To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

Notice

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

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

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

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

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

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

User's Manual

16

H8/36087 Group

Hardware Manual

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

H8/36087F	HD64F36087
H8/36087	HD64336087
H8/36086	HD64336086
H8/36085	HD64336085
H8/36084	HD64336084
H8/36083	HD64336083
H8/36082	HD64336082

Renesas Electronics
www.renesas.com

Rev.2.00 2005.09

a third party.

2. Renesas Technology Corp. assumes no responsibility for any damage, or infringement of any party's rights, originating in the use of any product data, diagrams, charts, programs, algorithm circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, program algorithms represents information on products at the time of publication of these materials, and subject to change by Renesas Technology Corp. without notice due to product improvements other reasons. It is therefore recommended that customers contact Renesas Technology Co 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 from these inaccuracies or errors.
   Please also pay attention to information published by Renesas Technology Corp. by various 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 to system before making a final decision on the applicability of the information and products. Re Technology Corp. assumes no responsibility for any damage, liability or other loss resulting fr information contained herein.
5. Renesas Technology Corp. semiconductors are not designed or manufactured for use in a de system that is used under circumstances in which human life is potentially at stake. Please c Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor w considering the use of a product contained herein for any specific purposes, such as apparat 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 whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they be exported under a license from the Japanese government and cannot be imported into a co other than the approved destination.
   Any diversion or reexport contrary to the export control laws and regulations of Japan and/or country of destination is prohibited.
8. Please contact Renesas Technology Corp. for further details on these materials or the produc contained therein.

RENESAS

are in their open states, intermediate levels are induced by noise in the vicinity, a 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 throughou chip and a low level is input on the reset pin. During the period where the states a undefined, the register settings and the output state of each pin are also undefined your system so that it does not malfunction because of processing while it is in th undefined state. For those products which have a reset function, reset the LSI imr 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 reg may have been be allocated to these addresses. Do not access these registers; the operation is not guaranteed if they are accessed.

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

  The configuration of the functional description of each module differs according
  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. Ea
includes notes in relation to the descriptions given, and usage notes are given, as require
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 ve
This does not include all of the revised contents. For details, see the actual locations in t
manual.

11. Index

Objective: This manual was written to explain the hardware functions and electrical characteristics of the H8/36087 Group to the target users.
Refer to the H8/300H Series Software Manual for a detailed description o 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 in on the CPU, system control functions, peripheral functions and electrical characteristi
- 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 register. The addresses, bits, and initial values of the registers are summarized in secti List of Registers.

Example:	Register name:	The following notation is used for cases when the sa similar function, e.g. serial communication interface implemented on more than one channel: XXX_N (XXX is the register name and N is the cha number)
	Bit order:	The MSB is on the left and the LSB is on the right.

Notes:

When using an on-chip emulator (E7, E8) for H8/36087 program development and debug following restrictions must be noted.

7. Use channel 1 of the SCI3 (P21/RXD, P22/TXD) in on-board programming mode by mode.

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

H8/36087 Group manuals:

Document Title	Docume
H8/36087 Group Hardware Manual	This ma
H8/300H Series Software Manual	REJ09B

User's manuals for development tools:

Document Title	Docume
H8S, H8/300 Series C/C++ Compiler, Assembler, Optimizing Linkage Editor User's Manual	REJ10B
Microcomputer Development Environment System H8S, H8/300 Series Simulator/Debugger User's Manual	ADE-702
H8S, H8/300 Series High-Performance Embedded Workshop 3, Tutorial	REJ10B
H8S, H8/300 Series High-Performance Embedded Workshop 3, User's Manual	REJ10B

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

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

3.5.1 Interrupts after Reset
3.5.2 Notes on Stack Area Use
3.5.3 Notes on Rewriting Port Mode Registers

Section 4 Address Break
4.1 Register Descriptions
4.1.1 Address Break Control Register (ABRKCR)
4.1.2 Address Break Status Register (ABRKSR)
4.1.3 Break Address Registers (BARH, BARL)
4.1.4 Break Data Registers (BDRH, BDRL)
4.2 Operation

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

6.2.4 Subactive Mode
6.3 Operating Frequency in Active Mode
6.4 Direct Transition
6.4.1 Direct Transition from Active Mode to Subactive Mode
6.4.2 Direct Transition from Subactive Mode to Active Mode
6.5 Module Standby Function

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

9.2.2 Port Data Register 2 (PDR2)
9.2.3 Port Mode Register 3 (PMR3)
9.2.4 Pin Functions
9.3 Port 3
9.3.1 Port Control Register 3 (PCR3)
9.3.2 Port Data Register 3 (PDR3)
9.3.3 Pin Functions
9.4 Port 5
9.4.1 Port Mode Register 5 (PMR5)
9.4.2 Port Control Register 5 (PCR5)
9.4.3 Port Data Register 5 (PDR5)
9.4.4 Port Pull-Up Control Register 5 (PUCR5)
9.4.5 Pin Functions
9.5 Port 6
9.5.1 Port Control Register 6 (PCR6)
9.5.2 Port Data Register 6 (PDR6)
9.5.3 Pin Functions
9.6 Port 7
9.6.1 Port Control Register 7 (PCR7)
9.6.2 Port Data Register 7 (PDR7)
9.6.3 Pin Functions
9.7 Port 8
9.7.1 Port Control Register 8 (PCR8)
9.7.2 Port Data Register 8 (PDR8)
9.7.3 Pin Functions
9.8 Port B
9.8.1 Port Data Register B (PDRB)

10.3.7 Clock Source Select Register (RTCCSR)
10.4 Operation
10.4.1 Initial Settings of Registers after Power-On
10.4.2 Initial Setting Procedure
10.4.3 Data Reading Procedure
10.5 Interrupt Source

Section 11 Timer B1
11.1 Features
11.2 Input/Output Pin
11.3 Register Descriptions
11.3.1 Timer Mode Register B1 (TMB1)
11.3.2 Timer Counter B1 (TCB1)
11.3.3 Timer Load Register B1 (TLB1)
11.4 Operation
11.4.1 Interval Timer Operation
11.4.2 Auto-Reload Timer Operation
11.4.3 Event Counter Operation
11.5 Timer B1 Operating Modes

Section 12 Timer V
12.1 Features
12.2 Input/Output Pins
12.3 Register Descriptions
12.3.1 Timer Counter V (TCNTV)
12.3.2 Time Constant Registers A and B (TCORA, TCORB)
12.3.3 Timer Control Register V0 (TCRV0)
12.3.4 Timer Control/Status Register V (TCSRV)
12.3.5 Timer Control Register V1 (TCRV1)

13.3.1 Timer Start Register (TSTR)
13.3.2 Timer Mode Register (TMDR)
13.3.3 Timer PWM Mode Register (TPMR)
13.3.4 Timer Function Control Register (TFCR)
13.3.5 Timer Output Master Enable Register (TOER)
13.3.6 Timer Output Control Register (TOCR)
13.3.7 Timer Counter (TCNT)
13.3.8 General Registers A, B, C, and D (GRA, GRB, GRC, and GRD)
13.3.9 Timer Control Register (TCR)
13.3.10 Timer I/O Control Register (TIORA and TIORC)
13.3.11 Timer Status Register (TSR)
13.3.12 Timer Interrupt Enable Register (TIER)
13.3.13 PWM Mode Output Level Control Register (POCR)
13.3.14 Interface with CPU
13.4 Operation
13.4.1 Counter Operation
13.4.2 Waveform Output by Compare Match
13.4.3 Input Capture Function
13.4.4 Synchronous Operation
13.4.5 PWM Mode
13.4.6 Reset Synchronous PWM Mode
13.4.7 Complementary PWM Mode
13.4.8 Buffer Operation
13.4.9 Timer Z Output Timing
13.5 Interrupts
13.5.1 Status Flag Set Timing
13.5.2 Status Flag Clearing Timing
13.6 Usage Notes

