11.3.3 Timer Control Register V0 (TCRV0) 148
11.3.4 Timer Control/Status Register V (TCSRV) 150
11.3.5 Timer Control Register V1 (TCRV1) 151
11.4 Operation 152
11.4.1 Timer V Operation 152
11.5 Timer V Application Examples 155
11.5.1 Pulse Output with Arbitrary Duty Cycle 155
11.5.2 Pulse Output with Arbitrary Pulse Width and Delay from TRGV Input 156
11.6 Usage Notes 157

Section 12 Timer W 159
12.1 Features 159
12.2 Input/Output Pins 162
12.3 Register Descriptions 162
12.3.1 Timer Mode Register W (TMRW) 163
12.3.2 Timer Control Register W (TCRW) 164
12.3.3 Timer Interrupt Enable Register W (TIERW) 165
12.3.4 Timer Status Register W (TSRW) 166
12.3.5 Timer I/O Control Register 0 (TIOR0) 167
12.3.6 Timer I/O Control Register 1 (TIOR1) 169
12.3.7 Timer Counter (TCNT) 170
12.3.8 General Registers A to D (GRA to GRD) 171
12.4 Operation 172
12.4.1 Normal Operation 172
12.4.2 PWM Operation 176
12.5 Operation Timing 181
12.5.1 TCNT Count Timing 181
12.5.2 Output Compare Output Timing 182
12.5.3 Input Capture Timing 183
12.5.4 Timing of Counter Clearing by Compare Match 183
12.5.5 Buffer Operation Timing 184
12.5.6 Timing of IMFA to IMFD Flag Setting at Compare Match 185
12.5.7 Timing of IMFA to IMFD Setting at Input Capture 186
12.5.8 Timing of Status Flag Clearing 186
12.6 Usage Notes 187

Section 13 Watchdog Timer 191
13.1 Features 191
13.2 Register Descriptions 191
13.2.1 Timer Control/Status Register WD (TCSRWD) 192
13.2.2 Timer Counter WD (TCWD) 193
13.2.3 Timer Mode Register WD (TMWD) 194
13.3 Operation 195

Section 14 Serial Communication Interface 3 (SCI3) 197
14.1 Features 197
14.2 Input/Output Pins 198
14.3 Register Descriptions 199
14.3.1 Receive Shift Register (RSR) 199
14.3.2 Receive Data Register (RDR) 199
14.3.3 Transmit Shift Register (TSR) 199
14.3.4 Transmit Data Register (TDR) 200
14.3.5 Serial Mode Register (SMR) 200
14.3.6 Serial Control Register 3 (SCR3) 201
14.3.7 Serial Status Register (SSR) 203
14.3.8 Bit Rate Register (BRR) 205
14.4 Operation in Asynchronous Mode 210
14.4.1 Clock 210
14.4.2 SCI3 Initialization 211
14.4.3 Data Transmission 212
14.4.4 Serial Data Reception 214
14.5 Operation in Clocked Synchronous Mode 218
14.5.1 Clock 218
14.5.2 SCI3 Initialization 218
14.5.3 Serial Data Transmission 219
14.5.4 Serial Data Reception (Clocked Synchronous Mode) 221
14.5.5 Simultaneous Serial Data Transmission and Reception 223
14.6 Multiprocessor Communication Function 224
14.6.1 Multiprocessor Serial Data Transmission 226
14.6.2 Multiprocessor Serial Data Reception 227
14.7 Interrupts 230
14.8 Usage Notes 230
14.8.1 Break Detection and Processing 230
14.8.2 Mark State and Break Sending 231
14.8.3 Receive Error Flags and Transmit Operations (Clocked Synchronous Mode Only) 231

14.8.4 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode 231

Section 15 I²C Bus Interface (IIC) 233
15.1 Features 233
15.2 Input/Output Pins 235
15.3 Register Descriptions 236
15.3.1 I²C Bus Data Register (ICDR) 236
15.3.2 Slave Address Register (SAR) 238
15.3.3 Second Slave Address Register (SARX) 238
15.3.4 I²C Bus Mode Register (ICMR) 239
15.3.5 I²C Bus Control Register (ICCR) 242
15.3.6 I²C Bus Status Register (ICSR) 245
15.3.7 Timer Serial Control Register (TSCR) 248
15.4 Operation 249
15.4.1 I²C Bus Data Format 249
15.4.2 Master Transmit Operation 251
15.4.3 Master Receive Operation 253
15.4.4 Slave Receive Operation 255
15.4.5 Slave Transmit Operation 258
15.4.6 Clock Synchronous Serial Format 259
15.4.7 IRIC Setting Timing and SCL Control 260
15.4.8 Noise Canceler 261
15.4.9 Sample Flowcharts 262
15.5 Usage Notes 266

Section 16 A/D Converter 275
16.1 Features 275
16.2 Input/Output Pins 277
16.3 Register Description 278
16.3.1 A/D Data Registers A to D (ADDRA to ADDRD) 278
16.3.2 A/D Control/Status Register (ADCSR) 279
16.3.3 A/D Control Register (ADCR) 280
16.4 Operation 281
16.4.1 Single Mode 281
16.4.2 Scan Mode 281
16.4.3 Input Sampling and A/D Conversion Time 282
16.4.4 External Trigger Input Timing 283
16.5 A/D Conversion Accuracy Definitions 284
16.6 Usage Notes 286

16.6.1 Permissible Signal Source Impedance 286
16.6.2 Influences on Absolute Accuracy 286

Section 17 EEPROM 287
17.1 Features 287
17.2 Input/Output Pins 289
17.3 Register Description 289
17.3.1 EEPROM Key Register (EKR) 289
17.4 Operation 290
17.4.1 EEPROM Interface 290
17.4.2 Bus Format and Timing 290
17.4.3 Start Condition 291
17.4.4 Stop Condition 291
17.4.5 Acknowledge 291
17.4.6 Slave Addressing 292
17.4.7 Write Operations 293
17.4.8 Acknowledge Polling 294
17.4.9 Read Operation 295
17.5 Usage Notes 297
17.5.1 Data Protection at V_{CC} On/Off 297
17.5.2 Write/Erase Endurance 297
17.5.3 Noise Suppression Time 298

Section 18 Power Supply Circuit 299
18.1 When Using Internal Power Supply Step-Down Circuit 299
18.2 When Not Using Internal Power Supply Step-Down Circuit 300

Section 19 List of Registers 301
19.1 Register Addresses (Address Order) 302
19.2 Register Bits 305
19.3 Register States in Each Operating Mode 308

Section 20 Electrical Characteristics 311
20.1 Absolute Maximum Ratings 311
20.2 Electrical Characteristics (F-ZTAT™ Version, F-ZTAT™ Version with EEPROM) 311
20.2.1 Power Supply Voltage and Operating Ranges 311
20.2.2 DC Characteristics 314
20.2.3 AC Characteristics 320
20.2.4 A/D Converter Characteristics 324
20.2.5 Watchdog Timer Characteristics 325

20.2.6 Memory Characteristics 326
20.2.7 EEPROM Characteristics 328
20.3 Electrical Characteristics (Mask ROM Version) 329
20.3.1 Power Supply Voltage and Operating Ranges 329
20.3.2 DC Characteristics 331
20.3.3 AC Characteristics 337
20.3.4 A/D Converter Characteristics 341
20.3.5 Watchdog Timer Characteristics 342
20.4 Operation Timing 343
20.5 Output Load Condition 346

Appendix A Instruction Set 347
A.1 Instruction List 347
A.2 Operation Code Map 362
A.3 Number of Execution States 365
A.4 Combinations of Instructions and Addressing Modes 376

Appendix B I/O Port Block Diagrams 377
B.1 I/O Port Block 377
B.2 Port States in Each Operating State 394

Appendix C Product Code Lineup 395

Appendix D Package Dimensions 397

Appendix E EEPROM Stacked-Structure Cross-Sectional View 401

Main Revisions and Additions in this Edition 403

Index 409

Figures

Section 1 Overview
Figure 1.1 Internal Block Diagram of H8/3664 of F-ZTAT™ and Mask-ROM Versions 3
Figure 1.2 Internal Block Diagram of H8/3664N of F-ZTAT™ Version with EEPROM 4
Figure 1.3 Pin Arrangement of H8/3664 of F-ZTAT™ and Mask-ROM Versions (FP-64E, FP-64A) 5
Figure 1.4 Pin Arrangement of H8/3664 of F-ZTAT™ and Mask-ROM Versions (FP-48F, FP-48B) 6
Figure 1.5 Pin Arrangement of H8/3664 of F-ZTAT™ and Mask-ROM Versions (DS-42S) 7
Figure 1.6 Pin Arrangement of H8/3664N of F-ZTAT™ Version with EEPROM (FP-64E) 8

Section 2 CPU
Figure 2.1 Memory Map (1) 14
Figure 2.1 Memory Map (2) 15
Figure 2.1 Memory Map (3) 16
Figure 2.2 CPU Registers 17
Figure 2.3 Usage of General Registers 18
Figure 2.4 Relationship between Stack Pointer and Stack Area 19
Figure 2.5 General Register Data Formats (1) 21
Figure 2.5 General Register Data Formats (2) 22
Figure 2.6 Memory Data Formats 23
Figure 2.7 Instruction Formats 34
Figure 2.8 Branch Address Specification in Memory Indirect Mode 38
Figure 2.9 On-Chip Memory Access Cycle 41
Figure 2.10 On-Chip Peripheral Module Access Cycle (3-State Access) 42
Figure 2.11 CPU Operation States 43
Figure 2.12 State Transitions 44
Figure 2.13 Example of Timer Configuration with Two Registers Allocated to Same Address 45

Section 3 Exception Handling
Figure 3.1 Reset Sequence 60
Figure 3.2 Stack Status after Exception Handling 62
Figure 3.3 Interrupt Sequence 63
Figure 3.4 Port Mode Register Setting and Interrupt Request Flag Clearing Procedure 65

Section 4 Address Break
Figure 4.1 Block Diagram of Address Break 67

Figure 4.2 Address Break Interrupt Operation Example (1) 71
Figure 4.2 Address Break Interrupt Operation Example (2) 72
Figure 4.3 Operation when Condition is not Satisfied in Branch Instruction 73
Figure 4.4 Operation when Another Interrupt is Accepted at Address Break Setting Instruction 74
Figure 4.5 Operation when the Instruction Set is not Executed and does not Branch due to Conditions not Being Satisfied 75

Section 5 Clock Pulse Generators

Figure 5.1 Block Diagram of Clock Pulse Generators 77
Figure 5.2 Block Diagram of System Clock Generator 78
Figure 5.3 Typical Connection to Crystal Resonator 78
Figure 5.4 Equivalent Circuit of Crystal Resonator 78
Figure 5.5 Typical Connection to Ceramic Resonator 79
Figure 5.6 Example of External Clock Input 79
Figure 5.7 Block Diagram of Subclock Generator 80
Figure 5.8 Typical Connection to 32.768-kHz Crystal Resonator 80
Figure 5.9 Equivalent Circuit of 32.768-kHz Crystal Resonator 80
Figure 5.10 Pin Connection when not Using Subclock 81
Figure 5.11 Example of Incorrect Board Design 82

Section 6 Power-Down Modes

Figure 6.1 Mode Transition Diagram 88

Section 7 ROM

Figure 7.1 Flash Memory Block Configuration 96
Figure 7.2 Programming/Erasing Flowchart Example in User Program Mode 103
Figure 7.3 Program/Program-Verify Flowchart 105
Figure 7.4 Erase/Erase-Verify Flowchart 108

Section 9 I/O Ports

Figure 9.1 Port 1 Pin Configuration 115
Figure 9.2 Port 2 Pin Configuration 121
Figure 9.3 Port 5 Pin Configuration 124
Figure 9.4 Port 7 Pin Configuration 130
Figure 9.5 Port 8 Pin Configuration 133
Figure 9.6 Port B Pin Configuration 138

Section 10 Timer A

Figure 10.1 Block Diagram of Timer A 140

Section 11 Timer V

Figure 11.1 Block Diagram of Timer V 146

Figure 11.2 Increment Timing with Internal Clock 153
Figure 11.3 Increment Timing with External Clock 153
Figure 11.4 OVF Set Timing 153
Figure 11.5 CMFA and CMFB Set Timing 154
Figure 11.6 TMOV Output Timing 154
Figure 11.7 Clear Timing by Compare Match 154
Figure 11.8 Clear Timing by TMRIV Input 155
Figure 11.9 Pulse Output Example 155
Figure 11.10 Example of Pulse Output Synchronized to TRGV Input 156
Figure 11.11 Contention between TCNTV Write and Clear 157
Figure 11.12 Contention between TCORA Write and Compare Match 158
Figure 11.13 Internal Clock Switching and TCNTV Operation 158

Section 12 Timer W

Figure 12.1 Timer W Block Diagram 161
Figure 12.2 Free-Running Counter Operation 172
Figure 12.3 Periodic Counter Operation 173
Figure 12.4 0 and 1 Output Example (TOA = 0, TOB = 1) 173
Figure 12.5 Toggle Output Example (TOA = 0, TOB = 1) 174
Figure 12.6 Toggle Output Example (TOA = 0, TOB = 1) 174
Figure 12.7 Input Capture Operating Example 175
Figure 12.8 Buffer Operation Example (Input Capture) 176
Figure 12.9 PWM Mode Example (1) 177
Figure 12.10 PWM Mode Example (2) 177
Figure 12.11 Buffer Operation Example (Output Compare) 178
Figure 12.12 PWM Mode Example
(TOB, TOC, and TOD = 0: initial output values are set to 0) 179
Figure 12.13 PWM Mode Example
(TOB, TOC, and TOD = 1: initial output values are set to 1) 180
Figure 12.14 Count Timing for Internal Clock Source 181
Figure 12.15 Count Timing for External Clock Source 181
Figure 12.16 Output Compare Output Timing 182
Figure 12.17 Input Capture Input Signal Timing 183
Figure 12.18 Timing of Counter Clearing by Compare Match 183
Figure 12.19 Buffer Operation Timing (Compare Match) 184
Figure 12.20 Buffer Operation Timing (Input Capture) 184
Figure 12.21 Timing of IMFA to IMFD Flag Setting at Compare Match 185
Figure 12.22 Timing of IMFA to IMFD Flag Setting at Input Capture 186
Figure 12.23 Timing of Status Flag Clearing by CPU 186
Figure 12.24 Contention between TCNT Write and Clear 187
Figure 12.25 Internal Clock Switching and TCNT Operation 188

Figure 12.26 When Compare Match and Bit Manipulation Instruction to TCRW Occur at the Same Timing 189

Section 13 Watchdog Timer

Figure 13.1 Block Diagram of Watchdog Timer 191
Figure 13.2 Watchdog Timer Operation Example 195

Section 14 Serial Communication Interface 3 (SCI3)

Figure 14.1 Block Diagram of SCI3 198
Figure 14.2 Data Format in Asynchronous Communication 210
Figure 14.3 Relationship between Output Clock and Transfer Data Phase (Asynchronous Mode) (Example with 8-Bit Data, Parity, Two Stop Bits) 210
Figure 14.4 Sample SCI3 Initialization Flowchart 211
Figure 14.5 Example SCI3 Operation in Transmission in Asynchronous Mode (8-Bit Data, Parity, One Stop Bit) 212
Figure 14.6 Sample Serial Transmission Flowchart (Asynchronous Mode) 213
Figure 14.7 Example SCI3 Operation in Reception in Asynchronous Mode (8-Bit Data, Parity, One Stop Bit) 214
Figure 14.8 Sample Serial Data Reception Flowchart (Asynchronous mode) (1) 216
Figure 14.8 Sample Serial Reception Data Flowchart (2) 217
Figure 14.9 Data Format in Clocked Synchronous Communication 218
Figure 14.10 Example of SCI3 Operation in Transmission in Clocked Synchronous Mode 219
Figure 14.11 Sample Serial Transmission Flowchart (Clocked Synchronous Mode) 220
Figure 14.12 Example of SCI3 Reception Operation in Clocked Synchronous Mode 221
Figure 14.13 Sample Serial Reception Flowchart (Clocked Synchronous Mode) 222
Figure 14.14 Sample Flowchart of Simultaneous Serial Transmit and Receive Operations (Clocked Synchronous Mode) 223
Figure 14.15 Example of Communication Using Multiprocessor Format (Transmission of Data H'AA to Receiving Station A) 225
Figure 14.16 Sample Multiprocessor Serial Transmission Flowchart 226
Figure 14.17 Sample Multiprocessor Serial Reception Flowchart (1) 227
Figure 14.17 Sample Multiprocessor Serial Reception Flowchart (2) 228
Figure 14.18 Example of SCI3 Operation in Reception Using Multiprocessor Format (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit) 229
Figure 14.19 Receive Data Sampling Timing in Asynchronous Mode 232

Section 15 I^2C Bus Interface (IIC)

Figure 15.1 Block Diagram of I^2C Bus Interface 234
Figure 15.2 I^2C Bus Interface Connections (Example: This LSI as Master) 235
Figure 15.3 I^2C Bus Data Formats (I^2C Bus Formats) 250
Figure 15.4 I^2C Bus Timing 250
Figure 15.5 Master Transmit Mode Operation Timing Example (MLS = WAIT = 0) 252

Figure 15.6 Master Receive Mode Operation Timing Example (1)
(MLS = ACKB = 0, WAIT = 1) 254
Figure 15.6 Master Receive Mode Operation Timing Example (2)
(MLS = ACKB = 0, WAIT = 1) 255
Figure 15.7 Example of Slave Receive Mode Operation Timing (1) (MLS = ACKB = 0) 256
Figure 15.8 Example of Slave Receive Mode Operation Timing (2) (MLS = ACKB = 0) 257
Figure 15.9 Example of Slave Transmit Mode Operation Timing (MLS = 0) 259
Figure 15.10 I^2C Bus Data Format (Serial Format) 259
Figure 15.11 IRIC Setting Timing and SCL Control 260
Figure 15.12 Block Diagram of Noise Canceler 261
Figure 15.13 Sample Flowchart for Master Transmit Mode 262
Figure 15.14 Sample Flowchart for Master Receive Mode 263
Figure 15.15 Sample Flowchart for Slave Receive Mode 264
Figure 15.16 Sample Flowchart for Slave Transmit Mode 265
Figure 15.17 Flowchart and Timing of Start Condition Instruction Issuance
for Retransmission 270
Figure 15.18 IRIC Flag Clear Timing on WAIT Operation 271
Figure 15.19 Notes on ICDR Reading with TRS = 1 Setting in Master Mode 272
Figure 15.20 Notes on ICDR Writing with TRS = 0 Setting in Slave Mode 273

Section 16 A/D Converter

Figure 16.1 Block Diagram of A/D Converter 276
Figure 16.2 A/D Conversion Timing 282
Figure 16.3 External Trigger Input Timing 283
Figure 16.4 A/D Conversion Accuracy Definitions (1) 285
Figure 16.5 A/D Conversion Accuracy Definitions (2) 285
Figure 16.6 Analog Input Circuit Example 286

Section 17 EEPROM

Figure 17.1 Block Diagram of EEPROM 288
Figure 17.2 EEPROM Bus Format and Bus Timing 290
Figure 17.3 Byte Write Operation 293
Figure 17.4 Page Write Operation 294
Figure 17.5 Current Address Read Operation 295
Figure 17.6 Random Address Read Operation 296
Figure 17.7 Sequential Read Operation (when current address read is used) 297

Section 18 Power Supply Circuit

Figure 18.1 Power Supply Connection when Internal Step-Down Circuit is Used 299
Figure 18.2 Power Supply Connection when Internal Step-Down Circuit is Not Used 300

Section 20 Electrical Characteristics
Figure 20.1 System Clock Input Timing 343
Figure 20.2 $\overline{\text{RES}}$ Low Width Timing 343
Figure 20.3 Input Timing 343
Figure 20.4 I^2C Bus Interface Input/Output Timing 344
Figure 20.5 SCK3 Input Clock Timing 344
Figure 20.6 SCI3 Input/Output Timing in Clocked Synchronous Mode 345
Figure 20.7 EEPROM Bus Timing 345
Figure 20.8 Output Load Circuit 346

Appendix B I/O Port Block Diagrams
Figure B.1 Port 1 Block Diagram (P17) 377
Figure B.2 Port 1 Block Diagram (P16 to P14) 378
Figure B.3 Port 1 Block Diagram (P12, P11) 379
Figure B.4 Port 1 Block Diagram (P10) 380
Figure B.5 Port 2 Block Diagram (P22) 381
Figure B.6 Port 2 Block Diagram (P21) 382
Figure B.7 Port 2 Block Diagram (P20) 383
Figure B.8 Port 5 Block Diagram (P57, P56) 384
Figure B.9 Port 5 Block Diagram (P55) 385
Figure B.10 Port 5 Block Diagram (P54 to P50) 386
Figure B.11 Port 7 Block Diagram (P76) 387
Figure B.12 Port 7 Block Diagram (P75) 388
Figure B.13 Port 7 Block Diagram (P74) 389
Figure B.14 Port 8 Block Diagram (P87 to P85) 390
Figure B.15 Port 8 Block Diagram (P84 to P81) 391
Figure B.16 Port 8 Block Diagram (P80) 392
Figure B.17 Port B Block Diagram (PB7 to PB0) 393

Appendix D Package Dimensions
Figure D.1 FP-64E Package Dimensions 397
Figure D.2 FP-64A Package Dimensions 398
Figure D.3 FP-48F Package Dimensions 399
Figure D.4 FP-48B Package Dimensions 400
Figure D.5 DP-42S Package Dimensions 400

Appendix E EEPROM Stacked-Structure Cross-Sectional View
Figure E.1 EEPROM Stacked-Structure Cross-Sectional View 401

Tables

Section 1 Overview
Table 1.1 Pin Functions 9

Section 2 CPU
Table 2.1 Operation Notation 24
Table 2.2 Data Transfer Instructions 25
Table 2.3 Arithmetic Operations Instructions (1) 26
Table 2.3 Arithmetic Operations Instructions (2) 27
Table 2.4 Logic Operations Instructions 28
Table 2.5 Shift Instructions 28
Table 2.6 Bit Manipulation Instructions (1) 29
Table 2.6 Bit Manipulation Instructions (2) 30
Table 2.7 Branch Instructions 31
Table 2.8 System Control Instructions 32
Table 2.9 Block Data Transfer Instructions 33
Table 2.10 Addressing Modes 35
Table 2.11 Absolute Address Access Ranges 37
Table 2.12 Effective Address Calculation (1) 39
Table 2.12 Effective Address Calculation (2) 40

Section 3 Exception Handling
Table 3.1 Exception Sources and Vector Address 52
Table 3.2 Interrupt Wait States 62

Section 4 Address Break
Table 4.1 Access and Data Bus Used 69

Section 5 Clock Pulse Generators
Table 5.1 Crystal Resonator Parameters 79

Section 6 Power-Down Modes
Table 6.1 Operating Frequency and Waiting Time 85
Table 6.2 Transition Mode after SLEEP Instruction Execution and Interrupt Handling 89
Table 6.3 Internal State in Each Operating Mode 90

Section 7 ROM
Table 7.1 Setting Programming Modes 100
Table 7.2 Boot Mode Operation 102
Table 7.3 System Clock Frequencies for which Automatic Adjustment of LSI Bit Rate is Possible 103

Table 7.4 Reprogram Data Computation Table 106
Table 7.5 Additional-Program Data Computation Table 106
Table 7.6 Programming Time 106
Table 7.7 Flash Memory Operating States 111

Section 10 Timer A
Table 10.1 Pin Configuration 140

Section 11 Timer V
Table 11.1 Pin Configuration 147
Table 11.2 Clock Signals to Input to TCNTV and Counting Conditions 149

Section 12 Timer W
Table 12.1 Timer W Functions 160
Table 12.2 Pin Configuration 162

Section 14 Serial Communication Interface 3 (SCI3)
Table 14.1 Pin Configuration 198
Table 14.2 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (1) 206
Table 14.2 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (2) 207
Table 14.2 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (3) 208
Table 14.3 Maximum Bit Rate for Each Frequency (Asynchronous Mode) 208
Table 14.4 BRR Settings for Various Bit Rates (Clocked Synchronous Mode) 209
Table 14.5 SSR Status Flags and Receive Data Handling 215
Table 14.6 SCI3 Interrupt Requests 230

Section 15 I^2C Bus Interface (IIC)
Table 15.1 I^2C Bus Interface Pins 235
Table 15.2 Communication Format 239
Table 15.3 I^2C Transfer Rate 241
Table 15.4 Flags and Transfer States 249
Table 15.5 I^2C Bus Timing (SCL and SDA Output) 266
Table 15.6 Permissible SCL Rise Time (t_{sr}) Values 267
Table 15.7 I^2C Bus Timing (with Maximum Influence of t_{sr}/t_{sf}) 268

Section 16 A/D Converter
Table 16.1 Pin Configuration 277
Table 16.2 Analog Input Channels and Corresponding ADDR Registers 278
Table 16.3 A/D Conversion Time (Single Mode) 283

Section 17 EEPROM
Table 17.1 Pin Configuration 289
Table 17.2 Slave Addresses 292

Section 20 Electrical Characteristics
Table 20.1 Absolute Maximum Ratings 311
Table 20.2 DC Characteristics (1) 314
Table 20.2 DC Characteristics (2) 318
Table 20.2 DC Characteristics (3) 319
Table 20.3 AC Characteristics 320
Table 20.4 I^2C Bus Interface Timing 322
Table 20.5 Serial Interface (SCI3) Timing 323
Table 20.6 A/D Converter Characteristics 324
Table 20.7 Watchdog Timer Characteristics 325
Table 20.8 Flash Memory Characteristics 326
Table 20.9 EEPROM Characteristics 328
Table 20.10 DC Characteristics (1) 331
Table 20.10 DC Characteristics (2) 336
Table 20.11 AC Characteristics 337
Table 20.12 I^2C Bus Interface Timing 339
Table 20.13 Serial Interface (SCI3) Timing 340
Table 20.14 A/D Converter Characteristics 341
Table 20.15 Watchdog Timer Characteristics 342

Appendix A Instruction Set
Table A.1 Instruction Set 349
Table A.2 Operation Code Map (1) 362
Table A.2 Operation Code Map (2) 363
Table A.2 Operation Code Map (3) 364
Table A.3 Number of Cycles in Each Instruction 366
Table A.4 Number of Cycles in Each Instruction 367
Table A.5 Combinations of Instructions and Addressing Modes 376

Section 1 Overview

1.1 Features

- High-speed H8/300H central processing unit with an internal 16-bit architecture
  - — Upward-compatible with H8/300 CPU on an object level
  - — Sixteen 16-bit general registers
  - — 62 basic instructions
- Various peripheral functions
  - — Timer A (can be used as a time base for a clock)
  - — Timer V (8-bit timer)
  - — Timer W (16-bit timer)
  - — Watchdog timer
  - — SCI3 (Asynchronous or clocked synchronous serial communication interface)
  - — I^2C Bus Interface (conforms to the I^2C bus interface format that is advocated by Philips Electronics)
  - — 10-bit A/D converter
- On-chip memory

Product Classification		Model	EEPROM	ROM	RAM
Flash memory version	H8/3664N	HD64N3664	512 bytes	32 kbytes	2,048 bytes
(F-ZTAT™ version)	H8/3664F	HD64F3664	—	32 kbytes	2,048 bytes
Mask ROM version	H8/3664	HD6433664	—	32 kbytes	1,024 bytes
	H8/3663	HD6433663	—	24 kbytes	1,024 bytes
	H8/3662	HD6433662	—	16 kbytes	512 bytes
	H8/3661	HD6433661	—	12 kbytes	512 bytes
	H8/3660	HD6433660	—	8 kbytes	512 bytes

- General I/O ports
  - — I/O pins: 29 I/O pins (H8/3664N has 27 I/O pins), including 8 large current ports (I_{OL} = 20 mA, @V_{OL} = 1.5 V)
  - — Input-only pins: 8 input pins (also used for analog input)
- EEPROM interface (only for H8/3664N)
  - — I^2C Bus Interface (conforms to the I^2C bus interface format that is advocated by Philips Electronics)

- Supports various power-down modes

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

- Compact package

Package	Code	Body Size	Pin Pitch
LQFP-64	FP-64E	10.0 × 10.0 mm	0.5 mm
QFP-64	FP-64A	14.0 × 14.0 mm	0.8 mm
LQFP-48	FP-48F	10.0 × 10.0 mm	0.65 mm
LQFP-48	FP-48B	7.0 × 7.0 mm	0.5 mm
SDIP-42	DP-42S	14.0 × 37.3 mm	1.78 mm

Only LQFP-64 (FP-64E) for H8/3664N package

1.2 Internal Block Diagram

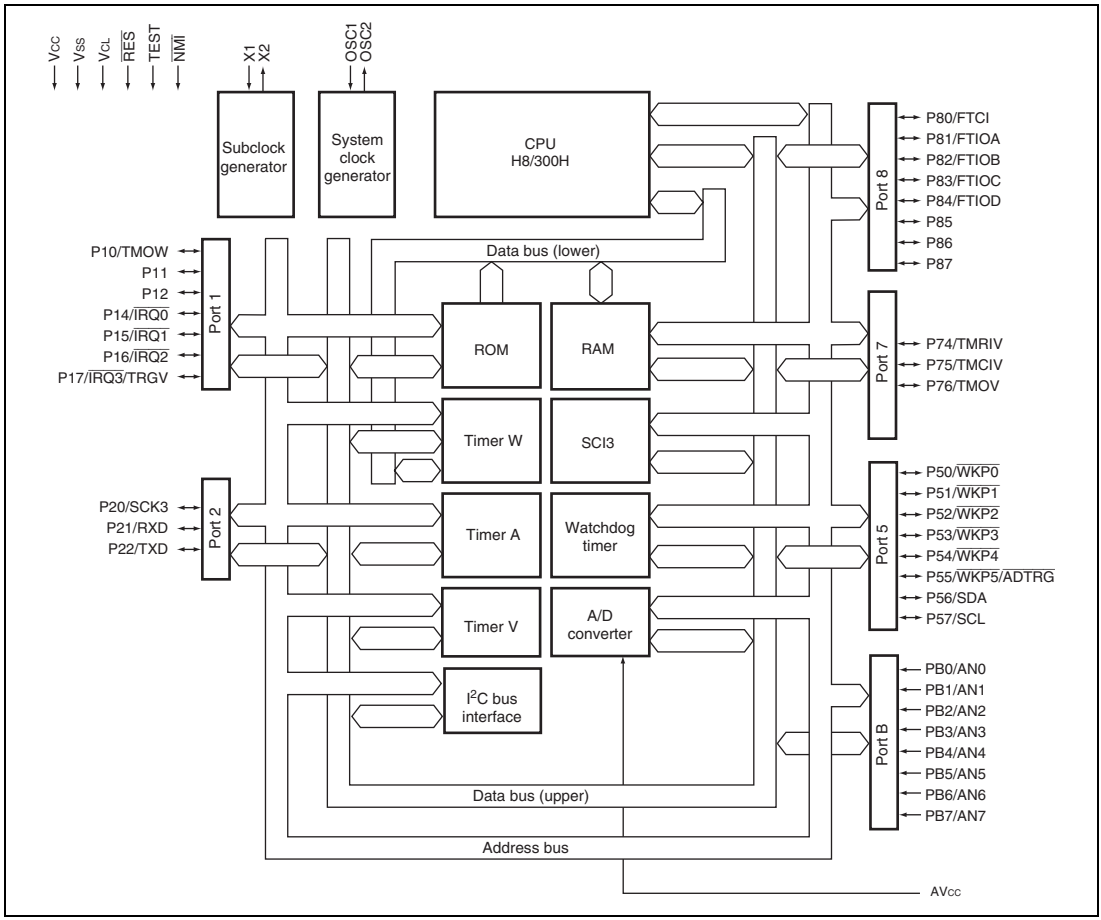
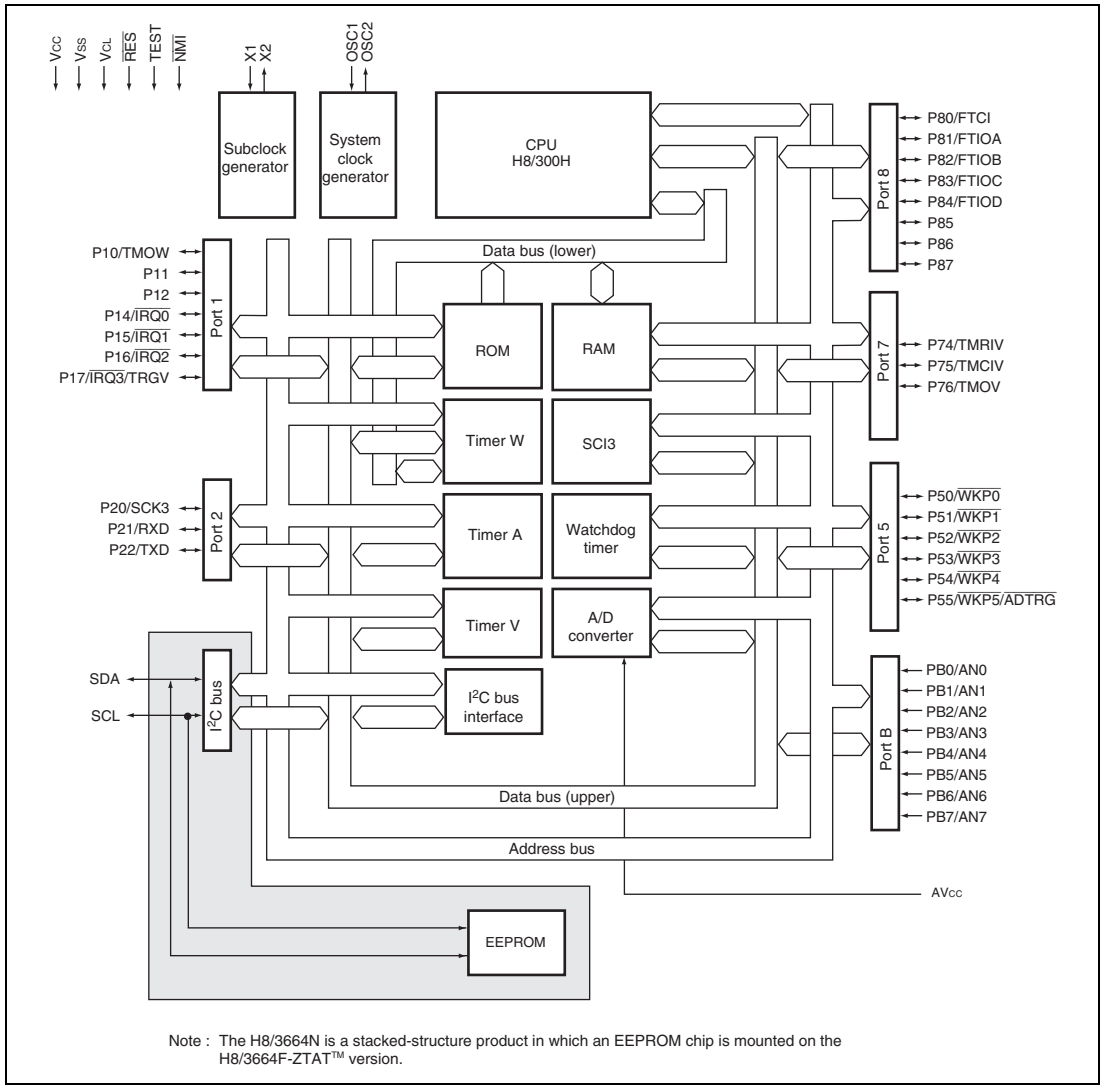


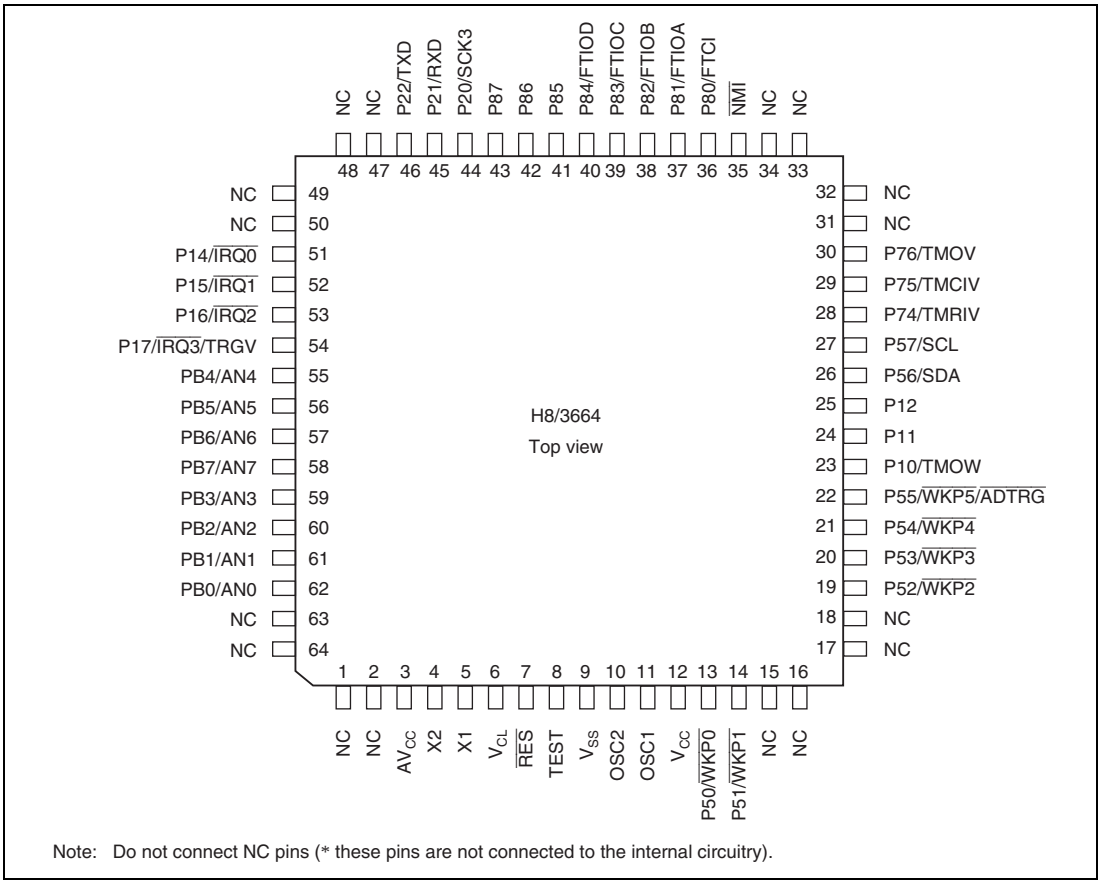
Figure 1.1 Internal Block Diagram of H8/3664 of F-ZTAT™ and Mask-ROM Versions



Note : The H8/3664N is a stacked-structure product in which an EEPROM chip is mounted on the H8/3664F-ZTAT™ version.

Figure 1.2 Internal Block Diagram of H8/3664N of F-ZTAT™ Version with EEPROM

1.3 Pin Arrangement



Note: Do not connect NC pins (* these pins are not connected to the internal circuitry).

Figure 1.3 Pin Arrangement of H8/3664 of F-ZTAT™ and Mask-ROM Versions (FP-64E, FP-64A)

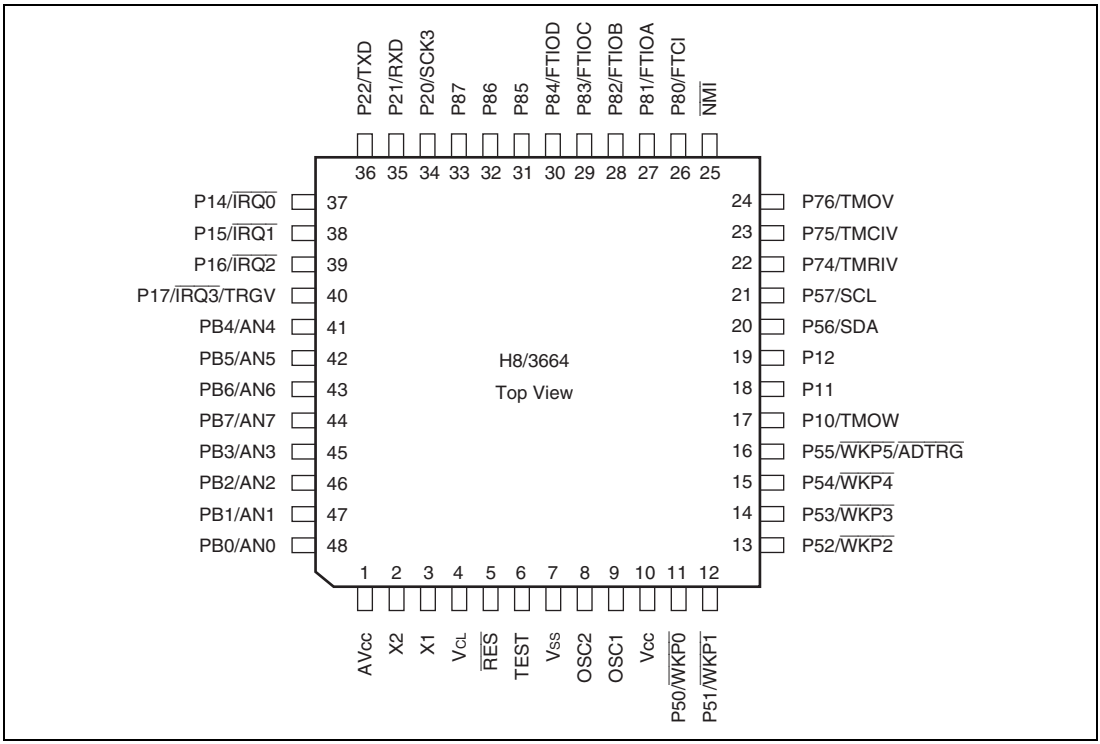
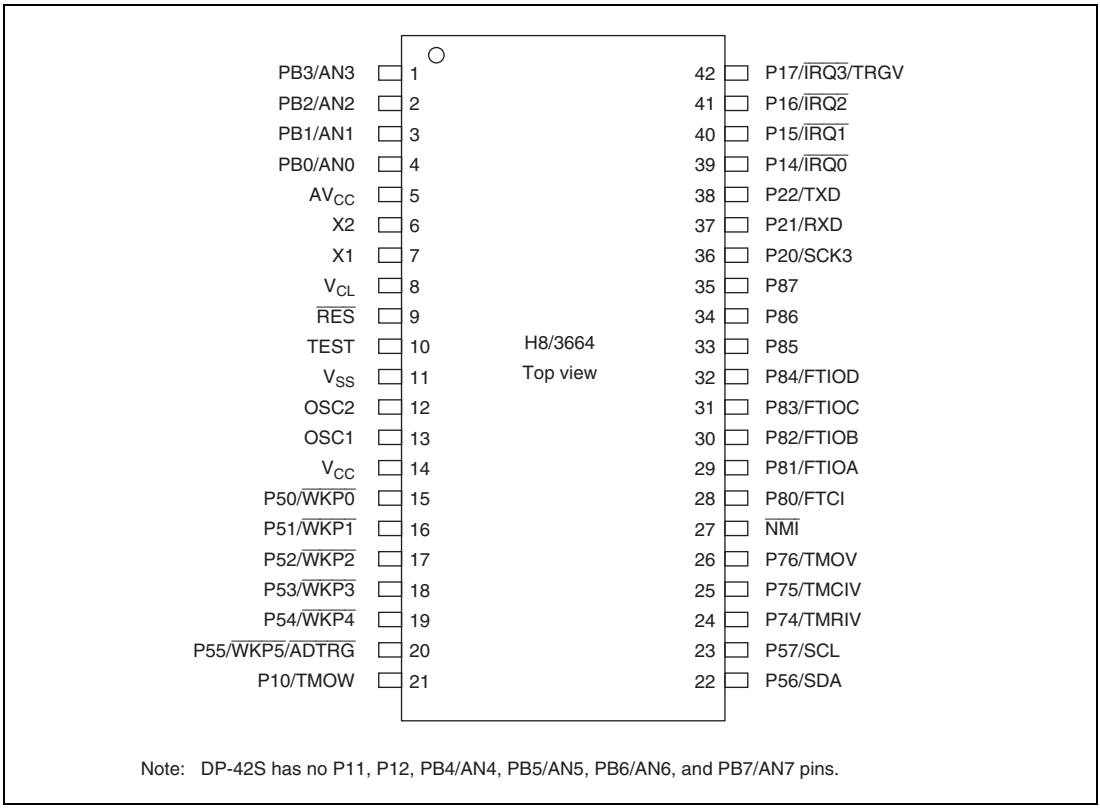
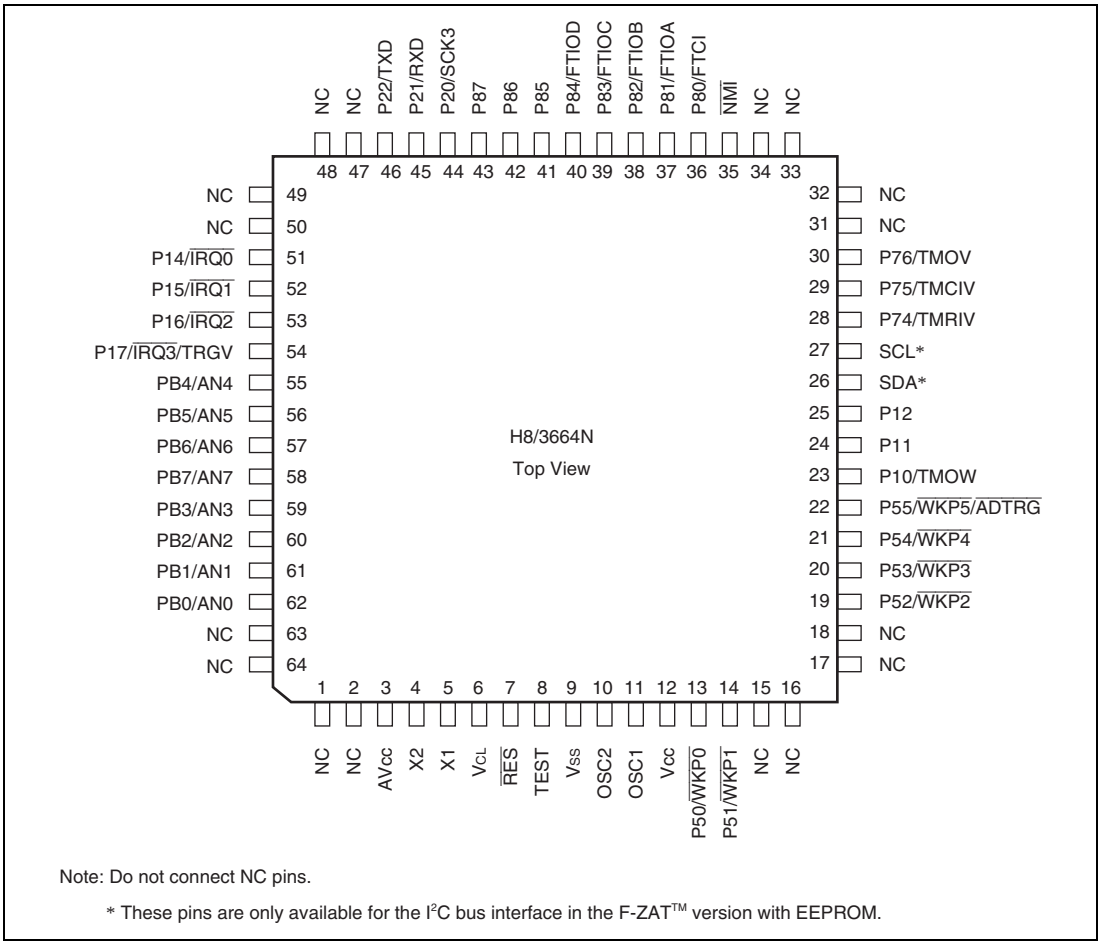


Figure 1.4 Pin Arrangement of H8/3664 of F-ZTAT™ and Mask-ROM Versions (FP-48F, FP-48B)



Note: DP-42S has no P11, P12, PB4/AN4, PB5/AN5, PB6/AN6, and PB7/AN7 pins.

Figure 1.5 Pin Arrangement of H8/3664 of F-ZTAT™ and Mask-ROM Versions (DS-42S)



Note: Do not connect NC pins.

* These pins are only available for the I²C bus interface in the F-ZAT™ version with EEPROM.

Figure 1.6 Pin Arrangement of H8/3664N of F-ZTAT™ Version with EEPROM (FP-64E)

1.4 Pin Functions

Table 1.1 Pin Functions

		Pin No.					
		H8/3664			H8/3664N		
Type	Symbol	FP-64E, FP-64A	FP-48F, FP-48B	DP-42S	FP-64E	I/O	Functions
Power source pins	V_{CC}	12	10	14	12	Input	Power supply pin. Connect this pin to the system power supply.
	V_{SS}	9	7	11	9	Input	Ground pin. Connect all these pins to the system power supply (0V).
	AV_{CC}	3	1	5	3	Input	Analog power supply pin for the A/D converter. When the A/D converter is not used, connect this pin to the system power supply.
	V_{CL}	6	4	8	6	Input	Internal step-down power supply pin. Connect a capacitor of around 0.1 μF between this pin and the Vss pin for stabilization.
Clock pins	OSC1	11	9	13	11	Input	These pins connect to a crystal or ceramic resonator for system clocks, or can be used to input an external clock. These pins can be used to input an external clock. See section 5, Clock Pulse Generators, for a typical connection.
	OSC2	10	8	12	10	Output	
	X1	5	3	7	5	Input	For connection to a 32.768 kHz crystal resonator for subclocks. See section 5, Clock Pulse Generators, for a typical connection.
	X2	4	2	6	4	Output	
System control	$\overline{RES}$	7	5	9	7	Input	Reset pin. When this driven low, the chip is reset.
	TEST	8	6	10	8	Input	Test pin. Connect this pin to Vss.

		Pin No.					
		H8/3664			H8/3664N		
Type	Symbol	FP-64E, FP-64A	FP-48F, FP-48B	DP-42S	FP-64E	I/O	Functions
Interrupt pins	$\overline{NMI}$	35	25	27	35	Input	Non-maskable interrupt request input pin. Be sure to pull-up by a pull-up resistor.
	$\overline{IRQ0}$ to $\overline{IRQ3}$	51 to 54	37 to 40	39 to 42	51 to 54	Input	External interrupt request input pins. Can select the rising or falling edge.
	$\overline{WKP0}$ to $\overline{WKP5}$	13, 14, 19 to 22	11 to 16	15 to 20	13, 14, 19 to 22	Input	External interrupt request input pins. Can select the rising or falling edge.
Timer A	TMOW	23	17	21	23	Output	This is an output pin for divided clocks
Timer V	TMOV	30	24	26	30	Output	This is an output pin for waveforms generated by the output compare function
	TMCIV	29	23	25	29	Input	External event input pin
	TMRIV	28	22	24	28	Input	Counter reset input pin
	TRGV	54	40	42	54	Input	Counter start trigger input pin
Timer W	FTCI	36	26	28	36	Input	External event input pin
	FTIOA to FTIOD	37 to 40	27 to 30	29 to 32	37 to 40	I/O	Output compare output/ input capture input/ PWM output pin
I^2C bus inerface	SDA	26*[2]	20	22	26*[1]	I/O	IIC data I/O pin. Can directly drive a bus by NMOS open-drain output. When using this pin, external pull-up resistance is required.
	SCL	27*[2]	21	23	27*[1]	I/O (EEPROM: input)	IIC clock I/O pin. Can directly drive a bus by NMOS open-drain output. When using this pin, external pull-up resistance is required.
Serial communication interface (SCI)	TXD	46	36	38	46	Output	Transmit data output pin
	RXD	45	35	37	45	Input	Receive data input pin
	SCK3	44	34	36	44	I/O	Clock I/O pin
A/D converter	AN7 to AN0	55 to 62	41 to 48	1 to 4*[2]	55 to 62	Input	Analog input pin
	$\overline{ADTRG}$	22	16	20	22	Input	A/D converter trigger input pin

		Pin No.					
		H8/3664			H8/3664N		
Type	Symbol	FP-64E, FP-64A	FP-48F, FP-48B	DP-42S	FP-64E	I/O	Functions
I/O ports	PB7 to PB0	55 to 62	41 to 48	1 to 4*[2]	55 to 62	Input	8-bit input port
	P17 to P14, P12 to P10	51 to 54 23 to 25	37 to 40 17 to 19	39 to 42, 21*[2]	51 to 54, 23 to 25	I/O	7-bit I/O port
	P22 to P20	44 to 46	34 to 36	36 to 38	44 to 46	I/O	3-bit I/O port
	P57 to P50 (P55 to P50 for H8/3664N)	13,14, 19 to 22 26, 27	21, 20, 16 to 11	15 to 20, 22, 23	13, 14, 19 to 22	I/O	8-bit I/O port (6-bit I/O port for H8/3664N)
	P76 to P74	28 to 30	22 to 24	24 to 26	28 to 30	I/O	3-bit I/O port
	P87 to P80	36 to 43	26 to 33	28 to 35	36 to 43	I/O	8-bit I/O port

Notes: 1. These pins are only available for the I^2C bus interface in the F-ZAT™ version with EEPROM. Since the I^2C bus is disabled after canceling a reset, the ICE bit in ICCR must be set to 1 by using the program.

2. The DP-42S does not have the P11, P12, PB4/AN4, PB5/AN5, PB6/AN6, and PB7/AN7 pins.

Section 2 CPU

This LSI has an H8/300H CPU with an internal 32-bit architecture that is upward-compatible with the H8/300CPU, and supports only normal mode, which has a 64-kbyte address space.

- Upward-compatible with H8/300 CPUs
  - Can execute H8/300 CPUs object programs
  - Additional eight 16-bit extended registers
  - 32-bit transfer and arithmetic and logic instructions are added
  - Signed multiply and divide instructions are added.
- General-register architecture
  - Sixteen 16-bit general registers also usable as sixteen 8-bit registers or eight 32-bit registers
- Sixty-two basic instructions
  - 8/16/32-bit data transfer and 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:24,ERn)]
  - Register indirect with post-increment or pre-decrement [@ERn+ or @–ERn]
  - Absolute address [@aa:8, @aa:16, @aa:24]
  - Immediate [#xx:8, #xx:16, or #xx:32]
  - Program-counter relative [@(d:8,PC) or @(d:16,PC)]
  - Memory indirect [@@aa:8]
- 64-kbyte address space
- High-speed operation
  - All frequently-used instructions execute in one or two states
  - 8/16/32-bit register-register add/subtract : 2 state
  - 8 × 8-bit register-register multiply : 14 states
  - 16 ÷ 8-bit register-register divide : 14 states
  - 16 × 16-bit register-register multiply : 22 states
  - 32 ÷ 16-bit register-register divide : 22 states
- Power-down state
  - Transition to power-down state by SLEEP instruction

2.1 Address Space and Memory Map

The address space of this LSI is 64 kbytes, which includes the program area and the data area. Figures 2.1 show the memory map.

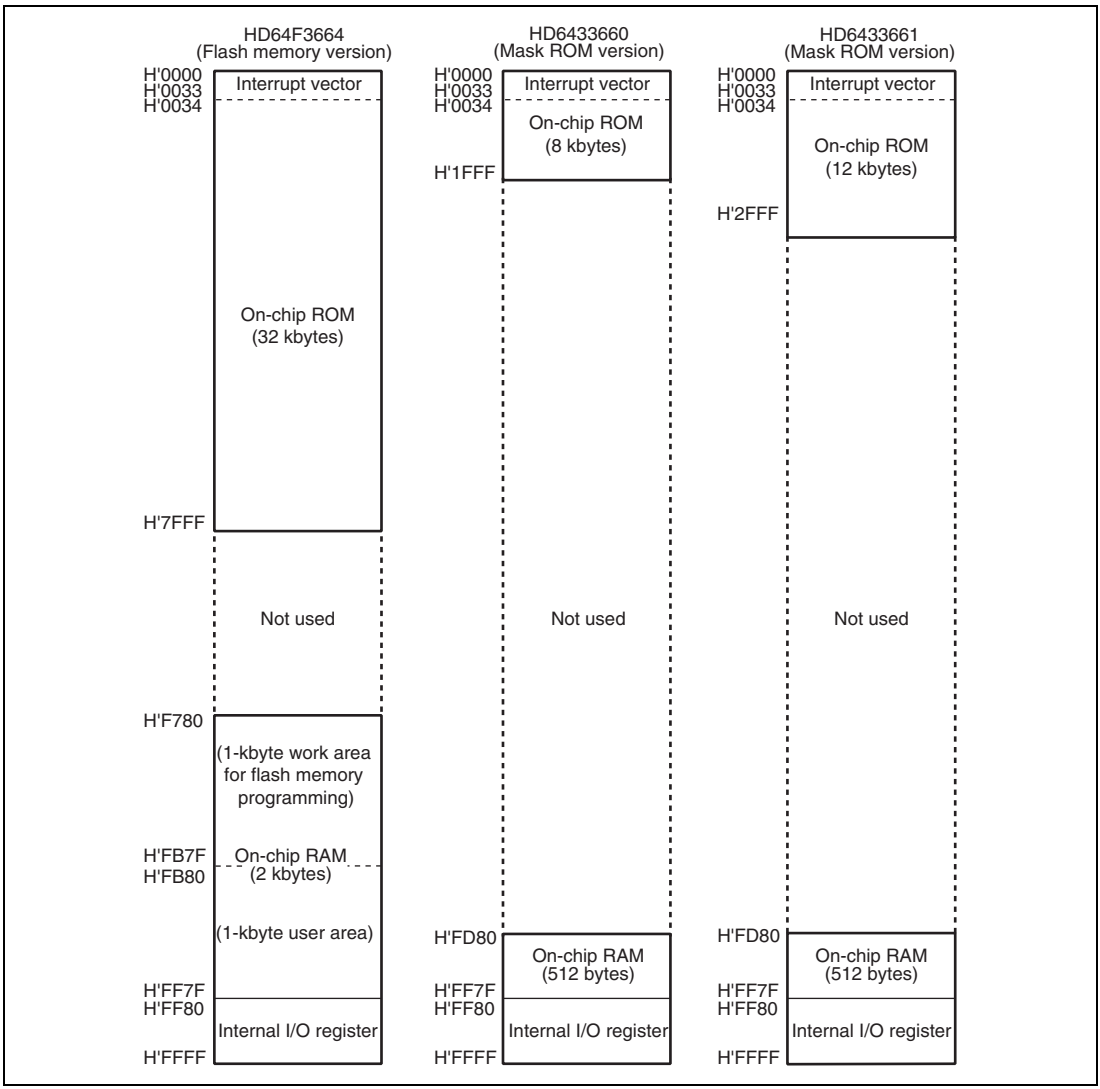


Figure 2.1 Memory Map (1)

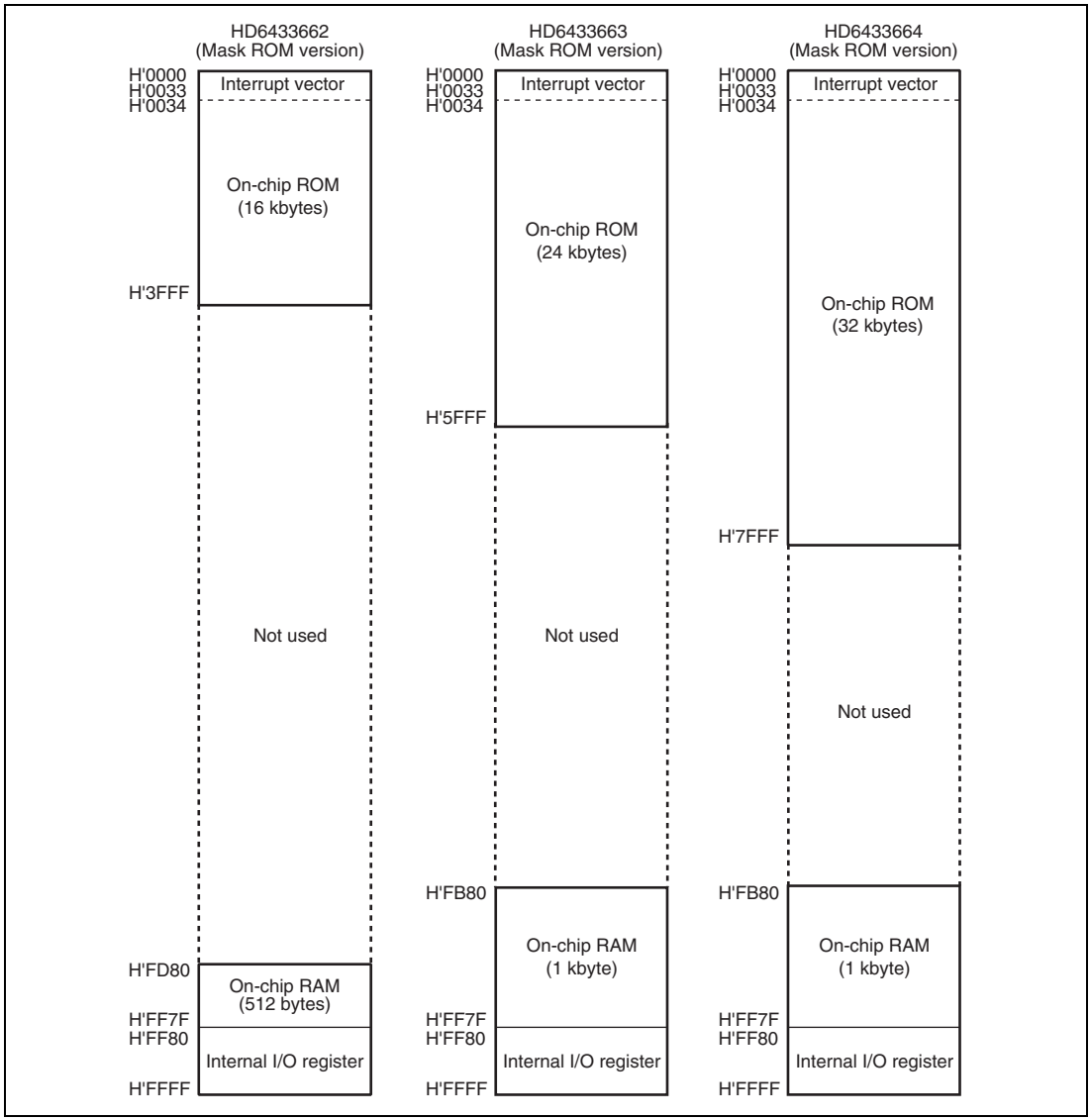


Figure 2.1 Memory Map (2)

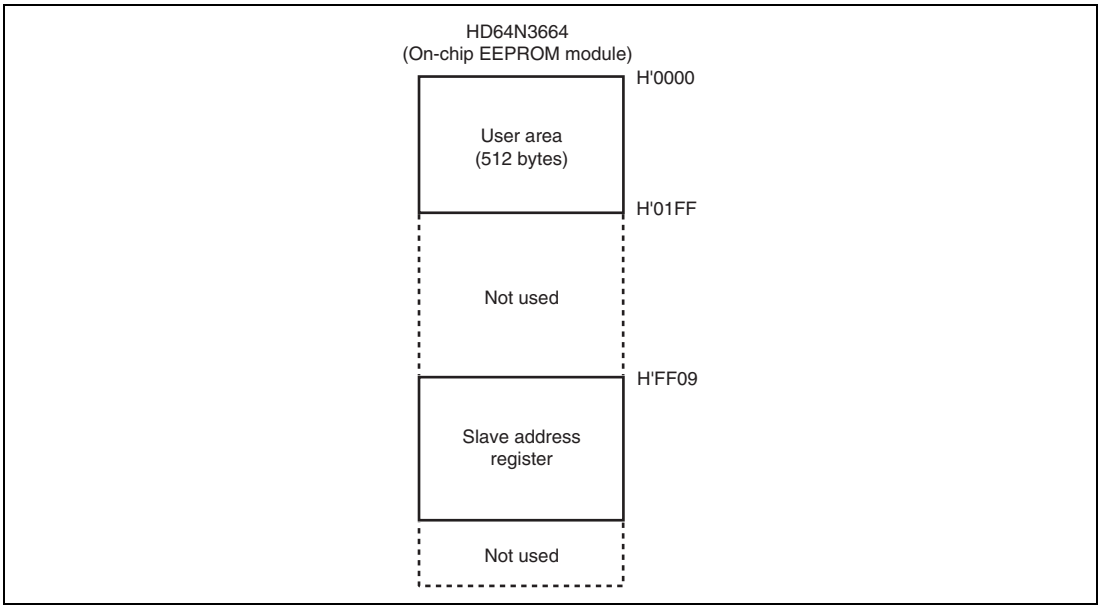


Figure 2.1 Memory Map (3)

2.2 Register Configuration

The H8/300H CPU has the internal registers shown in figure 2.2. There are two types of registers; general registers and control registers. The control registers are a 24-bit program counter (PC), and an 8-bit condition code register (CCR).

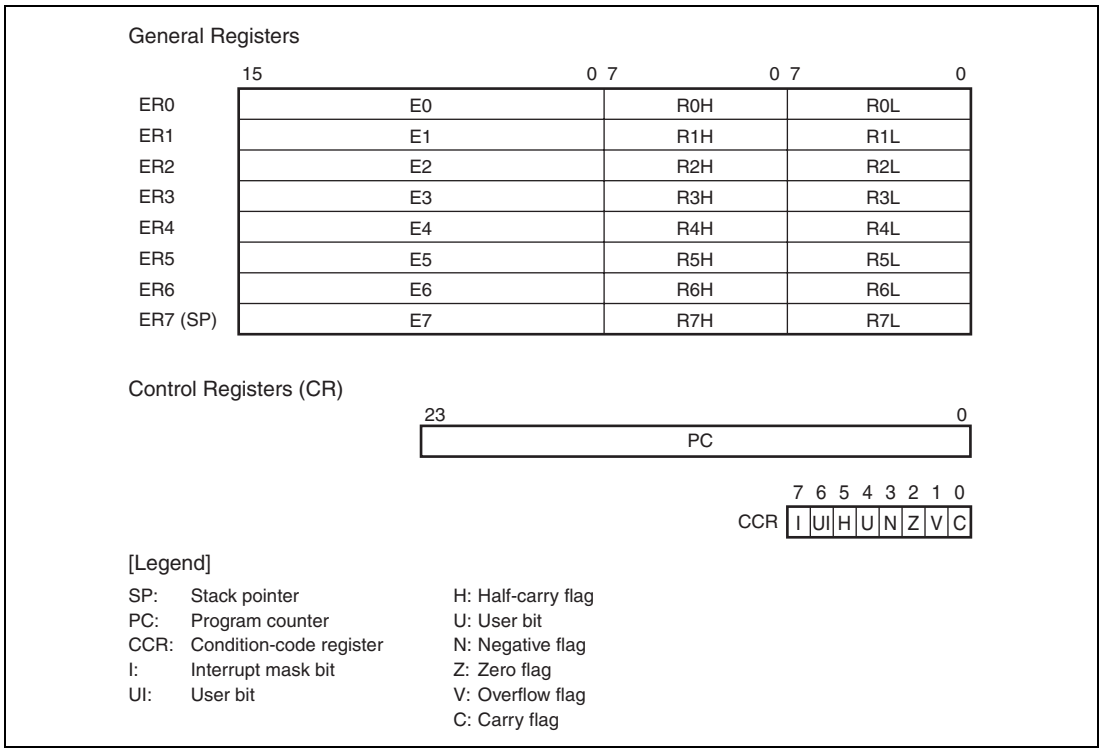


Figure 2.2 CPU Registers

2.2.1 General Registers

The H8/300H CPU has eight 32-bit general registers. These general registers are all functionally identical and can be used as both address registers and data registers. When a general register is used as a data register, it can be accessed as a 32-bit, 16-bit, or 8-bit register. Figure 2.3 illustrates the usage of the general registers. When the general registers are used as 32-bit registers or address registers, they are designated by the letters ER (ER0 to ER7).

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

The R registers divide into 8-bit registers designated by the letters RH (R0H to R7H) and RL (R0L to R7L). These registers are functionally equivalent, providing a maximum of sixteen 8-bit registers.

The usage of each register can be selected independently.

General register ER7 has the function of stack pointer (SP) in addition to its general-register function, and is used implicitly in exception handling and subroutine calls. Figure 2.4 shows the stack.

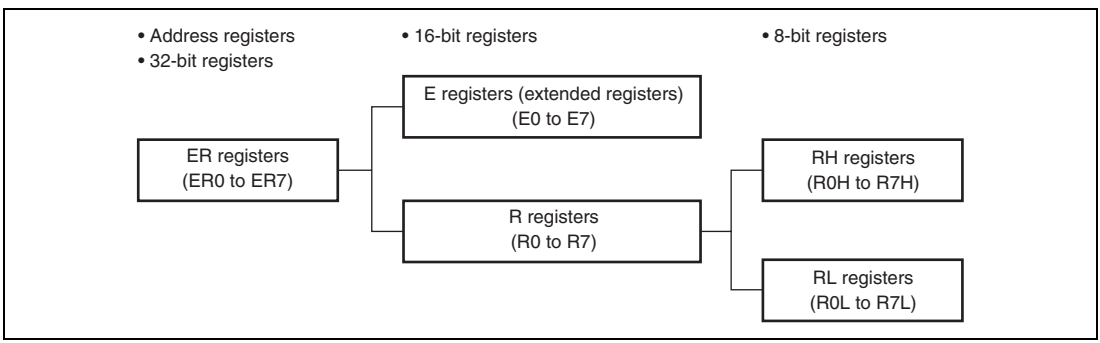


Figure 2.3 Usage of General Registers

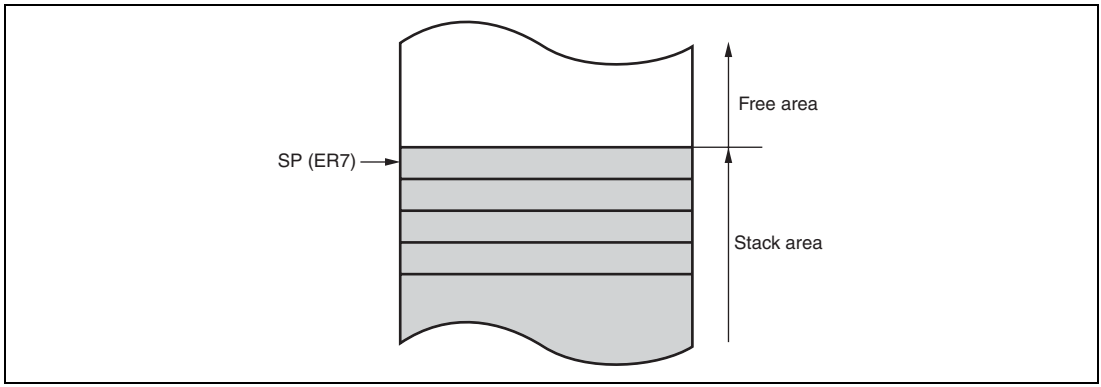


Figure 2.4 Relationship between Stack Pointer and Stack Area

2.2.2 Program Counter (PC)

This 24-bit counter indicates the address of the next instruction the CPU will execute. The length of all CPU instructions is 2 bytes (one word), so the least significant PC bit is ignored. (When an instruction is fetched, the least significant PC bit is regarded as 0). The PC is initialized when the start address is loaded by the vector address generated during reset exception-handling sequence.

2.2.3 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. The I bit is initialized to 1 by reset exception-handling sequence, but other bits are not initialized.

Some instructions leave flag bits unchanged. 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.

For the action of each instruction on the flag bits, see appendix A.1, Instruction List.

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.
6	UI	Undefined	R/W	User Bit Can be written and read 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 and read by software using the LDC, STC, ANDC, ORC, and XORC instructions.
3	N	Undefined	R/W	Negative Flag Stores the value of the most significant bit of data as a sign bit.
2	Z	Undefined	R/W	Zero Flag Set to 1 to indicate zero data, and cleared to 0 to indicate non-zero data.
1	V	Undefined	R/W	Overflow Flag Set to 1 when an arithmetic overflow occurs, and cleared to 0 at other times.
0	C	Undefined	R/W	Carry Flag Set to 1 when a carry occurs, and cleared to 0 otherwise. Used by: • Add instructions, to indicate a carry • Subtract instructions, to indicate a borrow • Shift and rotate instructions, to indicate a carry The carry flag is also used as a bit accumulator by bit manipulation instructions.

2.3 Data Formats

The H8/300H 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.

2.3.1 General Register Data Formats

Figure 2.5 shows the data formats in general registers.

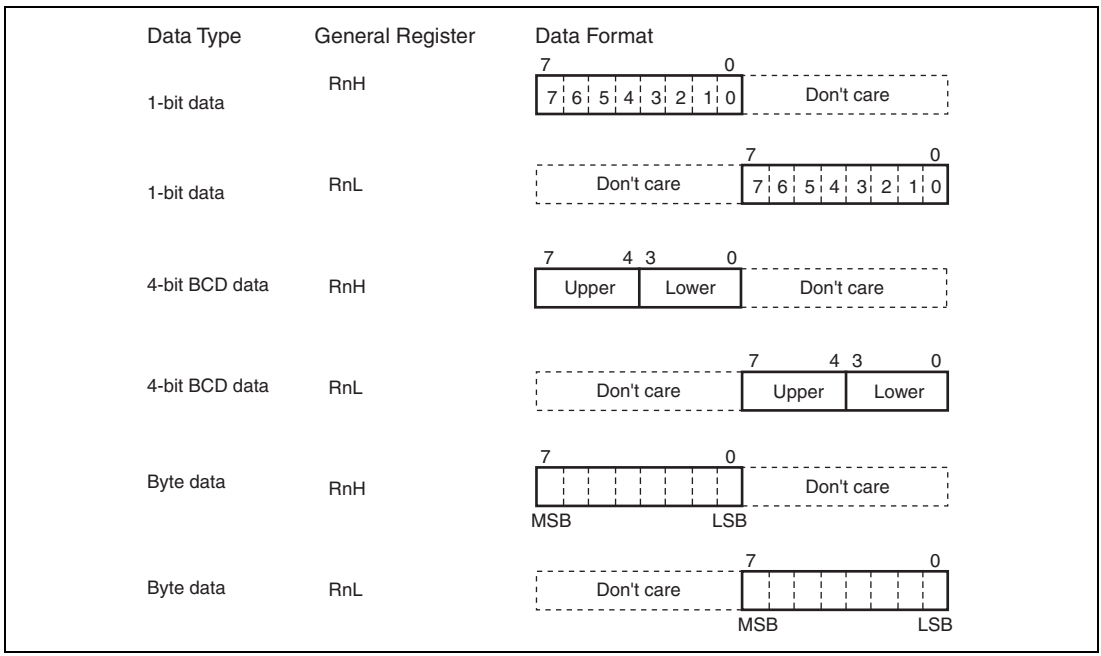


Figure 2.5 General Register Data Formats (1)

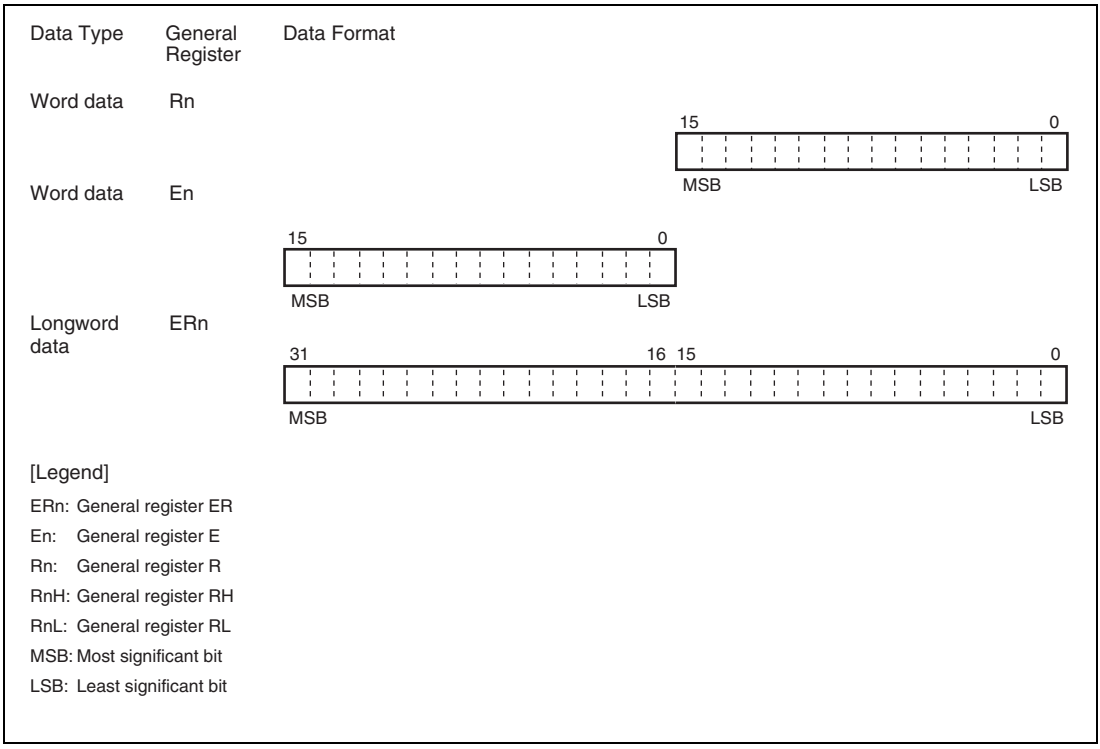


Figure 2.5 General Register Data Formats (2)

2.3.2 Memory Data Formats

Figure 2.6 shows the data formats in memory. The H8/300H CPU can access word data and longword data in memory, however word or longword data must begin at an even address. If an attempt is made to access word or longword data at an odd address, an address error does not occur, however the least significant bit of the address is regarded as 0, so access begins the preceding address. This also applies to instruction fetches.

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

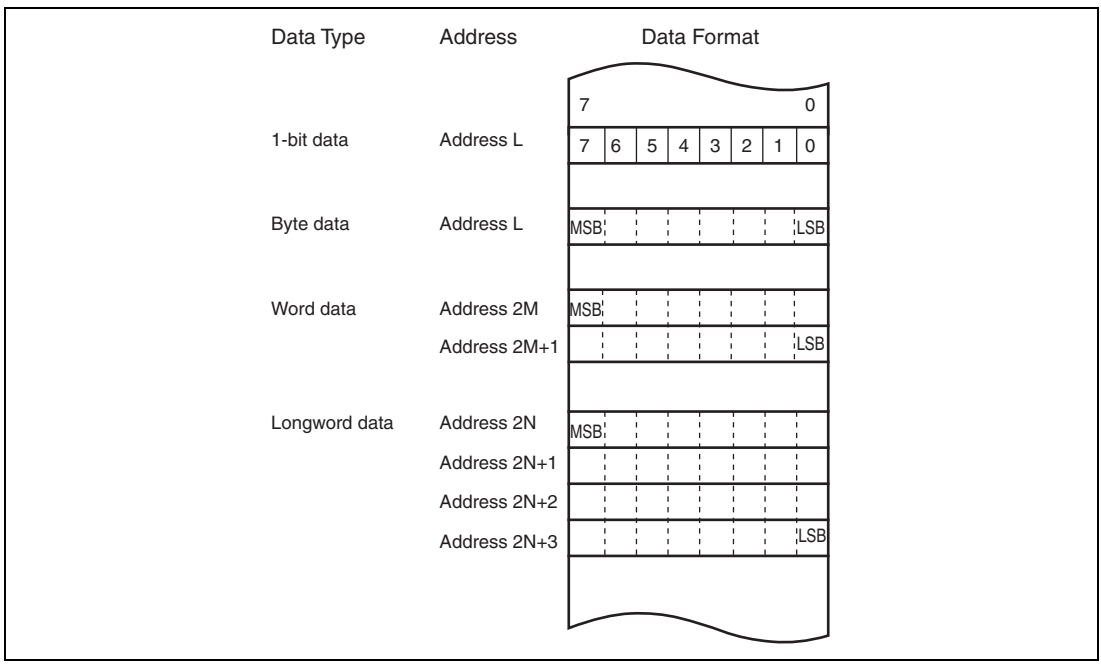


Figure 2.6 Memory Data Formats

2.4 Instruction Set

2.4.1 Table of Instructions Classified by Function

The H8/300H CPU has 62 instructions. Tables 2.2 to 2.9 summarize the instructions in each functional category. The notation used in tables 2.2 to 2.9 is defined below.

Table 2.1 Operation Notation

Symbol	Description
Rd	General register (destination)*
Rs	General register (source)*
Rn	General register*
ERn	General register (32-bit register or address register)
(EAd)	Destination operand
(EAs)	Source operand
CCR	Condition-code register
N	N (negative) flag in CCR
Z	Z (zero) flag in CCR
V	V (overflow) flag in CCR
C	C (carry) flag in CCR
PC	Program counter
SP	Stack pointer
#IMM	Immediate data
disp	Displacement
+	Addition
–	Subtraction
×	Multiplication
÷	Division
∧	Logical AND
∨	Logical OR
⊕	Logical XOR
→	Move
¬	NOT (logical complement)
:3/:8/:16/:24	3-, 8-, 16-, or 24-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/address register (ER0 to ER7).

Table 2.2 Data Transfer Instructions

Instruction	Size*	Function
MOV	B/W/L	(EAs) → Rd, Rs → (EAd) Moves data between two general registers or between a general register and memory, or moves immediate data to a general register.
MOVFPE	B	(EAs) → Rd, Cannot be used in this LSI.
MOVTPE	B	Rs → (EAs) 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.

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

Table 2.3 Arithmetic Operations Instructions (1)

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

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

Table 2.3 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 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.

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

Table 2.4 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 of general register contents.

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

Table 2.5 Shift Instructions

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

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

Table 2.6 Bit Manipulation Instructions (1)

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

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

Table 2.6 Bit Manipulation Instructions (2)

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

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

Table 2.7 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		

Note: * Bcc is the general name for conditional branch instructions.

Table 2.8 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 Moves the source operand contents to the CCR. The CCR size is one byte, but in transfer from memory, data is read by word access.
STC	B/W	CCR → (EAd), EXR → (EAd) Transfers the CCR contents to a destination location. The condition code register size is one byte, but in transfer to memory, data is written by word access.
ANDC	B	CCR ∧ #IMM → CCR, EXR ∧ #IMM → EXR Logically ANDs the CCR with immediate data.
ORC	B	CCR ∨ #IMM → CCR, EXR ∨ #IMM → EXR Logically ORs the CCR with immediate data.
XORC	B	CCR ⊕ #IMM → CCR, EXR ⊕ #IMM → EXR Logically XORs the CCR with immediate data.
NOP	—	PC + 2 → PC Only increments the program counter.

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

Table 2.9 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.

2.4.2 Basic Instruction Formats

H8/300H CPU instructions consist of 2-byte (1-word) units. An instruction consists of an operation field (op field), a register field (r field), an effective address extension (EA field), and a condition field (cc).

Figure 2.7 shows examples of instruction formats.

- Operation Field

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

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

  8, 16, or 32 bits specifying immediate data, an absolute address, or a displacement. A24-bit address or displacement is treated as a 32-bit data in which the first 8 bits are 0 (H'00).
- Condition Field

  Specifies the branching condition of Bcc instructions.

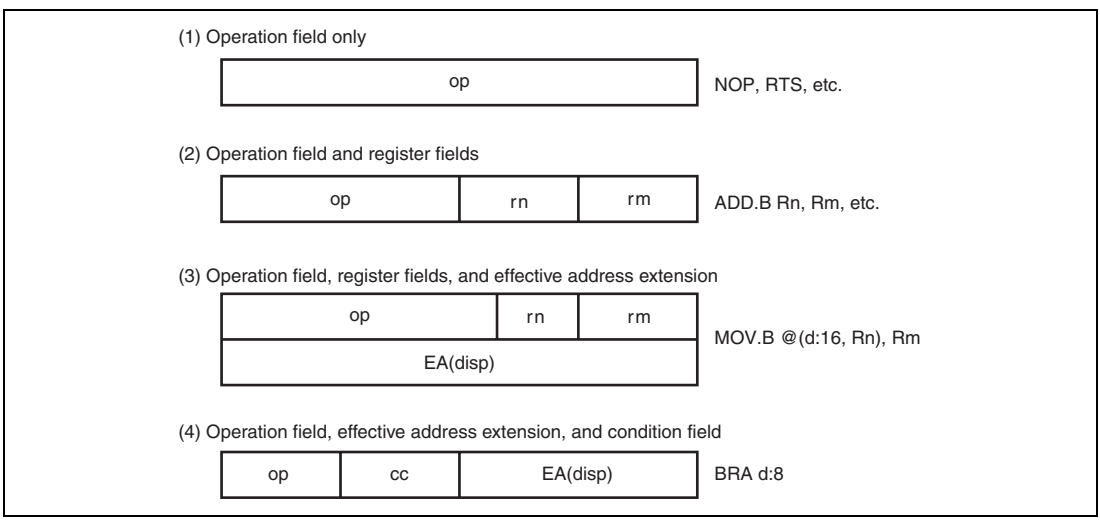


Figure 2.7 Instruction Formats

2.5 Addressing Modes and Effective Address Calculation

The following describes the H8/300H CPU. In this LSI, the upper eight bits are ignored in the generated 24-bit address, so the effective address is 16 bits.

2.5.1 Addressing Modes

The H8/300H CPU supports the eight addressing modes listed in table 2.10. Each instruction uses a subset of these addressing modes. Addressing modes that can be used differ depending on the instruction. For details, refer to appendix A.4, Combinations of Instructions and Addressing Modes.

Arithmetic and logic instructions can use the register direct and immediate modes. Data transfer instructions can use all addressing modes except program-counter relative and memory indirect. Bit manipulation instructions use register direct, register indirect, or the absolute addressing mode to specify an operand, and register direct (BSET, BCLR, BNOT, and BTST instructions) or immediate (3-bit) addressing mode to specify a bit number in the operand.

Table 2.10 Addressing Modes

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

(1) Register Direct—Rn

The register field of the instruction specifies an 8-, 16-, or 32-bit general register containing the operand. R0H to R7H and R0L to R7L can be specified as 8-bit registers. R0 to R7 and E0 to E7 can be specified as 16-bit registers. ER0 to ER7 can be specified as 32-bit registers.

(2) Register Indirect—@ERn

The register field of the instruction code specifies an address register (ERn), the lower 24 bits of which contain the address of the operand on memory.

(3) Register Indirect with Displacement—@(d:16, ERn) or @(d:24, ERn)

A 16-bit or 24-bit displacement contained in the instruction is added to an address register (ERn) specified by the register field of the instruction, and the lower 24 bits of the sum the address of a memory operand. A 16-bit displacement is sign-extended when added.

(4) Register Indirect with Post-Increment or Pre-Decrement—@ERn+ or @-ERn

(a) Register indirect with post-increment—@ERn+

The register field of the instruction code specifies an address register (ERn) the lower 24 bits of which contains the address of a memory operand. After the operand is accessed, 1, 2, or 4 is added to the address register contents (32 bits) and the sum is stored in the address register. The value added is 1 for byte access, 2 for word access, or 4 for longword access. For the word or longword access, the register value should be even.

(b) 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 lower 24 bits of the result is the address of a memory operand. The result is also stored in the address register. The value subtracted is 1 for byte access, 2 for word access, or 4 for longword access. For the word or longword access, the register value should be even.

(5) Absolute Address—@aa:8, @aa:16, @aa:24

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)

For an 8-bit absolute address, the upper 16 bits are all assumed to be 1 (H'FFFF). For a 16-bit absolute address the upper 8 bits are a sign extension. A 24-bit absolute address can access the entire address space.

The access ranges of absolute addresses for the group of this LSI are those shown in table 2.11, because the upper 8 bits are ignored.

Table 2.11 Absolute Address Access Ranges

Absolute Address	Access Range
8 bits (@aa:8)	H'FF00 to H'FFFF
16 bits (@aa:16)	H'0000 to H'FFFF
24 bits (@aa:24)	H'0000 to H'FFFF

(6) Immediate—#xx:8, #xx:16, or #xx:32

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

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

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

This mode is used in the BSR instruction. An 8-bit or 16-bit displacement contained in the instruction is sign-extended and added to the 24-bit PC contents to generate a branch address. The PC value to which the displacement is added is the address of the first byte of the next instruction, so the possible branching range is –126 to +128 bytes (–63 to +64 words) or –32766 to +32768 bytes (–16383 to +16384 words) from the branch instruction. The resulting value should be an even number.

(8) Memory Indirect—@@aa:8

This mode can be used by the JMP and JSR instructions. The instruction code contains an 8-bit absolute address specifying a memory operand. This memory operand contains a branch address. The memory operand is accessed by longword access. The first byte of the memory operand is ignored, generating a 24-bit branch address. Figure 2.8 shows how to specify branch address for in memory indirect mode. The upper bits of the absolute address are all assumed to be 0, so the address range is 0 to 255 (H'0000 to H'00FF).

Note that the first part of the address range is also the exception vector area.

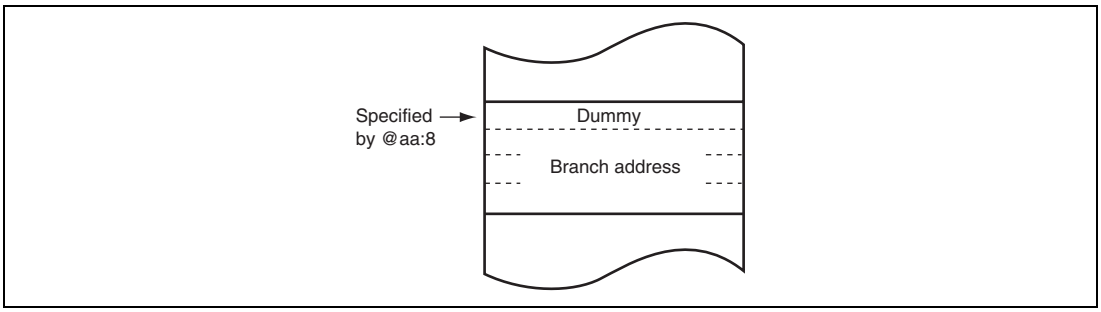


Figure 2.8 Branch Address Specification in Memory Indirect Mode

2.5.2 Effective Address Calculation

Table 2.12 indicates how effective addresses are calculated in each addressing mode. In this LSI the upper 8 bits of the effective address are ignored in order to generate a 16-bit effective address.

Table 2.12 Effective Address Calculation (1)

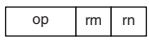
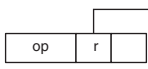
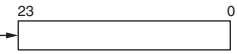
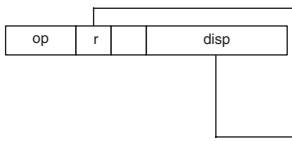
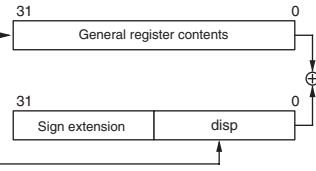
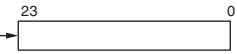
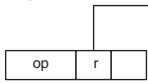
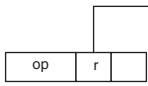
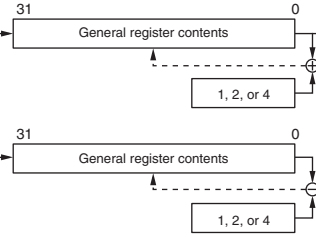
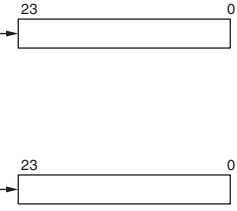
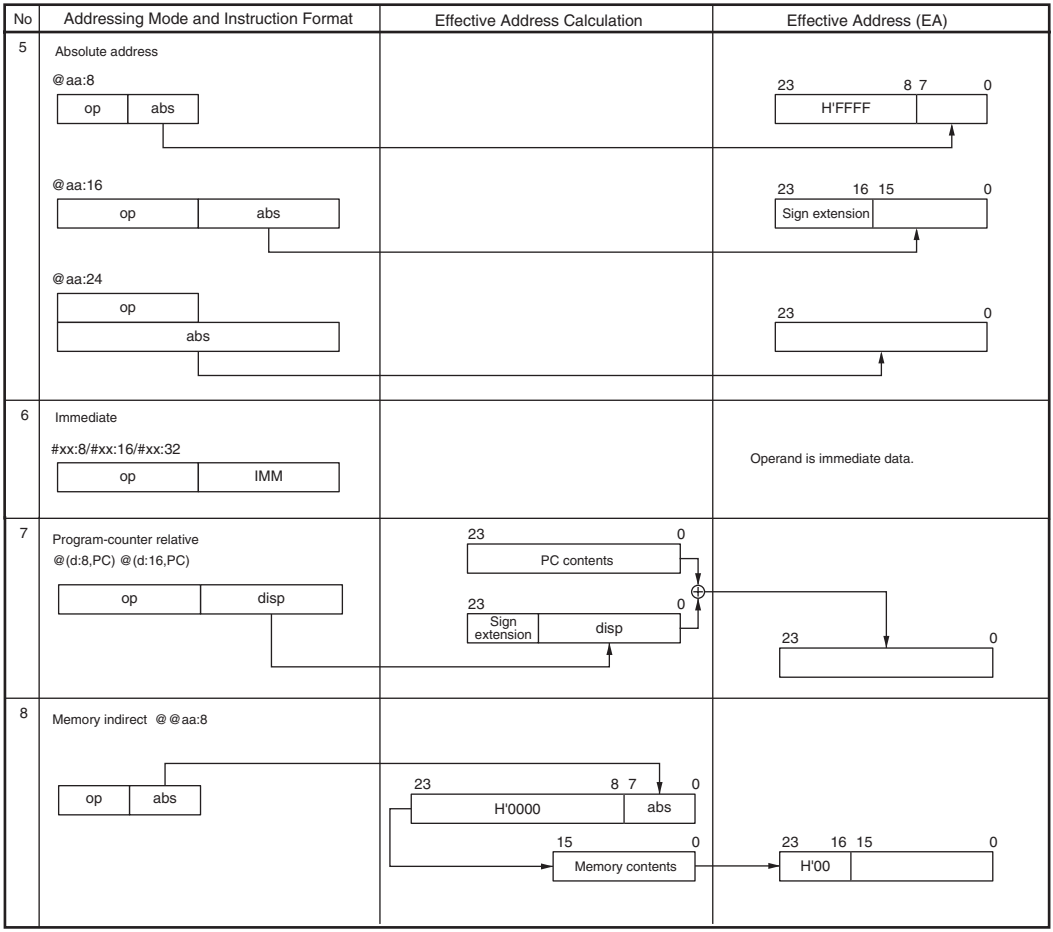
No	Addressing Mode and Instruction Format	Effective Address Calculation	Effective Address (EA)
1	Register direct(Rn) 		Operand is general register contents.
2	Register indirect(@ERn) 		
3	Register indirect with displacement @(d:16,ERn) or @(d:24,ERn) 		
4	Register indirect with post-increment or pre-decrement •Register indirect with post-increment @ERn+  •Register indirect with pre-decrement @-ERn 	 The value to be added or subtracted is 1 when the operand is byte size, 2 for word size, and 4 for longword size.	

Table 2.12 Effective Address Calculation (2)



[Legend]
r, rm,rn : Register field
op : Operation field
disp : Displacement
IMM : Immediate data
abs : Absolute address

2.6 Basic Bus Cycle

CPU operation is synchronized by a system clock (ϕ) or a subclock (ϕ_{SUB}). The period from a rising edge of ϕ or ϕ_{SUB} to the next rising edge is called one state. A bus cycle consists of two states or three states. The cycle differs depending on whether access is to on-chip memory or to on-chip peripheral modules.

2.6.1 Access to On-Chip Memory (RAM, ROM)

Access to on-chip memory takes place in two states. The data bus width is 16 bits, allowing access in byte or word size. Figure 2.9 shows the on-chip memory access cycle.

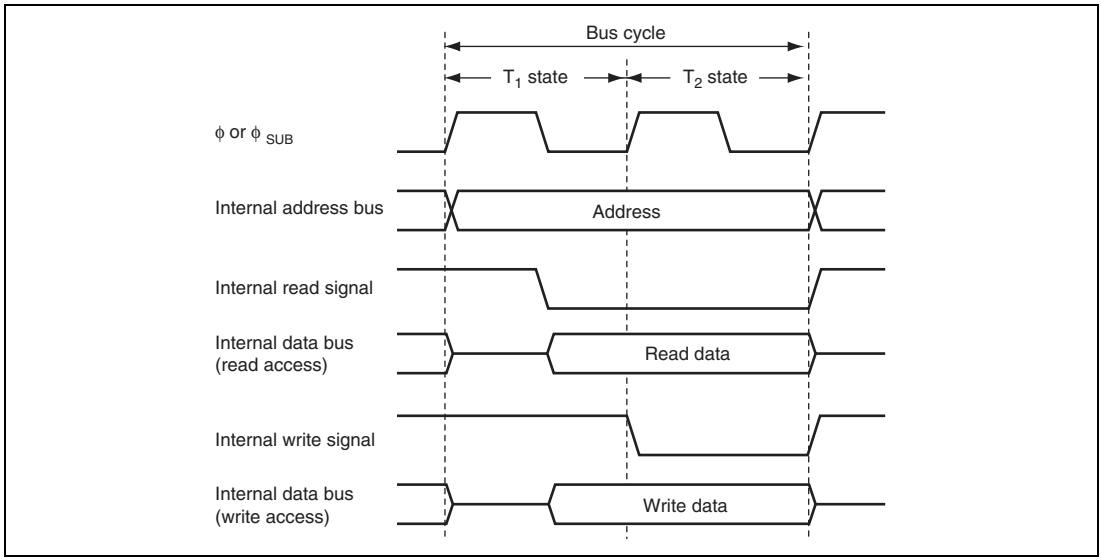


Figure 2.9 On-Chip Memory Access Cycle

2.6.2 On-Chip Peripheral Modules

On-chip peripheral modules are accessed in two states or three states. The data bus width is 8 bits or 16 bits depending on the register. For description on the data bus width and number of accessing states of each register, refer to section 19.1, Register Addresses (Address Order). Registers with 16-bit data bus width can be accessed by word size only. Registers with 8-bit data bus width can be accessed by byte or word size. When a register with 8-bit data bus width is accessed by word size, access is completed in two cycles. In two-state access, the operation timing is the same as that for on-chip memory.

Figure 2.10 shows the operation timing in the case of three-state access to an on-chip peripheral module.