15.3 Register Descriptions
15.3.1 PWM Control Register (PWCR)
15.3.2 PWM Data Registers U and L (PWDRU, PWDRL)
15.4 Operation

Section 16 Serial Communication Interface 3 (SCI3)
16.1 Features
16.2 Input/Output Pins
16.3 Register Descriptions
16.3.1 Receive Shift Register (RSR)
16.3.2 Receive Data Register (RDR)
16.3.3 Transmit Shift Register TSR (SCI3)
16.3.4 Transmit Data Register (TDR)
16.3.5 Serial Mode Register (SMR)
16.3.6 Serial Control Register 3 (SCR3)
16.3.7 Serial Status Register (SSR)
16.3.8 Bit Rate Register (BRR)
16.4 Operation in Asynchronous Mode
16.4.1 Clock
16.4.2 SCI3 Initialization
16.4.3 Data Transmission
16.4.4 Serial Data Reception
16.5 Operation in Clocked Synchronous Mode
16.5.1 Clock
16.5.2 SCI3 Initialization
16.5.3 Serial Data Transmission
16.5.4 Serial Data Reception (Clocked Synchronous Mode)
16.5.5 Simultaneous Serial Data Transmission and Reception
16.6 Multiprocessor Communication Function

Section 17 I²C Bus Interface 2 (IIC2)
17.1 Features
17.2 Input/Output Pins
17.3 Register Descriptions
17.3.1 I²C Bus Control Register 1 (ICCR1)
17.3.2 I²C Bus Control Register 2 (ICCR2)
17.3.3 I²C Bus Mode Register (ICMR)
17.3.4 I²C Bus Interrupt Enable Register (ICIER)
17.3.5 I²C Bus Status Register (ICSR)
17.3.6 Slave Address Register (SAR)
17.3.7 I²C Bus Transmit Data Register (ICDRT)
17.3.8 I²C Bus Receive Data Register (ICDRR)
17.3.9 I²C Bus Shift Register (ICDRS)
17.4 Operation
17.4.1 I²C Bus Format
17.4.2 Master Transmit Operation
17.4.3 Master Receive Operation
17.4.4 Slave Transmit Operation
17.4.5 Slave Receive Operation
17.4.6 Clocked Synchronous Serial Format
17.4.7 Noise Canceler
17.4.8 Example of Use
17.5 Interrupt Request
17.6 Bit Synchronous Circuit
17.7 Usage Notes
17.7.1 Issue (Retransmission) of Start/Stop Conditions
17.7.2 WAIT Setting in I²C Bus Mode Register (ICMR)

18.4.3 Input Sampling and A/D Conversion Time
18.4.4 External Trigger Input Timing
18.5 A/D Conversion Accuracy Definitions
18.6 Usage Notes
18.6.1 Permissible Signal Source Impedance
18.6.2 Influences on Absolute Accuracy

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

Section 20 Electrical Characteristics
20.1 Absolute Maximum Ratings
20.2 Electrical Characteristics (F-ZTAT™ Version)
20.2.1 Power Supply Voltage and Operating Ranges
20.2.2 DC Characteristics
20.2.3 AC Characteristics
20.2.4 A/D Converter Characteristics
20.2.5 Watchdog Timer Characteristics
20.2.6 Flash Memory Characteristics
20.3 Electrical Characteristics (Mask-ROM Version)
20.3.1 Power Supply Voltage and Operating Ranges
20.3.2 DC Characteristics
20.3.3 AC Characteristics
20.3.4 A/D Converter Characteristics
20.3.5 Watchdog Timer Characteristics
20.4 Operation Timing
20.5 Output Load Condition

Appendix D Package Dimensions .....
Main Revisions and Additions in this Edition.....
Index .....

Figure 2.1 Memory Map (2)
Figure 2.2 CPU Registers
Figure 2.3 Usage of General Registers
Figure 2.4 Relationship between Stack Pointer and Stack Area
Figure 2.5 General Register Data Formats (1)
Figure 2.5 General Register Data Formats (2)
Figure 2.6 Memory Data Formats
Figure 2.7 Instruction Formats
Figure 2.8 Branch Address Specification in Memory Indirect Mode
Figure 2.9 On-Chip Memory Access Cycle
Figure 2.10 On-Chip Peripheral Module Access Cycle (3-State Access)
Figure 2.11 CPU Operation States
Figure 2.12 State Transitions
Figure 2.13 Example of Timer Configuration with Two Registers Allocated to Same Address

Section 3 Exception Handling

Figure 3.1 Reset Sequence
Figure 3.2 Stack Status after Exception Handling
Figure 3.3 Interrupt Sequence
Figure 3.4 Port Mode Register Setting and Interrupt Request Flag Clearing Procedure

Section 4 Address Break

Figure 4.1 Block Diagram of Address Break
Figure 4.2 Address Break Interrupt Operation Example (1)
Figure 4.2 Address Break Interrupt Operation Example (2)

Section 5 Clock Pulse Generators

Figure 5.1 Block Diagram of Clock Pulse Generators
Figure 5.2 Block Diagram of System Clock Generator
Figure 5.3 Typical Connection to Crystal Resonator

Section 7 ROM
Figure 7.1 Flash Memory Block Configuration
Figure 7.2 Programming/Erasing Flowchart Example in User Program Mode
Figure 7.3 Program/Program-Verify Flowchart
Figure 7.4 Erase/Erase-Verify Flowchart

Section 9 I/O Ports
Figure 9.1 Port 1 Pin Configuration
Figure 9.2 Port 2 Pin Configuration
Figure 9.3 Port 3 Pin Configuration
Figure 9.4 Port 5 Pin Configuration
Figure 9.5 Port 6 Pin Configuration
Figure 9.6 Port 7 Pin Configuration
Figure 9.7 Port 8 Pin Configuration
Figure 9.8 Port B Pin Configuration

Section 10 Realtime Clock (RTC)
Figure 10.1 Block Diagram of RTC
Figure 10.2 Definition of Time Expression
Figure 10.3 Initial Setting Procedure
Figure 10.4 Example: Reading of Inaccurate Time Data

Section 11 Timer B1
Figure 11.1 Block Diagram of Timer B1

Section 12 Timer V
Figure 12.1 Block Diagram of Timer V
Figure 12.2 Increment Timing with Internal Clock
Figure 12.3 Increment Timing with External Clock
Figure 12.4 OVF Set Timing
Figure 12.5 CMFA and CMFB Set Timing

Figure 13.2 Timer Z (Channel 0) Block Diagram .....
Figure 13.3 Timer Z (Channel 1) Block Diagram .....
Figure 13.4 Example of Outputs in Reset Synchronous PWM Mode and Complementa PWM Mode .....
Figure 13.5 Accessing Operation of 16-Bit Register (between CPU and TCNT (16 bits
Figure 13.6 Accessing Operation of 8-Bit Register (between CPU and TSTR (8 bits)).
Figure 13.7 Example of Counter Operation Setting Procedure .....
Figure 13.8 Free-Running Counter Operation .....
Figure 13.9 Periodic Counter Operation .....
Figure 13.10 Count Timing at Internal Clock Operation .....
Figure 13.11 Count Timing at External Clock Operation (Both Edges Detected).....
Figure 13.12 Example of Setting Procedure for Waveform Output by Compare Match.
Figure 13.13 Example of 0 Output/1 Output Operation .....
Figure 13.14 Example of Toggle Output Operation .....
Figure 13.15 Output Compare Timing.....
Figure 13.16 Example of Input Capture Operation Setting Procedure .....
Figure 13.17 Example of Input Capture Operation.....
Figure 13.18 Input Capture Signal Timing .....
Figure 13.19 Example of Synchronous Operation Setting Procedure .....
Figure 13.20 Example of Synchronous Operation.....
Figure 13.21 Example of PWM Mode Setting Procedure .....
Figure 13.22 Example of PWM Mode Operation (1) .....
Figure 13.23 Example of PWM Mode Operation (2) .....
Figure 13.24 Example of PWM Mode Operation (3) .....
Figure 13.25 Example of PWM Mode Operation (4) .....
Figure 13.26 Example of Reset Synchronous PWM Mode Setting Procedure.....
Figure 13.27 Example of Reset Synchronous PWM Mode Operation (OLS0 = OLS1 =
Figure 13.28 Example of Reset Synchronous PWM Mode Operation (OLS0 = OLS1 =
Figure 13.29 Example of Complementar y PWM Mode Setting Procedure.....
Figure 13.30 Canceling Procedure of Complementary PWM Mode .....

(Buffer Operation for Output Compare Register) .....
Figure 13.39 Example of Compare Match Timing for Buffer Operation .....
Figure 13.40 Example of Buffer Operation (2)
(Buffer Operation for Input Capture Register) .....
Figure 13.41 Input Capture Timing of Buffer Operation.....
Figure 13.42 Buffer Operation (3)
(Buffer Operation in Complementary PWM Mode CMD1 = CMD0 = 1)...
Figure 13.43 Buffer Operation (4)
(Buffer Operation in Complementary PWM Mode CMD1 = CMD0 = 1)...
Figure 13.44 Example of Output Disable Timing of Timer Z by Writing to TOER .....
Figure 13.45 Example of Output Disable Timing of Timer Z by External Trigger.....
Figure 13.46 Example of Output Inverse Timing of Timer Z by Writing to TFCR .....
Figure 13.47 Example of Output Inverse Timing of Timer Z by Writing to POCR.....
Figure 13.48 IMF Flag Set Timing when Compare Match Occurs .....
Figure 13.49 IMF Flag Set Timing at Input Capture .....
Figure 13.50 OVF Flag Set Timing.....
Figure 13.51 Status Flag Clearing Timing.....
Figure 13.52 Contention between TCNT Write and Clear Operations.....
Figure 13.53 Contention between TCNT Write and Increment Operations .....
Figure 13.54 Contention between GR Write and Compare Match.....
Figure 13.55 Contention between TCNT Write and Overflow.....
Figure 13.56 Contention between GR Read and Input Capture.....
Figure 13.57 Contention between Count Clearing and Increment Operations by Input Capture .....
Figure 13.58 Contention between GR Write and Input Capture.....
Figure 13.59 When Compare Match and Bit Manipulation Instruction to TOCR Occur a Same Timing .....

Figure 16.3 Relationship between Output Clock and Transfer Data Phase (Asynchronous Mode)(Example with 8-Bit Data, Parity, Two Stop Bits) ...
Figure 16.4 Sample SCI3 Initialization Flowchart ...
Figure 16.5 Example of SCI3 Transmission in Asynchronous Mode (8-Bit Data, Parity, One Stop Bit) ...
Figure 16.6 Sample Serial Transmission Data Flowchart (Asynchronous Mode) ...
Figure 16.7 Example of SCI3 Reception in Asynchronous Mode (8-Bit Data, Parity, One Stop Bit) ...
Figure 16.8 Sample Serial Reception Data Flowchart (Asynchronous Mode)(1) ...
Figure 16.8 Sample Serial Reception Data Flowchart (Asynchronous Mode)(2) ...
Figure 16.9 Data Format in Clocked Synchronous Communication ...
Figure 16.10 Example of SCI3 Transmission in Clocked Synchronous Mode ...
Figure 16.11 Sample Serial Transmission Flowchart (Clocked Synchronous Mode) ...
Figure 16.12 Example of SCI3 Reception in Clocked Synchronous Mode ...
Figure 16.13 Sample Serial Reception Flowchart (Clocked Synchronous Mode) ...
Figure 16.14 Sample Flowchart of Simultaneous Serial Transmit and Receive Operatio (Clocked Synchronous Mode) ...
Figure 16.15 Example of Inter-Processor Communication Using Multiprocessor Forma (Transmission of Data H'AA to Receiving Station A) ...
Figure 16.16 Sample Multiprocessor Serial Transmission Flowchart ...
Figure 16.17 Sample Multiprocessor Serial Reception Flowchart (1) ...
Figure 16.17 Sample Multiprocessor Serial Reception Flowchart (2) ...
Figure 16.18 Example of SCI3 Reception Using Multiprocessor Format (Example with Data, Multiprocessor Bit, One Stop Bit) ...
Figure 16.19 Receive Data Sampling Timing in Asynchronous Mode ...

Section 17 I^2C Bus Interface 2 (IIC2)

Figure 17.1 Block Diagram of I^2C Bus Interface 2 ...
Figure 17.2 External Circuit Connections of I/O Pins ...
Figure 17.3 I^2C Bus Formats ...

Figure 17.15 Receive Mode Operation Timing
Figure 17.16 Block Diagram of Noise Conceler
Figure 17.17 Sample Flowchart for Master Transmit Mode
Figure 17.18 Sample Flowchart for Master Receive Mode
Figure 17.19 Sample Flowchart for Slave Transmit Mode
Figure 17.20 Sample Flowchart for Slave Receive Mode
Figure 17.21 The Timing of the Bit Synchronous Circuit

Section 18 A/D Converter

Figure 18.1 Block Diagram of A/D Converter
Figure 18.2 A/D Conversion Timing
Figure 18.3 External Trigger Input Timing
Figure 18.4 A/D Conversion Accuracy Definitions (1)
Figure 18.5 A/D Conversion Accuracy Definitions (2)
Figure 18.6 Analog Input Circuit Example

Section 20 Electrical Characteristics

Figure 20.1 System Clock Input Timing
Figure 20.2 $\overline{\text{RES}}$ 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
Figure 20.6 SCI Input/Output Timing in Clocked Synchronous Mode
Figure 20.7 Output Load Circuit

Appendix

Figure B.1 Port 1 Block Diagram (P17)
Figure B.2 Port 1 Block Diagram (P14, P16)
Figure B.3 Port 1 Block Diagram (P15)
Figure B.4 Port 1 Block Diagram (P12)
Figure B.5 Port 2 Block Diagram (P11)

Figure B.17 Port 7 Block Diagram (P75) .....
Figure B.18 Port 7 Block Diagram (P74) .....
Figure B.19 Port 7 Block Diagram (P72) .....
Figure B.20 Port 7 Block Diagram (P71) .....
Figure B.21 Port 7 Block Diagram (P70) .....
Figure B.22 Port 8 Block Diagram (P87 to P85) .....
Figure B.23 Port B Block Diagram (PB7 to PB0) .....
Figure D.1 FP-64E Package Dimensions.....
Figure D.2 FP-64A Package Dimensions .....