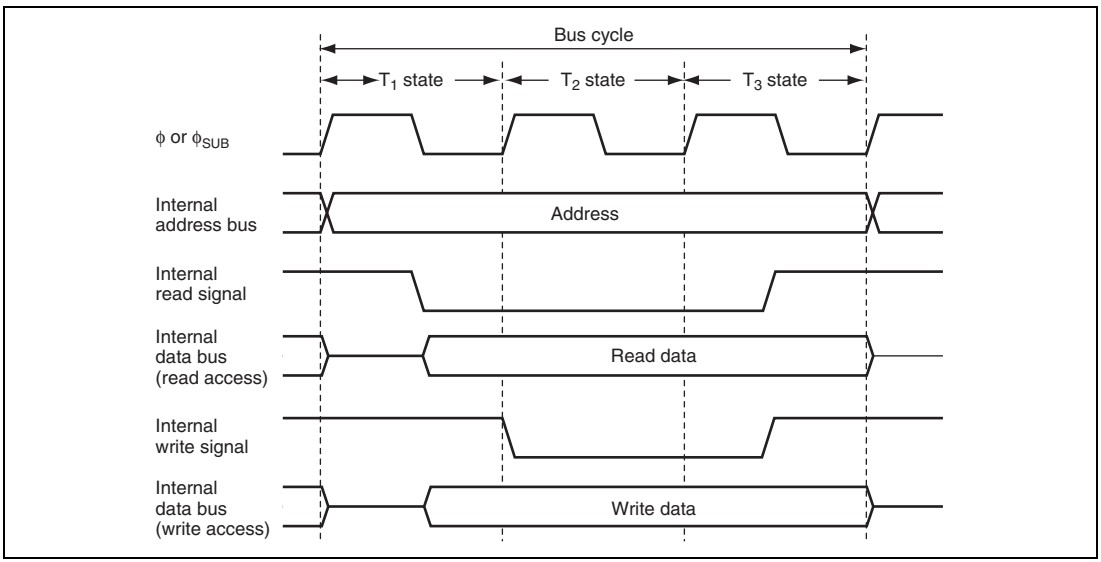


Figure 2.10 On-Chip Peripheral Module Access Cycle (3-State Access)

2.7 CPU States

There are four CPU states: the reset state, program execution state, program halt state, and exception-handling state. The program execution state includes active mode and subactive mode. In the program halt state there are a sleep mode, standby mode, and sub-sleep mode. These states are shown in figure 2.11. Figure 2.12 shows the state transitions. For details on program execution state and program halt state, refer to section 6, Power-Down Modes. For details on exception processing, refer to section 3, Exception Handling.

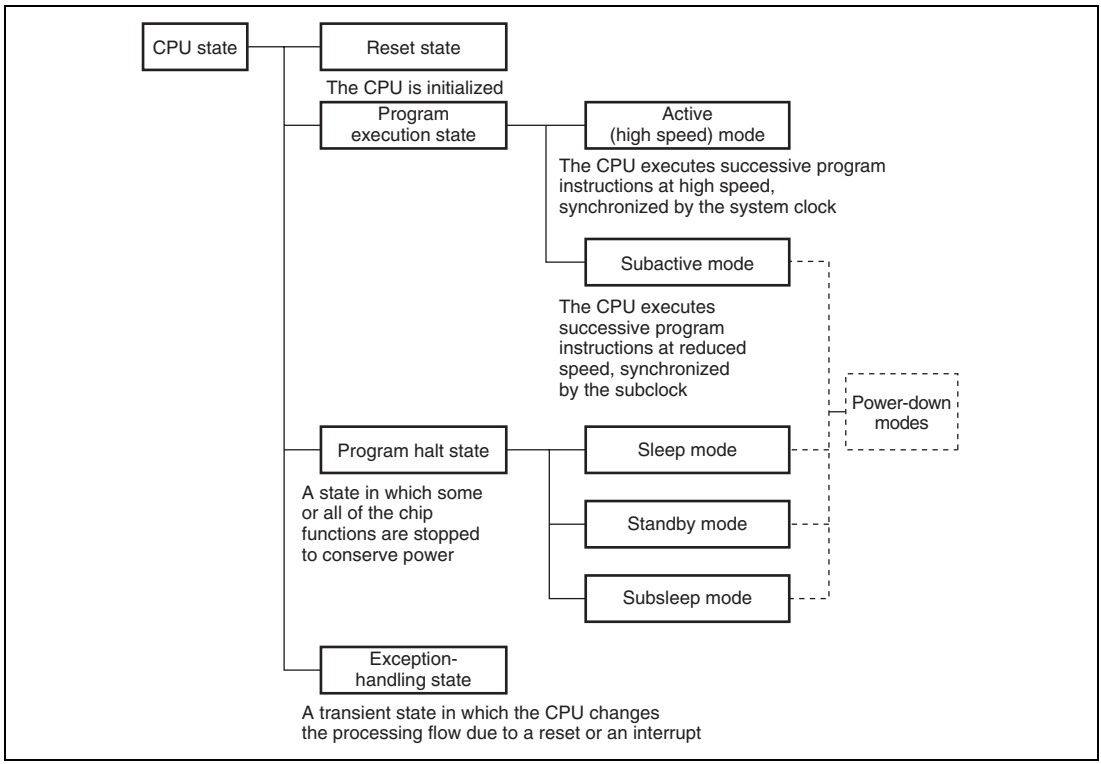


Figure 2.11 CPU Operation States

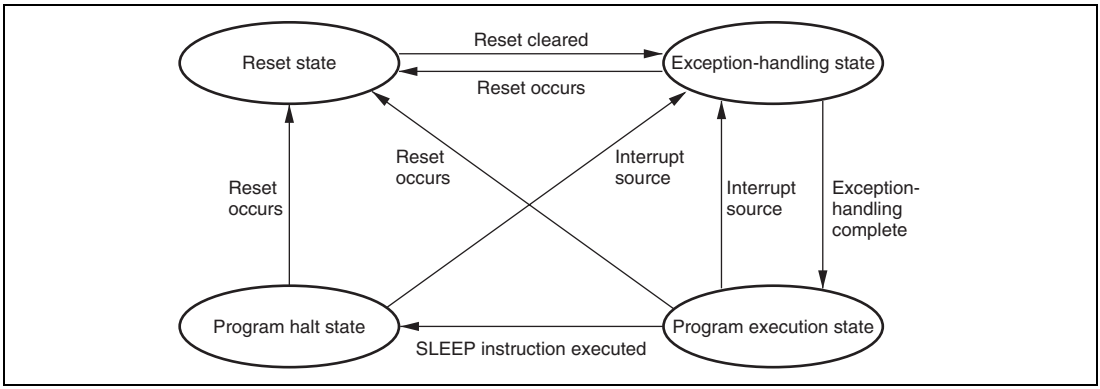


Figure 2.12 State Transitions

2.8 Usage Notes

2.8.1 Notes on Data Access to Empty Areas

The address space of this LSI includes empty areas in addition to the ROM, RAM, and on-chip I/O registers areas available to the user. When data is transferred from CPU to empty areas, the transferred data will be lost. This action may also cause the CPU to malfunction. When data is transferred from an empty area to CPU, the contents of the data cannot be guaranteed.

2.8.2 EEPMOV Instruction

EEPMOV is a block-transfer instruction and transfers the byte size of data indicated by R4L, which starts from the address indicated by R5, to the address indicated by R6. Set R4L and R6 so that the end address of the destination address (value of R6 + R4L) does not exceed H'FFFF (the value of R6 must not change from H'FFFF to H'0000 during execution).

2.8.3 Bit Manipulation Instruction

The BSET, BCLR, BNOT, BST, and BIST instructions read data from the specified address in byte units, manipulate the data of the target bit, and write data to the same address again in byte units. Special care is required when using these instructions in cases where two registers are assigned to the same address or when a bit is directly manipulated for a port, because this may rewrite data of a bit other than the bit to be manipulated.

(1) Bit manipulation for two registers assigned to the same address

Example 1: Bit manipulation for the timer load register and timer counter

(Applicable for timer B and timer C, not for the group of this LSI.)

Figure 2.13 shows an example of a timer in which two timer registers are assigned to the same address. When a bit manipulation instruction accesses the timer load register and timer counter of a reloadable timer, since these two registers share the same address, the following operations takes place.

1. Data is read in byte units.
2. The CPU sets or resets the bit to be manipulated with the bit manipulation instruction.
3. The written data is written again in byte units to the timer load register.

The timer is counting, so the value read is not necessarily the same as the value in the timer load register. As a result, bits other than the intended bit in the timer counter may be modified and the modified value may be written to the timer load register.

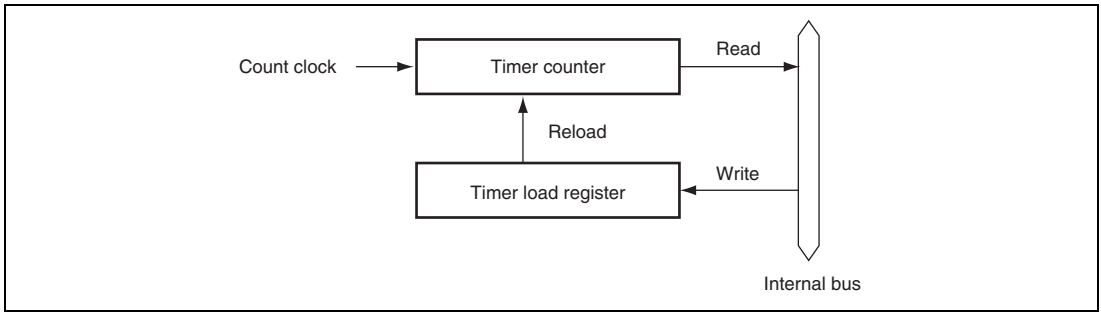


Figure 2.13 Example of Timer Configuration with Two Registers Allocated to Same Address

Example 2: The BSET instruction is executed for port 5.

P57 and P56 are input pins, with a low-level signal input at P57 and a high-level signal input at P56. P55 to P50 are output pins and output low-level signals. An example to output a high-level signal at P50 with a BSET instruction is shown below.

Prior to executing BSET

	P57	P56	P55	P54	P53	P52	P51	P50
Input/output	Input	Input	Output	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	Low level
PCR5	0	0	1	1	1	1	1	1
PDR5	1	0	0	0	0	0	0	0

BSET instruction executed

```
BSET    #0,     @PDR5
```

The BSET instruction is executed for port 5.

After executing BSET

	P57	P56	P55	P54	P53	P52	P51	P50
Input/output	Input	Input	Output	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	High level
PCR5	0	0	1	1	1	1	1	1
PDR5	0	1	0	0	0	0	0	1

Description on operation

When the BSET instruction is executed, first the CPU reads port 5.

Since P57 and P56 are input pins, the CPU reads the pin states (low-level and high-level input). P55 to P50 are output pins, so the CPU reads the value in PDR5. In this example PDR5 has a value of H'80, but the value read by the CPU is H'40.

Next, the CPU sets bit 0 of the read data to 1, changing the PDR5 data to H'41.

Finally, the CPU writes H'41 to PDR5, completing execution of BSET.

As a result of the BSET instruction, bit 0 in PDR5 becomes 1, and P50 outputs a high-level signal. However, bits 7 and 6 of PDR5 end up with different values. To prevent this problem, store a copy of the PDR5 data in a work area in memory. Perform the bit manipulation on the data in the work area, then write this data to PDR5.

Prior to executing BSET

```
MOV.B   #80,  R0L
MOV.B   R0L,  @RAM0
MOV.B   R0L,  @PDR5
```

The PDR5 value (H'80) is written to a work area in memory (RAM0) as well as to PDR5.

	P57	**P56**	**P55**	**P54**	**P53**	**P52**	**P51**	**P50**
Input/output	Input	Input	Output	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	Low level
PCR5	0	0	1	1	1	1	1	1
PDR5	1	0	0	0	0	0	0	0
RAM0	1	0	0	0	0	0	0	0

BSET instruction executed

```
BSET    #0,     @RAM0
```

The BSET instruction is executed designating the PDR5 work area (RAM0).

After executing BSET

```
MOV.B   @RAM0, R0L
MOV.B   R0L,  @PDR5
```

The work area (RAM0) value is written to PDR5.

	P57	**P56**	**P55**	**P54**	**P53**	**P52**	**P51**	**P50**
Input/output	Input	Input	Output	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	High level
PCR5	0	0	1	1	1	1	1	1
PDR5	1	0	0	0	0	0	0	1
RAM0	1	0	0	0	0	0	0	1

(2) Bit Manipulation in a Register Containing a Write-Only Bit

Example 3: BCLR instruction executed designating port 5 control register PCR5

P57 and P56 are input pins, with a low-level signal input at P57 and a high-level signal input at P56. P55 to P50 are output pins that output low-level signals. An example of setting the P50 pin as an input pin by the BCLR instruction is shown below. It is assumed that a high-level signal will be input to this input pin.

Prior to executing BCLR

	P57	**P56**	**P55**	**P54**	**P53**	**P52**	**P51**	**P50**
Input/output	Input	Input	Output	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	Low level
PCR5	0	0	1	1	1	1	1	1
PDR5	1	0	0	0	0	0	0	0

BCLR instruction executed

```
BCLR   #0,    @PCR5
```

The BCLR instruction is executed for PCR5.

After executing BCLR

	P57	**P56**	**P55**	**P54**	**P53**	**P52**	**P51**	**P50**
Input/output	Output	Output	Output	Output	Output	Output	Output	Input
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	High level
PCR5	1	1	1	1	1	1	1	0
PDR5	1	0	0	0	0	0	0	0

Description on operation

When the BCLR instruction is executed, first the CPU reads PCR5. Since PCR5 is a write-only register, the CPU reads a value of H'FF, even though the PCR5 value is actually H'3F.

Next, the CPU clears bit 0 in the read data to 0, changing the data to H'FE.

Finally, H'FE is written to PCR5 and BCLR instruction execution ends.

As a result of this operation, bit 0 in PCR5 becomes 0, making P50 an input port. However, bits 7 and 6 in PCR5 change to 1, so that P57 and P56 change from input pins to output pins. To prevent this problem, store a copy of the PDR5 data in a work area in memory and manipulate data of the bit in the work area, then write this data to PDR5.

Prior to executing BCLR

```
MOV.B   #3F,  R0L
MOV.B   R0L,  @RAM0
MOV.B   R0L,  @PCR5
```

The PCR5 value (H'3F) is written to a work area in memory (RAM0) as well as to PCR5.

	P57	**P56**	**P55**	**P54**	**P53**	**P52**	**P51**	**P50**
Input/output	Input	Input	Output	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	Low level
PCR5	0	0	1	1	1	1	1	1
PDR5	1	0	0	0	0	0	0	0
RAM0	0	0	1	1	1	1	1	1

BCLR instruction executed

```
BCLR   #0,    @RAM0
```

The BCLR instructions executed for the PCR5 work area (RAM0).

After executing BCLR

```
MOV.B   @RAM0, R0L
MOV.B   R0L,  @PCR5
```

The work area (RAM0) value is written to PCR5.

	P57	**P56**	**P55**	**P54**	**P53**	**P52**	**P51**	**P50**
Input/output	Input	Input	Output	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	High level
PCR5	0	0	1	1	1	1	1	0
PDR5	1	0	0	0	0	0	0	0
RAM0	0	0	1	1	1	1	1	0

Section 3 Exception Handling

Exception handling may be caused by a reset, a trap instruction (TRAPA), or interrupts.

- Reset

A reset has the highest exception priority. Exception handling starts as soon as the reset is cleared by the $\overline{\text{RES}}$ pin. The chip is also reset when the watchdog timer overflows, and exception handling starts. Exception handling is the same as exception handling by the $\overline{\text{RES}}$ pin.

- Trap Instruction

Exception handling starts when a trap instruction (TRAPA) is executed. The TRAPA instruction generates a vector address corresponding to a vector number from 0 to 3, as specified in the instruction code. Exception handling can be executed at all times in the program execution state.

- Interrupts

External interrupts other than NMI and internal interrupts other than address break are masked by the I bit in CCR, and kept masked while the I bit is set to 1. Exception handling starts when the current instruction or exception handling ends, if an interrupt request has been issued.

3.1 Exception Sources and Vector Address

Table 3.1 shows the vector addresses and priority of each exception handling. When more than one interrupt is requested, handling is performed from the interrupt with the highest priority.

Table 3.1 Exception Sources and Vector Address

Relative Module	Exception Sources	Vector Number	Vector Address	Priority
$\overline{\text{RES}}$ pin	Reset	0	H'0000 to H'0001	High
Watchdog timer				▲
—	Reserved for system use	1 to 6	H'0002 to H'000D	
External interrupt pin	NMI	7	H'000E to H'000F	
CPU	Trap instruction (#0)	8	H'0010 to H'0011	
	(#1)	9	H'0012 to H'0013	
	(#2)	10	H'0014 to H'0015	
	(#3)	11	H'0016 to H'0017	
Address break	Break conditions satisfied	12	H'0018 to H'0019	
CPU	Direct transition by executing the SLEEP instruction	13	H'001A to H'001B	
External interrupt pin	IRQ0	14	H'001C to H'001D	
	IRQ1	15	H'001E to H'001F	
	IRQ2	16	H'0020 to H'0021	
	IRQ3	17	H'0022 to H'0023	
	WKP	18	H'0024 to H'0025	
Timer A	Overflow	19	H'0026 to H'0027	
—	Reserved for system use	20	H'0028 to H'0029	
Timer W	Input capture A/compare match A	21	H'002A to H'002B	
	Input capture B/compare match B			
	Input capture C/compare match C			
	Input capture D/compare match D			
	Timer W overflow			
Timer V	Timer V compare match A	22	H'002C to H'002D	
	Timer V compare match B			
	Timer V overflow			
SCI3	SCI3 receive data full	23	H'002E to H'002F	
	SCI3 transmit data empty			
	SCI3 transmit end			
	SCI3 receive error			
IIC	Data transfer end	24	H'0030 to H'0031	
	Address inequality			
	Stop conditions detected			
A/D converter	A/D conversion end	25	H'0032 to H'0033	Low

3.2 Register Descriptions

Interrupts are controlled by the following registers.

- Interrupt edge select register 1 (IEGR1)
- Interrupt edge select register 2 (IEGR2)
- Interrupt enable register 1 (IENR1)
- Interrupt flag register 1 (IRR1)
- Wakeup interrupt flag register (IWPR)

3.2.1 Interrupt Edge Select Register 1 (IEGR1)

IEGR1 selects the direction of an edge that generates interrupt requests of pins $\overline{\text{NMI}}$ and $\overline{\text{IRQ3}}$ to $\overline{\text{IRQ0}}$.

Bit	Bit Name	Initial Value	R/W	Description
7	NMIEG	0	R/W	NMI Edge Select 0: Falling edge of $\overline{\text{NMI}}$ pin input is detected 1: Rising edge of $\overline{\text{NMI}}$ pin input is detected
6	—	1	—	Reserved
5	—	1	—	These bits are always read as 1.
4	—	1	—	
3	IEG3	0	R/W	IRQ3 Edge Select 0: Falling edge of $\overline{\text{IRQ3}}$ pin input is detected 1: Rising edge of $\overline{\text{IRQ3}}$ pin input is detected
2	IEG2	0	R/W	IRQ2 Edge Select 0: Falling edge of $\overline{\text{IRQ2}}$ pin input is detected 1: Rising edge of $\overline{\text{IRQ2}}$ pin input is detected
1	IEG1	0	R/W	IRQ1 Edge Select 0: Falling edge of $\overline{\text{IRQ1}}$ pin input is detected 1: Rising edge of $\overline{\text{IRQ1}}$ pin input is detected
0	IEG0	0	R/W	IRQ0 Edge Select 0: Falling edge of $\overline{\text{IRQ0}}$ pin input is detected 1: Rising edge of $\overline{\text{IRQ0}}$ pin input is detected

3.2.2 Interrupt Edge Select Register 2 (IEGR2)

IEGR2 selects the direction of an edge that generates interrupt requests of the pins $\overline{\text{ADTRG}}$ and $\overline{\text{WKP5}}$ to $\overline{\text{WKP0}}$.

Bit	Bit Name	Initial Value	R/W	Description
7	—	1	—	Reserved
6	—	1	—	These bits are always read as 1.
5	WPEG5	0	R/W	WKP5 Edge Select 0: Falling edge of $\overline{\text{WKP5}}$ ($\overline{\text{ADTRG}}$) pin input is detected 1: Rising edge of $\overline{\text{WKP5}}$ ($\overline{\text{ADTRG}}$) pin input is detected
4	WPEG4	0	R/W	WKP4 Edge Select 0: Falling edge of $\overline{\text{WKP4}}$ pin input is detected 1: Rising edge of $\overline{\text{WKP4}}$ pin input is detected
3	WPEG3	0	R/W	WKP3 Edge Select 0: Falling edge of $\overline{\text{WKP3}}$ pin input is detected 1: Rising edge of $\overline{\text{WKP3}}$ pin input is detected
2	WPEG2	0	R/W	WKP2 Edge Select 0: Falling edge of $\overline{\text{WKP2}}$ pin input is detected 1: Rising edge of $\overline{\text{WKP2}}$ pin input is detected
1	WPEG1	0	R/W	WKP1Edge Select 0: Falling edge of $\overline{\text{WKP1}}$ pin input is detected 1: Rising edge of $\overline{\text{WKP1}}$ pin input is detected
0	WPEG0	0	R/W	WKP0 Edge Select 0: Falling edge of $\overline{\text{WKP0}}$ pin input is detected 1: Rising edge of $\overline{\text{WKP0}}$ pin input is detected

3.2.3 Interrupt Enable Register 1 (IENR1)

IENR1 enables direct transition interrupts, timer A overflow interrupts, and external pin interrupts.

Bit	Bit Name	Initial Value	R/W	Description
7	IENDT	0	R/W	Direct Transfer Interrupt Enable When this bit is set to 1, direct transition interrupt requests are enabled.
6	IENTA	0	R/W	Timer A Interrupt Enable When this bit is set to 1, timer A overflow interrupt requests are enabled.
5	IENWP	0	R/W	Wakeup Interrupt Enable This bit is an enable bit, which is common to the pins $\overline{\text{WKP5}}$ to $\overline{\text{WKP0}}$. When the bit is set to 1, interrupt requests are enabled.
4	—	1	—	Reserved This bit is always read as 1.
3	IEN3	0	R/W	IRQ3 Interrupt Enable When this bit is set to 1, interrupt requests of the $\overline{\text{IRQ3}}$ pin are enabled.
2	IEN2	0	R/W	IRQ2 Interrupt Enable When this bit is set to 1, interrupt requests of the $\overline{\text{IRQ2}}$ pin are enabled.
1	IEN1	0	R/W	IRQ1 Interrupt Enable When this bit is set to 1, interrupt requests of the $\overline{\text{IRQ1}}$ pin are enabled.
0	IEN0	0	R/W	IRQ0 Interrupt Enable When this bit is set to 1, interrupt requests of the $\overline{\text{IRQ0}}$ pin are enabled.

When disabling interrupts by clearing bits in an interrupt enable register, or when clearing bits in an interrupt flag register, always do so while interrupts are masked (I = 1). If the above clear operations are performed while I = 0, and as a result a conflict arises between the clear instruction and an interrupt request, exception handling for the interrupt will be executed after the clear instruction has been executed.

3.2.4 Interrupt Flag Register 1 (IRR1)

IRR1 is a status flag register for direct transition interrupts, timer A overflow interrupts, and $\overline{\text{IRQ3}}$ to $\overline{\text{IRQ0}}$ interrupt requests.

Bit	Bit Name	Initial Value	R/W	Description
7	IRRDT	0	R/W	Direct Transfer Interrupt Request Flag [Setting condition] When a direct transfer is made by executing a SLEEP instruction while DTON in SYSCR2 is set to 1. [Clearing condition] When IRRDT is cleared by writing 0
6	IRRTA	0	R/W	Timer A Interrupt Request Flag [Setting condition] When the timer A counter value overflows [Clearing condition] When IRRTA is cleared by writing 0
5	—	1	—	Reserved
4	—	1	—	These bits are always read as 1.
3	IRRI3	0	R/W	IRQ3 Interrupt Request Flag [Setting condition] When $\overline{\text{IRQ3}}$ pin is designated for interrupt input and the designated signal edge is detected. [Clearing condition] When IRRI3 is cleared by writing 0
2	IRRI2	0	R/W	IRQ2 Interrupt Request Flag [Setting condition] When $\overline{\text{IRQ2}}$ pin is designated for interrupt input and the designated signal edge is detected. [Clearing condition] When IRRI2 is cleared by writing 0
1	IRRI1	0	R/W	IRQ1 Interrupt Request Flag [Setting condition] When $\overline{\text{IRQ1}}$ pin is designated for interrupt input and the designated signal edge is detected. [Clearing condition] When IRRI1 is cleared by writing 0

Bit	Bit Name	Initial Value	R/W	Description
0	IRRI0	0	R/W	IRQ0 Interrupt Request Flag [Setting condition] When $\overline{\text{IRQ0}}$ pin is designated for interrupt input and the designated signal edge is detected. [Clearing condition] When IRRI0 is cleared by writing 0

3.2.5 Wakeup Interrupt Flag Register (IWPR)

IWPR is a status flag register for $\overline{\text{WKP5}}$ to $\overline{\text{WKP0}}$ interrupt requests.

Bit	Bit Name	Initial Value	R/W	Description
7	—	1	—	Reserved
6	—	1	—	These bits are always read as 1.
5	IWPF5	0	R/W	WKP5 Interrupt Request Flag [Setting condition] When $\overline{\text{WKP5}}$ pin is designated for interrupt input and the designated signal edge is detected. [Clearing condition] When IWPF5 is cleared by writing 0.
4	IWPF4	0	R/W	WKP4 Interrupt Request Flag [Setting condition] When $\overline{\text{WKP4}}$ pin is designated for interrupt input and the designated signal edge is detected. [Clearing condition] When IWPF4 is cleared by writing 0.
3	IWPF3	0	R/W	WKP3 Interrupt Request Flag [Setting condition] When $\overline{\text{WKP3}}$ pin is designated for interrupt input and the designated signal edge is detected. [Clearing condition] When IWPF3 is cleared by writing 0.

Bit	Bit Name	Initial Value	R/W	Description
2	IWPF2	0	R/W	WKP2 Interrupt Request Flag [Setting condition] When $\overline{\text{WKP2}}$ pin is designated for interrupt input and the designated signal edge is detected. [Clearing condition] When IWPF2 is cleared by writing 0.
1	IWPF1	0	R/W	WKP1 Interrupt Request Flag [Setting condition] When $\overline{\text{WKP1}}$ pin is designated for interrupt input and the designated signal edge is detected. [Clearing condition] When IWPF1 is cleared by writing 0.
0	IWPF0	0	R/W	WKP0 Interrupt Request Flag [Setting condition] When $\overline{\text{WKP0}}$ pin is designated for interrupt input and the designated signal edge is detected. [Clearing condition] When IWPF0 is cleared by writing 0.

3.3 Reset Exception Handling

When the $\overline{\text{RES}}$ pin goes low, all processing halts and this LSI enters the reset. The internal state of the CPU and the registers of the on-chip peripheral modules are initialized by the reset. To ensure that this LSI is reset at power-up, hold the $\overline{\text{RES}}$ pin low until the clock pulse generator output stabilizes. To reset the chip during operation, hold the $\overline{\text{RES}}$ pin low for at least 10 system clock cycles. When the $\overline{\text{RES}}$ pin goes high after being held low for the necessary time, this LSI starts reset exception handling. The reset exception handling sequence is shown in figure 3.1.

The reset exception handling sequence is as follows:

1. Set the I bit in the condition code register (CCR) to 1.
2. The CPU generates a reset exception handling vector address (from H'0000 to H'0001), the data in that address is sent to the program counter (PC) as the start address, and program execution starts from that address.

3.4 Interrupt Exception Handling

3.4.1 External Interrupts

There are external interrupts, NMI, IRQ3 to IRQ0, and WKP5 to WKP0.

(1) NMI Interrupt

NMI interrupt is requested by input signal edge to pin $\overline{\text{NMI}}$. This interrupt is detected by either rising edge sensing or falling edge sensing, depending on the setting of bit NMIEG in IEGR1.

NMI is the highest-priority interrupt, and can always be accepted without depending on the I bit value in CCR.

(2) IRQ3 to IRQ0 Interrupts

IRQ3 to IRQ0 interrupts are requested by input signals to pins $\overline{\text{IRQ3}}$ to $\overline{\text{IRQ0}}$. These four interrupts are given different vector addresses, and are detected individually by either rising edge sensing or falling edge sensing, depending on the settings of bits IEG3 to IEG0 in IEGR1.

When pins $\overline{\text{IRQ3}}$ to $\overline{\text{IRQ0}}$ are designated for interrupt input in PMR1 and the designated signal edge is input, the corresponding bit in IRR1 is set to 1, requesting the CPU of an interrupt. When IRQ3 to IRQ0 interrupt is accepted, the I bit is set to 1 in CCR. These interrupts can be masked by setting bits IEN3 to IEN0 in IENR1.

(3) WKP5 to WKP0 Interrupts

WKP5 to WKP0 interrupts are requested by input signals to pins $\overline{\text{WKP5}}$ to $\overline{\text{WKP0}}$. These six interrupts have the same vector addresses, and are detected individually by either rising edge sensing or falling edge sensing, depending on the settings of bits WPEG5 to WPEG0 in IEGR2.

When pins $\overline{\text{WKP5}}$ to $\overline{\text{WKP0}}$ are designated for interrupt input in PMR5 and the designated signal edge is input, the corresponding bit in IWPR is set to 1, requesting the CPU of an interrupt. These interrupts can be masked by setting bit IENWP in IENR1.

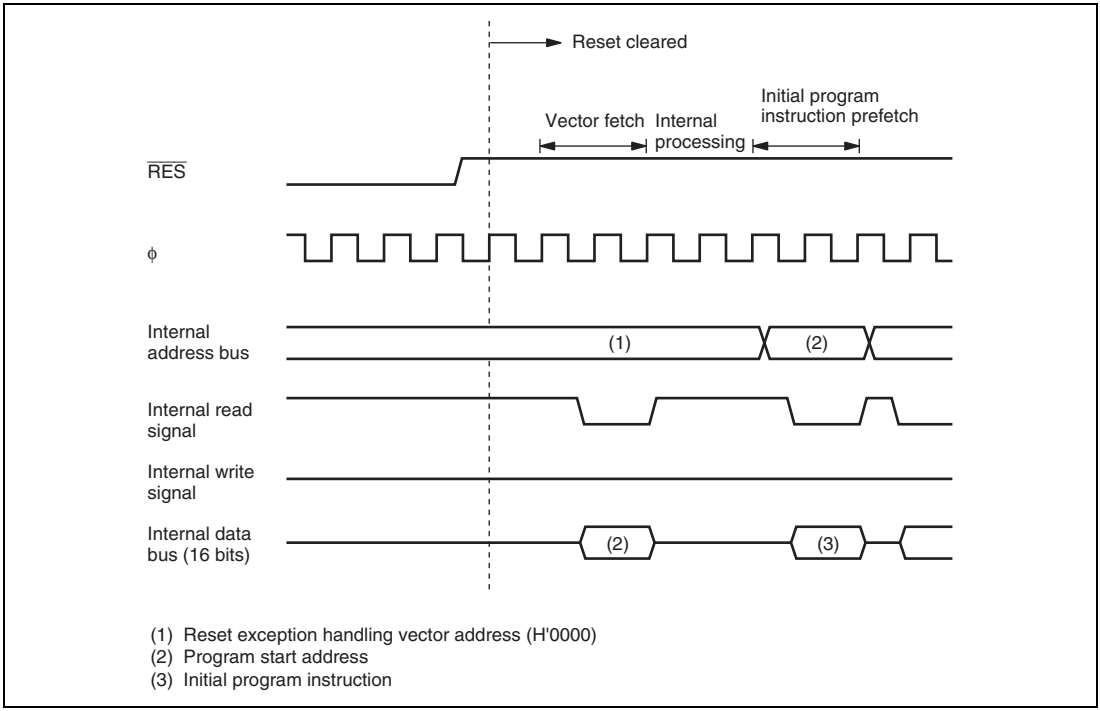


Figure 3.1 Reset Sequence

3.4.2 Internal Interrupts

Each on-chip peripheral module has a flag to show the interrupt request status and the enable bit to enable or disable the interrupt. For timer A interrupt requests and direct transfer interrupt requests generated by execution of a SLEEP instruction, this function is included in IRR1 and IENR1.

When an on-chip peripheral module requests an interrupt, the corresponding interrupt request status flag is set to 1, requesting the CPU of an interrupt. When this interrupt is accepted, the I bit is set to 1 in CCR. These interrupts can be masked by writing 0 to clear the corresponding enable bit.

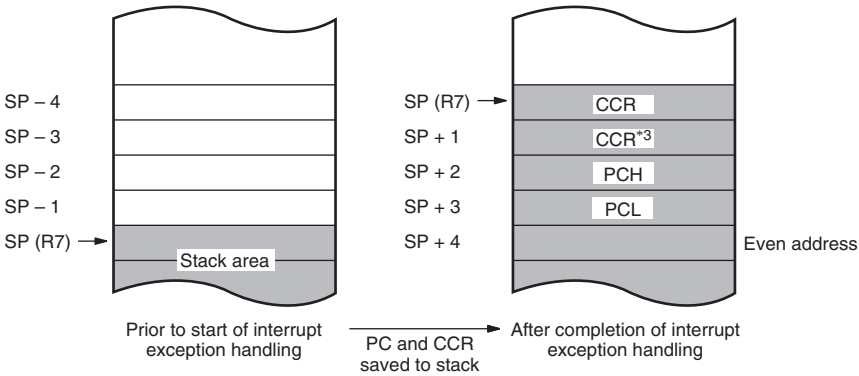
3.4.3 Interrupt Handling Sequence

Interrupts are controlled by an interrupt controller.

Interrupt operation is described as follows.

1. If an interrupt occurs while the NMI or interrupt enable bit is set to 1, an interrupt request signal is sent to the interrupt controller.
2. When multiple interrupt requests are generated, the interrupt controller requests to the CPU for the interrupt handling with the highest priority at that time according to table 3.1. Other interrupt requests are held pending.
3. The CPU accepts the NMI and address break without depending on the I bit value. Other interrupt requests are accepted, if the I bit is cleared to 0 in CCR; if the I bit is set to 1, the interrupt request is held pending.
4. If the CPU accepts the interrupt after processing of the current instruction is completed, interrupt exception handling will begin. First, both PC and CCR are pushed onto the stack. The state of the stack at this time is shown in figure 3.2. The PC value pushed onto the stack is the address of the first instruction to be executed upon return from interrupt handling.
5. Then, the I bit of CCR is set to 1, masking further interrupts excluding the NMI and address break. Upon return from interrupt handling, the values of I bit and other bits in CCR will be restored and returned to the values prior to the start of interrupt exception handling.
6. Next, the CPU generates the vector address corresponding to the accepted interrupt, and transfers the address to PC as a start address of the interrupt handling-routine. Then a program starts executing from the address indicated in PC.

Figure 3.3 shows a typical interrupt sequence where the program area is in the on-chip ROM and the stack area is in the on-chip RAM.



[Legend]
PCH: Upper 8 bits of program counter (PC)
PCL: Lower 8 bits of program counter (PC)
CCR: Condition code register
SP: Stack pointer

Notes: 1. PC shows the address of the first instruction to be executed upon return from the interrupt handling routine.
2. Register contents must always be saved and restored by word length, starting from an even-numbered address.
3. Ignored when returning from the interrupt handling routine.

Figure 3.2 Stack Status after Exception Handling

3.4.4 Interrupt Response Time

Table 3.2 shows the number of wait states after an interrupt request flag is set until the first instruction of the interrupt handling-routine is executed.

Table 3.2 Interrupt Wait States

Item	States	Total
Waiting time for completion of executing instruction*	1 to 23	15 to 37
Saving of PC and CCR to stack	4	
Vector fetch	2	
Instruction fetch	4	
Internal processing	4	

Note: * Not including EEPMOV instruction.

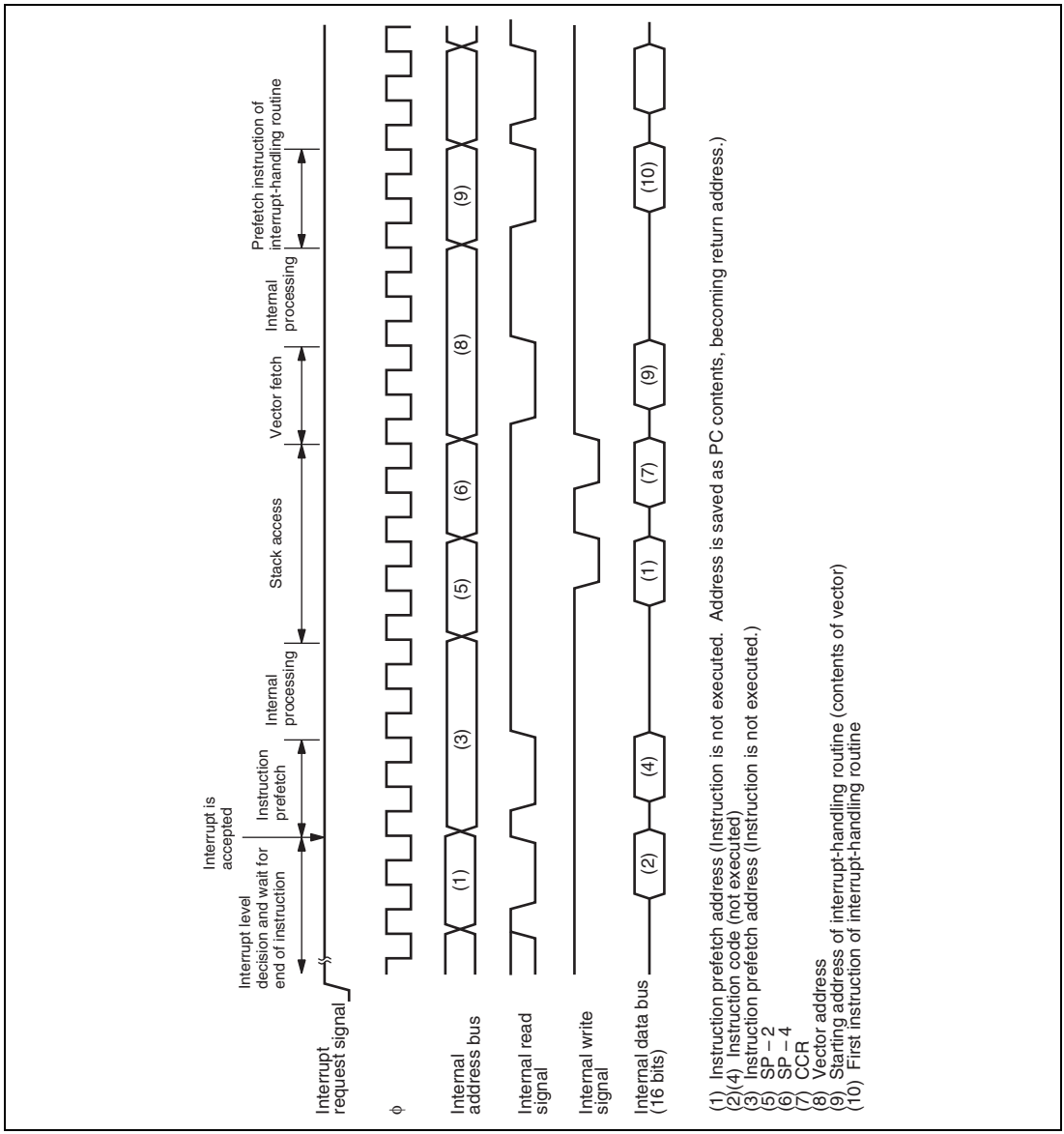


Figure 3.3 Interrupt Sequence

3.5 Usage Notes

3.5.1 Interrupts after Reset

If an interrupt is accepted 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 the reset state ends, make sure that this instruction initializes the stack pointer (example: MOV.W #xx: 16, SP).

3.5.2 Notes on Stack Area Use

When word data is accessed the least significant bit of the address is regarded as 0. Access to the stack always takes place in word size, so the stack pointer (SP: R7) should never indicate an odd address. Use PUSH Rn (MOV.W Rn, @–SP) or POP Rn (MOV.W @SP+, Rn) to save or restore register values.

3.5.3 Notes on Rewriting Port Mode Registers

When a port mode register is rewritten to switch the functions of external interrupt pins, $\overline{\text{IRQ3}}$ to $\overline{\text{IRQ0}}$, and $\overline{\text{WKP5}}$ to $\overline{\text{WKP0}}$, the interrupt request flag may be set to 1.

Figure 3.4 shows a port mode register setting and interrupt request flag clearing procedure.

When switching a pin function, mask the interrupt before setting the bit in the port mode register. After accessing the port mode register, execute at least one instruction (e.g., NOP), then clear the interrupt request flag from 1 to 0.

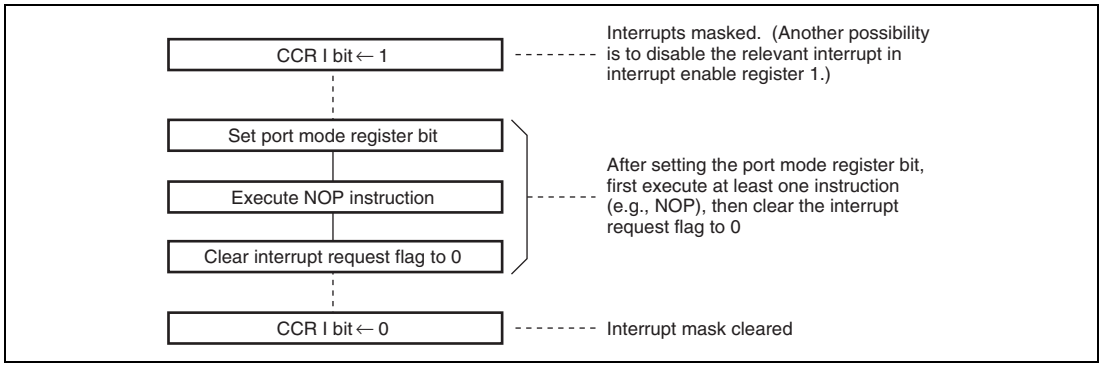


Figure 3.4 Port Mode Register Setting and Interrupt Request Flag Clearing Procedure

Section 4 Address Break

The address break simplifies on-board program debugging. It requests an address break interrupt when the set break condition is satisfied. The interrupt request is not affected by the I bit of CCR. Break conditions that can be set include instruction execution at a specific address and a combination of access and data at a specific address. With the address break function, the execution start point of a program containing a bug is detected and execution is branched to the correcting program. Figure 4.1 shows a block diagram of the address break.

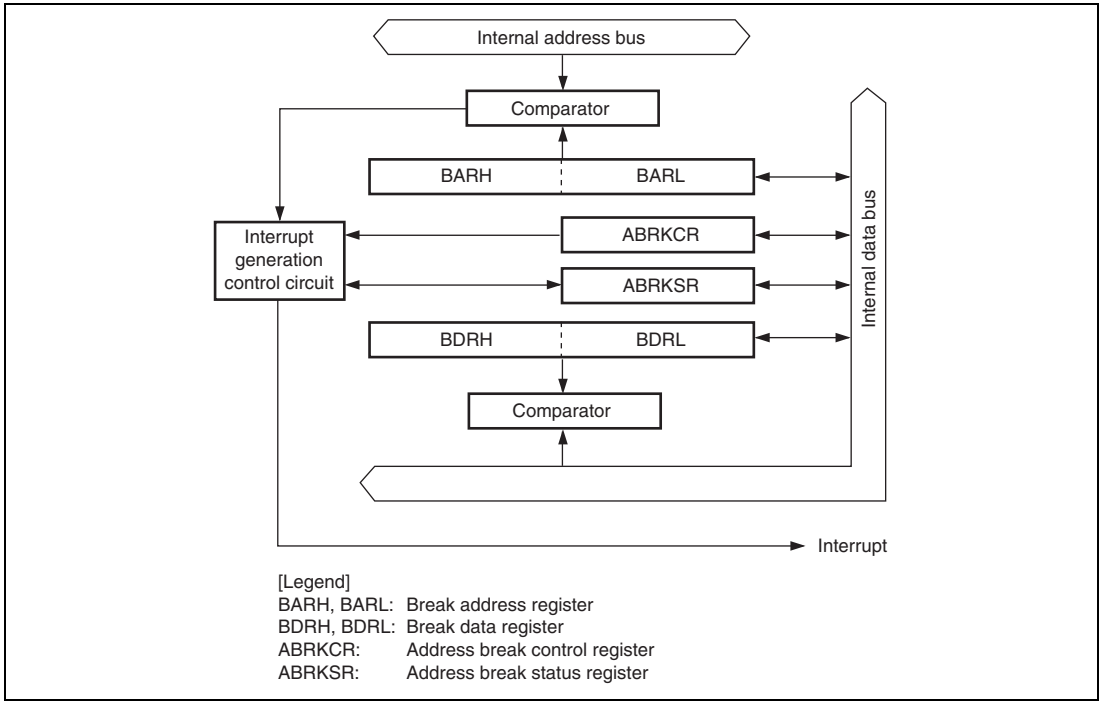


Figure 4.1 Block Diagram of Address Break

4.1 Register Descriptions

Address break has the following registers.

- Address break control register (ABRKCR)
- Address break status register (ABRKSR)
- Break address register (BARH, BARL)
- Break data register (BDRH, BDRL)

4.1.1 Address Break Control Register (ABRKCR)

ABRKCR sets address break conditions.

Bit	Bit Name	Initial Value	R/W	Description
7	RTINTE	1	R/W	RTE Interrupt Enable When this bit is 0, the interrupt immediately after executing RTE is masked and then one instruction must be executed. When this bit is 1, the interrupt is not masked.
6	CSEL1	0	R/W	Condition Select 1 and 0 These bits set address break conditions. 00: Instruction execution cycle 01: CPU data read cycle 10: CPU data write cycle 11: CPU data read/write cycle
5	CSEL0	0	R/W	
4	ACMP2	0	R/W	Address Compare Condition Select 2 to 0 These bits comparison condition between the address set in BAR and the internal address bus. 000: Compares 16-bit addresses 001: Compares upper 12-bit addresses 010: Compares upper 8-bit addresses 011: Compares upper 4-bit addresses 1XX: Reserved (setting prohibited)
3	ACMP1	0	R/W	
2	ACMP0	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
1	DCMP1	0	R/W	Data Compare Condition Select 1 and 0
0	DCMP0	0	R/W	These bits set the comparison condition between the data set in BDR and the internal data bus. 00: No data comparison 01: Compares lower 8-bit data between BDRL and data bus 10: Compares upper 8-bit data between BDRH and data bus 11: Compares 16-bit data between BDR and data bus

[Legend]

X: Don't care.

When an address break is set in the data read cycle or data write cycle, the data bus used will depend on the combination of the byte/word access and address. Table 4.1 shows the access and data bus used. When an I/O register space with an 8-bit data bus width is accessed in word size, a byte access is generated twice. For details on data widths of each register, see section 19.1, Register Addresses (Address Order).

Table 4.1 Access and Data Bus Used

	Word Access		Byte Access	
	Even Address	Odd Address	Even Address	Odd Address
ROM space	Upper 8 bits	Lower 8 bits	Upper 8 bits	Upper 8 bits
RAM space	Upper 8 bits	Lower 8 bits	Upper 8 bits	Upper 8 bits
I/O register with 8-bit data bus width	Upper 8 bits	Upper 8 bits	Upper 8 bits	Upper 8 bits
I/O register with 16-bit data bus width	Upper 8 bits	Lower 8 bits	—	—

4.1.2 Address Break Status Register (ABRKSR)

ABRKSR consists of the address break interrupt flag and the address break interrupt enable bit.

Bit	Bit Name	Initial Value	R/W	Description
7	ABIF	0	R/W	Address Break Interrupt Flag [Setting condition] When the condition set in ABRKCR is satisfied [Clearing condition] When 0 is written after ABIF=1 is read
6	ABIE	0	R/W	Address Break Interrupt Enable When this bit is 1, an address break interrupt request is enabled.
5 to 0	—	All 1	—	Reserved These bits are always read as 1.

4.1.3 Break Address Registers (BARH, BARL)

BARH and BARL are 16-bit read/write registers that set the address for generating an address break interrupt. When setting the address break condition to the instruction execution cycle, set the first byte address of the instruction. The initial value of this register is H'FFFF.

4.1.4 Break Data Registers (BDRH, BDRL)

BDRH and BDRL are 16-bit read/write registers that set the data for generating an address break interrupt. BDRH is compared with the upper 8-bit data bus. BDRL is compared with the lower 8-bit data bus. When memory or registers are accessed by byte, the upper 8-bit data bus is used for even and odd addresses in the data transmission. Therefore, comparison data must be set in BDRH for byte access. For word access, the data bus used depends on the address. See section 4.1.1, Address Break Control Register (ABRKCR), for details. The initial value of this register is undefined.

4.2 Operation

When the ABIF and ABIE bits in ABRKSR are set to 1, the address break function generates an interrupt request to the CPU. The ABIF bit in ABRKSR is set to 1 by the combination of the address set in BAR, the data set in BDR, and the conditions set in ABRKCR. When the interrupt request is accepted, interrupt exception handling starts after the instruction being executed ends. The address break interrupt is not masked because of the I bit in CCR of the CPU.

Figures 4.2 show the operation examples of the address break interrupt setting.

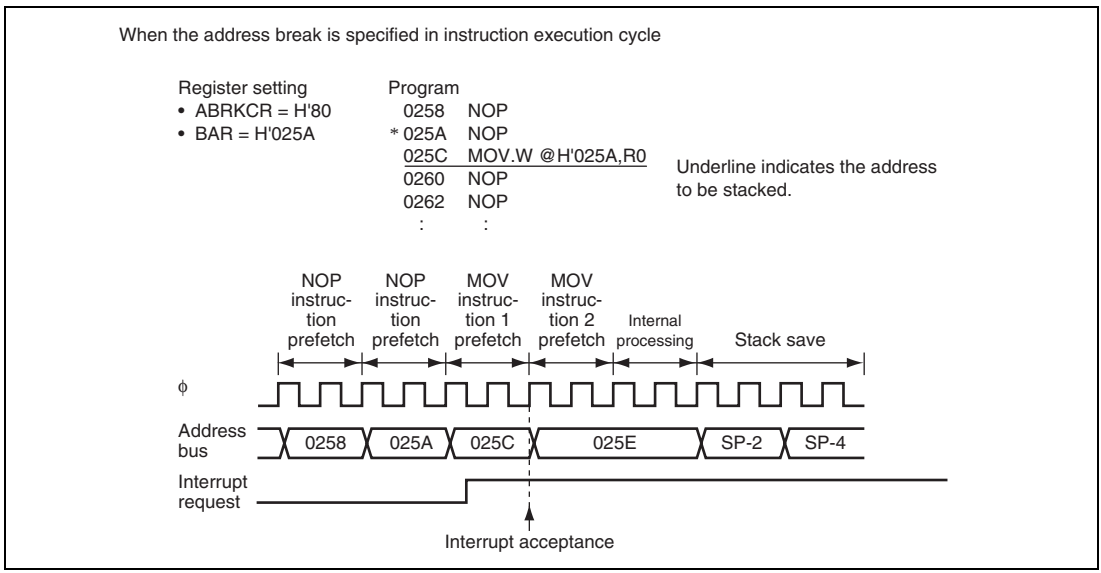


Figure 4.2 Address Break Interrupt Operation Example (1)

When the address break is specified in the data read cycle

Register setting
- ABRKCR = H'A0
- BAR = H'025A

Program

```
  0258  NOP
  025A  NOP
* 025C  MOV.W @H'025A,R0
  0260  NOP
  0262  NOP
   :     :
```

Underline indicates the address to be stacked.

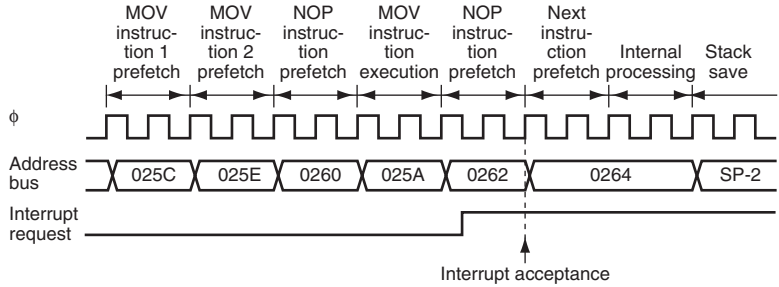


Figure 4.2 Address Break Interrupt Operation Example (2)

4.3 Usage Notes

When an address break is set to an instruction after a conditional branch instruction, and the instruction set when the condition of the branch instruction is not satisfied is executed (see figure 4.3), note that an address break interrupt request is not generated. Therefore an address break must not be set to the instruction after a conditional branch instruction.

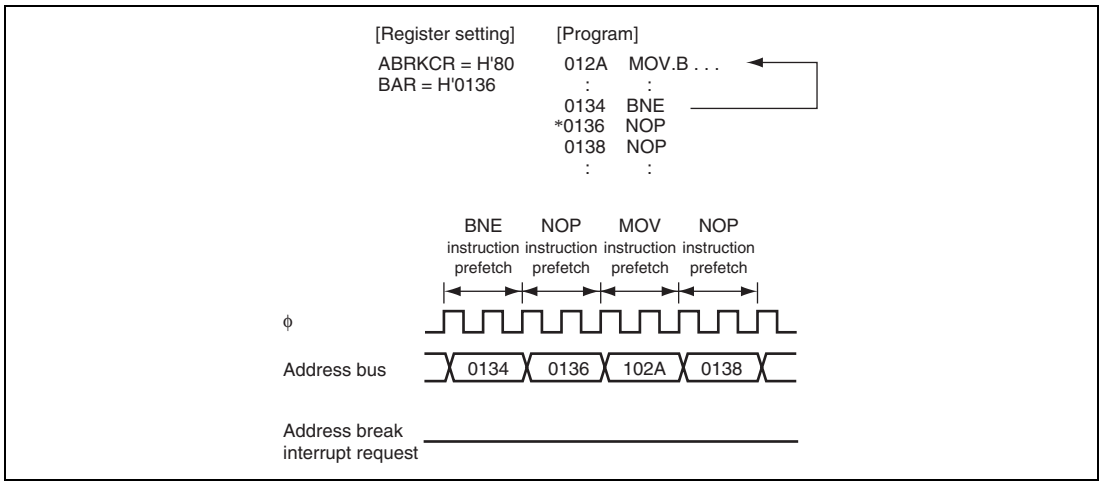


Figure 4.3 Operation when Condition is not Satisfied in Branch Instruction

When another interrupt request is accepted before an instruction to which an address break is set is executed, exception handling of an address break interrupt is not executed. However, the ABIF bit is set to 1 (see figure 4.4). Therefore the ABIF bit must be read during exception handling of an address break interrupt.

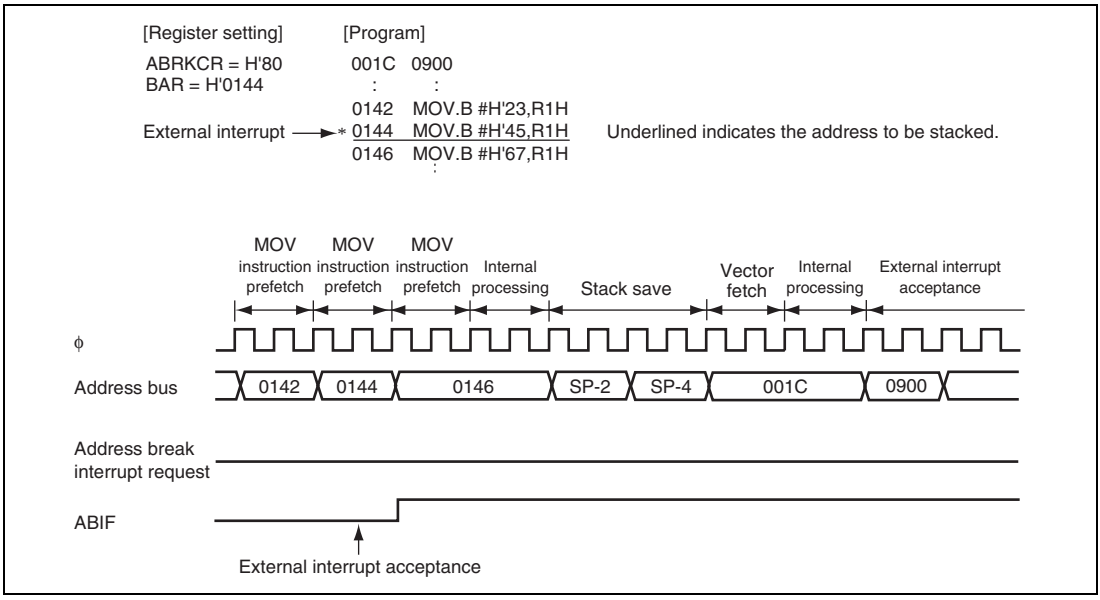


Figure 4.4 Operation when Another Interrupt is Accepted at Address Break Setting Instruction

When an address break is set to an instruction as a branch destination of a conditional branch instruction, the instruction set when the condition of the branch instruction is not satisfied is not executed, and an address break is generated. Therefore an address break must not be set to the instruction as a branch destination of a conditional branch instruction.

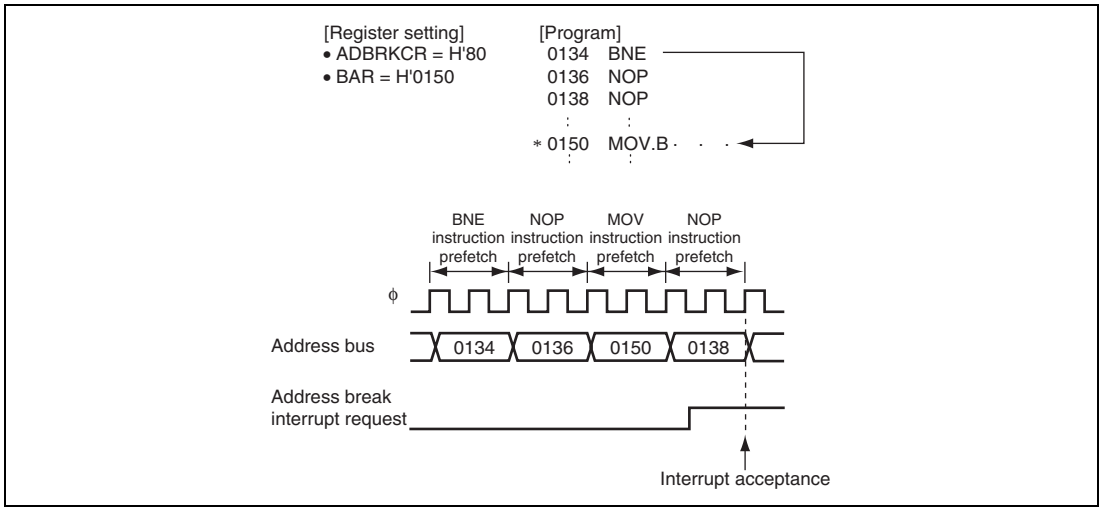


Figure 4.5 Operation when the Instruction Set is not Executed and does not Branch due to Conditions not Being Satisfied

Section 5 Clock Pulse Generators

Clock oscillator circuitry (CPG: clock pulse generator) is provided on-chip, including both a system clock pulse generator and a subclock pulse generator. The system clock pulse generator consists of a system clock oscillator, a duty correction circuit, and system clock dividers. The subclock pulse generator consists of a subclock oscillator circuit and a subclock divider.

Figure 5.1 shows a block diagram of the clock pulse generators.

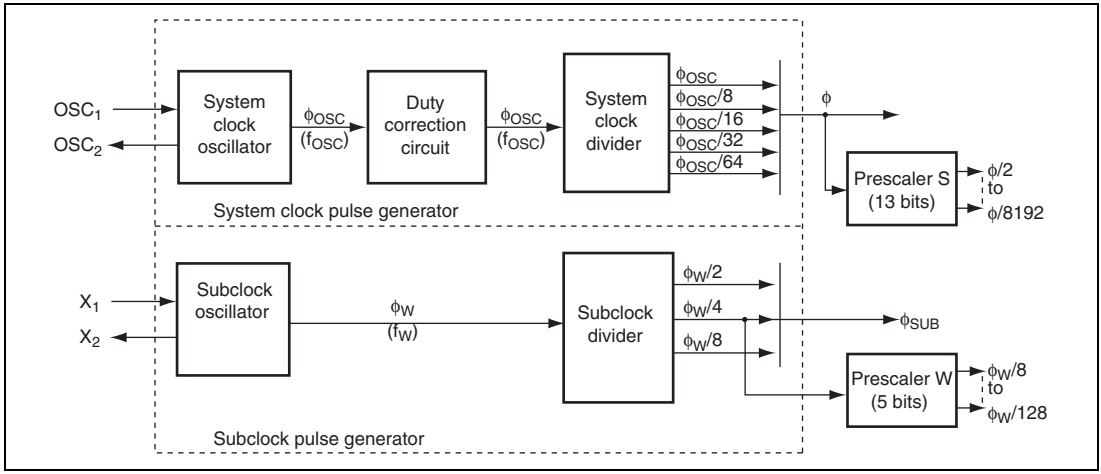
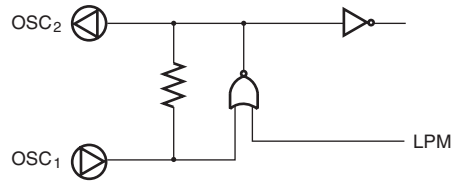


Figure 5.1 Block Diagram of Clock Pulse Generators

The basic clock signals that drive the CPU and on-chip peripheral modules are ϕ and ϕ_{SUB}. The system clock is divided by prescaler S to become a clock signal from $\phi/8192$ to $\phi/2$, and the subclock is divided by prescaler W to become a clock signal from $\phi w/128$ to $\phi w/8$. Both the system clock and subclock signals are provided to the on-chip peripheral modules.

5.1 System Clock Generator

Clock pulses can be supplied to the system clock divider either by connecting a crystal or ceramic resonator, or by providing external clock input. Figure 5.2 shows a block diagram of the system clock generator.

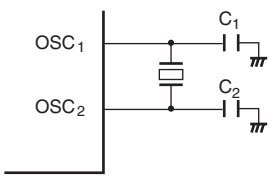


LPM: Low-power mode (standby mode, subactive mode, subsleep mode)

Figure 5.2 Block Diagram of System Clock Generator

5.1.1 Connecting Crystal Resonator

Figure 5.3 shows a typical method of connecting a crystal resonator. An AT-cut parallel-resonance crystal resonator should be used. Figure 5.4 shows the equivalent circuit of a crystal resonator. A resonator having the characteristics given in table 5.1 should be used.



$C_1 = C_2 = 12$ pF ±20%

Note: Capacitances are reference values.

Figure 5.3 Typical Connection to Crystal Resonator

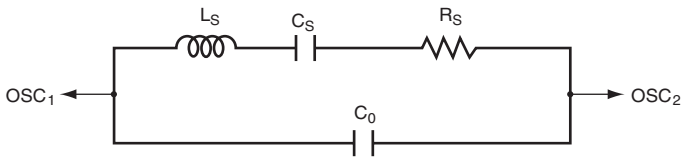


Figure 5.4 Equivalent Circuit of Crystal Resonator

Table 5.1 Crystal Resonator Parameters

Frequency (MHz)	2	4	8	10	16
R_s (max)	500 Ω	120 Ω	80 Ω	60 Ω	50 Ω
C_0 (max)	7 pF	7 pF	7 pF	7 pF	7 pF

5.1.2 Connecting Ceramic Resonator

Figure 5.5 shows a typical method of connecting a ceramic resonator.

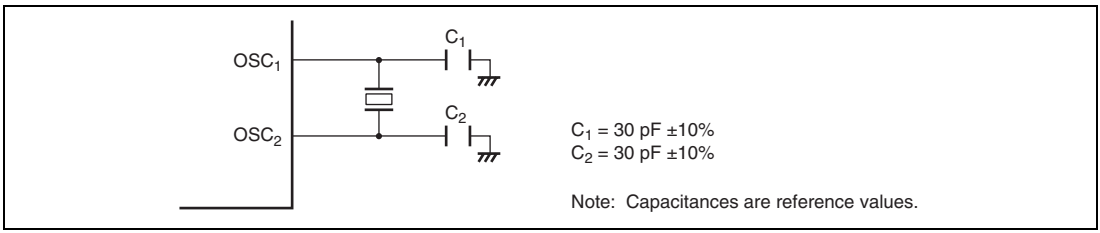


Figure 5.5 Typical Connection to Ceramic Resonator

5.1.3 External Clock Input Method

Connect an external clock signal to pin OSC_1, and leave pin OSC_2 open. Figure 5.6 shows a typical connection. The duty cycle of the external clock signal must be 45 to 55%.

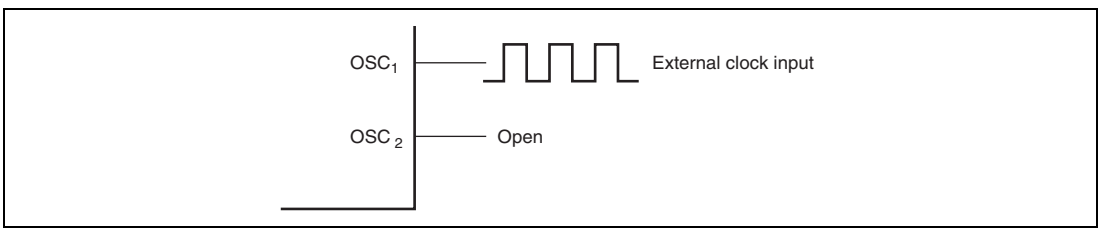
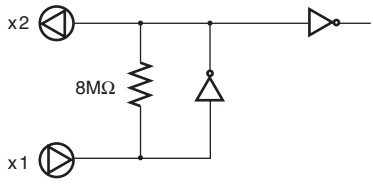


Figure 5.6 Example of External Clock Input

5.2 Subclock Generator

Figure 5.7 shows a block diagram of the subclock generator.

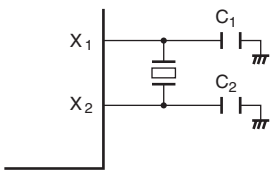


Note : Resistance is a reference value.

Figure 5.7 Block Diagram of Subclock Generator

5.2.1 Connecting 32.768-kHz Crystal Resonator

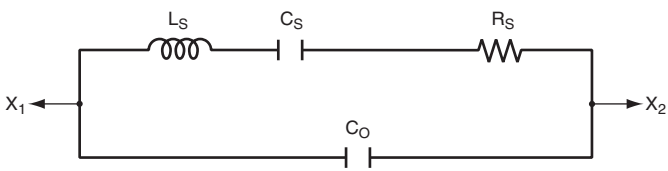
Clock pulses can be supplied to the subclock divider by connecting a 32.768-kHz crystal resonator, as shown in figure 5.8. Figure 5.9 shows the equivalent circuit of the 32.768-kHz crystal resonator.



$C_1 = C_2 = 15$ pF (typ.)

Note: Capacitances are reference values.

Figure 5.8 Typical Connection to 32.768-kHz Crystal Resonator



$C_O = 1.5$ pF (typ.)
$R_S = 14$ kΩ (typ.)
$f_W = 32.768$ kHz

Note: Constants are reference values.

Figure 5.9 Equivalent Circuit of 32.768-kHz Crystal Resonator

5.2.2 Pin Connection when Not Using Subclock

When the subclock is not used, connect pin X_1 to V_{CL} or V_{SS} and leave pin X_2 open, as shown in figure 5.10.

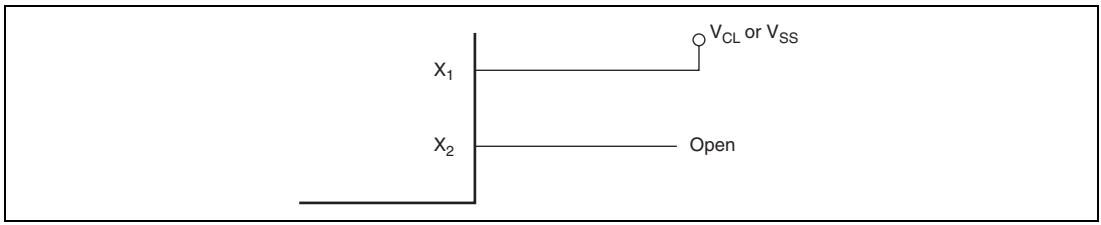


Figure 5.10 Pin Connection when not Using Subclock

5.3 Prescalers

5.3.1 Prescaler S

Prescaler S is a 13-bit counter using the system clock (ϕ) as its input clock. It is incremented once per clock period. Prescaler S is initialized to H'0000 by a reset, and starts counting on exit from the reset state. In standby mode, subactive mode, and subsleep mode, the system clock pulse generator stops. Prescaler S also stops and is initialized to H'0000. The CPU cannot read or write prescaler S. The output from prescaler S is shared by the on-chip peripheral modules. The divider ratio can be set separately for each on-chip peripheral function. In active mode and sleep mode, the clock input to prescaler S is determined by the division factor designated by MA2 to MA0 in SYSCR2.

5.3.2 Prescaler W

Prescaler W is a 5-bit counter using a 32.768 kHz signal divided by 4 ($\phi_w/4$) as its input clock. The divided output is used for clock time base operation of timer A. Prescaler W is initialized to H'00 by a reset, and starts counting on exit from the reset state. Even in standby mode, subactive mode, or subsleep mode, prescaler W continues functioning so long as clock signals are supplied to pins X_1 and X_2. Prescaler W can be reset by setting 1s in bits TMA3 and TMA2 of timer mode register A (TMA).

5.4 Usage Notes

5.4.1 Note on Resonators

Resonator characteristics are closely related to board design and should be carefully evaluated by the user, referring to the examples shown in this section. Resonator circuit constants will differ depending on the resonator element, stray capacitance in its interconnecting circuit, and other factors. Suitable constants should be determined in consultation with the resonator element manufacturer. Design the circuit so that the resonator element never receives voltages exceeding its maximum rating.

5.4.2 Notes on Board Design

When using a crystal resonator (ceramic resonator), place the resonator and its load capacitors as close as possible to the OSC_1 and OSC_2 pins. Other signal lines should be routed away from the resonator circuit to prevent induction from interfering with correct oscillation (see figure 5.11).

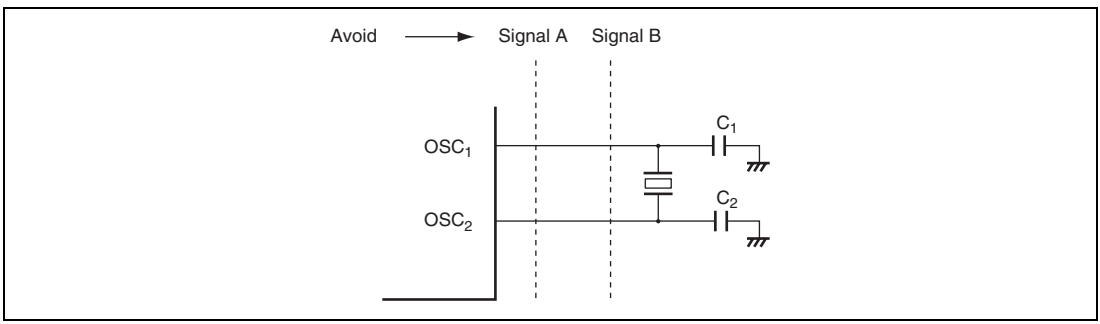


Figure 5.11 Example of Incorrect Board Design

Section 6 Power-Down Modes

This LSI has six modes of operation after a reset. These include a normal active mode and four power-down modes, in which power consumption is significantly reduced. Module standby mode reduces power consumption by selectively halting on-chip module functions.

- Active mode

  The CPU and all on-chip peripheral modules are operable on the system clock. The system clock frequency can be selected from ϕosc, ϕosc/8, ϕosc/16, ϕosc/32, and ϕosc/64.
- Subactive mode

  The CPU and all on-chip peripheral modules are operable on the subclock. The subclock frequency can be selected from ϕw/2, ϕw/4, and ϕw/8.
- Sleep mode

  The CPU halts. On-chip peripheral modules are operable on the system clock.
- Subsleep mode

  The CPU halts. On-chip peripheral modules are operable on the subclock.
- Standby mode

  The CPU and all on-chip peripheral modules halt. When the clock time-base function is selected, timer A is operable.
- Module standby mode

  Independent of the above modes, power consumption can be reduced by halting on-chip peripheral modules that are not used in module units.

6.1 Register Descriptions

The registers related to power-down modes are listed below.

- System control register 1 (SYSCR1)
- System control register 2 (SYSCR2)
- Module standby control register 1 (MSTCR1)

6.1.1 System Control Register 1 (SYSCR1)

SYSCR1 controls the power-down modes, as well as SYSCR2.

Bit	Bit Name	Initial Value	R/W	Description
7	SSBY	0	R/W	Software Standby This bit selects the mode to transit after the execution of the SLEEP instruction. 0: a transition is made to sleep mode or subsleep mode. 1: a transition is made to standby mode. For details, see table 6.2.
6	STS2	0	R/W	Standby Timer Select 2 to 0 These bits designate the time the CPU and peripheral modules wait for stable clock operation after exiting from standby mode, subactive mode, or subsleep mode to active mode or sleep mode due to an interrupt. The designation should be made according to the clock frequency so that the waiting time is at least 6.5 ms. The relationship between the specified value and the number of wait states is shown in table 6.1. When an external clock is to be used, the minimum value (STS2 = STS1 = STS0 = 1) is recommended.
5	STS1	0	R/W	
4	STS0	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
3	NESEL	0	R/W	Noise Elimination Sampling Frequency Select The subclock pulse generator generates the watch clock signal (ϕ_w) and the system clock pulse generator generates the oscillator clock (ϕ_{OSC}). This bit selects the sampling frequency of the oscillator clock when the watch clock signal (ϕ_w) is sampled. When ϕ_{OSC} = 4 to 16 MHz, clear NESEL to 0. 0: Sampling rate is ϕ_{OSC}/16 1: Sampling rate is ϕ_{OSC}/4
2	—	0	—	Reserved
1	—	0	—	These bits are always read as 0.
0	—	0	—	

Table 6.1 Operating Frequency and Waiting Time

STS2	STS1	STS0	Waiting Time	16 MHz	10 MHz	8 MHz	4 MHz	2 MHz	1 MHz	0.5 MHz
0	0	0	8,192 states	0.5	0.8	1.0	2.0	4.1	8.1	16.4
		1	16,384 states	1.0	1.6	2.0	4.1	8.2	16.4	32.8
	1	0	32,768 states	2.0	3.3	4.1	8.2	16.4	32.8	65.5
		1	65,536 states	4.1	6.6	8.2	16.4	32.8	65.5	131.1
1	0	0	131,072 states	8.2	13.1	16.4	32.8	65.5	131.1	262.1
		1	1,024 states	0.06	0.10	0.13	0.26	0.51	1.02	2.05
	1	0	128 states	0.00	0.01	0.02	0.03	0.06	0.13	0.26
		1	16 states	0.00	0.00	0.00	0.00	0.01	0.02	0.03

Note: Time unit is ms.

6.1.2 System Control Register 2 (SYSCR2)

SYSCR2 controls the power-down modes, as well as SYSCR1.

Bit	Bit Name	Initial Value	R/W	Description
7	SMSEL	0	R/W	Sleep Mode Selection
6	LSON	0	R/W	Low Speed on Flag
5	DTON	0	R/W	Direct Transfer on Flag These bits select the mode to transit after the execution of a SLEEP instruction, as well as bit SSBY of SYSCR1. For details, see table 6.2.
4	MA2	0	R/W	Active Mode Clock Select 2 to 0
3	MA1	0	R/W	These bits select the operating clock frequency in active and sleep modes. The operating clock frequency changes to the set frequency after the SLEEP instruction is executed.
2	MA0	0	R/W	0XX: ϕ_{osc} 100: $\phi_{osc}/8$ 101: $\phi_{osc}/16$ 110: $\phi_{osc}/32$ 111: $\phi_{osc}/64$
1	SA1	0	R/W	Subactive Mode Clock Select 1 and 0
0	SA0	0	R/W	These bits select the operating clock frequency in subactive and subsleep modes. The operating clock frequency changes to the set frequency after the SLEEP instruction is executed. 00: $\phi_{w}/8$ 01: $\phi_{w}/4$ 1X: $\phi_{w}/2$

[Legend]

X: Don't care.

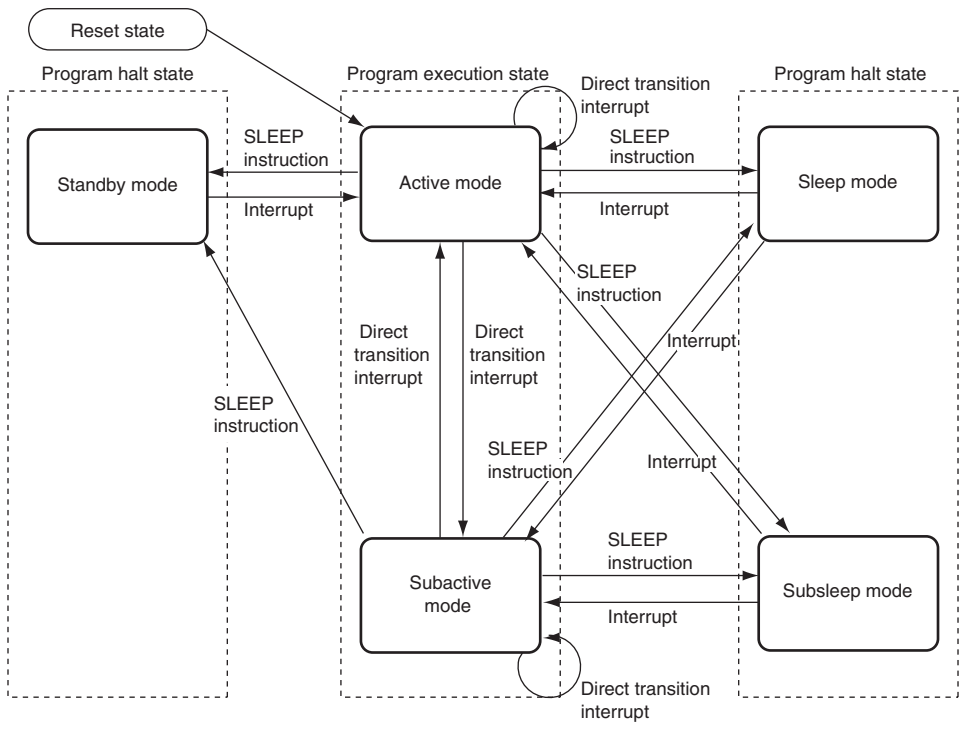
6.1.3 Module Standby Control Register 1 (MSTCR1)

MSTCR1 allows the on-chip peripheral modules to enter a standby state in module units.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved This bit is always read as 0.
6	MSTIIC	0	R/W	IIC Module Standby IIC enters standby mode when this bit is set to 1
5	MSTS3	0	R/W	SCI3 Module Standby SCI3 enters standby mode when this bit is set to 1
4	MSTAD	0	R/W	A/D Converter Module Standby A/D converter enters standby mode when this bit is set to 1
3	MSTWD	0	R/W	Watchdog Timer Module Standby Watchdog timer enters standby mode when this bit is set to 1.When the internal oscillator is selected for the watchdog timer clock, the watchdog timer operates regardless of the setting of this bit
2	MSTTW	0	R/W	Timer W Module Standby Timer W enters standby mode when this bit is set to 1
1	MSTTV	0	R/W	Timer V Module Standby Timer V enters standby mode when this bit is set to 1
0	MSTTA	0	R/W	Timer A Module Standby Timer A enters standby mode when this bit is set to 1

6.2 Mode Transitions and States of LSI

Figure 6.1 shows the possible transitions among these operating modes. A transition is made from the program execution state to the program halt state of the program by executing a SLEEP instruction. Interrupts allow for returning from the program halt state to the program execution state of the program. A direct transition between active mode and subactive mode, which are both program execution states, can be made without halting the program. The operating frequency can also be changed in the same modes by making a transition directly from active mode to active mode, and from subactive mode to subactive mode. $\overline{\text{RES}}$ input enables transitions from a mode to the reset state. Table 6.2 shows the transition conditions of each mode after the SLEEP instruction is executed and a mode to return by an interrupt. Table 6.3 shows the internal states of the LSI in each mode.



Notes: 1. To make a transition to another mode by an interrupt, make sure interrupt handling is after the interrupt is accepted.
2. Details on the mode transition conditions are given in table 6.2.

Figure 6.1 Mode Transition Diagram

Table 6.2 Transition Mode after SLEEP Instruction Execution and Interrupt Handling

DTON	SSBY	SMSEL	LSON	Transition Mode after SLEEP Instruction Execution	Transition Mode due to Interrupt
0	0	0	0	Sleep mode	Active mode
			1		Subactive mode
		1	0	Subsleep mode	Active mode
			1		Subactive mode
	1	X	X	Standby mode	Active mode
1	X	0*	0	Active mode (direct transition)	—
	X	X	1	Subactive mode (direct transition)	—

[Legend]

X: Don't care.

Note: * When a state transition is performed while SMSEL is 1, timer V, SCI3, and the A/D converter are reset, and all registers are set to their initial values. To use these functions after entering active mode, reset the registers.

Table 6.3 Internal State in Each Operating Mode

Function		Active Mode	Sleep Mode	Subactive Mode	Subsleep Mode	Standby Mode
System clock oscillator		Functioning	Functioning	Halted	Halted	Halted
Subclock oscillator		Functioning	Functioning	Functioning	Functioning	Functioning
CPU operations	Instructions	Functioning	Halted	Functioning	Halted	Halted
	Registers	Functioning	Retained	Functioning	Retained	Retained
RAM		Functioning	Retained	Functioning	Retained	Retained
IO ports		Functioning	Retained	Functioning	Retained	Register contents are retained, but output is the high-impedance state.
External interrupts	IRQ3 to IRQ0	Functioning	Functioning	Functioning	Functioning	Functioning
	WKP5 to WKP0	Functioning	Functioning	Functioning	Functioning	Functioning
Peripheral functions	Timer A	Functioning	Functioning	Functioning if the timekeeping time-base function is selected, and retained if not selected		
	Timer V	Functioning	Functioning	Reset	Reset	Reset
	Timer W	Functioning	Functioning	Retained (if internal clock ϕ is selected as a count clock, the counter is incremented by a subclock*)		Retained
	Watchdog timer	Functioning	Functioning	Retained (functioning if the internal oscillator is selected as a count clock*)		
	SCI3	Functioning	Functioning	Reset	Reset	Reset
	IIC	Functioning	Functioning	Retained*	Retained	Retained
	A/D converter	Functioning	Functioning	Reset	Reset	Reset

Note: * Registers can be read or written in subactive mode.

6.2.1 Sleep Mode

In sleep mode, CPU operation is halted but the on-chip peripheral modules function at the clock frequency set by the MA2, MA1, and MA0 bits in SYSCR2. CPU register contents are retained. When an interrupt is requested, sleep mode is cleared and interrupt exception handling starts. Sleep mode is not cleared if the I bit of the condition code register (CCR) is set to 1 or the requested interrupt is disabled in the interrupt enable register. After sleep mode is cleared, a transition is made to active mode when the LSON bit in SYSCR2 is 0, and a transition is made to subactive mode when the bit is 1.

When the $\overline{\text{RES}}$ pin goes low, the CPU goes into the reset state and sleep mode is cleared.

6.2.2 Standby Mode

In standby mode, the clock pulse generator stops, so the CPU and on-chip peripheral modules stop functioning. However, as long as the rated voltage is supplied, the contents of CPU registers, on-chip RAM, and some on-chip peripheral module registers are retained. On-chip RAM contents will be retained as long as the voltage set by the RAM data retention voltage is provided. The I/O ports go to the high-impedance state.

Standby mode is cleared by an interrupt. When an interrupt is requested, the system clock pulse generator starts. After the time set in bits STS2–STS0 in SYSCR1 has elapsed, and interrupt exception handling starts. Standby mode is not cleared if the I bit of CCR is set to 1 or the requested interrupt is disabled in the interrupt enable register.

When the $\overline{\text{RES}}$ pin goes low, the system clock pulse generator starts. Since system clock signals are supplied to the entire chip as soon as the system clock pulse generator starts functioning, the $\overline{\text{RES}}$ pin must be kept low until the pulse generator output stabilizes. After the pulse generator output has stabilized, the CPU starts reset exception handling if the $\overline{\text{RES}}$ pin is driven high.

6.2.3 Subsleep Mode

In subsleep mode, operation of the CPU and on-chip peripheral modules other than timer A is halted. As long as a required voltage is applied, the contents of CPU registers, the on-chip RAM, and some registers of the on-chip peripheral modules are retained. I/O ports keep the same states as before the transition.

Subsleep mode is cleared by an interrupt. When an interrupt is requested, subsleep mode is cleared and interrupt exception handling starts. Subsleep mode is not cleared if the I bit of CCR is set to 1 or the requested interrupt is disabled in the interrupt enable register. After subsleep mode is cleared, a transition is made to active mode when the LSON bit in SYSCR2 is 0, and a transition is made to subactive mode when the bit is 1.

When the $\overline{\text{RES}}$ pin goes low, the system clock pulse generator starts. Since system clock signals are supplied to the entire chip as soon as the system clock pulse generator starts functioning, the $\overline{\text{RES}}$ pin must be kept low until the pulse generator output stabilizes. After the pulse generator output has stabilized, the CPU starts reset exception handling if the $\overline{\text{RES}}$ pin is driven high.

6.2.4 Subactive Mode

The operating frequency of subactive mode is selected from $\phi_w/2$, $\phi_w/4$, and $\phi_w/8$ by the SA1 and SA0 bits in SYSCR2. After the SLEEP instruction is executed, the operating frequency changes to the frequency which is set before the execution. When the SLEEP instruction is executed in subactive mode, a transition to sleep mode, subsleep mode, standby mode, active mode, or subactive mode is made, depending on the combination of SYSCR1 and SYSCR2. When the $\overline{\text{RES}}$ pin goes low, the system clock pulse generator starts. Since system clock signals are supplied to the entire chip as soon as the system clock pulse generator starts functioning, the $\overline{\text{RES}}$ pin must be kept low until the pulse generator output stabilizes. After the pulse generator output has stabilized, the CPU starts reset exception handling if the $\overline{\text{RES}}$ pin is driven high.

6.3 Operating Frequency in Active Mode

Operation in active mode is clocked at the frequency designated by the MA2, MA1, and MA0 bits in SYSCR2. The operating frequency changes to the set frequency after SLEEP instruction execution.

6.4 Direct Transition

The CPU can execute programs in two modes: active and subactive mode. A direct transition is a transition between these two modes without stopping program execution. A direct transition can be made by executing a SLEEP instruction while the DTON bit in SYSCR2 is set to 1. The direct transition also enables operating frequency modification in active or subactive mode. After the mode transition, direct transition interrupt exception handling starts.

If the direct transition interrupt is disabled in interrupt enable register 1, a transition is made instead to sleep or subsleep mode. Note that if a direct transition is attempted while the I bit in CCR is set to 1, sleep or subsleep mode will be entered, and the resulting mode cannot be cleared by means of an interrupt.

6.4.1 Direct Transition from Active Mode to Subactive Mode

The time from the start of SLEEP instruction execution to the end of interrupt exception handling (the direct transition time) is calculated by equation (1).

Direct transition time = {(number of SLEEP instruction execution states) +
(number of internal processing states)}× (tcyc before transition) +
(number of interrupt exception handling states) × (tsubcyc after transition).……(1)

Example:
Direct transition time = (2 + 1) × tosc + 14 × 8tw = 3tosc + 112tw
(when the CPU operating clock of $\phi_{osc} \rightarrow \phi_w/8$ is selected)

[Legend]
tosc: OSC clock cycle time
tw: watch clock cycle time
tcyc: system clock (ϕ) cycle time
tsubcyc: subclock (ϕ_{SUB}) cycle time

6.4.2 Direct Transition from Subactive Mode to Active Mode

The time from the start of SLEEP instruction execution to the end of interrupt exception handling (the direct transition time) is calculated by equation (2).

Direct transition time = {(number of SLEEP instruction execution states) + (number of internal processing states)} × (tsubcyc before transition) + {(waiting time set in bits STS2 to STS0) + (number of interrupt exception handling states)} × (tcyc after transition)……(2)

Example
Direct transition time = (2 + 1) × 8 tw + (8192 + 14) × tosc = 24tw + 8206 tosc
(when the CPU operating clock of $\phi_w/8 \rightarrow \phi_{osc}$ and a waiting time of 8192 states are selected)

[Legend]
tosc: OSC clock cycle time
tw: watch clock cycle time
tcyc: system clock (ϕ) cycle time
tsubcyc: subclock (ϕ_{SUB}) cycle time

6.5 Module Standby Function

The module-standby function can be set to any peripheral module. In module standby mode, the clock supply to modules stops to enter the power-down mode. Module standby mode enables each on-chip peripheral module to enter the standby state by setting a bit that corresponds to each module to 1 and cancels the mode by clearing the bit to 0.

6.6 Usage Note

When subsleep mode is entered by setting the SMSEL bit to 1 while the subclock is not used (the X_1 pin is fixed), note that active mode cannot be re-entered by using an interrupt. To use a power-down mode while a port is retained, connect the subclock to the X_1 and X_2 pins.

Section 7 ROM

The features of the 32-kbyte flash memory built into the flash memory version are summarized below.

- Programming/erase methods
  - The flash memory is programmed 128 bytes at a time. Erase is performed in single-block units. The flash memory is configured as follows: 1 kbyte × 4 blocks and 28 kbytes × 1 block. To erase the entire flash memory, each block must be erased in turn.
- Reprogramming capability
  - The flash memory can be reprogrammed up to 1,000 times.
- On-board programming
  - On-board programming/erasing can be done in boot mode, in which the boot program built into the chip is started to erase or program of the entire flash memory. In normal user program mode, individual blocks can be erased or programmed.
- Programmer mode
  - Flash memory can be programmed/erased in programmer mode using a PROM programmer, as well as in on-board programming mode.
- Automatic bit rate adjustment
  - For data transfer in boot mode, this LSI's bit rate can be automatically adjusted to match the transfer bit rate of the host.
- Programming/erasing protection
  - Sets software protection against flash memory programming/erasing.
- Power-down mode
  - Operation of the power supply circuit can be partly halted in subactive mode. As a result, flash memory can be read with low power consumption.

7.1 Block Configuration

Figure 7.1 shows the block configuration of 32-kbyte flash memory. The thick lines indicate erasing units, the narrow lines indicate programming units, and the values are addresses. The flash memory is divided into 1 kbyte × 4 blocks and 28 kbytes × 1 block. Erasing is performed in these units. Programming is performed in 128-byte units starting from an address with lower eight bits H'00 or H'80.

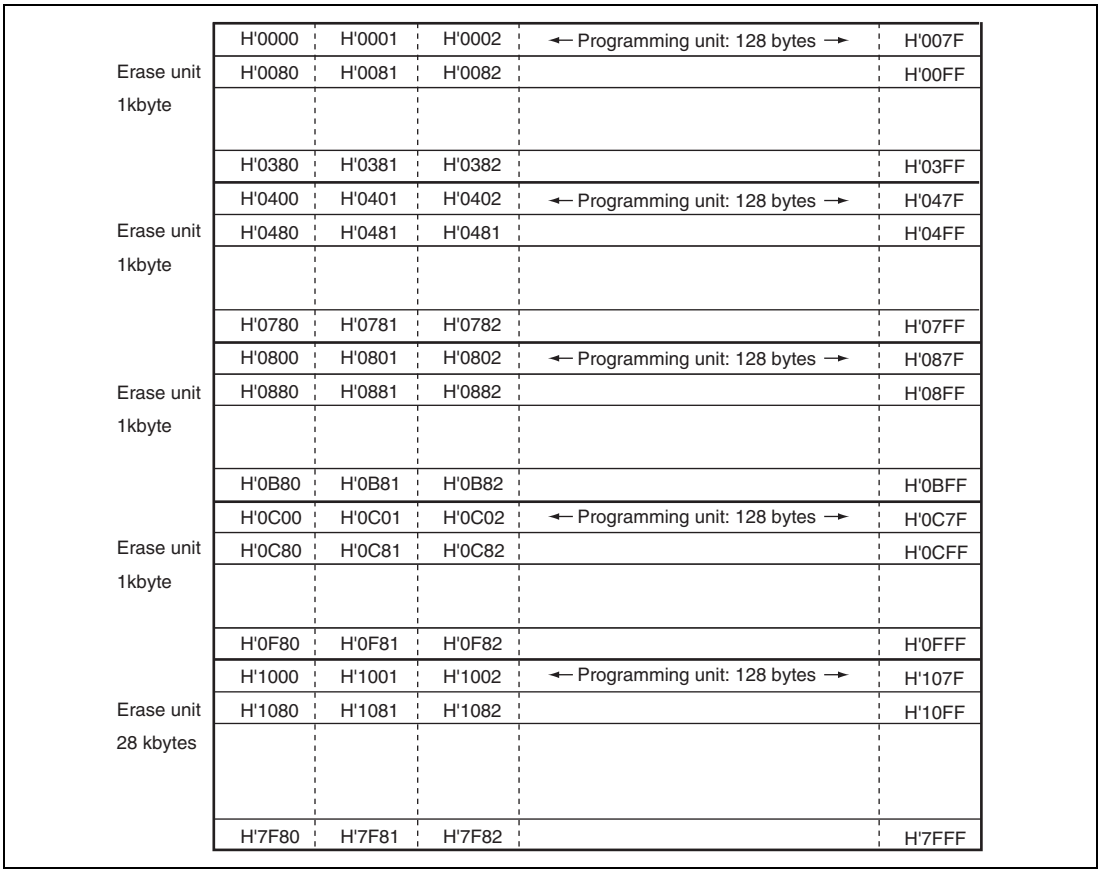


Figure 7.1 Flash Memory Block Configuration

7.2 Register Descriptions

The flash memory has the following registers.

- Flash memory control register 1 (FLMCR1)
- Flash memory control register 2 (FLMCR2)
- Erase block register 1 (EBR1)
- Flash memory power control register (FLPWCR)
- Flash memory enable register (FENR)

7.2.1 Flash Memory Control Register 1 (FLMCR1)

FLMCR1 is a register that makes the flash memory change to program mode, program-verify mode, erase mode, or erase-verify mode. For details on register setting, refer to section 7.4, Flash Memory Programming/Erasing.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved This bit is always read as 0.
6	SWE	0	R/W	Software Write Enable When this bit is set to 1, flash memory programming/erasing is enabled. When this bit is cleared to 0, other FLMCR1 register bits and all EBR1 bits cannot be set.
5	ESU	0	R/W	Erase Setup When this bit is set to 1, the flash memory changes to the erase setup state. When it is cleared to 0, the erase setup state is cancelled. Set this bit to 1 before setting the E bit to 1 in FLMCR1.
4	PSU	0	R/W	Program Setup When this bit is set to 1, the flash memory changes to the program setup state. When it is cleared to 0, the program setup state is cancelled. Set this bit to 1 before setting the P bit in FLMCR1.

Bit	Bit Name	Initial Value	R/W	Description
3	EV	0	R/W	Erase-Verify When this bit is set to 1, the flash memory changes to erase-verify mode. When it is cleared to 0, erase-verify mode is cancelled.
2	PV	0	R/W	Program-Verify When this bit is set to 1, the flash memory changes to program-verify mode. When it is cleared to 0, program-verify mode is cancelled.
1	E	0	R/W	Erase When this bit is set to 1, and while the SWE=1 and ESU=1 bits are 1, the flash memory changes to erase mode. When it is cleared to 0, erase mode is cancelled.
0	P	0	R/W	Program When this bit is set to 1, and while the SWE=1 and PSU=1 bits are 1, the flash memory changes to program mode. When it is cleared to 0, program mode is cancelled.

7.2.2 Flash Memory Control Register 2 (FLMCR2)

FLMCR2 is a register that displays the state of flash memory programming/erasing. FLMCR2 is a read-only register, and should not be written to.

Bit	Bit Name	Initial Value	R/W	Description
7	FLER	0	R	Flash Memory Error Indicates that an error has occurred during an operation on flash memory (programming or erasing). When FLER is set to 1, flash memory goes to the error-protection state. See section 7.5.3, Error Protection, for details.
6 to 0	—	All 0	—	Reserved These bits are always read as 0.

7.2.3 Erase Block Register 1 (EBR1)

EBR1 specifies the flash memory erase area block. EBR1 is initialized to H'00 when the SWE bit in FLMCR1 is 0. Do not set more than one bit at a time, as this will cause all the bits in EBR1 to be automatically cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	—	Reserved These bits are always read as 0.
4	EB4	0	R/W	When this bit is set to 1, 28 kbytes of H'1000 to H'7FFF will be erased.
3	EB3	0	R/W	When this bit is set to 1, 1 kbyte of H'0C00 to H'0FFF will be erased.
2	EB2	0	R/W	When this bit is set to 1, 1 kbyte of H'0800 to H'0BFF will be erased.
1	EB1	0	R/W	When this bit is set to 1, 1 kbyte of H'0400 to H'07FF will be erased.
0	EB0	0	R/W	When this bit is set to 1, 1 kbyte of H'0000 to H'03FF will be erased.

7.2.4 Flash Memory Power Control Register (FLPWCR)

FLPWCR enables or disables a transition to the flash memory power-down mode when the LSI switches to subactive mode. There are two modes: mode in which operation of the power supply circuit of flash memory is partly halted in power-down mode and flash memory can be read, and mode in which even if a transition is made to subactive mode, operation of the power supply circuit of flash memory is retained and flash memory can be read.

Bit	Bit Name	Initial Value	R/W	Description
7	PDWND	0	R/W	Power-Down Disable When this bit is 0 and a transition is made to subactive mode, the flash memory enters the power-down mode. When this bit is 1, the flash memory remains in the normal mode even after a transition is made to subactive mode.
6 to 0	—	All 0	—	Reserved These bits are always read as 0.

7.2.5 Flash Memory Enable Register (FENR)

Bit 7 (FLSHE) in FENR enables or disables the CPU access to the flash memory control registers, FLMCR1, FLMCR2, EBR1, and FLPWCR.

Bit	Bit Name	Initial Value	R/W	Description
7	FLSHE	0	R/W	Flash Memory Control Register Enable Flash memory control registers can be accessed when this bit is set to 1. Flash memory control registers cannot be accessed when this bit is set to 0.
6 to 0	—	All 0	—	Reserved These bits are always read as 0.

7.3 On-Board Programming Modes

There are two modes for programming/erasing of the flash memory; boot mode, which enables on-board programming/erasing, and programmer mode, in which programming/erasing is performed with a PROM programmer. On-board programming/erasing can also be performed in user program mode. At reset-start in reset mode, the series of HD64F3664 changes to a mode depending on the TEST pin settings, $\overline{\text{NMI}}$ pin settings, and input level of each port, as shown in table 7.1. The input level of each pin must be defined four states before the reset ends.