Table 2.4 Logic Operations Instructions
Table 2.5 Shift Instructions
Table 2.6 Bit Manipulation Instructions (1)
Table 2.6 Bit Manipulation Instructions (2)
Table 2.7 Branch Instructions
Table 2.8 System Control Instructions
Table 2.9 Block Data Transfer Instructions
Table 2.10 Addressing Modes
Table 2.11 Absolute Address Access Ranges
Table 2.12 Effective Address Calculation (1)
Table 2.12 Effective Address Calculation (2)

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

Section 4 Address Break
Table 4.1 Access and Data Bus Used

Section 5 Clock Pulse Generators
Table 5.1 Crystal Resonator Parameters

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

Section 7 ROM
Table 7.1 Setting Programming Modes
Table 7.2 Boot Mode Operation

Table 11.1 Pin Configuration
Table 11.2 Timer B1 Operating Modes

Section 12 Timer V
Table 12.1 Pin Configuration
Table 12.2 Clock Signals to Input to TCNTV and Counting Conditions

Section 13 Timer Z
Table 13.1 Timer Z Functions
Table 13.2 Pin Configuration
Table 13.3 Initial Output Level of FTIOB0 Pin
Table 13.4 Output Pins in Reset Synchronous PWM Mode
Table 13.5 Register Settings in Reset Synchronous PWM Mode
Table 13.6 Output Pins in Complementary PWM Mode
Table 13.7 Register Settings in Complementary PWM Mode
Table 13.8 Register Combinations in Buffer Operation

Section 15 14-Bit PWM
Table 15.1 Pin Configuration

Section 16 Serial Communication Interface 3 (SCI3)
Table 16.1 Channel Configuration
Table 16.2 Pin Configuration
Table 16.3 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (1
Table 16.3 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (2
Table 16.3 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (3
Table 16.4 Maximum Bit Rate for Each Frequency (Asynchronous Mode)
Table 16.5 Examples of BRR Settings for Various Bit Rates (Clocked Synchronous M (1)
Table 16.5 Examples of BRR Settings for Various Bit Rates (Clocked Synchronous M (2)

Table 18.3 A/D Conversion Time (Single Mode)

Section 20 Electrical Characteristics

Table 20.1 Absolute Maximum Ratings
Table 20.2 DC Characteristics (1)
Table 20.2 DC Characteristics (2)
Table 20.3 AC Characteristics
Table 20.4 I^2C Bus Interface Timing
Table 20.5 Serial Communication Interface (SCI) Timing
Table 20.6 A/D Converter Characteristics
Table 20.7 Watchdog Timer Characteristics
Table 20.8 Flash Memory Characteristics
Table 20.9 DC Characteristics (1)
Table 20.9 DC Characteristics (2)
Table 20.10 AC Characteristics
Table 20.11 I^2C Bus Interface Timing
Table 20.12 Serial Communication Interface (SCI) Timing
Table 20.13 A/D Converter Characteristics
Table 20.14 Watchdog Timer Characteristics

Appendix

Table A.1 Instruction Set
Table A.2 Operation Code Map (1)
Table A.2 Operation Code Map (2)
Table A.2 Operation Code Map (3)
Table A.3 Number of Cycles in Each Instruction
Table A.4 Number of Cycles in Each Instruction
Table A.5 Combinations of Instructions and Addressing Modes

— RTC (can be used as a free running counter)
— Timer B1 (8-bit timer)
— Timer V (8-bit timer)
— Timer Z (16-bit timer)
— 14-bit PWM
— Watchdog timer
— SCI (Asynchronous or clocked synchronous serial communication interface) × 2
— I^2C Bus Interface (conforms to the I^2C bus interface format that is advocated by P
Electronics)
— 10-bit A/D converter

H8/36083	HD64336083	24 kbytes	3 kbytes
H8/36082	HD64336082	16 kbytes	3 kbytes

- General I/O ports
  - I/O pins: 45 I/O pins including 8 large current ports (I_{OL} = 10 mA, @V_{OL} = 1.0V)
  - Input-only pins: 8 input pins (also used for analog input)
- Supports various power-down states

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

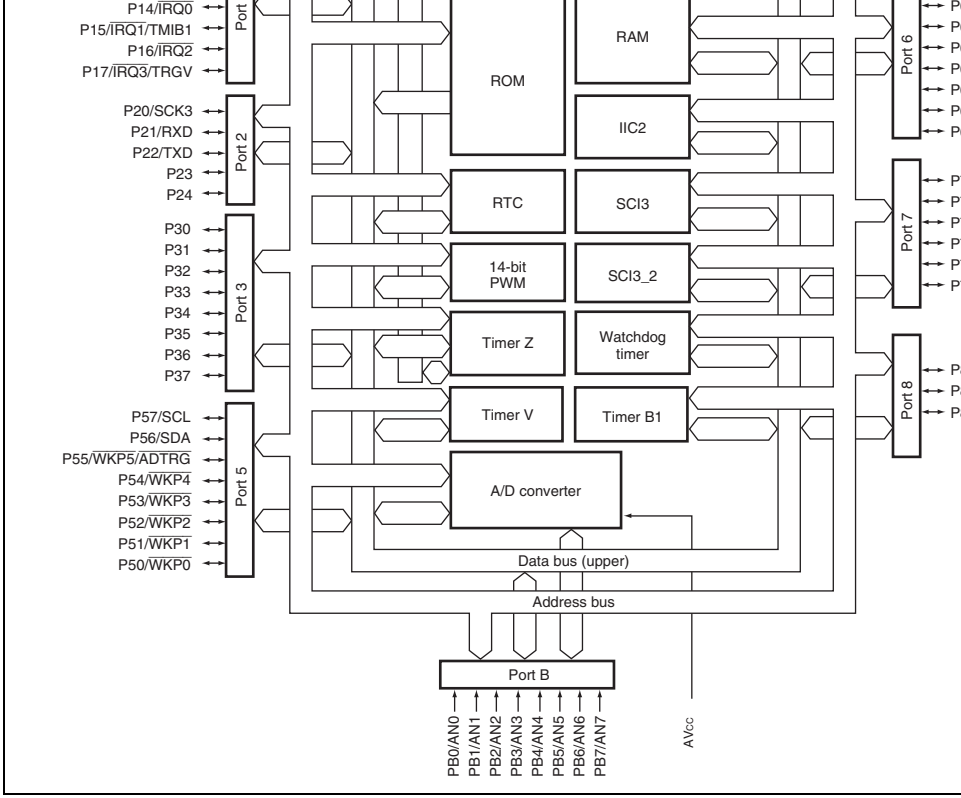


Figure 1.1 Internal Block Diagram of H8/36087 Group of F-ZTAT™ and Mask-ROM Versions

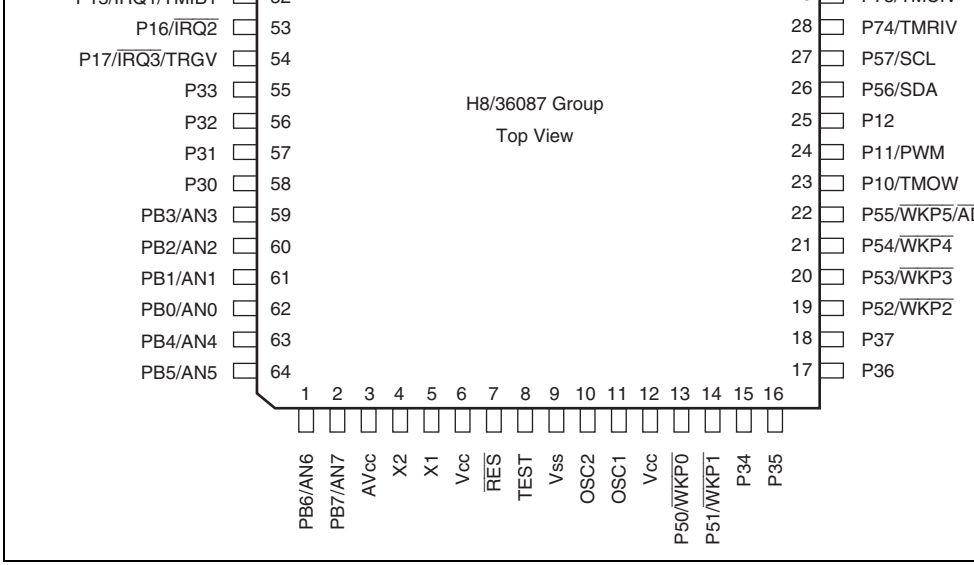


Figure 1.2 Pin Arrangement of H8/36087 Group of F-ZTAT™ and Mask-ROM V
(FP-64E, FP-64A)

	AV_{cc}	3	Input	Analog power supply pin for the A/D c When the A/D converter is not used, c this pin to the system power supply.
Clock pins	OSC1	11	Input	These pins connect with crystal or cer resonator for the system clock, or can to input an external clock. See section 5, Clock Pulse Generator typical connection.
	OSC2	10	Output	
	X1	5	Input	These pins connect with a 32.768 kHz resonator for the subclock. See sectio Pulse Generators, for a typical connec
	X2	4	Output	
System control	$\overline{RES}$	7	Input	Reset pin. The pull-up resistor (typ. 15 incorporated. When driven low, the ch
	TEST	8	Input	Test pin. Connect this pin to Vss.
Interrupt pins	$\overline{NMI}$	35	Input	Non-maskable interrupt request input sure to pull-up by a pull-up resistor.
	$\overline{IRQ0}$ to $\overline{IRQ3}$	51 to 54	Input	External interrupt request input pins. C the rising or falling edge.
	$\overline{WKP0}$ to $\overline{WKP5}$	13, 14, 19 to 22	Input	External interrupt request input pins. C the rising or falling edge.
RTC	TMOW	23	Output	This is an output pin for divided clocks
Timer B1	TMIB1	52	Input	External event input pin.
Timer V	TMOV	30	Output	This is an output pin for waveforms ge by the output compare function.
	TMCIV	29	Input	External event input pin.
	TMRIV	28	Input	Counter reset input pin.
	TRGV	54	Input	Counter start trigger input pin.

				input/PWM output pin
	FTIOA1	37	I/O	Output compare output/input capture input/PWM output pin (at a reset, complementary PWM mode)
	FTIOB1 to FTIOD1	38 to 40	I/O	Output compare output/input capture input/PWM output pin
14-bit PWM	PWM	24	Output	14-bit PWM square wave output pin
I^2C bus interface (IIC)	SDA	26	I/O	IIC data I/O pin. Can directly drive a b NMOS open-drain output. When using external pull-up resistance is required
	SCL	27	I/O	IIC clock I/O pin. Can directly drive a b NMOS open-drain output. When using external pull-up resistance is required
Serial communication interface (SCI)	TXD, TXD_2	46, 50	Output	Transmit data output pin
	RXD, RXD_2	45, 49	Input	Receive data input pin
	SCK3, SCK3_2	44, 48	I/O	Clock I/O pin
A/D converter	AN7 to AN0	1, 2, 59 to 64	Input	Analog input pin
	$\overline{\text{ADTRG}}$	22	Input	A/D converter trigger input pin.

P57 to P50	13, 14, 19 to 22, 26, 27	I/O	8-bit I/O port
P67 to P60	32 to 34, 36, 37 to 40	I/O	8-bit I/O port
P76 to P74, P72 to P70	28 to 30, 48 to 50	I/O	6-bit I/O port
P87 to P85	41 to 43	I/O	3-bit I/O port.

- General-register architecture
  - Sixteen 16-bit general registers also usable as sixteen 8-bit registers and eight 16 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

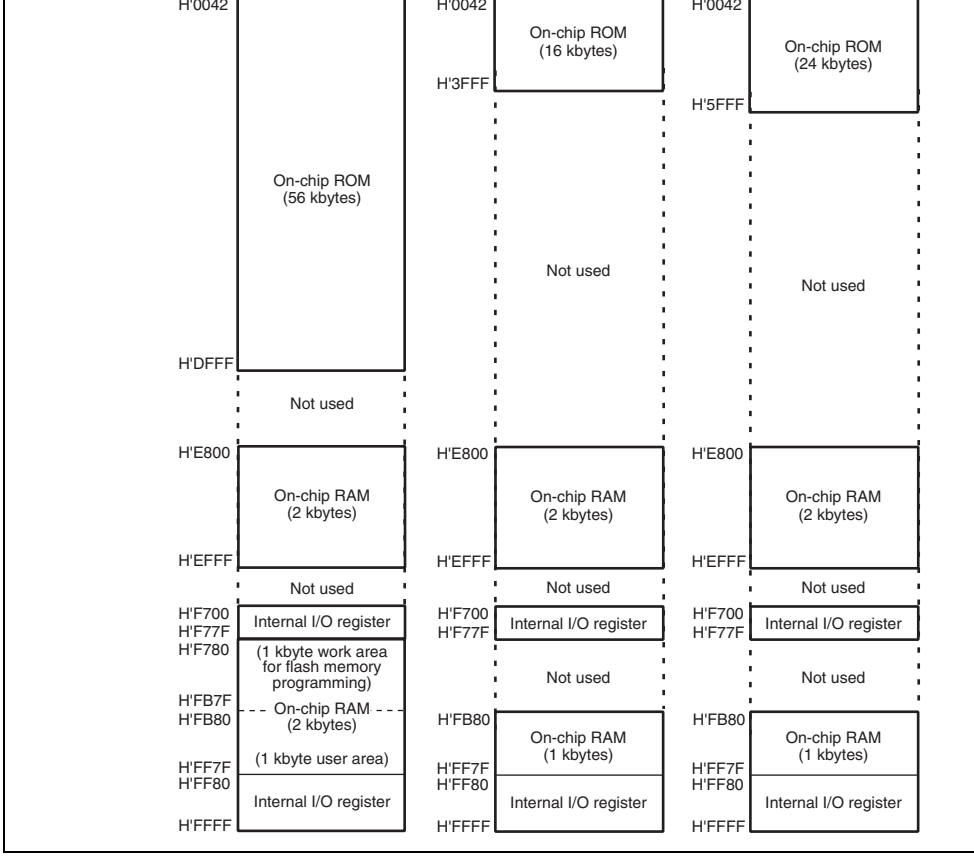


Figure 2.1 Memory Map (1)