When changing to boot mode, the boot program built into this LSI is initiated. The boot program transfers the programming control program from the externally-connected host to on-chip RAM via SCI3. After erasing the entire flash memory, the programming control program is executed. This can be used for programming initial values in the on-board state or for a forcible return when programming/erasing can no longer be done in user program mode. In user program mode, individual blocks can be erased and programmed by branching to the user program/erase control program prepared by the user.

Table 7.1 Setting Programming Modes

TEST	$\overline{\text{NMI}}$	P85	PB0	PB1	PB2	LSI State after Reset End
0	1	X	X	X	X	User Mode
0	0	1	X	X	X	Boot Mode
1	X	X	0	0	0	Programmer Mode

[Legend]

X: Don't care.

7.3.1 Boot Mode

Table 7.2 shows the boot mode operations between reset end and branching to the programming control program.

1. When boot mode is used, the flash memory programming control program must be prepared in the host beforehand. Prepare a programming control program in accordance with the description in section 7.4, Flash Memory Programming/Erasing.
2. SCI3 should be set to asynchronous mode, and the transfer format as follows: 8-bit data, 1 stop bit, and no parity.
3. When the boot program is initiated, the chip measures the low-level period of asynchronous SCI communication data (H'00) transmitted continuously from the host. The chip then calculates the bit rate of transmission from the host, and adjusts the SCI3 bit rate to match that of the host. The reset should end with the RxD pin high. The RxD and TxD pins should be pulled up on the board if necessary. After the reset is complete, it takes approximately 100 states before the chip is ready to measure the low-level period.
4. After matching the bit rates, the chip transmits one H'00 byte to the host to indicate the completion of bit rate adjustment. The host should confirm that this adjustment end indication (H'00) has been received normally, and transmit one H'55 byte to the chip. If reception could not be performed normally, initiate boot mode again by a reset. Depending on the host's transfer bit rate and system clock frequency of this LSI, there will be a discrepancy between the bit rates of the host and the chip. To operate the SCI properly, set the host's transfer bit rate and system clock frequency of this LSI within the ranges listed in table 7.3.
5. In boot mode, a part of the on-chip RAM area is used by the boot program. The area H'F780 to H'FEEF is the area to which the programming control program is transferred from the host. The boot program area cannot be used until the execution state in boot mode switches to the programming control program.
6. Before branching to the programming control program, the chip terminates transfer operations by SCI3 (by clearing the RE and TE bits in SCR to 0), however the adjusted bit rate value remains set in BRR. Therefore, the programming control program can still use it for transfer of write data or verify data with the host. The TxD pin is high (PCR22 = 1, P22 = 1). The contents of the CPU general registers are undefined immediately after branching to the programming control program. These registers must be initialized at the beginning of the programming control program, as the stack pointer (SP), in particular, is used implicitly in subroutine calls, etc.
7. Boot mode can be cleared by a reset. End the reset after driving the reset pin low, waiting at least 20 states, and then setting the TEST pin and $\overline{\text{NMI}}$ pin. Boot mode is also cleared when a WDT overflow occurs.
8. Do not change the TEST pin and $\overline{\text{NMI}}$ pin input levels in boot mode.

Table 7.2 Boot Mode Operation

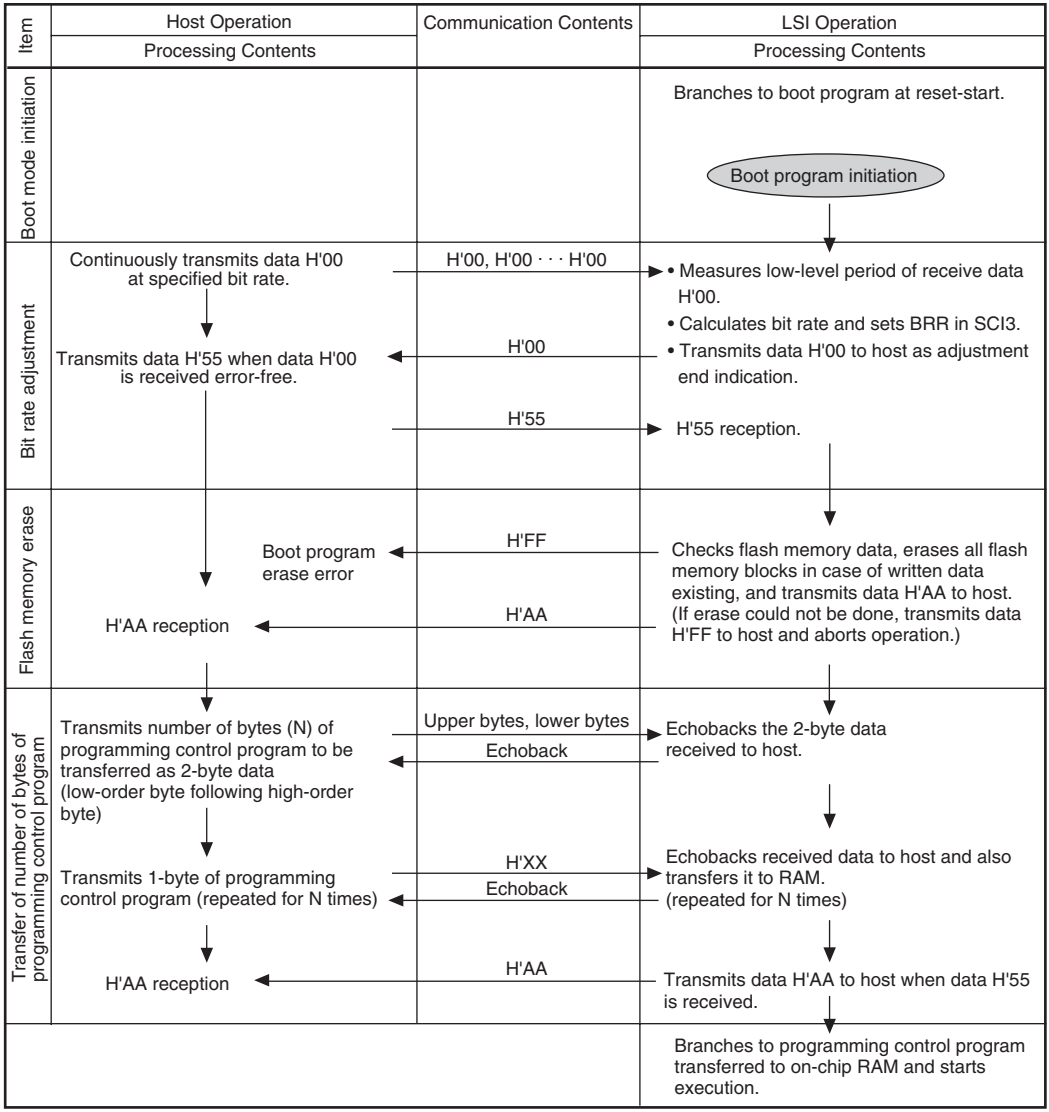


Table 7.3 System Clock Frequencies for which Automatic Adjustment of LSI Bit Rate is Possible

Host Bit Rate	System Clock Frequency Range of LSI
19,200 bps	16 MHz
9,600 bps	8 to 16 MHz
4,800 bps	4 to 16 MHz
2,400 bps	2 to 16 MHz

7.3.2 Programming/Erasing in User Program Mode

On-board programming/erasing of an individual flash memory block can also be performed in user program mode by branching to a user program/erase control program. The user must set branching conditions and provide on-board means of supplying programming data. The flash memory must contain the user program/erase control program or a program that provides the user program/erase control program from external memory. As the flash memory itself cannot be read during programming/erasing, transfer the user program/erase control program to on-chip RAM, as in boot mode. Figure 7.2 shows a sample procedure for programming/erasing in user program mode. Prepare a user program/erase control program in accordance with the description in section 7.4, Flash Memory Programming/Erasing.

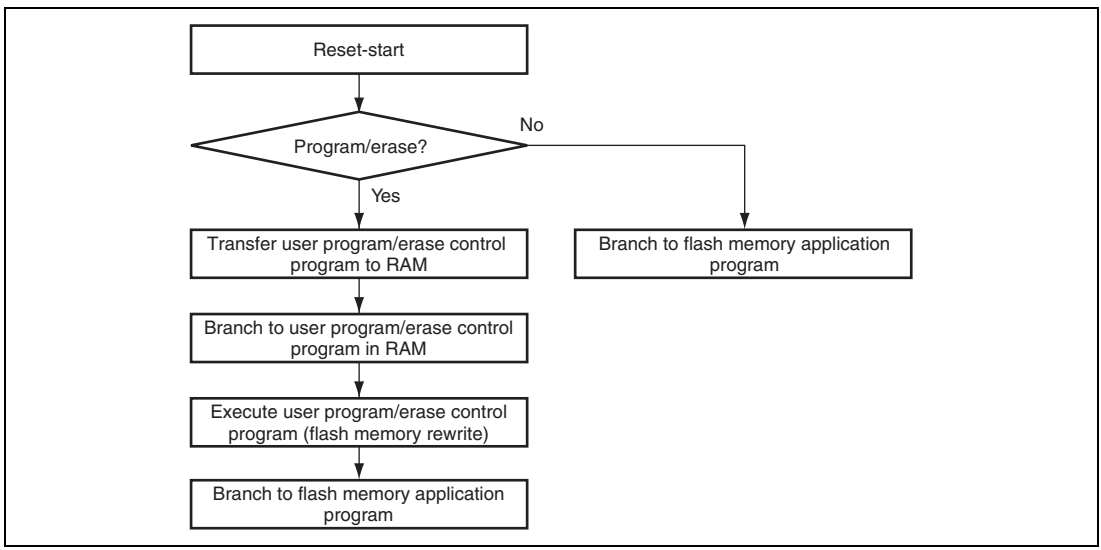


Figure 7.2 Programming/Erasing Flowchart Example in User Program Mode

7.4 Flash Memory Programming/Erasing

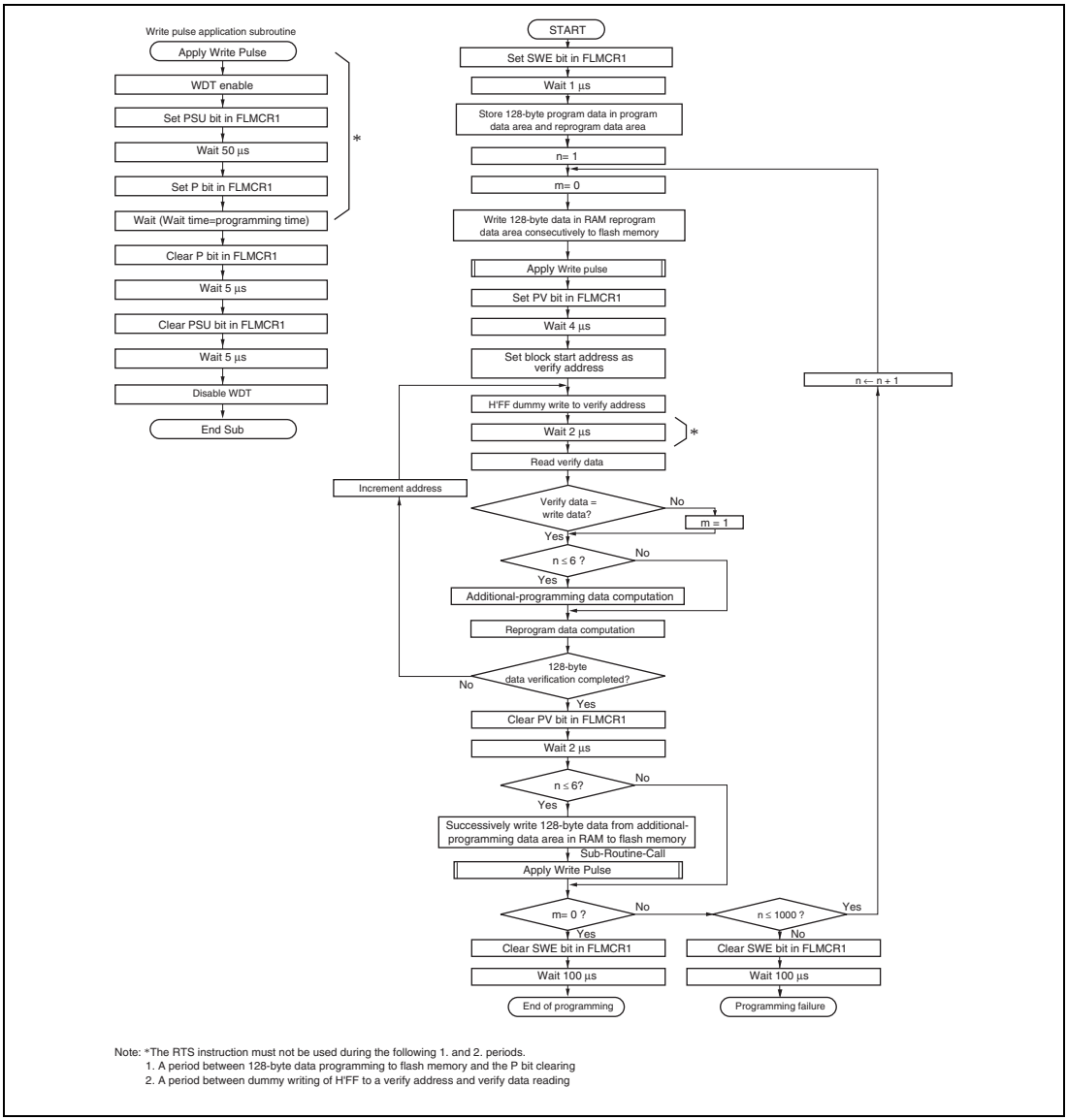
A software method using the CPU is employed to program and erase flash memory in the on-board programming modes. Depending on the FLMCR1 setting, the flash memory operates in one of the following four modes: Program mode, program-verify mode, erase mode, and erase-verify mode. The programming control program in boot mode and the user program/erase control program in user program mode use these operating modes in combination to perform programming/erasing. Flash memory programming and erasing should be performed in accordance with the descriptions in section 7.4.1, Program/Program-Verify and section 7.4.2, Erase/Erase-Verify, respectively.

7.4.1 Program/Program-Verify

When writing data or programs to the flash memory, the program/program-verify flowchart shown in figure 7.3 should be followed. Performing programming operations according to this flowchart will enable data or programs to be written to the flash memory without subjecting the chip to voltage stress or sacrificing program data reliability.

1. Programming must be done to an empty address. Do not reprogram an address to which programming has already been performed.
2. Programming should be carried out 128 bytes at a time. A 128-byte data transfer must be performed even if writing fewer than 128 bytes. In this case, H'FF data must be written to the extra addresses.
3. Prepare the following data storage areas in RAM: A 128-byte programming data area, a 128-byte reprogramming data area, and a 128-byte additional-programming data area. Perform reprogramming data computation according to table 7.4, and additional programming data computation according to table 7.5.
4. Consecutively transfer 128 bytes of data in byte units from the reprogramming data area or additional-programming data area to the flash memory. The program address and 128-byte data are latched in the flash memory. The lower 8 bits of the start address in the flash memory destination area must be H'00 or H'80.
5. The time during which the P bit is set to 1 is the programming time. Table 7.6 shows the allowable programming times.
6. The watchdog timer (WDT) is set to prevent overprogramming due to program runaway, etc. An overflow cycle of approximately 6.6 ms is allowed.
7. For a dummy write to a verify address, write 1-byte data H'FF to an address whose lower 2 bits are B'00. Verify data can be read in words or in longwords from the address to which a dummy write was performed.

8. The maximum number of repetitions of the program/program-verify sequence of the same bit is 1,000.



Note: *The RTS instruction must not be used during the following 1. and 2. periods.
1. A period between 128-byte data programming to flash memory and the P bit clearing
2. A period between dummy writing of H'FF to a verify address and verify data reading

Figure 7.3 Program/Program-Verify Flowchart

Table 7.4 Reprogram Data Computation Table

Program Data	Verify Data	Reprogram Data	Comments
0	0	1	Programming completed
0	1	0	Reprogram bit
1	0	1	—
1	1	1	Remains in erased state

Table 7.5 Additional-Program Data Computation Table

Reprogram Data	Verify Data	Additional-Program Data	Comments
0	0	0	Additional-program bit
0	1	1	No additional programming
1	0	1	No additional programming
1	1	1	No additional programming

Table 7.6 Programming Time

n (Number of Writes)	Programming Time	In Additional Programming	Comments
1 to 6	30	10	
7 to 1,000	200	—	

Note: Time shown in μs.

7.4.2 Erase/Erase-Verify

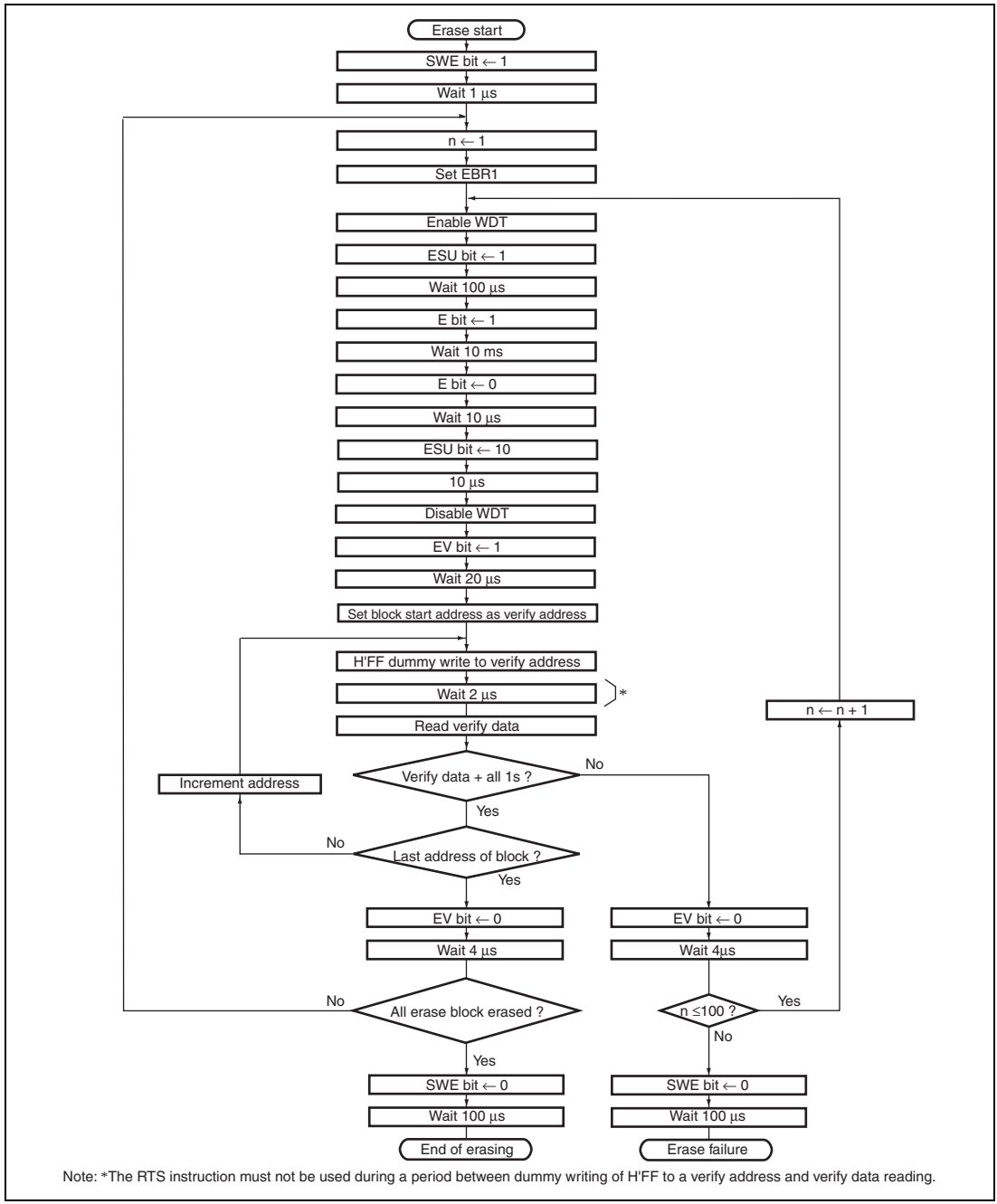
When erasing flash memory, the erase/erase-verify flowchart shown in figure 7.4 should be followed.

1. Prewriting (setting erase block data to all 0s) is not necessary.
2. Erasing is performed in block units. Make only a single-bit specification in the erase block register (EBR1). To erase multiple blocks, each block must be erased in turn.
3. The time during which the E bit is set to 1 is the flash memory erase time.
4. The watchdog timer (WDT) is set to prevent overerasing due to program runaway, etc. An overflow cycle of approximately 19.8 ms is allowed.
5. For a dummy write to a verify address, write 1-byte data H'FF to an address whose lower two bits are B'00. Verify data can be read in longwords from the address to which a dummy write was performed.
6. If the read data is not erased successfully, set erase mode again, and repeat the erase/erase-verify sequence as before. The maximum number of repetitions of the erase/erase-verify sequence is 100.

7.4.3 Interrupt Handling when Programming/Erasing Flash Memory

All interrupts, including the NMI interrupt, are disabled while flash memory is being programmed or erased, or while the boot program is executing, for the following three reasons:

1. Interrupt during programming/erasing may cause a violation of the programming or erasing algorithm, with the result that normal operation cannot be assured.
2. If interrupt exception handling starts before the vector address is written or during programming/erasing, a correct vector cannot be fetched and the CPU malfunctions.
3. If an interrupt occurs during boot program execution, normal boot mode sequence cannot be carried out.



Note: *The RTS instruction must not be used during a period between dummy writing of H'FF to a verify address and verify data reading.

Figure 7.4 Erase/Erase-Verify Flowchart

7.5 Program/Erase Protection

There are three kinds of flash memory program/erase protection; hardware protection, software protection, and error protection.

7.5.1 Hardware Protection

Hardware protection refers to a state in which programming/erasing of flash memory is forcibly disabled or aborted because of a transition to reset, subactive mode, subsleep mode, or standby mode. Flash memory control register 1 (FLMCR1), flash memory control register 2 (FLMCR2), and erase block register 1 (EBR1) are initialized. In a reset via the $\overline{\text{RES}}$ pin, the reset state is not entered unless the $\overline{\text{RES}}$ pin is held low until oscillation stabilizes after powering on. In the case of a reset during operation, hold the $\overline{\text{RES}}$ pin low for the $\overline{\text{RES}}$ pulse width specified in the AC Characteristics section.

7.5.2 Software Protection

Software protection can be implemented against programming/erasing of all flash memory blocks by clearing the SWE bit in FLMCR1. When software protection is in effect, setting the P or E bit in FLMCR1 does not cause a transition to program mode or erase mode. By setting the erase block register 1 (EBR1), erase protection can be set for individual blocks. When EBR1 is set to H'00, erase protection is set for all blocks.

7.5.3 Error Protection

In error protection, an error is detected when CPU runaway occurs during flash memory programming/erasing, or operation is not performed in accordance with the program/erase algorithm, and the program/erase operation is aborted. Aborting the program/erase operation prevents damage to the flash memory due to overprogramming or overerasing.

When the following errors are detected during programming/erasing of flash memory, the FLER bit in FLMCR2 is set to 1, and the error protection state is entered.

- When the flash memory of the relevant address area is read during programming/erasing (including vector read and instruction fetch)
- Immediately after exception handling excluding a reset during programming/erasing
- When a SLEEP instruction is executed during programming/erasing

The FLMCR1, FLMCR2, and EBR1 settings are retained, however program mode or erase mode is aborted at the point at which the error occurred. Program mode or erase mode cannot be re-entered by re-setting the P or E bit. However, PV and EV bit setting is enabled, and a transition can be made to verify mode. Error protection can be cleared only by a power-on reset.

7.6 Programmer Mode

In programmer mode, a PROM programmer can be used to perform programming/erasing via a socket adapter, just as a discrete flash memory. Use a PROM programmer that supports the MCU device type with the on-chip 64-kbyte flash memory (FZTAT64V5).

7.7 Power-Down States for Flash Memory

In user mode, the flash memory will operate in either of the following states:

- Normal operating mode
  The flash memory can be read and written to at high speed.
- Power-down operating mode
  The power supply circuit of flash memory can be partly halted. As a result, flash memory can be read with low power consumption.
- Standby mode
  All flash memory circuits are halted.

Table 7.7 shows the correspondence between the operating modes of this LSI and the flash memory. In subactive mode, the flash memory can be set to operate in power-down mode with the PDWND bit in FLPWCR. When the flash memory returns to its normal operating state from power-down mode or standby mode, a period to stabilize operation of the power supply circuits that were stopped is needed. When the flash memory returns to its normal operating state, bits STS2 to STS0 in SYSCR1 must be set to provide a wait time of at least 20 µs, even when the external clock is being used.

Table 7.7 Flash Memory Operating States

	Flash Memory Operating State	
LSI Operating State	**PDWND = 0 (Initial value)**	**PDWND = 1**
Active mode	Normal operating mode	Normal operating mode
Subactive mode	Power-down mode	Normal operating mode
Sleep mode	Normal operating mode	Normal operating mode
Subsleep mode	Standby mode	Standby mode
Standby mode	Standby mode	Standby mode

Section 8 RAM

This LSI has an on-chip high-speed static RAM. The RAM is connected to the CPU by a 16-bit data bus, enabling two-state access by the CPU to both byte data and word data.

Product Classification		RAM Size	RAM Address
Flash memory version (F-ZTAT™ version)	H8/3664N	2 kbytes	H'F780 to H'FF7F*
	H8/3664F	2 kbytes	H'F780 to H'FF7F*
Mask ROM version	H8/3664	1 kbyte	H'FB80 to H'FF7F
	H8/3663	1 kbyte	H'FB80 to H'FF7F
	H8/3662	512 bytes	H'FD80 to H'FF7F
	H8/3661	512 bytes	H'FD80 to H'FF7F
	H8/3660	512 bytes	H'FD80 to H'FF7F

Note: * Area H'F780 to H'FB7F must not be accessed.

Section 9 I/O Ports

The group of this LSI has twenty-nine general I/O ports (twenty-seven ports for H8/3664N) and eight general input-only ports. Port 8 is a large current port, which can drive 20 mA ($@V_{OL}$ = 1.5 V) when a low level signal is output. Any of these ports can become an input port immediately after a reset. They can also be used as I/O pins of the on-chip peripheral modules or external interrupt input pins, and these functions can be switched depending on the register settings. The registers for selecting these functions can be divided into two types: those included in I/O ports and those included in each on-chip peripheral module. General I/O ports are comprised of the port control register for controlling inputs/outputs and the port data register for storing output data and can select inputs/outputs in bit units. For functions in each port, see appendix B.1, I/O Port Block. For the execution of bit manipulation instructions to the port control register and port data register, see section 2.8.3, Bit Manipulation Instruction.

9.1 Port 1

Port 1 is a general I/O port also functioning as IRQ interrupt input pins, a timer A output pin, and a timer V input pin. Figure 9.1 shows its pin configuration.

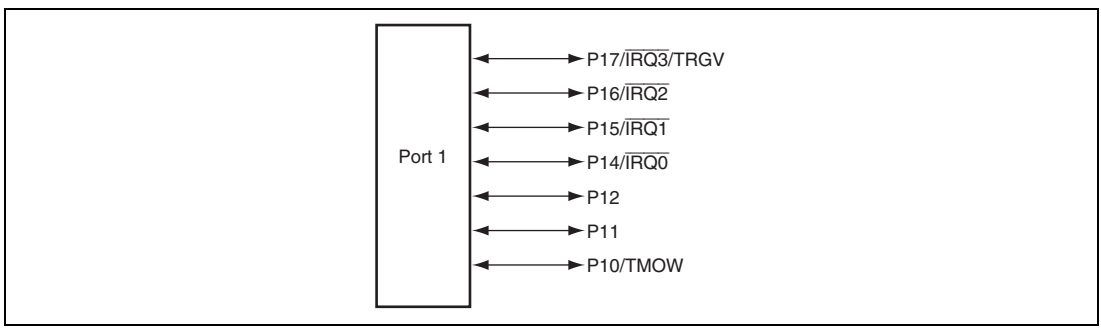


Figure 9.1 Port 1 Pin Configuration

Port 1 has the following registers.

- Port mode register 1 (PMR1)
- Port control register 1 (PCR1)
- Port data register 1 (PDR1)
- Port pull-up control register 1 (PUCR1)

9.1.1 Port Mode Register 1 (PMR1)

PMR1 switches the functions of pins in port 1 and port 2.

Bit	Bit Name	Initial Value	R/W	Description
7	IRQ3	0	R/W	P17/$\overline{\text{IRQ3}}$/TRGV Pin Function Switch This bit selects whether pin P17/$\overline{\text{IRQ3}}$/TRGV is used as P17 or as $\overline{\text{IRQ3}}$/TRGV. 0: General I/O port 1: $\overline{\text{IRQ3}}$/TRGV input pin
6	IRQ2	0	R/W	P16/$\overline{\text{IRQ2}}$ Pin Function Switch This bit selects whether pin P16/$\overline{\text{IRQ2}}$ is used as P16 or as $\overline{\text{IRQ2}}$. 0: General I/O port 1: $\overline{\text{IRQ2}}$ input pin
5	IRQ1	0	R/W	P15/$\overline{\text{IRQ1}}$ Pin Function Switch This bit selects whether pin P15/$\overline{\text{IRQ1}}$ is used as P15 or as $\overline{\text{IRQ1}}$. 0: General I/O port 1: $\overline{\text{IRQ1}}$ input pin
4	IRQ0	0	R/W	P14/$\overline{\text{IRQ0}}$ Pin Function Switch This bit selects whether pin P14/$\overline{\text{IRQ0}}$ is used as P14 or as $\overline{\text{IRQ0}}$. 0: General I/O port 1: $\overline{\text{IRQ0}}$ input pin
3	—	1	—	Reserved
2	—	1	—	These bits are always read as 1.
1	TXD	0	R/W	P22/TXD Pin Function Switch This bit selects whether pin P22/TXD is used as P22 or as TXD. 0: General I/O port 1: TXD output pin

Bit	Bit Name	Initial Value	R/W	Description
0	TMOW	0	R/W	P10/TMOW Pin Function Switch This bit selects whether pin P10/TMOW is used as P10 or as TMOW. 0: General I/O port 1: TMOW output pin

9.1.2 Port Control Register 1 (PCR1)

PCR1 selects inputs/outputs in bit units for pins to be used as general I/O ports of port 1.

Bit	Bit Name	Initial Value	R/W	Description
7	PCR17	0	W	When the corresponding pin is designated in PMR1 as a general I/O pin, setting a PCR1 bit to 1 makes the corresponding pin an output port, while clearing the bit to 0 makes the pin an input port. Bit 3 is a reserved bit.
6	PCR16	0	W	
5	PCR15	0	W	
4	PCR14	0	W	
3	—	—	—	
2	PCR12	0	W	
1	PCR11	0	W	
0	PCR10	0	W	

9.1.3 Port Data Register 1 (PDR1)

PDR1 is a general I/O port data register of port 1.

Bit	Bit Name	Initial Value	R/W	Description
7	P17	0	R/W	PDR1 stores output data for port 1 pins.
6	P16	0	R/W	If PDR1 is read while PCR1 bits are set to 1, the value stored in PDR1 are read. If PDR1 is read while PCR1 bits are cleared to 0, the pin states are read regardless of the value stored in PDR1.
5	P15	0	R/W	
4	P14	0	R/W	
3	—	1	—	Bit 3 is a reserved bit. This bit is always read as 1.
2	P12	0	R/W	
1	P11	0	R/W	
0	P10	0	R/W	

9.1.4 Port Pull-Up Control Register 1 (PUCR1)

PUCR1 controls the pull-up MOS in bit units of the pins set as the input ports.

Bit	Bit Name	Initial Value	R/W	Description
7	PUCR17	0	R/W	Only bits for which PCR1 is cleared are valid. The pull-up MOS of P17 to P14 and P12 to P10 pins enter the on-state when these bits are set to 1, while they enter the off-state when these bits are cleared to 0.
6	PUCR16	0	R/W	
5	PUCR15	0	R/W	
4	PUCR14	0	R/W	Bit 3 is a reserved bit. This bit is always read as 1.
3	—	1	—	
2	PUCR12	0	R/W	
1	PUCR11	0	R/W	
0	PUCR10	0	R/W	

9.1.5 Pin Functions

The correspondence between the register specification and the port functions is shown below.

- P17/$\overline{\text{IRQ3}}$/TRGV Pin

Register	PMR1	PCR1	
Bit Name	**IRQ3**	**PCR17**	**Pin Function**
Setting value	0	0	P17 input pin
		1	P17 output pin
	1	X	$\overline{\text{IRQ3}}$ input/TRGV input pin

[Legend]
X: Don't care.

- P16/$\overline{\text{IRQ2}}$ Pin

Register	PMR1	PCR1	
Bit Name	**IRQ2**	**PCR16**	**Pin Function**
Setting value	0	0	P16 input pin
		1	P16 output pin
	1	X	$\overline{\text{IRQ2}}$ input pin

[Legend]
X: Don't care.

- P15/$\overline{\text{IRQ1}}$ Pin

Register	PMR1	PCR1	
Bit Name	**IRQ1**	**PCR15**	**Pin Function**
Setting value	0	0	P15 input pin
		1	P15 output pin
	1	X	$\overline{\text{IRQ1}}$ input pin

[Legend]
X: Don't care.

- P14/$\overline{\text{IRQ0}}$ Pin

Register	PMR1	PCR1	
Bit Name	**IRQ0**	**PCR14**	**Pin Function**
Setting value	0	0	P14 input pin
		1	P14 output pin
	1	X	$\overline{\text{IRQ0}}$ input pin

[Legend]
X: Don't care.

- P12 Pin

Register	PCR1	
Bit Name	**PCR12**	**Pin Function**
Setting value	0	P12 input pin
	1	P12 output pin

- P11 Pin

Register	PCR1	
Bit Name	**PCR11**	**Pin Function**
Setting value	0	P11 input pin
	1	P11 output pin

- P10/TMOW Pin

Register	PMR1	PCR1	
Bit Name	**TMOW**	**PCR10**	**Pin Function**
Setting value	0	0	P10 input pin
		1	P10 output pin
	1	X	TMOW output pin

[Legend]
X: Don't care.

9.2 Port 2

Port 2 is a general I/O port also functioning as a SCI3 I/O pin. Each pin of the port 2 is shown in figure 9.2. The register settings of PMR1 and SCI3 have priority for functions of the pins for both uses.

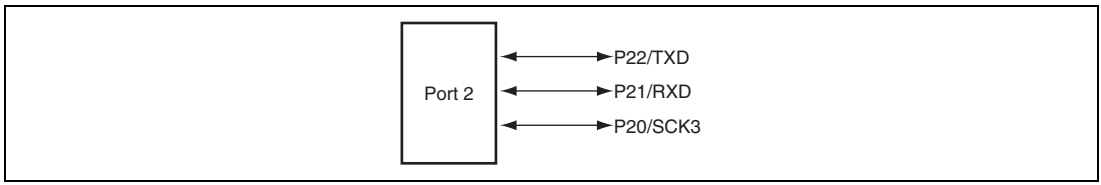


Figure 9.2 Port 2 Pin Configuration

Port 2 has the following registers.

- Port control register 2 (PCR2)
- Port data register 2 (PDR2)

9.2.1 Port Control Register 2 (PCR2)

PCR2 selects inputs/outputs in bit units for pins to be used as general I/O ports of port 2.

Bit	Bit Name	Initial Value	R/W	Description
7	—	—	—	Reserved
6	—	—	—	
5	—	—	—	
4	—	—	—	
3	—	—	—	
2	PCR22	0	W	When each of the port 2 pins P22 to P20 functions as an general I/O port, setting a PCR2 bit to 1 makes the corresponding pin an output port, while clearing the bit to 0 makes the pin an input port.
1	PCR21	0	W	
0	PCR20	0	W	

9.2.2 Port Data Register 2 (PDR2)

PDR2 is a general I/O port data register of port 2.

Bit	Bit Name	Initial Value	R/W	Description
7	—	1	—	Reserved
6	—	1	—	These bits are always read as 1.
5	—	1	—	
4	—	1	—	
3	—	1	—	
2	P22	0	R/W	PDR2 stores output data for port 2 pins.
1	P21	0	R/W	PDR2 is read while PCR2 bits are set to 1, the value stored in PDR2 is read. If PDR2 is read while PCR2 bits are cleared to 0, the pin states are read regardless of the value stored in PDR2.
0	P20	0	R/W	

9.2.3 Pin Functions

The correspondence between the register specification and the port functions is shown below.

- P22/TXD Pin

Register	PMR1	PCR2	
Bit Name	TXD	PCR22	Pin Function
Setting Value	0	0	P22 input pin
		1	P22 output pin
	1	X	TXD output pin

[Legend]
X: Don't care.

- P21/RXD Pin

Register	SCR3	PCR2	
Bit Name	**RE**	**PCR21**	**Pin Function**
Setting Value	0	0	P21 input pin
		1	P21 output pin
	1	X	RXD input pin

[Legend]
X: Don't care.

- P20/SCK3 Pin

Register	SCR3		SMR	PCR2	
Bit Name	**CKE1**	**CKE0**	**COM**	**PCR20**	**Pin Function**
Setting Value	0	0	0	0	P20 input pin
				1	P20 output pin
	0	0	1	X	SCK3 output pin
	0	1	X	X	SCK3 output pin
	1	X	X	X	SCK3 input pin

[Legend]
X: Don't care.

9.3 Port 5

Port 5 is a general I/O port also functioning as an I^2C bus interface I/O pin, an A/D trigger input pin, wakeup interrupt input pin. Each pin of the port 5 is shown in figure 9.3. The register setting of the I^2C bus interface register has priority for functions of the pins P57/SCL and P56/SDA. Since the output buffer for pins P56 and P57 has the NMOS push-pull structure, it differs from an output buffer with the CMOS structure in the high-level output characteristics (see section 20, Electrical Characteristics). The H8/3664N does not have P57 and P56.

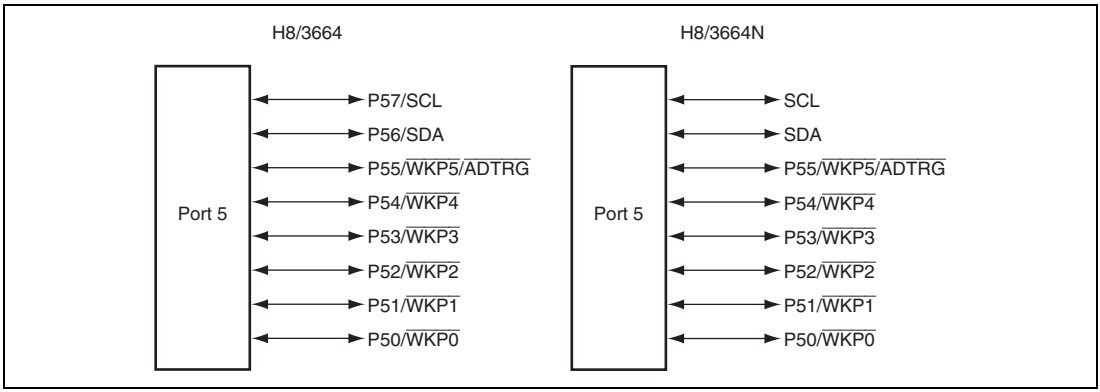


Figure 9.3 Port 5 Pin Configuration

Port 5 has the following registers.

- Port mode register 5 (PMR5)
- Port control register 5 (PCR5)
- Port data register 5 (PDR5)
- Port pull-up control register 5 (PUCR5)

9.3.1 Port Mode Register 5 (PMR5)

PMR5 switches the functions of pins in port 5.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved
6	—	0	—	These bits are always read as 0.
5	WKP5	0	R/W	P55/$\overline{\text{WKP5}}$/$\overline{\text{ADTRG}}$ Pin Function Switch Selects whether pin P55/$\overline{\text{WKP5}}$/$\overline{\text{ADTRG}}$ is used as P55 or as $\overline{\text{WKP5}}$/$\overline{\text{ADTRG}}$ input. 0: General I/O port 1: $\overline{\text{WKP5}}$/$\overline{\text{ADTRG}}$ input pin
4	WKP4	0	R/W	P54/$\overline{\text{WKP4}}$ Pin Function Switch Selects whether pin P54/$\overline{\text{WKP4}}$ is used as P54 or as $\overline{\text{WKP4}}$. 0: General I/O port 1: $\overline{\text{WKP4}}$ input pin
3	WKP3	0	R/W	P53/$\overline{\text{WKP3}}$ Pin Function Switch Selects whether pin P53/$\overline{\text{WKP3}}$ is used as P53 or as $\overline{\text{WKP3}}$. 0: General I/O port 1: $\overline{\text{WKP3}}$ input pin
2	WKP2	0	R/W	P52/$\overline{\text{WKP2}}$ Pin Function Switch Selects whether pin P52/$\overline{\text{WKP2}}$ is used as P52 or as $\overline{\text{WKP2}}$. 0: General I/O port 1: $\overline{\text{WKP2}}$ input pin
1	WKP1	0	R/W	P51/$\overline{\text{WKP1}}$ Pin Function Switch Selects whether pin P51/$\overline{\text{WKP1}}$ is used as P51 or as $\overline{\text{WKP1}}$. 0: General I/O port 1: $\overline{\text{WKP1}}$ input pin

Bit	Bit Name	Initial Value	R/W	Description
0	WKP0	0	R/W	P50/$\overline{\text{WKP0}}$ Pin Function Switch Selects whether pin P50/$\overline{\text{WKP0}}$ is used as P50 or as $\overline{\text{WKP0}}$. 0: General I/O port 1: $\overline{\text{WKP0}}$ input pin

9.3.2 Port Control Register 5 (PCR5)

PCR5 selects inputs/outputs in bit units for pins to be used as general I/O ports of port 5.

Bit	Bit Name	Initial Value	R/W	Description
7	PCR57	0	W	When each of the port 5 pins P57 to P50 functions as an general I/O port, setting a PCR5 bit to 1 makes the corresponding pin an output port, while clearing the bit to 0 makes the pin an input port. Note: Do not set PCR57 and PCR56 to 1 for H8/3664N.
6	PCR56	0	W	
5	PCR55	0	W	
4	PCR54	0	W	
3	PCR53	0	W	
2	PCR52	0	W	
1	PCR51	0	W	
0	PCR50	0	W	

9.3.3 Port Data Register 5 (PDR5)

PDR5 is a general I/O port data register of port 5.

Bit	Bit Name	Initial Value	R/W	Description
7	P57	0	R/W	Stores output data for port 5 pins.
6	P56	0	R/W	If PDR5 is read while PCR5 bits are set to 1, the value stored in PDR5 are read. If PDR5 is read while PCR5 bits are cleared to 0, the pin states are read regardless of the value stored in PDR5. Note: Do not set P57 and P56 to 1 for H8/3664N.
5	P55	0	R/W	
4	P54	0	R/W	
3	P53	0	R/W	
2	P52	0	R/W	
1	P51	0	R/W	
0	P50	0	R/W	

9.3.4 Port Pull-Up Control Register 5 (PUCR5)

PUCR5 controls the pull-up MOS in bit units of the pins set as the input ports.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved
6	—	0	—	These bits are always read as 0.
5	PUCR55	0	R/W	Only bits for which PCR5 is cleared are valid. The pull-up MOS of the corresponding pins enter the on-state when these bits are set to 1, while they enter the off-state when these bits are cleared to 0.
4	PUCR54	0	R/W	
3	PUCR53	0	R/W	
2	PUCR52	0	R/W	
1	PUCR51	0	R/W	
0	PUCR50	0	R/W	

9.3.5 Pin Functions

The correspondence between the register specification and the port functions is shown below.

- P57/SCL Pin

Register	ICCR	PCR5	
Bit Name	**ICE**	**PCR57**	**Pin Function**
Setting Value	0	0	P57 input pin
		1	P57 output pin
	1	X	SCL I/O pin

[Legend]
X: Don't care.

SCL performs the NMOS open-drain output, that enables a direct bus drive.

- P56/SDA Pin

Register	ICCR	PCR5	
Bit Name	**ICE**	**PCR56**	**Pin Function**
Setting Value	0	0	P56 input pin
		1	P56 output pin
	1	X	SDA I/O pin

[Legend]
X: Don't care.

SDA performs the NMOS open-drain output, that enables a direct bus drive.

- P55/$\overline{\text{WKP5}}$/$\overline{\text{ADTRG}}$ Pin

Register	PMR5	PCR5	
Bit Name	**WKP5**	**PCR55**	**Pin Function**
Setting Value	0	0	P55 input pin
		1	P55 output pin
	1	X	$\overline{\text{WKP5}}$/$\overline{\text{ADTRG}}$ input pin

[Legend]
X: Don't care.

- P54/$\overline{\text{WKP4}}$ Pin

Register	PMR5	PCR5	
Bit Name	**WKP4**	**PCR54**	**Pin Function**
Setting Value	0	0	P54 input pin
		1	P54 output pin
	1	X	$\overline{\text{WKP4}}$ input pin

[Legend]
X: Don't care.

- P53/$\overline{\text{WKP3}}$ Pin

Register	PMR5	PCR5	
Bit Name	**WKP3**	**PCR53**	**Pin Function**
Setting Value	0	0	P53 input pin
		1	P53 output pin
	1	X	$\overline{\text{WKP3}}$ input pin

[Legend]
X: Don't care.

- P52/$\overline{\text{WKP2}}$ Pin

Register	PMR5	PCR5	
Bit Name	**WKP2**	**PCR52**	**Pin Function**
Setting Value	0	0	P52 input pin
		1	P52 output pin
	1	X	$\overline{\text{WKP2}}$ input pin

[Legend]
X: Don't care.

- P51/$\overline{\text{WKP1}}$ Pin

Register	PMR5	PCR5	
Bit Name	**WKP1**	**PCR51**	**Pin Function**
Setting Value	0	0	P51 input pin
		1	P51 output pin
	1	X	$\overline{\text{WKP1}}$ input pin

[Legend]
X: Don't care.

- P50/$\overline{\text{WKP0}}$ Pin

Register	PMR5	PCR5	
Bit Name	**WKP0**	**PCR50**	**Pin Function**
Setting Value	0	0	P50 input pin
		1	P50 output pin
	1	X	$\overline{\text{WKP0}}$ input pin

[Legend]
X: Don't care.

9.4 Port 7

Port 7 is a general I/O port also functioning as a timer V I/O pin. Each pin of the port 7 is shown in figure 9.4. The register setting of TCSRV in timer V has priority for functions of pin P76/TMOV. The pins, P75/TMCIV and P74/TMRIV, are also functioning as timer V input ports that are connected to the timer V regardless of the register setting of port 7.

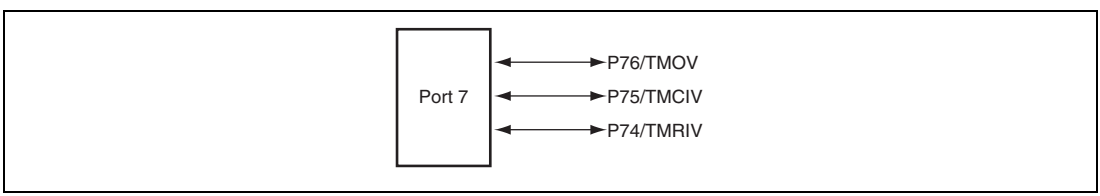


Figure 9.4 Port 7 Pin Configuration

Port 7 has the following registers.

- Port control register 7 (PCR7)
- Port data register 7 (PDR7)

9.4.1 Port Control Register 7 (PCR7)

PCR7 selects inputs/outputs in bit units for pins to be used as general I/O ports of port 7.

Bit	Bit Name	Initial Value	R/W	Description
7	—	—	—	Reserved
6	PCR76	0	W	Setting a PCR7 bit to 1 makes the corresponding pin an output port, while clearing the bit to 0 makes the pin an input port. Note that the TCSRV setting of the timer V has priority for deciding input/output direction of the P76/TMOV pin.
5	PCR75	0	W	
4	PCR74	0	W	
3	—	—	—	Reserved
2	—	—	—	
1	—	—	—	
0	—	—	—	

9.4.2 Port Data Register 7 (PDR7)

PDR7 is a general I/O port data register of port 7.

Bit	Bit Name	Initial Value	R/W	Description
7	—	1	—	Reserved This bit is always read as 1.
6	P76	0	R/W	PDR7 stores output data for port 7 pins.
5	P75	0	R/W	PDR7 is read while PCR7 bits are set to 1, the value stored in PDR7 is read. If PDR7 is read while PCR7 bits are cleared to 0, the pin states are read regardless of the value stored in PDR7.
4	P74	0	R/W	
3	—	1	—	Reserved
2	—	1	—	These bits are always read as 1.
1	—	1	—	
0	—	1	—	

9.4.3 Pin Functions

The correspondence between the register specification and the port functions is shown below.

- P76/TMOV Pin

Register	TCSRV	PCR7	
Bit Name	**OS3 to OS0**	**PCR76**	**Pin Function**
Setting Value	0000	0	P76 input pin
		1	P76 output pin
	Other than the above values	X	TMOV output pin

[Legend]
X: Don't care.

- P75/TMCIV Pin

Register	PCR7	
Bit Name	**PCR75**	**Pin Function**
Setting Value	0	P75 input/TMCIV input pin
	1	P75 output/TMCIV input pin

- P74/TMRIV Pin

Register	PCR7	
Bit Name	**PCR74**	**Pin Function**
Setting Value	0	P74 input/TMRIV input pin
	1	P74 output/TMRIV input pin

9.5 Port 8

Port 8 is a general I/O port also functioning as a timer W I/O pin. Each pin of the port 8 is shown in figure 9.5. The register setting of the timer W has priority for functions of the pins P84/FTIOD, P83/FTIOC, P82/FTIOB, and P81/FTIOA. P80/FTCI also functions as a timer W input port that is connected to the timer W regardless of the register setting of port 8.

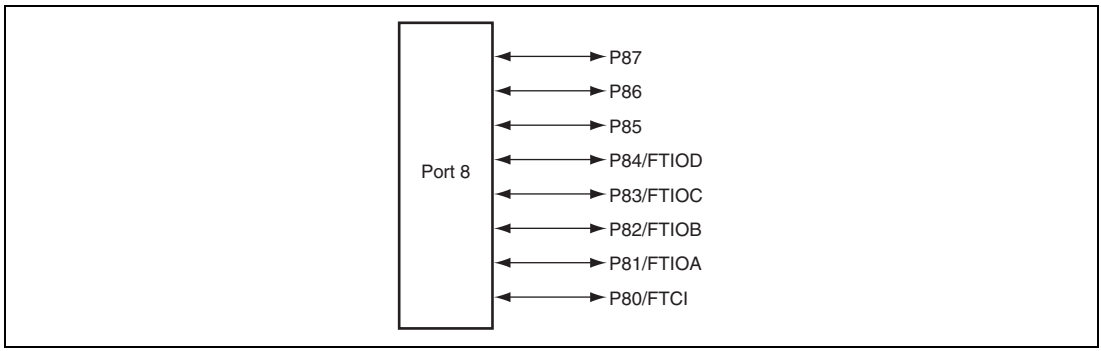


Figure 9.5 Port 8 Pin Configuration

Port 8 has the following registers.

- Port control register 8 (PCR8)
- Port data register 8 (PDR8)

9.5.1 Port Control Register 8 (PCR8)

PCR8 selects inputs/outputs in bit units for pins to be used as general I/O ports of port 8.

Bit	Bit Name	Initial Value	R/W	Description
7	PCR87	0	W	When each of the port 8 pins P87 to P80 functions as an general I/O port, setting a PCR8 bit to 1 makes the corresponding pin an output port, while clearing the bit to 0 makes the pin an input port.
6	PCR86	0	W	
5	PCR85	0	W	
4	PCR84	0	W	
3	PCR83	0	W	
2	PCR82	0	W	
1	PCR81	0	W	
0	PCR80	0	W	

9.5.2 Port Data Register 8 (PDR8)

PDR8 is a general I/O port data register of port 8.

Bit	Bit Name	Initial Value	R/W	Description
7	P87	0	R/W	PDR8 stores output data for port 8 pins.
6	P86	0	R/W	PDR8 is read while PCR8 bits are set to 1, the value stored in PDR8 is read. If PDR8 is read while PCR8 bits are cleared to 0, the pin states are read regardless of the value stored in PDR8.
5	P85	0	R/W	
4	P84	0	R/W	
3	P83	0	R/W	
2	P82	0	R/W	
1	P81	0	R/W	
0	P80	0	R/W	

9.5.3 Pin Functions

The correspondence between the register specification and the port functions is shown below.

- P87 Pin

Register	PCR8	
Bit Name	**PCR87**	**Pin Function**
Setting Value	0	P87 input pin
	1	P87 output pin

- P86 Pin

Register	PCR8	
Bit Name	**PCR86**	**Pin Function**
Setting Value	0	P86 input pin
	1	P86 output pin

- P85 Pin

Register	PCR8	
Bit Name	**PCR85**	**Pin Function**
Setting Value	0	P85 input pin
	1	P85 output pin

- P84/FTIOD Pin

Register	TMRW		TIOR1		PCR8	
Bit Name	**PWMD**	**IOD2**	**IOD1**	**IOD0**	**PCR84**	**Pin Function**
Setting Value	0	0	0	0	0	P84 input/FTIOD input pin
					1	P84 output/FTIOD input pin
		0	0	1	X	FTIOD output pin
		0	1	X	X	FTIOD output pin
		1	X	X	0	P84 input/FTIOD input pin
					1	P84 output/FTIOD input pin
	1	X	X	X	X	PWM output pin

[Legend]
X: Don't care.

- P83/FTIOC Pin

Register	TMRW		TIOR1		PCR8	
Bit Name	**PWMC**	**IOC2**	**IOC1**	**IOC0**	**PCR83**	**Pin Function**
Setting Value	0	0	0	0	0	P83 input/FTIOC input pin
					1	P83 output/FTIOC input pin
		0	0	1	X	FTIOC output pin
		0	1	X	X	FTIOC output pin
		1	X	X	0	P83 input/FTIOC input pin
					1	P83 output/FTIOC input pin
	1	X	X	X	X	PWM output pin

[Legend]
X: Don't care.

- P82/FTIOB Pin

Register	TMRW	TIOR0			PCR8	
Bit Name	**PWMB**	**IOB2**	**IOB1**	**IOB0**	**PCR82**	**Pin Function**
Setting Value	0	0	0	0	0	P82 input/FTIOB input pin
					1	P82 output/FTIOB input pin
		0	0	1	X	FTIOB output pin
		0	1	X	X	FTIOB output pin
		1	X	X	0	P82 input/FTIOB input pin
					1	P82 output/FTIOB input pin
	1	X	X	X	X	PWM output pin

[Legend]
X: Don't care.

- P81/FTIOA Pin

Register	TIOR0			PCR8	
Bit Name	**IOA2**	**IOA1**	**IOA0**	**PCR81**	**Pin Function**
Setting Value	0	0	0	0	P81 input/FTIOA input pin
			0	1	P81 output/FTIOA input pin
	0	0	1	X	FTIOA output pin
	0	1	X	X	FTIOA output pin
	1	X	X	0	P81 input/FTIOA input pin
				1	P81 output/FTIOA input pin

[Legend]
X: Don't care.

- P80/FTCI Pin

Register	PCR8	
Bit Name	**PCR80**	**Pin Function**
Setting Value	0	P80 input/FTCI input pin
	1	P80 output/FTCI input pin

9.6 Port B

Port B is an input port also functioning as an A/D converter analog input pin. Each pin of the port B is shown in figure 9.6.

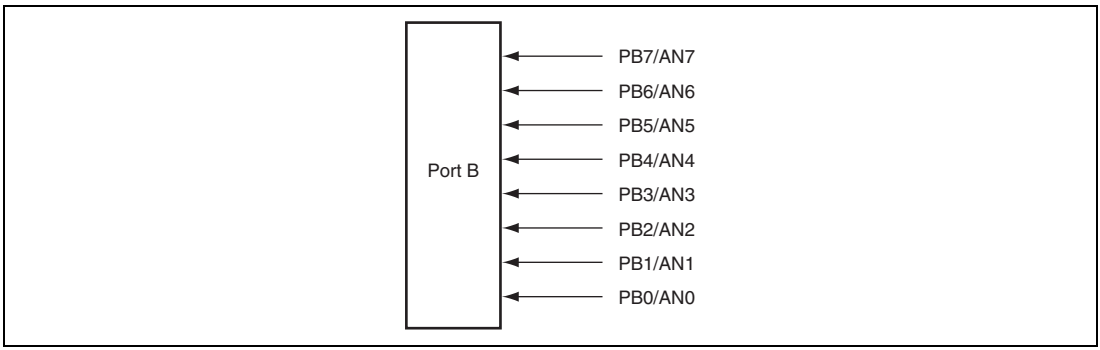


Figure 9.6 Port B Pin Configuration

Port B has the following register.

- Port data register B (PDRB)

9.6.1 Port Data Register B (PDRB)

PDRB is a general input-only port data register of port B.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7	—	R	The input value of each pin is read by reading this register. However, if a port B pin is designated as an analog input channel by ADCSR in A/D converter, 0 is read.
6	PB6	—	R	
5	PB5	—	R	
4	PB4	—	R	
3	PB3	—	R	
2	PB2	—	R	
1	PB1	—	R	
0	PB0	—	R	

Section 10 Timer A

Timer A is an 8-bit timer with interval timing and real-time clock time-base functions. The clock time-base function is available when a 32.768kHz crystal oscillator is connected. Figure 10.1 shows a block diagram of timer A.

10.1 Features

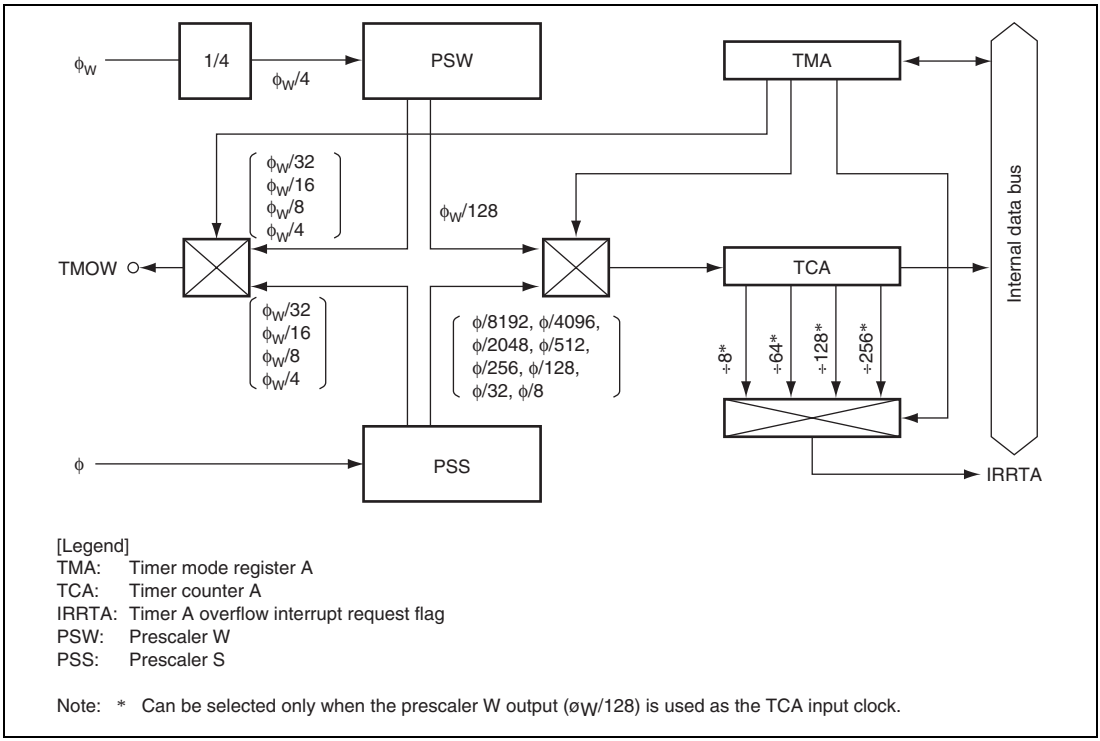
- Timer A can be used as an interval timer or a clock time base.
- An interrupt is requested when the counter overflows.
- Any of eight clock signals can be output from pin TMOW: 32.768 kHz divided by 32, 16, 8, or 4 (1 kHz, 2 kHz, 4 kHz, 8 kHz), or the system clock divided by 32, 16, 8, or 4.

Interval Timer:

- Choice of eight internal clock sources (ϕ/8192, ϕ/4096, ϕ/2048, ϕ/512, ϕ/256, ϕ/128, ϕ/32, ϕ8)

Clock Time Base:

- Choice of four overflow periods (1 s, 0.5 s, 0.25 s, 31.25 ms) when timer A is used as a clock time base (using a 32.768 kHz crystal oscillator).



[Legend]
TMA: Timer mode register A
TCA: Timer counter A
IRRTA: Timer A overflow interrupt request flag
PSW: Prescaler W
PSS: Prescaler S

Note: * Can be selected only when the prescaler W output ($\phi_W/128$) is used as the TCA input clock.

Figure 10.1 Block Diagram of Timer A

10.2 Input/Output Pins

Table 10.1 shows the timer A input/output pin.

Table 10.1 Pin Configuration

Name	Abbreviation	I/O	Function
Clock output	TMOW	Output	Output of waveform generated by timer A output circuit

10.3 Register Descriptions

Timer A has the following registers.

- Timer mode register A (TMA)
- Timer counter A (TCA)

10.3.1 Timer Mode Register A (TMA)

TMA selects the operating mode, the divided clock output, and the input clock.

Bit	Bit Name	Initial Value	R/W	Description
7	TMA7	0	R/W	Clock Output Select 7 to 5
6	TMA6	0	R/W	These bits select the clock output at the TMOW pin.
5	TMA5	0	R/W	000: ϕ/32
				001: ϕ/16
				010: ϕ/8
				011: ϕ/4
				100: ϕ_w/32
				101: ϕ_w/16
				110: ϕ_w/8
				111: ϕ_w/4
				For details on clock outputs, see section 10.4.3, Clock Output.
4	—	1	—	Reserved
				This bit is always read as 1.
3	TMA3	0	R/W	Internal Clock Select 3
				This bit selects the operating mode of the timer A.
				0: Functions as an interval timer to count the outputs of prescaler S.
				1: Functions as a clock-time base to count the outputs of prescaler W.

Bit	Bit Name	Initial Value	R/W	Description
2	TMA2	0	R/W	Internal Clock Select 2 to 0
1	TMA1	0	R/W	These bits select the clock input to TCA when TMA3 = 0.
0	TMA0	0	R/W	000: ϕ/8192
				001: ϕ/4096
				010: ϕ/2048
				011: ϕ/512
				100: ϕ/256
				101: ϕ/128
				110: ϕ/32
				111: ϕ/8
				These bits select the overflow period when TMA3 = 1 (when a 32.768 kHz crystal oscillator with is used as ϕW).
				000: 1s
				001: 0.5 s
				010: 0.25 s
				011: 0.03125 s
				1XX: Both PSW and TCA are reset

[Legend]

X: Don't care.

10.3.2 Timer Counter A (TCA)

TCA is an 8-bit readable up-counter, which is incremented by internal clock input. The clock source for input to this counter is selected by bits TMA3 to TMA0 in TMA. TCA values can be read by the CPU in active mode, but cannot be read in subactive mode. When TCA overflows, the IRRTA bit in interrupt request register 1 (IRR1) is set to 1. TCA is cleared by setting bits TMA3 and TMA2 in TMA to B’11. TCA is initialized to H'00.

10.4 Operation

10.4.1 Interval Timer Operation

When bit TMA3 in TMA is cleared to 0, timer A functions as an 8-bit interval timer.

Upon reset, TCA is cleared to H'00 and bit TMA3 is cleared to 0, so up-counting of timer A resume immediately as an interval timer. The clock input to timer A is selected by bits TMA2 to TMA0 in TMA; any of eight internal clock signals output by prescaler S can be selected.

After the count value in TCA reaches H'FF, the next clock signal input causes timer A to overflow, setting bit IRRTA to 1 in interrupt Flag Register 1 (IRR1). If IENTA = 1 in interrupt enable register 1 (IENR1), a CPU interrupt is requested. At overflow, TCA returns to H'00 and starts counting up again. In this mode timer A functions as an interval timer that generates an overflow output at intervals of 256 input clock pulses.

10.4.2 Clock Time Base Operation

When bit TMA3 in TMA is set to 1, timer A functions as a clock-timer base by counting clock signals output by prescaler W. When a clock signal is input after the TCA counter value has become H'FF, timer A overflows and IRRTA in IRR1 is set to 1. At that time, an interrupt request is generated to the CPU if IENTA in the interrupt enable register 1 (IENR1) is 1. The overflow period of timer A is set by bits TMA1 and TMA0 in TMA. A choice of four periods is available. In clock time base operation (TMA3 = 1), setting bit TMA2 to 1 clears both TCA and prescaler W to H'00.

10.4.3 Clock Output

Setting bit TMOW in port mode register 1 (PMR1) to 1 causes a clock signal to be output at pin TMOW. Eight different clock output signals can be selected by means of bits TMA7 to TMA5 in TMA. The system clock divided by 32, 16, 8, or 4 can be output in active mode and sleep mode. A 32.768 kHz signal divided by 32, 16, 8, or 4 can be output in active mode, sleep mode, and subactive mode.

10.5 Usage Note

When the clock time base function is selected as the internal clock of TCA in active mode or sleep mode, the internal clock is not synchronous with the system clock, so it is synchronized by a synchronizing circuit. This may result in a maximum error of 1/ϕ (s) in the count cycle.

Section 11 Timer V

Timer V is an 8-bit timer based on an 8-bit counter. Timer V counts external events. Compare-match signals with two registers can also be used to reset the counter, request an interrupt, or output a pulse signal with an arbitrary duty cycle. Counting can be initiated by a trigger input at the TRGV pin, enabling pulse output control to be synchronized to the trigger, with an arbitrary delay from the trigger input. Figure 11.1 shows a block diagram of timer V.

11.1 Features

- Choice of seven clock signals is available.
  Choice of six internal clock sources (ϕ/128, ϕ/64, ϕ/32, ϕ/16, ϕ/8, ϕ/4) or an external clock.
- Counter can be cleared by compare match A or B, or by an external reset signal. If the count stop function is selected, the counter can be halted when cleared.
- Timer output is controlled by two independent compare match signals, enabling pulse output with an arbitrary duty cycle, PWM output, and other applications.
- Three interrupt sources: compare match A, compare match B, timer overflow
- Counting can be initiated by trigger input at the TRGV pin. The rising edge, falling edge, or both edges of the TRGV input can be selected.

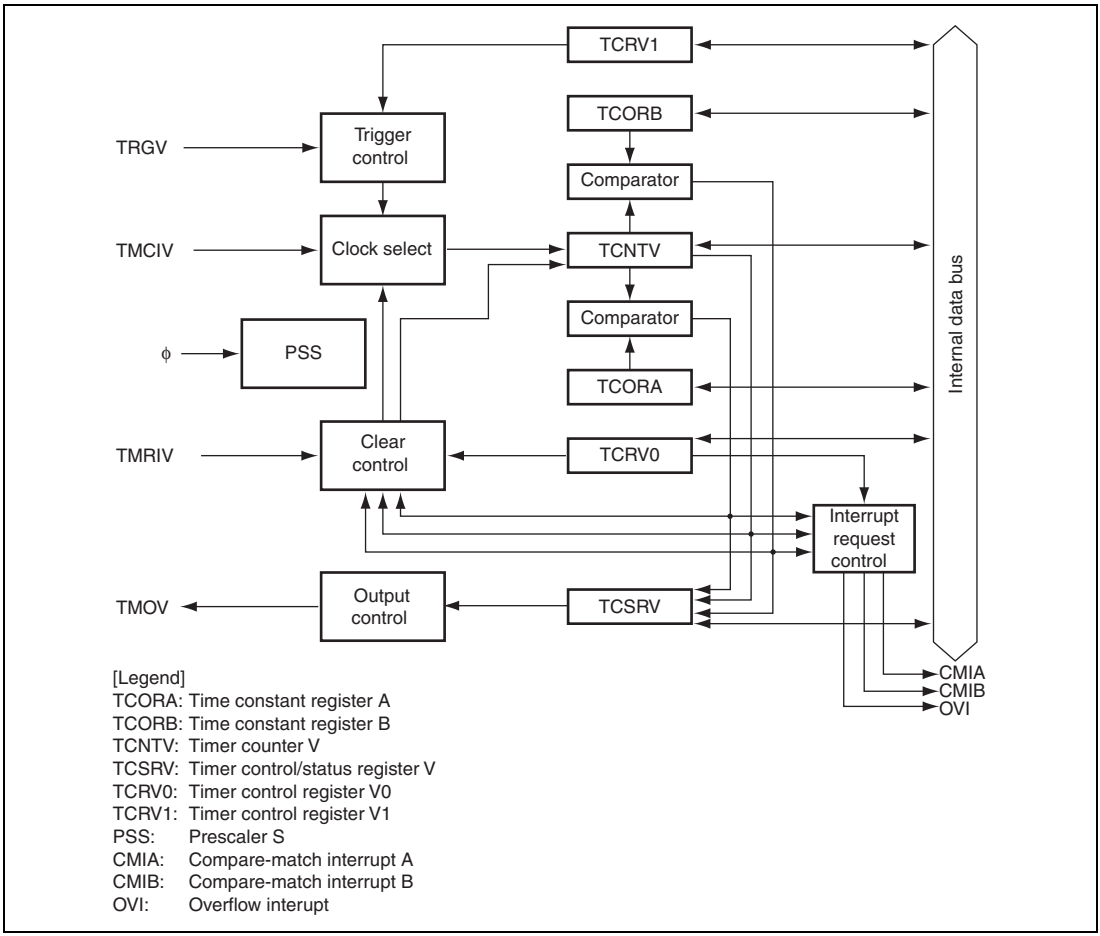


Figure 11.1 Block Diagram of Timer V

11.2 Input/Output Pins

Table 11.1 shows the timer V pin configuration.

Table 11.1 Pin Configuration

Name	Abbreviation	I/O	Function
Timer V output	TMOV	Output	Timer V waveform output
Timer V clock input	TMCIV	Input	Clock input to TCNTV
Timer V reset input	TMRIV	Input	External input to reset TCNTV
Trigger input	TRGV	Input	Trigger input to initiate counting

11.3 Register Descriptions

Time V has the following registers.

- Timer counter V (TCNTV)
- Timer constant register A (TCORA)
- Timer constant register B (TCORB)
- Timer control register V0 (TCRV0)
- Timer control/status register V (TCSRV)
- Timer control register V1 (TCRV1)

11.3.1 Timer Counter V (TCNTV)

TCNTV is an 8-bit up-counter. The clock source is selected by bits CKS2 to CKS0 in timer control register V0 (TCRV0). The TCNTV value can be read and written by the CPU at any time. TCNTV can be cleared by an external reset input signal, or by compare match A or B. The clearing signal is selected by bits CCLR1 and CCLR0 in TCRV0.

When TCNTV overflows, OVF is set to 1 in timer control/status register V (TCSRV).

TCNTV is initialized to H'00.

11.3.2 Time Constant Registers A and B (TCORA, TCORB)

TCORA and TCORB have the same function.

TCORA and TCORB are 8-bit read/write registers.

TCORA and TCNTV are compared at all times. When the TCORA and TCNTV contents match, CMFA is set to 1 in TCSRV. If CMIEA is also set to 1 in TCRV0, a CPU interrupt is requested. Note that they must not be compared during the T3 state of a TCORA write cycle.

Timer output from the TMOV pin can be controlled by the identifying signal (compare match A) and the settings of bits OS3 to OS0 in TCSRV.

TCORA and TCORB are initialized to H'FF.

11.3.3 Timer Control Register V0 (TCRV0)

TCRV0 selects the input clock signals of TCNTV, specifies the clearing conditions of TCNTV, and controls each interrupt request.

Bit	Bit Name	Initial Value	R/W	Description
7	CMIEB	0	R/W	Compare Match Interrupt Enable B When this bit is set to 1, interrupt request from the CMFB bit in TCSRV is enabled.
6	CMIEA	0	R/W	Compare Match Interrupt Enable A When this bit is set to 1, interrupt request from the CMFA bit in TCSRV is enabled.
5	OVIE	0	R/W	Timer Overflow Interrupt Enable When this bit is set to 1, interrupt request from the OVF bit in TCSRV is enabled.

Bit	Bit Name	Initial Value	R/W	Description
4	CCLR1	0	R/W	Counter Clear 1 and 0
3	CCLR0	0	R/W	These bits specify the clearing conditions of TCNTV. 00: Clearing is disabled 01: Cleared by compare match A 10: Cleared by compare match B 11: Cleared on the rising edge of the TMRIV pin. The operation of TCNTV after clearing depends on TRGE in TCRV1.
2	CKS2	0	R/W	Clock Select 2 to 0
1	CKS1	0	R/W	These bits select clock signals to input to TCNTV and the counting condition in combination with ICKS0 in TCRV1. Refer to table 11.2.
0	CKS0	0	R/W	

Table 11.2 Clock Signals to Input to TCNTV and Counting Conditions

TCRV0			TCRV1	
Bit 2	Bit 1	Bit 0	Bit 0	
CKS2	CKS1	CKS0	ICKS0	Description
0	0	0	—	Clock input prohibited
		1	0	Internal clock: counts on ϕ/4, falling edge
			1	Internal clock: counts on ϕ/8, falling edge
	1	0	0	Internal clock: counts on ϕ/16, falling edge
			1	Internal clock: counts on ϕ/32, falling edge
		1	0	Internal clock: counts on ϕ/64, falling edge
			1	Internal clock: counts on ϕ/128, falling edge
1	0	0	—	Clock input prohibited
		1	—	External clock: counts on rising edge
	1	0	—	External clock: counts on falling edge
		1	—	External clock: counts on rising and falling edge

11.3.4 Timer Control/Status Register V (TCSRV)

TCSRV indicates the status flag and controls outputs by using a compare match.

Bit	Bit Name	Initial Value	R/W	Description
7	CMFB	0	R/W	Compare Match Flag B Setting condition: When the TCNTV value matches the TCORB value Clearing condition: After reading CMFB = 1, cleared by writing 0 to CMFB
6	CMFA	0	R/W	Compare Match Flag A Setting condition: When the TCNTV value matches the TCORA value Clearing condition: After reading CMFA = 1, cleared by writing 0 to CMFA
5	OVF	0	R/W	Timer Overflow Flag Setting condition: When TCNTV overflows from H'FF to H'00 Clearing condition: After reading OVF = 1, cleared by writing 0 to OVF
4	—	1	—	Reserved This bit is always read as 1.
3	OS3	0	R/W	Output Select 3 and 2
2	OS2	0	R/W	These bits select an output method for the TMOV pin by the compare match of TCORB and TCNTV. 00: No change 01: 0 output 10: 1 output 11: Output toggles

Bit	Bit Name	Initial Value	R/W	Description
1	OS1	0	R/W	Output Select 1 and 0
0	OS0	0	R/W	These bits select an output method for the TMOV pin by the compare match of TCORA and TCNTV. 00: No change 01: 0 output 10: 1 output 11: Output toggles

OS3 and OS2 select the output level for compare match B. OS1 and OS0 select the output level for compare match A. The two output levels can be controlled independently. After a reset, the timer output is 0 until the first compare match.

11.3.5 Timer Control Register V1 (TCRV1)

TCRV1 selects the edge at the TRGV pin, enables TRGV input, and selects the clock input to TCNTV.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 1	—	Reserved These bits are always read as 1.
4	TVEG1	0	R/W	TRGV Input Edge Select
3	TVEG0	0	R/W	These bits select the TRGV input edge. 00: TRGV trigger input is prohibited 01: Rising edge is selected 10: Falling edge is selected 11: Rising and falling edges are both selected
2	TRGE	0	R/W	TCNTV starts counting up by the input of the edge which is selected by TVEG1 and TVEG0. 0: Disables starting counting-up TCNTV by the input of the TRGV pin and halting counting-up TCNTV when TCNTV is cleared by a compare match. 1: Enables starting counting-up TCNTV by the input of the TRGV pin and halting counting-up TCNTV when TCNTV is cleared by a compare match.

Bit	Bit Name	Initial Value	R/W	Description
1	—	1	—	Reserved This bit is always read as 1.
0	ICKS0	0	R/W	Internal Clock Select 0 This bit selects clock signals to input to TCNTV in combination with CKS2 to CKS0 in TCRV0. Refer to table 11.2.

11.4 Operation

11.4.1 Timer V Operation

1. According to table 11.2, six internal/external clock signals output by prescaler S can be selected as the timer V operating clock signals. When the operating clock signal is selected, TCNTV starts counting-up. Figure 11.2 shows the count timing with an internal clock signal selected, and figure 11.3 shows the count timing with both edges of an external clock signal selected.
2. When TCNTV overflows (changes from H'FF to H'00), the overflow flag (OVF) in TCRV0 will be set. The timing at this time is shown in figure 11.4. An interrupt request is sent to the CPU when OVIE in TCRV0 is 1.
3. TCNTV is constantly compared with TCORA and TCORB. Compare match flag A or B (CMFA or CMFB) is set to 1 when TCNTV matches TCORA or TCORB, respectively. The compare-match signal is generated in the last state in which the values match. Figure 11.5 shows the timing. An interrupt request is generated for the CPU when CMIEA or CMIEB in TCRV0 is 1.
4. When a compare match A or B is generated, the TMOV responds with the output value selected by bits OS3 to OS0 in TCSRV. Figure 11.6 shows the timing when the output is toggled by compare match A.
5. When CCLR1 or CCLR0 in TCRV0 is 01 or 10, TCNTV can be cleared by the corresponding compare match. Figure 11.7 shows the timing.
6. When CCLR1 or CCLR0 in TCRV0 is 11, TCNTV can be cleared by the rising edge of the input of TMRIV pin. A TMRIV input pulse-width of at least 1.5 system clocks is necessary. Figure 11.8 shows the timing.
7. When a counter-clearing source is generated with TRGE in TCRV1 set to 1, the counting-up is halted as soon as TCNTV is cleared. TCNTV resumes counting-up when the edge selected by TVEG1 or TVEG0 in TCRV1 is input from the TGRV pin.

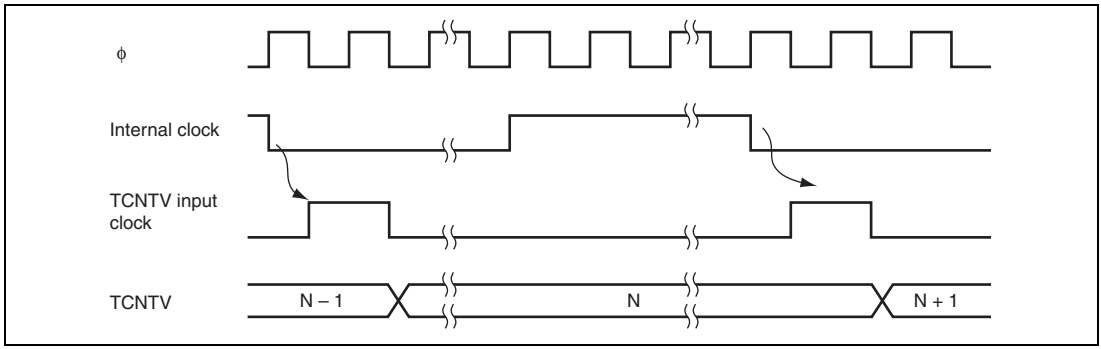


Figure 11.2 Increment Timing with Internal Clock

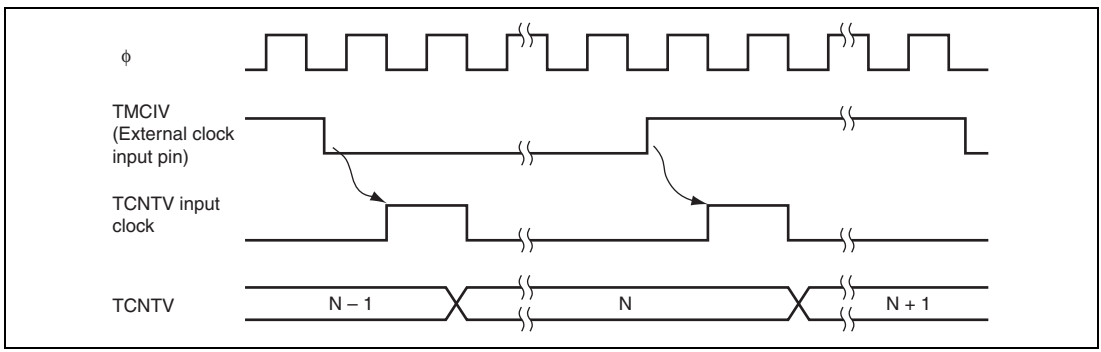


Figure 11.3 Increment Timing with External Clock

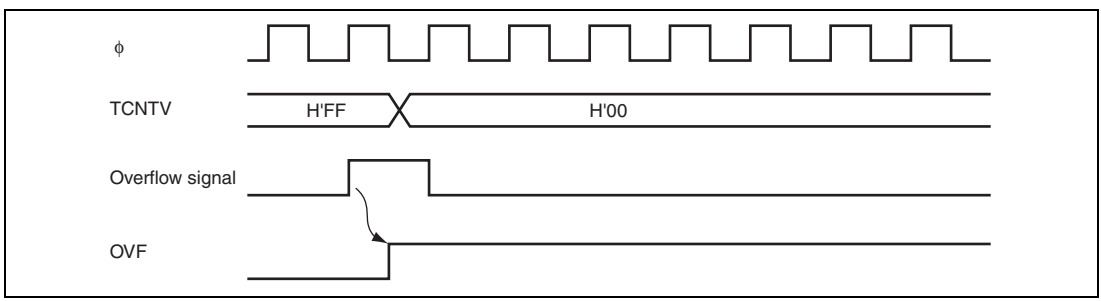


Figure 11.4 OVF Set Timing

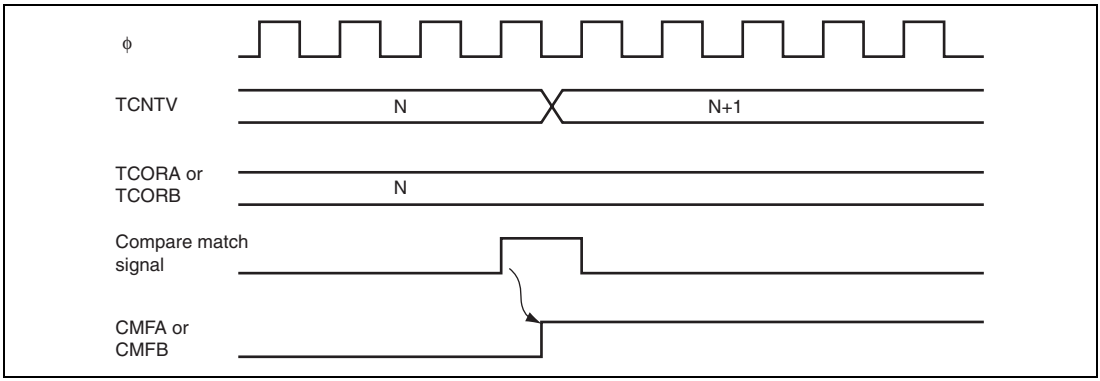


Figure 11.5 CMFA and CMFB Set Timing

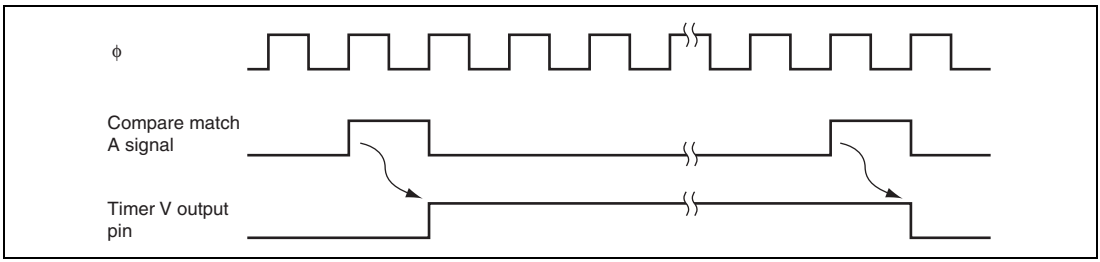


Figure 11.6 TMOV Output Timing

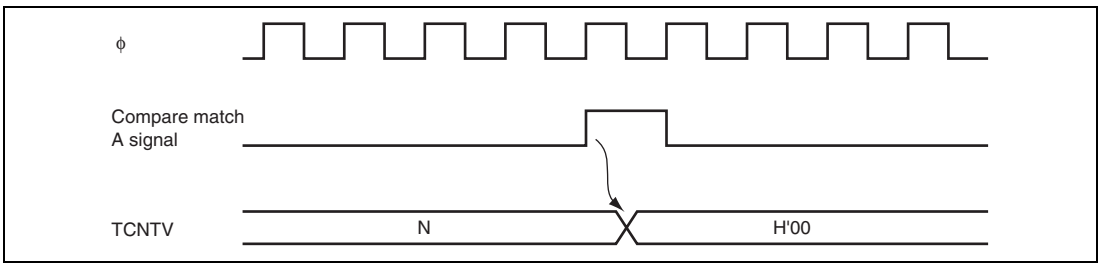


Figure 11.7 Clear Timing by Compare Match

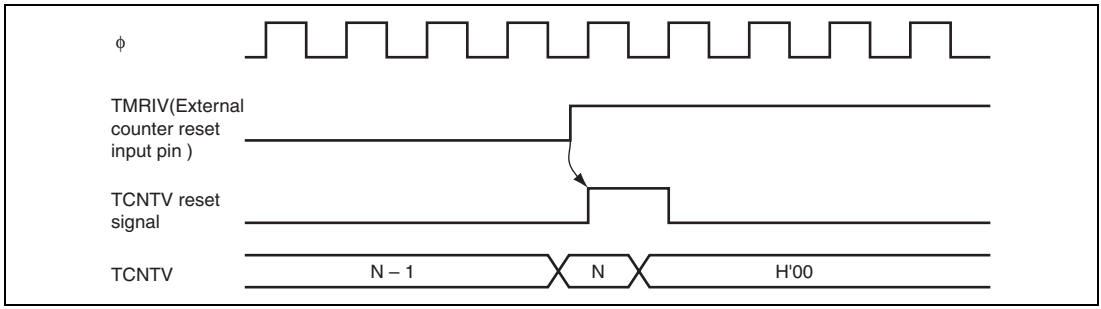


Figure 11.8 Clear Timing by TMRIV Input

11.5 Timer V Application Examples

11.5.1 Pulse Output with Arbitrary Duty Cycle

Figure 11.9 shows an example of output of pulses with an arbitrary duty cycle.

1. Set bits CCLR1 and CCLR0 in TCRV0 so that TCNTV will be cleared by compare match with TCORA.
2. Set bits OS3 to OS0 in TCSRV so that the output will go to 1 at compare match with TCORA and to 0 at compare match with TCORB.
3. Set bits CKS2 to CKS0 in TCRV0 and bit ICKS0 in TCRV1 to select the desired clock source.
4. With these settings, a waveform is output without further software intervention, with a period determined by TCORA and a pulse width determined by TCORB.

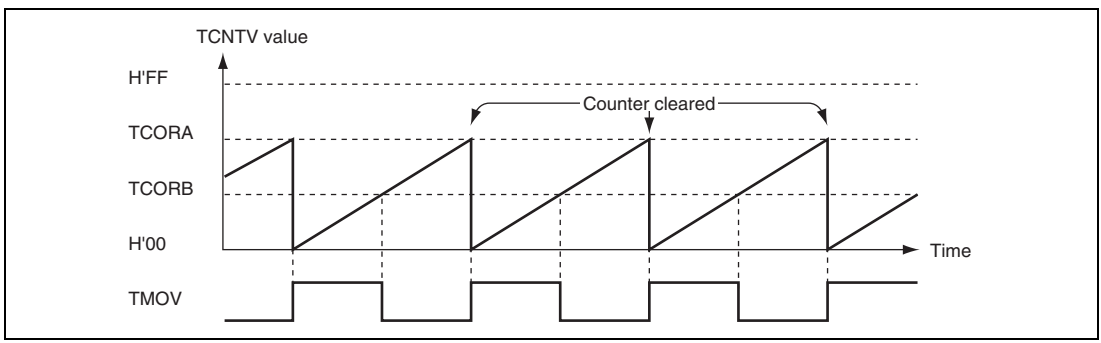


Figure 11.9 Pulse Output Example

11.5.2 Pulse Output with Arbitrary Pulse Width and Delay from TRGV Input

The trigger function can be used to output a pulse with an arbitrary pulse width at an arbitrary delay from the TRGV input, as shown in figure 11.10. To set up this output:

1. Set bits CCLR1 and CCLR0 in TCRV0 so that TCNTV will be cleared by compare match with TCORB.
2. Set bits OS3 to OS0 in TCSRV so that the output will go to 1 at compare match with TCORA and to 0 at compare match with TCORB.
3. Set bits TVEG1 and TVEG0 in TCRV1 and set TRGE to select the falling edge of the TRGV input.
4. Set bits CKS2 to CKS0 in TCRV0 and bit ICKS0 in TCRV1 to select the desired clock source.
5. After these settings, a pulse waveform will be output without further software intervention, with a delay determined by TCORA from the TRGV input, and a pulse width determined by (TCORB – TCORA).

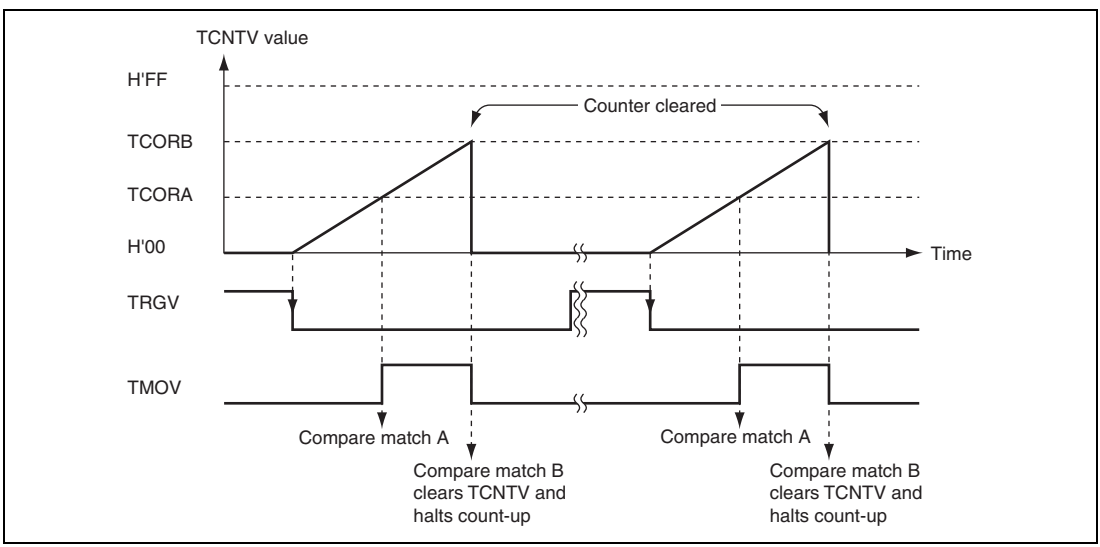


Figure 11.10 Example of Pulse Output Synchronized to TRGV Input

11.6 Usage Notes

The following types of contention or operation can occur in timer V operation.

1. Writing to registers is performed in the T3 state of a TCNTV write cycle. If a TCNTV clear signal is generated in the T3 state of a TCNTV write cycle, as shown in figure 11.11, clearing takes precedence and the write to the counter is not carried out. If counting-up is generated in the T3 state of a TCNTV write cycle, writing takes precedence.
2. If a compare match is generated in the T3 state of a TCORA or TCORB write cycle, the write to TCORA or TCORB takes precedence and the compare match signal is inhibited. Figure 11.12 shows the timing.
3. If compare matches A and B occur simultaneously, any conflict between the output selections for compare match A and compare match B is resolved by the following priority: toggle output > output 1 > output 0.
4. Depending on the timing, TCNTV may be incremented by a switch between different internal clock sources. When TCNTV is internally clocked, an increment pulse is generated from the falling edge of an internal clock signal, that is divided system clock (ϕ). Therefore, as shown in figure 11.3 the switch is from a high clock signal to a low clock signal, the switchover is seen as a falling edge, causing TCNTV to increment. TCNTV can also be incremented by a switch between internal and external clocks.

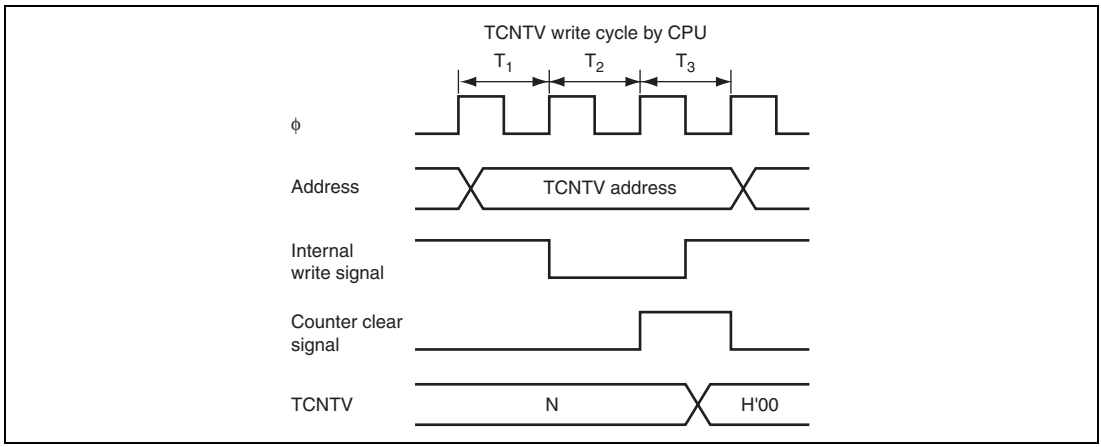


Figure 11.11 Contention between TCNTV Write and Clear

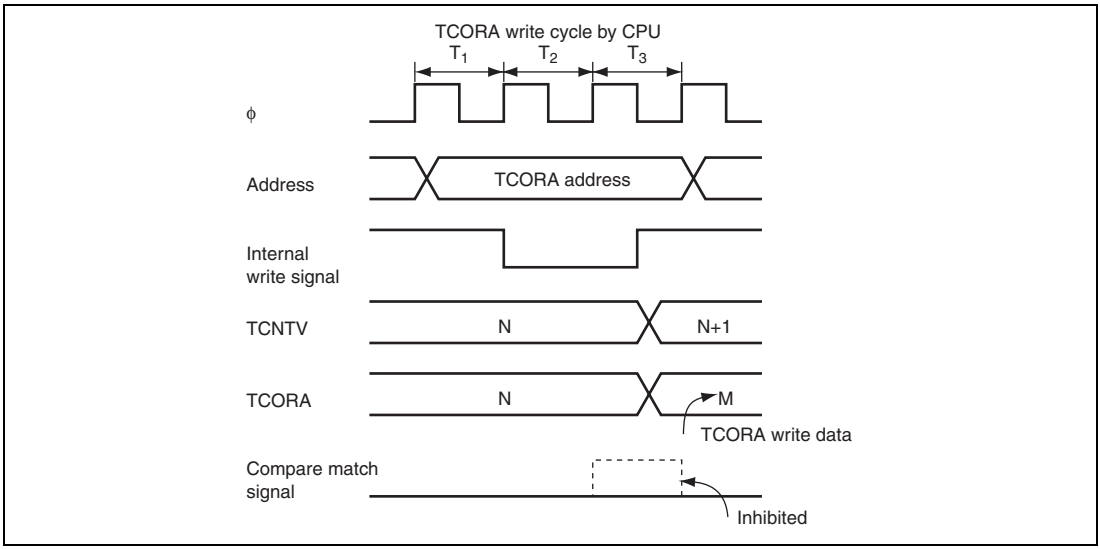


Figure 11.12 Contention between TCORA Write and Compare Match

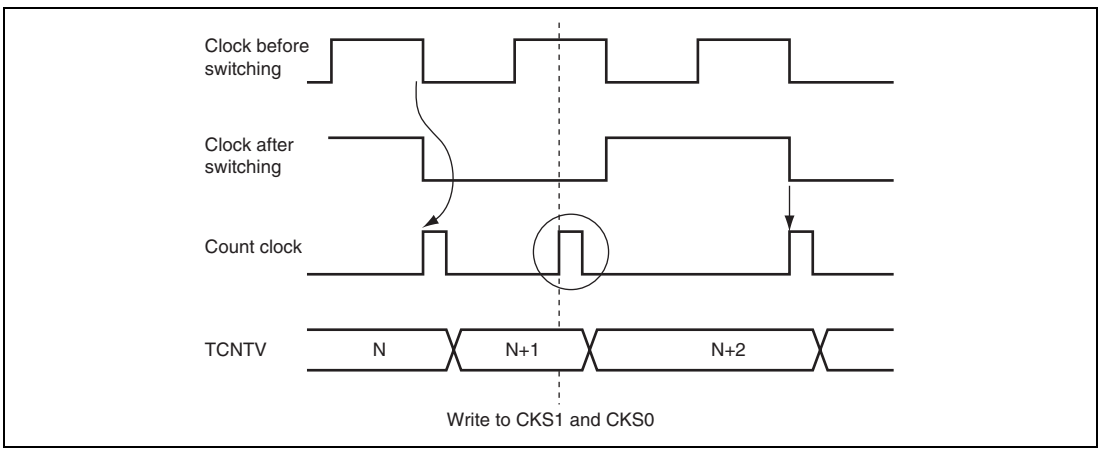


Figure 11.13 Internal Clock Switching and TCNTV Operation

Section 12 Timer W

The timer W has a 16-bit timer having output compare and input capture functions. The timer W can count external events and output pulses with an arbitrary duty cycle by compare match between the timer counter and four general registers. Thus, it can be applied to various systems.