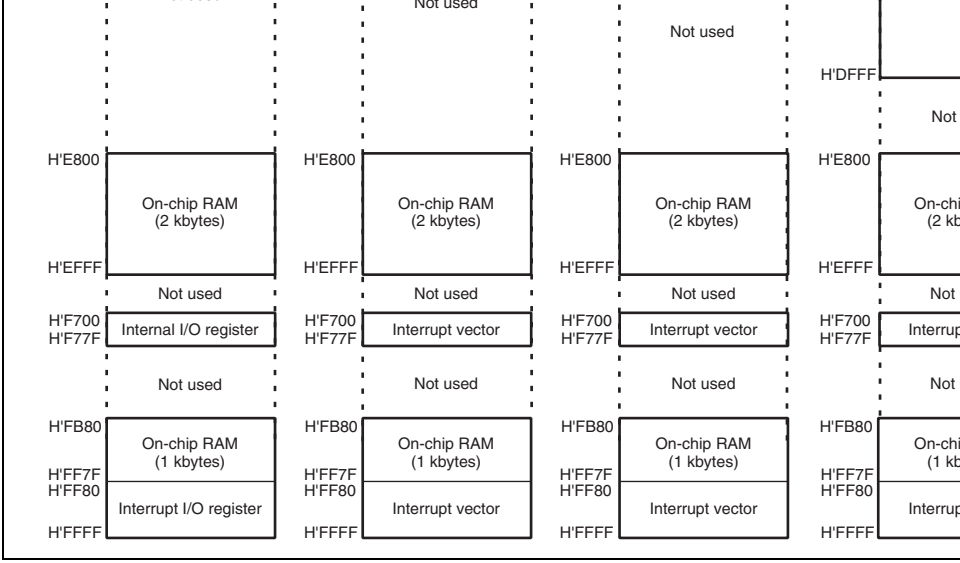
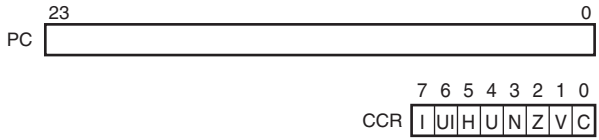


Figure 2.1 Memory Map (2)

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

Control Registers (CR)



[Legend]

SP: Stack pointer
PC: Program counter
CCR: Condition-code register
I: Interrupt mask bit
UI: User bit
H: Half-carry flag
U: User bit
N: Negative flag
Z: Zero flag
V: Overflow flag
C: Carry flag

Figure 2.2 CPU Registers

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

The usage of each register can be selected independently.

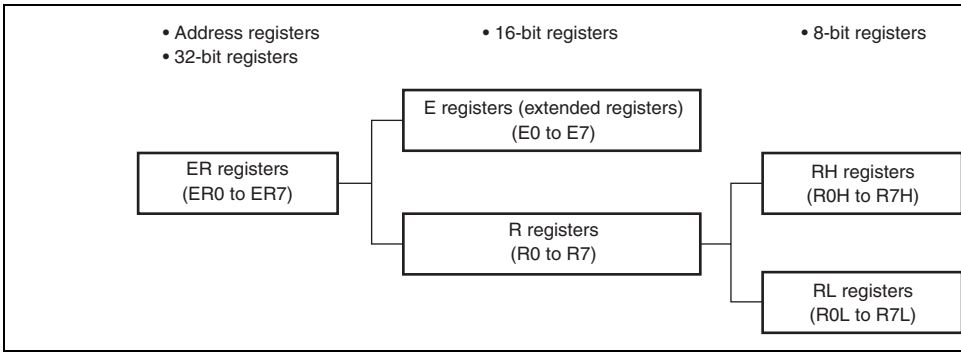


Figure 2.3 Usage of General Registers

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

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. Th
of all CPU instructions is 2 bytes (one word), so the least significant PC bit is ignored. (W
instruction is fetched, the least significant PC bit is regarded as 0). The PC is initialized w
start address is loaded by the vector address generated during reset exception-handling se

2.2.3 Condition-Code Register (CCR)

This 8-bit register contains internal CPU status information, including an interrupt mask l
half-carry (H), negative (N), zero (Z), overflow (V), and carry (C) flags. The I bit is initia
by reset exception-handling sequence, but other bits are not initialized.

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

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

				When the ADD.B, ADDX.B, SUB.B, SUBX.B, or NEG.B instruction is executed, this flag is s there is a carry or borrow at bit 3, and cleared otherwise. When the ADD.W, SUB.W, CMP.W NEG.W instruction is executed, the H flag is s there is a carry or borrow at bit 11, and cleare otherwise. When the ADD.L, SUB.L, CMP.L, instruction is executed, the H flag is set to 1 if carry or borrow at bit 27, and cleared to 0 oth
4	U	Undefined	R/W	User Bit Can be written and read by software using the STC, ANDC, ORC, and XORC instructions.
3	N	Undefined	R/W	Negative Flag Stores the value of the most significant bit of sign bit.
2	Z	Undefined	R/W	Zero Flag Set to 1 to indicate zero data, and cleared to indicate non-zero data.
1	V	Undefined	R/W	Overflow Flag Set to 1 when an arithmetic overflow occurs, cleared to 0 at other times.
0	C	Undefined	R/W	Carry Flag Set to 1 when a carry occurs, and cleared to otherwise. Used by: • Add instructions, to indicate a carry • Subtract instructions, to indicate a borrow • Shift and rotate instructions, to indicate a The carry flag is also used as a bit accumulat manipulation instructions.

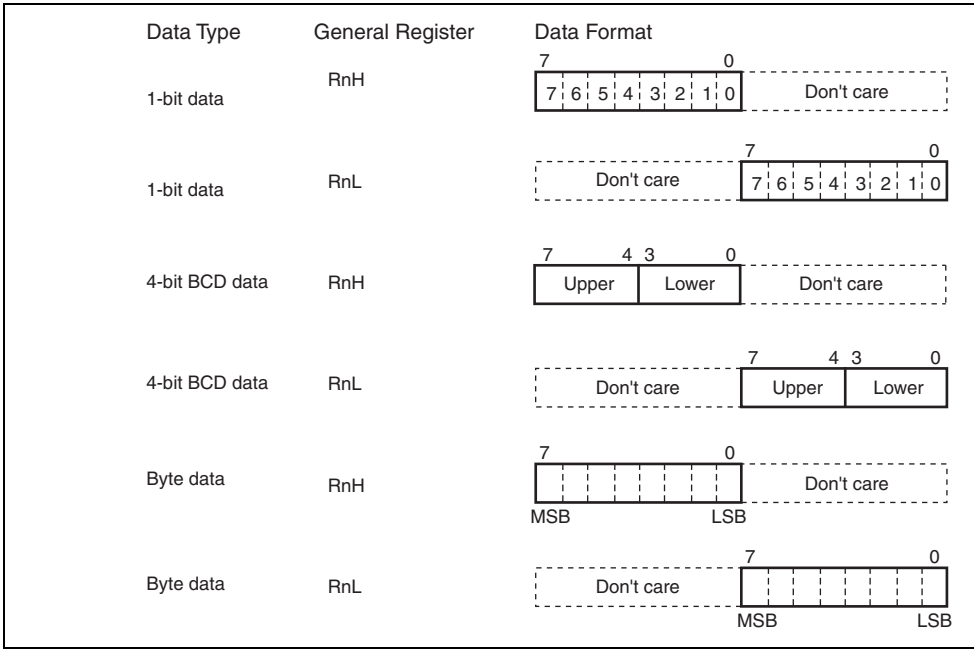
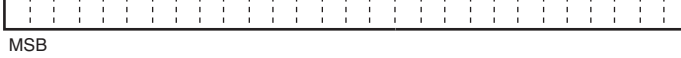


Figure 2.5 General Register Data Formats (1)



Legend

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

Figure 2.5 General Register Data Formats (2)

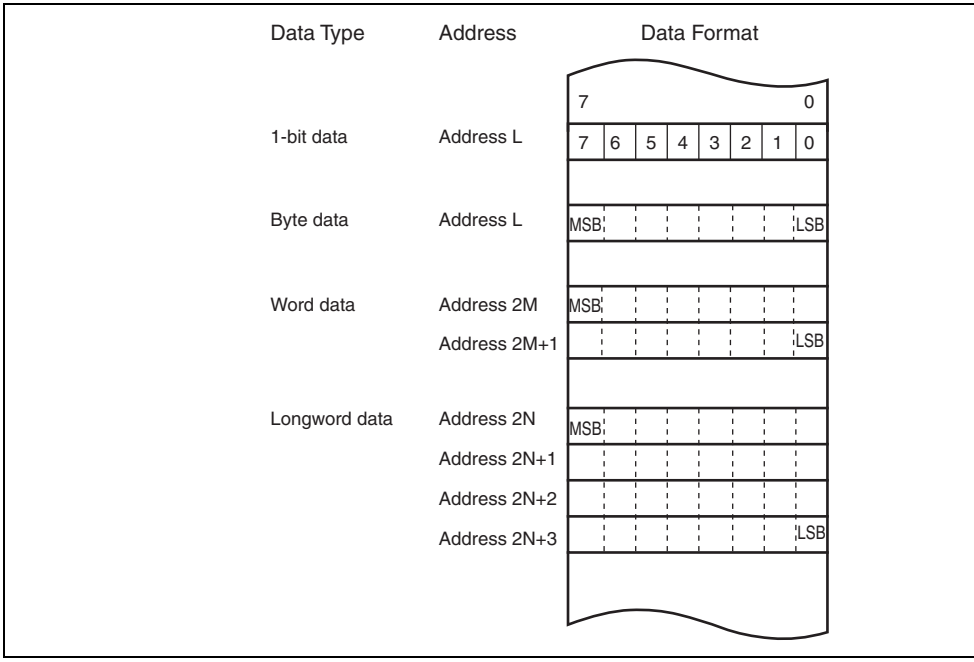


Figure 2.6 Memory Data Formats

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

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 t MOV.W @SP+, Rn. POP.L ERn is identical to MOV.L @SP+, E
PUSH	W/L	Rn → @–SP Pushes a general register onto the stack. PUSH.W Rn is identi MOV.W Rn, @–SP. PUSH.L ERn is identical to MOV.L ERn, @

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

DEC		Increments or decrements a general register by 1 or 2. (Byte c 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-b
DAA DAS	B	Rd (decimal adjust) → Rd Decimal-adjusts an addition or subtraction result in a general r referring to the CCR to produce 4-bit BCD data.
MULXU	B/W	Rd × Rs → Rd Performs unsigned multiplication on data in two general regist 8 bits × 8 bits → 16 bits or 16 bits × 16 bits → 32 bits.
MULXS	B/W	Rd × Rs → Rd Performs signed multiplication on data in two general registers bits × 8 bits → 16 bits or 16 bits × 16 bits → 32 bits.
DIVXU	B/W	Rd ÷ Rs → Rd Performs unsigned division on data in two general registers: e bits ÷ 8 bits → 8-bit quotient and 8-bit remainder or 32 bits ÷ 1 16-bit quotient and 16-bit remainder.

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

		general register.
EXTU	W/L	Rd (zero extension) → Rd Extends the lower 8 bits of a 16-bit register to word size, or the l bits of a 32-bit register to longword size, by padding with zeros o left.
EXTS	W/L	Rd (sign extension) → Rd Extends the lower 8 bits of a 16-bit register to word size, or the l bits of a 32-bit register to longword size, by extending the sign b

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

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

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

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

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

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

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

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

		code register size is one byte, but in transfer to memory, data by word access.
ANDC	B	CCR ∧ #IMM → CCR Logically ANDs the CCR with immediate data.
ORC	B	CCR ∨ #IMM → CCR Logically ORs the CCR with immediate data.
XORC	B	CCR ⊕ #IMM → CCR 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

	else next;
	Transfers a data block. Starting from the address set in ER5, tr data for the number of bytes set in R4L or R4 to the address loc in ER6.
	Execution of the next instruction begins as soon as the transfer completed.

Some instructions have two operation fields.

- Register Field

  Specifies a general register. Address registers are specified by 3 bits, and data registe
  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. A
  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.

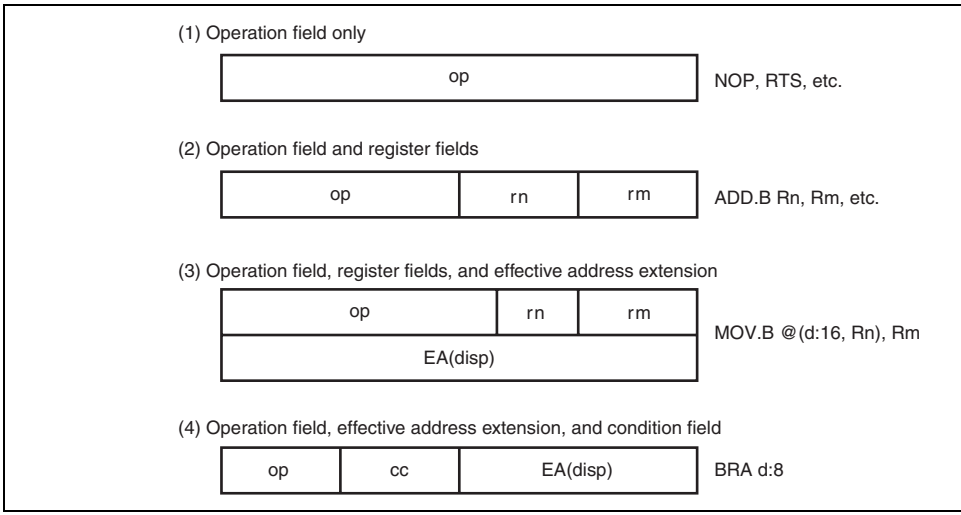


Figure 2.7 Instruction Formats

Arithmetic and logic instructions can use the register direct and immediate modes. Data t
instructions can use all addressing modes except program-counter relative and memory in
Bit-manipulation instructions use register direct, register indirect, or the absolute addressi
(@aa:8) to specify an operand, and register direct (BSET, BCLR, BNOT, and BTST instr
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

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

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

- Register indirect with post-increment—@ERn+

  The register field of the instruction code specifies an address register (ERn) the lowe
  of which contains the address of a memory operand. After the operand is accessed, 1
  added to the address register contents (32 bits) and the sum is stored in the address r
  The value added is 1 for byte access, 2 for word access, or 4 for longword access. Fo
  or longword access, the register value should be even.
- Register indirect with pre-decrement—@-ERn

  The value 1, 2, or 4 is subtracted from an address register (ERn) specified by the reg
  in the instruction code, and the lower 24 bits of the result is the address of a memory
  The result is also stored in the address register. The value subtracted is 1 for byte acc
  word access, or 4 for longword access. For the word or longword access, the register
  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 a
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 1
absolute address the upper 8 bits are a sign extension. A 24-bit absolute address can acc
entire address space.