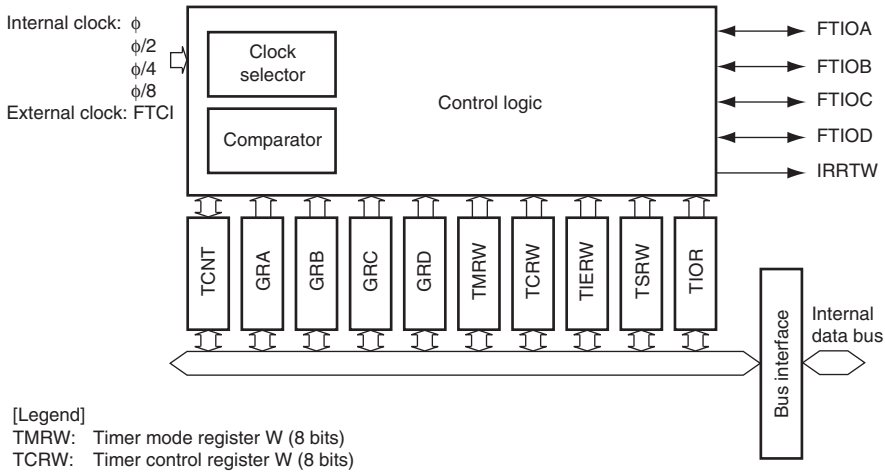
12.1 Features

- Selection of five counter clock sources: four internal clocks (ϕ, ϕ/2, ϕ/4, and ϕ/8) and an external clock (external events can be counted)
- Capability to process up to four pulse outputs or four pulse inputs
- Four general registers:
  - — Independently assignable output compare or input capture functions
  - — Usable as two pairs of registers; one register of each pair operates as a buffer for the output compare or input capture register
- Four selectable operating modes :
  - — Waveform output by compare match
    Selection of 0 output, 1 output, or toggle output
  - — Input capture function
    Rising edge, falling edge, or both edges
  - — Counter clearing function
    Counters can be cleared by compare match
  - — PWM mode
    Up to three-phase PWM output can be provided with desired duty ratio.
- Any initial timer output value can be set
- Five interrupt sources
  Four compare match/input capture interrupts and an overflow interrupt.

Table 12.1 summarizes the timer W functions, and figure 12.1 shows a block diagram of the timer W.

Table 12.1 Timer W Functions

Item		Counter	Input/Output Pins			
			FTIOA	FTIOB	FTIOC	FTIOD
Count clock		Internal clocks: φ, φ/2, φ/4, φ/8 External clock: FTCI				
General registers (output compare/input capture registers)		Period specified in GRA	GRA	GRB	GRC (buffer register for GRA in buffer mode)	GRD (buffer register for GRB in buffer mode)
Counter clearing function		GRA compare match	GRA compare match	—	—	—
Initial output value setting function		—	Yes	Yes	Yes	Yes
Buffer function		—	Yes	Yes	—	—
Compare match output	0	—	Yes	Yes	Yes	Yes
	1	—	Yes	Yes	Yes	Yes
	Toggle	—	Yes	Yes	Yes	Yes
Input capture function		—	Yes	Yes	Yes	Yes
PWM mode		—	—	Yes	Yes	Yes
Interrupt sources		Overflow	Compare match/input capture	Compare match/input capture	Compare match/input capture	Compare match/input capture



[Legend]
TMRW: Timer mode register W (8 bits)
TCRW: Timer control register W (8 bits)
TIERW: Timer interrupt enable register W (8 bits)
TSRW: Timer status register W (8 bits)
TIOR: Timer I/O control register (8 bits)
TCNT: Timer counter (16 bits)
GRA: General register A (input capture/output compare register: 16 bits)
GRB: General register B (input capture/output compare register: 16 bits)
GRC: General register C (input capture/output compare register: 16 bits)
GRD: General register D (input capture/output compare register: 16 bits)
IRRTW: Timer W interrupt request

Figure 12.1 Timer W Block Diagram

12.2 Input/Output Pins

Table 12.2 summarizes the timer W pins.

Table 12.2 Pin Configuration

Name	Abbreviation	Input/Output	Function
External clock input	FTCI	Input	External clock input pin
Input capture/output compare A	FTIOA	Input/output	Output pin for GRA output compare or input pin for GRA input capture
Input capture/output compare B	FTIOB	Input/output	Output pin for GRB output compare, input pin for GRB input capture, or PWM output pin in PWM mode
Input capture/output compare C	FTIOC	Input/output	Output pin for GRC output compare, input pin for GRC input capture, or PWM output pin in PWM mode
Input capture/output compare D	FTIOD	Input/output	Output pin for GRD output compare, input pin for GRD input capture, or PWM output pin in PWM mode

12.3 Register Descriptions

The timer W has the following registers.

- Timer mode register W (TMRW)
- Timer control register W (TCRW)
- Timer interrupt enable register W (TIERW)
- Timer status register W (TSRW)
- Timer I/O control register 0 (TIOR0)
- Timer I/O control register 1 (TIOR1)
- Timer counter (TCNT)
- General register A (GRA)
- General register B (GRB)
- General register C (GRC)
- General register D (GRD)

12.3.1 Timer Mode Register W (TMRW)

TMRW selects the general register functions and the timer output mode.

Bit	Bit Name	Initial Value	R/W	Description
7	CTS	0	R/W	Counter Start The counter operation is halted when this bit is 0, while it can be performed when this bit is 1.
6	—	1	—	Reserved This bit is always read as 1.
5	BUFEB	0	R/W	Buffer Operation B Selects the GRD function. 0: GRD operates as an input capture/output compare register 1: GRD operates as the buffer register for GRB
4	BUFEA	0	R/W	Buffer Operation A Selects the GRC function. 0: GRC operates as an input capture/output compare register 1: GRC operates as the buffer register for GRA
3	—	1	—	Reserved This bit is always read as 1.
2	PWMD	0	R/W	PWM Mode D Selects the output mode of the FTIOD pin. 0: FTIOD operates normally (output compare output) 1: PWM output
1	PWMC	0	R/W	PWM Mode C Selects the output mode of the FTIOC pin. 0: FTIOC operates normally (output compare output) 1: PWM output
0	PWMB	0	R/W	PWM Mode B Selects the output mode of the FTIOB pin. 0: FTIOB operates normally (output compare output) 1: PWM output

12.3.2 Timer Control Register W (TCRW)

TCRW selects the timer counter clock source, selects a clearing condition, and specifies the timer output levels.

Bit	Bit Name	Initial Value	R/W	Description
7	CCLR	0	R/W	Counter Clear The TCNT value is cleared by compare match A when this bit is 1. When it is 0, TCNT operates as a free-running counter.
6	CKS2	0	R/W	Clock Select 2 to 0
5	CKS1	0	R/W	Select the TCNT clock source.
4	CKS0	0	R/W	000: Internal clock: counts on ϕ 001: Internal clock: counts on $\phi/2$ 010: Internal clock: counts on $\phi/4$ 011: Internal clock: counts on $\phi/8$ 1XX: Counts on rising edges of the external event (FTCI) When the internal clock source (ϕ) is selected, subclock sources are counted in subactive and subsleep modes.
3	TOD	0	R/W	Timer Output Level Setting D Sets the output value of the FTIOD pin until the first compare match D is generated. 0: Output value is 0* 1: Output value is 1*
2	TOC	0	R/W	Timer Output Level Setting C Sets the output value of the FTIOC pin until the first compare match C is generated. 0: Output value is 0* 1: Output value is 1*
1	TOB	0	R/W	Timer Output Level Setting B Sets the output value of the FTIOB pin until the first compare match B is generated. 0: Output value is 0* 1: Output value is 1*

Bit	Bit Name	Initial Value	R/W	Description
0	TOA	0	R/W	Timer Output Level Setting A Sets the output value of the FTIOA pin until the first compare match A is generated. 0: Output value is 0* 1: Output value is 1*

[Legend]

X: Don't care.

Note: * The change of the setting is immediately reflected in the output value.

12.3.3 Timer Interrupt Enable Register W (TIERW)

TIERW controls the timer W interrupt request.

Bit	Bit Name	Initial Value	R/W	Description
7	OVIE	0	R/W	Timer Overflow Interrupt Enable When this bit is set to 1, FOVI interrupt requested by OVF flag in TSRW is enabled.
6	—	1	—	Reserved
5	—	1	—	These bits are always read as 1.
4	—	1	—	
3	IMIED	0	R/W	Input Capture/Compare Match Interrupt Enable D When this bit is set to 1, IMID interrupt requested by IMFD flag in TSRW is enabled.
2	IMIEC	0	R/W	Input Capture/Compare Match Interrupt Enable C When this bit is set to 1, IMIC interrupt requested by IMFC flag in TSRW is enabled.
1	IMIEB	0	R/W	Input Capture/Compare Match Interrupt Enable B When this bit is set to 1, IMIB interrupt requested by IMFB flag in TSRW is enabled.
0	IMIEA	0	R/W	Input Capture/Compare Match Interrupt Enable A When this bit is set to 1, IMIA interrupt requested by IMFA flag in TSRW is enabled.

12.3.4 Timer Status Register W (TSRW)

TSRW shows the status of interrupt requests.

Bit	Bit Name	Initial Value	R/W	Description
7	OVF	0	R/W	Timer Overflow Flag [Setting condition] When TCNT overflows from H'FFFF to H'0000 [Clearing condition] Read OVF when OVF = 1, then write 0 in OVF
6	—	1	—	Reserved These bits are always read as 1.
5	—	1	—	
4	—	1	—	
3	IMFD	0	R/W	Input Capture/Compare Match Flag D [Setting conditions] • TCNT = GRD when GRD functions as an output compare register • The TCNT value is transferred to GRD by an input capture signal when GRD functions as an input capture register [Clearing condition] Read IMFD when IMFD = 1, then write 0 in IMFD
2	IMFC	0	R/W	Input Capture/Compare Match Flag C [Setting conditions] • TCNT = GRC when GRC functions as an output compare register • The TCNT value is transferred to GRC by an input capture signal when GRC functions as an input capture register [Clearing condition] Read IMFC when IMFC = 1, then write 0 in IMFC

Bit	Bit Name	Initial Value	R/W	Description
1	IMFB	0	R/W	Input Capture/Compare Match Flag B [Setting conditions] • TCNT = GRB when GRB functions as an output compare register • The TCNT value is transferred to GRB by an input capture signal when GRB functions as an input capture register [Clearing condition] Read IMFB when IMFB = 1, then write 0 in IMFB
0	IMFA	0	R/W	Input Capture/Compare Match Flag A [Setting conditions] • TCNT = GRA when GRA functions as an output compare register • The TCNT value is transferred to GRA by an input capture signal when GRA functions as an input capture register [Clearing condition] Read IMFA when IMFA = 1, then write 0 in IMFA

12.3.5 Timer I/O Control Register 0 (TIOR0)

TIOR0 selects the functions of GRA and GRB, and specifies the functions of the FTIOA and FTIOB pins.

Bit	Bit Name	Initial Value	R/W	Description
7	—	1	—	Reserved This bit is always read as 1.
6	IOB2	0	R/W	I/O Control B2 Selects the GRB function. 0: GRB functions as an output compare register 1: GRB functions as an input capture register

Bit	Bit Name	Initial Value	R/W	Description
5	IOB1	0	R/W	I/O Control B1 and B0
4	IOB0	0	R/W	When IOB2 = 0, 00: No output at compare match 01: 0 output to the FTIOB pin at GRB compare match 10: 1 output to the FTIOB pin at GRB compare match 11: Output toggles to the FTIOB pin at GRB compare match When IOB2 = 1, 00: Input capture at rising edge at the FTIOB pin 01: Input capture at falling edge at the FTIOB pin 1X: Input capture at rising and falling edges of the FTIOB pin
3	—	1	—	Reserved This bit is always read as 1.
2	IOA2	0	R/W	I/O Control A2 Selects the GRA function. 0: GRA functions as an output compare register 1: GRA functions as an input capture register
1	IOA1	0	R/W	I/O Control A1 and A0
0	IOA0	0	R/W	When IOA2 = 0, 00: No output at compare match 01: 0 output to the FTIOA pin at GRA compare match 10: 1 output to the FTIOA pin at GRA compare match 11: Output toggles to the FTIOA pin at GRA compare match When IOA2 = 1, 00: Input capture at rising edge of the FTIOA pin 01: Input capture at falling edge of the FTIOA pin 1X: Input capture at rising and falling edges of the FTIOA pin

[Legend]

X: Don't care.

12.3.6 Timer I/O Control Register 1 (TIOR1)

TIOR1 selects the functions of GRC and GRD, and specifies the functions of the FTIOC and FTIOD pins.

Bit	Bit Name	Initial Value	R/W	Description
7	—	1	—	Reserved This bit is always read as 1.
6	IOD2	0	R/W	I/O Control D2 Selects the GRD function. 0: GRD functions as an output compare register 1: GRD functions as an input capture register
5	IOD1	0	R/W	I/O Control D1 and D0
4	IOD0	0	R/W	When IOD2 = 0, 00: No output at compare match 01: 0 output to the FTIOD pin at GRD compare match 10: 1 output to the FTIOD pin at GRD compare match 11: Output toggles to the FTIOD pin at GRD compare match When IOD2 = 1, 00: Input capture at rising edge at the FTIOD pin 01: Input capture at falling edge at the FTIOD pin 1X: Input capture at rising and falling edges at the FTIOD pin
3	—	1	—	Reserved This bit is always read as 1.
2	IOC2	0	R/W	I/O Control C2 Selects the GRC function. 0: GRC functions as an output compare register 1: GRC functions as an input capture register

Bit	Bit Name	Initial Value	R/W	Description
1	IOC1	0	R/W	I/O Control C1 and C0
0	IOC0	0	R/W	When IOC2 = 0, 00: No output at compare match 01: 0 output to the FTIOC pin at GRC compare match 10: 1 output to the FTIOC pin at GRC compare match 11: Output toggles to the FTIOC pin at GRC compare match When IOC2 = 1, 00: Input capture to GRC at rising edge of the FTIOC pin 01: Input capture to GRC at falling edge of the FTIOC pin 1X: Input capture to GRC at rising and falling edges of the FTIOC pin

[Legend]

X: Don't care.

12.3.7 Timer Counter (TCNT)

TCNT is a 16-bit readable/writable up-counter. The clock source is selected by bits CKS2 to CKS0 in TCRW. TCNT can be cleared to H'0000 through a compare match with GRA by setting the CCLR in TCRW to 1. When TCNT overflows (changes from H'FFFF to H'0000), the OVF flag in TSRW is set to 1. If OVIE in TIERW is set to 1 at this time, an interrupt request is generated. TCNT must always be read or written in 16-bit units; 8-bit access is not allowed. TCNT is initialized to H'0000 by a reset.

12.3.8 General Registers A to D (GRA to GRD)

Each general register is a 16-bit readable/writable register that can function as either an output-compare register or an input-capture register. The function is selected by settings in TIOR0 and TIOR1.

When a general register is used as an input-compare register, its value is constantly compared with the TCNT value. When the two values match (a compare match), the corresponding flag (IMFA, IMFB, IMFC, or IMFD) in TSRW is set to 1. An interrupt request is generated at this time, when IMIEA, IMIEB, IMIEC, or IMIED is set to 1. Compare match output can be selected in TIOR.

When a general register is used as an input-capture register, an external input-capture signal is detected and the current TCNT value is stored in the general register. The corresponding flag (IMFA, IMFB, IMFC, or IMFD) in TSRW is set to 1. If the corresponding interrupt-enable bit (IMIEA, IMIEB, IMIEC, or IMIED) in TSRW is set to 1 at this time, an interrupt request is generated. The edge of the input-capture signal is selected in TIOR.

GRC and GRD can be used as buffer registers of GRA and GRB, respectively, by setting BUFEA and BUFEB in TMRW.

For example, when GRA is set as an output-compare register and GRC is set as the buffer register for GRA, the value in the buffer register GRC is sent to GRA whenever compare match A is generated.

When GRA is set as an input-capture register and GRC is set as the buffer register for GRA, the value in TCNT is transferred to GRA and the value in the buffer register GRC is transferred to GRA whenever an input capture is generated.

GRA to GRD must be written or read in 16-bit units; 8-bit access is not allowed. GRA to GRD are initialized to H'FFFF by a reset.

12.4 Operation

The timer W has the following operating modes.

- Normal Operation
- PWM Operation

12.4.1 Normal Operation

TCNT performs free-running or periodic counting operations. After a reset, TCNT is set as a free-running counter. When the CTS bit in TMRW is set to 1, TCNT starts incrementing the count. When the count overflows from H'FFFF to H'0000, the OVF flag in TSRW is set to 1. If the OVIE in TIERW is set to 1, an interrupt request is generated. Figure 12.2 shows free-running counting.

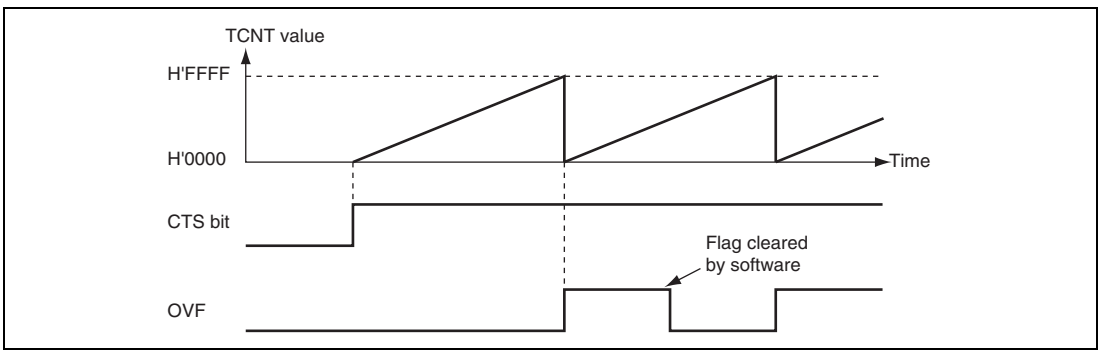


Figure 12.2 Free-Running Counter Operation

Periodic counting operation can be performed when GRA is set as an output compare register and bit CCLR in TCRW is set to 1. When the count matches GRA, TCNT is cleared to H'0000, the IMFA flag in TSRW is set to 1. If the corresponding IMIEA bit in TIERW is set to 1, an interrupt request is generated. TCNT continues counting from H'0000. Figure 12.3 shows periodic counting.

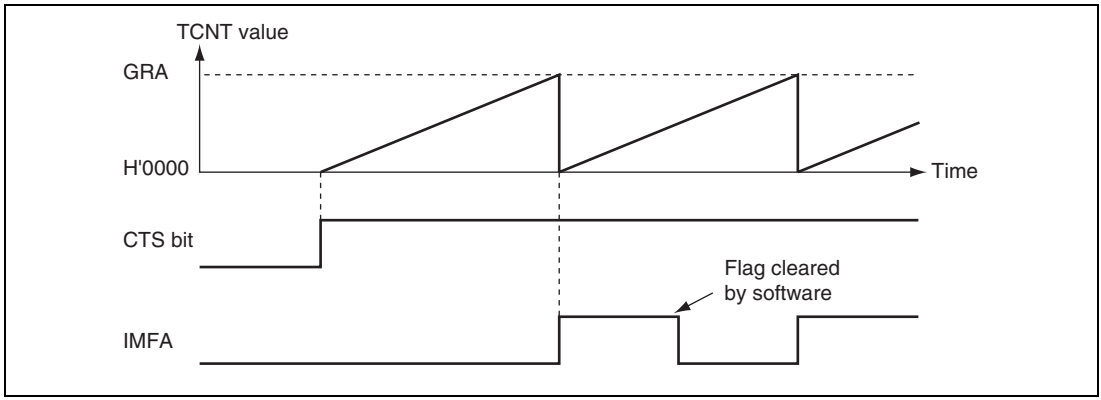


Figure 12.3 Periodic Counter Operation

By setting a general register as an output compare register, compare match A, B, C, or D can cause the output at the FTIOA, FTIOB, FTIOC, or FTIOD pin to output 0, output 1, or toggle. Figure 12.4 shows an example of 0 and 1 output when TCNT operates as a free-running counter, 1 output is selected for compare match A, and 0 output is selected for compare match B. When signal is already at the selected output level, the signal level does not change at compare match.

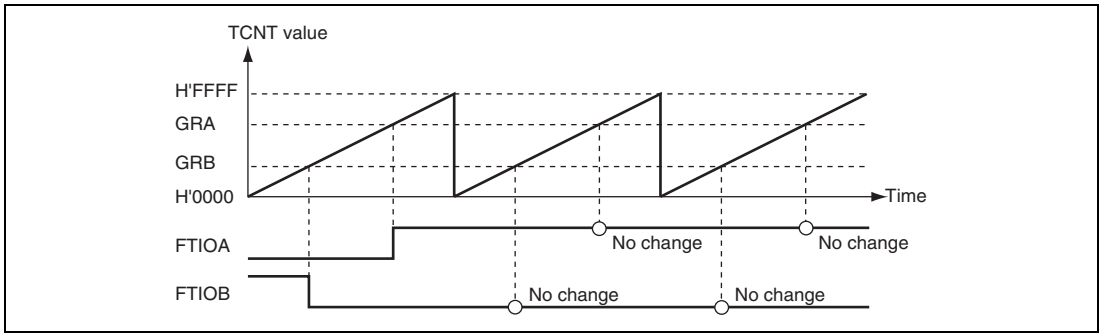


Figure 12.4 0 and 1 Output Example (TOA = 0, TOB = 1)

Figure 12.5 shows an example of toggle output when TCNT operates as a free-running counter, and toggle output is selected for both compare match A and B.

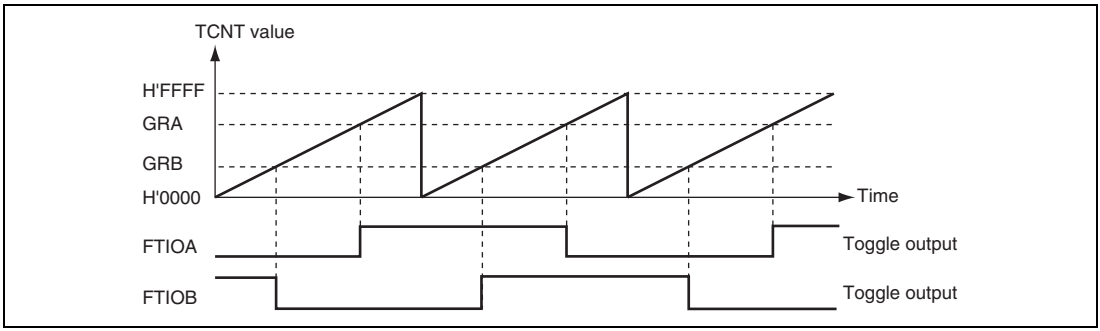


Figure 12.5 Toggle Output Example (TOA = 0, TOB = 1)

Figure 12.6 shows another example of toggle output when TCNT operates as a periodic counter, cleared by compare match A. Toggle output is selected for both compare match A and B.

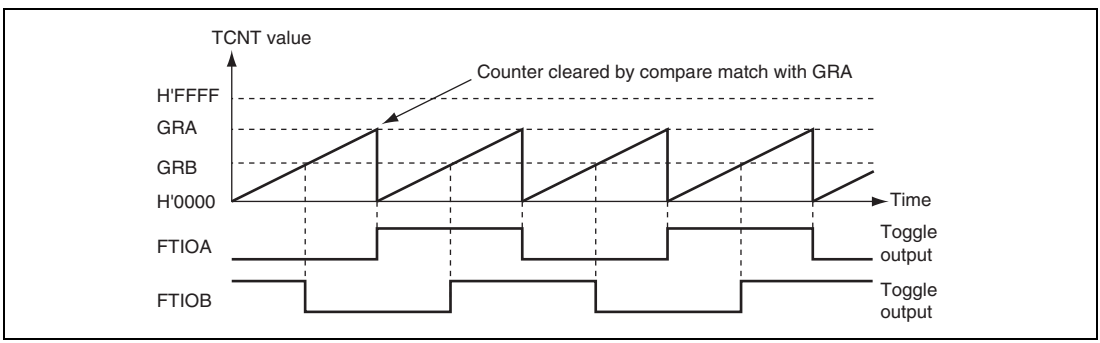


Figure 12.6 Toggle Output Example (TOA = 0, TOB = 1)

The TCNT value can be captured into a general register (GRA, GRB, GRC, or GRD) when a signal level changes at an input-capture pin (FTIOA, FTIOB, FTIOC, or FTIOD). Capture can take place on the rising edge, falling edge, or both edges. By using the input-capture function, the pulse width and periods can be measured. Figure 12.7 shows an example of input capture when both edges of FTIOA and the falling edge of FTIOB are selected as capture edges. TCNT operates as a free-running counter.

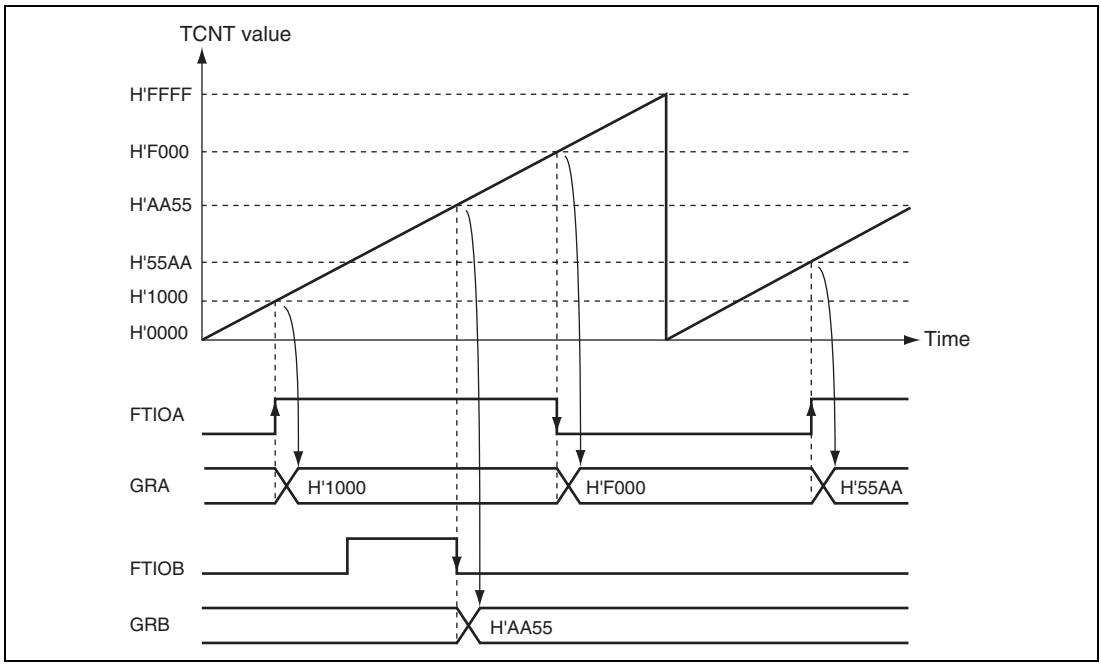


Figure 12.7 Input Capture Operating Example

Figure 12.8 shows an example of buffer operation when the GRA is set as an input-capture register and GRC is set as the buffer register for GRA. TCNT operates as a free-running counter, and FTIOA captures both rising and falling edge of the input signal. Due to the buffer operation, the GRA value is transferred to GRC by input-capture A and the TCNT value is stored in GRA.

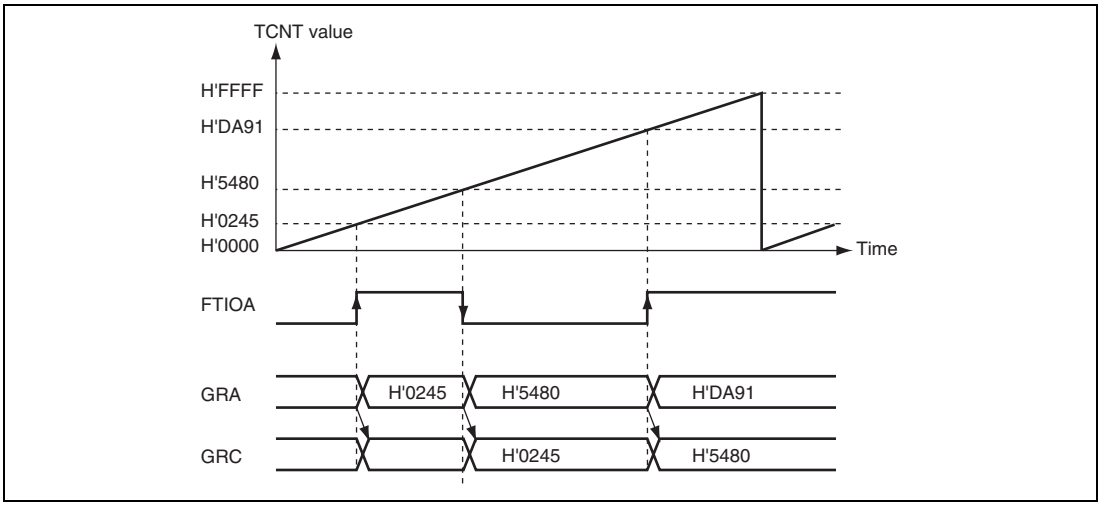


Figure 12.8 Buffer Operation Example (Input Capture)

12.4.2 PWM Operation

In PWM mode, PWM waveforms are generated by using GRA as the period register and GRB, GRC, and GRD as duty registers. PWM waveforms are output from the FTIOB, FTIOC, and FTIOD pins. Up to three-phase PWM waveforms can be output. In PWM mode, a general register functions as an output compare register automatically. The output level of each pin depends on the corresponding timer output level set bit (TOB, TOC, and TOD) in TCRW. When TOB is 1, the FTIOB output goes to 1 at compare match A and to 0 at compare match B. When TOB is 0, the FTIOB output goes to 0 at compare match A and to 1 at compare match B. Thus the compare match output level settings in TIOR0 and TIOR1 are ignored for the output pin set to PWM mode. If the same value is set in the cycle register and the duty register, the output does not change when a compare match occurs.

Figure 12.9 shows an example of operation in PWM mode. The output signals go to 1 and TCNT is cleared at compare match A, and the output signals go to 0 at compare match B, C, and D (TOB, TOC, and TOD = 1: initial output values are set to 1).

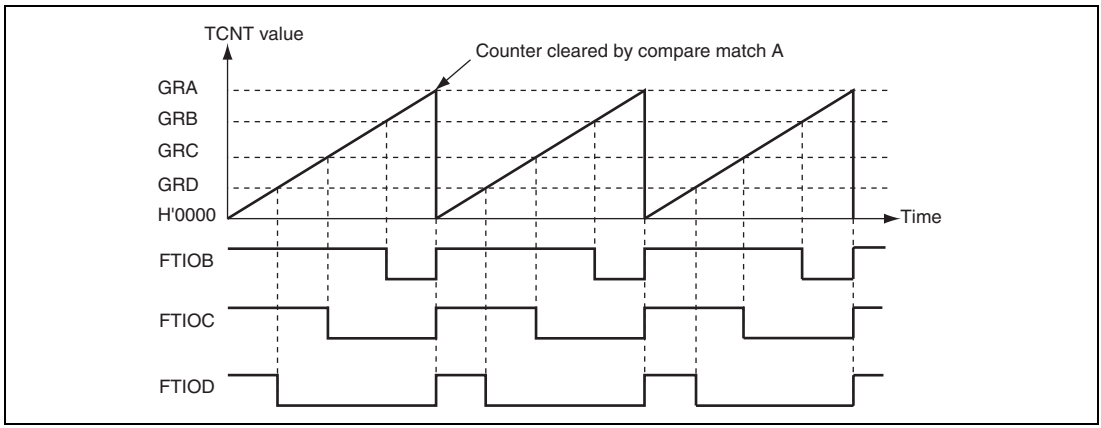


Figure 12.9 PWM Mode Example (1)

Figure 12.10 shows another example of operation in PWM mode. The output signals go to 0 and TCNT is cleared at compare match A, and the output signals go to 1 at compare match B, C, and D (TOB, TOC, and TOD = 0: initial output values are set to 1).

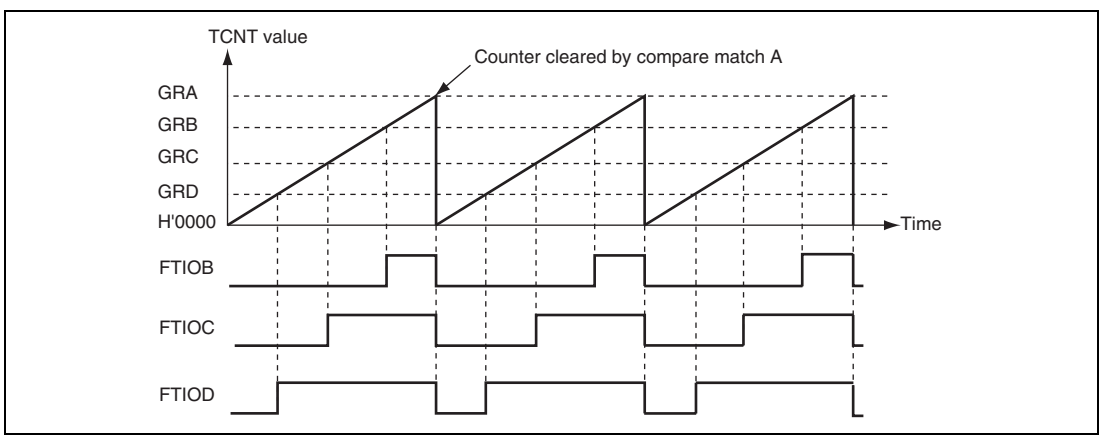


Figure 12.10 PWM Mode Example (2)

Figure 12.11 shows an example of buffer operation when the FTIOB pin is set to PWM mode and GRD is set as the buffer register for GRB. TCNT is cleared by compare match A, and FTIOB outputs 1 at compare match B and 0 at compare match A.

Due to the buffer operation, the FTIOB output level changes and the value of buffer register GRD is transferred to GRB whenever compare match B occurs. This procedure is repeated every time compare match B occurs.

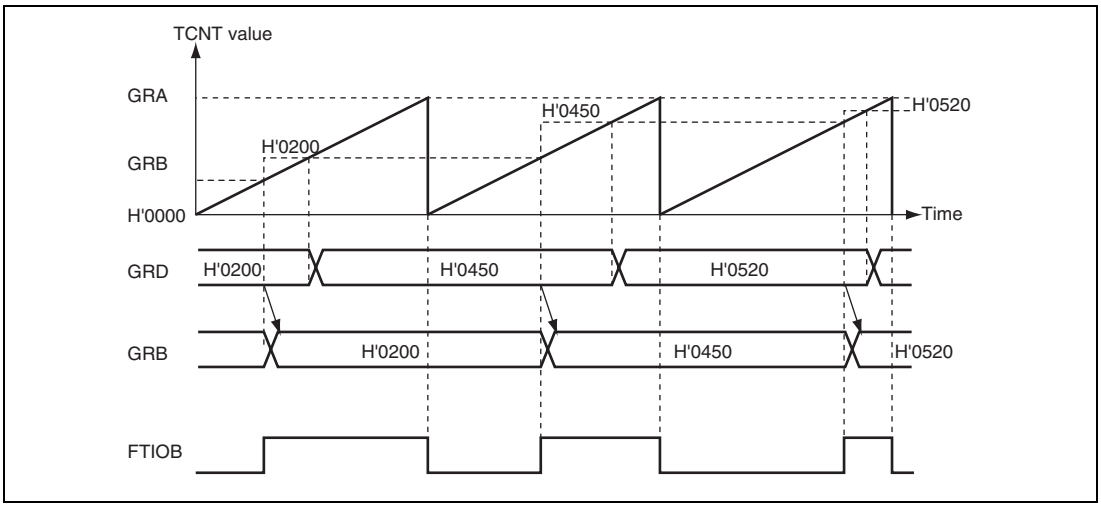


Figure 12.11 Buffer Operation Example (Output Compare)

Figures 12.12 and 12.13 show examples of the output of PWM waveforms with duty cycles of 0% and 100%.

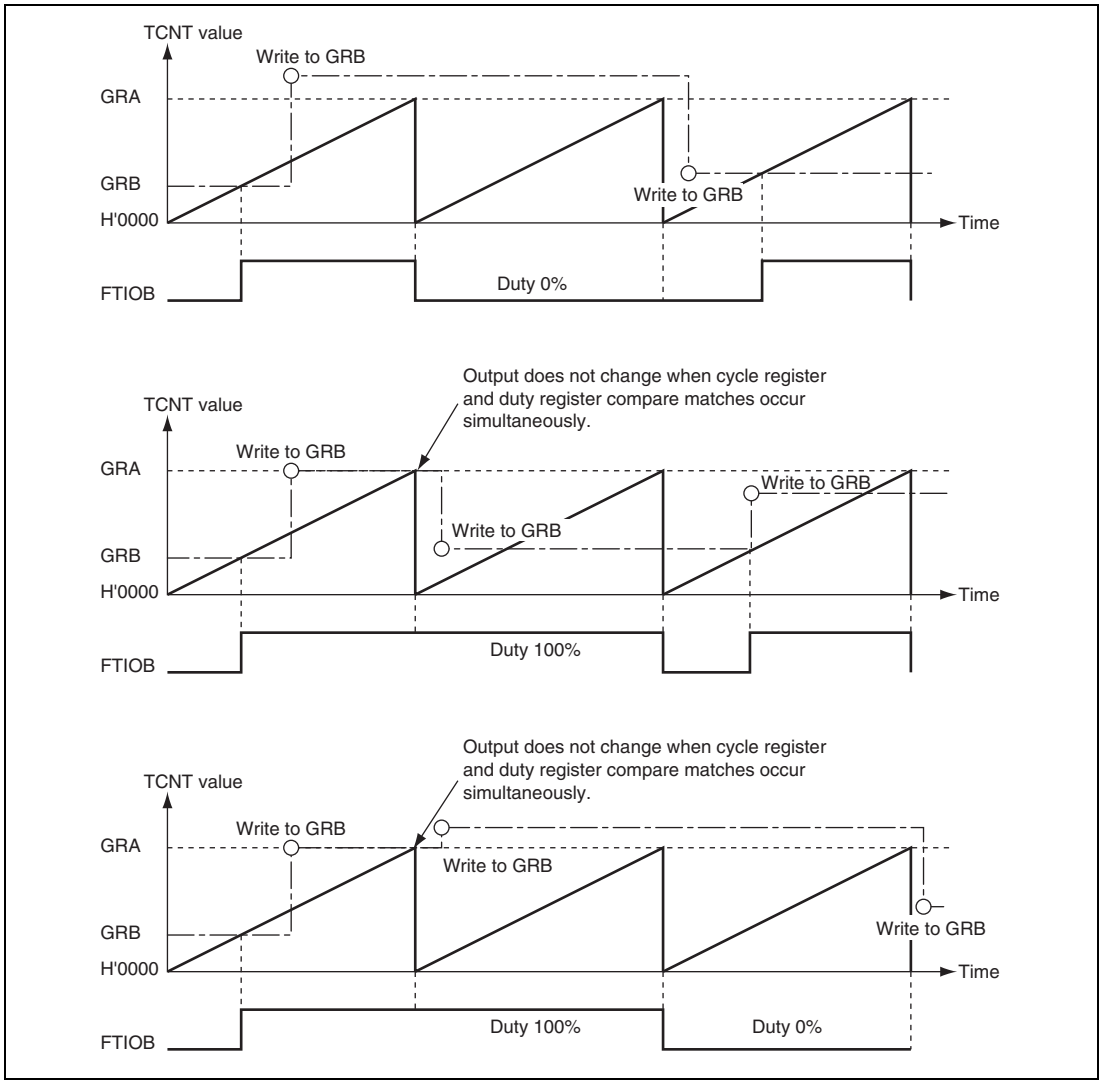


Figure 12.12 PWM Mode Example
(TOB, TOC, and TOD = 0: initial output values are set to 0)

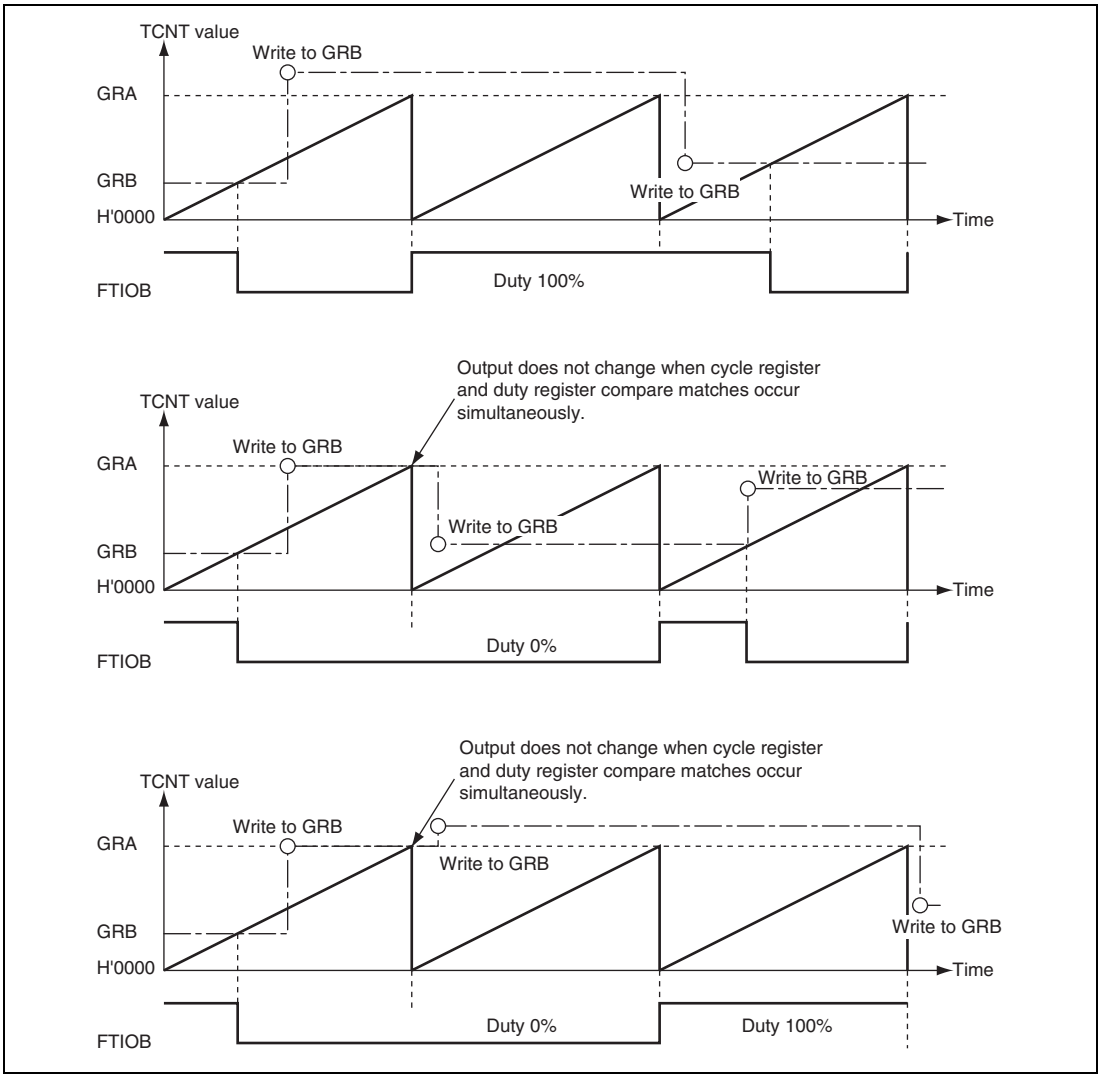


Figure 12.13 PWM Mode Example
(TOB, TOC, and TOD = 1: initial output values are set to 1)

12.5 Operation Timing

12.5.1 TCNT Count Timing

Figure 12.14 shows the TCNT count timing when the internal clock source is selected. Figure 12.15 shows the timing when the external clock source is selected. The pulse width of the external clock signal must be at least two system clock (ϕ) cycles; shorter pulses will not be counted correctly.

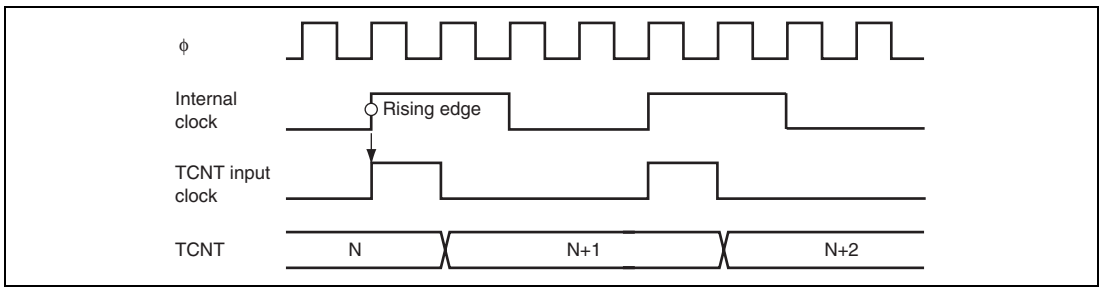


Figure 12.14 Count Timing for Internal Clock Source

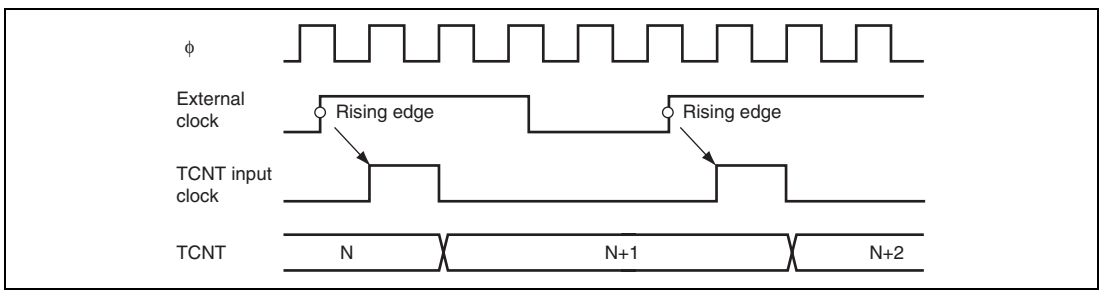


Figure 12.15 Count Timing for External Clock Source

12.5.2 Output Compare Output Timing

The compare match signal is generated in the last state in which TCNT and GR match (when TCNT changes from the matching value to the next value). When the compare match signal is generated, the output value selected in TIOR is output at the compare match output pin (FTIOA, FTIOB, FTIOC, or FTIOD).

When TCNT matches GR, the compare match signal is generated only after the next counter clock pulse is input.

Figure 12.16 shows the output compare timing.

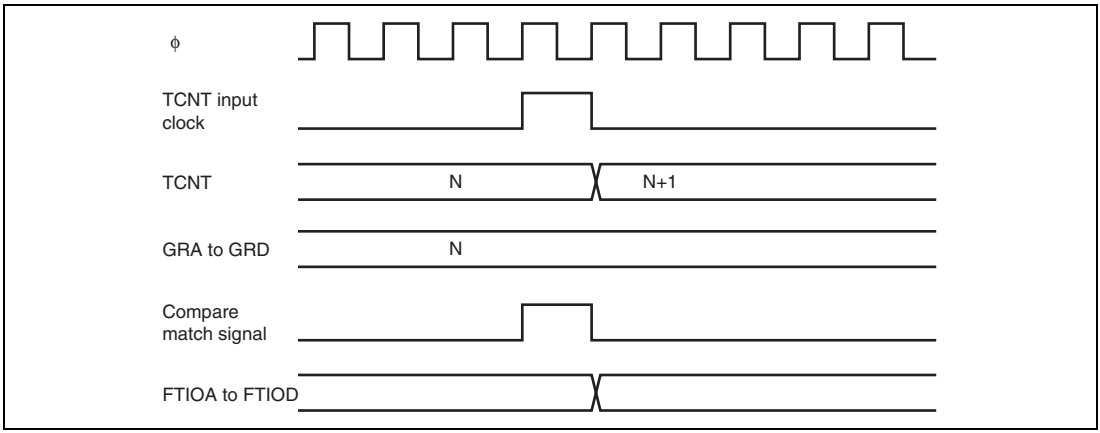


Figure 12.16 Output Compare Output Timing

12.5.3 Input Capture Timing

Input capture on the rising edge, falling edge, or both edges can be selected through settings in TIOR0 and TIOR1. Figure 12.17 shows the timing when the falling edge is selected. The pulse width of the input capture signal must be at least two system clock (ϕ) cycles; shorter pulses will not be detected correctly.

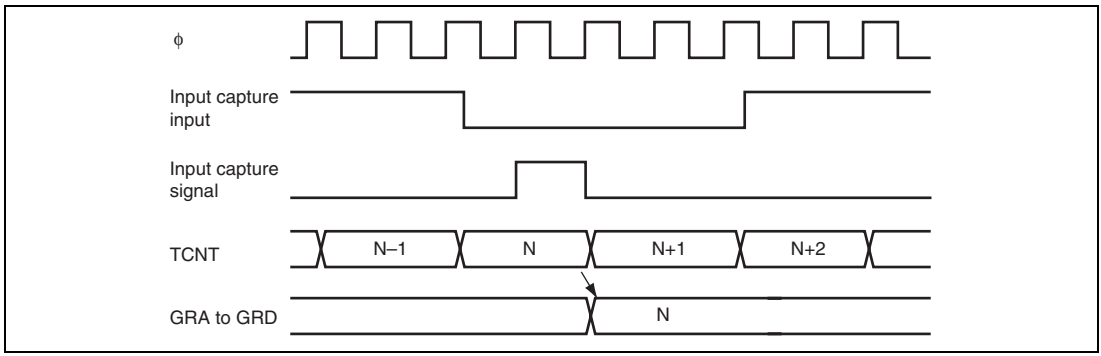


Figure 12.17 Input Capture Input Signal Timing

12.5.4 Timing of Counter Clearing by Compare Match

Figure 12.18 shows the timing when the counter is cleared by compare match A. When the GRA value is N, the counter counts from 0 to N, and its cycle is N + 1.

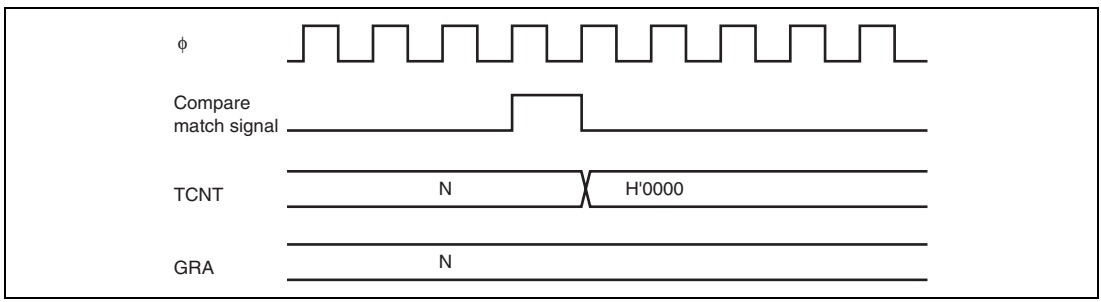


Figure 12.18 Timing of Counter Clearing by Compare Match

12.5.5 Buffer Operation Timing

Figures 12.19 and 12.20 show the buffer operation timing.

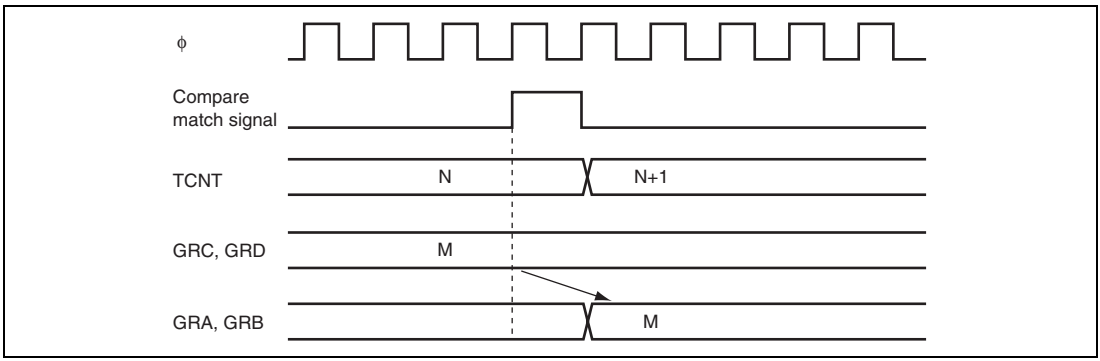


Figure 12.19 Buffer Operation Timing (Compare Match)

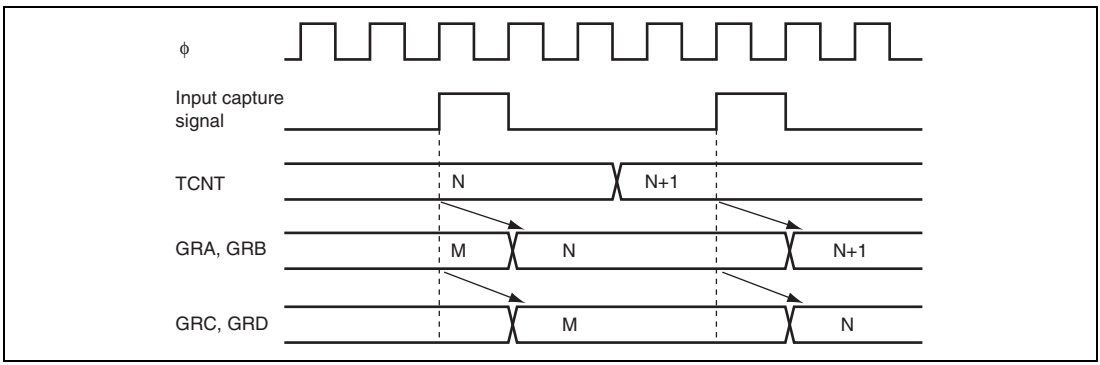


Figure 12.20 Buffer Operation Timing (Input Capture)

12.5.6 Timing of IMFA to IMFD Flag Setting at Compare Match

If a general register (GRA, GRB, GRC, or GRD) is used as an output compare register, the corresponding IMFA, IMFB, IMFC, or IMFD flag is set to 1 when TCNT matches the general register.

The compare match signal is generated in the last state in which the values match (when TCNT is updated from the matching count to the next count). Therefore, when TCNT matches a general register, the compare match signal is generated only after the next TCNT clock pulse is input.

Figure 12.21 shows the timing of the IMFA to IMFD flag setting at compare match.

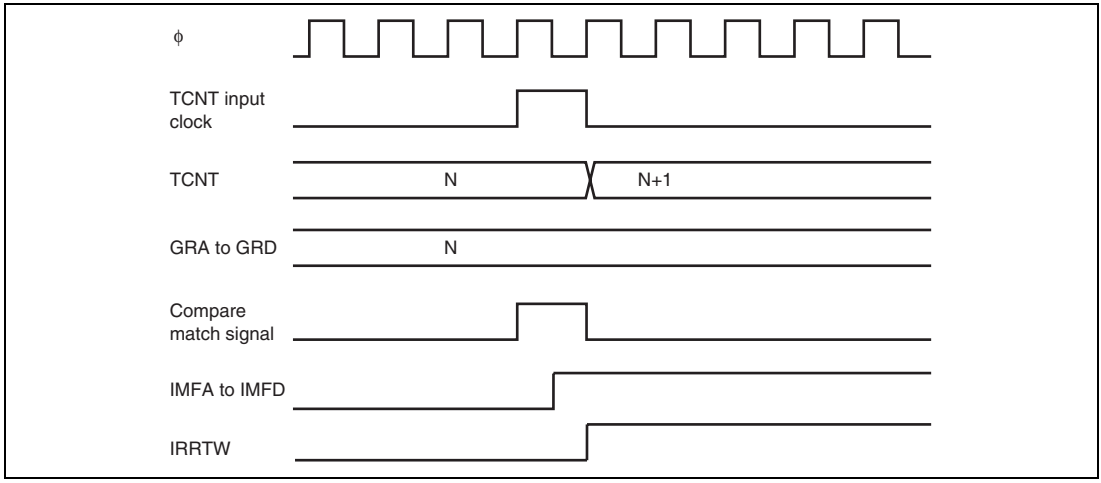


Figure 12.21 Timing of IMFA to IMFD Flag Setting at Compare Match

12.5.7 Timing of IMFA to IMFD Setting at Input Capture

If a general register (GRA, GRB, GRC, or GRD) is used as an input capture register, the corresponding IMFA, IMFB, IMFC, or IMFD flag is set to 1 when an input capture occurs. Figure 12.22 shows the timing of the IMFA to IMFD flag setting at input capture.

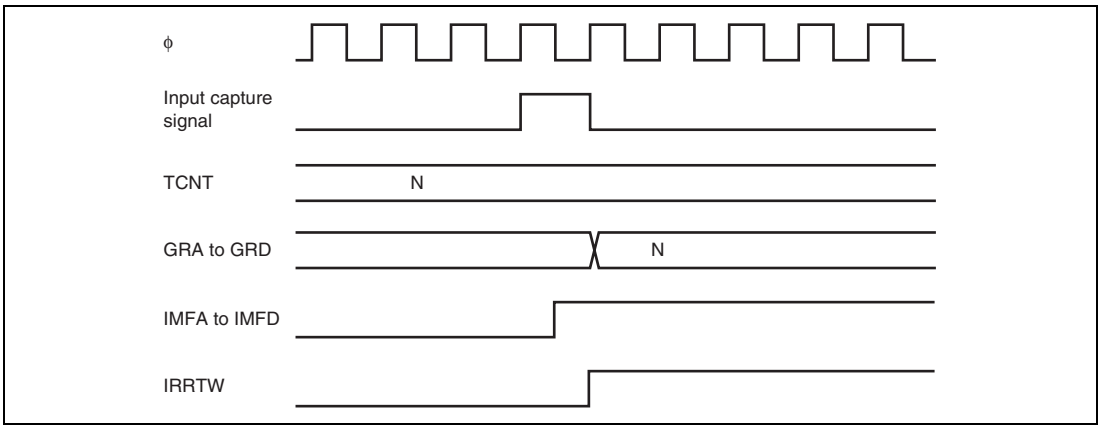


Figure 12.22 Timing of IMFA to IMFD Flag Setting at Input Capture

12.5.8 Timing of Status Flag Clearing

When the CPU reads a status flag while it is set to 1, then writes 0 in the status flag, the status flag is cleared. Figure 12.23 shows the status flag clearing timing.

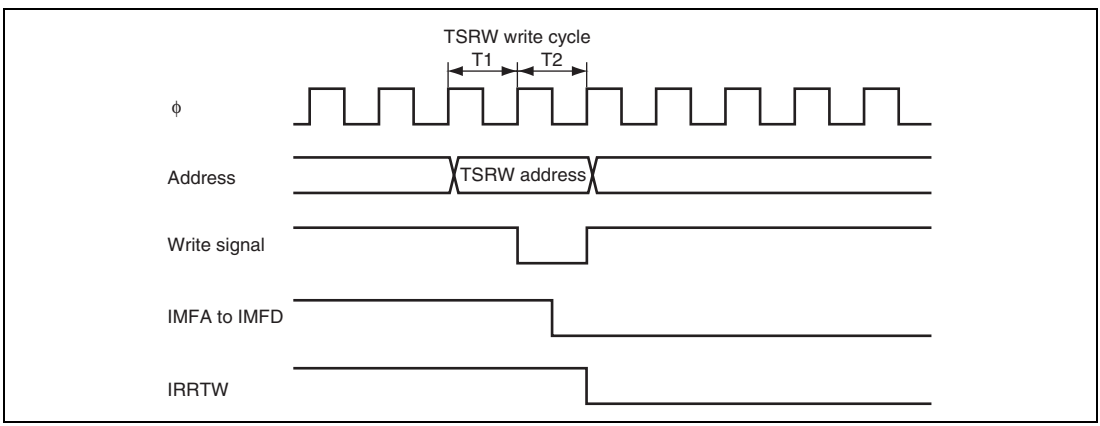


Figure 12.23 Timing of Status Flag Clearing by CPU

12.6 Usage Notes

The following types of contention or operation can occur in timer W operation.

1. The pulse width of the input clock signal and the input capture signal must be at least two system clock (ϕ) cycles; shorter pulses will not be detected correctly.
2. Writing to registers is performed in the T2 state of a TCNT write cycle.

   If counter clear signal occurs in the T2 state of a TCNT write cycle, clearing of the counter takes priority and the write is not performed, as shown in figure 12.24. If counting-up is generated in the TCNT write cycle to contend with the TCNT counting-up, writing takes precedence.
3. Depending on the timing, TCNT may be incremented by a switch between different internal clock sources. When TCNT is internally clocked, an increment pulse is generated from the rising edge of an internal clock signal, that is divided system clock (ϕ). Therefore, as shown in figure 12.25 the switch is from a low clock signal to a high clock signal, the switchover is seen as a rising edge, causing TCNT to increment.
4. If timer W enters module standby mode while an interrupt request is generated, the interrupt request cannot be cleared. Before entering module standby mode, disable interrupt requests.

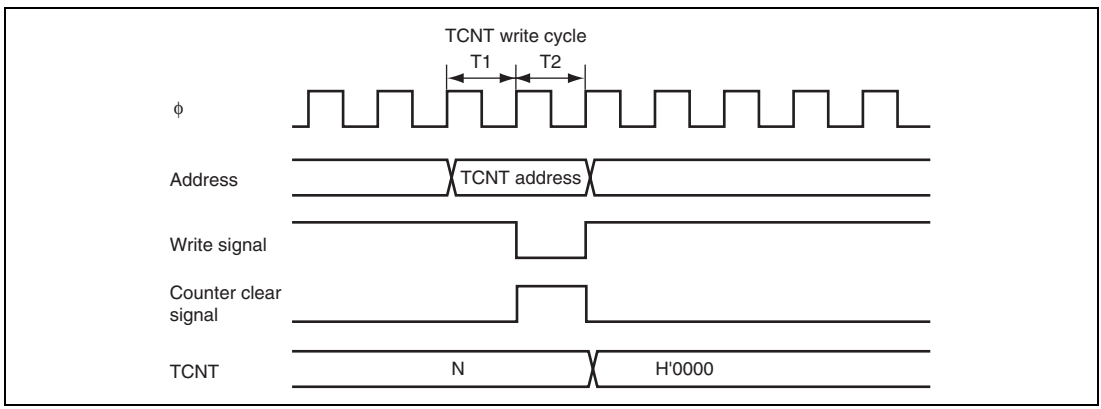


Figure 12.24 Contention between TCNT Write and Clear

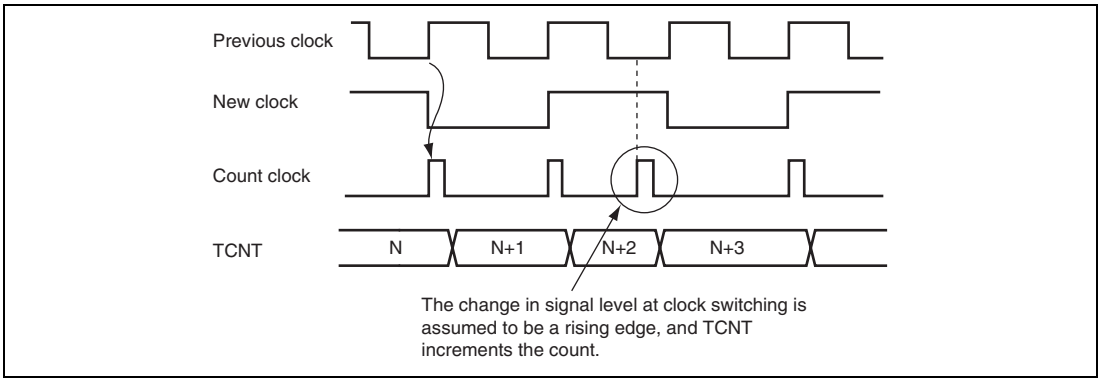


Figure 12.25 Internal Clock Switching and TCNT Operation

5. The TOA to TOD bits in TCRW decide the value of the FTIO pin, which is output until the first compare match occurs. Once a compare match occurs and this compare match changes the values of FTIOA to FTIOD output, the values of the FTIOA to FTIOD pin output and the values read from the TOA to TOD bits may differ. Moreover, when the writing to TCRW and the generation of the compare match A to D occur at the same timing, the writing to TCRW has the priority. Thus, output change due to the compare match is not reflected to the FTIOA to FTIOD pins. Therefore, when bit manipulation instruction is used to write to TCRW, the values of the FTIOA to FTIOD pin output may result in an unexpected result. When TCRW is to be written to while compare match is operating, stop the counter once before accessing to TCRW, read the port 8 state to reflect the values of FTIOA to FTIOD output, to TOA to TOD, and then restart the counter. Figure 12.26 shows an example when the compare match and the bit manipulation instruction to TCRW occur at the same timing.

TCRW has been set to H'06. Compare match B and compare match C are used. The FTIOB pin is the 1 output state, and is set to the toggle output or the 0 output on compare match B.
When BCLR#2, @TCRW is executed to clear the TOC bit (the FTIOC signal is low) and compare match B occurs at the same timing as shown below, the H'02 writing to TCRW has priority and compare match B does not drive the FTIOB signal low; the FTIOB signal remains high.

Bit	7	6	5	4	3	2	1	0
TRCCR1	CCLR	CKS2	CKS1	CKS0	TOD	TOC	TOB	TOA
Set value	0	0	0	0	0	1	1	0

BCLR#2, @TCRW
(1) TCRW read operation: Read H'06
(2) Modify operation: Modify H'06 to H'02
(3) Write operation to TCRW: Write H'02

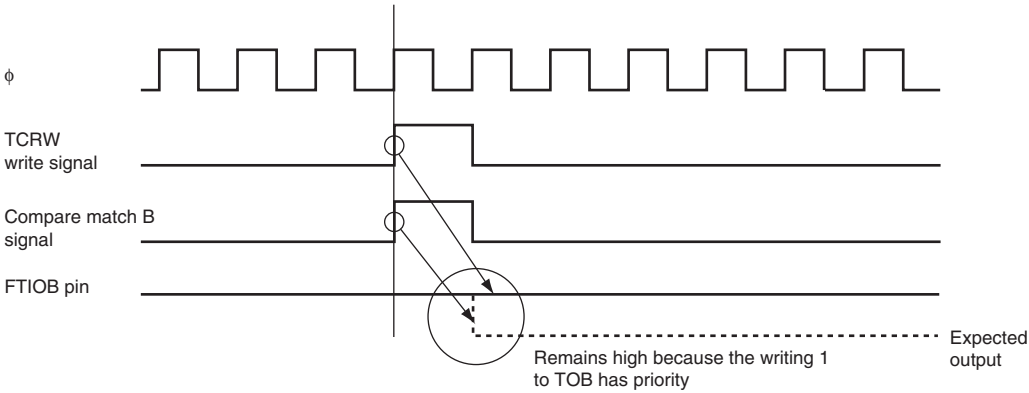


Figure 12.26 When Compare Match and Bit Manipulation Instruction to TCRW Occur at the Same Timing

Section 13 Watchdog Timer

The watchdog timer is an 8-bit timer that can generate an internal reset signal for this LSI if a system crash prevents the CPU from writing to the timer counter, thus allowing it to overflow.

The block diagram of the watchdog timer is shown in figure 13.1.

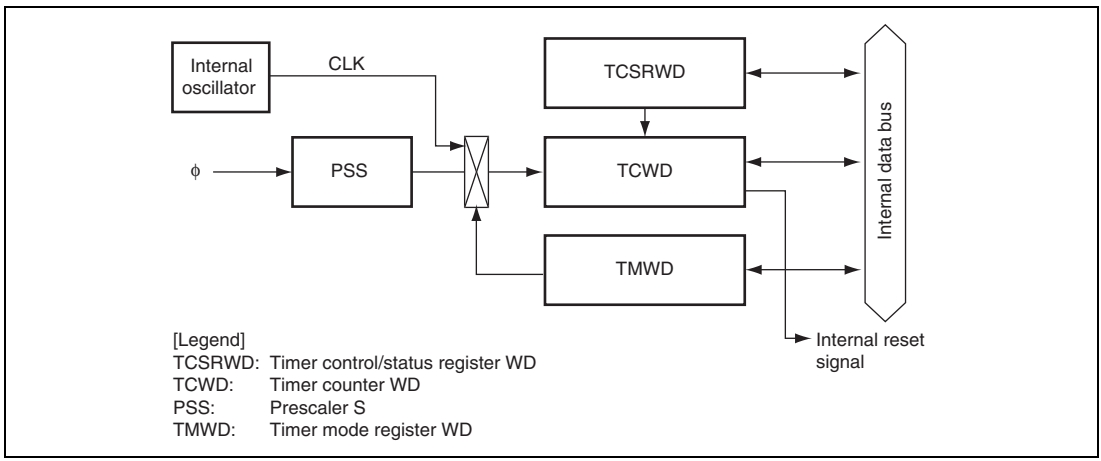


Figure 13.1 Block Diagram of Watchdog Timer

13.1 Features

- Selectable from nine counter input clocks.
  Eight clock sources (ϕ/64, ϕ/128, ϕ/256, ϕ/512, ϕ/1024, ϕ/2048, ϕ/4096, and ϕ/8192) or the internal oscillator can be selected as the timer-counter clock. When the internal oscillator is selected, it can operate as the watchdog timer in any operating mode.
- Reset signal generated on counter overflow
  An overflow period of 1 to 256 times the selected clock can be set.

13.2 Register Descriptions

The watchdog timer has the following registers.

- Timer control/status register WD (TCSRWD)
- Timer counter WD (TCWD)
- Timer mode register WD (TMWD)

13.2.1 Timer Control/Status Register WD (TCSRWD)

TCSRWD performs the TCSRWD and TCWD write control. TCSRWD also controls the watchdog timer operation and indicates the operating state. TCSRWD must be rewritten by using the MOV instruction. The bit manipulation instruction cannot be used to change the setting value.

Bit	Bit Name	Initial Value	R/W	Description
7	B6WI	1	R/W	Bit 6 Write Inhibit The TCWE bit can be written only when the write value of the B6WI bit is 0. This bit is always read as 1.
6	TCWE	0	R/W	Timer Counter WD Write Enable TCWD can be written when the TCWE bit is set to 1. When writing data to this bit, the value for bit 7 must be 0.
5	B4WI	1	R/W	Bit 4 Write Inhibit The TCSRWE bit can be written only when the write value of the B4WI bit is 0. This bit is always read as 1.
4	TCSRWE	0	R/W	Timer Control/Status Register WD Write Enable The WDON and WRST bits can be written when the TCSRWE bit is set to 1. When writing data to this bit, the value for bit 5 must be 0.
3	B2WI	1	R/W	Bit 2 Write Inhibit This bit can be written to the WDON bit only when the write value of the B2WI bit is 0. This bit is always read as 1.

Bit	Bit Name	Initial Value	R/W	Description
2	WDON	0	R/W	Watchdog Timer On TCWD starts counting up when WDON is set to 1 and halts when WDON is cleared to 0. [Setting condition] When 1 is written to the WDON bit while writing 0 to the B2WI bit when the TCSRWE bit=1 [Clearing conditions] • Reset by $\overline{RES}$ pin • When 0 is written to the WDON bit while writing 0 to the B2WI when the TCSRWE bit=1
1	B0WI	1	R/W	Bit 0 Write Inhibit This bit can be written to the WRST bit only when the write value of the B0WI bit is 0. This bit is always read as 1.
0	WRST	0	R/W	Watchdog Timer Reset [Setting condition] When TCWD overflows and an internal reset signal is generated [Clearing conditions] • Reset by $\overline{RES}$ pin • When 0 is written to the WRST bit while writing 0 to the B0WI bit when the TCSRWE bit=1

13.2.2 Timer Counter WD (TCWD)

TCWD is an 8-bit readable/writable up-counter. When TCWD overflows from H'FF to H'00, the internal reset signal is generated and the WRST bit in TCSRWD is set to 1. TCWD is initialized to H'00.

13.2.3 Timer Mode Register WD (TMWD)

TMWD selects the input clock.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	—	Reserved These bits are always read as 1.
3	CKS3	1	R/W	Clock Select 3 to 0
2	CKS2	1	R/W	Select the clock to be input to TCWD.
1	CKS1	1	R/W	1000: Internal clock: counts on ϕ/64
0	CKS0	1	R/W	1001: Internal clock: counts on ϕ/128 1010: Internal clock: counts on ϕ/256 1011: Internal clock: counts on ϕ/512 1100: Internal clock: counts on ϕ/1024 1101: Internal clock: counts on ϕ/2048 1110: Internal clock: counts on ϕ/4096 1111: Internal clock: counts on ϕ8192 0XXX: Internal oscillator For the internal oscillator overflow periods, see section 20, Electrical Characteristics.

[Legend]
X: Don't care.

13.3 Operation

The watchdog timer is provided with an 8-bit counter. If 1 is written to WDON while writing 0 to B2WI when the TCSRWE bit in TCSRWD is set to 1, TCWD begins counting up. (To operate the watchdog timer, two write accesses to TCSRWD are required.) When a clock pulse is input after the TCWD count value has reached H'FF, the watchdog timer overflows and an internal reset signal is generated. The internal reset signal is output for a period of 256 ϕ_{osc} clock cycles. TCWD is a writable counter, and when a value is set in TCWD, the count-up starts from that value. An overflow period in the range of 1 to 256 input clock cycles can therefore be set, according to the TCWD set value.

Figure 13.2 shows an example of watchdog timer operation.

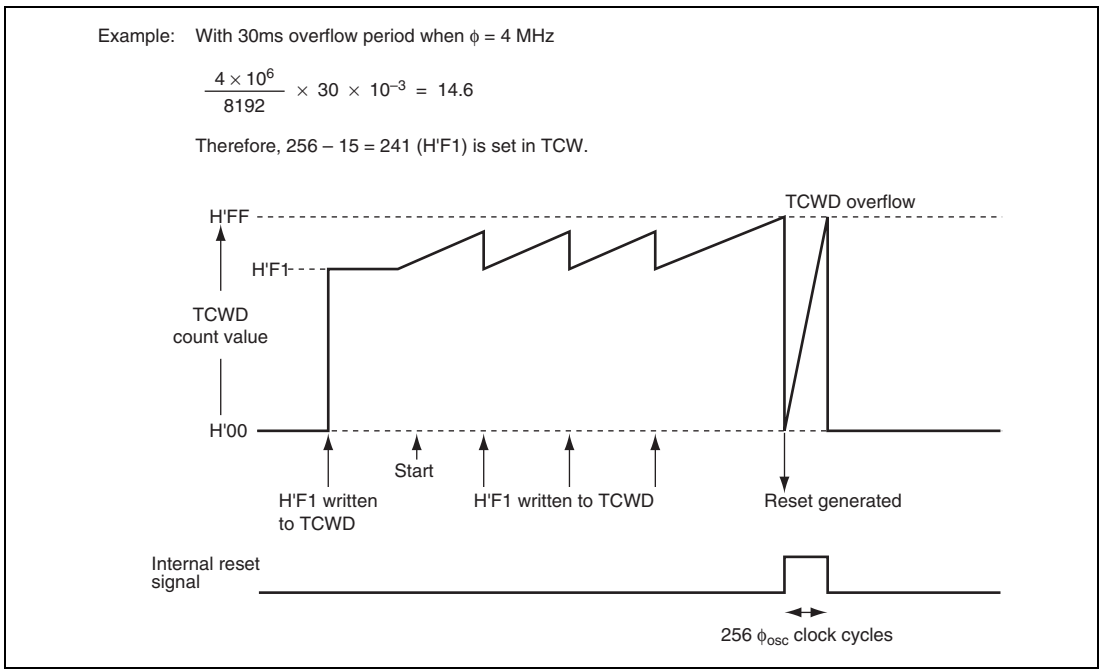


Figure 13.2 Watchdog Timer Operation Example

Section 14 Serial Communication Interface 3 (SCI3)

Serial Communication Interface 3 (SCI3) can handle both asynchronous and clocked synchronous serial communication. In the asynchronous method, serial data communication can be carried out using standard asynchronous communication chips such as a Universal Asynchronous Receiver/Transmitter (UART) or an Asynchronous Communication Interface Adapter (ACIA). A function is also provided for serial communication between processors (multiprocessor communication function).

Figure 14.1 shows a block diagram of the SCI3.

14.1 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
- External clock or on-chip baud rate generator can be selected as a transfer clock source.
- Six interrupt sources

  Transmit-end, transmit-data-empty, receive-data-full, overrun error, framing error, and parity error.

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 the case of a framing error

Clocked synchronous mode

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

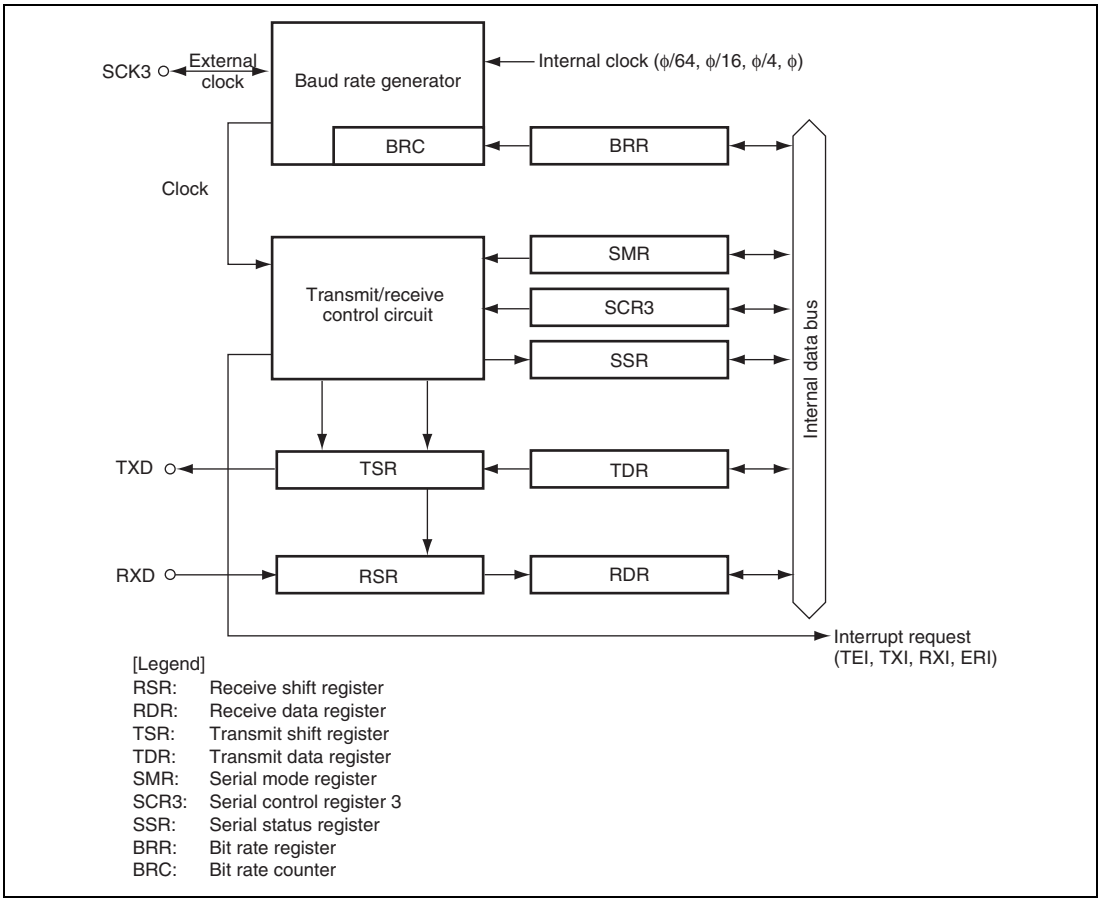


Figure 14.1 Block Diagram of SCI3

14.2 Input/Output Pins

Table 14.1 shows the SCI3 pin configuration.

Table 14.1 Pin Configuration

Pin Name	Abbreviation	I/O	Function
SCI3 clock	SCK3	I/O	SCI3 clock input/output
SCI3 receive data input	RXD	Input	SCI3 receive data input
SCI3 transmit data output	TXD	Output	SCI3 transmit data output

14.3 Register Descriptions

The SCI3 has the following registers.

- Receive shift register (RSR)
- Receive data register (RDR)
- Transmit shift register (TSR)
- Transmit data register (TDR)
- Serial mode register (SMR)
- Serial control register 3 (SCR3)
- Serial status register (SSR)
- Bit rate register (BRR)

14.3.1 Receive Shift Register (RSR)

RSR is a shift register that is used to receive serial data input from the RxD pin and convert it into parallel data. When one byte of data has been received, it is transferred to RDR automatically. RSR cannot be directly accessed by the CPU.

14.3.2 Receive Data Register (RDR)

RDR is an 8-bit register that stores received data. When the SCI3 has received one byte of serial data, it transfers the received serial data from RSR to RDR, where it is stored. After this, RSR is receive-enabled. As RSR and RDR function as a double buffer in this way, continuous receive operations are possible. After confirming that the RDRF bit in SSR is set to 1, read RDR only once. RDR cannot be written to by the CPU. RDR is initialized to H'00.

14.3.3 Transmit Shift Register (TSR)

TSR is a shift register that transmits serial data. To perform serial data transmission, the SCI3 first transfers transmit data from TDR to TSR automatically, then sends the data that starts from the LSB to the TXD pin. TSR cannot be directly accessed by the CPU.

14.3.4 Transmit Data Register (TDR)

TDR is an 8-bit register that stores data for transmission. When the SCI3 detects that TSR is empty, it transfers the transmit data written in TDR to TSR and starts transmission. The double-buffered structure of TDR and TSR enables continuous serial transmission. If the next transmit data has already been written to TDR during transmission of one-frame data, the SCI3 transfers the written data to TSR to continue transmission. To achieve reliable serial transmission, write transmit data to TDR only once after confirming that the TDRE bit in SSR is set to 1. TDR is initialized to H'FF.

14.3.5 Serial Mode Register (SMR)

SMR is used to set the SCI3's serial transfer format and select the on-chip baud rate generator clock source.

Bit	Bit Name	Initial Value	R/W	Description
7	COM	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.
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.
4	PM	0	R/W	Parity Mode (enabled only when the PE bit is 1 in asynchronous mode) 0: Selects even parity. 1: Selects odd parity.

Bit	Bit Name	Initial Value	R/W	Description
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 For reception, only the first stop bit is checked, regardless of the value in the bit. If the second stop bit is 0, it is treated as the start bit of the next transmit character.
2	MP	0	R/W	Multiprocessor Mode When this bit is set to 1, the multiprocessor communication function is enabled. The PE bit and PM bit settings are invalid. In clocked synchronous mode, this bit should be cleared to 0.
1	CKS1	0	R/W	Clock Select 0 and 1
0	CKS0	0	R/W	These bits select the clock source for the on-chip 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 relationship between the bit rate register setting and the baud rate, see section 14.3.8, Bit Rate Register (BRR). n is the decimal representation of the value of n in BRR (see section 14.3.8, Bit Rate Register (BRR)).

14.3.6 Serial Control Register 3 (SCR3)

SCR3 is a register that enables or disables SCI3 transfer operations and interrupt requests, and is also used to select the transfer clock source. For details on interrupt requests, refer to section 14.7, Interrupts.

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable When this bit is set to 1, the TXI interrupt request is enabled.

Bit	Bit Name	Initial Value	R/W	Description
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 OER status flags in SSR is prohibited. On receiving data in which the multiprocessor bit is 1, this bit is automatically cleared and normal reception is resumed. For details, refer to section 14.6, Multiprocessor Communication Function.
2	TEIE	0	R/W	Transmit End Interrupt Enable When this bit is set to 1, the TEI interrupt request is enabled.
1	CKE1	0	R/W	Clock Enable 0 and 1
0	CKE0	0	R/W	Selects the clock source. Asynchronous mode: 00: Internal baud rate generator 01: Internal baud rate generator Outputs a clock of the same frequency as the bit rate from the SCK3 pin. 10: External clock Inputs a clock with a frequency 16 times the bit rate from the SCK3 pin. 11:Reserved Clocked synchronous mode: 00: Internal clock (SCK3 pin functions as clock output) 01: Reserved 10: External clock (SCK3 pin functions as clock input) 11: Reserved

14.3.7 Serial Status Register (SSR)

SSR is a register containing status flags of the SCI3 and multiprocessor bits for transfer. 1 cannot be written to flags TDRE, RDRF, OER, PER, and FER; they can only be cleared.

Bit	Bit Name	Initial Value	R/W	Description
7	TDRE	1	R/W	Transmit Data Register Empty Displays whether TDR contains transmit data. [Setting conditions] • When the TE bit in SCR3 is 0 • When data is transferred from TDR to TSR [Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the transmit data is written to TDR
6	RDRF	0	R/W	Receive Data Register Full Indicates that the received 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 data is read from RDR
5	OER	0	R/W	Overrun Error [Setting condition] When an overrun error occurs in reception [Clearing condition] When 0 is written to OER after reading OER = 1
4	FER	0	R/W	Framing Error [Setting condition] When a framing error occurs in reception [Clearing condition] When 0 is written to FER after reading FER = 1

Bit	Bit Name	Initial Value	R/W	Description
3	PER	0	R/W	Parity Error [Setting condition] When a parity error is generated 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 SCR3 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 TEND after reading TEND = 1 • When the transmit data is written to TDR
1	MPBR	0	R	Multiprocessor Bit Receive MPBR stores the multiprocessor bit in the receive character data. When the RE bit in SCR3 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 character data.

14.3.8 Bit Rate Register (BRR)

BRR is an 8-bit register that adjusts the bit rate. The initial value of BRR is H'FF. Table 14.2 shows the relationship between the N setting in BRR and the n setting in bits CKS1 and CKS0 of SMR in asynchronous mode. Table 14.3 shows the maximum bit rate for each frequency in asynchronous mode. The values shown in both tables 14.2 and 14.3 are values in active (high-speed) mode. Table 14.4 shows the relationship between the N setting in BRR and the n setting in bits CKS1 and CKS0 in SMR in clocked synchronous mode. The values shown in table 14.4 are values in active (high-speed) mode. The N setting in BRR and error for other operating frequencies and bit rates can be obtained by the following formulas:

[Asynchronous Mode]

$$N = \frac{\phi}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

$$\text{Error (\%)} = \left\{ \frac{\phi \times 10^6}{(N + 1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$

[Clocked Synchronous Mode]

$$N = \frac{\phi}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

[Legend]

B: Bit rate (bit/s)

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

ϕ: Operating frequency (MHz)

n: CKS1 and CKS0 setting for SMR ($0 \leq N \leq 3$)

Table 14.2 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (1)

	Operating Frequency φ (MHz)											
	2			2.097152			2.4576			3		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	1	141	0.03	1	148	–0.04	1	174	–0.26	1	212	0.03
150	1	103	0.16	1	108	0.21	1	127	0.00	1	155	0.16
300	0	207	0.16	0	217	0.21	0	255	0.00	1	77	0.16
600	0	103	0.16	0	108	0.21	0	127	0.00	0	155	0.16
1200	0	51	0.16	0	54	–0.70	0	63	0.00	0	77	0.16
2400	0	25	0.16	0	26	1.14	0	31	0.00	0	38	0.16
4800	0	12	0.16	0	13	–2.48	0	15	0.00	0	19	–2.34
9600	0	6	–6.99	0	6	–2.48	0	7	0.00	0	9	–2.34
19200	0	2	8.51	0	2	13.78	0	3	0.00	0	4	–2.34
31250	0	1	0.00	0	1	4.86	0	1	22.88	0	2	0.00
38400	0	1	–18.62	0	1	–14.67	0	1	0.00	—	—	—

[Legend]

— : A setting is available but error occurs

	Operating Frequency φ (MHz)											
	3.6864			4			4.9152			5		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	64	0.70	2	70	0.03	2	86	0.31	2	88	–0.25
150	1	191	0.00	1	207	0.16	1	255	0.00	2	64	0.16
300	1	95	0.00	1	103	0.16	1	127	0.00	1	129	0.16
600	0	191	0.00	0	207	0.16	0	255	0.00	1	64	0.16
1200	0	95	0.00	0	103	0.16	0	127	0.00	0	129	0.16
2400	0	47	0.00	0	51	0.16	0	63	0.00	0	64	0.16
4800	0	23	0.00	0	25	0.16	0	31	0.00	0	32	–1.36
9600	0	11	0.00	0	12	0.16	0	15	0.00	0	15	1.73
19200	0	5	0.00	0	6	–6.99	0	7	0.00	0	7	1.73
31250	—	—	—	0	3	0.00	0	4	–1.70	0	4	0.00
38400	0	2	0.00	0	2	8.51	0	3	0.00	0	3	1.73

Table 14.2 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (2)

	Operating Frequency ϕ (MHz)											
	6			6.144			7.3728			8		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	106	–0.44	2	108	0.08	2	130	–0.07	2	141	0.03
150	2	77	0.16	2	79	0.00	2	95	0.00	2	103	0.16
300	1	155	0.16	1	159	0.00	1	191	0.00	1	207	0.16
600	1	77	0.16	1	79	0.00	1	95	0.00	1	103	0.16
1200	0	155	0.16	0	159	0.00	0	191	0.00	0	207	0.16
2400	0	77	0.16	0	79	0.00	0	95	0.00	0	103	0.16
4800	0	38	0.16	0	39	0.00	0	47	0.00	0	51	0.16
9600	0	19	–2.34	0	19	0.00	0	23	0.00	0	25	0.16
19200	0	9	–2.34	0	9	0.00	0	11	0.00	0	12	0.16
31250	0	5	0.00	0	5	2.40	0	6	5.33	0	7	0.00
38400	0	4	–2.34	0	4	0.00	0	5	0.00	0	6	-6.99

	Operating Frequency ϕ (MHz)											
	9.8304			10			12			12.888		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	174	–0.26	2	177	–0.25	2	212	0.03	2	217	0.08
150	2	127	0.00	2	129	0.16	2	155	0.16	2	159	0.00
300	1	255	0.00	2	64	0.16	2	77	0.16	2	79	0.00
600	1	127	0.00	1	129	0.16	1	155	0.16	1	159	0.00
1200	0	255	0.00	1	64	0.16	1	77	0.16	1	79	0.00
2400	0	127	0.00	0	129	0.16	0	155	0.16	0	159	0.00
4800	0	63	0.00	0	64	0.16	0	77	0.16	0	79	0.00
9600	0	31	0.00	0	32	–1.36	0	38	0.16	0	39	0.00
19200	0	15	0.00	0	15	1.73	0	19	–2.34	0	19	0.00
31250	0	9	–1.70	0	9	0.00	0	11	0.00	0	11	2.40
38400	0	7	0.00	0	7	1.73	0	9	–2.34	0	9	0.00

Table 14.2 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (3)

	Operating Frequency ϕ (MHz)								
	14			14.7456			16		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	248	–0.17	3	64	0.70	3	70	0.03
150	2	181	0.16	2	191	0.00	2	207	0.16
300	2	90	0.16	2	95	0.00	2	103	0.16
600	1	181	0.16	1	191	0.00	1	207	0.16
1200	1	90	0.16	1	95	0.00	1	103	0.16
2400	0	181	0.16	0	191	0.00	0	207	0.16
4800	0	90	0.16	0	95	0.00	0	103	0.16
9600	0	45	–0.93	0	47	0.00	0	51	0.16
19200	0	22	–0.93	0	23	0.00	0	25	0.16
31250	0	13	0.00	0	14	–1.70	0	15	0.00
38400	—	—	—	0	11	0.00	0	12	0.16

[Legend]

—: A setting is available but error occurs.

Table 14.3 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
2	62500	0	0	7.3728	230400	0	0
2.097152	65536	0	0	8	250000	0	0
2.4576	76800	0	0	9.8304	307200	0	0
3	93750	0	0	10	312500	0	0
3.6864	115200	0	0	12	375000	0	0
4	125000	0	0	12.288	384000	0	0
4.9152	153600	0	0	14	437500	0	0
5	156250	0	0	14.7456	460800	0	0
6	187500	0	0	16	500000	0	0
6.144	192000	0	0				

Table 14.4 BRR Settings for Various Bit Rates (Clocked Synchronous Mode)

Bit Rate (bit/s)	Operating Frequency ϕ (MHz)									
	2		4		8		10		16	
	n	N	n	N	n	N	n	N	n	N
110	3	70	—	—	—	—	—	—		
250	2	124	2	249	3	124	—	—	3	249
500	1	249	2	124	2	249	—	—	3	124
1k	1	124	1	249	2	124	—	—	2	249
2.5k	0	199	1	99	1	199	1	249	2	99
5k	0	99	0	199	1	99	1	124	1	199
10k	0	49	0	99	0	199	0	249	1	99
25k	0	19	0	39	0	79	0	99	0	159
50k	0	9	0	19	0	39	0	49	0	79
100k	0	4	0	9	0	19	0	24	0	39
250k	0	1	0	3	0	7	0	9	0	15
500k	0	0*	0	1	0	3	0	4	0	7
1M			0	0*	0	1	—	—	0	3
2M					0	0*	—	—	0	1
2.5M							0	0*	—	—
4M									0	0*

[Legend]

Blank: No setting is available.

—: A setting is available but error occurs.

*: Continuous transfer is not possible.

14.4 Operation in Asynchronous Mode

Figure 14.2 shows the general format for asynchronous serial communication. One frame consists of a start bit (low level), followed by data (in LSB-first order), a parity bit (high or low level), and finally stop bits (high level). Inside the SCI3, the transmitter and receiver are independent units, enabling full duplex. Both the transmitter and the receiver also have a double-buffered structure, so data can be read or written during transmission or reception, enabling continuous data transfer.

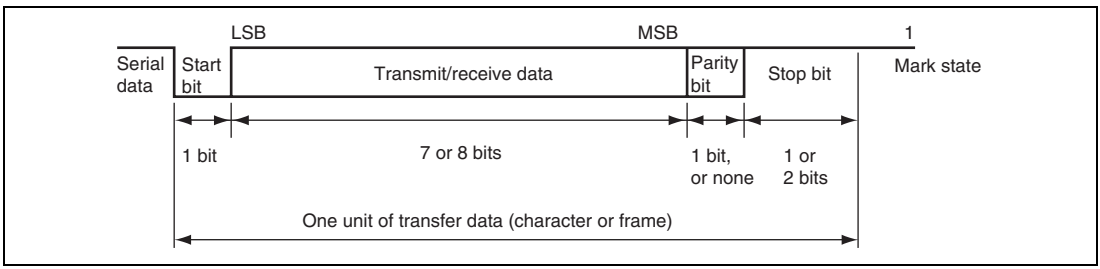


Figure 14.2 Data Format in Asynchronous Communication

14.4.1 Clock

Either an internal clock generated by the on-chip baud rate generator or an external clock input at the SCK3 pin can be selected as the SCI3's serial clock source, according to the setting of the COM bit in SMR and the CKE0 and CKE1 bits in SCR3. When an external clock is input at the SCK3 pin, the clock frequency should be 16 times the bit rate used.

When the SCI3 is operated on an internal clock, the clock can be output from the SCK3 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 14.3.

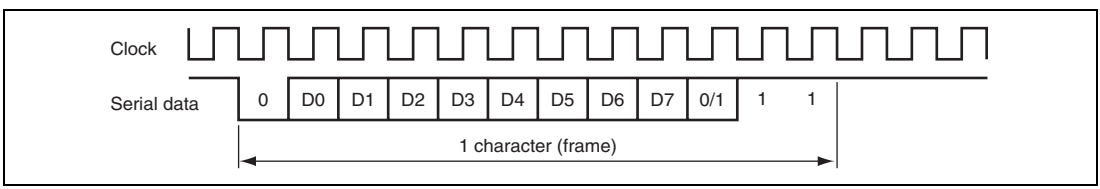
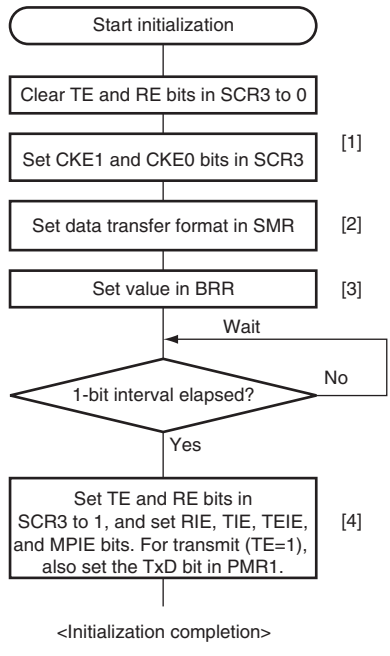


Figure 14.3 Relationship between Output Clock and Transfer Data Phase (Asynchronous Mode) (Example with 8-Bit Data, Parity, Two Stop Bits)

14.4.2 SCI3 Initialization

Follow the flowchart as shown in figure 14.4 to initialize the SCI3. When the TE bit is cleared to 0, the TDRE flag is set to 1. Note that clearing the RE bit to 0 does not initialize the contents of the RDRF, PER, FER, and OER flags, 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 SCR3. Be sure to clear bits RIE, TIE, TEIE, and MPIE, and bits TE and RE, to 0.

When the clock output is selected in asynchronous mode, clock is output immediately after CKE1 and CKE0 settings are made. When the clock output is selected at reception in clocked synchronous mode, clock is output immediately after CKE1, CKE0, and RE are set to 1.

[2] Set the data transfer format in SMR.

[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 SCR3 to 1. RE settings enable the RXD pin to be used. For transmission, set the TXD bit in PMR1 to 1 to enable the TXD output pin to be used. Also set the RIE, TIE, TEIE, and MPIE bits, depending on whether interrupts are required. In asynchronous mode, the bits are marked at transmission and idled at reception to wait for the start bit.

Figure 14.4 Sample SCI3 Initialization Flowchart

14.4.3 Data Transmission

Figure 14.5 shows an example of operation for transmission in asynchronous mode. In transmission, the SCI3 operates as described below.

1. The SCI3 monitors the TDRE flag in SSR. If the flag is cleared to 0, the SCI3 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 SCI3 sets the TDRE flag to 1 and starts transmission. If the TIE bit is set to 1 at this time, a TXI interrupt request is generated. Continuous transmission is possible because the TXI interrupt routine writes next transmit data to TDR before transmission of the current transmit data has been completed.
3. The SCI3 checks the TDRE flag at the timing for sending the stop bit.
4. 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.
5. 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 SCR3 is set to 1 at this time, a TEI interrupt request is generated.
6. Figure 14.6 shows a sample flowchart for transmission in asynchronous mode.

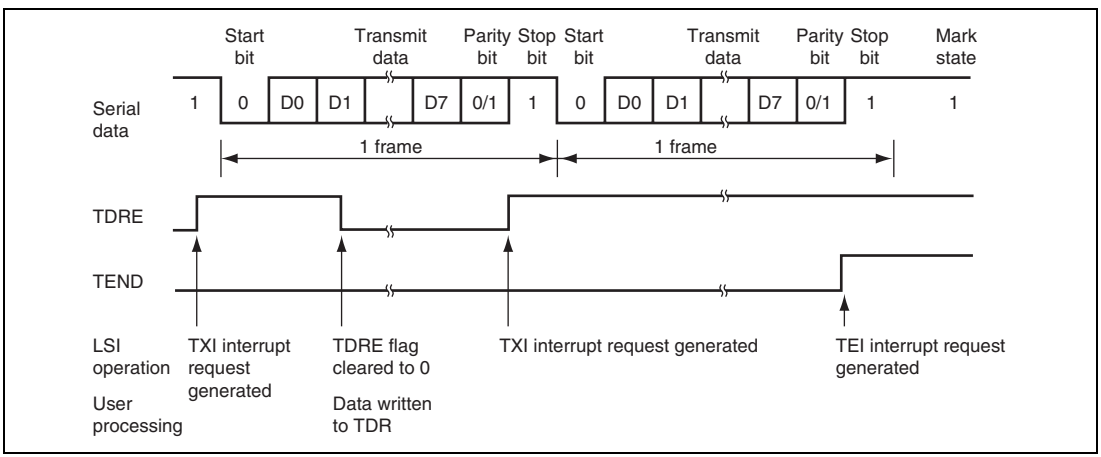
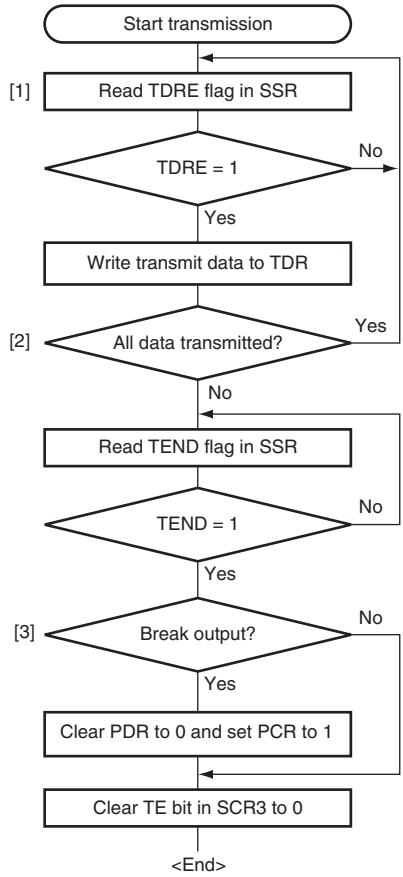


Figure 14.5 Example SCI3 Operation in Transmission in Asynchronous Mode (8-Bit Data, Parity, One Stop Bit)



[1] Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR. When data is written to TDR, the TDRE flag is automaticaly cleared to 0.

[2] To continue serial transmission, read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR. When data is written to TDR, the TDRE flag is automaticaly cleared to 0.

[3] To output a break in serial transmission, after setting PCR to 1 and PDR to 0, clear TxD in PMR1 to 0, then clear the TE bit in SCR3 to 0.

Figure 14.6 Sample Serial Transmission Flowchart (Asynchronous Mode)

14.4.4 Serial Data Reception

Figure 14.7 shows an example of operation for reception in asynchronous mode. In serial reception, the SCI operates as described below.

1. The SCI3 monitors the communication line. If a start bit is detected, the SCI3 performs internal synchronization, receives data in RSR, and checks the parity bit and stop bit.
2. If an overrun error occurs (when reception of the next data is completed while the RDRF flag is still set to 1), the OER bit in SSR is set to 1. If the RIE bit in SCR3 is set to 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to RDR.
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 SCR3 is set to 1 at this time, an ERI interrupt request is generated.
4. If a framing error is detected (when the stop bit is 0), the FER bit in SSR is set to 1 and receive data is transferred to RDR. If the RIE bit in SCR3 is set to 1 at this time, an ERI interrupt request is generated.
5. If reception is completed successfully, the RDRF bit in SSR is set to 1, and receive data is transferred to RDR. If the RIE bit in SCR3 is set to 1 at this time, an RXI interrupt request is generated. Continuous reception is possible because the RXI interrupt routine reads the receive data transferred to RDR before reception of the next receive data has been completed.

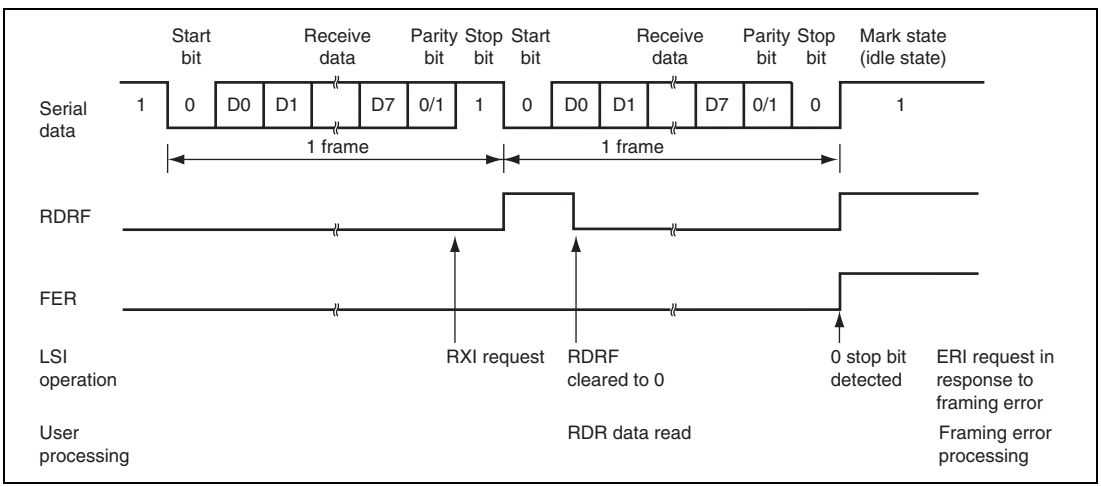


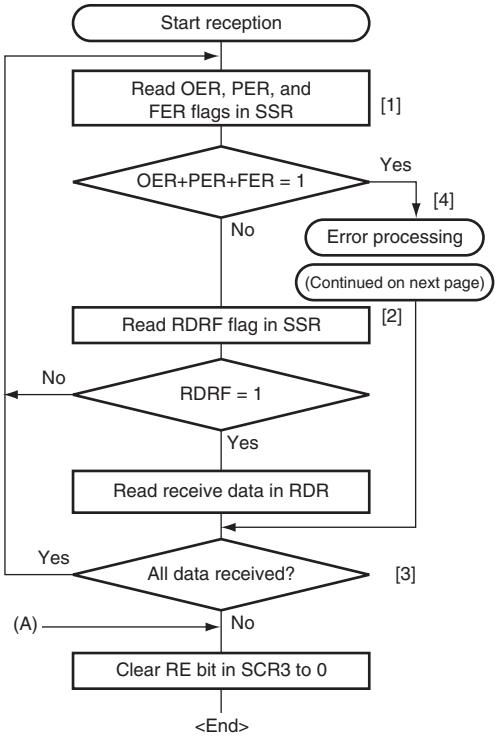
Figure 14.7 Example SCI3 Operation in Reception in Asynchronous Mode (8-Bit Data, Parity, One Stop Bit)

Table 14.5 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 OER, FER, PER, and RDRF bits to 0 before resuming reception. Figure 14.8 shows a sample flowchart for serial data reception.

Table 14.5 SSR Status Flags and Receive Data Handling

	SSR Status Flag				
RDRF*	**OER**	**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] Read the OER, PER, and FER flags in SSR to identify the error. If a receive error occurs, performs the appropriate error processing.

[2] Read SSR and check that RDRF = 1, then read the receive data in RDR. The RDRF flag is cleared automatically.

[3] To continue serial reception, before the stop bit for the current frame is received, read the RDRF flag and read RDR. The RDRF flag is cleared automatically.

[4] If a receive error occurs, read the OER, PER, and FER flags in SSR to identify the error. After performing the appropriate error processing, ensure that the OER, 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.

Figure 14.8 Sample Serial Data Reception Flowchart (Asynchronous mode) (1)

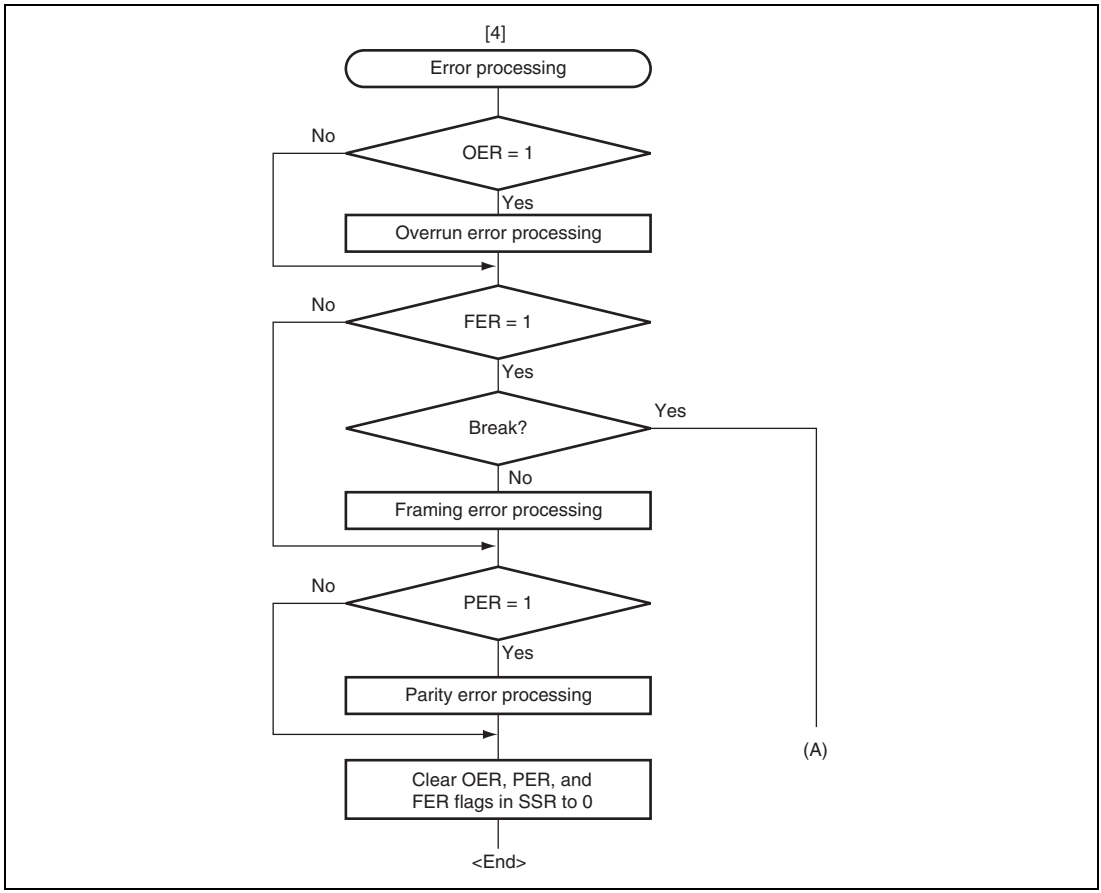


Figure 14.8 Sample Serial Reception Data Flowchart (2)

14.5 Operation in Clocked Synchronous Mode

Figure 14.9 shows the general format for clocked synchronous communication. In clocked synchronous mode, data is transmitted or received synchronous with clock pulses. A single character in the transmit data consists of the 8-bit data starting from the LSB. In clocked synchronous serial communication, data on the transmission line is output from one falling edge of the serial clock to the next. In clocked synchronous mode, the SCI3 receives data in synchronous with the rising edge of the serial 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 SCI3, the transmitter and receiver are independent units, enabling full-duplex communication through the use of a common clock. Both the transmitter and the receiver also have a double-buffered structure, so data can be read or written during transmission or reception, enabling continuous data transfer.

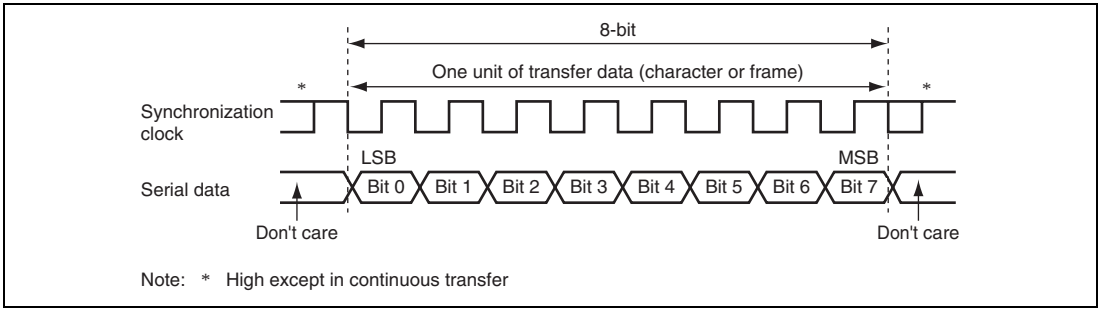


Figure 14.9 Data Format in Clocked Synchronous Communication

14.5.1 Clock

Either an internal clock generated by the on-chip baud rate generator or an external synchronization clock input at the SCK3 pin can be selected, according to the setting of the COM bit in SMR and CKE0 and CKE1 bits in SCR3. When the SCI3 is operated on an internal clock, the serial clock is output from the SCK3 pin. Eight serial clock pulses are output in the transfer of one character, and when no transfer is performed the clock is fixed high.

14.5.2 SCI3 Initialization

Before transmitting and receiving data, the SCI3 should be initialized as described in a sample flowchart in figure 14.4.

14.5.3 Serial Data Transmission

Figure 14.10 shows an example of SCI3 operation for transmission in clocked synchronous mode. In serial transmission, the SCI3 operates as described below.

1. The SCI3 monitors the TDRE flag in SSR, and if the flag is 0, the SCI recognizes that data has been written to TDR, and transfers the data from TDR to TSR.
2. The SCI3 sets the TDRE flag to 1 and starts transmission. If the TIE bit in SCR3 is set to 1 at this time, a transmit data empty interrupt (TXI) is generated.
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. Serial data is transmitted sequentially from the LSB (bit 0), from the TXD pin.
4. The SCI checks the TDRE flag at the timing for sending the MSB (bit 7).
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 TDRE flag maintains the output state of the last bit. If the TEIE bit in SCR3 is set to 1 at this time, a TEI interrupt request is generated.
7. The SCK3 pin is fixed high.

Figure 14.11 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 (OER, FER, or PER) is set to 1. Make sure that the receive error flags are cleared to 0 before starting transmission.

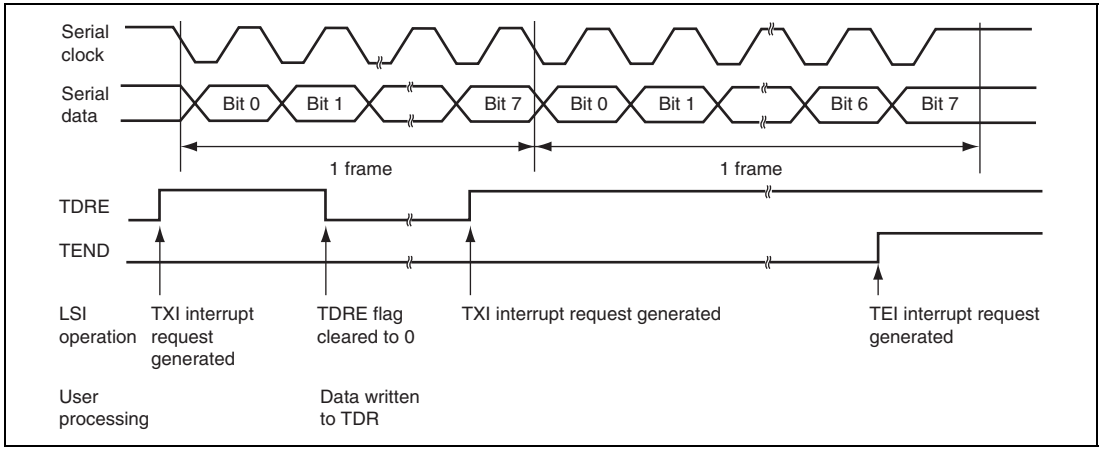
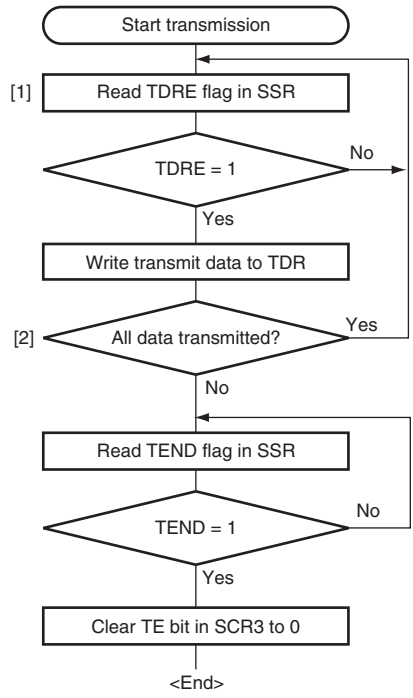


Figure 14.10 Example of SCI3 Operation in Transmission in Clocked Synchronous Mode



[1] Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR. When data is written to TDR, the TDRE flag is automatically cleared to 0 and clocks are output to start the data transmission.

[2] To continue serial transmission, be sure to read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR.
When data is written to TDR, the TDRE

Figure 14.11 Sample Serial Transmission Flowchart (Clocked Synchronous Mode)

14.5.4 Serial Data Reception (Clocked Synchronous Mode)

Figure 14.12 shows an example of SCI3 operation for reception in clocked synchronous mode. In serial reception, the SCI3 operates as described below.

1. The SCI3 performs internal initialization synchronous with a synchronous clock input or output, starts receiving data.
2. The SCI3 stores the received data in RSR.
3. If an overrun error occurs (when reception of the next data is completed while the RDRF flag in SSR is still set to 1), the OER bit in SSR is set to 1. If the RIE bit in SCR3 is set to 1 at this time, an ERI interrupt request is generated, receive data is not transferred to RDR, and the RDRF flag remains to be set to 1.
4. If reception is completed successfully, the RDRF bit in SSR is set to 1, and receive data is transferred to RDR. If the RIE bit in SCR3 is set to 1 at this time, an RXI interrupt request is generated.

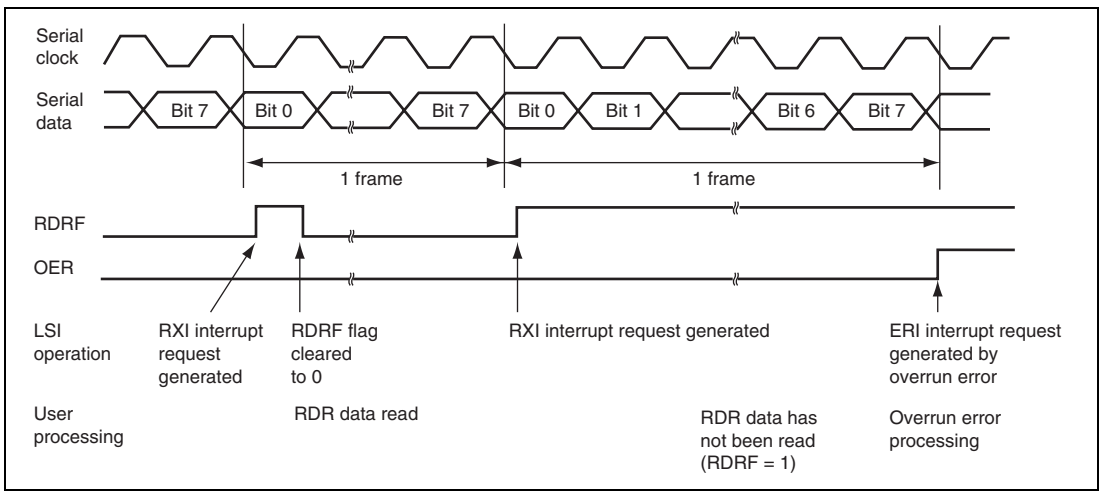
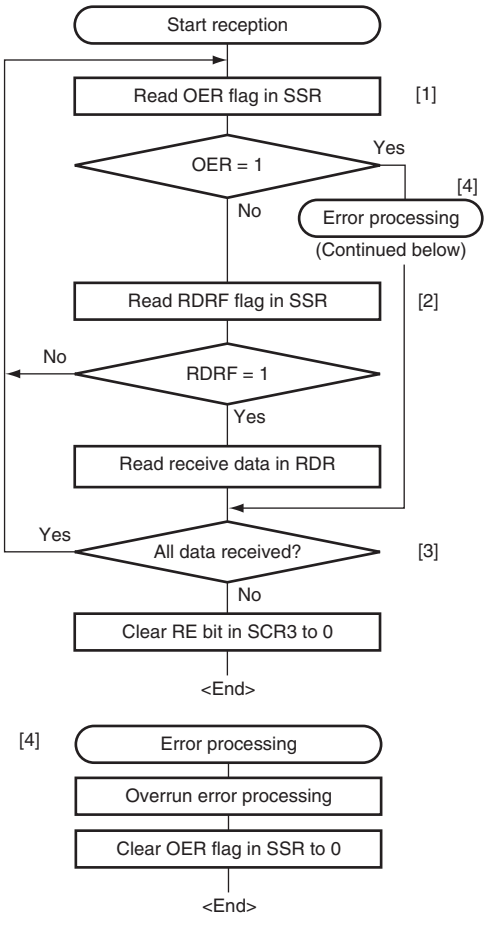


Figure 14.12 Example of SCI3 Reception Operation in Clocked Synchronous Mode

Reception cannot be resumed while a receive error flag is set to 1. Accordingly, clear the OER, FER, PER, and RDRF bits to 0 before resuming reception. Figure 14.13 shows a sample flowchart for serial data reception.



[1] Read the OER flag in SSR to determine if there is an error. If an overrun error has occurred, execute overrun error processing.

[2] Read SSR and check that the RDRF flag is set to 1, then read the receive data in RDR. When data is read from RDR, the RDRF flag is automatically cleared to 0.

[3] To continue serial reception, before the MSB (bit 7) of the current frame is received, reading the RDRF flag and reading RDR should be finished. When data is read from RDR, the RDRF flag is automatically cleared to 0.

[4] If an overrun error occurs, read the OER flag in SSR, and after performing the appropriate error processing, clear the OER flag to 0. Reception cannot be

Figure 14.13 Sample Serial Reception Flowchart (Clocked Synchronous Mode)

14.5.5 Simultaneous Serial Data Transmission and Reception

Figure 14.14 shows a sample flowchart for simultaneous serial transmit and receive operations. 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 SCI3 has finished transmission and the TDRE and TEND flags are set to 1, clear TE to 0. Then simultaneously set TE and RE to 1 with a single instruction. To switch from receive mode to simultaneous transmit and receive mode, after checking that the SCI3 has finished reception, clear RE to 0. Then after checking that the RDRF and receive error flags (OER, FER, and PER) are cleared to 0, simultaneously set TE and RE to 1 with a single instruction.

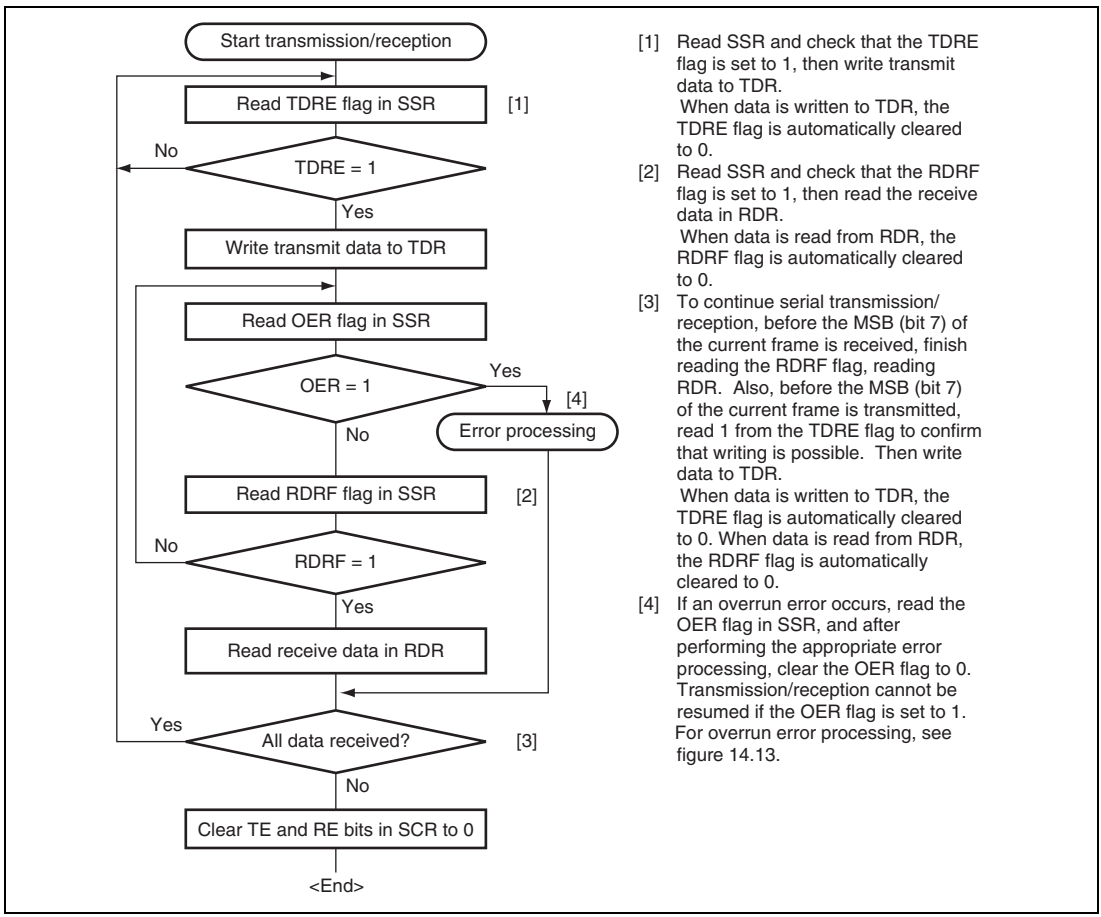


Figure 14.14 Sample Flowchart of Simultaneous Serial Transmit and Receive Operations (Clocked Synchronous Mode)

14.6 Multiprocessor Communication Function

Use of the multiprocessor communication function enables data transfer between a number of processors sharing communication lines by asynchronous serial communication using the multiprocessor format, in which a multiprocessor bit is added to the transfer data. When multiprocessor communication is performed, each receiving station is addressed by a unique ID code. The serial communication cycle consists of two component cycles; an ID transmission cycle that specifies the receiving station, and a data transmission cycle. 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; if the multiprocessor bit is 0, the cycle is a data transmission cycle. Figure 14.15 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 IDs do not match continue to skip data until data with a 1 multiprocessor bit is again received.

The SCI3 uses the MPIE bit in SCR3 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 SSR status flags, RDRF, FER, and OER to 1, are inhibited until data with a 1 multiprocessor bit is received. On reception of a receive character with a 1 multiprocessor bit, the MPBR bit in SSR is set to 1 and the MPIE bit is automatically cleared, thus normal reception is resumed. If the RIE bit in SCR3 is set to 1 at this time, an RXI interrupt is generated.

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

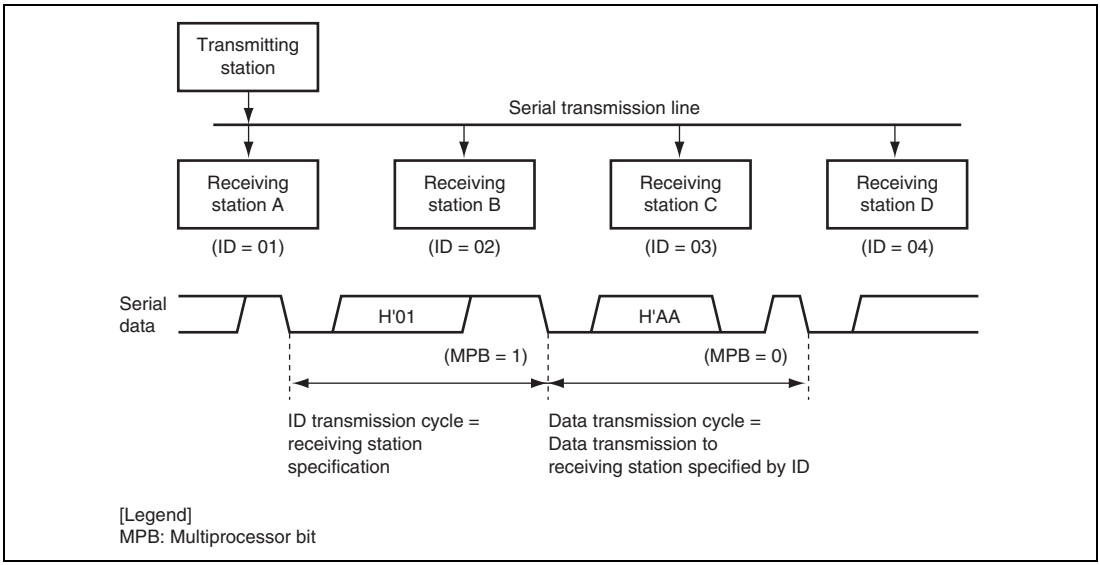


Figure 14.15 Example of Communication Using Multiprocessor Format (Transmission of Data H'AA to Receiving Station A)

14.6.1 Multiprocessor Serial Data Transmission

Figure 14.16 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 SCI3 operations are the same as those in asynchronous mode.

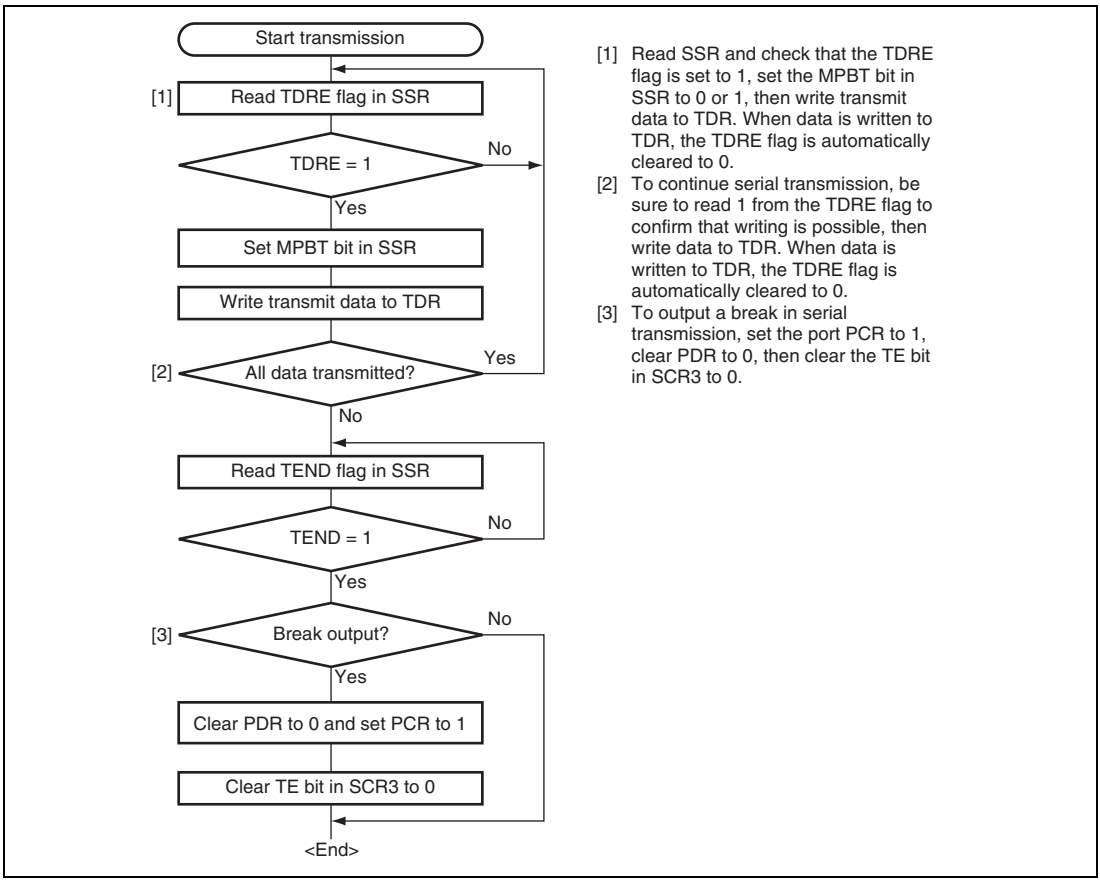


Figure 14.16 Sample Multiprocessor Serial Transmission Flowchart

14.6.2 Multiprocessor Serial Data Reception

Figure 14.17 shows a sample flowchart for multiprocessor serial data reception. If the MPIE bit in SCR3 is set to 1, data is skipped until data with a 1 multiprocessor bit is received. 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 SCI3 operations are the same as in asynchronous mode. Figure 14.18 shows an example of SCI3 operation for multiprocessor format reception.

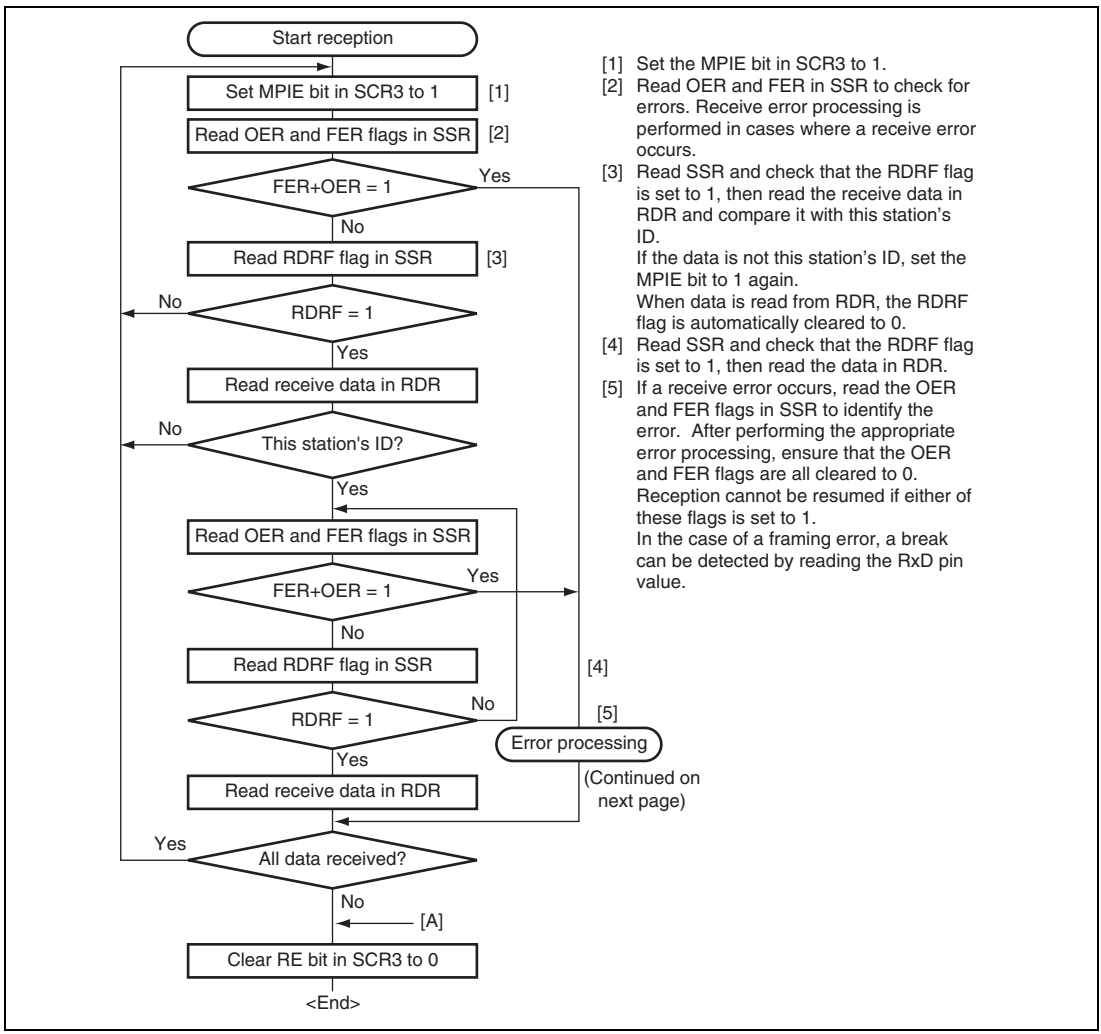


Figure 14.17 Sample Multiprocessor Serial Reception Flowchart (1)

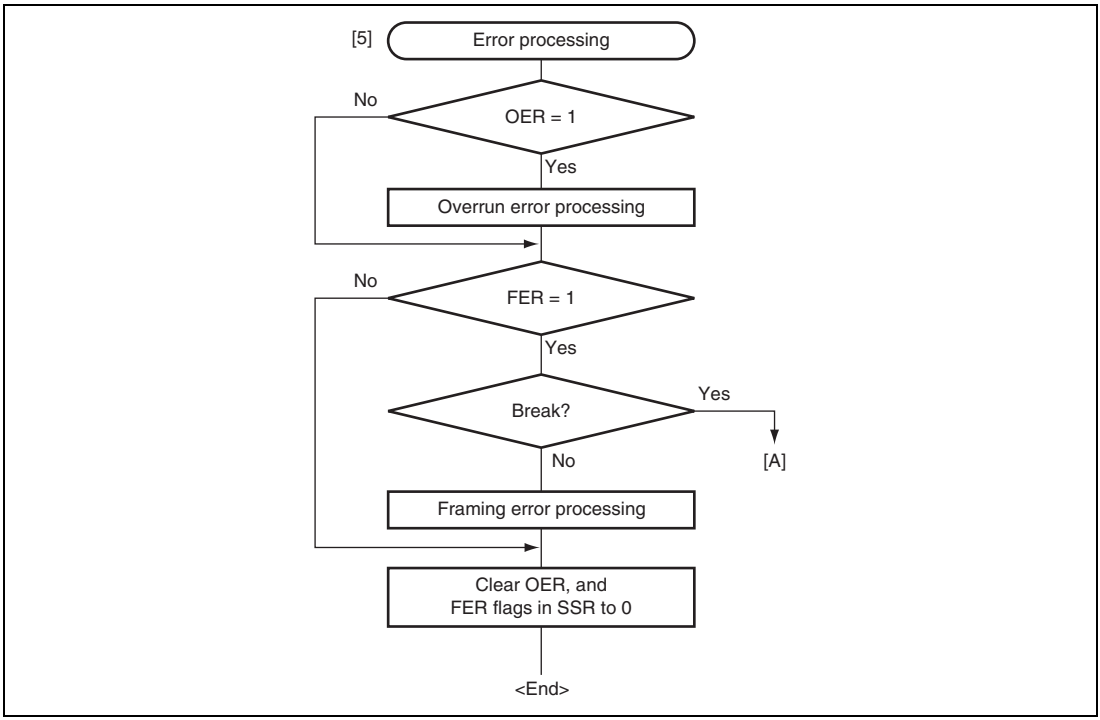


Figure 14.17 Sample Multiprocessor Serial Reception Flowchart (2)

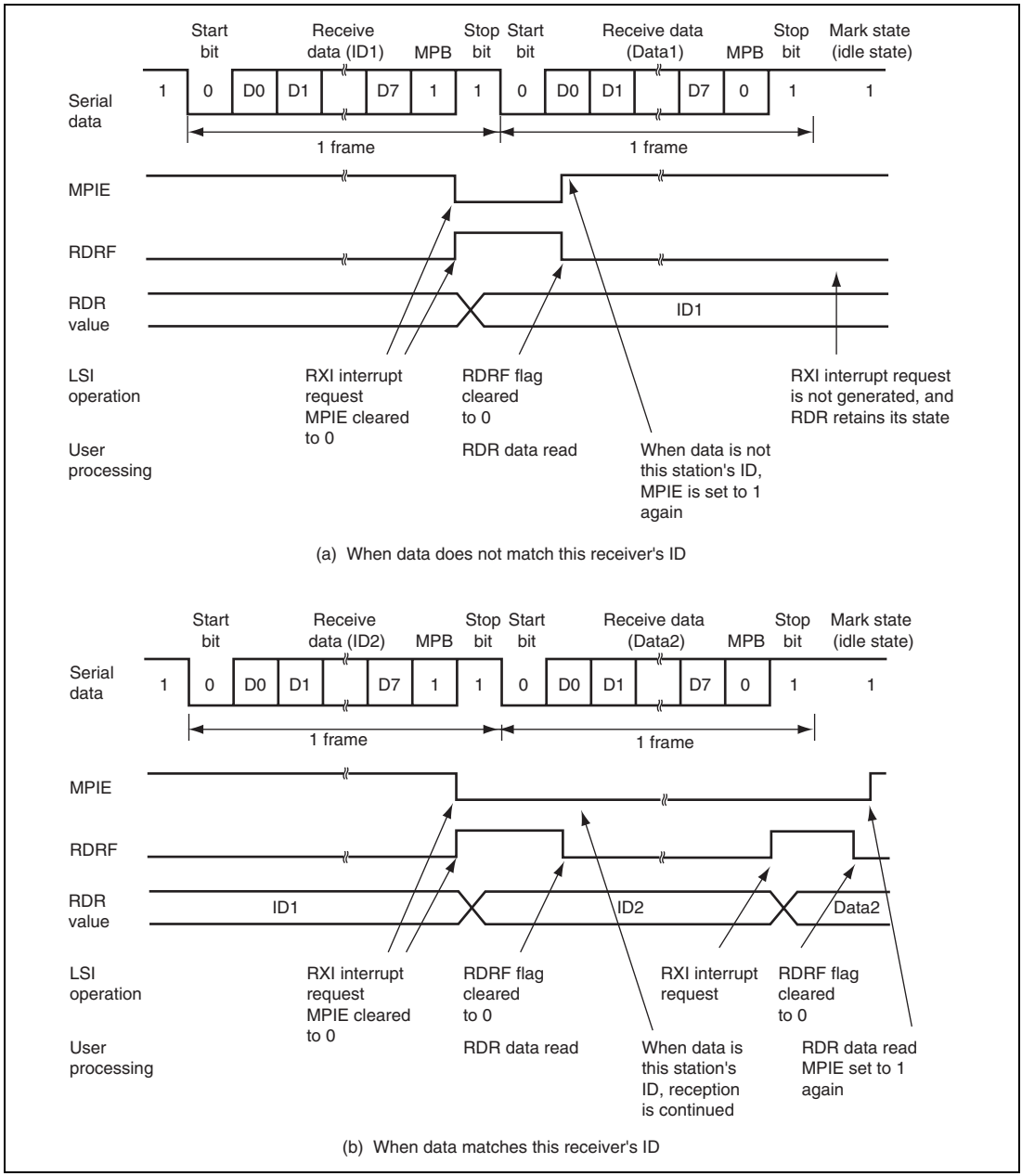


Figure 14.18 Example of SCI3 Operation in Reception Using Multiprocessor Format (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)

14.7 Interrupts

The SCI3 creates the following six interrupt requests: transmission end, transmit data empty, receive data full, and receive errors (overrun error, framing error, and parity error). Table 14.6 shows the interrupt sources.

Table 14.6 SCI3 Interrupt Requests

Interrupt Requests	Abbreviation	Interrupt Sources
Receive Data Full	RXI	Setting RDRF in SSR
Transmit Data Empty	TXI	Setting TDRE in SSR
Transmission End	TEI	Setting TEND in SSR
Receive Error	ERI	Setting OER, FER, and PER in SSR

The initial value of the TDRE flag in SSR is 1. Thus, when the TIE bit in SCR3 is set to 1 before transferring the transmit data to TDR, a TXI interrupt request is generated even if the transmit data is not ready. The initial value of the TEND flag in SSR is 1. Thus, when the TEIE bit in SCR3 is set to 1 before transferring the transmit data to TDR, a TEI interrupt request is generated even if the transmit data has not been sent. It is possible to make use of the most of these interrupt requests efficiently by transferring the transmit data to TDR in the interrupt routine. To prevent the generation of these interrupt requests (TXI and TEI), set the enable bits (TIE and TEIE) that correspond to these interrupt requests to 1, after transferring the transmit data to TDR.

14.8 Usage Notes

14.8.1 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 0, setting the FER flag, and possibly the PER flag. Note that as the SCI3 continues the receive operation after receiving a break, even if the FER flag is cleared to 0, it will be set to 1 again.

14.8.2 Mark State and Break Sending

When TE is 0, the TxD pin is used as an I/O port whose direction (input or output) and level are determined by PCR and PDR. 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 PCR and PDR to 1. As TE 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 PCR to 1 and PDR to 0, and then clear TE to 0. When TE 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.

14.8.3 Receive Error Flags and Transmit Operations (Clocked Synchronous Mode Only)

Transmission cannot be started when a receive error flag (OER, PER, or FER) is set to 1, even if the TDRE flag is cleared to 0. Be sure to clear the receive error flags to 0 before starting transmission. Note also that receive error flags cannot be cleared to 0 even if the RE bit is cleared to 0.

14.8.4 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode

In asynchronous mode, the SCI3 operates on a basic clock with a frequency of 16 times the transfer rate. In reception, the SCI3 samples the falling edge of the start bit using the basic clock, and performs internal synchronization. Receive data is latched internally at the rising edge of the 8th pulse of the basic clock as shown in figure 14.19.

Thus, the reception margin in asynchronous mode is given by formula (1) below.

$$M = \left\{ (0.5 - \frac{1}{2N}) - \frac{D - 0.5}{N} - (L - 0.5)\,F \right\} \times 100(\%)$$

... Formula (1)

Where 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 (absolute value of clock rate deviation) = 0 and D (clock duty) = 0.5 in formula (1), the reception margin can be given by the formula.

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

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

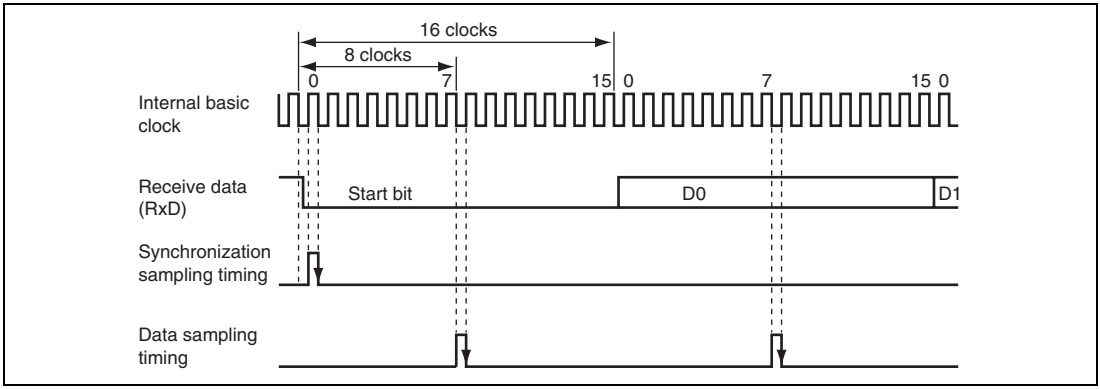


Figure 14.19 Receive Data Sampling Timing in Asynchronous Mode

Section 15 I²C Bus Interface (IIC)

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.

15.1 Features

- Selection of I²C format or clocked synchronous serial format
  - I²C bus format: addressing format with acknowledge bit, for master/slave operation
  - Clocked synchronous serial format: non-addressing format without acknowledge bit, for master operation only
- I²C bus format
- Two ways of setting slave address
- Start and stop conditions generated automatically in master mode
- Selection of acknowledge output levels when receiving
- Automatic loading of acknowledge bit when transmitting
- Wait function in master mode

  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 in slave mode

  A wait request can be generated by driving the SCL pin low after data transfer, excluding acknowledgement. The wait request is cleared when the next transfer becomes possible.
- Three interrupt sources
  - Data transfer end (including transmission mode transition with I²C bus format and address reception after loss of master arbitration)
  - Address match: when any slave address matches or the general call address is received in slave receive mode
  - Stop condition detection
- Selection of 16 internal clocks (in master mode)
- Direct bus drive
  - Two pins, SCL and SDA pins function as NMOS open-drain outputs when the bus drive function is selected.

Figure 15.1 shows a block diagram of the I²C bus interface.

Figure 15.2 shows an example of I/O pin connections to external circuits. The I/O pins are NMOS open drains. Set the upper limit of voltage applied to the power supply (V_{CC}) voltage range + 0.3 V, i.e. 5.8 V.

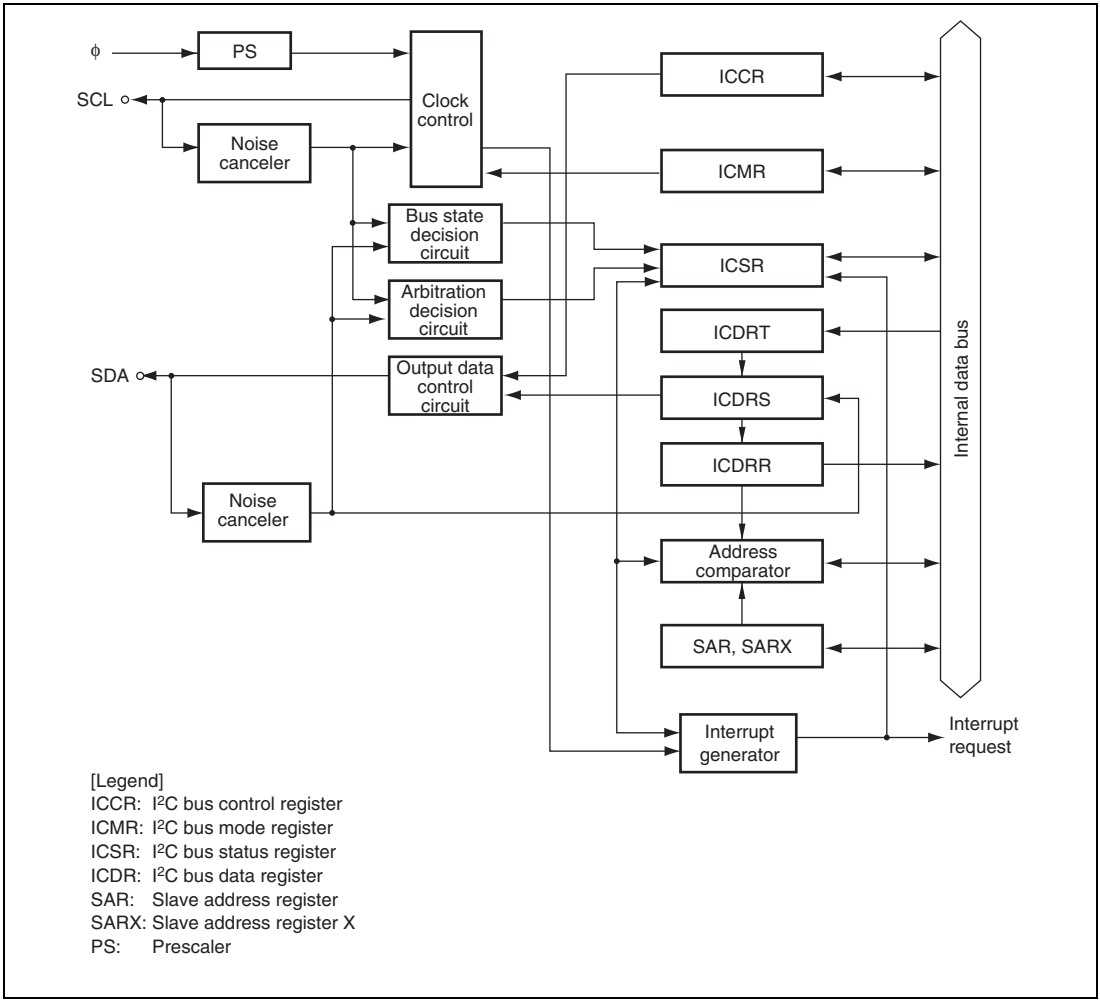


Figure 15.1 Block Diagram of I²C Bus Interface

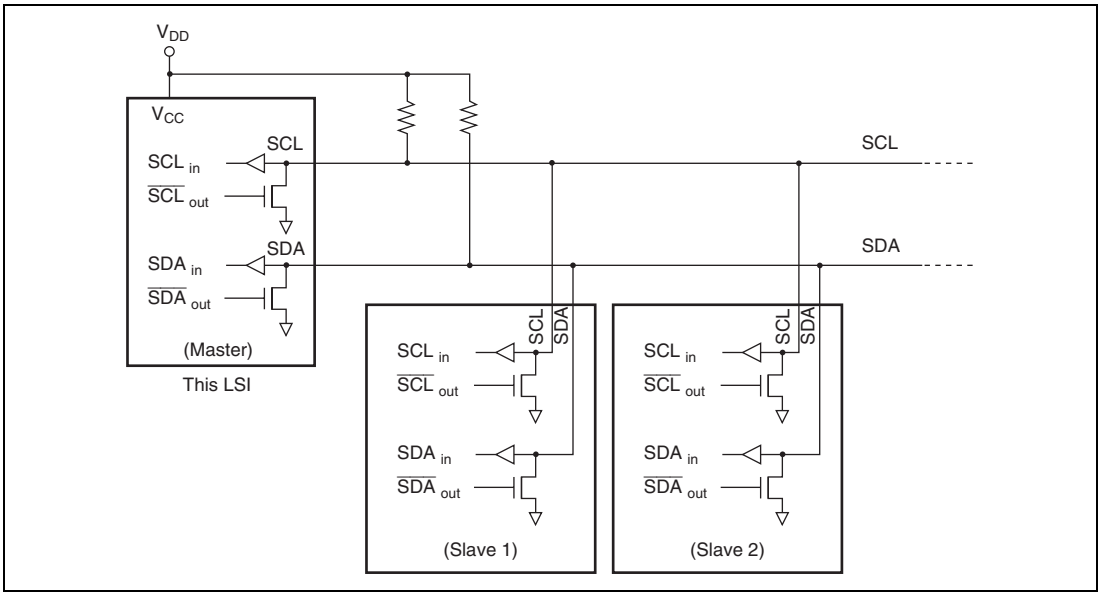


Figure 15.2 I²C Bus Interface Connections (Example: This LSI as Master)

15.2 Input/Output Pins

Table 15.1 summarizes the input/output pins used by the I²C bus interface.

Table 15.1 I²C Bus Interface Pins

Name	Abbreviation	I/O	Function
Serial clock	SCL	I/O	IIC serial clock input/output
Serial data	SDA	I/O	IIC serial data input/output

15.3 Register Descriptions

The I²C bus interface has the following registers. ICDR, SARX, ICMR, and SAR are allocated to one address, and registers that can be accessed depend on the ICE bit in ICCR. When ICE = 0. SAR and SARX can be accessed. When ICE = 1, ICMR and ICDR can be accessed.

- 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)
- Timer serial control register (TSCR)

15.3.1 I²C Bus Data Register (ICDR)

ICDR is an 8-bit readable/writable register that is used as a transmit data register when transmitting and a receive data register when receiving. ICDR is divided internally into a shift register (ICDRS), receive buffer (ICDRR), and transmit buffer (ICDRT). Data transfers among the three registers are performed automatically in coordination with changes in the bus state, and affect the status of internal flags such as TDRE and RDRF. When TDRE is 1 and the transmit buffer is empty, TDRE shows that the next transmit data can be written from the CPU. When RDRF is 1, it shows that the valid receive data is stored in the receive buffer.

If I²C is in transmit mode and the next data is in ICDRT (the TDRE flag is 0) following transmission/reception of one frame of data using ICDRS, data is transferred automatically from ICDRT to ICDRS. If I²C is in receive mode and no previous data remains in ICDRR (the RDRF flag is 0) following transmission/reception of one frame of data using ICDRS, data is transferred automatically from ICDRS to ICDRR.

If the number of bits in a frame, excluding the acknowledge bit, is less than 8, transmit data and receive data are stored differently. Transmit data should be written justified toward the MSB side when MLS = 0, and toward the LSB side when MLS = 1. Receive data bits read from the LSB side should be treated as valid when MLS = 0, and bits read from the MSB side when MLS = 1.

ICDR can be written and read only when the ICE bit is set to 1 in ICCR.

The value of ICDR is undefined after a reset.

The TDRE and RDRF flags are set and cleared under the conditions shown below. Setting the TDRE and RDRF flags affects the status of the interrupt flags.

Bit	Bit Name	Initial Value	R/W	Description
—	TDRE	—	—	Transmit Data Register Empty [Setting conditions] • In transmit mode, when a start condition is detected in the bus line state after a start condition is issued in master mode with the I²C bus format or serial format selected • When transmit mode (TRS = 1) is set without a format • When data is transferred from ICDRT to ICDRS • When a switch is made from receive mode to transmit mode after detection of a start condition [Clearing conditions] • When transmit data is written in ICDR in transmit mode • When a stop condition is detected in the bus line state after a stop condition is issued with the I2C bus format or serial format selected • When a stop condition is detected with the I2C bus format selected • In receive mode
—	RDRF	—	—	Receive Data Register Full [Setting condition] When data is transferred from ICDRS to ICDRR [Clearing condition] When ICDR (ICDRR) receive data is read in receive mode

15.3.2 Slave Address Register (SAR)

SAR selects the slave address and selects the communication format. SAR can be written and read only when the ICE bit is cleared to 0 in ICCR.

Bit	Bit Name	Initial Value	R/W	Description
7	SVA6	0	R/W	Slave Address 6 to 0
6	SVA5	0	R/W	Sets 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	Selects the communication format together with the FSX bit in SARX. Refer to table 15.2.

15.3.3 Second Slave Address Register (SARX)

SARX stores the second slave address and selects the communication format. SARX can be written and read only when the ICE bit is cleared to 0 in ICCR.

Bit	Bit Name	Initial Value	R/W	Description
7	SVAX6	0	R/W	Slave Address 6 to 0
6	SVAX5	0	R/W	Sets 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	0	R/W	Selects the communication format together with the FS bit in SAR. Refer to table 15.2.

Table 15.2 Communication Format

SAR	SARX	
FS	**FSX**	**I²C Transfer Format**
0	0	SAR and SARX are used as the slave addresses with the I²C bus format.
0	1	Only SAR is used as the slave address with the I²C bus format.
1	0	Only SARX is used as the slave address with the I²C bus format.
1	1	Clock synchronous serial format (SAR and SARX are invalid)

15.3.4 I²C Bus Mode Register (ICMR)

The I²C bus mode register (ICMR) sets the transfer format and transfer rate. It can only be accessed when the ICE bit in ICCR is 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. When WAIT is set to 1, after the fall of the clock for the final data bit, 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. If WAIT is cleared to 0, data and acknowledge bits are transferred consecutively with no wait inserted. The IRIC flag in ICCR is set to 1 on completion of the acknowledge bit transfer, regardless of the WAIT setting.
5	CKS2	0	R/W	Serial Clock Select 2 to 0
4	CKS1	0	R/W	This bit is valid only in master mode.
3	CKS0	0	R/W	These bits select the required transfer rate, together with the IICX bit in TSCR. Refer table 15.3.

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. With the I²C bus format, the data is transferred with one addition acknowledge bit. 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. The value returns to 000 at the end of a data transfer, including the acknowledge bit.	
0	BC0	0	R/W	I²C Bus Format	Clocked Synchronous Mode
				000: 9	000: 8
				001: 2	001: 1
				010: 3	010: 2
				011: 4	011: 3
				100: 5	100: 4
				101: 6	101: 5
				110: 7	110: 6
				111: 8	111: 7

Table 15.3 I²C Transfer Rate

TSCR	ICMR							
Bit 0	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
0	0	0	0	φ/28	179MHz	286kHz	357kHz	571kHz
0	0	0	1	φ/40	125kHz	200kHz	250kHz	400kHz
0	0	1	0	φ/48	104kHz	167kHz	208kHz	333kHz
0	0	1	1	φ/64	78.1kHz	125kHz	156kHz	250kHz
0	1	0	0	φ/80	62.5kHz	100kHz	125kHz	200kHz
0	1	0	1	φ/100	50.0kHz	80.0kHz	100kHz	160kHz
0	1	1	0	φ/112	44.6kHz	71.4kHz	89.3kHz	143kHz
0	1	1	1	φ/128	39.1kHz	62.5kHz	78.1kHz	125kHz
1	0	0	0	φ/56	89.3kHz	143kHz	179kHz	286kHz
1	0	0	1	φ/80	62.5kHz	100kHz	125kHz	200kHz
1	0	1	0	φ/96	52.1kHz	83.3kHz	104kHz	167kHz
1	0	1	1	φ/128	39.1kHz	62.5kHz	78.1kHz	125kHz
1	1	0	0	φ/160	31.3kHz	50.0kHz	62.5kHz	100kHz
1	1	0	1	φ/200	25.0kHz	40.0kHz	50.0kHz	80.0kHz
1	1	1	0	φ/224	22.3kHz	35.7kHz	44.6kHz	71.4kHz
1	1	1	1	φ/256	19.5kHz	31.3kHz	39.1kHz	62.5kHz

15.3.5 I²C Bus Control Register (ICCR)

I²C bus control register (ICCR) consists of the control bits and interrupt request flags of I²C bus interface.

Bit	Bit Name	Initial Value	R/W	Description
7	ICE	0	R/W	I²C Bus Interface Enable When this bit is set to 1, the I²C bus interface module is enabled to send/receive data and drive the bus since it is connected to the SCL and SDA pins. ICMR and ICDR can be accessed. When this bit is cleared, the module is halted and separated from the SCL and SDA pins. SAR and SARX can be accessed.
6	IEIC	0	R/W	I²C Bus Interface Interrupt Enable When this bit is 1, Interrupts are enabled by IRIC.
5	MST	0	R/W	Master/Slave Select
4	TRS	0	R/W	Transmit/Receive Select 00: Slave receive mode 01: Slave transmit mode 10: Master receive mode 11: Master transmit mode Both these bits will be cleared by hardware when they lose in a bus contention in master mode of the I²C bus format. In slave receive mode, the R/W bit in the first frame immediately after the start automatically sets these bits in receive mode or transmit mode by using hardware. The settings can be made again for the bits that were set/cleared by hardware, by reading these bits. When the TRS bit is intended to change during a transfer, the bit will not be switched until the frame transfer is completed, including acknowledgement.

Bit	Bit Name	Initial Value	R/W	Description
3	ACKE	0	R/W	Acknowledge Bit Judgement 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, which is always 0. 1: If the acknowledge bit is 1, continuous transfer is interrupted.
2	BBSY	0	R/W	Bus Busy In slave mode, reading the BBSY flag enables to confirm whether the I²C bus is occupied or released. The BBSY flag is set to 0 when the SDA level changes from high to low under the condition of SCl = high, assuming that the start condition has been issued. The BBSY flag is cleared to 0 when the SDA level changes from low to high under the condition of SCl = high, assuming that the start condition has been issued. Writing to the BBSY flag in slave mode is disabled. In master mode, the BBSY flag is used to issue start and stop conditions. Write 1 to BBSY and 0 to SCP to issue a start condition. Follow this procedure when also re-transmitting a start condition. 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.

Bit	Bit Name	Initial Value	R/W	Description
1	IRIC	0	R/W	I²C Bus Interface Interrupt Request Flag Also see table 15.4. [Setting conditions] In master mode with I²C bus format • When a start condition is detected in the bus line state after a start condition is issued • When a wait is inserted between the data and acknowledge bit when WAIT = 1 • At the rising edge of the ninth transfer/receive clock, and at the falling edge of the eighth transfer/receive clock when a wait is inseted • When a slave address is received after bus arbitration is lost (when the AL flag is set to1) • When 1 is received as the acknowledge bit when the ACKE bit is 1 (when the ACKB bit is set to 1) I²C bus format slave mode • When the slave address (SVA, SVAX) matches (when the AAS and AASX flags are set to 1) and at the end of data transfer up to the subsequent retransmission start condition or stop condition detection (FS = 0 and when the TDRE or RDRF flag is set to 1) • When the general call address is detected (when the ADZ flag is set to 1) and at the end of data transfer up to the subsequent retransmission start condition or stop condition detection (when the TDRE or RDRF flag is set to 1) • When 1 is received as the acknowledge bit when the ACKE bit is 1 (when the ACKB bit is set to 1) • When a stop condition is detected (when the STOP or ESTP flag is set to 1) Clocked synchronous serial format • At the end of data transfer (when the TDRE or RDRF flag is set to 1) • When a start condition is detected with serial format selected [Clearing condition] When 0 is written in IRIC after reading IRIC = 1

Bit	Bit Name	Initial Value	R/W	Description
0	SCP	1	W	Start Condition/Stop Condition Prohibit The SCP bit controls the issue of start/stop conditions in master mode. To issue a start condition, write 1 in BBSY and 0 in SCP. A retransmit start condition is issued in the same way. To issue a stop condition, write 0 in BBSY and 0 in SCP. This bit is always read as 1. If 1 is written, the data is not stored.

15.3.6 I²C Bus Status Register (ICSR)

The I²C bus status register (ICSR) consists of status flags. Also see table 15.4.

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 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 during frame transfer. [Clearing conditions] • When 0 is written in STOP after reading STOP = 1 • When the IRIC flag is cleared to 0

Bit	Bit Name	Initial Value	R/W	Description
5	IRTR	0	R/W	I²C Bus Interface Continuous Transmission/Reception Interrupt Request Flag [Setting conditions] In I²C bus interface slave mode • When the TDRE or RDRF flag is set to 1 when AASX = 1 In I²C bus interface other modes • When the TDRE or RDRF flag is set to 1 [Clearing conditions] • When 0 is written in IRTR after reading IRTR = 1 • When the IRIC flag is cleared to 0
4	AASX	0	R/W	Second Slave Address Recognition Flag [Setting condition] When the second slave address is detected in slave receive mode and FSX = 0 [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 [Setting condition] When bus arbitration was lost in master mode. [Clearing conditions] • When 0 is written in AL after reading AL = 1 • When ICDR data is written (transmit mode) or read (receive mode)

Bit	Bit Name	Initial Value	R/W	Description
2	AAS	0	R/W	Slave Address Recognition Flag [Setting condition] When the slave address or general call address is detected in slave receive mode and FS = 0. [Clearing conditions] • When ICDR data is written (transmit mode) or read (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 This bit is valid in I²C bus format slave receive mode. [Setting condition] When the general call address is detected in slave receive mode and FSX = 0 or FS = 0. [Clearing conditions] • When ICDR data is written (transmit mode) or read (receive mode) • When 0 is written in ADZ after reading ADZ = 1 • In master mode
0	ACKB	0	R/W	Acknowledge Bit In transmit mode, the acknowledge data that are returned by the receive device is loaded. In receive mode, the acknowledge data originally specified to this bit is sent to the transmit device, after receiving data. When this bit is read, the loaded value (return value from the receive device) is read at transmission and the specified value is read at reception.

15.3.7 Timer Serial Control Register (TSCR)

The timer serial control register (TSCR) is an 8-bit readable/writable register that controls the operating modes.

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	–	All 1	–	Reserved This bit is always read as 1 and cannot be modified.
1	IICRST	0	R/W	I²C Control Unit Reset Resets the control unit except for the I²C registers. When a hang up occurs due to illegal communication during I²C operation, setting IICRST to 1 can set a port or reset the I²C control unit without initializing registers.
0	IICX	0	R/W	I²C Transfer Rate Select Selects the transfer rate in master mode, together with bits CKS2 to CKS0 in ICMR. Refer to table 15.3.

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 TDRE or RDRF internal flag is set, the readable IRTR flag may or may not be set. Even when the IRIC flag and IRTR flag are set, the TDRE or RDRF internal flag may not be set. Table 15.4 shows the relationship between the flags and the transfer states.

Table 15.4 Flags and Transfer States

MST	TRS	BBSY	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB	State
1/0	1/0	0	0	0	0	0	0	0	0	0	Idle state (flag clearing required)
1	1	0	0	0	0	0	0	0	0	0	Start condition issuance
1	1	1	0	0	1	0	0	0	0	0	Start condition established
1	1/0	1	0	0	0	0	0	0	0	0/1	Master mode wait
1	1/0	1	0	0	1	0	0	0	0	0/1	Master mode transmit/receive end
0	0	1	0	0	0	1/0	1	1/0	1/0	0	Arbitration lost
0	0	1	0	0	0	0	0	1	0	0	SAR match by first frame in slave mode
0	0	1	0	0	0	0	0	1	1	0	General call address match
0	0	1	0	0	0	1	0	0	0	0	SARX match
0	1/0	1	0	0	0	0	0	0	0	0/1	Slave mode transmit/receive end (except after SARX match)
0	1/0	1	0	0	1	1	0	0	0	0	Slave mode transmit/receive end (after SARX match)
0	1	1	0	0	0	1	0	0	0	1	
0	1/0	0	1/0	1/0	0	0	0	0	0	0/1	Stop condition detected

15.4 Operation

The I²C bus interface has serial and I²C bus formats.

15.4.1 I²C Bus Data Format

The I²C bus formats are addressing formats and an acknowledge bit is inserted. These are shown in figures 15.3. Figure 15.5 shows the I²C bus timing. The first frame following a start condition always consists of 8 bits.

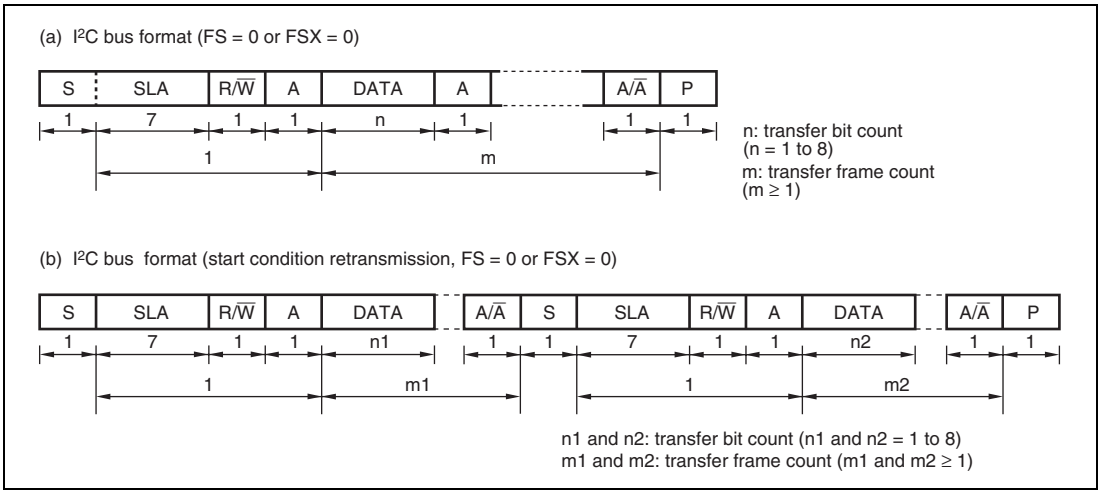


Figure 15.3 I²C Bus Data Formats (I²C Bus Formats)

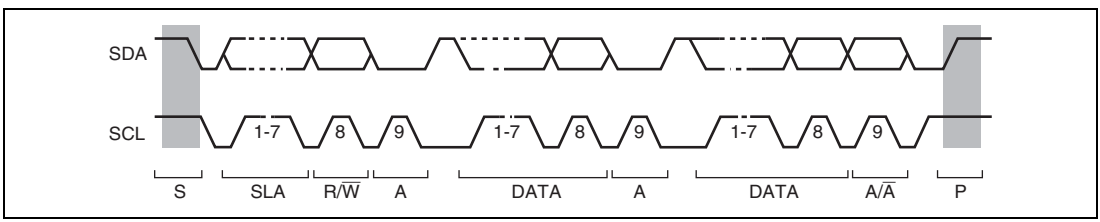


Figure 15.4 I²C Bus Timing

[Legend]

S: Start condition. The master device drives SDA from high to low while SCL is high

SLA: Slave address

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

DATA: Transferred data

P: Stop condition. The master device drives SDA from low to high while SCL is high

15.4.2 Master Transmit Operation

When data is set to ICDR during the period between the execution of an instruction to issue a start condition and the creation of the start condition, the data may not be output normally, because there will be a contention between a generation of a start condition and an output of data. Although data H'FF is to be sent to the ICDR register by a dummy write operation before an issue of a stop condition, the H'FF data may be output by the dummy write operation if the execution of the instruction to issue a stop condition is delayed. To prevent these problems, follow the flowchart shown below during the 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. The transmission procedure and operations synchronize with the ICDR writing are described below.

1. Set the ICE bit in ICCR to 1. Set bits MLS, WAIT, and CKS2 to CKS0 in ICMR, and bit IICX in TSCR, according to the operating mode.
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. This changes SDA from high to low when SCL is high, and generates the start condition.
5. Then 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. As indicating the end of the transfer, and so the IRIC flag is cleared to 0. After writing ICDR, clear IRIC continuously not to execute other interrupt handling routine. If one frame of data has been transmitted before the IRIC clearing, it can not be determine the end of transmission. The master device sequentially sends the transmission clock and the data written to ICDR using the timing shown in figure 15.5. 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 the step [12] to end transmission, and retry the transmit operation.

9. Write the transmit data to ICDR. As indicating the end of the transfer, and so the IRIC flag is cleared to 0. Perform the ICDR write and the IRIC flag clearing sequentially, just as in the 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 data to be transmitted, go to the step [9] to continue next transmission. When the slave device has not acknowledged (ACKB bit is set to 1), operate the step [12] to end transmission.
12. Clear the IRIC flag to 0. 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.

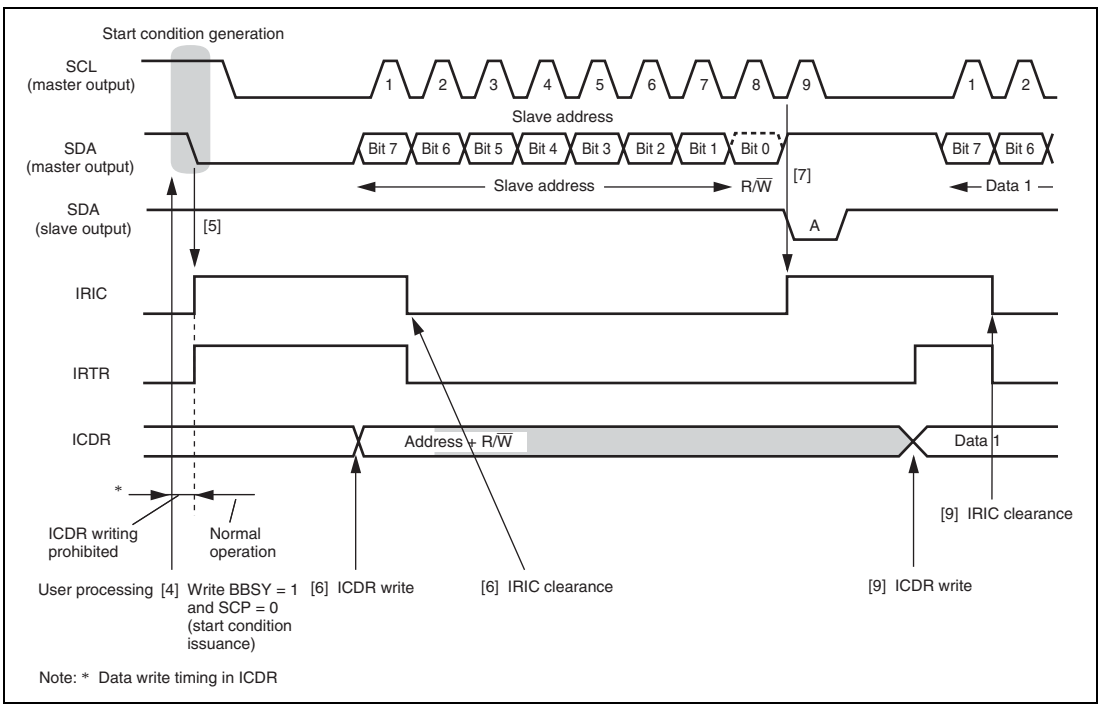


Figure 15.5 Master Transmit Mode Operation Timing Example (MLS = WAIT = 0)

15.4.3 Master Receive Operation

The data buffer of the I²C module can receive data consecutively since it consists of ICDRR and ICDRS. However, if the completion of receiving the last data is delayed, there will be a contention between the instruction to issue a stop condition and the SCl clock output to receive the next data, and may generate unnecessary clocks or fix the output level of the SDA line as low. The switch timing of the ACKB bit in the ICSR register should be controlled because the acknowledge bit does not return acknowledgement after receiving the last data in master mode. These problems can be avoided by using the WAIT function. Follow the flowchart shown below.

In master receive mode, the master device outputs the receive clock, receives data, and returns an acknowledge signal. The slave device transmits data. The reception procedure and operations with the wait function synchronized with the ICDR read operation to receive data in sequence are shown below.

1. Clear the TRS bit in ICCR to 0 to switch from transmit mode to receive mode, and set the WAIT bit in ICMR to 1. Also clear the bit in ICSR to ACKB 0 (acknowledge data setting).
2. When ICDR is read (dummy data read), reception is started, and the receive clock is output, and data received, in synchronization with the internal clock. In order to detect wait operation, set the IRIC flag in ICCR must be cleared to 0. After reading ICDR, clear IRIC continuously not to execute other interrupt handling routine. If one frame of data has been received before the IRIC clearing, it can not be determine the end of reception.
3. The IRIC flag is set to 1 at the fall of the 8th receive clock pulse. If the IEIC bit in ICCR has been set to 1, an interrupt request is sent to the CPU. SCL is automatically fixed low in synchronization with the internal clock until the IRIC flag clearing. If the first frame is the last receive data, execute the step [10] to halt reception.
4. Clear the IRIC flag to release from the Wait State. The master device outputs the 9th clock and drives SDA at the 9th receive clock pulse to return an acknowledge signal.
5. When one frame of data has been received, the IRIC flag in ICCR and the IRTR flag in ICSR are set to 1 at the rise of the 9th receive clock pulse. The master device outputs SCL clock to receive next data.
6. Read ICDR.
7. Clear the IRIC flag to detect next wait operation. Data reception process from the step [5] to [7] should be executed during one byte reception period after IRIC flag clearing in the step [4] or [9] to release wait status.
8. The IRIC flags set to 1 at the fall of 8th receive clock pulse. SCL is automatically fixed low in synchronization with the internal clock until the IRIC flag clearing. If this frame is the last receive data, execute the step [10] to halt reception.

9. Clear the IRIC flag in ICCR to cancel wait operation. The master device outputs the 9th clock and drives SDA at the 9th receive clock pulse to return an ackowledge signal. Data can be received continuously by repeating the step [5] to [9].
10. Set the ACKB bit in ICSR to 1 so as to return “No acknowledge” data. Also set the TRS bit in ICCR to 1 to switch from receive mode to transmit mode.
11. Clear IRIC flag to 0 to release from the Wait State.
12. When one frame of data has been received, the IRIC flag is set to 1 at the rise of the 9th receive clock pulse.
13. Clear the WAIT bit to 0 to switch from wait mode to no wait mode. Read ICDR and the IRIC flag to 0. Clearing of the IRIC flag should be after the 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 cannot be issued because the output level of the SDA line is fixed as low.
14. Clear the BBSY bit and SCP bit to 0. This changes SDA from low to high when SCL is high, and generates the stop condition.

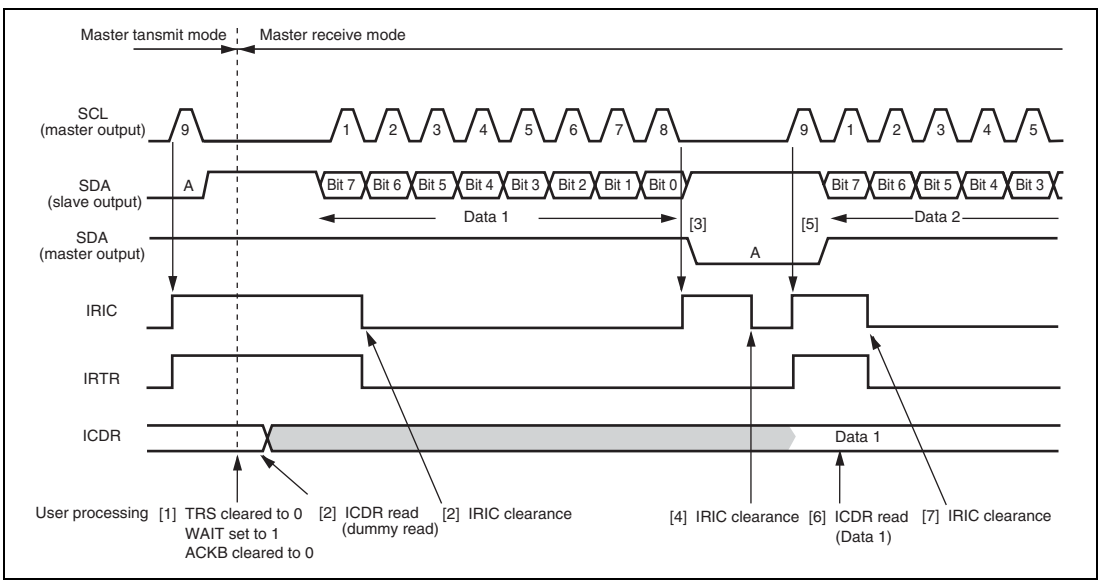


Figure 15.6 Master Receive Mode Operation Timing Example (1)
(MLS = ACKB = 0, WAIT = 1)

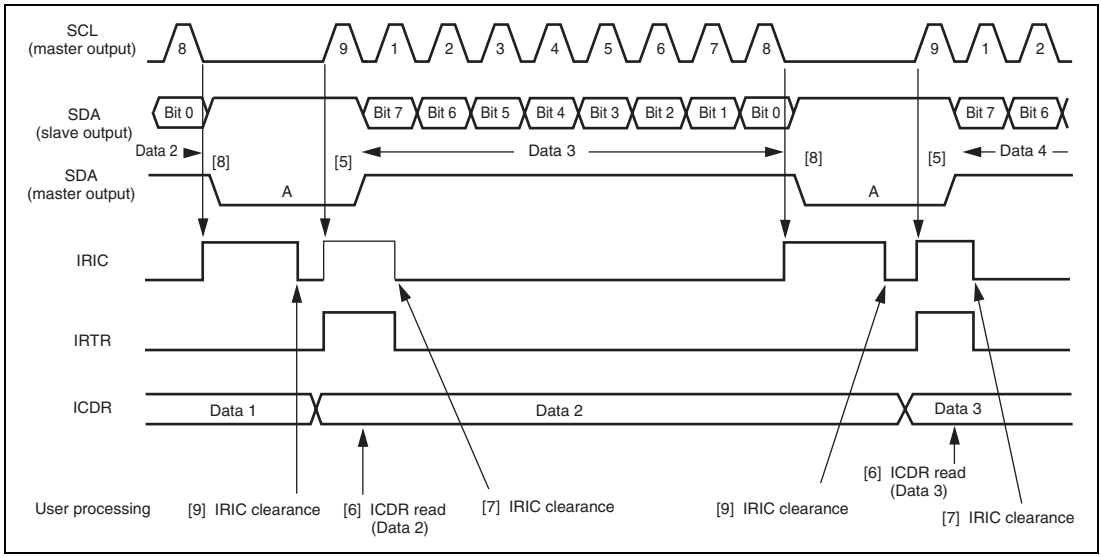


Figure 15.6 Master Receive Mode Operation Timing Example (2)
(MLS = ACKB = 0, WAIT = 1)

15.4.4 Slave Receive Operation

In slave receive mode, the master device outputs the transmit clock and transmit data, and the slave device returns an acknowledge signal. The reception procedure and operations in slave receive mode are described below.

1. Set the ICE bit in ICCR to 1. Set the MLS bit in ICMR and the MST and TRS bits in ICCR according to the operating mode.
2. When the start condition output by the master device is detected, the BBSY flag in ICCR is set to 1.
3. 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 in ICCR remains cleared to 0, and slave receive operation is performed.
4. At the 9th clock pulse of the receive frame, the slave device drives SDA low and returns an acknowledge signal. At the same time, the IRIC flag in ICCR is set to 1. If the IEIC bit in ICCR has been set to 1, an interrupt request is sent to the CPU. If the RDRF internal flag has been cleared to 0, it is set to 1, and the receive operation continues. If the RDRF internal flag has been set to 1 and ninth clock is received for the following data receival, the slave device drives SCL low from the falling edge of the receive clock until data is read into ICDR.

5. Read ICDR and clear the IRIC flag in ICCR to 0. The RDRF flag is cleared to 0.

Receive operations can be performed continuously by repeating steps [4] and [5]. When SDA is changed from low to high when SCL is high, and the stop condition is detected, the BBSY flag in ICCR is cleared to 0.

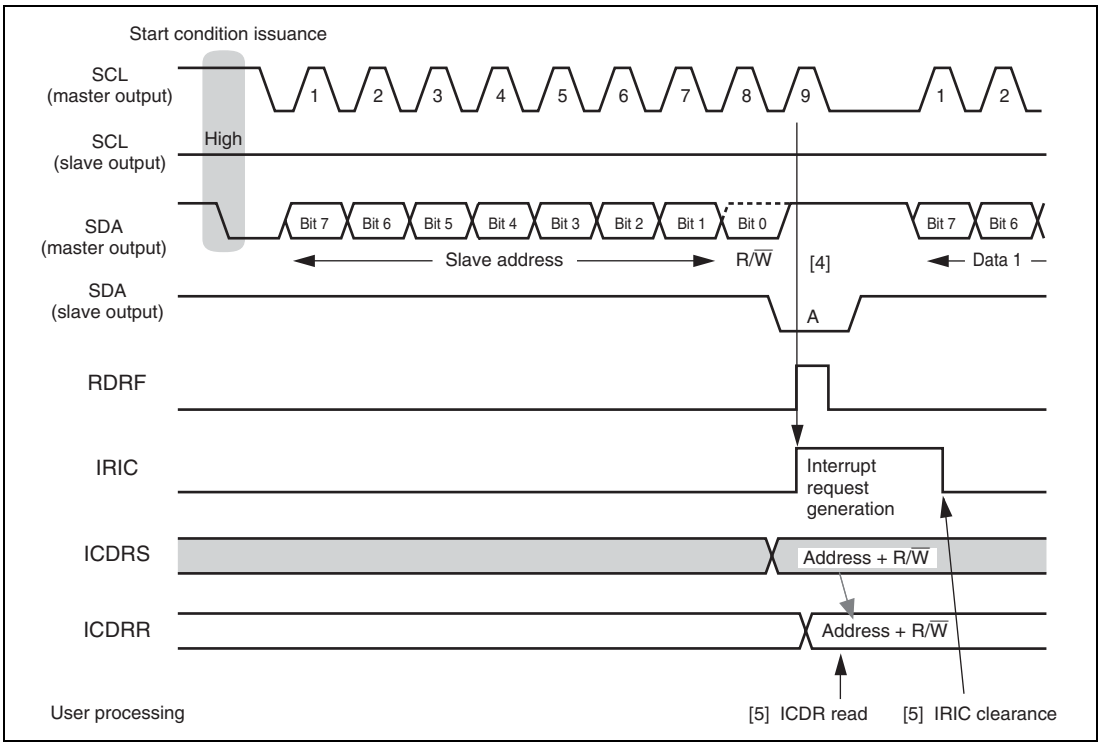


Figure 15.7 Example of Slave Receive Mode Operation Timing (1)
(MLS = ACKB = 0)

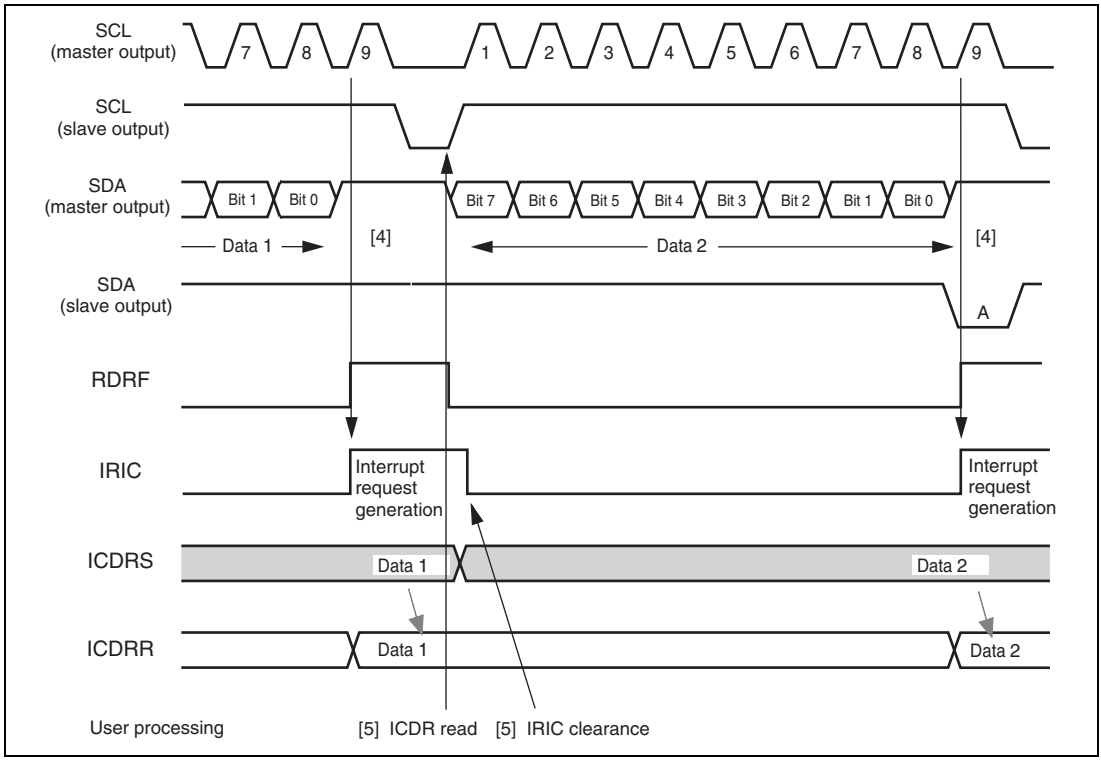


Figure 15.8 Example of Slave Receive Mode Operation Timing (2)
(MLS = ACKB = 0)

15.4.5 Slave Transmit Operation

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. Set the ICE bit in ICCR to 1. Set the MLS bit in ICMR and the MST and TRS bits in ICCR according to the operating mode.
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. At the same time, the IRIC flag in ICCR is set to 1. If the IEIC bit in ICCR has been set to 1, an interrupt request is sent to the CPU. If the 8th data bit (R/$\overline{W}$) is 1, the TRS bit in ICCR is set to 1, and the mode changes to slave transmit mode automatically. The TDRE internal flag is set to 1. The slave device drives SCL low from the fall of the transmit clock until ICDR data is written.
3. After clearing the IRIC flag to 0, write data to ICDR. The TDRE internal flag is cleared to 0. The written data is transferred to ICDRS, and the TDRE internal flag and the IRIC and IRTR flags are set to 1 again. After clearing the IRIC flag to 0, write the next data to ICDR. The slave device sequentially sends the data written into ICDR in accordance with the clock output by the master device at the timing shown in figure 15.9.
4. 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. If the TDRE internal flag has been set to 1, this slave device drives SCL low from the fall of the transmit clock until data is written to ICDR. 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 normally. When the TDRE internal flag is 0, the data written into ICDR is transferred to ICDRS, transmission is started, and the TDRE internal flag and the IRIC and IRTR flags are set to 1 again.
5. To continue transmission, clear the IRIC flag to 0, then write the next data to be transmitted into ICDR. The TDRE flag is cleared to 0.

Transmit operations can be performed continuously by repeating steps [4] and [5]. To end transmission, write H'FF to ICDR. When SDA is changed from low to high when SCL is high, and the stop condition is detected, the BBSY flag in ICCR is cleared to 0.

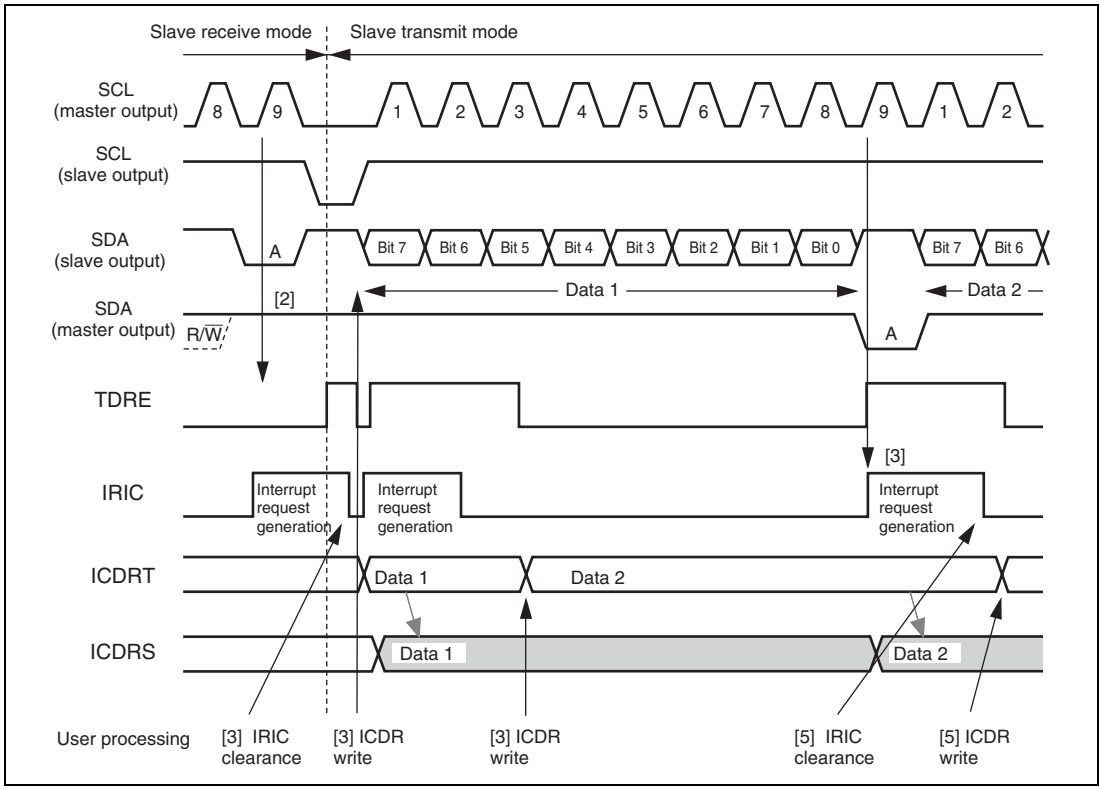


Figure 15.9 Example of Slave Transmit Mode Operation Timing (MLS = 0)

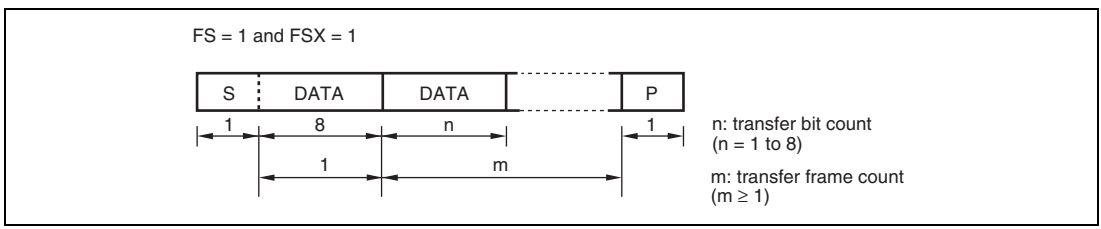


Figure 15.10 I²C Bus Data Format (Serial Format)

15.4.6 Clock Synchronous Serial Format

Serial format is a non-addressing format that has no acknowledge bit. Figure 15.10 shows this format.

15.4.7 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 TDRE or RDRF internal flag is set to 1, SCL is automatically held low after one frame has been transferred; this timing is synchronized with the internal clock. Figure 15.11 shows the IRIC set timing and SCL control.

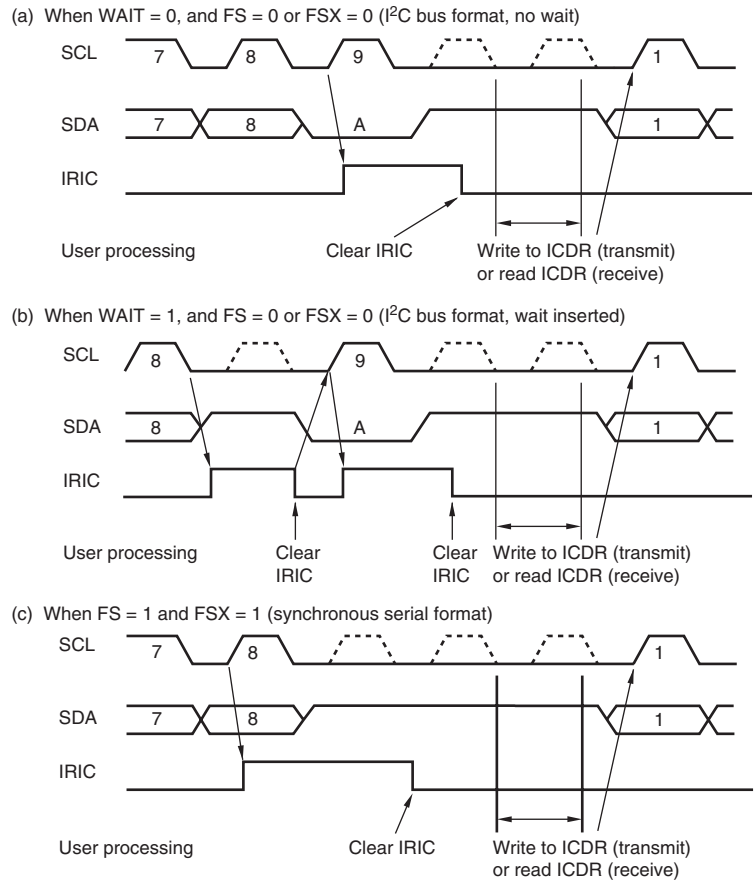


Figure 15.11 IRIC Setting Timing and SCL Control

15.4.8 Noise Canceler

The logic levels at the SCL and SDA pins are routed through noise cancelers before being latched internally. Figure 15.12 shows a block diagram of the noise canceler circuit.

The noise canceler consists of two cascaded latches and a match detector. The SCL (or SDA) 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.

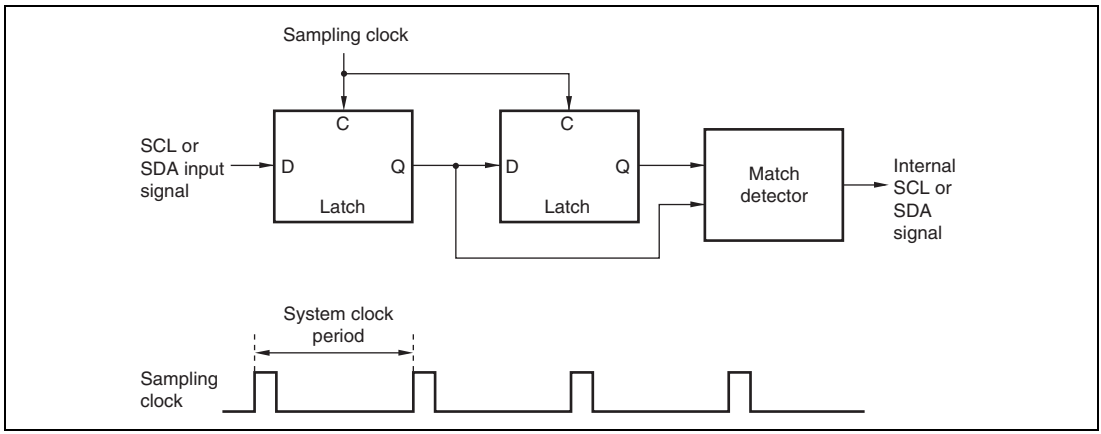


Figure 15.12 Block Diagram of Noise Canceler

15.4.9 Sample Flowcharts

Figures 15.13 to 15.16 show sample flowcharts for using the I²C bus interface in each mode.

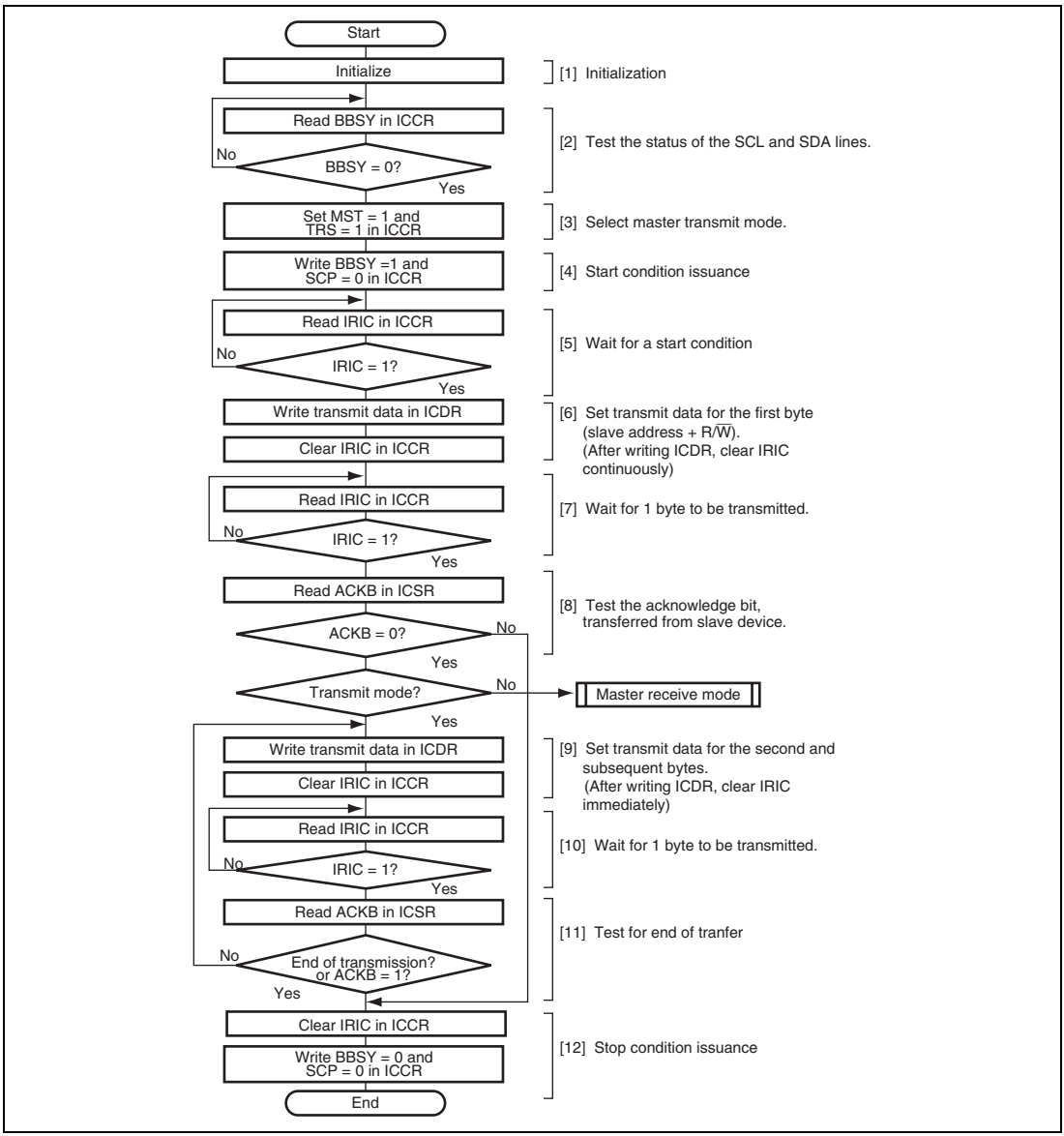


Figure 15.13 Sample Flowchart for Master Transmit Mode

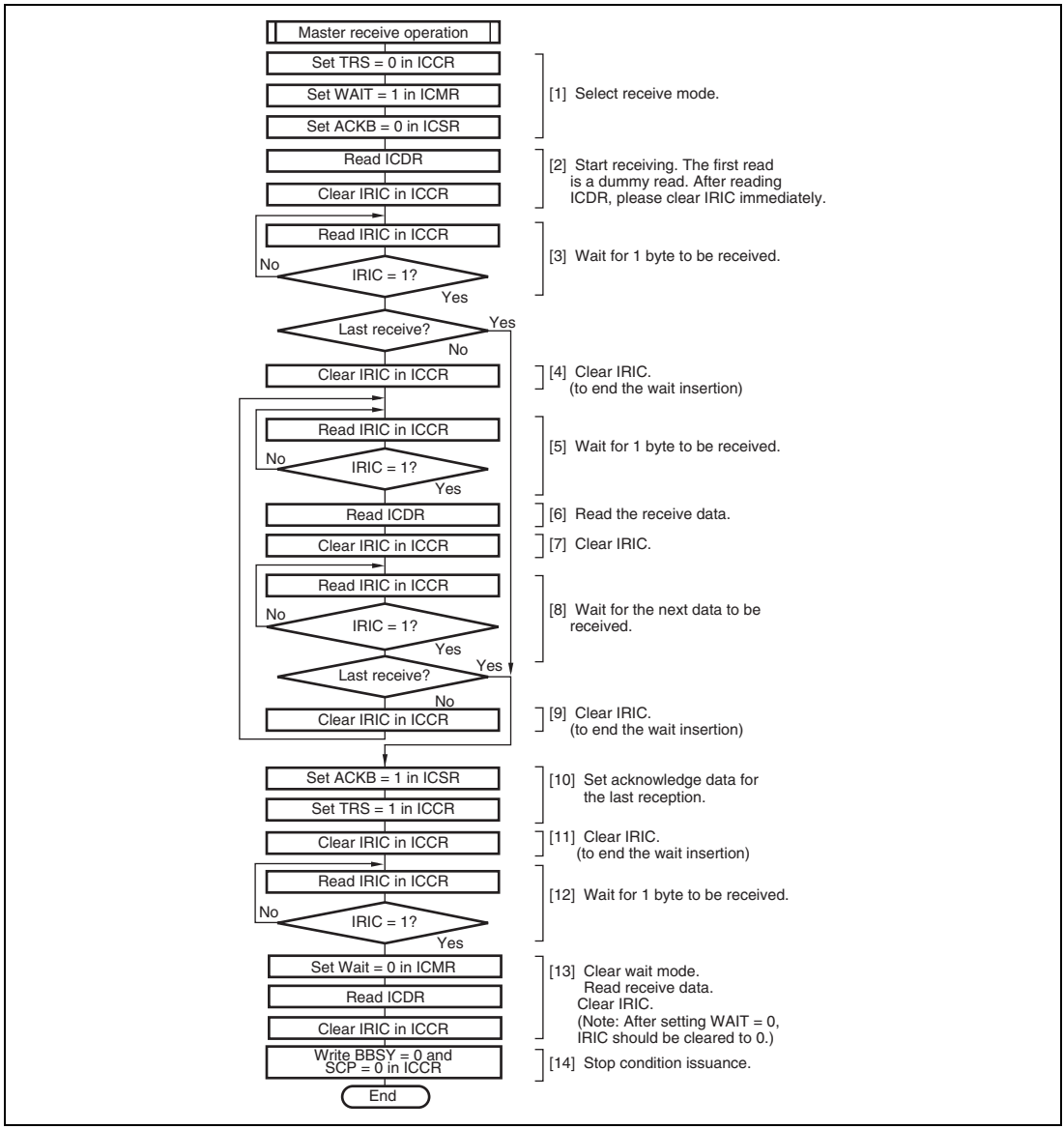


Figure 15.14 Sample Flowchart for Master Receive Mode

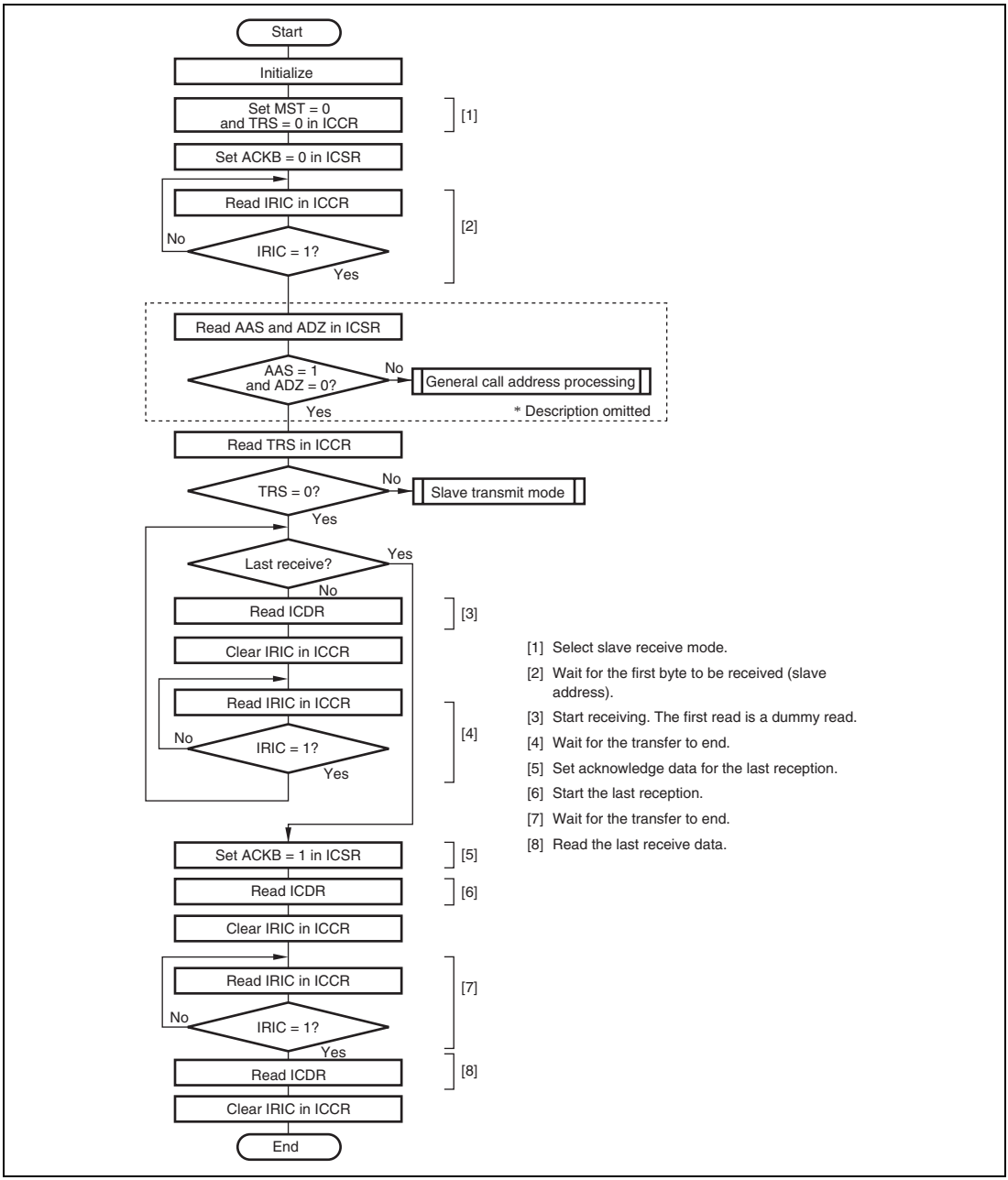
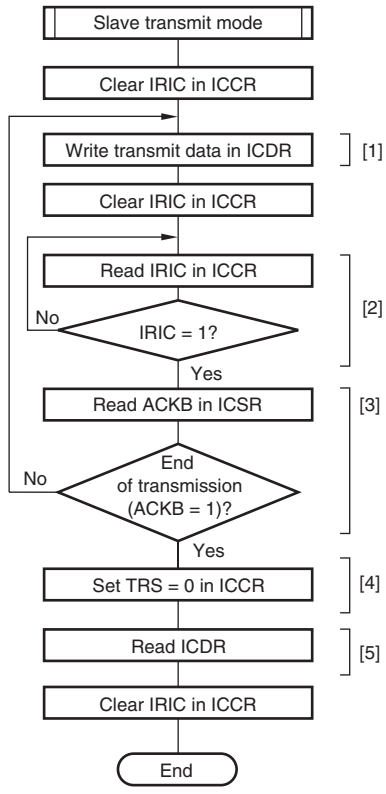


Figure 15.15 Sample Flowchart for Slave Receive Mode



[1] Set transmit data for the second and subsequent bytes.

[2] Wait for 1 byte to be transmitted.

[3] Test for end of transfer.

[4] Set slave receive mode.

[5] Dummy read (to release the SCL line).

Figure 15.16 Sample Flowchart for Slave Transmit Mode

15.5 Usage Notes

1. In master mode, if an instruction to generate a start condition is immediately followed by an instruction to generate a stop condition, neither condition will be output correctly. To output consecutive start and stop conditions, after issuing the instruction that generates the start condition, read the relevant ports, check that SCL and SDA are both low, then issue the instruction that generates the stop condition. Note that SCL may not yet have gone low when BBSY is cleared to 0.
2. Either of the following two conditions will start the next transfer. Pay attention to these conditions when reading or writing to ICDR.
   — Write access to ICDR when ICE = 1 and TRS = 1 (including automatic transfer from ICDRT to ICDRS)
   — Read access to ICDR when ICE = 1 and TRS = 0 (including automatic transfer from ICDRS to ICDRR)
3. Table 15.5 shows the timing of SCL and SDA output 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 15.5 I²C Bus Timing (SCL and SDA Output)

Item	Symbol	Output Timing	Unit	Notes
SCL output cycle time	t_{SCLO}	$28t_{cyc}$ to $256t_{cyc}$	ns	
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)		$1t_{SCLL} – 3t_{cyc}$	ns	
Data output hold time	t_{SDAHO}	$3t_{cyc}$	ns	

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 table 20-4 in section 20, 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 under 1000 ns (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 the table in table 15.6.

Table 15.6 Permissible SCL Rise Time (t_{sr}) Values

			Time Indication				
IICX	t_{cyc} Indication		I²C Bus Specification (Max.)	φ = 5 MHz	φ = 8 MHz	φ = 10 MHz	φ = 16 MHz
0	$7.5t_{cyc}$	Normal mode	1000 ns	1000 ns	937 ns	750 ns	468 ns
		High-speed mode	300 ns	300 ns	300 ns	300 ns	300 ns
1	$17.5t_{cyc}$	Normal mode	1000 ns	1000 ns	1000 ns	1000 ns	1000 ns
		High-speed mode	300 ns	300 ns	300 ns	300 ns	300 ns

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_{Scyc} and t_{cyc}, as shown in table 15.5. However, because of the rise and fall times, the I²C bus interface specifications may not be satisfied at the maximum transfer rate. Table 15.7 shows output timing calculations for different operating frequencies, including the worst-case influence of rise and fall times. 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.

   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 15.7 I²C Bus Timing (with Maximum Influence of t_{sr}/t_{sf})

			Time Indication (at Maximum Transfer Rate) [ns]					
Item	t_{cyc} Indication		t_{sr}/t_{sf} Influence (Max.)	I²C Bus Specification (Min.)	ϕ = 5 MHz	ϕ = 8 MHz	ϕ = 10 MHz	ϕ = 16 MHz
t_{SCLHO}	$0.5t_{SCLO}$ ($-t_{Sr}$)	Standard mode	–1000	4000	4000	4000	4000	4000
		High-speed mode	–300	600	950	950	950	950
t_{SCLLO}	$0.5t_{SCLO}$ ($-t_{Sf}$)	Standard mode	–250	4700	4750	4750	4750	4750
		High-speed mode	–250	1300	1000*[1]	1000*[1]	1000*[1]	1000*[1]
t_{BUFO}	$0.5t_{SCLO}-1t_{cyc}$ ($-t_{Sr}$)	Standard mode	–1000	4700	3800*[1]	3875*[1]	3900*[1]	3938*[1]
		High-speed mode	–300	1300	750*[1]	825*[1]	850*[1]	888*[1]
t_{STAHO}	$0.5t_{SCLO}-1t_{cyc}$ ($-t_{Sf}$)	Standard mode	–250	4000	4550	4625	4650	4688
		High-speed mode	–250	600	800	875	900	938
t_{STASO}	$1t_{SCLO}$ ($-t_{Sr}$)	Standard mode	–1000	4700	9000	9000	9000	9000
		High-speed mode	–300	600	2200	2200	2200	2200
t_{STOSO}	$0.5t_{SCLO}+2t_{cyc}$ ($-t_{Sr}$)	Standard mode	–1000	4000	4400	4250	4200	4125
		High-speed mode	–300	600	1350	1200	1150	1075
t_{SDASO} (master)	$1t_{SCLLO}$*[2] $-3t_{cyc}$ ($-t_{Sr}$)	Standard mode	–1000	250	3100	3325	3400	3513
		High-speed mode	–300	100	400	625	700	813
t_{SDASO} (slave)	$1t_{SCLL}$*[2] $-3t_{cyc}$ ($-t_{Sr}$)	Standard mode	–1000	250	3100	3325	3400	3513
		High-speed mode	–300	100	400	625	700	813
t_{SDAHO}	$3t_{cyc}$	Standard mode	0	0	600	375	300	188
		High-speed mode	0	0	600	375	300	188

Notes: 1. Does not meet the I²C bus interface specification

2. Calculated using the I²C bus specification values (standard mode: 4700 ns min.; high-speed mode: 1300 ns min.).

7. Note on ICDR Read at end of Master Reception

   To halt reception after completion 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 the SDA pin from low to high when the SCL pin 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, 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.

8. Notes on Start Condition Issuance for Retransmission

   Depending on the timing combination with the start condition issuance and the subsequently writing data to ICDR, it may not be possible to issue the retransmission and the data transmission after retransmission condition issuance.

   After start condition issuance is done and determined the start condition, write the transmit data to ICDR, as shown below. Figure 15.17 shows the timing of start condition issuance for retransmission, and the timing for subsequently writing data to ICDR, together with the corresponding flowchart.

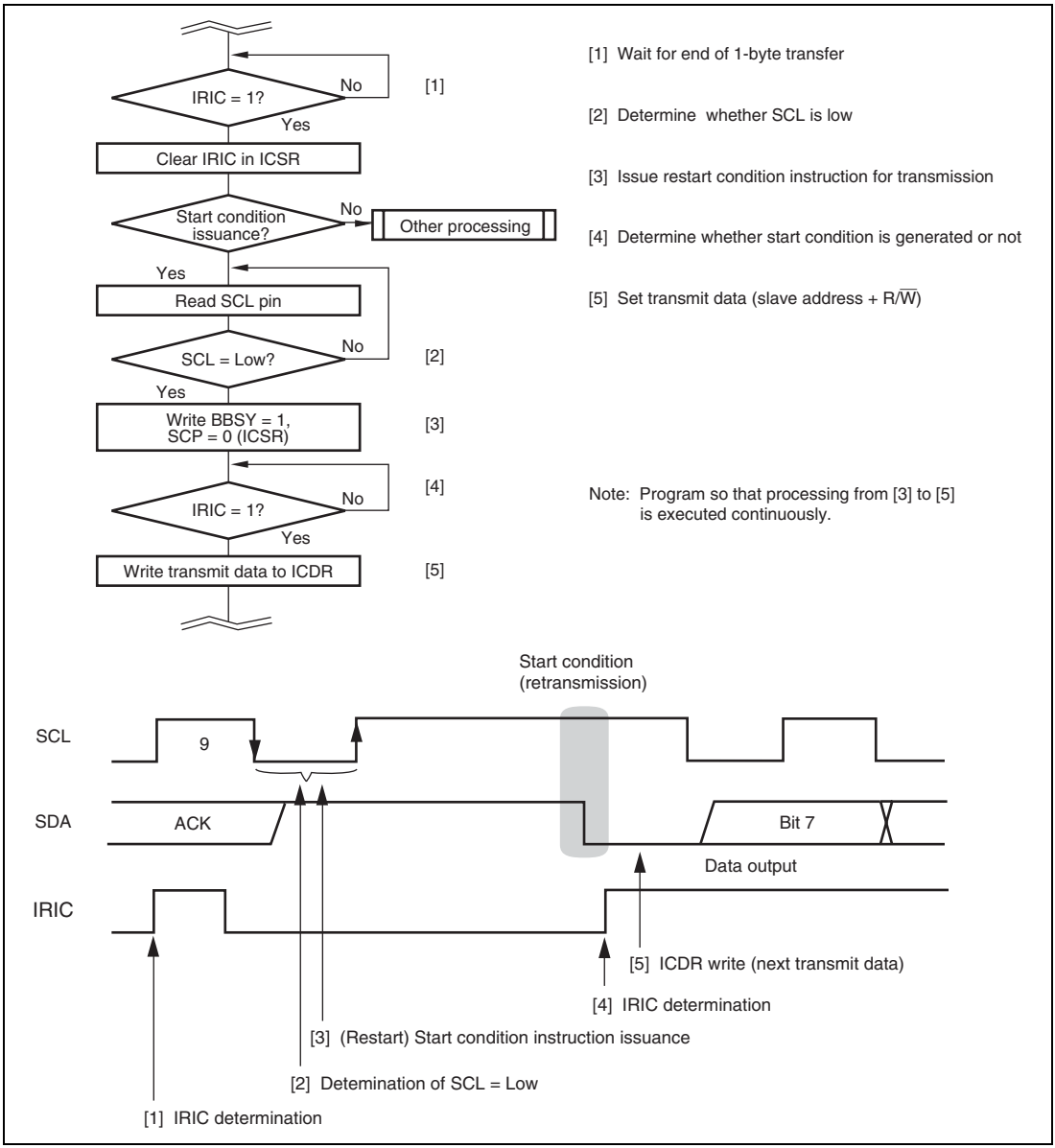


Figure 15.17 Flowchart and Timing of Start Condition Instruction Issuance for Retransmission

- Notes on WAIT Function
  - — Conditions to cause this phenomenon

    When both of the following conditions are satisfied, the clock pulse of the 9th clock could be outputted continuously in master mode using the WAIT function due to the failure of the WAIT insertion after the 8th clock fall.

    (1) Setting the WAIT bit of the ICMR register to 1 and operating WAIT, in master mode

    (2) If the IRIC bit of interrupt flag is cleared from 1 to 0 between the fall of the 7th clock and the fall of the 8th clock.

  - — Error phenomenon

    Normally, WAIT state will be cancelled by clearing the IRIC flag bit from 1 to 0 after the fall of the 8th clock in WAIT state. In this case, if the IRIC flag bit is cleared between the 7th clock fall and the 8th clock fall, the IRIC flag clear- data will be retained internally. Therefore, the WAIT state will be cancelled right after WAIT insertion on 8th clock fall.

  - — Restrictions

    Please clear the IRIC flag before the rise of the 7th clock (the counter value of BC2 through BC0 should be 2 or greater), after the IRIC flag is set to 1 on the rise of the 9th clock.

    If the IRIC flag-clear is delayed due to the interrupt or other processes and the value of BC counter is turned to 1 or 0, please confirm the SCL pins are in the low state after the counter value of BC2 through BC0 is turned to 0, and clear the IRIC flag. (See figure 15.18.)

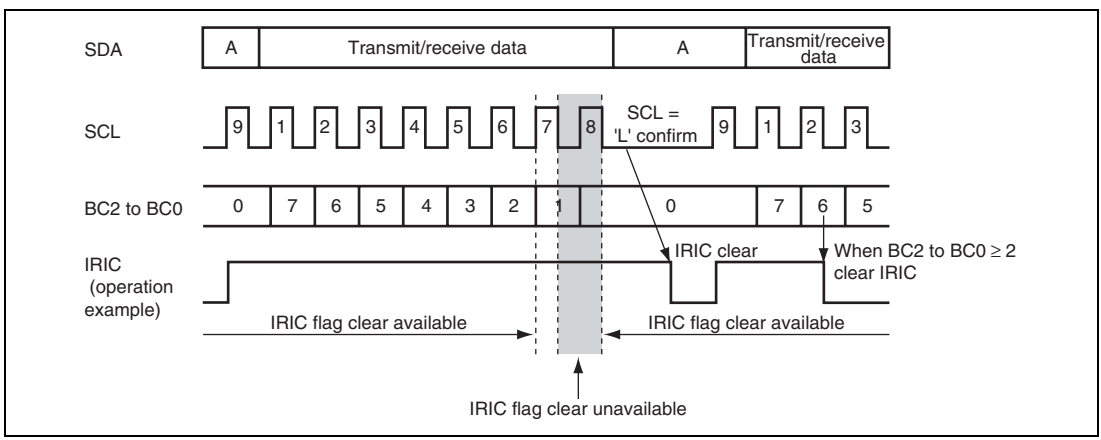


Figure 15.18 IRIC Flag Clear Timing on WAIT Operation

- Notes on TRS Bit Setting and ICDR Register Access

  — Conditions to cause this failure

  Low-fixation of the SCL pins is cancelled incorrectly when the following conditions are satisfied.

  Master mode:

  Figure 15.19 shows the notes on ICDR reading (TRS = 1) in master mode.

  (a) When previously received 2-byte data remains in ICDR unread (ICDRS are full).

  (b) Reads ICDR register after switching to transmit mode (TRS = 1). (RDRF = 0 state)

  (c) Sets to receive mode (TRS = 0), after transmitting the first frame of issued start condition by master mode.

  Slave mode:

  Figure 15.20 shows the notes on ICDR writing (TRS = 0) in slave mode.

  (a) Writes ICDR register in receive mode (TRS = 0), after entering the start condition by slave mode (TDRE = 0 state).

  (b) Address match with the first frame, receive 1 by R/W bit, and switches to transmit mode (TRS = 1).

  When these conditions are satisfied, the low fixation of the SCL pins is cancelled without ICDR register access after the first frame is transferred.

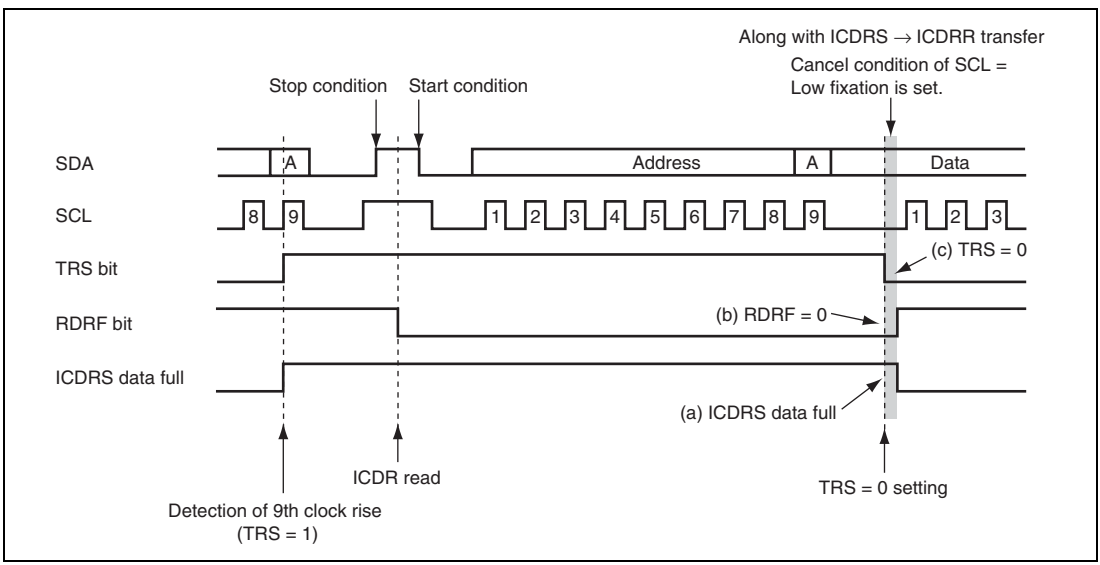


Figure 15.19 Notes on ICDR Reading with TRS = 1 Setting in Master Mode

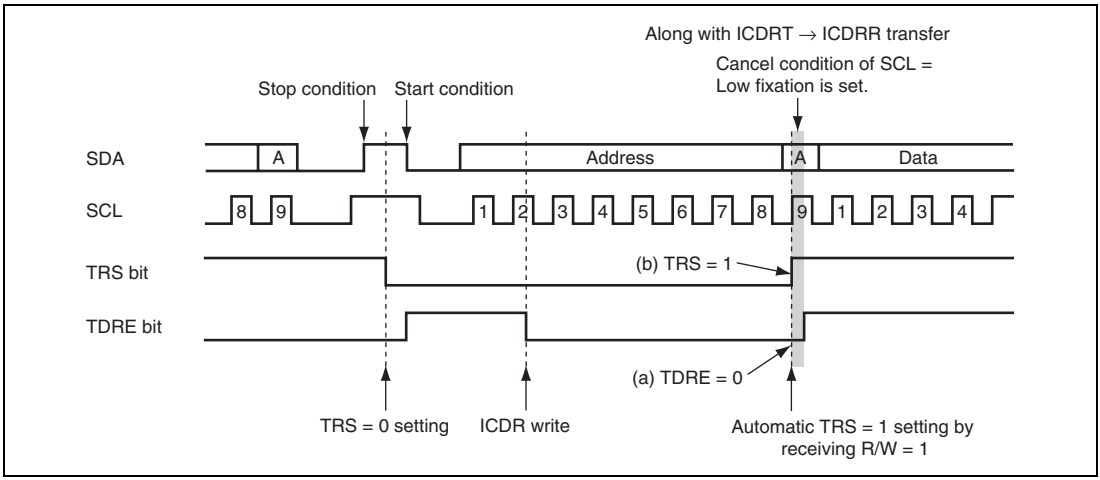


Figure 15.20 Notes on ICDR Writing with TRS = 0 Setting in Slave Mode

— Restriction

Please carry out the following countermeasures when transmitting/receiving via the IIC bus interface module.

(1) Please read the ICDR registers in receive mode, and write them in transmit mode.

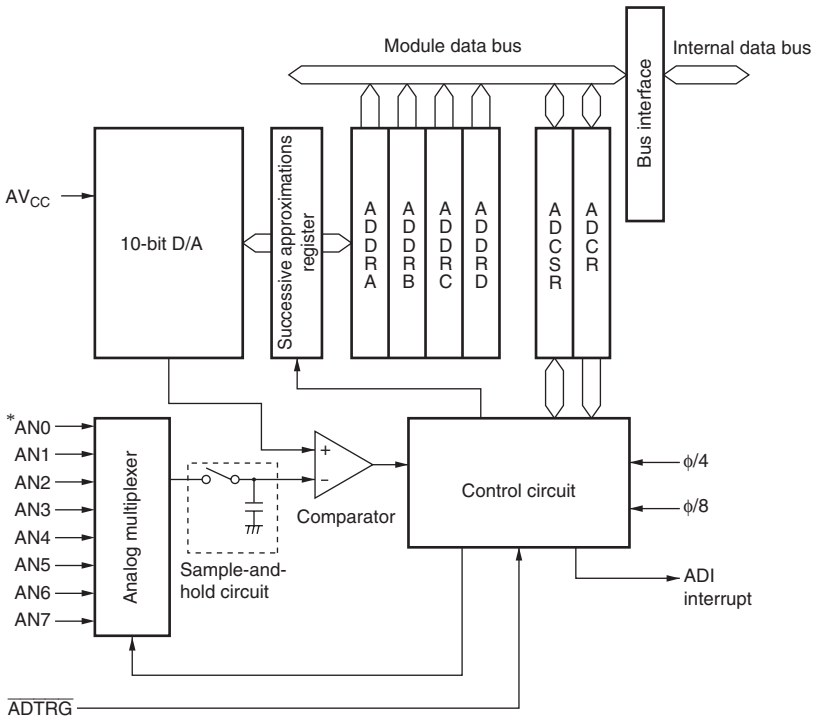
(2) In receiving operation with master mode, please issue the start condition after clearing the internal flag of the IIC bus interface module, using CLR3 to CLR0 bit of the DDCSWR register on bus-free state (BBSY = 0).

Section 16 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. The block diagram of the A/D converter is shown in figure 16.1.

16.1 Features

- 10-bit resolution
- Eight input channels (four channels for the 42-pin version)
- Conversion time: at least 4.4 μs per channel (at 16 MHz operation)
- Two operating modes
  - — Single mode: Single-channel A/D conversion
  - — Scan mode: Continuous A/D conversion on 1 to 4 channels
- Four data registers
  - — Conversion results are held in a 16-bit data register for each channel
- Sample and hold function
- Two conversion start methods
  - — Software
  - — External trigger signal
- Interrupt request
  - — An A/D conversion end interrupt request (ADI) can be generated



[Legend]
ADCR: A/D control register
ADCSR: A/D control/status register
ADDRA: A/D data register A
ADDRB: A/D data register B
ADDRC: A/D data register C
ADDRD: A/D data register D

Note: AN4, AN5, AN6, and AN7 do not exist in the 42-pin version.

Figure 16.1 Block Diagram of A/D Converter

16.2 Input/Output Pins

Table 16.1 summarizes the input pins used by the A/D converter. The eight analog input pins are divided into two groups; analog input pins 0 to 3 (AN0 to AN3) comprising group 0, analog input pins 4 to 7 (AN4 to AN7) comprising group 1. The AVcc pin is the power supply pin for the analog block in the A/D converter.

Table 16.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Analog power supply pin	AV_{CC}	Input	Analog block power supply pin
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

16.3 Register Description

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)

16.3.1 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 16.2.

The converted 10-bit data is stored in bits 6 to 15. The lower 6 bits are always read as 0.

The data bus between the CPU and the A/D converter is 8 bits wide. The upper byte can be read directly from the CPU, however the lower byte should be read via a temporary register. The temporary register contents are transferred from the ADDR when the upper byte data is read. Therefore, byte access to ADDR should be done by reading the upper byte first then the lower one. Word access is also possible. ADDR is initialized to H'0000.

Table 16.2 Analog Input Channels and Corresponding ADDR Registers

Analog Input Channel		
Group 0	**Group 1**	**A/D Data Register to be Stored Results of A/D Conversion**
AN0	AN4	ADDRA
AN1	AN5	ADDRB
AN2	AN6	ADDRC
AN3	AN7	ADDRD

16.3.2 A/D Control/Status Register (ADCSR)

ADCSR consists of the control bits and conversion end status bits of the A/D converter.

Bit	Bit Name	Initial Value	R/W	Description
7	ADF	0	R/W	A/D End Flag [Setting conditions] • When A/D conversion ends in single mode • When A/D conversion ends on all the channels selected in scan mode [Clearing condition] When 0 is written after reading ADF = 1
6	ADIE	0	R/W	A/D Interrupt Enable A/D conversion end interrupt (ADI) request enabled by ADF when 1 is set
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 is complete. 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.
4	SCAN	0	R/W	Scan Mode Selects single mode or scan mode as the A/D conversion operating mode. 0: Single mode 1: Scan mode
3	CKS	0	R/W	Clock Select Selects the A/D conversions time 0: Conversion time = 134 states (max.) 1: Conversion time = 70 states (max.) Clear the ADST bit to 0 before switching the conversion time.

Bit	Bit Name	Initial Value	R/W	Description	
2	CH2	0	R/W	Channel Select 0 to 2	
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 to AN1
				010: AN2	010: AN0 to AN2
				011: AN3	011: AN0 to AN3
				100: AN4	100: AN4
				101: AN5	101: AN4 to AN5
				110: AN6	110: AN4 to AN6
				111: AN7	111: AN4 to AN7
				AN4, AN5, AN6, and AN7 do not exist in the 42-pin version.	

16.3.3 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	TRGE	0	R/W	Trigger Enable A/D conversion is started at the falling edge and the rising edge of the external trigger signal ($\overline{\text{ADTRG}}$) when this bit is set to 1. The selection between the falling edge and rising edge of the external trigger pin ($\overline{\text{ADTRG}}$) conforms to the WPEG5 bit in the interrupt edge select register 2 (IEGR2).
6 to 1	—	All 1	—	Reserved These bits are always read as 1.
0	—	0	R/W	Reserved Do not set this bit to 1, though the bit is readable/writable.

16.4 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, in order to prevent incorrect operation, first clear the bit ADST to 0 in ADCSR. The ADST bit can be set at the same time as the operating mode or analog input channel is changed.

16.4.1 Single Mode

In single mode, A/D conversion is performed once for the analog input on the specified single channel as follows:

1. A/D conversion is started when the ADST bit in ADCSR is set to 1, according to software or external trigger input.
2. When A/D conversion is completed, the result is transferred to the corresponding A/D data register to the channel.
3. On completion of 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 A/D conversion ends, the ADST bit is automatically cleared to 0 and the A/D converter enters the wait state.

16.4.2 Scan Mode

In scan mode, A/D conversion is performed sequentially for the analog input on the specified channels (four channels maximum) as follows:

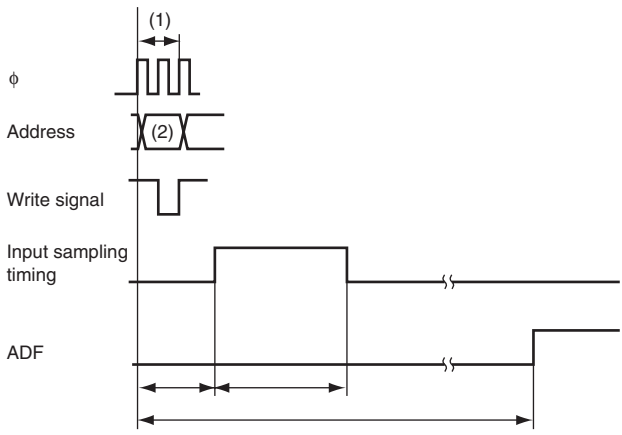
1. When the ADST bit is set to 1 by software, or external trigger input, A/D conversion starts on the first channel in the group (AN0 when CH2 = 0, AN4 when CH2 = 1).
2. When A/D conversion for each channel is completed, the result is sequentially transferred to the A/D data register corresponding to each channel.
3. When conversion of all the selected channels is completed, the ADF flag in ADCSR is set to 1. If the ADIE bit is set to 1 at this time, an ADI interrupt is requested. Conversion of the first channel in the group starts again.
4. The ADST bit is not automatically cleared to 0. Steps [2] to [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.

16.4.3 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) has passed after the ADST bit is set to 1, then starts conversion. Figure 16.2 shows the A/D conversion timing. Table 16.3 shows the A/D conversion time.

As indicated in figure 16.2, the A/D conversion time includes t_D and the input sampling time. The length of t_D varies depending on the timing of the write access to ADCSR. The total conversion time therefore varies within the ranges indicated in table 16.3.

In scan mode, the values given in table 16.3 apply to the first conversion time. In the second and subsequent conversions, the conversion time is 128 states (fixed) when CKS = 0 and 66 states (fixed) when CKS = 1.



[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 16.2 A/D Conversion Timing

Table 16.3 A/D Conversion Time (Single Mode)

		CKS = 0			CKS = 1		
Item	**Symbol**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**
A/D conversion start delay	t_D	6	—	9	4	—	5
Input sampling time	t_{SPL}	—	31	—	—	15	—
A/D conversion time	t_{CONV}	131	—	134	69	—	70

Note: All values represent the number of states.

16.4.4 External Trigger Input Timing

The A/D conversion can also be started by an external trigger input. When the TRGE bit is set to 1 in ADCR, external trigger input is enabled at the $\overline{\text{ADTRG}}$ pin. A falling edge at the $\overline{\text{ADTRG}}$ input pin sets the ADST bit to 1 in ADCSR, starting A/D conversion. Other operations, in both single and scan modes, are the same as when the bit ADST has been set to 1 by software. Figure 16.3 shows the timing.

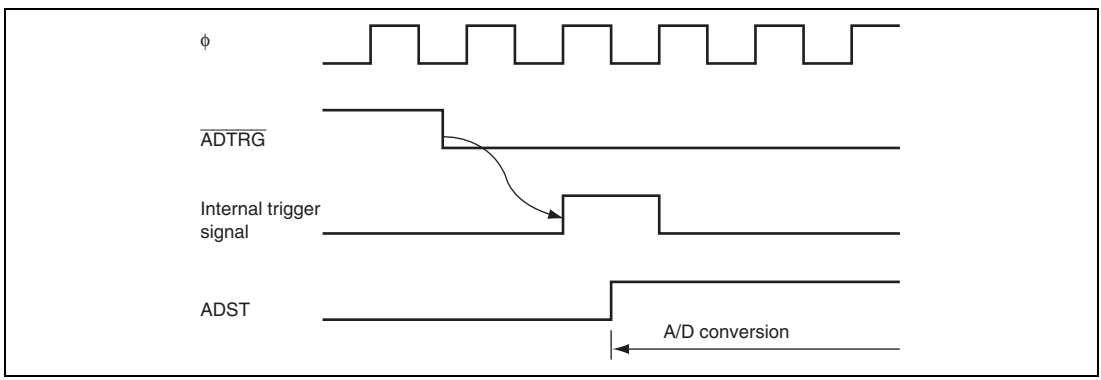


Figure 16.3 External Trigger Input Timing

16.5 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 16.4).
- Offset error

  The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from the minimum voltage value 0000000000 to 0000000001 (see figure 16.5).
- Full-scale error

  The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from 1111111110 to 1111111111 (see figure 16.5).
- Nonlinearity error

  The error with respect to the ideal A/D conversion characteristics between zero voltage and full-scale voltage. Does not include offset error, full-scale error, or quantization error.
- Absolute accuracy

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

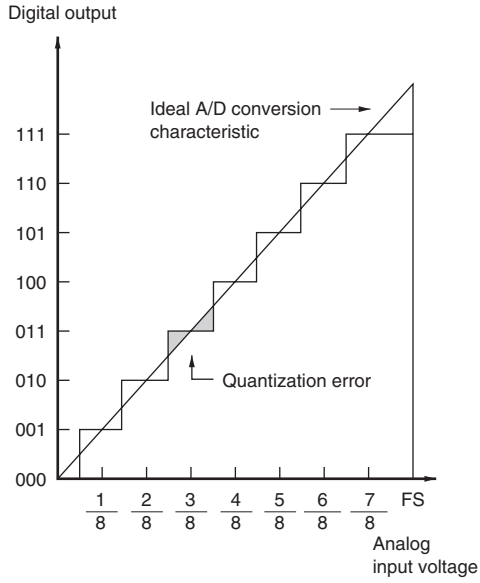


Figure 16.4 A/D Conversion Accuracy Definitions (1)

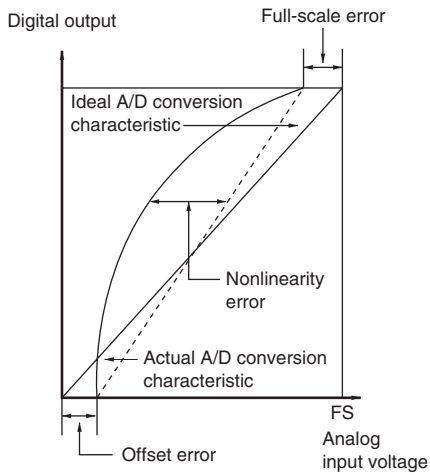


Figure 16.5 A/D Conversion Accuracy Definitions (2)

16.6 Usage Notes

16.6.1 Permissible Signal Source Impedance

This LSI's analog input is designed such that 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 A/D conversion accuracy. However, for A/D conversion in single mode with a large capacitance provided externally, the input load will essentially comprise only the internal input resistance of 10 kΩ, and the signal source impedance is ignored. However, as 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., 5 mV/µs or greater) (see figure 16.6). When converting a high-speed analog signal or converting in scan mode, a low-impedance buffer should be inserted.

16.6.2 Influences on Absolute Accuracy

Adding capacitance results in coupling with GND, and therefore noise in GND may adversely affect absolute accuracy. Be sure to make the connection to an electrically stable GND.

Care is also required to ensure that filter circuits do not interfere with digital signals or act as antennas on the mounting board.

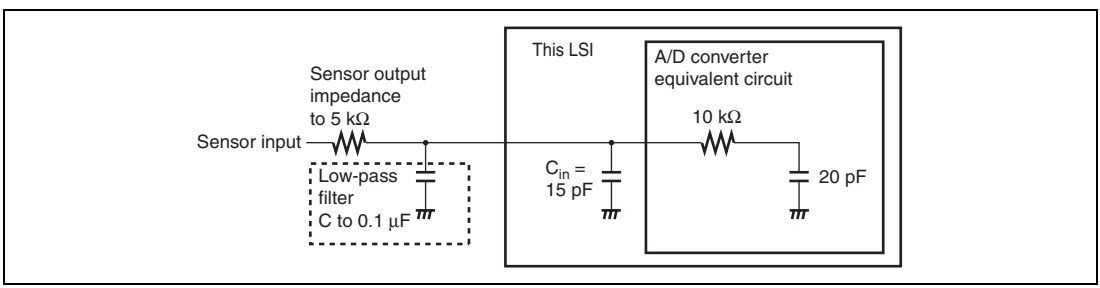


Figure 16.6 Analog Input Circuit Example

Section 17 EEPROM

This LSI has an on-chip 512-byte EEPROM. The block diagram of the EEPROM is shown in figure 17.1.

17.1 Features

- Two writing methods:
  1-byte write
  Page write: Page size 8 bytes
- Three reading methods:
  Current address read
  Random address read
  Sequential read
- Acknowledge polling possible
- Write cycle time:
  10 ms (power supply voltage Vcc = 2.7 V or more)
- Write/Erase Endurance:
  10^4 cycles/byte (byte write mode), 10^5 cycles/page (page write mode)
- Data retention:
  10 years after the write cycle of 10^4 cycles (page write mode)
- Interface with the CPU
  I^2C bus interface (complies with the standard of Philips Corporation)
  Device code 1010
  Sleep address code can be changed (initial value: 000))
  The I^2C bus is open to the outside, so the EEPROM can be directly accessed from the outside.

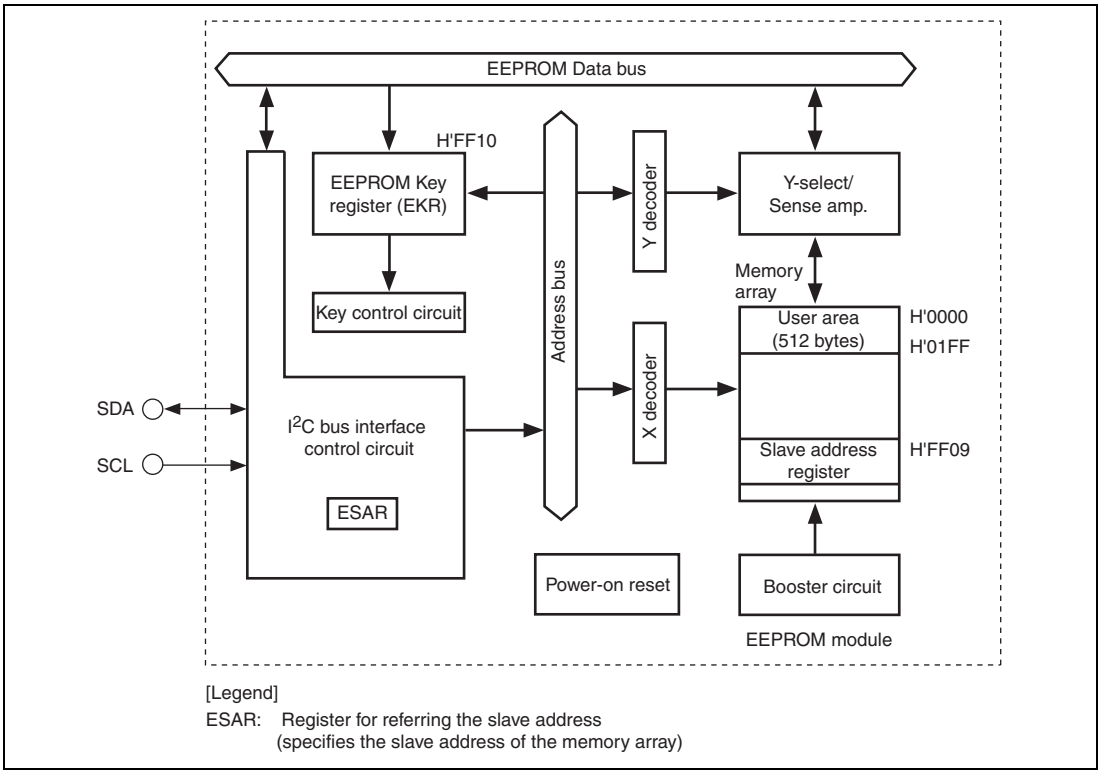


Figure 17.1 Block Diagram of EEPROM

17.2 Input/Output Pins

Pins used in the EEPROM are listed in table 17.1.

Table 17.1 Pin Configuration

Pin name	Symbol	Input/ Output	Function
Serial clock pin	SCL	Input	The SCL pin is used to control serial input/output data timing. The data is input at the rising edge of the clock and output at the falling edge of the clock. The SCL pin needs to be pulled up by resistor as that pin is open-drain driven structure of the I^2C pin. Use proper resistor value for your system by considering V_{OL}, I_{OL}, and the C_{IN} pin capacitance in section 20.2.2, DC Characteristics and in section 20.2.3, AC Characteristics. Maximum clock frequency is 400 kHz.
Serial data pin	SDA	Input/ Output	The SDA pin is bidirectional for serial data transfer. The SDA pin needs to be pulled up by resistor as that pin is open-drain driven structure. Use proper resistor value for your system by considering V_{OL}, I_{OL}, and the C_{IN} pin capacitance in section 20.2.2, DC Characteristics and in section 20.2.3, AC Characteristics. Except for a start condition and a stop condition which will be discussed later, the high-to-low and low-to-high change of SDA input should be done during SCL low periods.

17.3 Register Description

The EEPROM has a following register.

- EEPROM key register (EKR)

17.3.1 EEPROM Key Register (EKR)

EKR is an 8-bit readable/writable register, which changes the slave address code written in the EEPROM. The slave address code is changed by writing H'5F in EKR and then writing either of H'00 to H'07 as an address code to the H'FF09 address in the EEPROM by the byte write method. EKR is initialized to H'FF.

17.4 Operation

17.4.1 EEPROM Interface

This LSI has a multi-chip structure with two internal chips of F-ZTAT™ HD64F3664 and 512-byte EEPROM.

The EEPROM interface is the I²C bus interface. This I²C bus is open to the outside, so the communication with the external devices connected to the I²C bus can be made.

17.4.2 Bus Format and Timing

The I²C bus format and the I²C bus timing follow section 15.4.1, I²C Bus Data Format. The bus formats specific for the EEPROM are the following two.

1. The EEPROM address is configured of two bytes, the write data is transferred in the order of upper address and lower address from each MSB side.
2. The write data is transmitted from the MSB side.

The bus format and bus timing of the EEPROM are shown in figure 17.2.

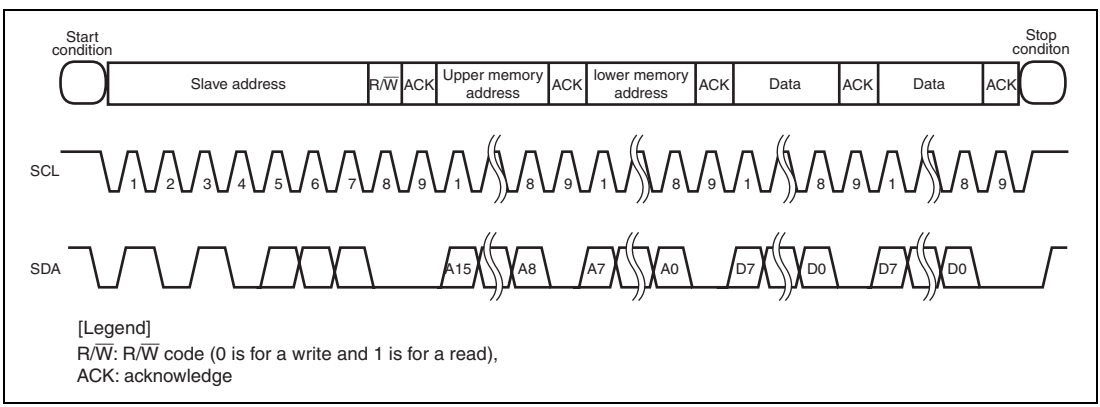


Figure 17.2 EEPROM Bus Format and Bus Timing

17.4.3 Start Condition

A high-to-low transition of the SDA input with the SCL input high is needed to generate the start condition for starting read, write operation.

17.4.4 Stop Condition

A low-to-high transition of the SDA input with the SCL input high is needed to generate the stop condition for stopping read, write operation.

The standby operation starts after a read sequence by a stop condition. In the case of write operation, a stop condition terminates the write data inputs and place the device in an internally-timed write cycle to the memories. After the internally-timed write cycle (t_{WC}) which is specified as t_{WC}, the device enters a standby mode.

17.4.5 Acknowledge

All address data and serial data such as read data and write data are transmitted to and from in 8-bit unit. The acknowledgement is the signal that indicates that this 8-bit data is normally transmitted to and from.

In the write operation, EEPROM sends "0" to acknowledge in the ninth cycle after receiving the data. In the read operation, EEPROM sends a read data following the acknowledgement after receiving the data. After sending read data, the EEPROM enters the bus open state. If the EEPROM receives "0" as an acknowledgement, it sends read data of the next address. If the EEPROM does not receive acknowledgement "0" and receives a following stop condition, it stops the read operation and enters a standby mode. If the EEPROM receives neither acknowledgement "0" nor a stop condition, the EEPROM keeps bus open without sending read data.

17.4.6 Slave Addressing

The EEPROM device receives a 7-bit slave address and a 1-bit R/$\overline{W}$ code following the generation of the start conditions. The EEPROM enables the chip for a read or a write operation with this operation.

The slave address consists of a former 4-bit device code and latter 3-bit slave address as shown in table 17.2. The device code is used to distinguish device type and this LSI uses "1010" fixed code in the same manner as in a general-purpose EEPROM. The slave address code selects one device out of all devices with device code 1010 (8 devices in maximum) which are connected to the I^2C bus. This means that the device is selected if the inputted slave address code received in the order of A2, A1, A0 is equal to the corresponding slave address reference register (ESAR).

The slave address code is stored in the address H'FF09 in the EEPROM. It is transferred to ESAR from the slave address register in the memory array during 10 ms after the reset is released. An access to the EEPROM is not allowed during transfer.

The initial value of the slave address code written in the EEPROM is H'00. It can be written in the range of H'00 to H'07. Be sure to write the data by the byte write method.

The next one bit of the slave address is the R/$\overline{W}$ code. 0 is for a write and 1 is for a read.

The EEPROM turns to a standby state if the device code is not "1010" or slave address code doesn't coincide.

Table 17.2 Slave Addresses

Bit	Bit name	Initial Value	Setting Value	Remarks
7	Device code D3	—	1	
6	Device code D2	—	0	
5	Device code D1	—	1	
4	Device code D0	—	0	
3	Slave address code A2	0	A2	The initial value can be changed
2	Slave address code A1	0	A1	The initial value can be changed
1	Slave address code A0	0	A0	The initial value can be changed

17.4.7 Write Operations

There are two types write operations; byte write operation and page write operation. To initiate the write operation, input 0 to R/$\overline{W}$ code following the slave address.

1. Byte Write

   A write operation requires an 8-bit data of a 7-bit slave address with R/$\overline{W}$ code = "0". Then the EEPROM sends acknowledgement "0" at the ninth bit. This enters the write mode. Then, two bytes of the memory address are received from the MSB side in the order of upper and lower. Upon receipt of one-byte memory address, the EEPROM sends acknowledgement "0" and receives a following a one-byte write data. After receipt of write data, the EEPROM sends acknowledgement "0". If the EEPROM receives a stop condition, the EEPROM enters an internally controlled write cycle and terminates receipt of SCL and SDA inputs until completion of the write cycle. The EEPROM returns to a standby mode after completion of the write cycle.

   The byte write operation is shown in figure 17.3.

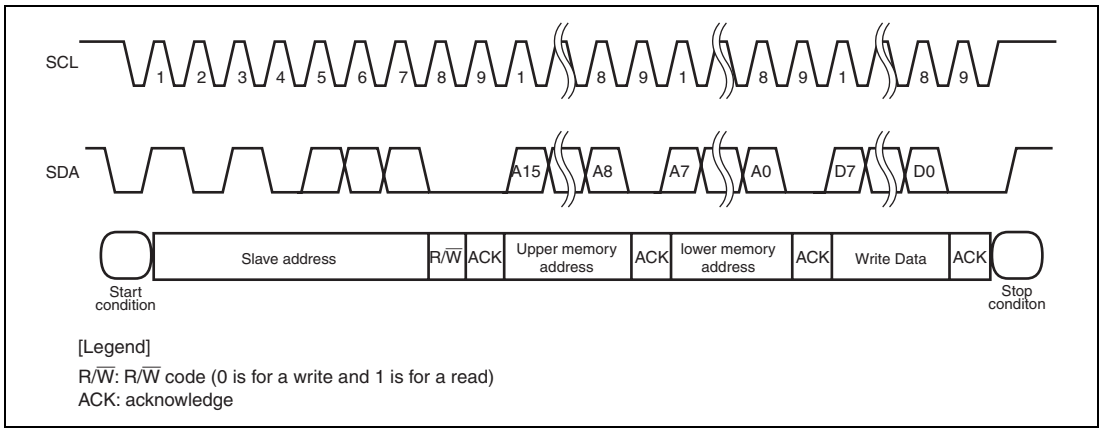


Figure 17.3 Byte Write Operation

2. Page Write

   This LSI is capable of the page write operation which allows any number of bytes up to 8 bytes to be written in a single write cycle. The write data is input in the same sequence as the byte write in the order of a start condition, slave address + R/$\overline{W}$ code, memory address (n), and write data (Dn) with every ninth bit acknowledgement "0" output. The EEPROM enters the page write operation if the EEPROM receives more write data (Dn+1) is input instead of receiving a stop condition after receiving the write data (Dn). LSB 3 bits (A2 to A0) in the EEPROM address are automatically incremented to be the (n+1) address upon receiving write data (Dn+1). Thus the write data can be received sequentially.

Addresses in the page are incremented at each receipt of the write data and the write data can be input up to 8 bytes. If the LSB 3 bits (A2 to A0) in the EEPROM address reach the last address of the page, the address will roll over to the first address of the same page. When the address is rolled over, write data is received twice or more to the same address, however, the last received data is valid. At the receipt of the stop condition, write data reception is terminated and the write operation is entered.

The page write operation is shown in figure 17.4.

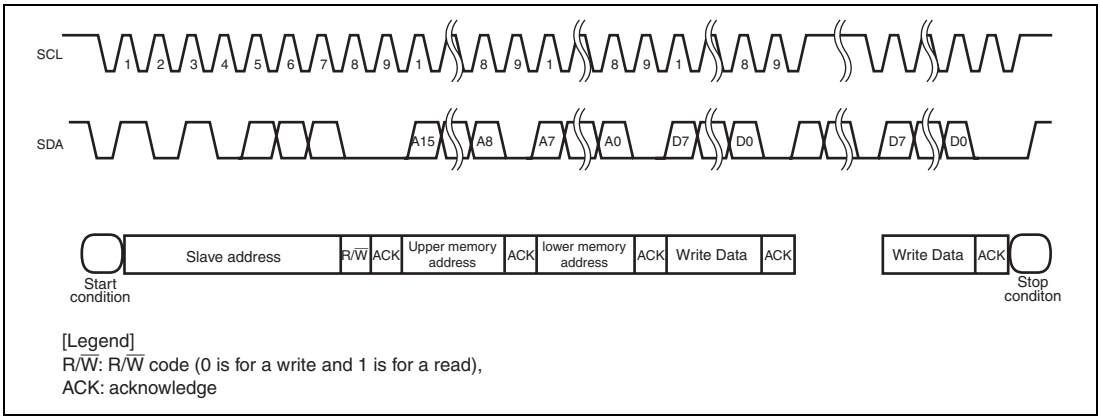


Figure 17.4 Page Write Operation

17.4.8 Acknowledge Polling

Acknowledge polling feature is used to show if the EEPROM is in an internally-timed write cycle or not. This feature is initiated by the input of the 8-bit slave address + R/$\overline{W}$ code following the start condition during an internally-timed write cycle. Acknowledge polling will operate R/W code = "0". The ninth acknowledgement judges if the EEPROM is an internally-timed write cycle or not. Acknowledgement "1" shows the EEPROM is in a internally-timed write cycle and acknowledgement "0" shows the internally-timed write cycle has been completed. The acknowledge polling starts to function after a write data is input, i.e., when the stop condition is input.

17.4.9 Read Operation

There are three read operations; current address read, random address read, and sequential read. Read operations are initiated in the same way as write operations with the exception of R/W = 1.

1. Current Address Read

   The internal address counter maintains the (n+1) address that is made by the last address (n) accessed during the last read or write operation, with incremented by one. Current address read accesses the (n+1) address kept by the internal address counter.

   After receiving in the order of a start condition and the slave address + R/$\overline{W}$ code (R/W = 1), the EEPROM outputs the 1-byte data of the (n+1) address from the most significant bit following acknowledgement "0". If the EEPROM receives in the order of acknowledgement "1" and a following stop condition, the EEPROM stops the read operation and is turned to a standby state.

   In case the EEPROM has accessed the last address H'01FF at previous read operation, the current address will roll over and returns to zero address. In case the EEPROM has accessed the last address of the page at previous write operation, the current address will roll over within page addressing and returns to the first address in the same page.

   The current address is valid while power is on. The current address after power on will be undefined. After power is turned on, define the address by the random address read operation described below is necessary.

   The current address read operation is shown in figure 17.5.

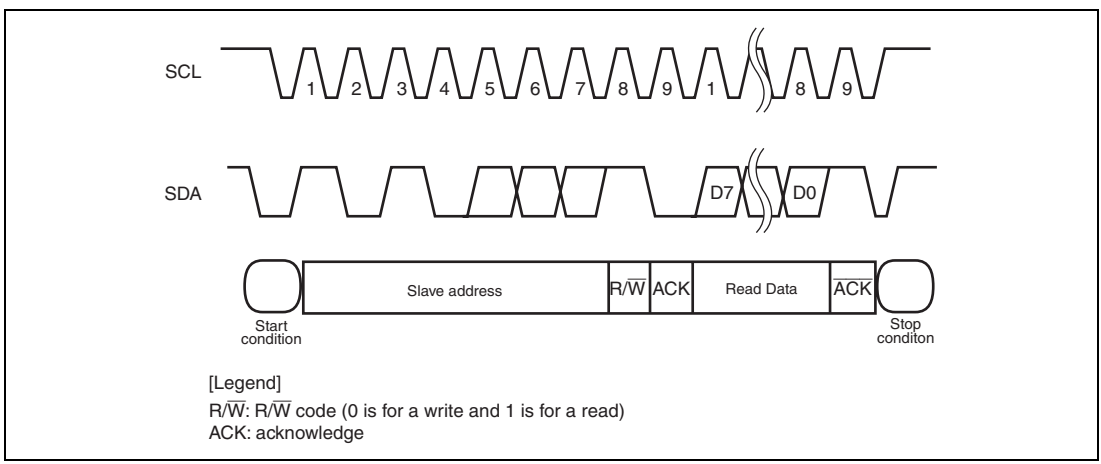


Figure 17.5 Current Address Read Operation

2. Random Address Read

This is a read operation with defined read address. A random address read requires a dummy write to set read address. The EEPROM receives a start condition, slave address + R/$\overline{W}$ code (R/$\overline{W}$ = 0), memory address (upper) and memory address (lower) sequentially. The EEPROM outputs acknowledgement "0" after receiving memory address (lower) then enters a current address read with receiving a start condition again. The EEPROM outputs the read data of the address which was defined in the dummy write operation. After receiving acknowledgement "1" and a following stop condition, the EEPROM stops the random read operation and returns to a standby state.

The random address read operation is shown in figure 17.6.

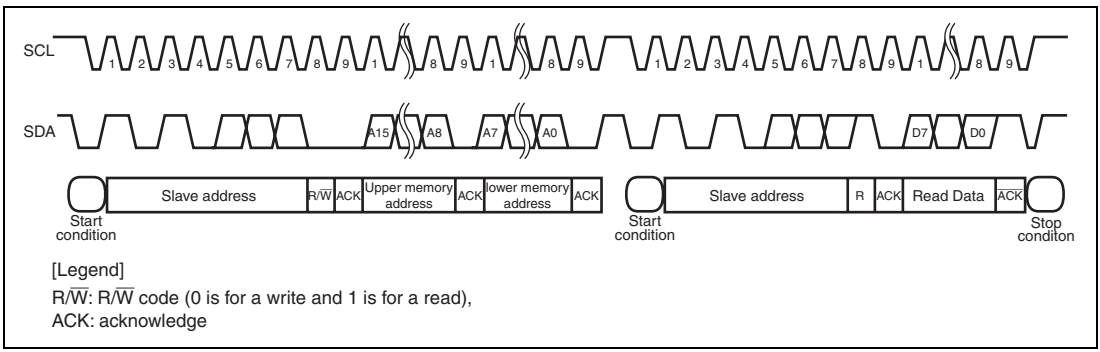


Figure 17.6 Random Address Read Operation

3. Sequential Read

This is a mode to read the data sequentially. Data is sequential read by either a current address read or a random address read. If the EEPROM receives acknowledgement "0" after 1-byte read data is output, the read address is incremented and the next 1-byte read data are coming out. Data is output sequentially by incrementing addresses as long as the EEPROM receives acknowledgement "0" after the data is output. The address will roll over and returns address zero if it reaches the last address H'01FF. The sequential read can be continued after roll over. The sequential read is terminated if the EEPROM receives acknowledgement "1" and a following stop condition as the same manner as in the random address read.

The condition of a sequential read when the current address read is used is shown in figure 17.7.

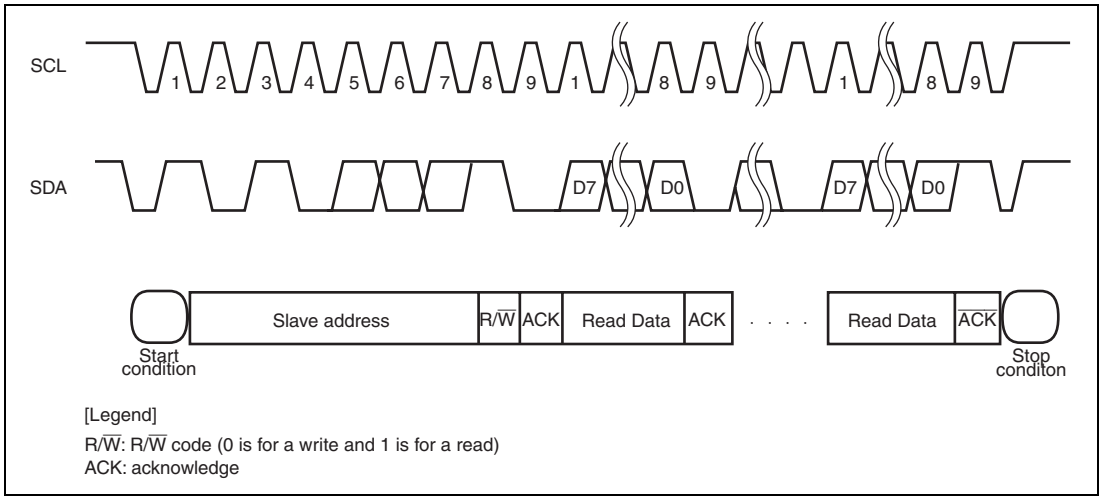


Figure 17.7 Sequential Read Operation (when current address read is used)

17.5 Usage Notes

17.5.1 Data Protection at V_{CC} On/Off

When V_{CC} is turned on or off, the data might be destroyed by malfunction. Be careful of the notices described below to prevent the data to be destroyed.

1. SCL and SDA should be fixed to V_{CC} or V_{SS} during V_{CC} on/off.
2. V_{CC} should be turned off after the EEPROM is placed in a standby state.
3. When V_{CC} is turned on from the intermediate level, malfunction is caused, so V_{CC} should be turned on from the ground level (V_{SS}).
4. V_{CC} turn on speed should be longer than 10 us.

17.5.2 Write/Erase Endurance

The endurance is 10^5 cycles/page (1% cumulative failure rate) in case of page programming and 10^4 cycles/byte in case of byte programming. The data retention time is more than 10 years when a device is page-programmed less than 10^4 cycles.

17.5.3 Noise Suppression Time

This EEPROM has a noise suppression function at SCL and SDA inputs, that cuts noise of width less than 50 ns. Be careful not to allow noise of width more than 50 ns because the noise of with more than 50 ms is recognized as an active pulse.

Section 18 Power Supply Circuit

This LSI incorporates an internal power supply step-down circuit. Use of this circuit enables the internal power supply to be fixed at a constant level of approximately 3.0 V, independently of the voltage of the power supply connected to the external V_{CC} pin. As a result, the current consumed when an external power supply is used at 3.0 V or above can be held down to virtually the same low level as when used at approximately 3.0 V. If the external power supply is 3.0 V or below, the internal voltage will be practically the same as the external voltage. It is, of course, also possible to use the same level of external power supply voltage and internal power supply voltage without using the internal power supply step-down circuit.

18.1 When Using Internal Power Supply Step-Down Circuit

Connect the external power supply to the V_{CC} pin, and connect a capacitance of approximately 0.1 µF between V_{CL} and V_{SS}, as shown in figure 18.1. The internal step-down circuit is made effective simply by adding this external circuit. In the external circuit interface, the external power supply voltage connected to V_{CC} and the GND potential connected to V_{SS} are the reference levels. For example, for port input/output levels, the V_{CC} level is the reference for the high level, and the V_{SS} level is that for the low level. The A/D converter analog power supply is not affected by the internal step-down circuit.

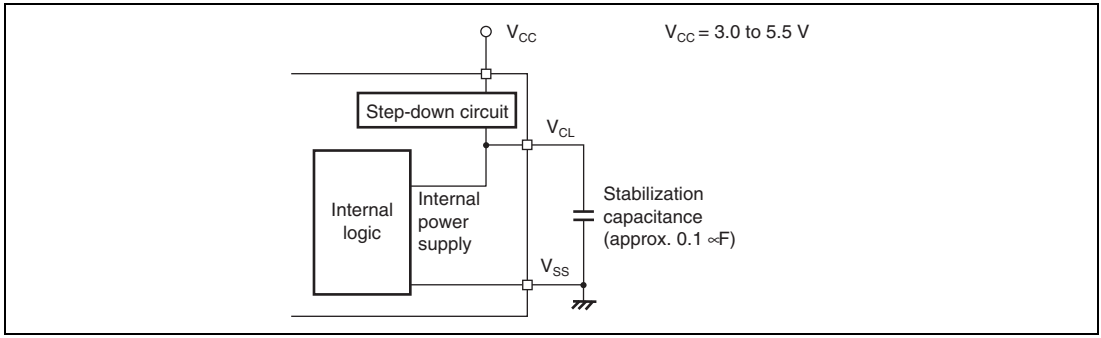


Figure 18.1 Power Supply Connection when Internal Step-Down Circuit is Used

18.2 When Not Using Internal Power Supply Step-Down Circuit

When the internal power supply step-down circuit is not used, connect the external power supply to the V_{CL} pin and V_{CC} pin, as shown in figure 18.2. The external power supply is then input directly to the internal power supply. The permissible range for the power supply voltage is 3.0 V to 3.6 V. Operation cannot be guaranteed if a voltage outside this range (less than 3.0 V or more than 3.6 V) is input.

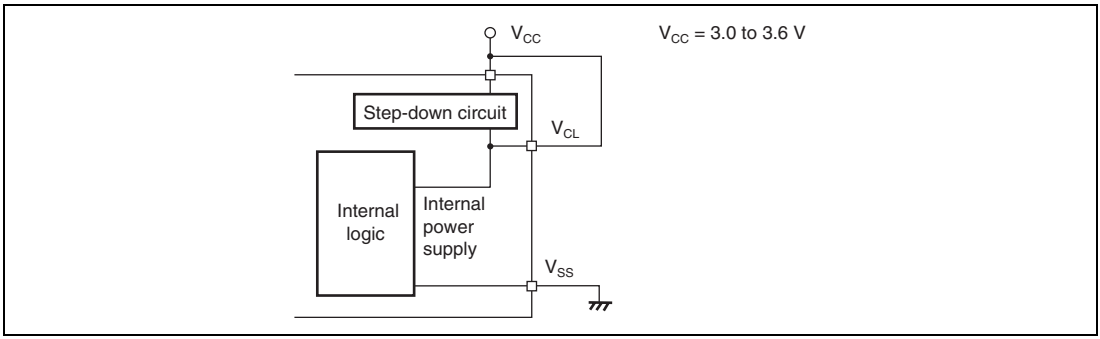


Figure 18.2 Power Supply Connection when Internal Step-Down Circuit is Not Used

Section 19 List of Registers

The register list gives information on the on-chip I/O register addresses, how the register bits are configured, and the register states in each operating mode. The information is given as shown below.

1. Register addresses (address order)
- Registers are listed from the lower allocation addresses.
- Registers are classified by functional modules.
- The data bus width is indicated.
- The number of access states is indicated.

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.
- When registers consist of 16 bits, bits are described from the MSB side.

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, refer to the section on that on-chip peripheral module.

19.1 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	Abbre-viation	Bit No.	Address	Module Name	Data Bus Width	Access State
Timer mode register W	TMRW	8	H'FF80	Timer W	8	2
Timer control register W	TCRW	8	H'FF81	Timer W	8	2
Timer interrupt enable register W	TIERW	8	H'FF82	Timer W	8	2
Timer status register W	TSRW	8	H'FF83	Timer W	8	2
Timer I/O control register 0	TIOR0	8	H'FF84	Timer W	8	2
Timer I/O control register 1	TIOR1	8	H'FF85	Timer W	8	2
Timer counter	TCNT	16	H'FF86	Timer W	16*[1]	2
General register A	GRA	16	H'FF88	Timer W	16*[1]	2
General register B	GRB	16	H'FF8A	Timer W	16*[1]	2
General register C	GRC	16	H'FF8C	Timer W	16*[1]	2
General register D	GRD	16	H'FF8E	Timer W	16*[1]	2
Flash memory control register 1	FLMCR1	8	H'FF90	ROM	8	2
Flash memory control register 2	FLMCR2	8	H'FF91	ROM	8	2
Flash memory power control register	FLPWCR	8	H'FF92	ROM	8	2
Erase block register 1	EBR1	8	H'FF93	ROM	8	2
Flash memory enable register	FENR	8	H'FF9B	ROM	8	2
Timer control register V0	TCRV0	8	H'FFA0	Timer V	8	3
Timer control/status register V	TCSRV	8	H'FFA1	Timer V	8	3
Timer constant register A	TCORA	8	H'FFA2	Timer V	8	3
Timer constant register B	TCORB	8	H'FFA3	Timer V	8	3
Timer counter V	TCNTV	8	H'FFA4	Timer V	8	3
Timer control register V1	TCRV1	8	H'FFA5	Timer V	8	3
Timer mode register A	TMA	8	H'FFA6	Timer A	8	2
Timer counter A	TCA	8	H'FFA7	Timer A	8	2
Serial mode register	SMR	8	H'FFA8	SCI3	8	3
Bit rate register	BRR	8	H'FFA9	SCI3	8	3
Serial control register 3	SCR3	8	H'FFAA	SCI3	8	3

Register Name	Abbre-viation	Bit No.	Address	Module Name	Data Bus Width	Access State
Transmit data register	TDR	8	H'FFAB	SCI3	8	3
Serial status register	SSR	8	H'FFAC	SCI3	8	3
Receive data register	RDR	8	H'FFAD	SCI3	8	3
A/D data register A	ADDRA	16	H'FFB0	A/D converter	8	3
A/D data register B	ADDRB	16	H'FFB2	A/D converter	8	3
A/D data register C	ADDRC	16	H'FFB4	A/D converter	8	3
A/D data register D	ADDRD	16	H'FFB6	A/D converter	8	3
A/D control/status register	ADCSR	8	H'FFB8	A/D converter	8	3
A/D control register	ADCR	8	H'FFB9	A/D converter	8	3
Timer control/status register WD	TCSRWD	8	H'FFC0	WDT*[2]	8	2
Timer counter WD	TCWD	8	H'FFC1	WDT*[2]	8	2
Timer mode register WD	TMWD	8	H'FFC2	WDT*[2]	8	2
I^2C bus control register	ICCR	8	H'FFC4	IIC	8	2
I^2C bus status register	ICSR	8	H'FFC5	IIC	8	2
I^2C bus data register	ICDR	8	H'FFC6	IIC	8	2
Second slave address register	SARX	8	H'FFC6	IIC	8	2
I^2C bus mode register	ICMR	8	H'FFC7	IIC	8	2
Slave address register	SAR	8	H'FFC7	IIC	8	2
Address break control register	ABRKCR	8	H'FFC8	Address break	8	2
Address break status register	ABRKSR	8	H'FFC9	Address break	8	2
Break address register H	BARH	8	H'FFCA	Address break	8	2
Break address register L	BARL	8	H'FFCB	Address break	8	2
Break data register H	BDRH	8	H'FFCC	Address break	8	2
Break data register L	BDRL	8	H'FFCD	Address break	8	2
Port pull-up control register 1	PUCR1	8	H'FFD0	I/O port	8	2
Port pull-up control register 5	PUCR5	8	H'FFD1	I/O port	8	2
Port data register 1	PDR1	8	H'FFD4	I/O port	8	2
Port data register 2	PDR2	8	H'FFD5	I/O port	8	2
Port data register 5	PDR5	8	H'FFD8	I/O port	8	2
Port data register 7	PDR7	8	H'FFDA	I/O port	8	2
Port data register 8	PDR8	8	H'FFDB	I/O port	8	2

Register Name	Abbre-viation	Bit No.	Address	Module Name	Data Bus Width	Access State
Port data register B	PDRB	8	H'FFDD	I/O port	8	2
Port mode register 1	PMR1	8	H'FFE0	I/O port	8	2
Port mode register 5	PMR5	8	H'FFE1	I/O port	8	2
Port control register 1	PCR1	8	H'FFE4	I/O port	8	2
Port control register 2	PCR2	8	H'FFE5	I/O port	8	2
Port control register 5	PCR5	8*[3]	H'FFE8	I/O port	8	2
Port control register 7	PCR7	8	H'FFEA	I/O port	8	2
Port control register 8	PCR8	8	H'FFEB	I/O port	8	2
System control register 1	SYSCR1	8	H'FFF0	Power-down	8	2
System control register 2	SYSCR2	8	H'FFF1	Power-down	8	2
Interrupt edge select register 1	IEGR1	8	H'FFF2	Interrupts	8	2
Interrupt edge select register 2	IEGR2	8	H'FFF3	Interrupts	8	2
Interrupt enable register 1	IENR1	8	H'FFF4	Interrupts	8	2
Interrupt flag register 1	IRR1	8	H'FFF6	Interrupts	8	2
Wake-up interrupt flag register	IWPR	8	H'FFF8	Interrupts	8	2
Module standby control register 1	MSTCR1	8	H'FFF9	Power-down	8	2
Timer serial control register	TSCR	8	H'FFFC	IIC	8	2

Notes: 1. Only word access can be used.
2. WDT: Watchdog timer.
3. The number of bits is six for H8/3664N.

- EEPROM

Register Name	Abbre-viation	Bit No.	Address	Module Name	Data Bus Width	Access State
EEPROM key register	EKR	8	H'FF10	IEEPROM	8	2

19.2 Register Bits

Register 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 Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name
TMRW	CTS	—	BUFEB	BUFEA	—	PWMD	PWMC	PWMB	Timer W
TCRW	CCLR	CKS2	CKS1	CKS0	TOD	TOC	TOB	TOA	
TIERW	OVIE	—	—	—	IMIED	IMIEC	IMIEB	IMIEA	
TSRW	OVF	—	—	—	IMFD	IMFC	IMFB	IMFA	
TIOR0	—	IOB2	IOB1	IOB0	—	IOA2	IOA1	IOA0	
TIOR1	—	IOD2	IOD1	IOD0	—	IOC2	IOC1	IOC0	
TCNT	TCNT15	TCNT14	TCNT13	TCNT12	TCNT11	TCNT10	TCNT9	TCNT8	
	TCNT7	TCNT6	TCNT5	TCNT4	TCNT3	TCNT2	TCNT1	TCNT0	
GRA	GRA15	GRA14	GRA13	GRA12	GRA11	GRA10	GRA9	GRA8	
	GRA7	GRA6	GRA5	GRA4	GRA3	GRA2	GRA1	GRA0	
GRB	GRB15	GRB14	GRB13	GRB12	GRB11	GRB10	GRB9	GRB8	
	GRB7	GRB6	GRB5	GRB4	GRB3	GRB2	GRB1	GRB0	
GRC	GRC15	GRC14	GRC13	GRC12	GRC11	GRC10	GRC9	GRC8	
	GRC7	GRC6	GRC5	GRC4	GRC3	GRC2	GRC1	GRC0	
GRD	GRD15	GRD14	GRD13	GRD12	GRD11	GRD10	GRD9	GRD8	
	GRD7	GRD6	GRD5	GRD4	GRD3	GRD2	GRD1	GRD0	
FLMCR1	—	SWE	ESU	PSU	EV	PV	E	P	ROM
FLMCR2	FLER	—	—	—	—	—	—	—	
FLPWCR	PDWND	—	—	—	—	—	—	—	
EBR1	—	—	—	EB4	EB3	EB2	EB1	EB0	
FENR	FLSHE	—	—	—	—	—	—	—	
TCRV0	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	Timer V
TCSRV	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0	
TCORA	TCORA7	TCORA6	TCORA5	TCORA4	TCORA3	TCORA2	TCORA1	TCORA0	
TCORB	TCORB7	TCORB6	TCORB5	TCORB4	TCORB3	TCORB2	TCORB1	TCORB0	
TCNTV	TCNTV7	TCNTV6	TCNTV5	TCNTV4	TCNTV3	TCNTV2	TCNTV1	TCNTV0	
TCRV1	—	—	—	TVEG1	TVEG0	TRGE	—	ICKS0	
TMA	TMA7	TMA6	TMA5	—	TMA3	TMA2	TMA1	TMA0	Timer A
TCA	TCA7	TCA6	TCA5	TCA4	TCA3	TCA2	TCA1	TCA0	

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name
SMR	COM	CHR	PE	PM	STOP	MP	CKS1	CKS0	SCI3
BRR	BRR7	BRR6	BRR5	BRR4	BRR3	BRR2	BRR1	BRR0	
SCR3	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1	TDR0	
SSR	TDRE	RDRF	OER	FER	PER	TEND	MPBR	MPBT	
RDR	RDR7	RDR6	RDR5	RDR4	RDR3	RDR2	RDR1	RDR0	
ADDRA	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D converter
	AD1	AD0	—	—	—	—	—	—	
ADDRB	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDRC	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDRD	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADCSR	ADF	ADIE	ADST	SCAN	CKS	CH2	CH1	CH0	
ADCR	TRGE	—	—	—	—	—	—	—	
TCSRWD	B6WI	TCWE	B4WI	TCSRWE	B2WI	WDON	B0WI	WRST	WDT*[1]
TCWD	TCWD7	TCWD6	TCWD5	TCWD4	TCWD3	TCWD2	TCWD1	TCWD0	
TMWD	—	—	—	—	CKS3	CKS2	CKS1	CKS0	
ICCR	ICE	IEIC	MST	TRS	ACKE	BBSY	IRIC	SCP	IIC
ICSR	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB	
ICDR	ICDR7	ICDR6	ICDR5	ICDR4	ICDR3	ICDR2	ICDR1	ICDR0	
SARX	SVAX6	SVAX5	SVAX4	SVAX3	SVAX2	SVAX1	SVAX0	FSX	
ICMR	MLS	WAIT	CKS2	CKS1	CKS0	BC2	BC1	BC0	
SAR	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS	
ABRKCR	RTINTE	CSEL1	CSEL0	ACMP2	ACMP1	ACMP0	DCMP1	DCMP0	Address break
ABRKSR	ABIF	ABIE	—	—	—	—	—	—	
BARH	BARH7	BARH6	BARH5	BARH4	BARH3	BARH2	BARH1	BARH0	
BARL	BARL7	BARL6	BARL5	BARL4	BARL3	BARL2	BARL1	BARL0	
BDRH	BDRH7	BDRH6	BDRH5	BDRH4	BDRH3	BDRH2	BDRH1	BDRH0	
BDRL	BDRL7	BDRL6	BDRL5	BDRL4	BDRL3	BDRL2	BDRL1	BDRL0	
PUCR1	PUCR17	PUCR16	PUCR15	PUCR14	—	PUCR12	PUCR11	PUCR10	I/O port
PUCR5	—	—	PUCR55	PUCR54	PUCR53	PUCR52	PUCR51	PUCR50	
PDR1	P17	P16	P15	P14	—	P12	P11	P10	
PDR2	—	—	—	—	—	P22	P21	P20	
PDR5	P57*[2]	P56*[2]	P55	P54	P53	P52	P51	P50	

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name
PDR7	—	P76	P75	P74	—	—	—	—	I/O port
PDR8	P87	P86	P85	P84	P83	P82	P81	P80	
PDRB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	
PMR1	IRQ3	IRQ2	IRQ1	IRQ0	—	—	TXD	TMOW	
PMR5	—	—	WKP5	WKP4	WKP3	WKP2	WKP1	WKP0	
PCR1	PCR17	PCR16	PCR15	PCR14	—	PCR12	PCR11	PCR10	
PCR2	—	—	—	—	—	PCR22	PCR21	PCR20	
PCR5	PCR57*[2]	PCR56*[2]	PCR55	PCR54	PCR53	PCR52	PCR51	PCR50	
PCR7	—	PCR76	PCR75	PCR74	—	—	—	—	
PCR8	PCR87	PCR86	PCR85	PCR84	PCR83	PCR82	PCR81	PCR80	
SYSCR1	SSBY	STS2	STS1	STS0	NESEL	—	—	—	Power-down
SYSCR2	SMSEL	LSON	DTON	MA2	MA1	MA0	SA1	SA0	
IEGR1	NMIEG	—	—	—	IEG3	IEG2	IEG1	IEG0	Interrupts
IEGR2	—	—	WPEG5	WPEG4	WPEG3	WPEG2	WPEG1	WPEG0	
IENR1	IENDT	IENTA	IENWP	—	IEN3	IEN2	IEN1	IEN0	
IRR1	IRRDT	IRRTA	—	—	IRRI3	IRRI2	IRRI1	IRRI0	
IWPR	—	—	IWPF5	IWPF4	IWPF3	IWPF2	IWPF1	IWPF0	
MSTCR1	—	MSTIIC	MSTS3	MSTAD	MSTWD	MSTTW	MSTTV	MSTTA	Power-down
TSCR	—	—	—	—	—	—	IICRST	IICX	IIC

Notes: 1. WDT: Watchdog timer
2. This bit is not included in H8/3664N.

- EEPROM

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name
EKR	EKR7	EKR6	EKR5	EKR4	EKR3	EKR2	EKR1	EKR0	EEPROM

19.3 Register States in Each Operating Mode

Register Name	Reset	Active	Sleep	Subactive	Subsleep	Standby	Module
TMRW	Initialized	—	—	—	—	—	Timer W
TCRW	Initialized	—	—	—	—	—	
TIERW	Initialized	—	—	—	—	—	
TSRW	Initialized	—	—	—	—	—	
TIOR0	Initialized	—	—	—	—	—	
TIOR1	Initialized	—	—	—	—	—	
TCNT	Initialized	—	—	—	—	—	
GRA	Initialized	—	—	—	—	—	
GRB	Initialized	—	—	—	—	—	
GRC	Initialized	—	—	—	—	—	
GRD	Initialized	—	—	—	—	—	
FLMCR1	Initialized	—	—	Initialized	Initialized	Initialized	ROM
FLMCR2	Initialized	—	—	—	—	—	
FLPWCR	Initialized	—	—	—	—	—	
EBR1	Initialized	—	—	Initialized	Initialized	Initialized	
FENR	Initialized	—	—	—	—	—	
TCRV0	Initialized	—	—	Initialized	Initialized	Initialized	Timer V
TCSRV	Initialized	—	—	Initialized	Initialized	Initialized	
TCORA	Initialized	—	—	Initialized	Initialized	Initialized	
TCORB	Initialized	—	—	Initialized	Initialized	Initialized	
TCNTV	Initialized	—	—	Initialized	Initialized	Initialized	
TCRV1	Initialized	—	—	Initialized	Initialized	Initialized	
TMA	Initialized	—	—	—	—	—	Timer A
TCA	Initialized	—	—	—	—	—	
SMR	Initialized	—	—	Initialized	Initialized	Initialized	SCI3
BRR	Initialized	—	—	Initialized	Initialized	Initialized	
SCR3	Initialized	—	—	Initialized	Initialized	Initialized	
TDR	Initialized	—	—	Initialized	Initialized	Initialized	
SSR	Initialized	—	—	Initialized	Initialized	Initialized	
RDR	Initialized	—	—	Initialized	Initialized	Initialized	

Register Name	Reset	Active	Sleep	Subactive	Subsleep	Standby	Module
ADDRA	Initialized	—	—	Initialized	Initialized	Initialized	A/D converter
ADDRB	Initialized	—	—	Initialized	Initialized	Initialized	
ADDRC	Initialized	—	—	Initialized	Initialized	Initialized	
ADDRD	Initialized	—	—	Initialized	Initialized	Initialized	
ADCSR	Initialized	—	—	Initialized	Initialized	Initialized	
ADCR	Initialized	—	—	Initialized	Initialized	Initialized	
TCSRWD	Initialized	—	—	—	—	—	WDT*
TCWD	Initialized	—	—	—	—	—	
TMWD	Initialized	—	—	—	—	—	
ICCR	Initialized	—	—	—	—	—	IIC
ICSR	Initialized	—	—	—	—	—	
ICDR	Initialized	—	—	—	—	—	
SARX	Initialized	—	—	—	—	—	
ICMR	Initialized	—	—	—	—	—	
SAR	Initialized	—	—	—	—	—	
ABRKCR	Initialized	—	—	—	—	—	Address Break
ABRKSR	Initialized	—	—	—	—	—	
BARH	Initialized	—	—	—	—	—	
BARL	Initialized	—	—	—	—	—	
BDRH	Initialized	—	—	—	—	—	
BDRL	Initialized	—	—	—	—	—	
PUCR1	Initialized	—	—	—	—	—	I/O port
PUCR5	Initialized	—	—	—	—	—	
PDR1	Initialized	—	—	—	—	—	
PDR2	Initialized	—	—	—	—	—	
PDR5	Initialized	—	—	—	—	—	
PDR7	Initialized	—	—	—	—	—	
PDR8	Initialized	—	—	—	—	—	
PDRB	Initialized	—	—	—	—	—	
PMR1	Initialized	—	—	—	—	—	
PMR5	Initialized	—	—	—	—	—	

Register Name	Reset	Active	Sleep	Subactive	Subsleep	Standby	Module
PCR1	Initialized	—	—	—	—	—	I/O port
PCR2	Initialized	—	—	—	—	—	
PCR5	Initialized	—	—	—	—	—	
PCR7	Initialized	—	—	—	—	—	
PCR8	Initialized	—	—	—	—	—	
SYSCR1	Initialized	—	—	—	—	—	Power-down
SYSCR2	Initialized	—	—	—	—	—	Power-down
IEGR1	Initialized	—	—	—	—	—	Interrupts
IEGR2	Initialized	—	—	—	—	—	Interrupts
IENR1	Initialized	—	—	—	—	—	Interrupts
IRR1	Initialized	—	—	—	—	—	Interrupts
IWPR	Initialized	—	—	—	—	—	Interrupts
MSTCR1	Initialized	—	—	—	—	—	Power-down
TSCR	Initialized	—	—	—	—	—	IIC

Note : — is not initialized

* WDT: Watchdog timer

- EEPROM

Register Name	Reset	Active	Sleep	Subactive	Subsleep	Standby	Module
EKR	Initialized	—	—	—	—	—	EEPROM

Section 20 Electrical Characteristics

20.1 Absolute Maximum Ratings

Table 20.1 Absolute Maximum Ratings

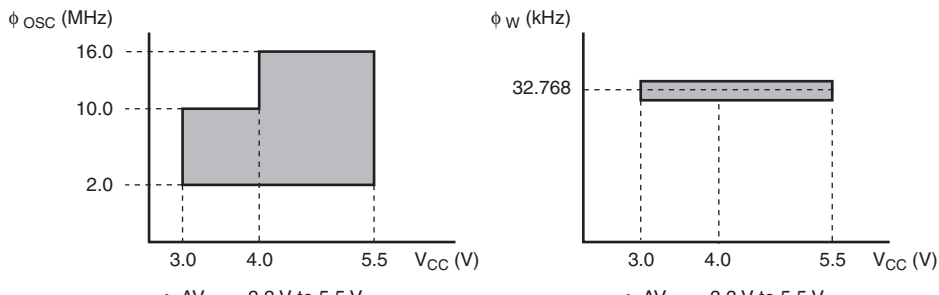
Item		Symbol	Value	Unit	Note
Power supply voltage		V_{CC}	–0.3 to +7.0	V	*
Analog power supply voltage		AV_{CC}	–0.3 to +7.0	V	
Input voltage	Ports other than Port B and X1	V_{IN}	–0.3 to V_{CC} +0.3	V	
	Port B		–0.3 to AV_{CC} +0.3	V	
	X1		–0.3 to 4.3	V	
Operating temperature		T_{opr}	–20 to +75	°C	
Storage temperature		T_{stg}	–55 to +125	°C	

Note: * Permanent damage may result if maximum ratings are exceeded. Normal operation should be under the conditions specified in Electrical Characteristics. Exceeding these values can result in incorrect operation and reduced reliability.

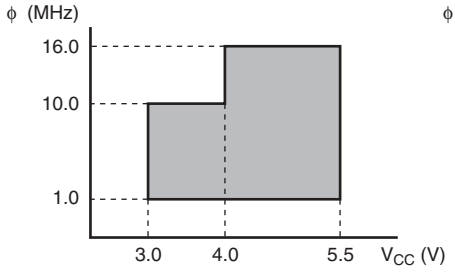
20.2 Electrical Characteristics (F-ZTAT™ Version, F-ZTAT™ Version with EEPROM)

20.2.1 Power Supply Voltage and Operating Ranges

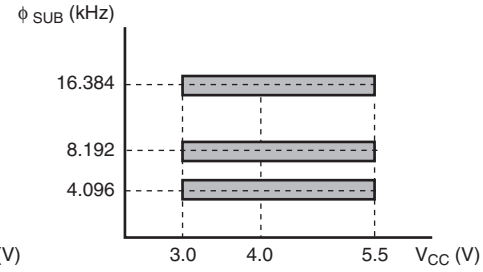
Power Supply Voltage and Oscillation Frequency Range



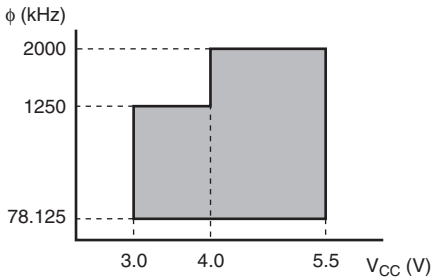
Power Supply Voltage and Operating Frequency Range



- AV_{CC} = 3.3 V to 5.5 V
- Active mode
- Sleep mode
  (When MA2 = 0 in SYSCR2)

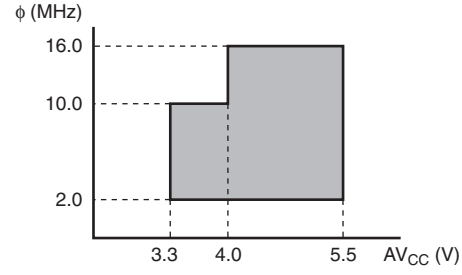


- AV_{CC} = 3.3 V to 5.5 V
- Subactive mode
- Subsleep mode



- AV_{CC} = 3.3 V to 5.5 V
- Active mode
- Sleep mode
  (When MA2 = 1 in SYSCR2)

Analog Power Supply Voltage and A/D Converter Accuracy Guarantee Range



- V_{CC} = 3.0 V to 5.5 V
- Active mode
- Sleep mode

20.2.2 DC Characteristics

Table 20.2 DC Characteristics (1)

V_{CC} = 3.0 V to 5.5 V, V_{SS} = 0.0 V, T_a = –20°C to +75°C unless otherwise indicated.

				Values				
Item	**Symbol**	**Applicable Pins**	**Test Condition**	**Min**	**Typ**	**Max**	**Unit**	**Notes**
Input high voltage	V_{IH}	$\overline{RES}$, $\overline{NMI}$, $\overline{WKP0}$ to $\overline{WKP5}$, $\overline{IRQ0}$ to $\overline{IRQ3}$, $\overline{ADTRG}$, TMRIV,	V_{CC} = 4.0 V to 5.5 V	$V_{CC}\times 0.8$	—	$V_{CC}+0.3$	V	
		TMCIV, FTCI, FTIOA to FTIOD, SCK3, TRGV		$V_{CC}\times 0.9$	—	$V_{CC}+0.3$		
		RXD, SCL, SDA, P10 to P12, P14 to P17, P20 to P22,	V_{CC} = 4.0 V to 5.5 V	$V_{CC}\times 0.7$	—	$V_{CC}+0.3$	V	
		P50 to P57*, P74 to P76, P80 to P87		$V_{CC}\times 0.8$	—	$V_{CC}+0.3$		
		PB0 to PB7	V_{CC} = 4.0 V to 5.5 V	$V_{CC}\times 0.7$	—	$AV_{CC}+0.3$	V	
				$V_{CC}\times 0.8$	—	$AV_{CC}+0.3$		
		OSC1	V_{CC} = 4.0 V to 5.5 V	$V_{CC}-0.5$	—	$V_{CC}+0.3$	V	
				$V_{CC}-0.3$	—	$V_{CC}+0.3$		
Input low voltage	V_{IL}	$\overline{RES}$, $\overline{NMI}$, $\overline{WKP0}$ to $\overline{WKP5}$, $\overline{IRQ0}$ to $\overline{IRQ3}$, $\overline{ADTRG}$, TMRIV,	V_{CC} = 4.0 V to 5.5 V	–0.3	—	$V_{CC}\times 0.2$	V	
		TMCIV, FTCI, FTIOA to FTIOD, SCK3, TRGV		–0.3	—	$V_{CC}\times 0.1$		
		RXD, SCL, SDA, P10 to P12, P14 to P17, P20 to P22,	V_{CC} = 4.0 V to 5.5 V	–0.3	—	$V_{CC}\times 0.3$	V	
		P50 to P57*, P74 to P76, P80 to P87, PB0 to PB7		–0.3	—	$V_{CC}\times 0.2$		
		OSC1	V_{CC} = 4.0 V to 5.5 V	–0.3	—	0.5	V	
				–0.3	—	0.3		

Note: * P50 to P55 for H8/3664N

Item	Symbol	Applicable Pins	Test Condition	Values Min	Typ	Max	Unit	Notes
Output high voltage	V_{OH}	P10 to P12, P14 to P17, P20 to P22, P50 to P55, P74 to P76, P80 to P87,	V_{CC} = 4.0 V to 5.5 V $-I_{OH}$ = 1.5 mA	V_{CC} – 1.0	—	—	V	
			$-I_{OH}$ = 0.1 mA	V_{CC} – 0.5	—	—		
		P56, P57*	V_{CC} = 4.0 V to 5.5 V $-I_{OH}$ = 0.1 mA	V_{CC} – 2.5	—	—	V	
			V_{CC} = 3.0 V to 4.0 V $-I_{OH}$ = 0.1 mA	V_{CC} – 2.0	—	—		
Output low voltage	V_{OL}	P10 to P12, P14 to P17, P20 to P22, P50 to P57*, P74 to P76,	V_{CC} = 4.0 V to 5.5 V I_{OL} = 1.6 mA	—	—	0.6	V	
			I_{OL} = 0.4 mA	—	—	0.4		
		P80 to P87	V_{CC} = 4.0 V to 5.5 V I_{OL} = 20.0 mA	—	—	1.5	V	
			V_{CC} = 4.0 V to 5.5 V I_{OL} = 10.0 mA	—	—	1.0		
			V_{CC} = 4.0 V to 5.5 V I_{OL} = 1.6 mA	—	—	0.4		
			I_{OL} = 0.4 mA	—	—	0.4		
		SCL, SDA	V_{CC} = 4.0 V to 5.5 V I_{OL} = 6.0 mA	—	—	0.6	V	
			I_{OL} = 3.0 mA	—	—	0.4		

Note: * P50 to P55 for H8/3664N

Item	Symbol	Applicable Pins	Test Condition	Values Min	Typ	Max	Unit	Notes
Input/ output leakage current	$\lvert I_{IL} \rvert$	OSC1, $\overline{RES}$, $\overline{NMI}$, $\overline{WKP0}$ to $\overline{WKP5}$, $\overline{IRQ0}$ to $\overline{IRQ3}$, $\overline{ADTRG}$, TRGV, TMRIV, TMCIV, FTCI, FTIOA to FTIOD, RXD, SCK3, SCL, SDA	V_{IN} = 0.5 V to (V_{CC} – 0.5 V)	—	—	1.0	µA	
		P10 to P12, P14 to P17, P20 to P22, P50 to P57*[1], P74 to P76, P80 to P87,	V_{IN} = 0.5 V to (V_{CC} – 0.5 V)	—	—	1.0	µA	
		PB0 to PB7	V_{IN} = 0.5 V to (AV_{CC} – 0.5 V)	—	—	1.0	µA	
Pull-up MOS current	$-I_p$	P10 to P12, P14 to P17, P50 to P55	V_{CC} = 5.0 V, V_{IN} = 0.0 V	50.0	—	300.0	µA	
			V_{CC} = 3.0 V, V_{IN} = 0.0 V	—	60.0	—		Reference value
Input capacitance	C_{in}	All input pins except power supply pins	f = 1 MHz, V_{IN} = 0.0 V, T_a = 25°C	—	—	15.0	pF	H8/3664N
		SCL, SDA		—	—	25.0		
Active mode supply current	I_{OPE1}	V_{CC}	Active mode 1 V_{CC} = 5.0 V, f_{OSC} = 16 MHz	—	15.0	22.5	mA	*[2]
			Active mode 1 V_{CC} = 3.0 V, f_{OSC} = 10 MHz	—	8.0	—		*[2] Reference value
	I_{OPE2}	V_{CC}	Active mode 2 V_{CC} = 5.0 V, f_{OSC} = 16 MHz	—	1.8	2.7	mA	*[2]
			Active mode 2 V_{CC} = 3.0 V, f_{OSC} = 10 MHz	—	1.2	—		*[2] Reference value

Notes: 1. P50 to P55 for H8/3664N

2. Pin states during supply current measurement are given below (excluding current in the pull-up MOS transistors and output buffers).

Item	Symbol	Applicable Pins	Test Condition	Values Min	Typ	Max	Unit	Notes
Sleep mode supply current	I_{SLEEP1}	V_{CC}	Sleep mode 1 V_{CC} = 5.0 V, f_{OSC} = 16 MHz	—	11.5	17.0	mA	*
			Sleep mode 1 V_{CC} = 3.0 V, f_{OSC} = 10 MHz	—	6.5	—		* Reference value
	I_{SLEEP2}	V_{CC}	Sleep mode 2 V_{CC} = 5.0 V, f_{OSC} = 16 MHz	—	1.7	2.5	mA	*
			Sleep mode 2 V_{CC} = 3.0 V, f_{OSC} = 10 MHz	—	1.1	—		* Reference value
Subactive mode supply current	I_{SUB}	V_{CC}	V_{CC} = 3.0 V 32-kHz crystal resonator ($\phi_{SUB} = \phi_W/2$)	—	35.0	70.0	µA	*
			V_{CC} = 3.0 V 32-kHz crystal resonator ($\phi_{SUB} = \phi_W/8$)	—	25.0	—		* Reference value
Subsleep mode supply current	I_{SUBSP}	V_{CC}	V_{CC} = 3.0 V 32-kHz crystal resonator ($\phi_{SUB} = \phi_W/2$)	—	25.0	50.0	µA	*
Standby mode supply current	I_{STBY}	V_{CC}	32-kHz crystal resonator not used	—	—	5.0	µA	*
RAM data retaining voltage	V_{RAM}	V_{CC}		2.0	—	—	V	

Note: * Pin states during supply current measurement are given below (excluding current in the pull-up MOS transistors and output buffers).

Mode	$\overline{RES}$ Pin	Internal State	Other Pins	Oscillator Pins
Active mode 1	V_{CC}	Operates	V_{CC}	Main clock: ceramic or crystal resonator Subclock: Pin X1 = V_{SS}
Active mode 2		Operates (ϕosc/64)		
Sleep mode 1	V_{CC}	Only timers operate	V_{CC}	
Sleep mode 2		Only timers operate (ϕosc/64)		
Subactive mode	V_{CC}	Operates	V_{CC}	Main clock: ceramic or crystal resonator Subclock: crystal resonator
Subsleep mode	V_{CC}	Only timers operate	V_{CC}	
Standby mode	V_{CC}	CPU and timers both stop	V_{CC}	Main clock: ceramic or crystal resonator Subclock: Pin X1 = V_{SS}

Table 20.2 DC Characteristics (2)

V_{CC} = 3.0 V to 5.5 V, V_{SS} = 0.0 V, T_a = –20°C to +75°C, unless otherwise indicated.

				Values				
Item	Symbol	Applicable Pins	Test Condition	Min	Typ	Max	Unit	Notes
EEPROM supply current	I_{EEW}	V_{CC}	V_{CC} = 5.0 V, t_{SCL} = 2.5 µs (when writing)	—	—	2.0	mA	*
	I_{EER}	V_{CC}	V_{CC} = 5.0 V, t_{SCL} = 2.5 µs (when reading)	—	—	0.3	mA	
	I_{EESTBY}	V_{CC}	V_{CC} = 5.0 V, t_{SCL} = 2.5 µs (at standby)	—	—	3.0	µA	

Note: * The supply current of the EEPROM chip is shown.
For the supply current of H8/3664N, add the above current values to the supply current of H8/3664F.

Table 20.2 DC Characteristics (3)

V_{CC} = 3.0 V to 5.5 V, V_{SS} = 0.0 V, T_a = –20°C to +75°C, unless otherwise indicated.

Item	Symbol	Applicable Pins	Test Condition	Values Min	Typ	Max	Unit
Allowable output low current (per pin)	I_{OL}	Output pins except port 8, SCL, and SDA	V_{CC} = 4.0 V to 5.5 V	—	—	2.0	mA
		Port 8		—	—	20.0	mA
		Port 8		—	—	10.0	mA
		SCL and SDA		—	—	6.0	mA
		Output pins except port 8, SCL, and SDA		—	—	0.5	mA
Allowable output low current (total)	ΣI_{OL}	Output pins except port 8, SCL, and SDA	V_{CC} = 4.0 V to 5.5 V	—	—	40.0	mA
		Port 8, SCL, and SDA		—	—	80.0	mA
		Output pins except port 8, SCL, and SDA		—	—	20.0	mA
		Port 8, SCL, and SDA		—	—	40.0	mA
Allowable output high current (per pin)	$\lvert -I_{OH} \rvert$	All output pins	V_{CC} = 4.0 V to 5.5 V	—	—	2.0	mA
				—	—	0.2	mA
Allowable output high current (total)	$\lvert -\Sigma I_{OH} \rvert$	All output pins	V_{CC} = 4.0 V to 5.5 V	—	—	30.0	mA
				—	—	8.0	mA

20.2.3 AC Characteristics

Table 20.3 AC Characteristics

V_{CC} = 3.0 V to 5.5 V, V_{SS} = 0.0 V, T_a = –20°C to +75°C, unless otherwise specified.

Item	Symbol	Applicable Pins	Test Condition	Values			Unit	Reference Figure
				Min	Typ	Max		
System clock oscillation frequency	f_{OSC}	OSC1, OSC2	V_{CC} = 4.0 V to 5.5 V	2.0	—	16.0	MHz	*1
				2.0	—	10.0	MHz	
System clock (φ) cycle time	t_{cyc}			1	—	64	t_{OSC}	*2
				—	—	12.8	µs	
Subclock oscillation frequency	f_W	X1, X2		—	32.768	—	kHz	
Watch clock (ϕ_W) cycle time	t_W	X1, X2		—	30.5	—	µs	
Subclock (ϕ_{SUB}) cycle time	t_{subcyc}			2	—	8	t_W	*2
Instruction cycle time				2	—	—	t_{cyc} t_{subcyc}	
Oscillation stabilization time (crystal resonator)	t_{rc}	OSC1, OSC2		—	—	10.0	ms	
Oscillation stabilization time (ceramic resonator)	t_{rc}	OSC1, OSC2		—	—	5.0	ms	
Oscillation stabilization time	t_{rcx}	X1, X2		—	—	2.0	s	
External clock high width	t_{CPH}	OSC1	V_{CC} = 4.0 V to 5.5 V	25.0	—	—	ns	Figure 20.1
				40.0	—	—		
External clock low width	t_{CPL}	OSC1	V_{CC} = 4.0 V to 5.5 V	25.0	—	—	ns	
				40.0	—	—		
External clock rise time	t_{CPr}	OSC1	V_{CC} = 4.0 V to 5.5 V	—	—	10.0	ns	
				—	—	15.0		
External clock fall time	t_{CPf}	OSC1	V_{CC} = 4.0 V to 5.5 V	—	—	10.0	ns	
				—	—	15.0		

Item	Symbol	Applicable Pins	Test Condition	Values Min	Typ	Max	Unit	Reference Figure
$\overline{\text{RES}}$ pin low width	t_{REL}	$\overline{\text{RES}}$	At power-on and in modes other than those below	t_{rc}	—	—	ms	Figure 20.2
			In active mode and sleep mode operation	10	—	—	t_{cyc}	
Input pin high width	t_{IH}	$\overline{\text{NMI}}$, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ3}}$, $\overline{\text{WKP0}}$ to $\overline{\text{WKP5}}$, TMCIV, TMRIV, TRGV, $\overline{\text{ADTRG}}$, FTCI, FTIOA to FTIOD		2	—	—	t_{cyc} t_{subcyc}	Figure 20.3
Input pin low width	t_{IL}	$\overline{\text{NMI}}$, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ3}}$, $\overline{\text{WKP0}}$ to $\overline{\text{WKP5}}$, TMCIV, TMRIV, TRGV, $\overline{\text{ADTRG}}$, FTCI, FTIOA to FTIOD		2	—	—	t_{cyc} t_{subcyc}	

Notes: 1. When an external clock is input, the minimum system clock oscillator frequency is 1.0 MHz.

2. Determined by MA2, MA1, MA0, SA1, and SA0 of system control register 2 (SYSCR2).

Table 20.4 I²C Bus Interface Timing

V_{CC} = 3.0 V to 5.5 V, V_{SS} = 0.0 V, T_a = –20 to +75°C, unless otherwise specified.

Item	Symbol	Test Condition	Values Min	Typ	Max	Unit	Reference Figure
SCL input cycle time	t_{SCL}		$12t_{cyc}$ + 600	—	—	ns	Figure 20.4
SCL input high width	t_{SCLH}		$3t_{cyc}$ + 300	—	—	ns	
SCL input low width	t_{SCLL}		$5t_{cyc}$ + 300	—	—	ns	
Input fall time of SCL and SDA	t_{Sf}		—	—	300	ns	
SCL and SDA input spike pulse removal time	t_{SP}		—	—	$1t_{cyc}$	ns	
SDA input bus-free time	t_{BUF}		$5t_{cyc}$	—	—	ns	
Start condition input hold time	t_{STAH}		$3t_{cyc}$	—	—	ns	
Retransmission start condition input setup time	t_{STAS}		$3t_{cyc}$	—	—	ns	
Setup time for stop condition input	t_{STOS}		$3t_{cyc}$	—	—	ns	
Data-input setup time	t_{SDAS}		$1t_{cyc}$+20	—	—	ns	
Data-input hold time	t_{SDAH}		0	—	—	ns	
Capacitive load of SCL and SDA	c_b		0	—	400	pF	
SCL and SDA output fall time	t_{Sf}	V_{CC} = 4.0 V to 5.5 V	—	—	250	ns	
			—	—	300		

Table 20.5 Serial Interface (SCI3) Timing

V_{CC} = 3.0 V to 5.5 V, V_{SS} = 0.0 V, T_a = –20°C to +75°C, unless otherwise specified.

Item		Symbol	Applicable Pins	Test Condition	Values Min	Typ	Max	Unit	Reference Figure
Input clock cycle	Asynchronous	t_{Scyc}	SCK3		4	—	—	t_{cyc}	Figure 20.5
	Clocked synchronous				6	—	—	t_{cyc}	
Input clock pulse width		t_{SCKW}	SCK3		0.4	—	0.6	t_{Scyc}	
Transmit data delay time (clocked synchronous)		t_{TXD}	TXD	V_{CC} = 4.0 V to 5.5 V	—	—	1	t_{cyc}	Figure 20.6
					—	—	1	t_{cyc}	
Receive data setup time (clocked synchronous)		t_{RXS}	RXD	V_{CC} = 4.0 V to 5.5 V	62.5	—	—	ns	
					100.0	—	—	ns	
Receive data hold time (clocked synchronous)		t_{RXH}	RXD	V_{CC} = 4.0 V to 5.5 V	62.5	—	—	ns	
					100.0	—	—	ns	

20.2.4 A/D Converter Characteristics

Table 20.6 A/D Converter Characteristics

V_{CC} = 3.0 V to 5.5 V, V_{SS} = 0.0 V, T_a = –20°C to +75°C, unless otherwise specified.

Item	Symbol	Applicable Pins	Test Condition	Values Min	Typ	Max	Unit	Reference Figure
Analog power supply voltage	AV_{CC}	AV_{CC}		3.3	V_{CC}	5.5	V	*1
Analog input voltage	AV_{IN}	AN0 to AN7		V_{SS} – 0.3	—	AV_{CC} + 0.3	V	
Analog power supply current	AI_{OPE}	AV_{CC}	AV_{CC} = 5.0 V f_{OSC} = 16 MHz	—	—	2.0	mA	
	AI_{STOP1}	AV_{CC}		—	50	—	µA	*2 Reference value
	AI_{STOP2}	AV_{CC}		—	—	5.0	µA	*3
Analog input capacitance	C_{AIN}	AN0 to AN7		—	—	30.0	pF	
Allowable signal source impedance	R_{AIN}	AN0 to AN7		—	—	5.0	kΩ	
Resolution (data length)				10	10	10	bit	
Conversion time (single mode)			AV_{CC} = 3.3 V to 5.5 V	134	—	—	t_{cyc}	
Nonlinearity error				—	—	±7.5	LSB	
Offset error				—	—	±7.5	LSB	
Full-scale error				—	—	±7.5	LSB	
Quantization error				—	—	±0.5	LSB	
Absolute accuracy				—	—	±8.0	LSB	
Conversion time (single mode)			AV_{CC} = 4.0 V to 5.5 V	70	—	—	t_{cyc}	
Nonlinearity error				—	—	±7.5	LSB	
Offset error				—	—	±7.5	LSB	
Full-scale error				—	—	±7.5	LSB	
Quantization error				—	—	±0.5	LSB	
Absolute accuracy				—	—	±8.0	LSB	

Item	Symbol	Applicable Pins	Test Condition	Values Min	Typ	Max	Unit	Reference Figure
Conversion time (single mode)			AV_{CC} = 4.0 V to 5.5 V	134	—	—	t_{cyc}	
Nonlinearity error				—	—	±3.5	LSB	
Offset error				—	—	±3.5	LSB	
Full-scale error				—	—	±3.5	LSB	
Quantization error				—	—	±0.5	LSB	
Absolute accuracy				—	—	±4.0	LSB	

Notes: 1. Set $AV_{CC} = V_{CC}$ when the A/D converter is not used.
2. AI_{STOP1} is the current in active and sleep modes while the A/D converter is idle.
3. AI_{STOP2} is the current at reset and in standby, subactive, and subsleep modes while the A/D converter is idle.

20.2.5 Watchdog Timer Characteristics

Table 20.7 Watchdog Timer Characteristics

V_{CC} = 3.0 V to 5.5 V, V_{SS} = 0.0 V, T_a = –20°C to +75°C, unless otherwise specified.

Item	Symbol	Applicable Pins	Test Condition	Values Min	Typ	Max	Unit	Reference Figure
On-chip oscillator overflow time	t_{OVF}			0.2	0.4	—	s	*

Note: * Shows the time to count from 0 to 255, at which point an internal reset is generated, when the internal oscillator is selected.

20.2.6 Memory Characteristics

Table 20.8 Flash Memory Characteristics

V_{CC} = 3.0 V to 5.5 V, V_{SS} = 0.0 V, T_a = –20°C to +75°C, unless otherwise specified.

Item		Symbol	Test Condition	Values Min	Typ	Max	Unit
Programming time (per 128 bytes)*1*2*4		t_P		—	7	200	ms
Erase time (per block) *1*3*6		t_E		—	100	1200	ms
Reprogramming count		N_{WEC}		1000	10000	—	Times
Programming	Wait time after SWE bit setting*1	x		1	—	—	µs
	Wait time after PSU bit setting*1	y		50	—	—	µs
	Wait time after P bit setting*1*4	z1	1 ≤ n ≤ 6	28	30	32	µs
		z2	7 ≤ n ≤ 1000	198	200	202	µs
		z3	Additional-programming	8	10	12	µs
	Wait time after P bit clear*1	α		5	—	—	µs
	Wait time after PSU bit clear*1	β		5	—	—	µs
	Wait time after PV bit setting*1	γ		4	—	—	µs
	Wait time after dummy write*1	ε		2	—	—	µs
	Wait time after PV bit clear*1	η		2	—	—	µs
	Wait time after SWE bit clear*1	θ		100	—	—	µs
	Maximum programming count*1*4*5	N		—	—	1000	Times

Item		Symbol	Test Condition	Values Min	Typ	Max	Unit
Erase	Wait time after SWE bit setting*[1]	x		1	—	—	µs
	Wait time after ESU bit setting*[1]	y		100	—	—	µs
	Wait time after E bit setting*[1]*[6]	z		10	—	100	ms
	Wait time after E bit clear*[1]	α		10	—	—	µs
	Wait time after ESU bit clear*[1]	β		10	—	—	µs
	Wait time after EV bit setting*[1]	γ		20	—	—	µs
	Wait time after dummy write*[1]	ε		2	—	—	µs
	Wait time after EV bit clear*[1]	η		4	—	—	µs
	Wait time after SWE bit clear*[1]	θ		100	—	—	µs
	Maximum erase count*[1]*[6]*[7]	N		—	—	120	Times

Notes:
1. Make the time settings in accordance with the program/erase algorithms.
2. The programming time for 128 bytes. (Indicates the total time for which the P bit in flash memory control register 1 (FLMCR1) is set. The program-verify time is not included.)
3. The time required to erase one block. (Indicates the time for which the E bit in flash memory control register 1 (FLMCR1) is set. The erase-verify time is not included.)
4. Programming time maximum value (t_P(MAX)) = wait time after P bit setting (z) × maximum programming count (N)
5. Set the maximum programming count (N) according to the actual set values of z1, z2, and z3, so that it does not exceed the programming time maximum value (t_P(MAX)). The wait time after P bit setting (z1, z2) should be changed as follows according to the value of the programming count (n).

   Programming count (n)

   $1 \leq n \leq 6$ z1 = 30 µs

   $7 \leq n \leq 1000$ z2 = 200 µs
6. Erase time maximum value (t_E(max)) = wait time after E bit setting (z) × maximum erase count (N)
7. Set the maximum maximum erase count (N) according to the actual set value of (z), so that it does not exceed the erase time maximum value (t_E(max)).

20.2.7 EEPROM Characteristics

Table 20.9 EEPROM Characteristics

V_{CC} = 3.0 V to 5.5 V, V_{SS} = 0.0 V, T_a = –20°C to +75°C, unless otherwise specified.

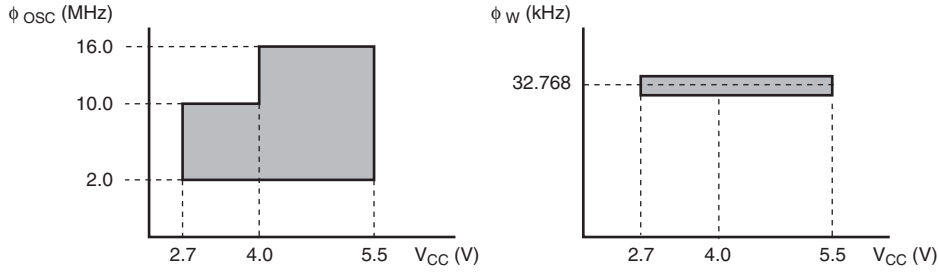
Item	Symbol	Test Condition	Values Min	Typ	Max	Unit	Reference Figure
SCL input cycle time	t_{SCL}		2500	—		ns	Figure 20.7
SCL input high pulse width	t_{SCLH}		600	—	—	µs	
SCL input low pulse width	t_{SCLL}		1200	—	—	ns	
SCL, SDA input spike pulse removal time	t_{SP}		—	—	50	ns	
SDA input bus-free time	t_{BUF}		1200	—	—	ns	
Start condition input hold time	t_{STAH}		600	—	—	ns	
Retransmit start condition input setup time	t_{STAS}		600	—	—	ns	
Stop condition input setup time	t_{STOS}		600	—	—	ns	
Data input setup time	t_{SDAS}		160	—	—	ns	
Data input hold time	t_{SDAH}		0	—	—	ns	
SCL, SDA input fall time	t_{Sf}		—	—	300	ns	
SDA input rise time	t_{Sr}		—	—	300	ns	
Data output hold time	t_{DH}		50	—	—	ns	
SCL, SDA capacitive load	C_b		0	—	400	pF	
Access time	t_{AA}		100	—	900	ns	
Cycle time at writing*	t_{WC}		—	—	10	ms	
Reset release time	t_{RES}		—	—	13	ms	

Note: * Cycle time at writing is a time from the stop condition to write completion (internal control).

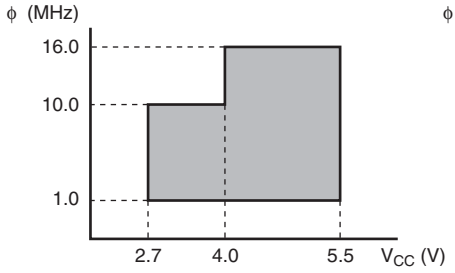
20.3 Electrical Characteristics (Mask ROM Version)

20.3.1 Power Supply Voltage and Operating Ranges

Power Supply Voltage and Oscillation Frequency Range

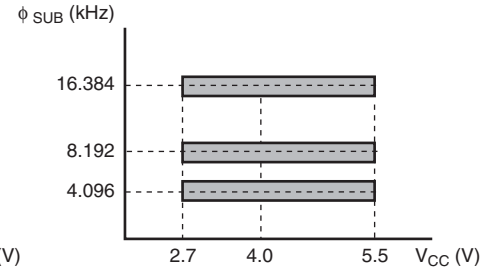


Power Supply Voltage and Operating Frequency Range

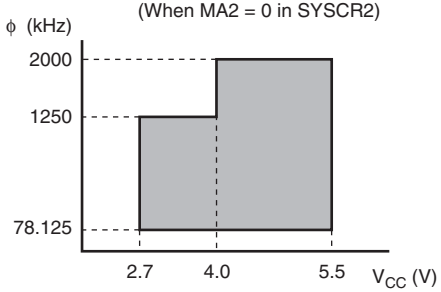


- AV_{CC} = 3.0 V to 5.5 V
- Active mode
- Sleep mode

(When MA2 = 0 in SYSCR2)



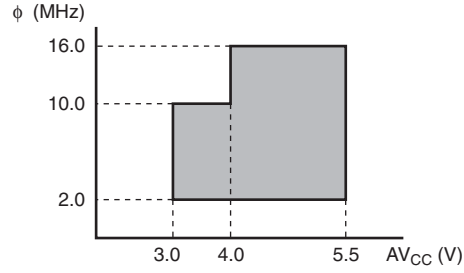
- AV_{CC} = 3.0 V to 5.5 V
- Subactive mode
- Subsleep mode



- AV_{CC} = 3.0 V to 5.5 V
- Active mode
- Sleep mode

(When MA2 = 1 in SYSCR2)

Analog Power Supply Voltage and A/D Converter Accuracy Guarantee Range:



20.3.2 DC Characteristics

Table 20.10 DC Characteristics (1)

V_{CC} = 2.7 V to 5.5 V, V_{SS} = 0.0 V, T_a = –20°C to +75°C unless otherwise indicated.

Item	Symbol	Applicable Pins	Test Condition	Values Min	Typ	Max	Unit	Notes
Input high voltage	V_{IH}	$\overline{RES}$, $\overline{NMI}$, $\overline{WKP0}$ to $\overline{WKP5}$, $\overline{IRQ0}$ to $\overline{IRQ3}$, $\overline{ADTRG}$, TMRIV, TMCIV, FTCI, FTIOA to FTIOD, SCK3, TRGV	V_{CC} = 4.0 V to 5.5 V	V_{CC} × 0.8	—	V_{CC} + 0.3	V	
				V_{CC} × 0.9	—	V_{CC} + 0.3	V	
		RXD, SCL, SDA, P10 to P12, P14 to P17, P20 to P22, P50 to P57, P74 to P76, P80 to P87	V_{CC} = 4.0 V to 5.5 V	V_{CC} × 0.7	—	V_{CC} + 0.3	V	
				V_{CC} × 0.8	—	V_{CC} + 0.3	V	
		PB0 to PB7	V_{CC} = 4.0 V to 5.5 V	V_{CC} × 0.7	—	AV_{CC} + 0.3	V	
				V_{CC} × 0.8	—	AV_{CC} + 0.3	V	
		OSC1	V_{CC} = 4.0 V to 5.5 V	V_{CC} – 0.5	—	V_{CC} + 0.3	V	
				V_{CC} – 0.3	—	V_{CC} + 0.3	V	

Item	Symbol	Applicable Pins	Test Condition	Values Min	Typ	Max	Unit	Notes
Input low voltage	V_{IL}	$\overline{RES}$, $\overline{NMI}$, $\overline{WKP0}$ to $\overline{WKP5}$, $\overline{IRQ0}$ to $\overline{IRQ3}$, $\overline{ADTRG}$, TMRIV, TMCIV, FTCI, FTIOA to FTIOD, SCK3, TRGV	V_{CC} = 4.0 V to 5.5 V	–0.3	—	V_{CC} × 0.2	V	
				–0.3	—	V_{CC} × 0.1	V	
		RXD, SCL, SDA, P10 to P12, P14 to P17, P20 to P22, P50 to P57, P74 to P76, P80 to P87, PB0 to PB7	V_{CC} = 4.0 V to 5.5 V	–0.3	—	V_{CC} × 0.3	V	
				–0.3	—	V_{CC} × 0.2	V	
		OSC1	V_{CC} = 4.0 V to 5.5 V	–0.3	—	0.5	V	
				–0.3	—	0.3	V	
Output high voltage	V_{OH}	P10 to P12, P14 to P17, P20 to P22, P50 to P55, P74 to P76, P80 to P87	V_{CC} = 4.0 V to 5.5 V $-I_{OH}$ = 1.5 mA	V_{CC} – 1.0	—	—	V	
			$-I_{OH}$ = 0.1 mA	V_{CC} – 0.5	—	—	V	
		P56, P57	V_{CC} = 4.0 V to 5.5 V $-I_{OH}$ = 0.1 mA	V_{CC} – 2.5	—	—	V	
			V_{CC} =2.7 V to 4.0 V $-I_{OH}$ = 0.1 mA	V_{CC} – 2.0	—	—	V	

Item	Symbol	Applicable Pins	Test Condition	Values Min	Typ	Max	Unit	Notes
Output low voltage	V_{OL}	P10 to P12, P14 to P17, P20 to P22, P50 to P57, P74 to P76	V_{CC} = 4.0 V to 5.5 V I_{OL} = 1.6 mA	—	—	0.6	V	
			I_{OL} = 0.4 mA	—	—	0.4	V	
		P80 to P87	V_{CC} = 4.0 V to 5.5 V I_{OL} = 20.0 mA	—	—	1.5	V	
			V_{CC} = 4.0 V to 5.5 V I_{OL} = 10.0 mA	—	—	1.0	V	
			V_{CC} = 4.0 V to 5.5 V I_{OL} = 1.6 mA	—	—	0.4	V	
			I_{OL} = 0.4 mA	—	—	0.4	V	
		SCL, SDA	V_{CC} = 4.0 to I_{OL} = 6.0 mA	—	—	0.6	V	
			I_{OL} = 3.0 mA	—	—	0.4	V	
Input/ output leakage current	$\lvert I_{IL} \rvert$	OSC1, $\overline{RES}$, $\overline{NMI}$, $\overline{WKP0}$ to $\overline{WKP5}$, $\overline{IRQ0}$ to $\overline{IRQ3}$, $\overline{ADTRG}$, TRGV, TMRIV, TMCIV, FTCI, FTIOA to FTIOD, RXD, SCK3, SCL, SDA	V_{IN} = 0.5 V to (V_{CC} – 0.5 V)	—	—	1.0	µA	
		P10 to P12, P14 to P17, P20 to P22, P50 to P57, P74 to P76, P80 to P87	V_{IN} = 0.5 V to (V_{CC} – 0.5 V)	—	—	1.0	µA	
		PB0 to PB7	V_{IN} = 0.5 V to (AV_{CC} – 0.5 V)	—	—	1.0	µA	
Pull-up MOS current	$-I_p$	P10 to P12, P14 to P17,	V_{CC} = 5.0 V, V_{IN} = 0.0 V	50.0	—	300.0	µA	
		P50 to P55	V_{CC} = 3.0 V, V_{IN} = 0.0 V	—	60.0	—	µA	Reference value

Item	Symbol	Applicable Pins	Test Condition	Values Min	Values Typ	Values Max	Unit	Notes
Input capacitance	C_{in}	All input pins except power supply pins	f = 1 MHz, V_{IN} = 0.0 V, T_a = 25°C	—	—	15.0	pF	
Active mode supply current	I_{OPE1}	V_{CC}	Active mode 1 V_{CC} = 5.0 V, f_{OSC} = 16 MHz	—	15.0	22.5	mA	*
			Active mode 1 V_{CC} = 3.0 V, f_{OSC} = 10 MHz	—	8.0	—	mA	* Reference value
	I_{OPE2}	V_{CC}	Active mode 2 V_{CC} = 5.0 V, f_{OSC} = 16 MHz	—	1.8	2.7	mA	*
			Active mode 2 V_{CC} = 3.0 V, f_{OSC} = 10 MHz	—	1.2	—	mA	* Reference value
Sleep mode supply current	I_{SLEEP1}	V_{CC}	Sleep mode 1 V_{CC} = 5.0 V, f_{OSC} = 16 MHz	—	7.1	13.0	mA	*
			Sleep mode 1 V_{CC} = 3.0 V, f_{OSC} = 10 MHz	—	4.0	—	mA	* Reference value
	I_{SLEEP2}	V_{CC}	Sleep mode 2 V_{CC} = 5.0 V, f_{OSC} = 16 MHz	—	1.1	2.0	mA	*
			Sleep mode 2 V_{CC} = 3.0 V, f_{OSC} = 10 MHz	—	0.5	—	mA	* Reference value
Subactive mode supply current	I_{SUB}	V_{CC}	V_{CC} = 3.0 V 32-kHz crystal resonator ($\phi_{SUB} = \phi_W/2$)	—	35.0	70.0	µA	*
			V_{CC} = 3.0 V 32-kHz crystal resonator ($\phi_{SUB} = \phi_W/8$)	—	25.0	—	µA	* Reference value
Subsleep mode supply current	I_{SUBSP}	V_{CC}	V_{CC} = 3.0 V 32-kHz crystal resonator ($\phi_{SUB} = \phi_W/2$)	—	25.0	50.0	µA	*
Standby mode supply current	I_{STBY}	V_{CC}	32-kHz crystal resonator not used	—	—	5.0	µA	*

Item	Symbol	Applicable Pins	Test Condition	Values Min	Typ	Max	Unit	Notes
RAM data retaining voltage	V_{RAM}	V_{CC}		2.0	—	—	V	

Note: * Pin states during supply current measurement are given below (excluding current in the pull-up MOS transistors and output buffers).

Mode	$\overline{RES}$ Pin	Internal State	Other Pins	Oscillator Pins
Active mode 1	V_{CC}	Operates	V_{CC}	Main clock: ceramic or crystal resonator Subclock: Pin X1 = V_{SS}
Active mode 2		Operates (ϕosc/64)		
Sleep mode 1	V_{CC}	Only timers operate	V_{CC}	
Sleep mode 2		Only timers operate (ϕosc/64)		
Subactive mode	V_{CC}	Operates	V_{CC}	Main clock: ceramic or crystal resonator Subclock: crystal resonator
Subsleep mode	V_{CC}	Only timers operate	V_{CC}	
Standby mode	V_{CC}	CPU and timers both stop	V_{CC}	Main clock: ceramic or crystal resonator Subclock: Pin X1 = V_{SS}

Table 20.10 DC Characteristics (2)

V_{CC} = 2.7 V to 5.5 V, V_{SS} = 0.0 V, T_a = –20°C to +75°C, unless otherwise indicated.

Item	Symbol	Applicable Pins	Test Condition	Values Min	Typ	Max	Unit
Allowable output low current (per pin)	I_{OL}	Output pins except port 8, SCL, and SDA	V_{CC} = 4.0 V to 5.5 V	—	—	2.0	mA
		Port 8		—	—	20.0	mA
		Port 8		—	—	10.0	mA
		SCL and SDA		—	—	6.0	mA
		Output pins except port 8, SCL,, and SDA		—	—	0.5	mA
Allowable output low current (total)	ΣI_{OL}	Output pins except port 8, SCL and SDA	V_{CC} = 4.0 V to 5.5 V	—	—	40.0	mA
		Port 8, SCL, and SDA		—	—	80.0	mA
		Output pins except port 8, SCL, and SDA		—	—	20.0	mA
		Port 8, SCL, and SDA		—	—	40.0	mA
Allowable output high current (per pin)	$I -I_{OH} I$	All output pins	V_{CC} = 4.0 V to 5.5 V	—	—	2.0	mA
				—	—	0.2	mA
Allowable output high current (total)	$I -\Sigma I_{OH} I$	All output pins	V_{CC} = 4.0 V to 5.5 V	—	—	30.0	mA
				—	—	8.0	mA

20.3.3 AC Characteristics

Table 20.11 AC Characteristics

V_{CC} = 2.7 V to 5.5 V, V_{SS} = 0.0 V, T_a = –20°C to +75°C, unless otherwise specified.

Item	Symbol	Applicable Pins	Test Condition	Values Min	Typ	Max	Unit	Reference Figure
System clock oscillation frequency	f_{OSC}	OSC1, OSC2	V_{CC} = 4.0 V to 5.5 V	2.0	—	16.0	MHz	*1
				2.0		10.0		
System clock (ϕ) cycle time	t_{cyc}			1	—	64	t_{OSC}	*2
				—	—	12.8	µs	
Subclock oscillation frequency	f_W	X1, X2		—	32.768	—	kHz	
Watch clock (ϕ_W) cycle time	t_W	X1, X2		—	30.5	—	µs	
Subclock (ϕ_{SUB}) cycle time	t_{subcyc}			2	—	8	t_W	*2
Instruction cycle time				2	—	—	t_{cyc} t_{subcyc}	
Oscillation stabilization time (crystal resonator)	t_{rc}	OSC1, OSC2		—	—	10.0	ms	
Oscillation stabilization time (ceramic resonator)	t_{rc}	OSC1, OSC2		—	—	5.0	ms	
Oscillation stabilization time	t_{rcx}	X1, X2		—	—	2.0	s	
External clock high width	t_{CPH}	OSC1	V_{CC} = 4.0 V to 5.5 V	25.0	—	—	ns	Figure 20.1
				40.0	—	—	ns	
External clock low width	t_{CPL}	OSC1	V_{CC} = 4.0 V to 5.5 V	25.0	—	—	ns	
				40.0	—	—	ns	
External clock rise time	t_{CPr}	OSC1	V_{CC} = 4.0 V to 5.5 V	—	—	10.0	ns	
				—	—	15.0	ns	
External clock fall time	t_{CPf}	OSC1	V_{CC} = 4.0 V to 5.5 V	—	—	10.0	ns	
				—	—	15.0	ns	

Item	Symbol	Applicable Pins	Test Condition	Values Min	Typ	Max	Unit	Reference Figure
$\overline{\text{RES}}$ pin low width	t_{REL}	$\overline{\text{RES}}$	At power-on and in modes other than those below	t_{rc}	—	—	ms	Figure 20.2
			In active mode and sleep mode operation	10	—	—	t_{cyc}	
Input pin high width	t_{IH}	$\overline{\text{NMI}}$, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ3}}$, $\overline{\text{WKP0}}$ to $\overline{\text{WKP5}}$, TMCIV, TMRIV, TRGV, $\overline{\text{ADTRG}}$, FTCI, FTIOA to FTIOD		2	—	—	t_{cyc} t_{subcyc}	Figure 20.3
Input pin low width	t_{IL}	$\overline{\text{NMI}}$, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ3}}$, $\overline{\text{WKP0}}$ to $\overline{\text{WKP5}}$, TMCIV, TMRIV, TRGV, $\overline{\text{ADTRG}}$, FTCI, FTIOA to FTIOD		2	—	—	t_{cyc} t_{subcyc}	

Notes: 1. When an external clock is input, the minimum system clock oscillator frequency is 1.0 MHz.

2. Determined by MA2, MA1, MA0, SA1, and SA0 of system control register 2 (SYSCR2).

Table 20.12 I²C Bus Interface Timing

Item	Symbol	Values			Unit	Test Condition	Reference Figure
		Min	Typ	Max			
SCL input cycle time	t_{SCL}	$12t_{cyc}$ + 600	—	—	ns		Figure 20.4
SCL input high width	t_{SCLH}	$3t_{cyc}$ + 300	—	—	ns		
SCL input low width	t_{SCLL}	$5t_{cyc}$ + 300	—	—	ns		
Input fall time of SCL and SDA	t_{Sf}	—	—	300	ns		
SCL and SDA input spike pulse removal time	t_{SP}	—	—	$1t_{cyc}$	ns		
SDA input bus-free time	t_{BUF}	$5t_{cyc}$	—	—	ns		
Start condition input hold time	t_{STAH}	$3t_{cyc}$	—	—	ns		
Retransmission start condition input setup time	t_{STAS}	$3t_{cyc}$	—	—	ns		
Setup time for stop condition input	t_{STOS}	$3t_{cyc}$	—	—	ns		
Data-input setup time	t_{SDAS}	$1t_{cyc}$+20	—	—	ns		
Data-input hold time	t_{SDAH}	0	—	—	ns		
Capacitive load of SCL and SDA	c_b	0	—	400	pF		
SCL and SDA output fall time	t_{Sf}	—	—	250	ns	V_{CC} = 4.0 V to 5.5 V	
		—	—	300	ns		

Table 20.13 Serial Interface (SCI3) Timing

V_{CC} = 2.7 V to 5.5 V, V_{SS} = 0.0 V, T_a = –20°C to +75°C, unless otherwise specified.

Item		Symbol	Applicable Pins	Test Condition	Values Min	Typ	Max	Unit	Reference Figure
Input clock cycle	Asynchronous	t_{Scyc}	SCK3		4	—	—	t_{cyc}	Figure 20.5
	Clocked synchronous				6	—	—	t_{cyc}	
Input clock pulse width		t_{SCKW}	SCK3		0.4	—	0.6	t_{Scyc}	
Transmit data delay time (clocked synchronous)		t_{TXD}	TXD	V_{CC} = 4.0 V to 5.5 V	—	—	1	t_{cyc}	Figure 20.6
					—	—	1	t_{cyc}	
Receive data setup time (clocked synchronous)		t_{RXS}	RXD	V_{CC} = 4.0 V to 5.5 V	62.5	—	—	ns	
					100.0	—	—	ns	
Receive data hold time (clocked synchronous)		t_{RXH}	RXD	V_{CC} = 4.0 V to 5.5 V	62.5	—	—	ns	
					100.0	—	—	ns	

20.3.4 A/D Converter Characteristics

Table 20.14 A/D Converter Characteristics

V_{CC} = 2.7 V to 5.5 V, V_{SS} = 0.0 V, T_a = –20°C to +75°C, unless otherwise specified.

Item	Symbol	Applicable Pins	Test Condition	Values Min	Values Typ	Values Max	Unit	Reference Figure
Analog power supply voltage	AV_{CC}	AV_{CC}		3.0	V_{CC}	5.5	V	*1
Analog input voltage	AV_{IN}	AN0 to AN7		V_{SS} – 0.3	—	AV_{CC} + 0.3	V	
Analog power supply current	AI_{OPE}	AV_{CC}	AV_{CC} = 5.0 V f_{OSC} = 16 MHz	—	—	2.0	mA	
	AI_{STOP1}	AV_{CC}		—	50	—	µA	*2 Reference value
	AI_{STOP2}	AV_{CC}		—	—	5.0	µA	*3
Analog input capacitance	C_{AIN}	AN0 to AN7		—	—	30.0	pF	
Allowable signal source impedance	R_{AIN}	AN0 to AN7		—	—	5.0	kΩ	
Resolution (data length)				10	10	10	bit	
Conversion time (single mode)			AV_{CC} = 3.0 V to 5.5 V	134	—	—	t_{cyc}	
Nonlinearity error				—	—	±7.5	LSB	
Offset error				—	—	±7.5	LSB	
Full-scale error				—	—	±7.5	LSB	
Quantization error				—	—	±0.5	LSB	
Absolute accuracy				—	—	±8.0	LSB	
Conversion time (single mode)			AV_{CC} = 4.0 V to 5.5 V	70	—	—	t_{cyc}	
Nonlinearity error				—	—	±7.5	LSB	
Offset error				—	—	±7.5	LSB	
Full-scale error				—	—	±7.5	LSB	
Quantization error				—	—	±0.5	LSB	
Absolute accuracy				—	—	±8.0	LSB	

Item	Symbol	Applicable Pins	Test Condition	Values Min	Typ	Max	Unit	Reference Figure
Conversion time (single mode)			AV_{CC} = 4.0 V to 5.5 V	134	—	—	t_{cyc}	
Nonlinearity error				—	—	±3.5	LSB	
Offset error				—	—	±3.5	LSB	
Full-scale error				—	—	±3.5	LSB	
Quantization error				—	—	±0.5	LSB	
Absolute accuracy				—	—	±4.0	LSB	

Notes: 1. Set $AV_{CC} = V_{CC}$ when the A/D converter is not used.
2. AI_{STOP1} is the current in active and sleep modes while the A/D converter is idle.
3. AI_{STOP2} is the current at reset and in standby, subactive, and subsleep modes while the A/D converter is idle.

20.3.5 Watchdog Timer Characteristics

Table 20.15 Watchdog Timer Characteristics

V_{CC} = 2.7 V to 5.5 V, V_{SS} = 0.0 V, T_a = –20°C to +75°C, unless otherwise specified.

Item	Symbol	Applicable Pins	Test Condition	Values Min	Typ	Max	Unit	Reference Figure
On-chip oscillator overflow time	t_{OVF}			0.2	0.4	—	s	*

Note: * Shows the time to count from 0 to 255, at which point an internal reset is generated, when the internal oscillator is selected.

20.4 Operation Timing

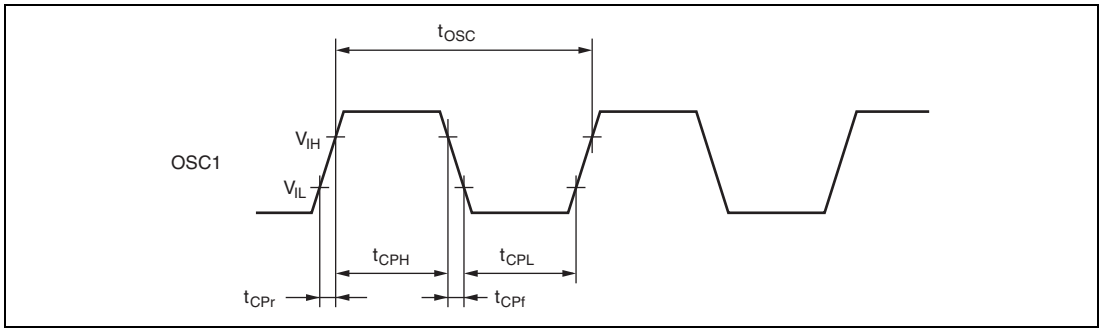


Figure 20.1 System Clock Input Timing

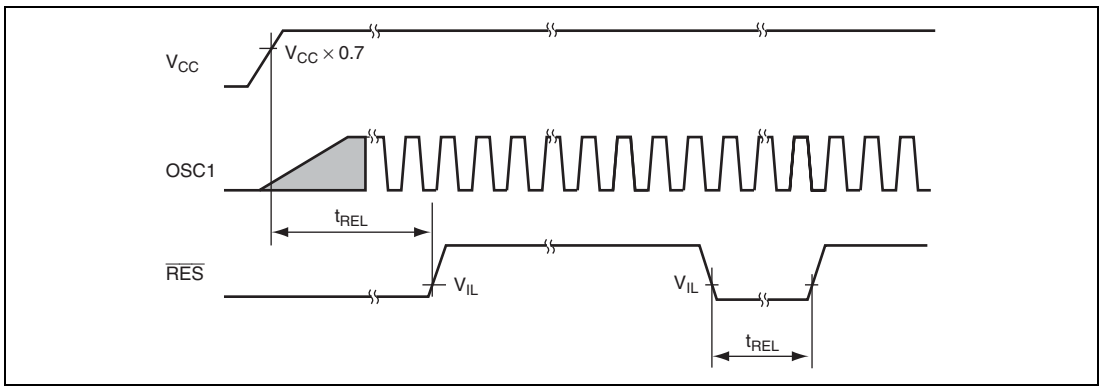


Figure 20.2 $\overline{RES}$ Low Width Timing

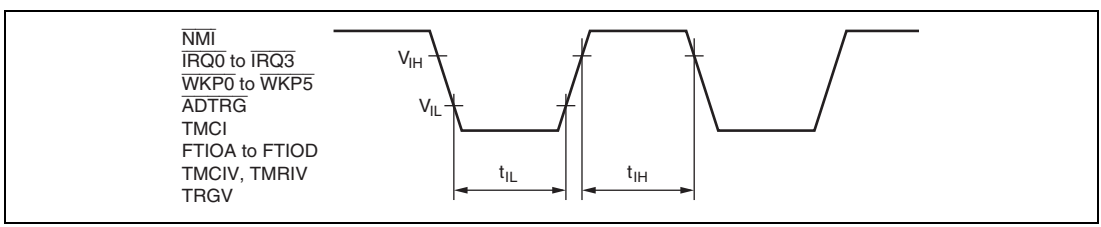


Figure 20.3 Input Timing

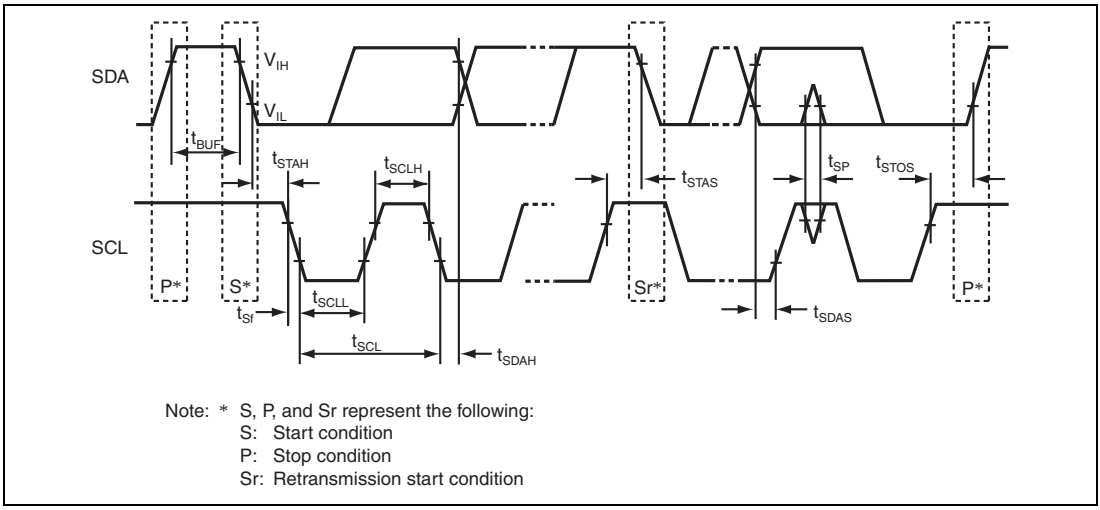


Figure 20.4 I^2C Bus Interface Input/Output Timing

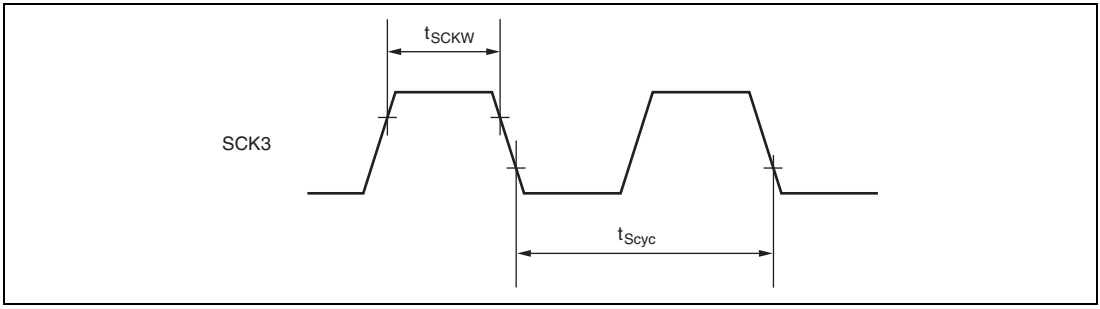
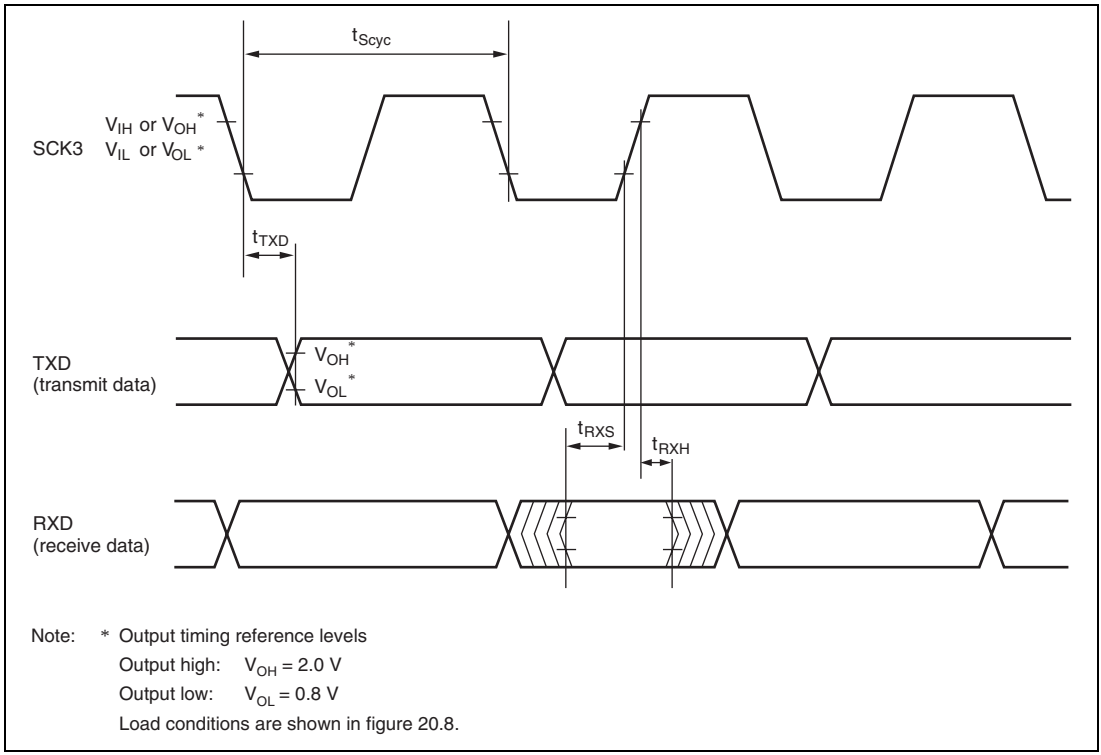


Figure 20.5 SCK3 Input Clock Timing



Note: * Output timing reference levels
Output high: V_{OH} = 2.0 V
Output low: V_{OL} = 0.8 V
Load conditions are shown in figure 20.8.

Figure 20.6 SCI3 Input/Output Timing in Clocked Synchronous Mode

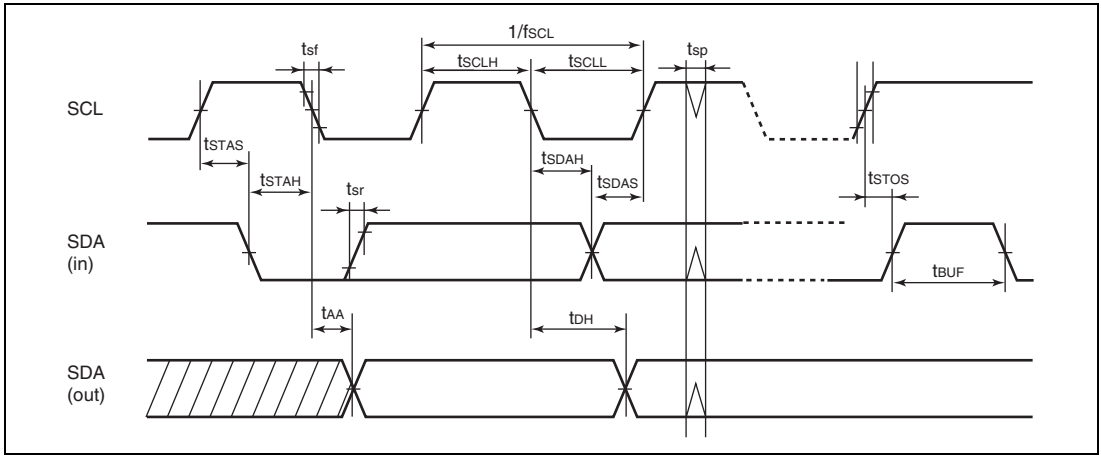


Figure 20.7 EEPROM Bus Timing

20.5 Output Load Condition

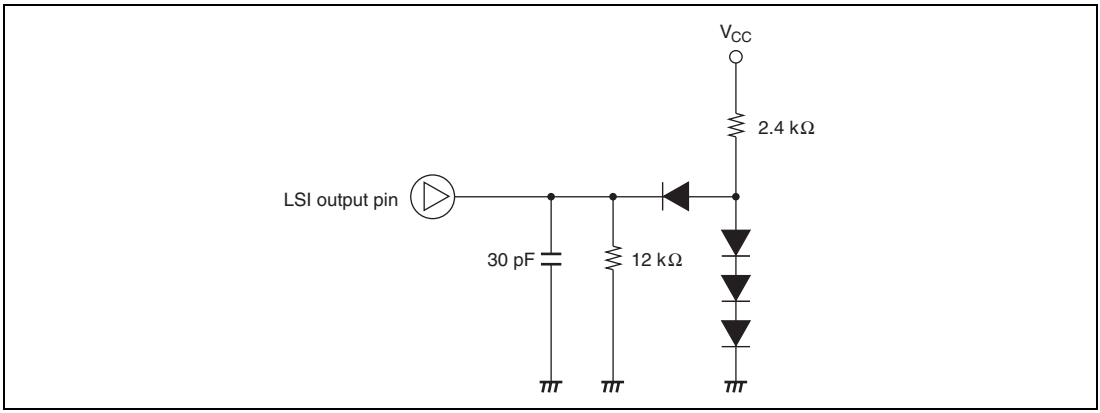


Figure 20.8 Output Load Circuit

Appendix A Instruction Set

A.1 Instruction List

Operand Notation

Symbol	Description
Rd	General (destination*) register
Rs	General (source*) register
Rn	General register*
ERd	General destination register (address register or 32-bit register)
ERs	General source register (address register or 32-bit register)
ERn	General register (32-bit register)
(EAd)	Destination operand
(EAs)	Source operand
PC	Program counter
SP	Stack pointer
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
disp	Displacement
→	Transfer from the operand on the left to the operand on the right, or transition from the state on the left to the state on the right
+	Addition of the operands on both sides
–	Subtraction of the operand on the right from the operand on the left
×	Multiplication of the operands on both sides
÷	Division of the operand on the left by the operand on the right
∧	Logical AND of the operands on both sides
∨	Logical OR of the operands on both sides

Symbol	Description
⊕	Logical exclusive OR of the operands on both sides
¬	NOT (logical complement)
( ), < >	Contents of operand

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

Condition Code Notation

Symbol	Description
↕	Changed according to execution result
*	Undetermined (no guaranteed value)
0	Cleared to 0
1	Set to 1
—	Not affected by execution of the instruction
Δ	Varies depending on conditions, described in notes

Table A.1 Instruction Set

1. Data transfer instructions

Mnemonic		Operand Size	Addressing Mode and Instruction Length (bytes)									Operation	Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	—		I	H	N	Z	V	C	Normal	Advanced
MOV	MOV.B #xx:8, Rd	B	2									#xx:8 → Rd8	—	—	↕	↕	0	—	2	
	MOV.B Rs, Rd	B		2								Rs8 → Rd8	—	—	↕	↕	0	—	2	
	MOV.B @ERs, Rd	B			2							@ERs → Rd8	—	—	↕	↕	0	—	4	
	MOV.B @(d:16, ERs), Rd	B				4						@(d:16, ERs) → Rd8	—	—	↕	↕	0	—	6	
	MOV.B @(d:24, ERs), Rd	B				8						@(d:24, ERs) → Rd8	—	—	↕	↕	0	—	10	
	MOV.B @ERs+, Rd	B					2					@ERs → Rd8 ERs32+1 → ERs32	—	—	↕	↕	0	—	6	
	MOV.B @aa:8, Rd	B						2				@aa:8 → Rd8	—	—	↕	↕	0	—	4	
	MOV.B @aa:16, Rd	B						4				@aa:16 → Rd8	—	—	↕	↕	0	—	6	
	MOV.B @aa:24, Rd	B						6				@aa:24 → Rd8	—	—	↕	↕	0	—	8	
	MOV.B Rs, @ERd	B			2							Rs8 → @ERd	—	—	↕	↕	0	—	4	
	MOV.B Rs, @(d:16, ERd)	B				4						Rs8 → @(d:16, ERd)	—	—	↕	↕	0	—	6	
	MOV.B Rs, @(d:24, ERd)	B				8						Rs8 → @(d:24, ERd)	—	—	↕	↕	0	—	10	
	MOV.B Rs, @–ERd	B					2					ERd32–1 → ERd32 Rs8 → @ERd	—	—	↕	↕	0	—	6	
	MOV.B Rs, @aa:8	B						2				Rs8 → @aa:8	—	—	↕	↕	0	—	4	
	MOV.B Rs, @aa:16	B						4				Rs8 → @aa:16	—	—	↕	↕	0	—	6	
	MOV.B Rs, @aa:24	B						6				Rs8 → @aa:24	—	—	↕	↕	0	—	8	
	MOV.W #xx:16, Rd	W	4									#xx:16 → Rd16	—	—	↕	↕	0	—	4	
	MOV.W Rs, Rd	W		2								Rs16 → Rd16	—	—	↕	↕	0	—	2	
	MOV.W @ERs, Rd	W			2							@ERs → Rd16	—	—	↕	↕	0	—	4	
	MOV.W @(d:16, ERs), Rd	W				4						@(d:16, ERs) → Rd16	—	—	↕	↕	0	—	6	
	MOV.W @(d:24, ERs), Rd	W				8						@(d:24, ERs) → Rd16	—	—	↕	↕	0	—	10	
	MOV.W @ERs+, Rd	W					2					@ERs → Rd16 ERs32+2 → @ERd32	—	—	↕	↕	0	—	6	
	MOV.W @aa:16, Rd	W						4				@aa:16 → Rd16	—	—	↕	↕	0	—	6	
	MOV.W @aa:24, Rd	W						6				@aa:24 → Rd16	—	—	↕	↕	0	—	8	
	MOV.W Rs, @ERd	W			2							Rs16 → @ERd	—	—	↕	↕	0	—	4	
	MOV.W Rs, @(d:16, ERd)	W				4						Rs16 → @(d:16, ERd)	—	—	↕	↕	0	—	6	
	MOV.W Rs, @(d:24, ERd)	W				8						Rs16 → @(d:24, ERd)	—	—	↕	↕	0	—	10	

Mnemonic		Operand Size	Addressing Mode and Instruction Length (bytes)									Operation	Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	—		I	H	N	Z	V	C	Normal	Advanced
MOV	MOV.W Rs, @–ERd	W					2					ERd32–2 → ERd32 Rs16 → @ERd	—	—	↕	↕	0	—	6	
	MOV.W Rs, @aa:16	W						4				Rs16 → @aa:16	—	—	↕	↕	0	—	6	
	MOV.W Rs, @aa:24	W						6				Rs16 → @aa:24	—	—	↕	↕	0	—	8	
	MOV.L #xx:32, Rd	L	6									#xx:32 → Rd32	—	—	↕	↕	0	—	6	
	MOV.L ERs, ERd	L		2								ERs32 → ERd32	—	—	↕	↕	0	—	2	
	MOV.L @ERs, ERd	L			4							@ERs → ERd32	—	—	↕	↕	0	—	8	
	MOV.L @(d:16, ERs), ERd	L				6						@(d:16, ERs) → ERd32	—	—	↕	↕	0	—	10	
	MOV.L @(d:24, ERs), ERd	L				10						@(d:24, ERs) → ERd32	—	—	↕	↕	0	—	14	
	MOV.L @ERs+, ERd	L					4					@ERs → ERd32 ERs32+4 → ERs32	—	—	↕	↕	0	—	10	
	MOV.L @aa:16, ERd	L						6				@aa:16 → ERd32	—	—	↕	↕	0	—	10	
	MOV.L @aa:24, ERd	L						8				@aa:24 → ERd32	—	—	↕	↕	0	—	12	
	MOV.L ERs, @ERd	L			4							ERs32 → @ERd	—	—	↕	↕	0	—	8	
	MOV.L ERs, @(d:16, ERd)	L				6						ERs32 → @(d:16, ERd)	—	—	↕	↕	0	—	10	
	MOV.L ERs, @(d:24, ERd)	L				10						ERs32 → @(d:24, ERd)	—	—	↕	↕	0	—	14	
	MOV.L ERs, @–ERd	L					4					ERd32–4 → ERd32 ERs32 → @ERd	—	—	↕	↕	0	—	10	
	MOV.L ERs, @aa:16	L						6				ERs32 → @aa:16	—	—	↕	↕	0	—	10	
	MOV.L ERs, @aa:24	L						8				ERs32 → @aa:24	—	—	↕	↕	0	—	12	
POP	POP.W Rn	W									2	@SP → Rn16 SP+2 → SP	—	—	↕	↕	0	—	6	
	POP.L ERn	L									4	@SP → ERn32 SP+4 → SP	—	—	↕	↕	0	—	10	
PUSH	PUSH.W Rn	W									2	SP–2 → SP Rn16 → @SP	—	—	↕	↕	0	—	6	
	PUSH.L ERn	L									4	SP–4 → SP ERn32 → @SP	—	—	↕	↕	0	—	10	
MOVFPE	MOVFPE @aa:16, Rd	B						4				Cannot be used in this LSI	Cannot be used in this LSI							
MOVTPE	MOVTPE Rs, @aa:16	B						4				Cannot be used in this LSI	Cannot be used in this LSI							

2. Arithmetic instructions

Mnemonic		Operand Size	Addressing Mode and Instruction Length (bytes)									Operation	Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	—		I	H	N	Z	V	C	Normal	Advanced
ADD	ADD.B #xx:8, Rd	B	2									Rd8+#xx:8 → Rd8	—	↕	↕	↕	↕	↕	2	
	ADD.B Rs, Rd	B		2								Rd8+Rs8 → Rd8	—	↕	↕	↕	↕	↕	2	
	ADD.W #xx:16, Rd	W	4									Rd16+#xx:16 → Rd16	—	(1)	↕	↕	↕	↕	4	
	ADD.W Rs, Rd	W		2								Rd16+Rs16 → Rd16	—	(1)	↕	↕	↕	↕	2	
	ADD.L #xx:32, ERd	L	6									ERd32+#xx:32 → ERd32	—	(2)	↕	↕	↕	↕	6	
	ADD.L ERs, ERd	L		2								ERd32+ERs32 → ERd32	—	(2)	↕	↕	↕	↕	2	
ADDX	ADDX.B #xx:8, Rd	B	2									Rd8+#xx:8 +C → Rd8	—	↕	↕	(3)	↕	↕	2	
	ADDX.B Rs, Rd	B		2								Rd8+Rs8 +C → Rd8	—	↕	↕	(3)	↕	↕	2	
ADDS	ADDS.L #1, ERd	L		2								ERd32+1 → ERd32	—	—	—	—	—	—	2	
	ADDS.L #2, ERd	L		2								ERd32+2 → ERd32	—	—	—	—	—	—	2	
	ADDS.L #4, ERd	L		2								ERd32+4 → ERd32	—	—	—	—	—	—	2	
INC	INC.B Rd	B		2								Rd8+1 → Rd8	—	—	↕	↕	↕	—	2	
	INC.W #1, Rd	W		2								Rd16+1 → Rd16	—	—	↕	↕	↕	—	2	
	INC.W #2, Rd	W		2								Rd16+2 → Rd16	—	—	↕	↕	↕	—	2	
	INC.L #1, ERd	L		2								ERd32+1 → ERd32	—	—	↕	↕	↕	—	2	
	INC.L #2, ERd	L		2								ERd32+2 → ERd32	—	—	↕	↕	↕	—	2	
DAA	DAA Rd	B		2								Rd8 decimal adjust → Rd8	—	*	↕	↕	*	↕	2	
SUB	SUB.B Rs, Rd	B		2								Rd8–Rs8 → Rd8	—	↕	↕	↕	↕	↕	2	
	SUB.W #xx:16, Rd	W	4									Rd16–#xx:16 → Rd16	—	(1)	↕	↕	↕	↕	4	
	SUB.W Rs, Rd	W		2								Rd16–Rs16 → Rd16	—	(1)	↕	↕	↕	↕	2	
	SUB.L #xx:32, ERd	L	6									ERd32–#xx:32 → ERd32	—	(2)	↕	↕	↕	↕	6	
	SUB.L ERs, ERd	L		2								ERd32–ERs32 → ERd32	—	(2)	↕	↕	↕	↕	2	
SUBX	SUBX.B #xx:8, Rd	B	2									Rd8–#xx:8–C → Rd8	—	↕	↕	(3)	↕	↕	2	
	SUBX.B Rs, Rd	B		2								Rd8–Rs8–C → Rd8	—	↕	↕	(3)	↕	↕	2	
SUBS	SUBS.L #1, ERd	L		2								ERd32–1 → ERd32	—	—	—	—	—	—	2	
	SUBS.L #2, ERd	L		2								ERd32–2 → ERd32	—	—	—	—	—	—	2	
	SUBS.L #4, ERd	L		2								ERd32–4 → ERd32	—	—	—	—	—	—	2	
DEC	DEC.B Rd	B		2								Rd8–1 → Rd8	—	—	↕	↕	↕	—	2	
	DEC.W #1, Rd	W		2								Rd16–1 → Rd16	—	—	↕	↕	↕	—	2	
	DEC.W #2, Rd	W		2								Rd16–2 → Rd16	—	—	↕	↕	↕	—	2	

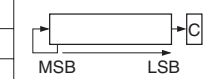
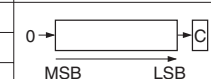
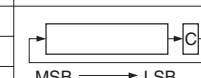
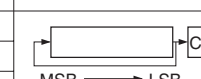
Mnemonic		Operand Size	Addressing Mode and Instruction Length (bytes)									Operation	Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	—		I	H	N	Z	V	C	Normal	Advanced
DEC	DEC.L #1, ERd	L		2								ERd32–1 → ERd32	—	—	↕	↕	↕	—	2	
	DEC.L #2, ERd	L		2								ERd32–2 → ERd32	—	—	↕	↕	↕	—	2	
DAS	DAS.Rd	B		2								Rd8 decimal adjust → Rd8	—	*	↕	↕	*	—	2	
MULXU	MULXU. B Rs, Rd	B		2								Rd8 × Rs8 → Rd16 (unsigned multiplication)	—	—	—	—	—	—	14	
	MULXU. W Rs, ERd	W		2								Rd16 × Rs16 → ERd32 (unsigned multiplication)	—	—	—	—	—	—	22	
MULXS	MULXS. B Rs, Rd	B		4								Rd8 × Rs8 → Rd16 (signed multiplication)	—	—	↕	↕	—	—	16	
	MULXS. W Rs, ERd	W		4								Rd16 × Rs16 → ERd32 (signed multiplication)	—	—	↕	↕	—	—	24	
DIVXU	DIVXU. B Rs, Rd	B		2								Rd16 ÷ Rs8 → Rd16 (RdH: remainder, RdL: quotient) (unsigned division)	—	—	(6)	(7)	—	—	14	
	DIVXU. W Rs, ERd	W		2								ERd32 ÷ Rs16 → ERd32 (Ed: remainder, Rd: quotient) (unsigned division)	—	—	(6)	(7)	—	—	22	
DIVXS	DIVXS. B Rs, Rd	B		4								Rd16 ÷ Rs8 → Rd16 (RdH: remainder, RdL: quotient) (signed division)	—	—	(8)	(7)	—	—	16	
	DIVXS. W Rs, ERd	W		4								ERd32 ÷ Rs16 → ERd32 (Ed: remainder, Rd: quotient) (signed division)	—	—	(8)	(7)	—	—	24	
CMP	CMP.B #xx:8, Rd	B	2									Rd8–#xx:8	—	↕	↕	↕	↕	↕	2	
	CMP.B Rs, Rd	B		2								Rd8–Rs8	—	↕	↕	↕	↕	↕	2	
	CMP.W #xx:16, Rd	W	4									Rd16–#xx:16	—	(1)	↕	↕	↕	↕	4	
	CMP.W Rs, Rd	W		2								Rd16–Rs16	—	(1)	↕	↕	↕	↕	2	
	CMP.L #xx:32, ERd	L	6									ERd32–#xx:32	—	(2)	↕	↕	↕	↕	4	
	CMP.L ERs, ERd	L		2								ERd32–ERs32	—	(2)	↕	↕	↕	↕	2	

Mnemonic		Operand Size	Operation	Addressing Mode and Instruction Length (bytes)									Condition Code						No. of States*1	
				#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	—	I	H	N	Z	V	C	Normal	Advanced
NEG	NEG.B Rd	B	0–Rd8 → Rd8		2								—	↕	↕	↕	↕	↕	2	
	NEG.W Rd	W	0–Rd16 → Rd16		2								—	↕	↕	↕	↕	↕	2	
	NEG.L ERd	L	0–ERd32 → ERd32		2								—	↕	↕	↕	↕	↕	2	
EXTU	EXTU.W Rd	W	0 → (<bits 15 to 8> of Rd16)		2								—	—	0	↕	0	—	2	
	EXTU.L ERd	L	0 → (<bits 31 to 16> of ERd32)		2								—	—	0	↕	0	—	2	
EXTS	EXTS.W Rd	W	(<bit 7> of Rd16) → (<bits 15 to 8> of Rd16)		2								—	—	↕	↕	0	—	2	
	EXTS.L ERd	L	(<bit 15> of ERd32) → (<bits 31 to 16> of ERd32)		2								—	—	↕	↕	0	—	2	

3. Logic instructions

Mnemonic		Operand Size	Addressing Mode and Instruction Length (bytes)									Operation	Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	—		I	H	N	Z	V	C	Normal	Advanced
AND	AND.B #xx:8, Rd	B	2									Rd8∧#xx:8 → Rd8	—	—	↕	↕	0	—	2	
	AND.B Rs, Rd	B		2								Rd8∧Rs8 → Rd8	—	—	↕	↕	0	—	2	
	AND.W #xx:16, Rd	W	4									Rd16∧#xx:16 → Rd16	—	—	↕	↕	0	—	4	
	AND.W Rs, Rd	W		2								Rd16∧Rs16 → Rd16	—	—	↕	↕	0	—	2	
	AND.L #xx:32, ERd	L	6									ERd32∧#xx:32 → ERd32	—	—	↕	↕	0	—	6	
	AND.L ERs, ERd	L		4								ERd32∧ERs32 → ERd32	—	—	↕	↕	0	—	4	
OR	OR.B #xx:8, Rd	B	2									Rd8∨#xx:8 → Rd8	—	—	↕	↕	0	—	2	
	OR.B Rs, Rd	B		2								Rd8∨Rs8 → Rd8	—	—	↕	↕	0	—	2	
	OR.W #xx:16, Rd	W	4									Rd16∨#xx:16 → Rd16	—	—	↕	↕	0	—	4	
	OR.W Rs, Rd	W		2								Rd16∨Rs16 → Rd16	—	—	↕	↕	0	—	2	
	OR.L #xx:32, ERd	L	6									ERd32∨#xx:32 → ERd32	—	—	↕	↕	0	—	6	
	OR.L ERs, ERd	L		4								ERd32∨ERs32 → ERd32	—	—	↕	↕	0	—	4	
XOR	XOR.B #xx:8, Rd	B	2									Rd8⊕#xx:8 → Rd8	—	—	↕	↕	0	—	2	
	XOR.B Rs, Rd	B		2								Rd8⊕Rs8 → Rd8	—	—	↕	↕	0	—	2	
	XOR.W #xx:16, Rd	W	4									Rd16⊕#xx:16 → Rd16	—	—	↕	↕	0	—	4	
	XOR.W Rs, Rd	W		2								Rd16⊕Rs16 → Rd16	—	—	↕	↕	0	—	2	
	XOR.L #xx:32, ERd	L	6									ERd32⊕#xx:32 → ERd32	—	—	↕	↕	0	—	6	
	XOR.L ERs, ERd	L		4								ERd32⊕ERs32 → ERd32	—	—	↕	↕	0	—	4	
NOT	NOT.B Rd	B		2								¬ Rd8 → Rd8	—	—	↕	↕	0	—	2	
	NOT.W Rd	W		2								¬ Rd16 → Rd16	—	—	↕	↕	0	—	2	
	NOT.L ERd	L		2								¬ Rd32 → Rd32	—	—	↕	↕	0	—	2	

4. Shift instructions

Mnemonic		Operand Size	Addressing Mode and Instruction Length (bytes)									Operation	Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	—		I	H	N	Z	V	C	Normal	Advanced
SHAL	SHAL.B Rd	B		2									—	—	↕	↕	↕	↕	2	
	SHAL.W Rd	W		2									—	—	↕	↕	↕	↕	2	
	SHAL.L ERd	L		2									—	—	↕	↕	↕	↕	2	
SHAR	SHAR.B Rd	B		2									—	—	↕	↕	0	↕	2	
	SHAR.W Rd	W		2									—	—	↕	↕	0	↕	2	
	SHAR.L ERd	L		2									—	—	↕	↕	0	↕	2	
SHLL	SHLL.B Rd	B		2									—	—	↕	↕	0	↕	2	
	SHLL.W Rd	W		2									—	—	↕	↕	0	↕	2	
	SHLL.L ERd	L		2									—	—	↕	↕	0	↕	2	
SHLR	SHLR.B Rd	B		2									—	—	↕	↕	0	↕	2	
	SHLR.W Rd	W		2									—	—	↕	↕	0	↕	2	
	SHLR.L ERd	L		2									—	—	↕	↕	0	↕	2	
ROTXL	ROTXL.B Rd	B		2									—	—	↕	↕	0	↕	2	
	ROTXL.W Rd	W		2									—	—	↕	↕	0	↕	2	
	ROTXL.L ERd	L		2									—	—	↕	↕	0	↕	2	
ROTXR	ROTXR.B Rd	B		2									—	—	↕	↕	0	↕	2	
	ROTXR.W Rd	W		2									—	—	↕	↕	0	↕	2	
	ROTXR.L ERd	L		2									—	—	↕	↕	0	↕	2	
ROTL	ROTL.B Rd	B		2									—	—	↕	↕	0	↕	2	
	ROTL.W Rd	W		2									—	—	↕	↕	0	↕	2	
	ROTL.L ERd	L		2									—	—	↕	↕	0	↕	2	
ROTR	ROTR.B Rd	B		2									—	—	↕	↕	0	↕	2	
	ROTR.W Rd	W		2									—	—	↕	↕	0	↕	2	
	ROTR.L ERd	L		2									—	—	↕	↕	0	↕	2	

5. Bit manipulation instructions

Mnemonic		Operand Size	Addressing Mode and Instruction Length (bytes)									Operation	Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	—		I	H	N	Z	V	C	Normal	Advanced
BSET	BSET #xx:3, Rd	B		2								(#xx:3 of Rd8) ← 1	—	—	—	—	—	—	2	
	BSET #xx:3, @ERd	B			4							(#xx:3 of @ERd) ← 1	—	—	—	—	—	—	8	
	BSET #xx:3, @aa:8	B						4				(#xx:3 of @aa:8) ← 1	—	—	—	—	—	—	8	
	BSET Rn, Rd	B		2								(Rn8 of Rd8) ← 1	—	—	—	—	—	—	2	
	BSET Rn, @ERd	B			4							(Rn8 of @ERd) ← 1	—	—	—	—	—	—	8	
	BSET Rn, @aa:8	B						4				(Rn8 of @aa:8) ← 1	—	—	—	—	—	—	8	
BCLR	BCLR #xx:3, Rd	B		2								(#xx:3 of Rd8) ← 0	—	—	—	—	—	—	2	
	BCLR #xx:3, @ERd	B			4							(#xx:3 of @ERd) ← 0	—	—	—	—	—	—	8	
	BCLR #xx:3, @aa:8	B						4				(#xx:3 of @aa:8) ← 0	—	—	—	—	—	—	8	
	BCLR Rn, Rd	B		2								(Rn8 of Rd8) ← 0	—	—	—	—	—	—	2	
	BCLR Rn, @ERd	B			4							(Rn8 of @ERd) ← 0	—	—	—	—	—	—	8	
	BCLR Rn, @aa:8	B						4				(Rn8 of @aa:8) ← 0	—	—	—	—	—	—	8	
BNOT	BNOT #xx:3, Rd	B		2								(#xx:3 of Rd8) ← ¬ (#xx:3 of Rd8)	—	—	—	—	—	—	2	
	BNOT #xx:3, @ERd	B			4							(#xx:3 of @ERd) ← ¬ (#xx:3 of @ERd)	—	—	—	—	—	—	8	
	BNOT #xx:3, @aa:8	B						4				(#xx:3 of @aa:8) ← ¬ (#xx:3 of @aa:8)	—	—	—	—	—	—	8	
	BNOT Rn, Rd	B		2								(Rn8 of Rd8) ← ¬ (Rn8 of Rd8)	—	—	—	—	—	—	2	
	BNOT Rn, @ERd	B			4							(Rn8 of @ERd) ← ¬ (Rn8 of @ERd)	—	—	—	—	—	—	8	
	BNOT Rn, @aa:8	B						4				(Rn8 of @aa:8) ← ¬ (Rn8 of @aa:8)	—	—	—	—	—	—	8	
BTST	BTST #xx:3, Rd	B		2								¬ (#xx:3 of Rd8) → Z	—	—	—	↕	—	—	2	
	BTST #xx:3, @ERd	B			4							¬ (#xx:3 of @ERd) → Z	—	—	—	↕	—	—	6	
	BTST #xx:3, @aa:8	B						4				¬ (#xx:3 of @aa:8) → Z	—	—	—	↕	—	—	6	
	BTST Rn, Rd	B		2								¬ (Rn8 of @Rd8) → Z	—	—	—	↕	—	—	2	
	BTST Rn, @ERd	B			4							¬ (Rn8 of @ERd) → Z	—	—	—	↕	—	—	6	
	BTST Rn, @aa:8	B						4				¬ (Rn8 of @aa:8) → Z	—	—	—	↕	—	—	6	
BLD	BLD #xx:3, Rd	B		2								(#xx:3 of Rd8) → C	—	—	—	—	—	↕	2	

Mnemonic		Operand Size	Addressing Mode and Instruction Length (bytes)									Operation	Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	—		I	H	N	Z	V	C	Normal	Advanced
BLD	BLD #xx:3, @ERd	B			4							(#xx:3 of @ERd) → C	—	—	—	—	—	↕	6	
	BLD #xx:3, @aa:8	B						4				(#xx:3 of @aa:8) → C	—	—	—	—	—	↕	6	
BILD	BILD #xx:3, Rd	B		2								¬ (#xx:3 of Rd8) → C	—	—	—	—	—	↕	2	
	BILD #xx:3, @ERd	B			4							¬ (#xx:3 of @ERd) → C	—	—	—	—	—	↕	6	
	BILD #xx:3, @aa:8	B						4				¬ (#xx:3 of @aa:8) → C	—	—	—	—	—	↕	6	
BST	BST #xx:3, Rd	B		2								C → (#xx:3 of Rd8)	—	—	—	—	—	—	2	
	BST #xx:3, @ERd	B			4							C → (#xx:3 of @ERd24)	—	—	—	—	—	—	8	
	BST #xx:3, @aa:8	B						4				C → (#xx:3 of @aa:8)	—	—	—	—	—	—	8	
BIST	BIST #xx:3, Rd	B		2								¬ C → (#xx:3 of Rd8)	—	—	—	—	—	—	2	
	BIST #xx:3, @ERd	B			4							¬ C → (#xx:3 of @ERd24)	—	—	—	—	—	—	8	
	BIST #xx:3, @aa:8	B						4				¬ C → (#xx:3 of @aa:8)	—	—	—	—	—	—	8	
BAND	BAND #xx:3, Rd	B		2								C∧(#xx:3 of Rd8) → C	—	—	—	—	—	↕	2	
	BAND #xx:3, @ERd	B			4							C∧(#xx:3 of @ERd24) → C	—	—	—	—	—	↕	6	
	BAND #xx:3, @aa:8	B						4				C∧(#xx:3 of @aa:8) → C	—	—	—	—	—	↕	6	
BIAND	BIAND #xx:3, Rd	B		2								C∧ ¬ (#xx:3 of Rd8) → C	—	—	—	—	—	↕	2	
	BIAND #xx:3, @ERd	B			4							C∧ ¬ (#xx:3 of @ERd24) → C	—	—	—	—	—	↕	6	
	BIAND #xx:3, @aa:8	B						4				C∧ ¬ (#xx:3 of @aa:8) → C	—	—	—	—	—	↕	6	
BOR	BOR #xx:3, Rd	B		2								C∨(#xx:3 of Rd8) → C	—	—	—	—	—	↕	2	
	BOR #xx:3, @ERd	B			4							C∨(#xx:3 of @ERd24) → C	—	—	—	—	—	↕	6	
	BOR #xx:3, @aa:8	B						4				C∨(#xx:3 of @aa:8) → C	—	—	—	—	—	↕	6	
BIOR	BIOR #xx:3, Rd	B		2								C∨ ¬ (#xx:3 of Rd8) → C	—	—	—	—	—	↕	2	
	BIOR #xx:3, @ERd	B			4							C∨ ¬ (#xx:3 of @ERd24) → C	—	—	—	—	—	↕	6	
	BIOR #xx:3, @aa:8	B						4				C∨ ¬ (#xx:3 of @aa:8) → C	—	—	—	—	—	↕	6	
BXOR	BXOR #xx:3, Rd	B		2								C⊕(#xx:3 of Rd8) → C	—	—	—	—	—	↕	2	
	BXOR #xx:3, @ERd	B			4							C⊕(#xx:3 of @ERd24) → C	—	—	—	—	—	↕	6	
	BXOR #xx:3, @aa:8	B						4				C⊕(#xx:3 of @aa:8) → C	—	—	—	—	—	↕	6	
BIXOR	BIXOR #xx:3, Rd	B		2								C⊕ ¬ (#xx:3 of Rd8) → C	—	—	—	—	—	↕	2	
	BIXOR #xx:3, @ERd	B			4							C⊕ ¬ (#xx:3 of @ERd24) → C	—	—	—	—	—	↕	6	
	BIXOR #xx:3, @aa:8	B						4				C⊕ ¬ (#xx:3 of @aa:8) → C	—	—	—	—	—	↕	6	

6. Branching instructions

Mnemonic		Operand Size	Addressing Mode and Instruction Length (bytes)								Operation	Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa		Branch Condition	I	H	N	Z	V	C	Normal	Advanced
Bcc	BRA d:8 (BT d:8)	—							2		If condition is true then PC ← PC+d else next;	Always	—	—	—	—	—	—	4	
	BRA d:16 (BT d:16)	—							4				—	—	—	—	—	—	6	
	BRN d:8 (BF d:8)	—							2			Never	—	—	—	—	—	—	4	
	BRN d:16 (BF d:16)	—							4				—	—	—	—	—	—	6	
	BHI d:8	—							2			C∨ Z = 0	—	—	—	—	—	—	4	
	BHI d:16	—							4				—	—	—	—	—	—	6	
	BLS d:8	—							2			C∨ Z = 1	—	—	—	—	—	—	4	
	BLS d:16	—							4				—	—	—	—	—	—	6	
	BCC d:8 (BHS d:8)	—							2			C = 0	—	—	—	—	—	—	4	
	BCC d:16 (BHS d:16)	—							4				—	—	—	—	—	—	6	
	BCS d:8 (BLO d:8)	—							2			C = 1	—	—	—	—	—	—	4	
	BCS d:16 (BLO d:16)	—							4				—	—	—	—	—	—	6	
	BNE d:8	—							2			Z = 0	—	—	—	—	—	—	4	
	BNE d:16	—							4				—	—	—	—	—	—	6	
	BEQ d:8	—							2			Z = 1	—	—	—	—	—	—	4	
	BEQ d:16	—							4				—	—	—	—	—	—	6	
	BVC d:8	—							2			V = 0	—	—	—	—	—	—	4	
	BVC d:16	—							4				—	—	—	—	—	—	6	
	BVS d:8	—							2			V = 1	—	—	—	—	—	—	4	
	BVS d:16	—							4				—	—	—	—	—	—	6	
	BPL d:8	—							2			N = 0	—	—	—	—	—	—	4	
	BPL d:16	—							4				—	—	—	—	—	—	6	
	BMI d:8	—							2			N = 1	—	—	—	—	—	—	4	
	BMI d:16	—							4				—	—	—	—	—	—	6	
	BGE d:8	—							2			N⊕V = 0	—	—	—	—	—	—	4	
	BGE d:16	—							4				—	—	—	—	—	—	6	
	BLT d:8	—							2			N⊕V = 1	—	—	—	—	—	—	4	
	BLT d:16	—							4				—	—	—	—	—	—	6	
	BGT d:8	—							2			Z∨ (N⊕V) = 0	—	—	—	—	—	—	4	
	BGT d:16	—							4				—	—	—	—	—	—	6	
	BLE d:8	—							2			Z∨ (N⊕V) = 1	—	—	—	—	—	—	4	
	BLE d:16	—							4				—	—	—	—	—	—	6	

Mnemonic		Operand Size	Addressing Mode and Instruction Length (bytes)									Operation	Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	—		I	H	N	Z	V	C	Normal	Advanced
JMP	JMP @ERn	—			2							PC ← ERn	—	—	—	—	—	—	4	
	JMP @aa:24	—						4				PC ← aa:24	—	—	—	—	—	—	6	
	JMP @@aa:8	—								2		PC ← @aa:8	—	—	—	—	—	—	8	10
BSR	BSR d:8	—							2			PC → @–SP PC ← PC+d:8	—	—	—	—	—	—	6	8
	BSR d:16	—							4			PC → @–SP PC ← PC+d:16	—	—	—	—	—	—	8	10
JSR	JSR @ERn	—			2							PC → @–SP PC ← ERn	—	—	—	—	—	—	6	8
	JSR @aa:24	—						4				PC → @–SP PC ← aa:24	—	—	—	—	—	—	8	10
	JSR @@aa:8	—								2		PC → @–SP PC ← @aa:8	—	—	—	—	—	—	8	12
RTS	RTS	—									2	PC ← @SP+	—	—	—	—	—	—	8	10

7. System control instructions

Mnemonic		Operand Size	Addressing Mode and Instruction Length (bytes)									Operation	Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	|		I	H	N	Z	V	C	Normal	Advanced
TRAPA	TRAPA #x:2	—									2	PC → @–SP CCR → @–SP <vector> → PC	1	—	—	—	—	—	14	16
RTE	RTE	—										CCR ← @SP+ PC ← @SP+	↕	↕	↕	↕	↕	↕	10	
SLEEP	SLEEP	—										Transition to power-down state	—	—	—	—	—	—	2	
LDC	LDC #xx:8, CCR	B	2									#xx:8 → CCR	↕	↕	↕	↕	↕	↕	2	
	LDC Rs, CCR	B		2								Rs8 → CCR	↕	↕	↕	↕	↕	↕	2	
	LDC @ERs, CCR	W			4							@ERs → CCR	↕	↕	↕	↕	↕	↕	6	
	LDC @(d:16, ERs), CCR	W				6						@(d:16, ERs) → CCR	↕	↕	↕	↕	↕	↕	8	
	LDC @(d:24, ERs), CCR	W				10						@(d:24, ERs) → CCR	↕	↕	↕	↕	↕	↕	12	
	LDC @ERs+, CCR	W					4					@ERs → CCR ERs32+2 → ERs32	↕	↕	↕	↕	↕	↕	8	
	LDC @aa:16, CCR	W						6				@aa:16 → CCR	↕	↕	↕	↕	↕	↕	8	
	LDC @aa:24, CCR	W						8				@aa:24 → CCR	↕	↕	↕	↕	↕	↕	10	
STC	STC CCR, Rd	B		2								CCR → Rd8	—	—	—	—	—	—	2	
	STC CCR, @ERd	W			4							CCR → @ERd	—	—	—	—	—	—	6	
	STC CCR, @(d:16, ERd)	W				6						CCR → @(d:16, ERd)	—	—	—	—	—	—	8	
	STC CCR, @(d:24, ERd)	W				10						CCR → @(d:24, ERd)	—	—	—	—	—	—	12	
	STC CCR, @–ERd	W					4					ERd32–2 → ERd32 CCR → @ERd	—	—	—	—	—	—	8	
	STC CCR, @aa:16	W						6				CCR → @aa:16	—	—	—	—	—	—	8	
	STC CCR, @aa:24	W						8				CCR → @aa:24	—	—	—	—	—	—	10	
ANDC	ANDC #xx:8, CCR	B	2									CCR∧#xx:8 → CCR	↕	↕	↕	↕	↕	↕	2	
ORC	ORC #xx:8, CCR	B	2									CCR∨#xx:8 → CCR	↕	↕	↕	↕	↕	↕	2	
XORC	XORC #xx:8, CCR	B	2									CCR⊕#xx:8 → CCR	↕	↕	↕	↕	↕	↕	2	
NOP	NOP	—									2	PC ← PC+2	—	—	—	—	—	—	2	

8. Block transfer instructions

Mnemonic		Operand Size	Addressing Mode and Instruction Length (bytes)									Operation	Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	|		I	H	N	Z	V	C	Normal	Advanced
EEPMOV	EEPMOV. B	—									4	if R4L ≠ 0 then repeat @R5 → @R6 R5+1 → R5 R6+1 → R6 R4L–1 → R4L until R4L=0 else next	—	—	—	—	—	—	8+ 4n*2	
	EEPMOV. W	—									4	if R4 ≠ 0 then repeat @R5 → @R6 R5+1 → R5 R6+1 → R6 R4–1 → R4 until R4=0 else next	—	—	—	—	—	—	8+ 4n*2	

Notes: 1. The number of states in cases where the instruction code and its operands are located in on-chip memory is shown here. For other cases see appendix A.3, Number of Execution States.

2. n is the value set in register R4L or R4.

(1) Set to 1 when a carry or borrow occurs at bit 11; otherwise cleared to 0.

(2) Set to 1 when a carry or borrow occurs at bit 27; otherwise cleared to 0.

(3) Retains its previous value when the result is zero; otherwise cleared to 0.

(4) Set to 1 when the adjustment produces a carry; otherwise retains its previous value.

(5) The number of states required for execution of an instruction that transfers data in synchronization with the E clock is variable.

(6) Set to 1 when the divisor is negative; otherwise cleared to 0.

(7) Set to 1 when the divisor is zero; otherwise cleared to 0.

(8) Set to 1 when the quotient is negative; otherwise cleared to 0.

A.2 Operation Code Map

Table A.2 Operation Code Map (1)

Instruction code:

1st byte	2nd byte
AH	AL	BH	BL

Instruction when most significant bit of BH is 0.

Instruction when most significant bit of BH is 1.

AH \ AL	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	Table A-2 (2)	STC	LDC	ORC	XORC	ANDC	LDC	ADD		Table A-2 (2)	Table A-2 (2)	MOV		ADDX	Table A-2 (2)
1	Table A-2 (2)	Table A-2 (2)	Table A-2 (2)	Table A-2 (2)	OR.B	XOR.B	AND.B	Table A-2 (2)	SUB		Table A-2 (2)	Table A-2 (2)	CMP		SUBX	Table A-2 (2)
2	MOV.B															
3																
4	BRA	BRN	BHI	BLS	BCC	BCS	BNE	BEQ	BVC	BVS	BPL	BMI	BGE	BLT	BGT	BLE
5	MULXU	DIVXU	MULXU	DIVXU	RTS	BSR	RTE	TRAPA	Table A-2 (2)	JMP			BSR	JSR		
6	BSET	BNOT	BCLR	BTST	OR	XOR	AND	BST	MOV				MOV			
7					BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD		Table A-2 (2)	Table A-2 (2)	EEPMOV	Table A-2 (3)			
8	ADD															
9	ADDX															
A	CMP															
B	SUBX															
C	OR															
D	XOR															
E	AND															
F	MOV															

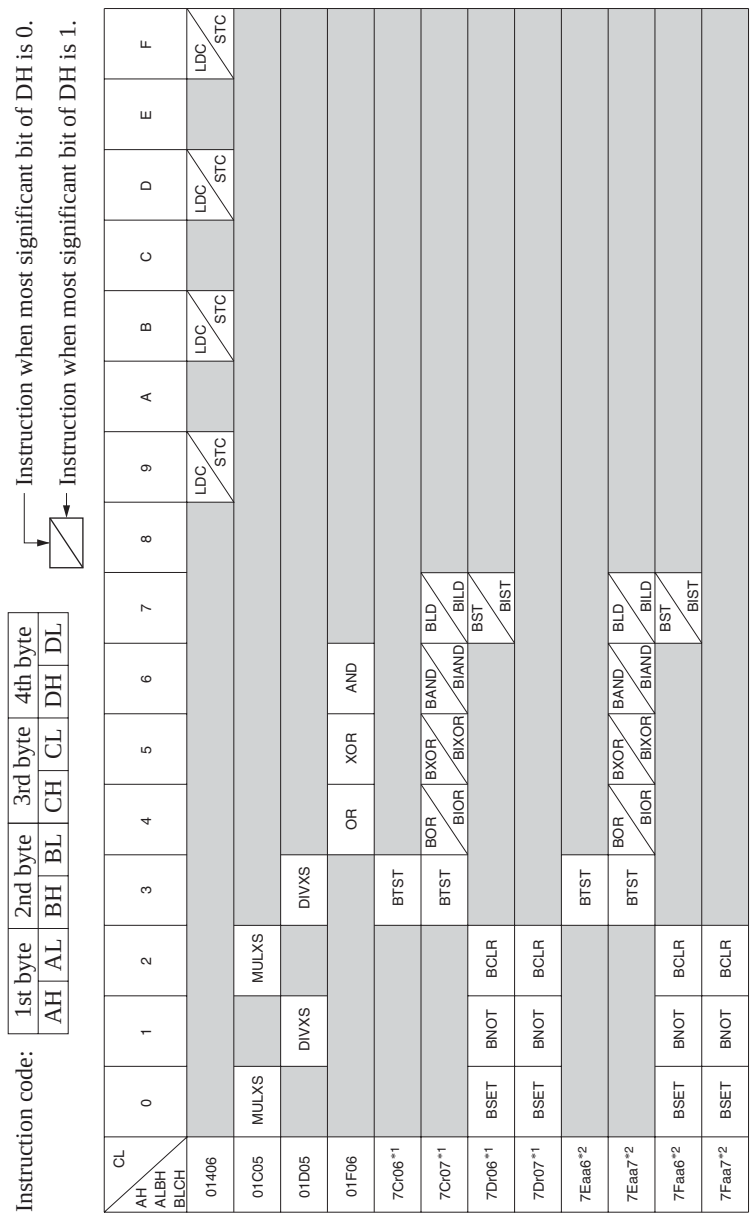
Table A.2 Operation Code Map (2)

Instruction code:

1st byte	2nd byte
AH	AL	BH	BL

AH AL \ BH	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
01	MOV				LDC/STC				SLEEP				Table A-2 (3)	Table A-2 (3)		Table A-2 (3)
0A	INC												ADD			
0B	ADDS					INC		INC		ADDS				INC		INC
0F	DAA												MOV			
10	SHLL			SHLL						SHAL		SHAL				
11	SHLR			SHLR						SHAR		SHAR				
12	ROTXL			ROTXL						ROTL		ROTL				
13	ROTXR			ROTXR						ROTR		ROTR				
17	NOT			NOT		EXTU		EXTU		NEG		NEG		EXTS		EXTS
1A	DEC												SUB			
1B	SUBS					DEC		DEC		SUB				DEC		DEC
1F	DAS												CMP			
58	BRA	BRN	BHI	BLS	BCC	BCS	BNE	BEQ	BVC	BVS	BPL	BMI	BGE	BLT	BGT	BLE
79	MOV	ADD	CMP	SUB	OR	XOR	AND									
7A	MOV	ADD	CMP	SUB	OR	XOR	AND									

Table A.2 Operation Code Map (3)



Notes: 1. r is the register designation field.
2. aa is the absolute address field.

A.3 Number of Execution States

The status of execution for each instruction of the H8/300H CPU and the method of calculating the number of states required for instruction execution are shown below. Table A.4 shows the number of cycles of each type occurring in each instruction, such as instruction fetch and data read/write. Table A.3 shows the number of states required for each cycle. The total number of states required for execution of an instruction can be calculated by the following expression:

$$\text{Execution states} = I \times S_I + J \times S_J + K \times S_K + L \times S_L + M \times S_M + N \times S_N$$

Examples: When instruction is fetched from on-chip ROM, and an on-chip RAM is accessed.

BSET #0, @FF00

From table A.4:
I = L = 2, J = K = M = N= 0

From table A.3:
$S_I = 2$, $S_L = 2$

Number of states required for execution = $2 \times 2 + 2 \times 2 = 8$

When instruction is fetched from on-chip ROM, branch address is read from on-chip ROM, and on-chip RAM is used for stack area.

JSR @@ 30

From table A.4:
I = 2, J = K = 1, L = M = N = 0

From table A.3:
$S_I = S_J = S_K = 2$

Number of states required for execution = $2 \times 2 + 1 \times 2 + 1 \times 2 = 8$

Table A.3 Number of Cycles in Each Instruction

Execution Status (Instruction Cycle)		Access Location	
		On-Chip Memory	On-Chip Peripheral Module
Instruction fetch	S_I	2	—
Branch address read	S_J		
Stack operation	S_K		
Byte data access	S_L		2 or 3*
Word data access	S_M		—
Internal operation	S_N	1	

Note: * Depends on which on-chip peripheral module is accessed. See section 19.1, Register Addresses (Address Order).

Table A.4 Number of Cycles in Each Instruction

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
ADD	ADD.B #xx:8, Rd	1					
	ADD.B Rs, Rd	1					
	ADD.W #xx:16, Rd	2					
	ADD.W Rs, Rd	1					
	ADD.L #xx:32, ERd	3					
	ADD.L ERs, ERd	1					
ADDS	ADDS #1/2/4, ERd	1					
ADDX	ADDX #xx:8, Rd	1					
	ADDX Rs, Rd	1					
AND	AND.B #xx:8, Rd	1					
	AND.B Rs, Rd	1					
	AND.W #xx:16, Rd	2					
	AND.W Rs, Rd	1					
	AND.L #xx:32, ERd	3					
	AND.L ERs, ERd	2					
ANDC	ANDC #xx:8, CCR	1					
BAND	BAND #xx:3, Rd	1					
	BAND #xx:3, @ERd	2			1		
	BAND #xx:3, @aa:8	2			1		
Bcc	BRA d:8 (BT d:8)	2					
	BRN d:8 (BF d:8)	2					
	BHI d:8	2					
	BLS d:8	2					
	BCC d:8 (BHS d:8)	2					
	BCS d:8 (BLO d:8)	2					
	BNE d:8	2					
	BEQ d:8	2					
	BVC d:8	2					
	BVS d:8	2					
	BPL d:8	2					
	BMI d:8	2					
	BGE d:8	2					

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
Bcc	BLT d:8	2					
	BGT d:8	2					
	BLE d:8	2					
	BRA d:16(BT d:16)	2					2
	BRN d:16(BF d:16)	2					2
	BHI d:16	2					2
	BLS d:16	2					2
	BCC d:16(BHS d:16)	2					2
	BCS d:16(BLO d:16)	2					2
	BNE d:16	2					2
	BEQ d:16	2					2
	BVC d:16	2					2
	BVS d:16	2					2
	BPL d:16	2					2
	BMI d:16	2					2
	BGE d:16	2					2
	BLT d:16	2					2
	BGT d:16	2					2
	BLE d:16	2					2
BCLR	BCLR #xx:3, Rd	1					
	BCLR #xx:3, @ERd	2			2		
	BCLR #xx:3, @aa:8	2			2		
	BCLR Rn, Rd	1					
	BCLR Rn, @ERd	2			2		
	BCLR Rn, @aa:8	2			2		
BIAND	BIAND #xx:3, Rd	1					
	BIAND #xx:3, @ERd	2			1		
	BIAND #xx:3, @aa:8	2			1		
BILD	BILD #xx:3, Rd	1					
	BILD #xx:3, @ERd	2			1		
	BILD #xx:3, @aa:8	2			1		

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
BIOR	BIOR #xx:8, Rd	1					
	BIOR #xx:8, @ERd	2			1		
	BIOR #xx:8, @aa:8	2			1		
BIST	BIST #xx:3, Rd	1					
	BIST #xx:3, @ERd	2			2		
	BIST #xx:3, @aa:8	2			2		
BIXOR	BIXOR #xx:3, Rd	1					
	BIXOR #xx:3, @ERd	2			1		
	BIXOR #xx:3, @aa:8	2			1		
BLD	BLD #xx:3, Rd	1					
	BLD #xx:3, @ERd	2			1		
	BLD #xx:3, @aa:8	2			1		
BNOT	BNOT #xx:3, Rd	1					
	BNOT #xx:3, @ERd	2			2		
	BNOT #xx:3, @aa:8	2			2		
	BNOT Rn, Rd	1					
	BNOT Rn, @ERd	2			2		
	BNOT Rn, @aa:8	2			2		
BOR	BOR #xx:3, Rd	1					
	BOR #xx:3, @ERd	2			1		
	BOR #xx:3, @aa:8	2			1		
BSET	BSET #xx:3, Rd	1					
	BSET #xx:3, @ERd	2			2		
	BSET #xx:3, @aa:8	2			2		
	BSET Rn, Rd	1					
	BSET Rn, @ERd	2			2		
	BSET Rn, @aa:8	2			2		
BSR	BSR d:8	2		1			
	BSR d:16	2		1			2
BST	BST #xx:3, Rd	1					
	BST #xx:3, @ERd	2			2		
	BST #xx:3, @aa:8	2			2		

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
BTST	BTST #xx:3, Rd	1					
	BTST #xx:3, @ERd	2			1		
	BTST #xx:3, @aa:8	2			1		
	BTST Rn, Rd	1					
	BTST Rn, @ERd	2			1		
	BTST Rn, @aa:8	2			1		
BXOR	BXOR #xx:3, Rd	1					
	BXOR #xx:3, @ERd	2			1		
	BXOR #xx:3, @aa:8	2			1		
CMP	CMP.B #xx:8, Rd	1					
	CMP.B Rs, Rd	1					
	CMP.W #xx:16, Rd	2					
	CMP.W Rs, Rd	1					
	CMP.L #xx:32, ERd	3					
	CMP.L ERs, ERd	1					
DAA	DAA Rd	1					
DAS	DAS Rd	1					
DEC	DEC.B Rd	1					
	DEC.W #1/2, Rd	1					
	DEC.L #1/2, ERd	1					
DUVXS	DIVXS.B Rs, Rd	2					12
	DIVXS.W Rs, ERd	2					20
DIVXU	DIVXU.B Rs, Rd	1					12
	DIVXU.W Rs, ERd	1					20
EEPMOV	EEPMOV.B	2			2n+2*[1]		
	EEPMOV.W	2			2n+2*[1]		
EXTS	EXTS.W Rd	1					
	EXTS.L ERd	1					
EXTU	EXTU.W Rd	1					
	EXTU.L ERd	1					

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
INC	INC.B Rd	1					
	INC.W #1/2, Rd	1					
	INC.L #1/2, ERd	1					
JMP	JMP @ERn	2					
	JMP @aa:24	2					2
	JMP @@aa:8	2	1				2
JSR	JSR @ERn	2		1			
	JSR @aa:24	2		1			2
	JSR @@aa:8	2	1	1			
LDC	LDC #xx:8, CCR	1					
	LDC Rs, CCR	1					
	LDC@ERs, CCR	2				1	
	LDC@(d:16, ERs), CCR	3				1	
	LDC@(d:24,ERs), CCR	5				1	
	LDC@ERs+, CCR	2				1	2
	LDC@aa:16, CCR	3				1	
	LDC@aa:24, CCR	4				1	
MOV	MOV.B #xx:8, Rd	1					
	MOV.B Rs, Rd	1					
	MOV.B @ERs, Rd	1			1		
	MOV.B @(d:16, ERs), Rd	2			1		
	MOV.B @(d:24, ERs), Rd	4			1		
	MOV.B @ERs+, Rd	1			1		2
	MOV.B @aa:8, Rd	1			1		
	MOV.B @aa:16, Rd	2			1		
	MOV.B @aa:24, Rd	3			1		
	MOV.B Rs, @Erd	1			1		
	MOV.B Rs, @(d:16, ERd)	2			1		
	MOV.B Rs, @(d:24, ERd)	4			1		
	MOV.B Rs, @-ERd	1			1		2
	MOV.B Rs, @aa:8	1			1		

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
MOV	MOV.B Rs, @aa:16	2			1		
	MOV.B Rs, @aa:24	3			1		
	MOV.W #xx:16, Rd	2					
	MOV.W Rs, Rd	1					
	MOV.W @ERs, Rd	1				1	
	MOV.W @(d:16,ERs), Rd	2				1	
	MOV.W @(d:24,ERs), Rd	4				1	
	MOV.W @ERs+, Rd	1				1	2
	MOV.W @aa:16, Rd	2				1	
	MOV.W @aa:24, Rd	3				1	
	MOV.W Rs, @ERd	1				1	
	MOV.W Rs, @(d:16,ERd)	2				1	
	MOV.W Rs, @(d:24,ERd)	4				1	
MOV	MOV.W Rs, @-ERd	1				1	2
	MOV.W Rs, @aa:16	2				1	
	MOV.W Rs, @aa:24	3				1	
	MOV.L #xx:32, ERd	3					
	MOV.L ERs, ERd	1					
	MOV.L @ERs, ERd	2				2	
	MOV.L @(d:16,ERs), ERd	3				2	
	MOV.L @(d:24,ERs), ERd	5				2	
	MOV.L @ERs+, ERd	2				2	2
	MOV.L @aa:16, ERd	3				2	
	MOV.L @aa:24, ERd	4				2	
	MOV.L ERs,@ERd	2				2	
	MOV.L ERs, @(d:16,ERd)	3				2	
	MOV.L ERs, @(d:24,ERd)	5				2	
	MOV.L ERs, @-ERd	2				2	2
	MOV.L ERs, @aa:16	3				2	
	MOV.L ERs, @aa:24	4				2	
MOVFPE	MOVFPE @aa:16, Rd*[2]	2			1		
MOVTPE	MOVTPE Rs,@aa:16*[2]	2			1		

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
MULXS	MULXS.B Rs, Rd	2					12
	MULXS.W Rs, ERd	2					20
MULXU	MULXU.B Rs, Rd	1					12
	MULXU.W Rs, ERd	1					20
NEG	NEG.B Rd	1					
	NEG.W Rd	1					
	NEG.L ERd	1					
NOP	NOP	1					
NOT	NOT.B Rd	1					
	NOT.W Rd	1					
	NOT.L ERd	1					
OR	OR.B #xx:8, Rd	1					
	OR.B Rs, Rd	1					
	OR.W #xx:16, Rd	2					
	OR.W Rs, Rd	1					
	OR.L #xx:32, ERd	3					
	OR.L ERs, ERd	2					
ORC	ORC #xx:8, CCR	1					
POP	POP.W Rn	1				1	2
	POP.L ERn	2				2	2
PUSH	PUSH.W Rn	1				1	2
	PUSH.L ERn	2				2	2
ROTL	ROTL.B Rd	1					
	ROTL.W Rd	1					
	ROTL.L ERd	1					
ROTR	ROTR.B Rd	1					
	ROTR.W Rd	1					
	ROTR.L ERd	1					
ROTXL	ROTXL.B Rd	1					
	ROTXL.W Rd	1					
	ROTXL.L ERd	1					

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
ROTXR	ROTXR.B Rd	1					
	ROTXR.W Rd	1					
	ROTXR.L ERd	1					
RTE	RTE	2		2			2
RTS	RTS	2		1			2
SHAL	SHAL.B Rd	1					
	SHAL.W Rd	1					
	SHAL.L ERd	1					
SHAR	SHAR.B Rd	1					
	SHAR.W Rd	1					
	SHAR.L ERd	1					
SHLL	SHLL.B Rd	1					
	SHLL.W Rd	1					
	SHLL.L ERd	1					
SHLR	SHLR.B Rd	1					
	SHLR.W Rd	1					
	SHLR.L ERd	1					
SLEEP	SLEEP	1					
STC	STC CCR, Rd	1					
	STC CCR, @ERd	2				1	
	STC CCR, @(d:16,ERd)	3				1	
	STC CCR, @(d:24,ERd)	5				1	
	STC CCR,@-ERd	2				1	2
	STC CCR, @aa:16	3				1	
	STC CCR, @aa:24	4				1	
SUB	SUB.B Rs, Rd	1					
	SUB.W #xx:16, Rd	2					
	SUB.W Rs, Rd	1					
	SUB.L #xx:32, ERd	3					
	SUB.L ERs, ERd	1					
SUBS	SUBS #1/2/4, ERd	1					

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
SUBX	SUBX #xx:8, Rd	1					
	SUBX. Rs, Rd	1					
TRAPA	TRAPA #xx:2	2	1	2			4
XOR	XOR.B #xx:8, Rd	1					
	XOR.B Rs, Rd	1					
	XOR.W #xx:16, Rd	2					
	XOR.W Rs, Rd	1					
	XOR.L #xx:32, ERd	3					
	XOR.L ERs, ERd	2					
XORC	XORC #xx:8, CCR	1					

Notes: 1. n:specified value in R4L and R4. The source and destination operands are accessed n+1 times respectively.

2. Cannot be used in this LSI.

A.4 Combinations of Instructions and Addressing Modes

Table A.5 Combinations of Instructions and Addressing Modes

Functions	Instructions	Addressing Mode												
		#xx	Rn	@ERn	@(d:16.ERn)	@(d:24.ERn)	@ERn+/@ERn	@aa:8	@aa:16	@aa:24	@(d:8.PC)	@(d:16.PC)	@@aa:8	|
Data transfer instructions	MOV	BWL	BWL	BWL	BWL	BWL	BWL	B	BWL	BWL	—	—	—	—
	POP, PUSH	—	—	—	—	—	—	—	—	—	—	—	—	WL
	MOVFPE, MOVTPE	—	—	—	—	—	—	—	—	—	—	—	—	—
Arithmetic operations	ADD, CMP	BWL	BWL	—	—	—	—	—	—	—	—	—	—	—
	SUB	WL	BWL	—	—	—	—	—	—	—	—	—	—	—
	ADDX, SUBX	B	B	—	—	—	—	—	—	—	—	—	—	—
	ADDS, SUBS	—	L	—	—	—	—	—	—	—	—	—	—	—
	INC, DEC	—	BWL	—	—	—	—	—	—	—	—	—	—	—
	DAA, DAS	—	B	—	—	—	—	—	—	—	—	—	—	—
	MULXU, MULXS, DIVXU, DIVXS	—	BW	—	—	—	—	—	—	—	—	—	—	—
	NEG	—	BWL	—	—	—	—	—	—	—	—	—	—	—
	EXTU, EXTS	—	WL	—	—	—	—	—	—	—	—	—	—	—
Logical operations	AND, OR, XOR	—	BWL	—	—	—	—	—	—	—	—	—	—	—
	NOT	—	BWL	—	—	—	—	—	—	—	—	—	—	—
Shift operations		—	BWL	—	—	—	—	—	—	—	—	—	—	—
Bit manipulations		—	B	B	—	—	—	B	—	—	—	—	—	—
Branching instructions	BCC, BSR	—	—	—	—	—	—	—	—	—	—	—	—	—
	JMP, JSR	—	—	◯	—	—	—	—	—	—	◯	◯	—	—
	RTS	—	—	—	—	—	—	—	—	◯	—	—	◯	—
System control instructions	TRAPA	—	—	—	—	—	—	—	—	—	—	—	—	◯
	RTE	—	—	—	—	—	—	—	—	—	—	—	—	◯
	SLEEP	—	—	—	—	—	—	—	—	—	—	—	—	◯
	LDC	B	B	W	W	W	W	—	W	W	—	—	—	◯
	STC	—	B	W	W	W	W	—	W	W	—	—	—	—
	ANDC, ORC, XORC	B	—	—	—	—	—	—	—	—	—	—	—	—
	NOP	—	—	—	—	—	—	—	—	—	—	—	—	◯
Block data transfer instructions		—	—	—	—	—	—	—	—	—	—	—	—	BW

Appendix B I/O Port Block Diagrams

B.1 I/O Port Block

$\overline{\text{RES}}$ goes low in a reset, and $\overline{\text{SBY}}$ goes low in a reset and in standby mode.

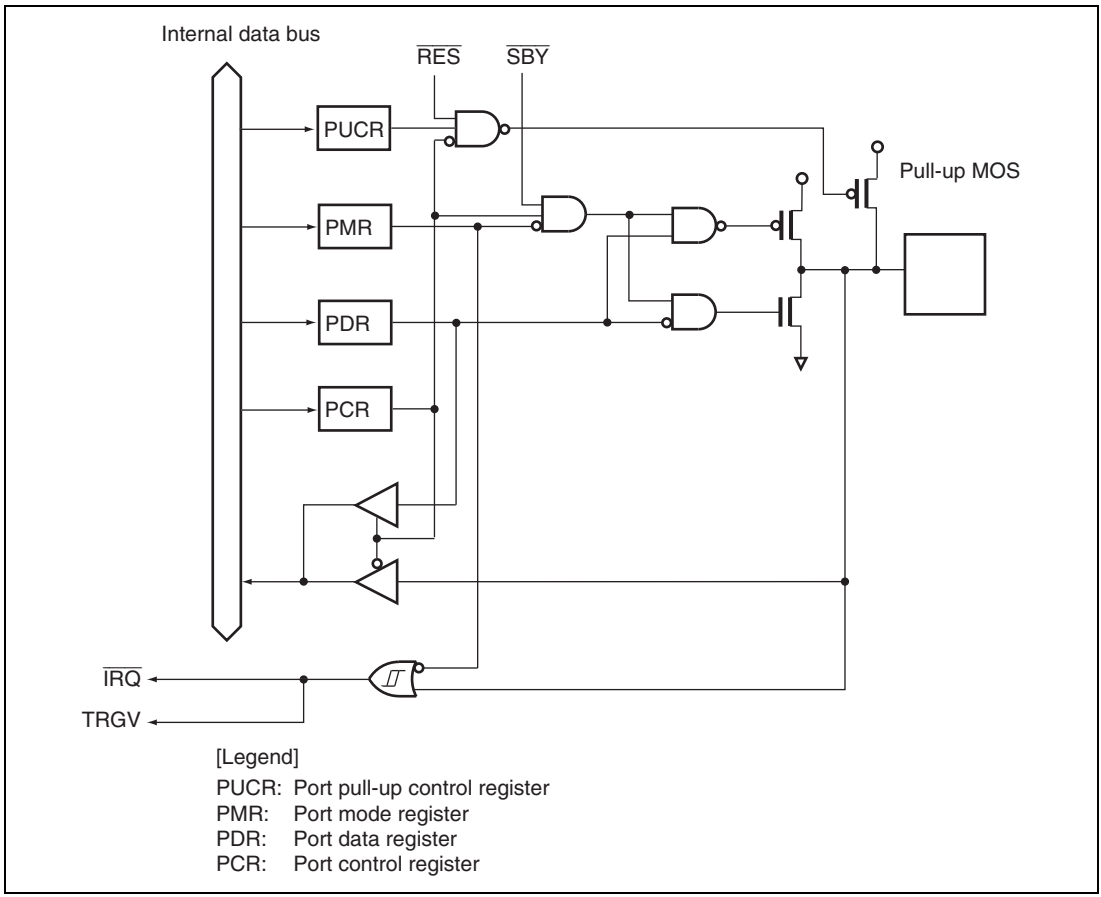


Figure B.1 Port 1 Block Diagram (P17)

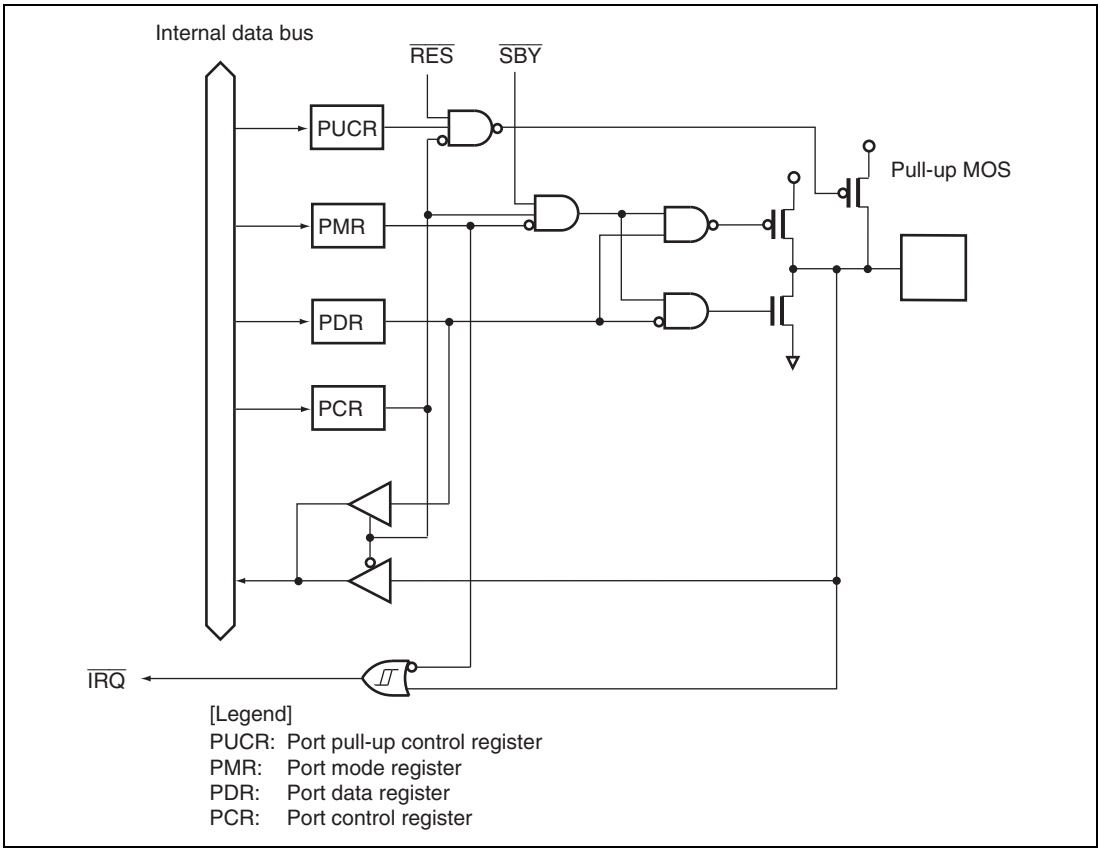


Figure B.2 Port 1 Block Diagram (P16 to P14)

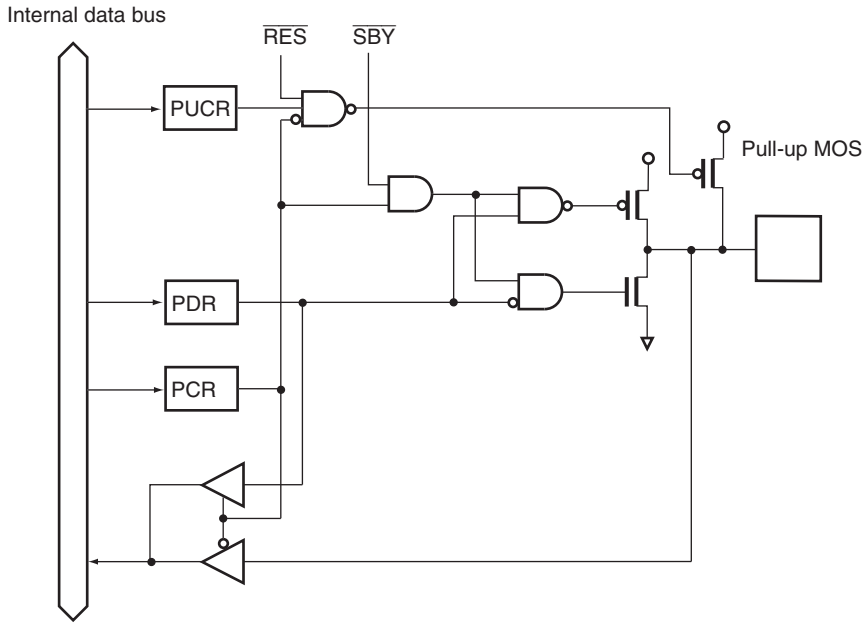


Figure B.3 Port 1 Block Diagram (P12, P11)

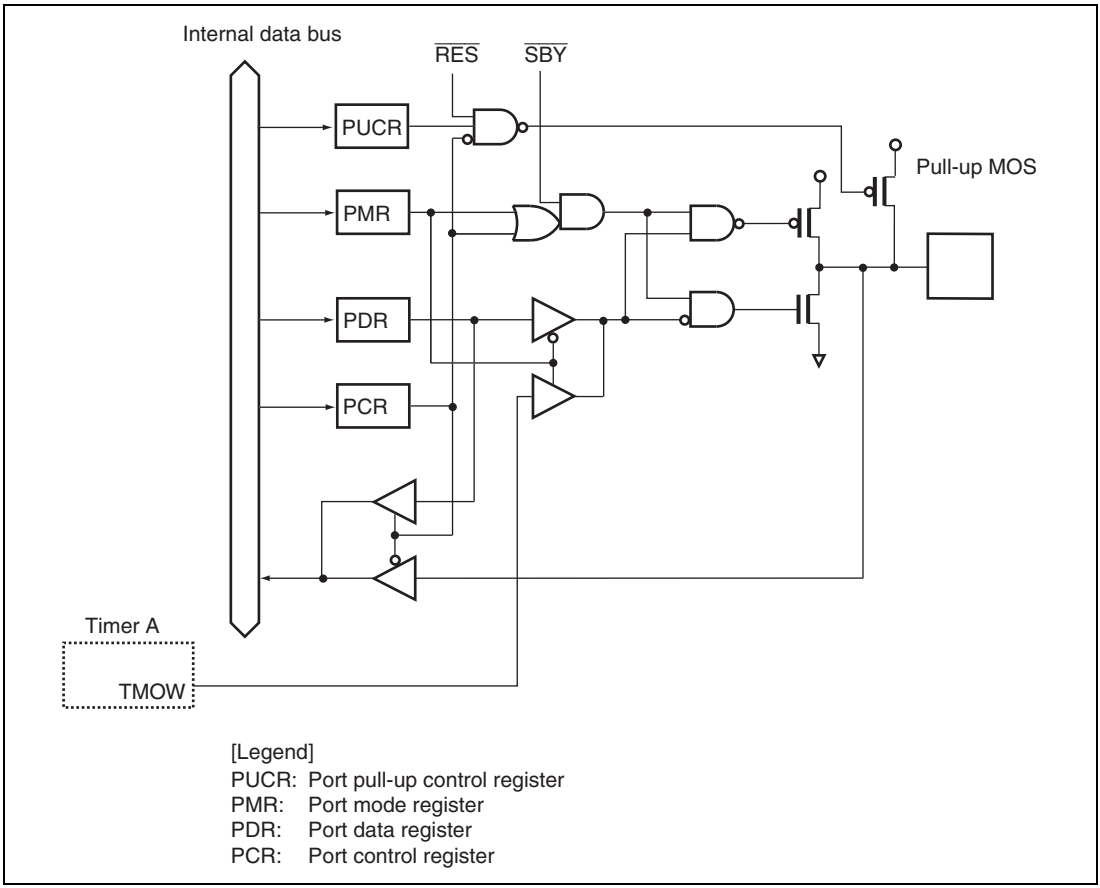


Figure B.4 Port 1 Block Diagram (P10)

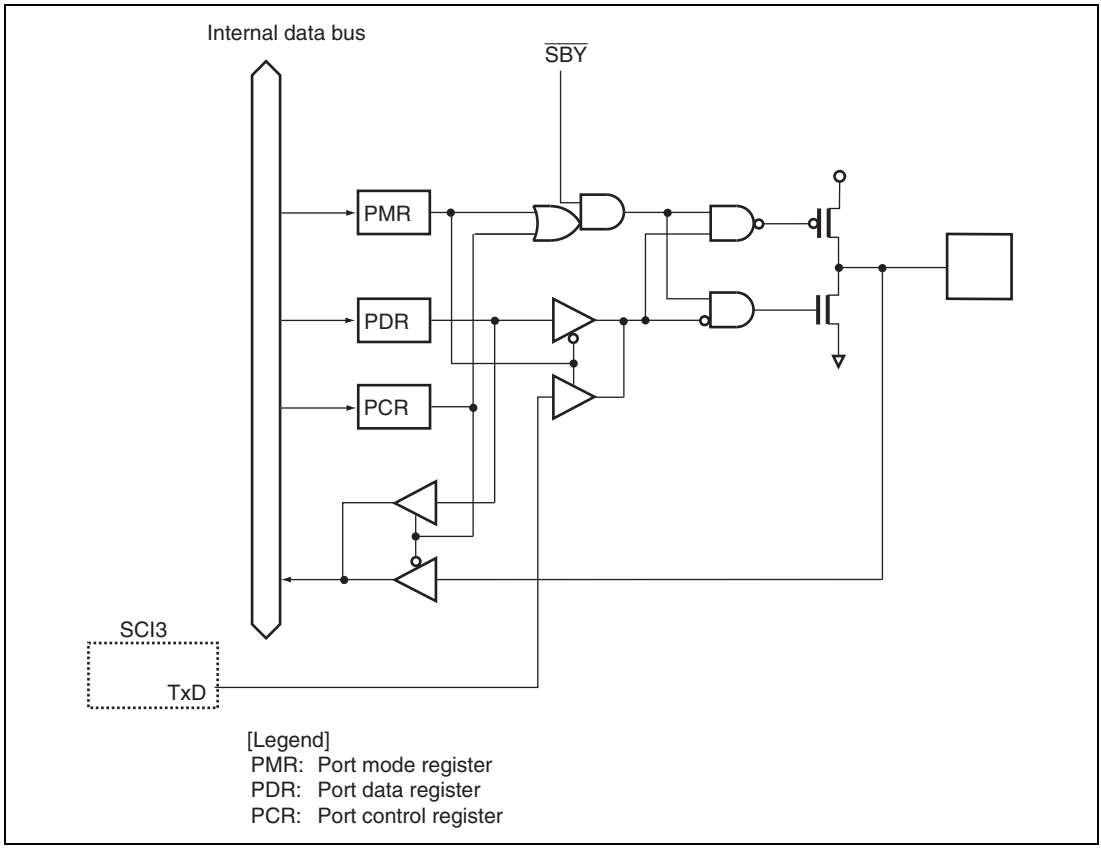


Figure B.5 Port 2 Block Diagram (P22)

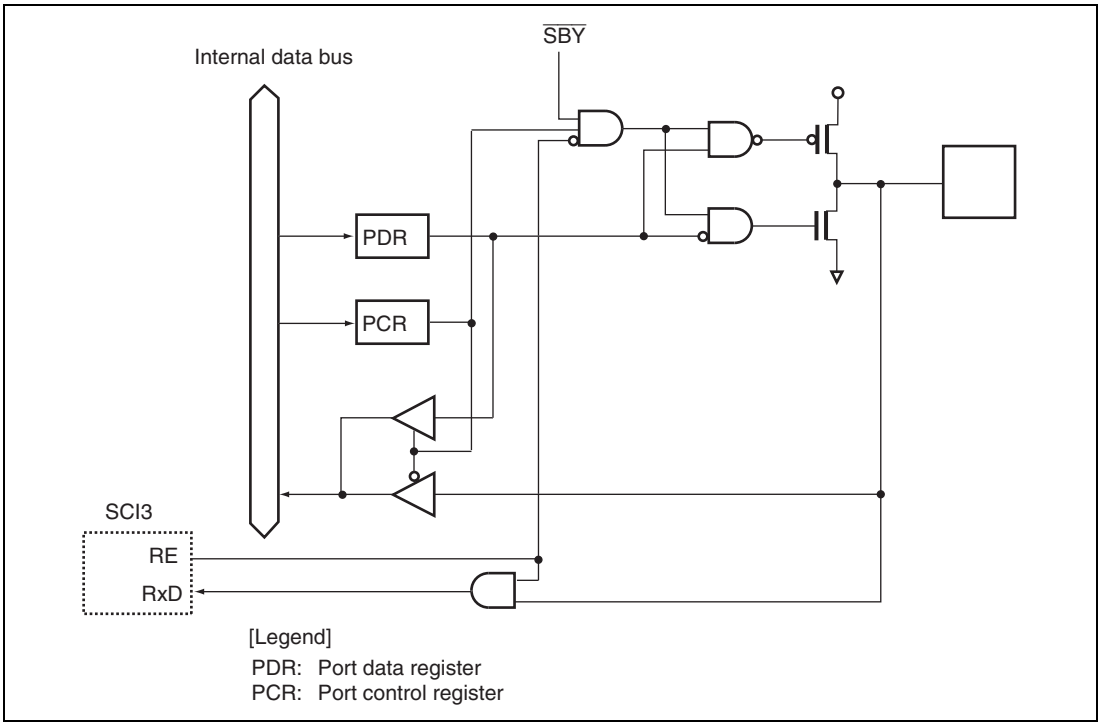


Figure B.6 Port 2 Block Diagram (P21)

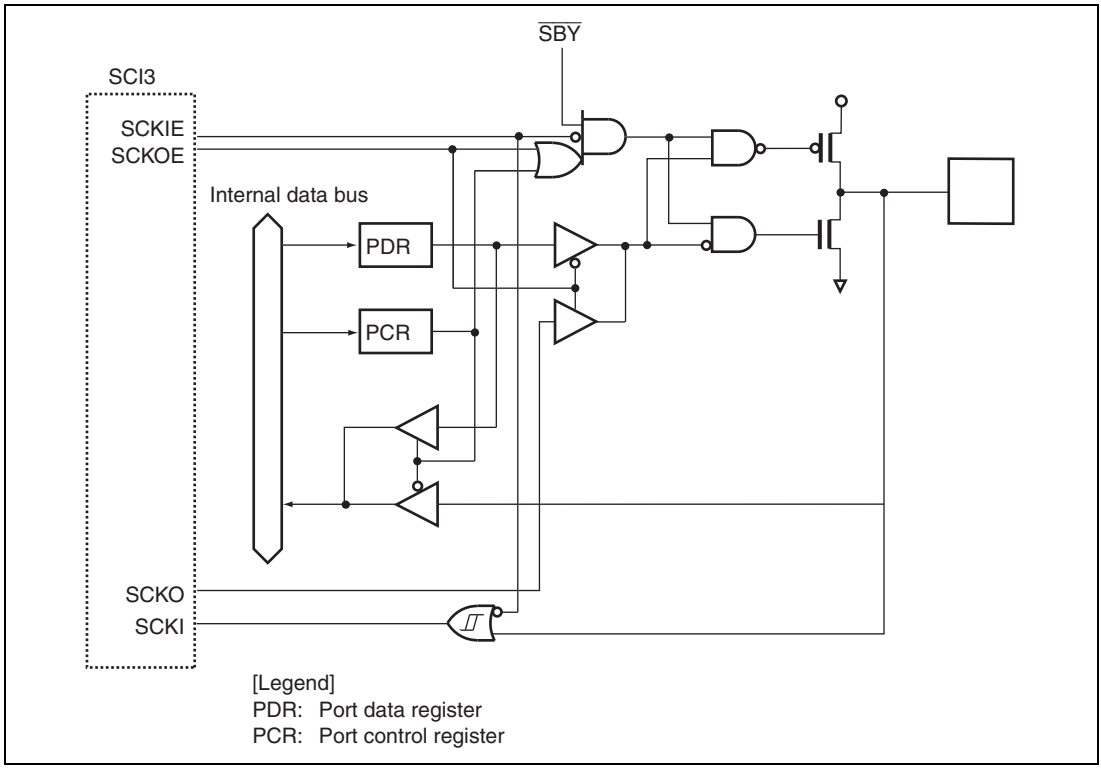


Figure B.7 Port 2 Block Diagram (P20)

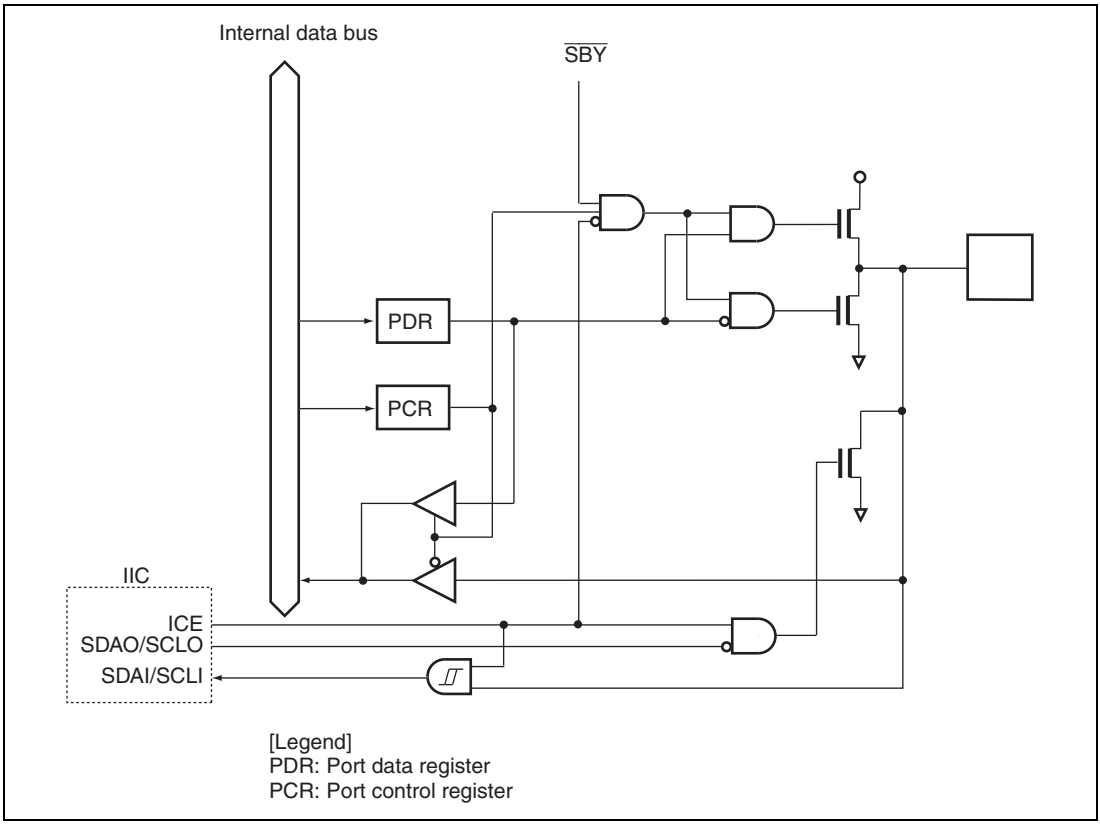


Figure B.8 Port 5 Block Diagram (P57, P56)*

Note: * Not included in H8/3664N.

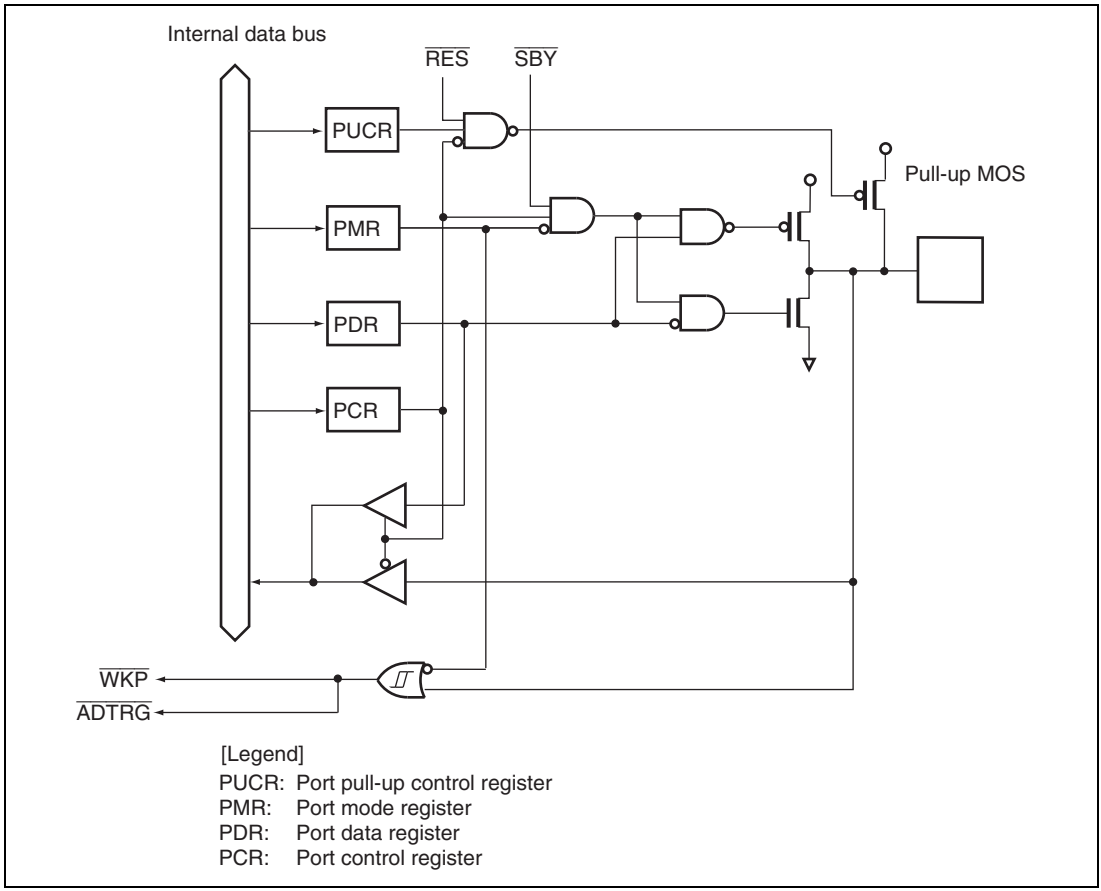


Figure B.9 Port 5 Block Diagram (P55)

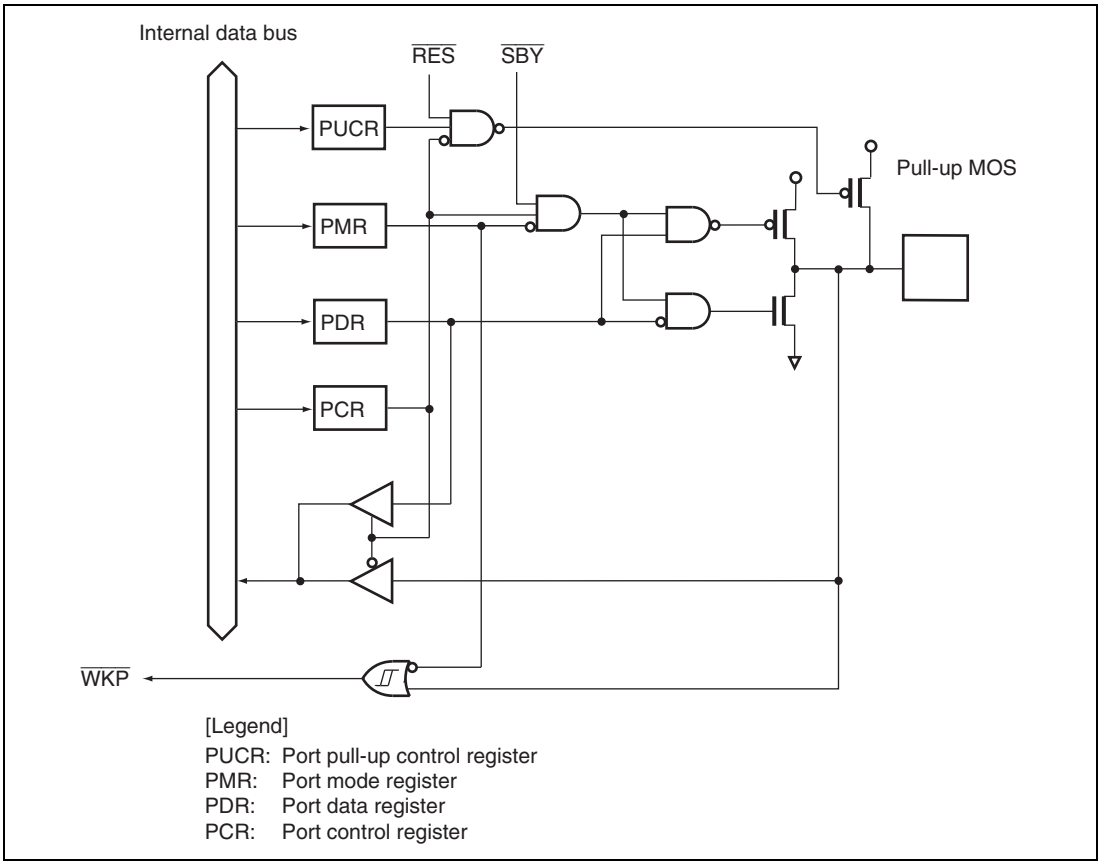


Figure B.10 Port 5 Block Diagram (P54 to P50)

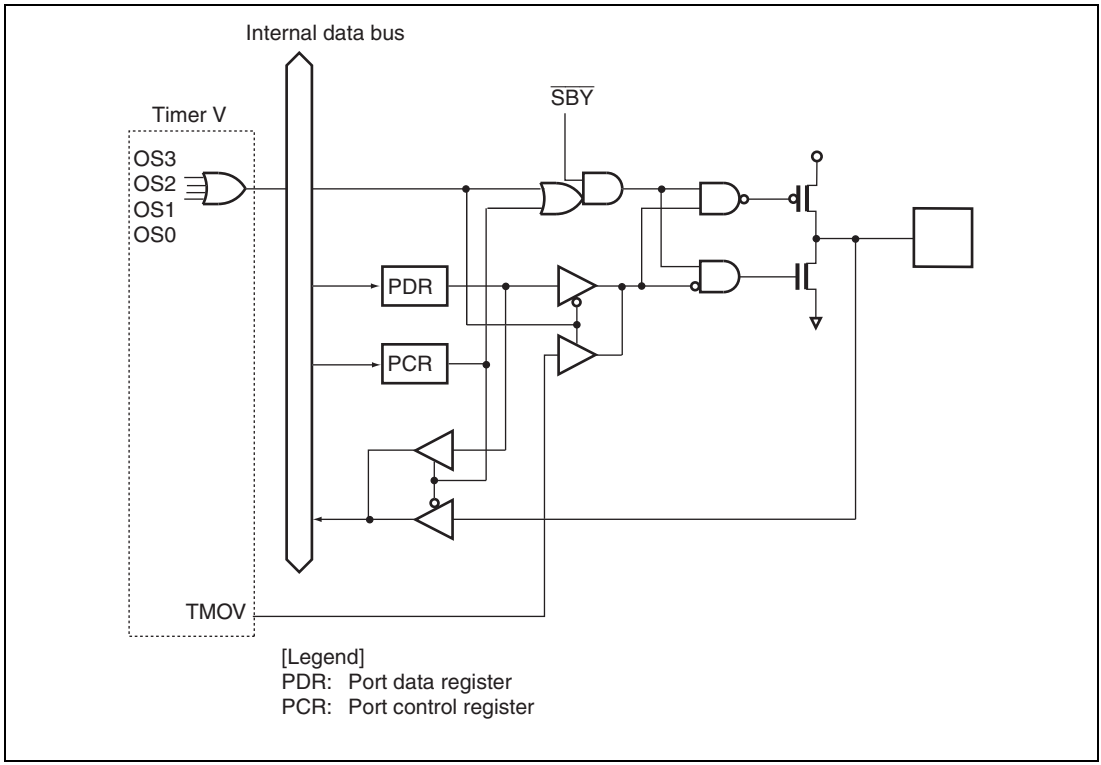


Figure B.11 Port 7 Block Diagram (P76)

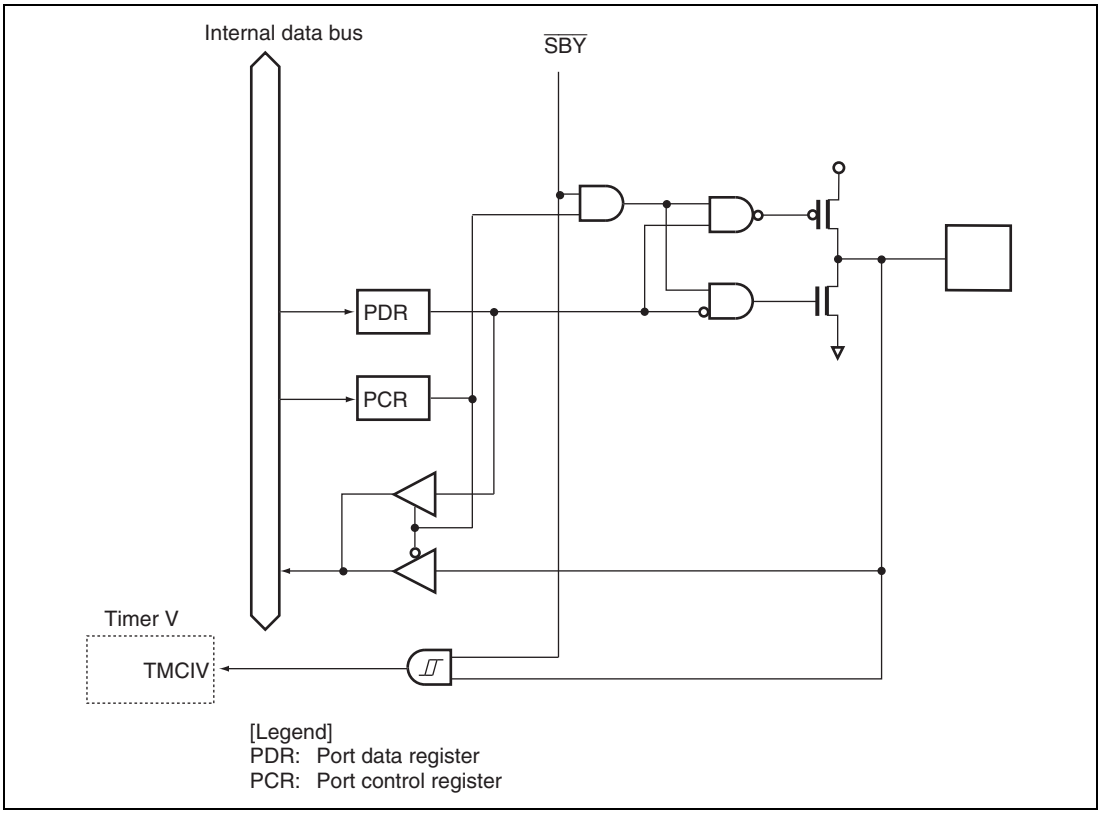


Figure B.12 Port 7 Block Diagram (P75)

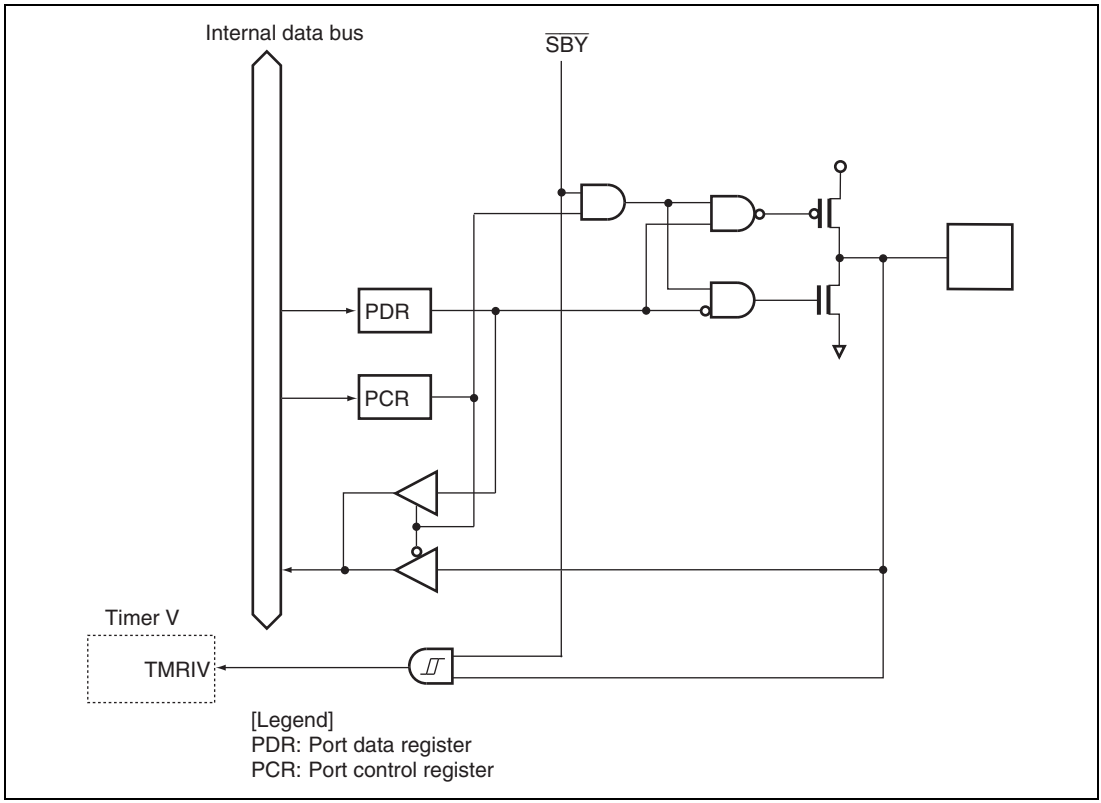


Figure B.13 Port 7 Block Diagram (P74)

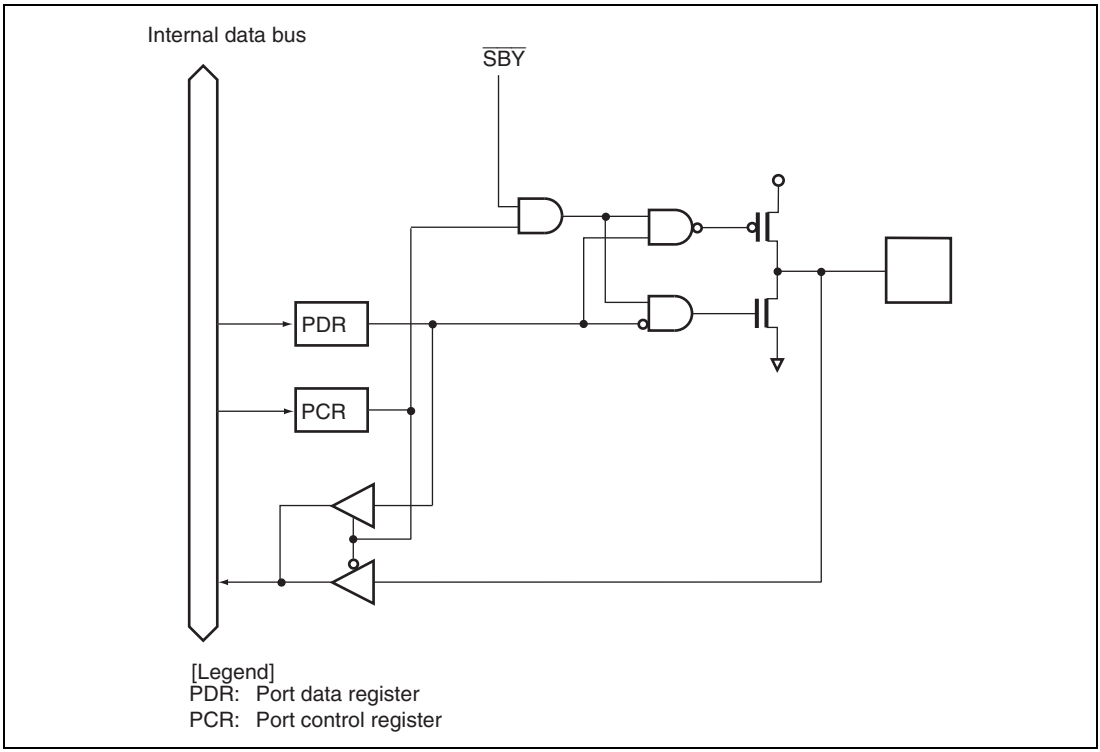


Figure B.14 Port 8 Block Diagram (P87 to P85)

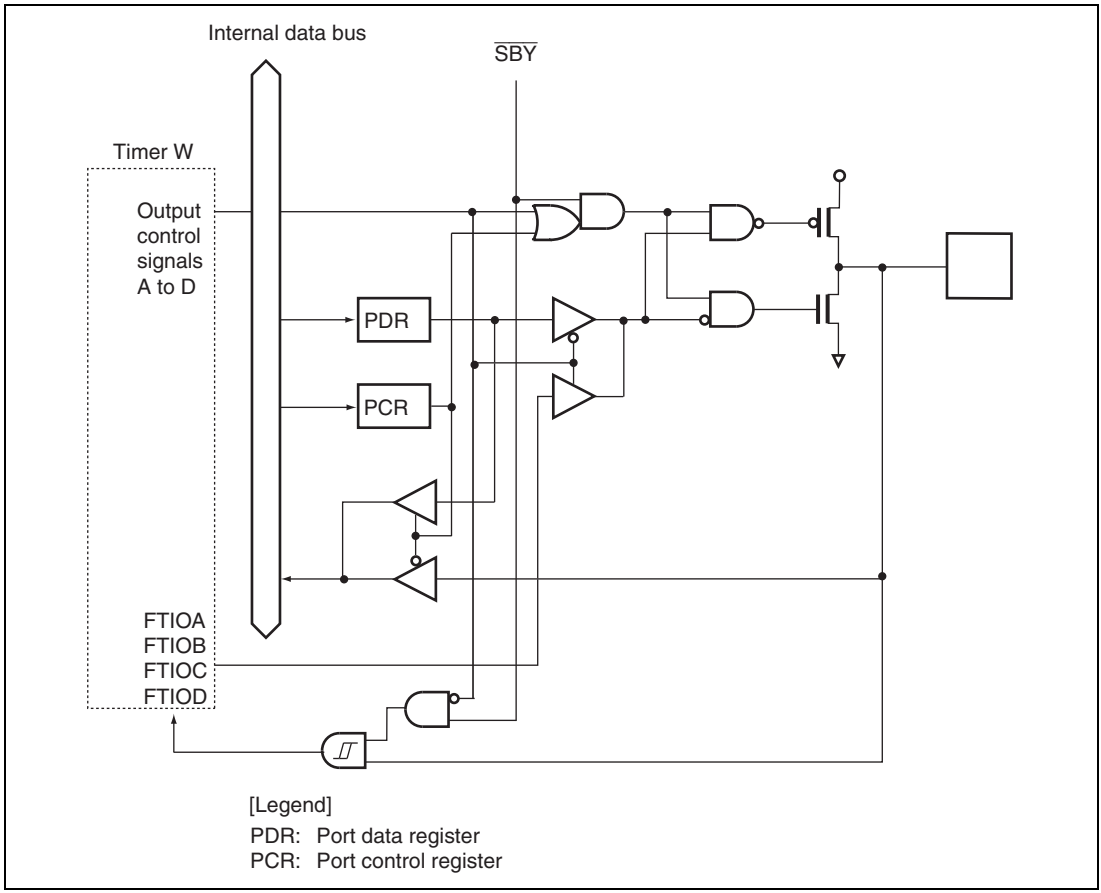


Figure B.15 Port 8 Block Diagram (P84 to P81)

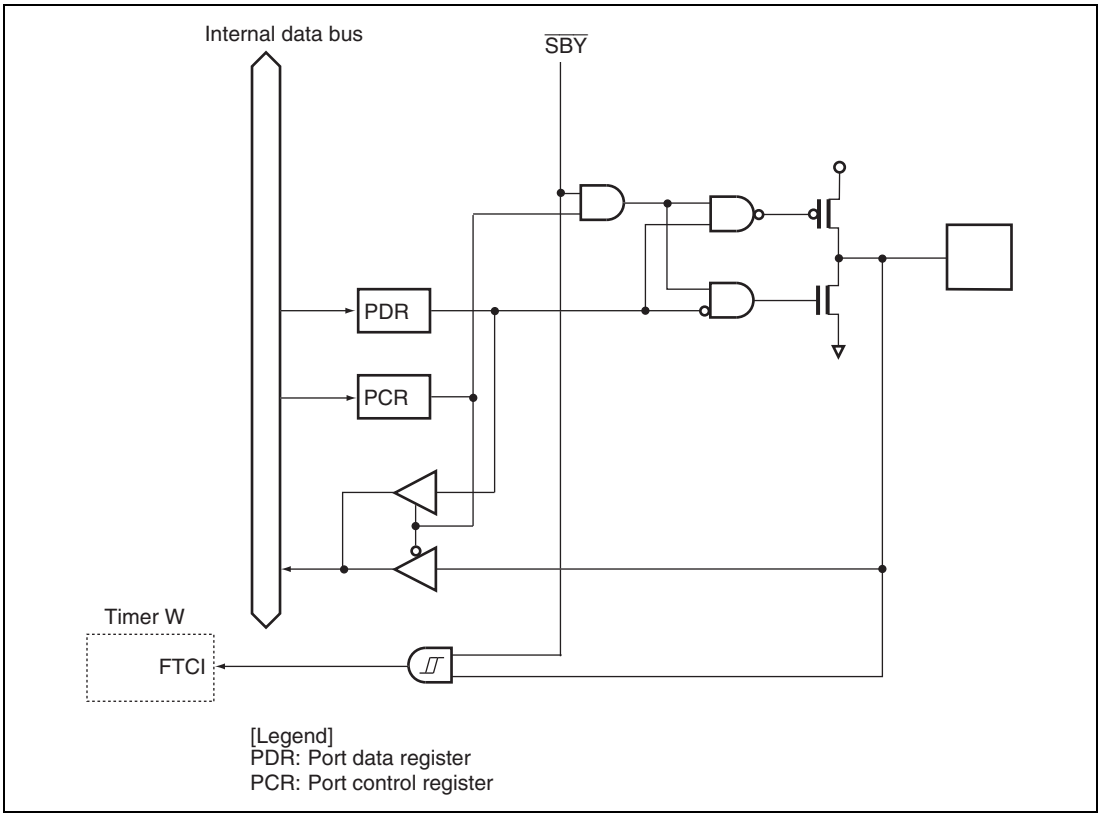


Figure B.16 Port 8 Block Diagram (P80)

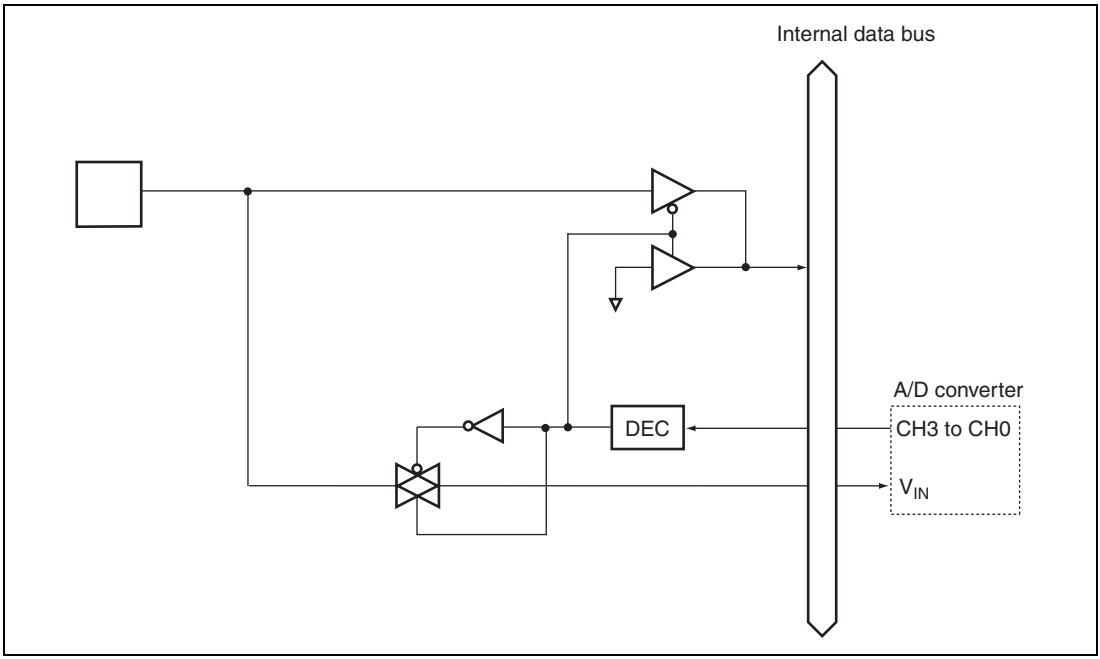


Figure B.17 Port B Block Diagram (PB7 to PB0)

B.2 Port States in Each Operating State

Port	Reset	Sleep	Subsleep	Standby	Subactive	Active
P17 to P14, P12 to P10	High impedance	Retained	Retained	High impedance*	Functioning	Functioning
P22 to P20	High impedance	Retained	Retained	High impedance*	Functioning	Functioning
P57 to P50 (P55 to P50 for H8/3664N)	High impedance	Retained	Retained	High impedance	Functioning	Functioning
P76 to P74	High impedance	Retained	Retained	High impedance	Functioning	Functioning
P87 to P80	High impedance	Retained	Retained	High impedance	Functioning	Functioning
PB7 to PB0	High impedance	High impedance	High impedance	High impedance	High impedance	High impedance

Note: * High level output when the pull-up MOS is in on state.

Appendix C Product Code Lineup

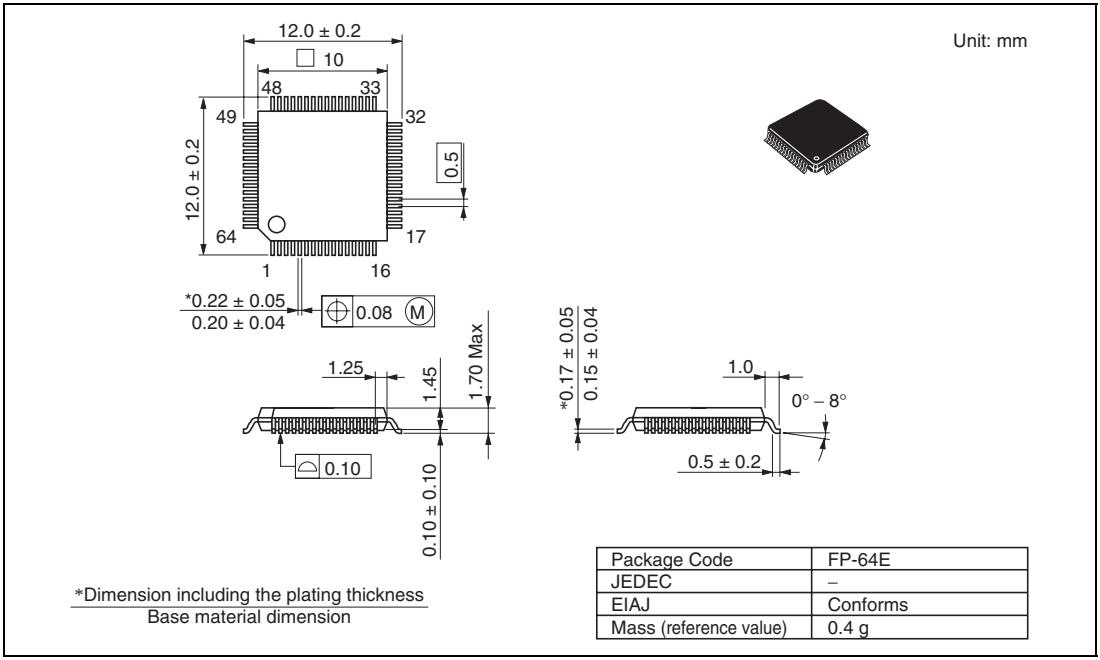
Product Type			Product Code	Model Marking	Package Code
H8/3664	Flash memory version with EEPROM	Standard product	HD64N3664FP	HD64N3664FP	LQFP-64 (FP-64E)
	Flash memory version	Standard product	HD64F3664FP	HD64F3664FP	LQFP-64 (FP-64E)
			HD64F3664H	HD64F3664H	QFP-64 (FP-64A)
			HD64F3664FX	HD64F3664FX	LQFP-48 (FP-48F)
			HD64F3664FY	HD64F3664FY	LQFP-48 (FP-48B)
			HD64F3664BP	HD64F3664BP	SDIP-42 (DP-42S)
	Mask ROM version	Standard product	HD6433664FP	HD6433664 (***) FP	LQFP-64 (FP-64E)
			HD6433664H	HD6433664 (***) H	QFP-64 (FP-64A)
			HD6433664FX	HD6433664 (***) FX	LQFP-48 (FP-48F)
			HD6433664FY	HD6433664 (***) FY	LQFP-48 (FP-48B)
			HD6433664BP	HD6433664 (***) BP	SDIP-42 (DP-42S)
H8/3663	Mask ROM version	Standard product	HD6433663FP	HD6433663 (***) FP	LQFP-64 (FP-64E)
			HD6433663H	HD6433663 (***) H	QFP-64 (FP-64A)
			HD6433663FX	HD6433663 (***) FX	LQFP-48 (FP-48F)
			HD6433663FY	HD6433663 (***) FY	LQFP-48 (FP-48B)
			HD6433663BP	HD6433663 (***) BP	SDIP-42 (DP-42S)
H8/3662	Mask ROM version	Standard product	HD6433662FP	HD6433662 (***) FP	LQFP-64 (FP-64E)
			HD6433662H	HD6433662 (***) H	QFP-64 (FP-64A)
			HD6433662FX	HD6433662 (***) FX	LQFP-48 (FP-48F)
			HD6433662FY	HD6433662 (***) FY	LQFP-48 (FP-48B)
			HD6433662BP	HD6433662 (***) BP	SDIP-42 (DP-42S)
H8/3661	Mask ROM version	Standard product	HD6433661FP	HD6433661 (***) FP	LQFP-64 (FP-64E)
			HD6433661H	HD6433661 (***) H	QFP-64 (FP-64A)
			HD6433661FX	HD6433661 (***) FX	LQFP-48 (FP-48F)
			HD6433661FY	HD6433661 (***) FY	LQFP-48 (FP-48B)
			HD6433661BP	HD6433661 (***) BP	SDIP-42 (DP-42S)

Product Type			Product Code	Model Marking	Package Code
H8/3660	Mask ROM version	Standard product	HD6433660FP	HD6433660 (***) FP	LQFP-64 (FP-64E)
			HD6433660H	HD6433660 (***) H	QFP-64 (FP-64A)
			HD6433660FX	HD6433660 (***) FX	LQFP-48 (FP-48F)
			HD6433660FY	HD6433660 (***) FY	LQFP-48 (FP-48B)
			HD6433660BP	HD6433660 (***) BP	SDIP-42 (DP-42S)

[Legend]
(***): ROM code

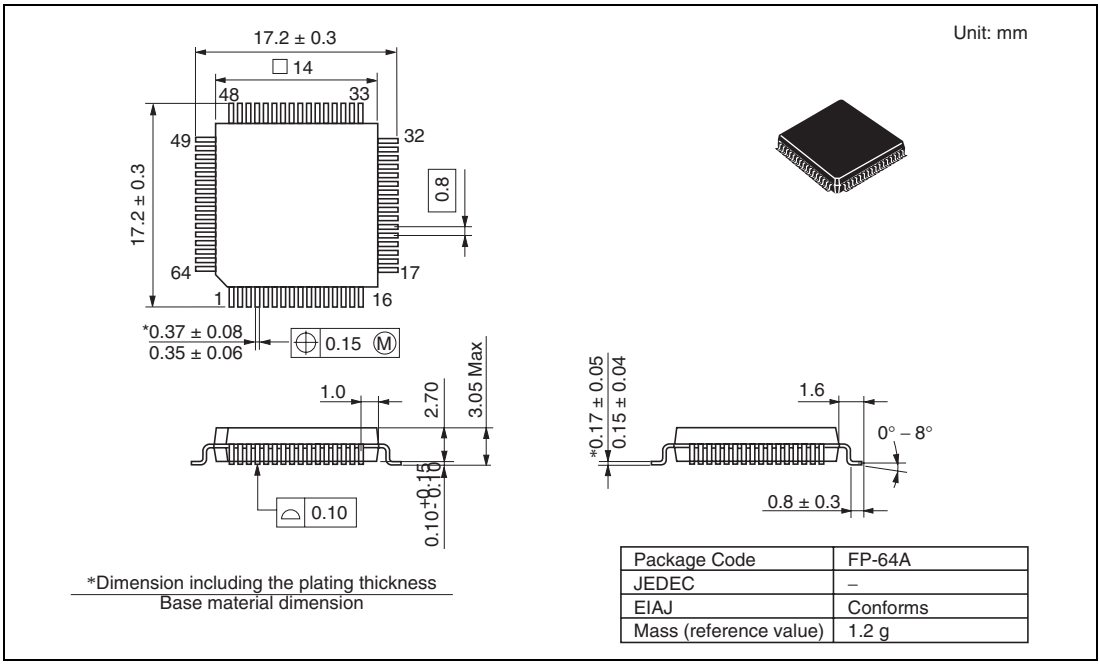
Appendix D Package Dimensions

The package dimensions that are shows in the Renesas Semiconductor Packages Data Book have priority.



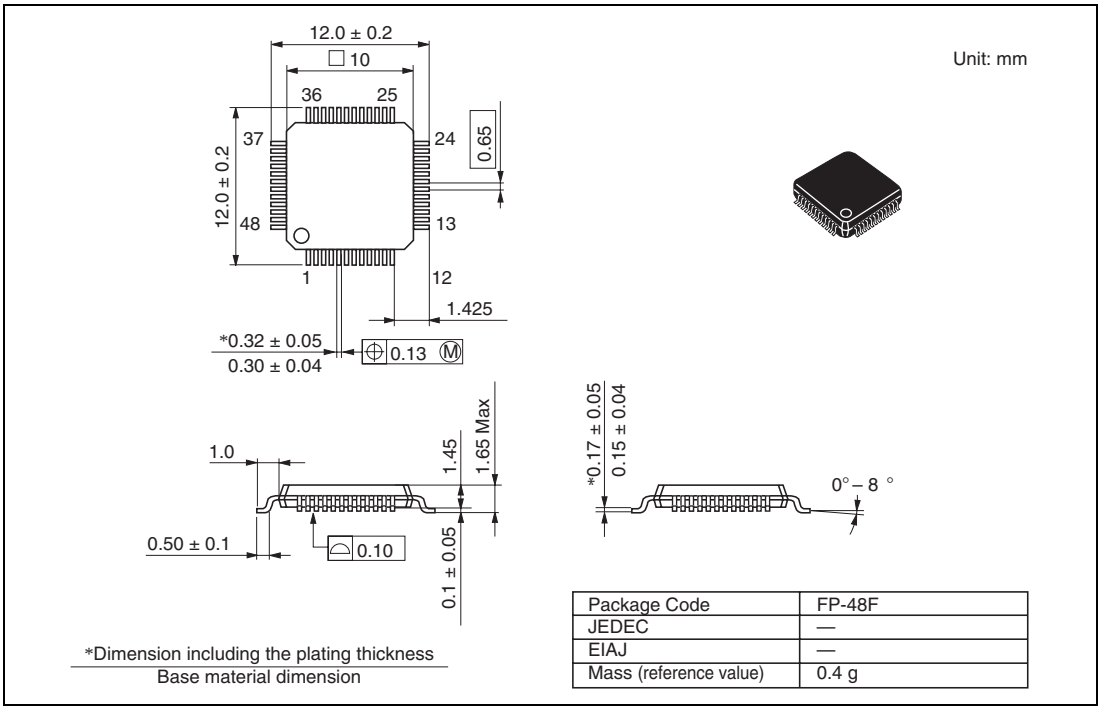
Package Code	FP-64E
JEDEC	–
EIAJ	Conforms
Mass (reference value)	0.4 g

Figure D.1 FP-64E Package Dimensions



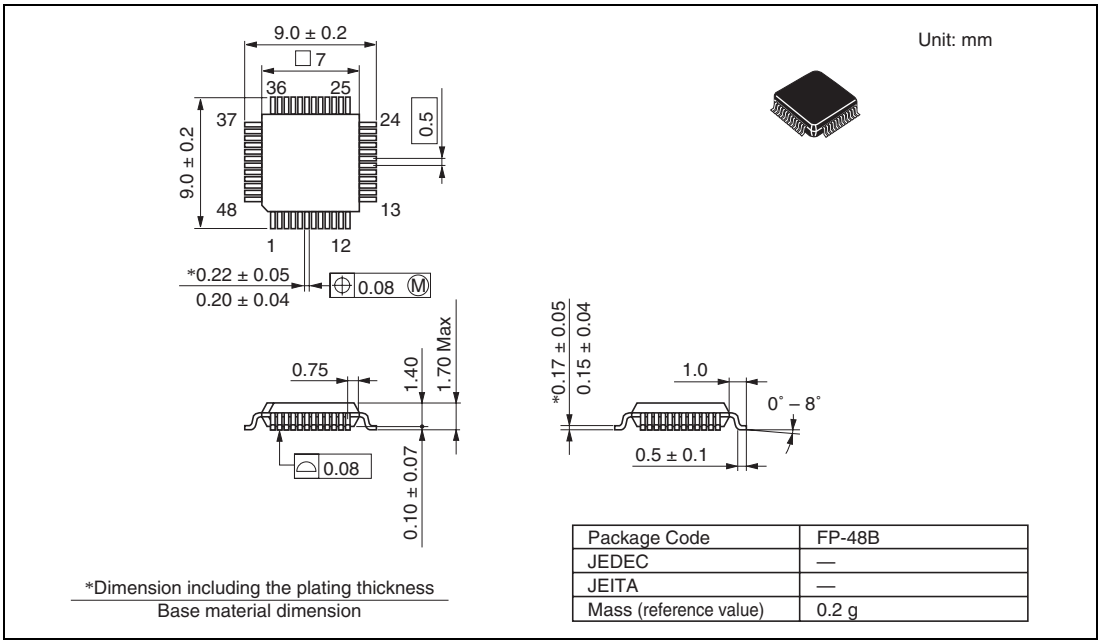
Package Code	FP-64A
JEDEC	–
EIAJ	Conforms
Mass (reference value)	1.2 g

Figure D.2 FP-64A Package Dimensions



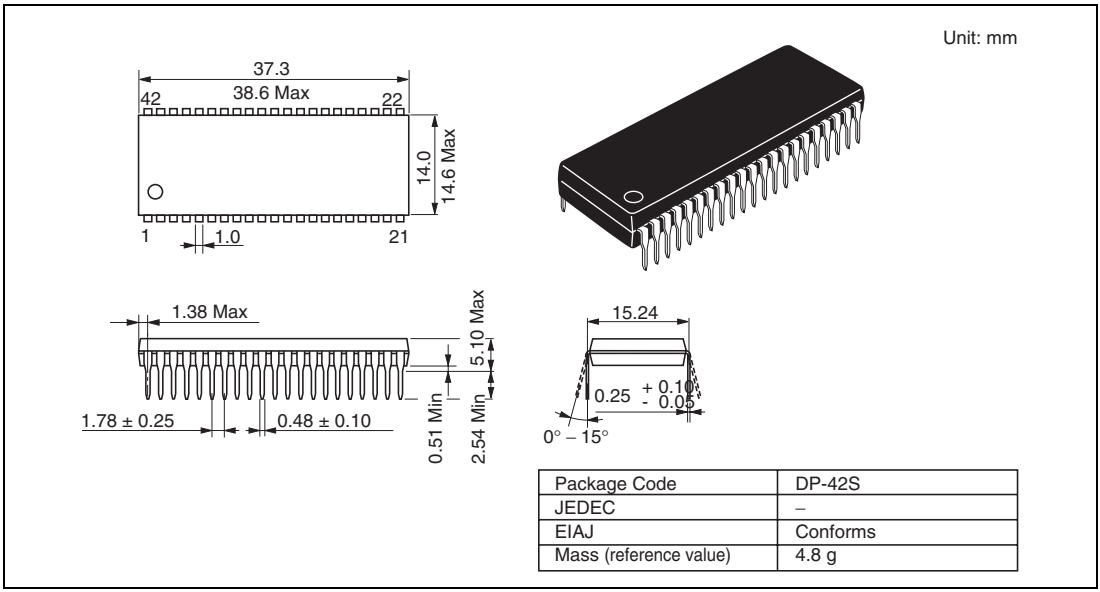
Package Code	FP-48F
JEDEC	—
EIAJ	—
Mass (reference value)	0.4 g

Figure D.3 FP-48F Package Dimensions



Package Code	FP-48B
JEDEC	—
JEITA	—
Mass (reference value)	0.2 g

Figure D.4 FP-48B Package Dimensions



Package Code	DP-42S
JEDEC	–
EIAJ	Conforms
Mass (reference value)	4.8 g

Figure D.5 DP-42S Package Dimensions

Appendix E EEPROM Stacked-Structure Cross-Sectional View

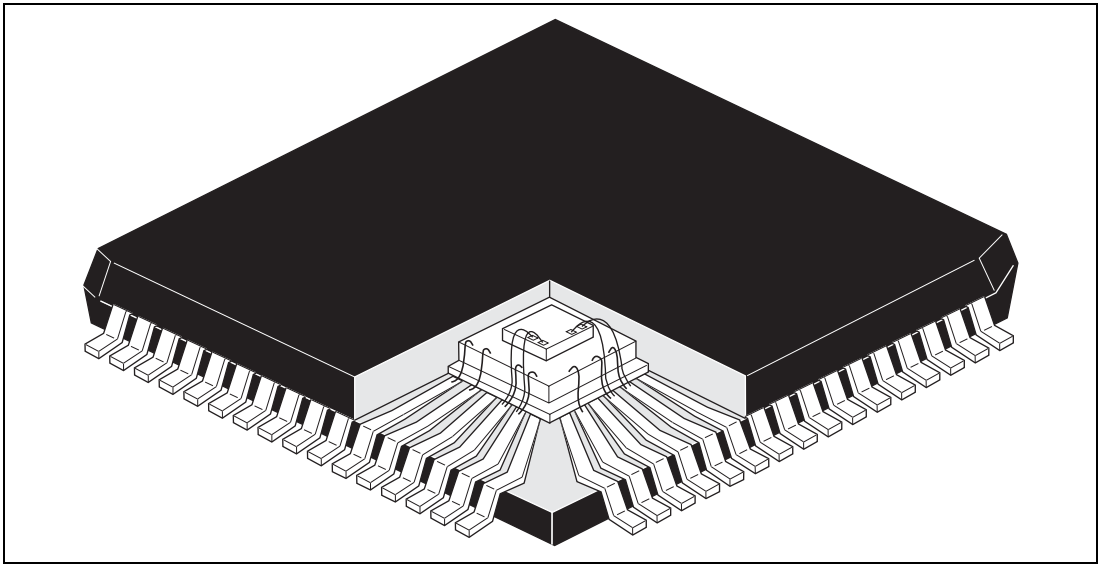


Figure E.1 EEPROM Stacked-Structure Cross-Sectional View

Main Revisions and Additions in this Edition

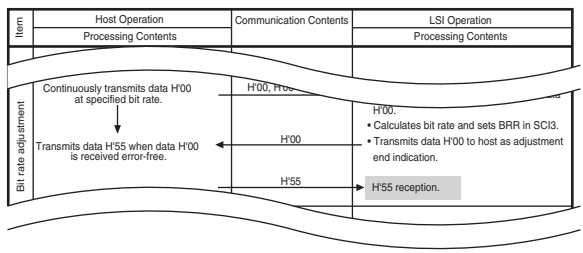
Item	Page	Revision (See Manual for Details)
Preface	vi, vii	Amended
Notes		When using the on-chip emulator (E7, E8) for H8/3664 program development and debugging, the following restrictions must be noted ~~(the on-chip debugging emulator (E7) can also be used).~~ 1. The NMI pin is reserved for the E7 or E8, and cannot be used. 2. Pins P85, P86, and P87 cannot be used. In order to use these pins, additional hardware must be provided on the user board. 3. Area H'7000 to H'7FFF is used by the E7 or E8, and is not available to the user. 4. Area H'F780 to H'FB7F must on no account be accessed. 5. When the E7 or E8 is used, address breaks can be set as either available to the user or for use by the E7 or E8. If address breaks are set as being used by the E7 or E8, the address break control registers must not be accessed. 6. When the E7 or E8 is used, NMI is an input/output pin (open-drain in output mode), P85 and P87 are input pins, and P86 is an output pin.
Figure 5.3 Typical Connection to Crystal Resonator Figure 5.5 Typical Connection to Ceramic Resonator Figure 5.8 Typical Connection to 32.768-kHz Crystal Resonator	78 to 80	Added Note: Capacitances are reference values.

Item	Page	Revision (See Manual for Details)

6.1.1 System Control Register 1 (SYSCR1) — 85 — Amended

Bit	Bit Name	Description
3	NESEL	Noise Elimination Sampling Frequency Select The subclock pulse generator generates the watch clock signal (ϕ_w) and the system clock pulse generator generates the oscillator clock (ϕ_{osc}). This bit selects the sampling frequency of the oscillator clock when the watch clock signal (ϕ_w) is sampled. When ϕ_{osc} = 4 to 16 MHz, clear NESEL to 0. 0: Sampling rate is ϕ_{osc}/16 1: Sampling rate is ϕ_{osc}/4

Table 7.2 Boot Mode Operation — 102 — Amended



9.5.3 Pin Functions — 136 — Amended

- P84/FTIOD Pin

Register	TMRW		TIOR1		PCR8	
Bit Name	PWMD	IOD2	IOD1	IOD0	PCR84	Pin Function
Setting Value	0	0	0	0	0	P84 input/FTIOD input pin
					1	P84 output/FTIOD input pin
		0	0	1	X	FTIOD output pin
		0	1	X	X	FTIOD output pin
		1	X	X	0	P84 input/FTIOD input pin
					1	P84 output/FTIOD input pin
	1	X	X	X	X	PWM output pin

Item	Page	Revision (See Manual for Details)
9.5.3 Pin Functions • P83/FTIOC Pin	136	Amended

Register	TMRW		TIOR1		PCR8	
Bit Name	PWMC	IOC2	IOC1	IOC0	PCR83	Pin Function
Setting Value	0	0	0	0	0	P83 input/FTIOC input pin
					1	P83 output/FTIOC input pin
		0	0	1	X	FTIOC output pin
		0	1	X	X	FTIOC output pin
		1	X	X	0	P83 input/FTIOC input pin
					1	P83 output/FTIOC input pin
	1	X	X	X	X	PWM output pin

9.5.3 Pin Functions • P82/FTIOB Pin	137	Amended

Register	TMRW		TIOR0		PCR8	
Bit Name	PWMB	IOB2	IOB1	IOB0	PCR82	Pin Function
Setting Value	0	0	0	0	0	P82 input/FTIOB input pin
					1	P82 output/FTIOB input pin
		0	0	1	X	FTIOB output pin
		0	1	X	X	FTIOB output pin
		1	X	X	0	P82 input/FTIOB input pin
					1	P82 output/FTIOB input pin
	1	X	X	X	X	PWM output pin

11.3.4 Timer Control/Status Register V (TCSRV)	150, 151	Amended

Bit	Bit Name	Description
3	OS3	Output Select 3 and 2 These bits select an output method for the TMOV pin by the compare match of TCORB and TCNTV. 00: No change 01: 0 output1 :
2	OS2	
1	OS1	Output Select 1 and 0 These bits select an output method for the TMOV pin by the compare match of TCORA and TCNTV. :
0	OS0	

13.2.1 Timer Control/Status Register WD (TCSRWD)	192	Amended

Bit	Bit Name	Description
4	TCSRWE	Timer Control/Status Register WD Write Enable :

Item	Page	Revision (See Manual for Details)
15.5 Usage Notes • Notes on WAIT Function • Notes on TRS Bit Setting and ICDR Register Access	271 to 274	Added.
16.3.1 A/D Data Registers A to D (ADDRA to ADDRD)	278	Amended … Therefore, byte access to ADDR should be done by reading the upper byte first then the lower one. Word access is also possible. ADDR is initialized to H'0000.
Table 20.2 DC Characteristics (1)	314	Amended

Item	Applicable Pins	Test Condition	Values Min
Input high voltage	PB0 to PB7	V_{cc} = 4.0 V to 5.5 V	V_{cc} × 0.7
		~~AV_{cc} = 3.3 V to 5.5 V~~	V_{cc} × 0.8
Input low voltage	P50 to P57*, P74 to P76, P80 to P87, PB0 to PB7		–0.3
	~~Pb0 to PB7~~	~~AV_{cc} = 4.0 V to 5.5 V~~	~~–0.3~~
		~~AV_{cc} = 3.3 V to 5.5 V~~	~~–0.3~~

Table 20.2 DC Characteristics (1) Table 20.10 DC Characteristics (1)	318, 335	Amended Note: * Pin states during supply current measurement are given below (excluding current in the pull-up MOS transistors and output buffers).

Mode	$\overline{RES}$ Pin	Internal State
Active mode 1	V_{CC}	Operates
Active mode 2		Operates (ϕosc/64)
Sleep mode 1	V_{CC}	Only timers operate
Sleep mode 2		Only timers operate (ϕosc/64)

Item	Page	Revision (See Manual for Details)
Table 20.10 DC Characteristics (1)	331, 332	Amended

Item	Applicable Pins	Test Condition	Values Min
Input high voltage	PB0 to PB7	V_{cc} = 4.0 V to 5.5 V	V_{cc} × 0.7
		AV_{cc} = 3.3 V to 5.5 V	V_{cc} × 0.8
Input low voltage	P50 to P57*, P74 to P76, P80 to P87, PB0 to PB7		–0.3
~~Pb0 to PB7~~		~~AV_{cc} = 4.0 V to 5.5 V~~	~~–0.3~~
		~~AV_{cc} = 3.3 V to 5.5 V~~	~~–0.3~~

Table A.1 Instruction Set	351	Amended
2. Arithmetic instructions		

Mnemonic		Operand Size	Condition Code						No. of States*1	
			I	H	N	Z	V	C	Normal	Advanced
DAA	DAA Rd	B		*	↕	↕	*	↕	2	

Index

A

A/D converter 275
  Sample-and-hold circuit 282
  Scan mode 281
  Single mode 281
Absolute maximum ratings 311
Address break 67
Addressing modes 35
  Absolute address 37
  Immediate 37
  Memory indirect 38
  Program-counter relative 37
  Register direct 36
  Register indirect 36
  Register indirect with displacement 36
  Register indirect with post-increment 36
  Register indirect with pre-decrement 36

C

Clock
  Clock pulse generators 77
  Subclock generator 80
  System clock generator 78
Condition-code register (CCR) 19
CPU 13

E

EEPROM 287
  acknowledge 291
  Acknowledge polling 294
  Byte write 293
  Corresponding slave address reference address (ESAR) 292
  Current address read 295
  EEPROM interface 290
  EEPROM key register (EKR) 289
  Page write 293
  Random address read 296
  Sequential read 296
  Slave addressing 292
  Start condition 291
  Stop condition 291
Effective address 39
Electrical characteristics (F-ZTAT™ version, F-ZTAT™ version with EEPROM)
  AC characteristics 320
Electrical characteristics (F-ZTAT™ version, F-ZTAT™ version with EEPROM) 311
  DC characteristics 314
Electrical characteristics (Mask ROM version) 329
  AC characteristics 337
  DC characteristics 331
Exception handling 51
  Reset exception handling 59
  Trap instruction 51

F

Flash memory
  On-board programming modes 100
Flash memory 95
  Boot mode 101
  Boot program 100
  Erase/erase-verify 107
  Error protection 109
  Hardware protection 109
  Power-down state 110
  Program/program-verify 104
  Programmer mode 110

Software protection 109

G

General registers 18

I

I/O ports 115
I/O port block diagrams 377
I²C bus interface (IIC) 233
Acknowledge 250
Clock synchronous serial format 259
General call address 247
I²C bus data formats 250
I²C transfer rate 241
Slave address 250
Start condition 250
Stop condition 250
Instruction set 24
Interrupt
Internal interrupts 61
Interrupt response time 62
IRQ3 to IRQ0 interrupts 59
NMI interrupt 59
WKP5 to WKP0 interrupts 60
Interrupt mask bit (I) 19

L

Laminated-structure cross section of H8/3664N 401
Large current ports 1

M

Memory map 14

P

Package 2
Package dimensions 397
Pin arrangement 5
Power supply circuit
Internal power supply step-down circuit 299
Power supply circuit 299
Power-down modes 83
Module standby function 94
Sleep mode 91
Standby mode 91
Subactive mode 92
Subsleep mode 92
Prescaler S 81
Prescaler W 81
Product code lineup 395
Program counter (PC) 19

R

Registers
ABRKCR 68, 303, 306, 309
ABRKSR 70, 303, 306, 309
ADCR 280, 303, 306, 309
ADCSR 279, 303, 306, 309
ADDRA 278, 303, 306, 309
ADDRB 278, 303, 306, 309
ADDRC 278, 303, 306, 309
ADDRD 278, 303, 306, 309
BARH 70, 303, 306, 309
BARL 70, 303, 306, 309
BDRH 70, 303, 306, 309
BDRL 70, 303, 306, 309
BRR 205, 302, 306, 308
EBR1 ..... 99, 302, 305, 308
EKR 289, 304, 307, 310
FENR 100, 302, 305, 308
FLMCR1 ..... 97, 302, 305, 308
FLMCR2 ..... 98, 302, 305, 308
FLPWCR 99, 302, 305, 308
GRA 171, 302, 305, 308

GRB 171, 302, 305, 308
GRC 171, 302, 305, 308
GRD 171, 302, 305, 308
ICCR 242, 303, 306, 309
ICDR 236, 303, 306, 309
ICMR 239, 303, 306, 309
ICSR 245, 303, 306, 309
IEGR1 ..... 53, 304, 307, 310
IEGR2 ..... 54, 304, 307, 310
IENR1 ..... 55, 304, 307, 310
IRR1 ..... 56, 304, 307, 310
IWPR 57, 304, 307, 310
MSTCR1 ..... 87, 304, 307, 310
PCR1 ..... 117, 304, 307, 310
PCR2 ..... 121, 304, 307, 310
PCR5 ..... 126, 304, 307, 310
PCR7 ..... 131, 304, 307, 310
PCR8 ..... 134, 304, 307, 310
PDR1 ..... 118, 303, 306, 309
PDR2 ..... 122, 303, 306, 309
PDR5 ..... 127, 303, 306, 309
PDR7 ..... 131, 303, 307, 309
PDR8 ..... 134, 303, 307, 309
PDRB 138, 304, 307, 309
PMR1 ..... 116, 304, 307, 309
PMR5 ..... 125, 304, 307, 309
PUCR1 ..... 118, 303, 306, 309
PUCR5 ..... 127, 303, 306, 309
RDR 199, 303, 306, 308
RSR 199
SAR 238, 303, 306, 309
SARX 238, 303, 306, 309
SCR3 ..... 201, 302, 306, 308
SMR 200, 302, 306, 308
SSR 203, 303, 306, 308
SYSCR1 ..... 84, 304, 307, 310
SYSCR2 ..... 86, 304, 307, 310
TCA 142, 302, 305, 308
TCNT 170, 302, 305, 308
TCNTV 147, 302, 305, 308
TCORA 148, 302, 305, 308
TCORB 148, 302, 305, 308
TCRV0 ..... 148, 302, 305, 308
TCRV1 ..... 151, 302, 305, 308
TCRW 164, 302, 305, 308
TCSRV 150, 302, 305, 308
TCSRWD 192, 303, 306, 309
TCWD 193, 303, 306, 309
TDR 200, 303, 306, 308
TIERW 165, 302, 305, 308
TIOR0 ..... 167, 302, 305, 308
TIOR1 ..... 169, 302, 305, 308
TMA 141, 302, 305, 308
TMRW 163, 302, 305, 308
TMWD 194, 303, 306, 309
TSCR 248, 304, 307, 310
TSR 199
TSRW 166, 302, 305, 308

S

Serial communication interface 3 (SCI3) 197
Asynchronous mode 210
Bit rate 205
Break detection 230
Clocked synchronous mode 218
Framing error 214
Mark state 231
Multiprocessor communication function 224
Overrun error 214
Parity error 214
Stack pointer (SP) 19

T

Timer A 139
Timer V 145
Timer W 159
PWM operation 176

V
Vector address 52

W
Watchdog timer..... 191

Renesas 16-Bit Single-Chip Microcomputer
Hardware Manual
H8/3664 Group

Publication Date: 1st Edition, Mar, 2000
Rev.6.00, Mar. 24, 2006
Published by: Sales Strategic Planning Div.
Renesas Technology Corp.
Edited by: Customer Support Department
Global Strategic Communication Div.
Renesas Solutions Corp.

© 2006. Renesas Technology Corp., All rights reserved. Printed in Japan.

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan

RENESAS SALES OFFICES

http://www.renesas.com

Refer to "**http://www.renesas.com/en/network**" for the latest and detailed information.

Renesas Technology America, Inc.
450 Holger Way, San Jose, CA 95134-1368, U.S.A
Tel: <1> (408) 382-7500, Fax: <1> (408) 382-7501

Renesas Technology Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

Renesas Technology (Shanghai) Co., Ltd.
Unit 204, 205, AZIACenter, No.1233 Lujiazui Ring Rd, Pudong District, Shanghai, China 200120
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7898

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 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

H8/3664 Group
Hardware Manual

Renesas Electronics Corporation
1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan

REJ09B0142-0600

Downloaded from Arrow.com.