(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 operand.

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

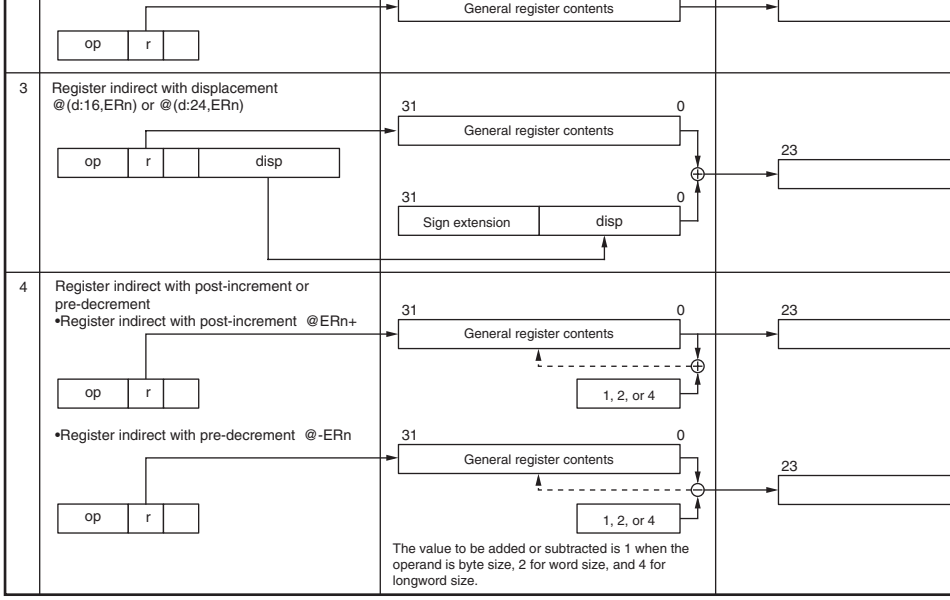
(7) Program-Counter Relative—@(d:8, PC) or @(d:16, PC)

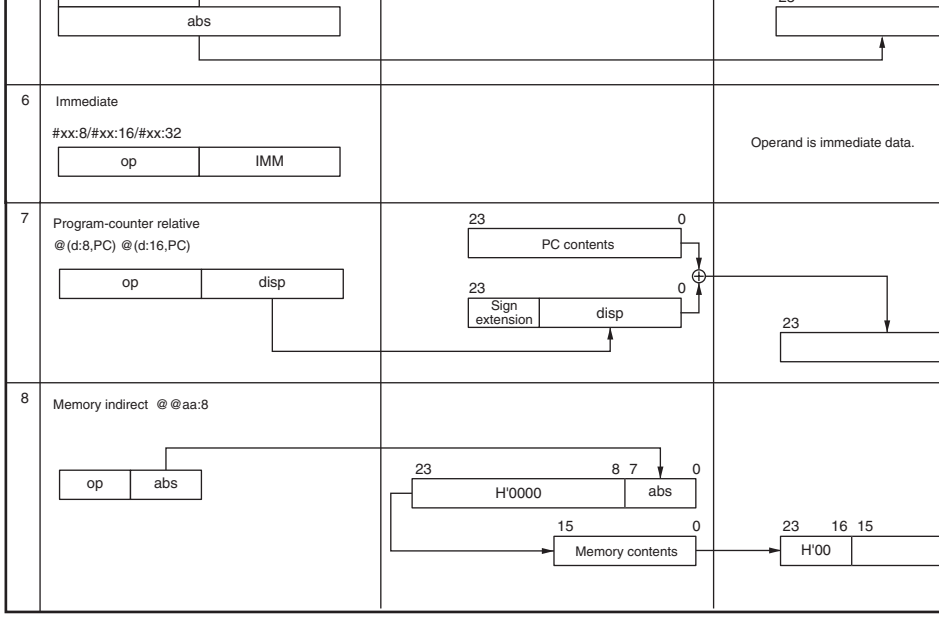
This mode is used in the BSR instruction. An 8-bit or 16-bit displacement contained in th instruction is sign-extended and added to the 24-bit PC contents to generate a branch add PC value to which the displacement is added is the address of the first byte of the next ins so the possible branching range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

(8) Memory Indirect—@@aa:8

This mode can be used by the JMP and JSR instructions. The instruction code contains a absolute address specifying a memory operand. This memory operand contains a branch The memory operand is accessed by longword access. The first byte of the memory opera ignored, generating a 24-bit branch address. Figure 2.8 shows how to specify branch addr memory indirect mode. The upper bits of the absolute address are all assumed to be 0, so 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.


General register contents
op
r
3
Register indirect with displacement
@(d:16,ERn) or @(d:24,ERn)
31
0
General register contents
op
r
disp
23
31
0
Sign extension
disp
4
Register indirect with post-increment or
pre-decrement
•Register indirect with post-increment @ERn+
31
0
General register contents
23
op
r
1, 2, or 4
•Register indirect with pre-decrement @-ERn
31
0
General register contents
23
op
r
1, 2, or 4
The value to be added or subtracted is 1 when the operand is byte size, 2 for word size, and 4 for longword size.



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

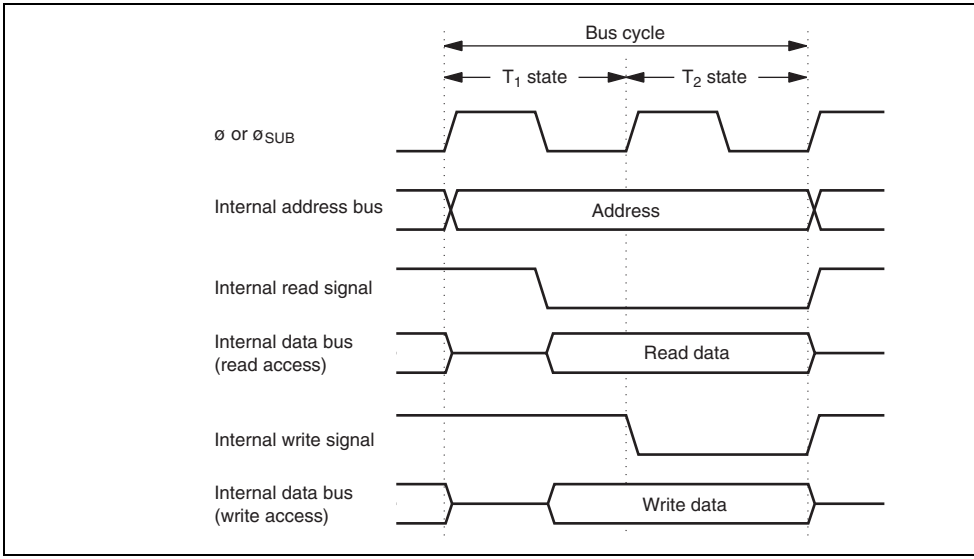


Figure 2.9 On-Chip Memory Access Cycle

module.

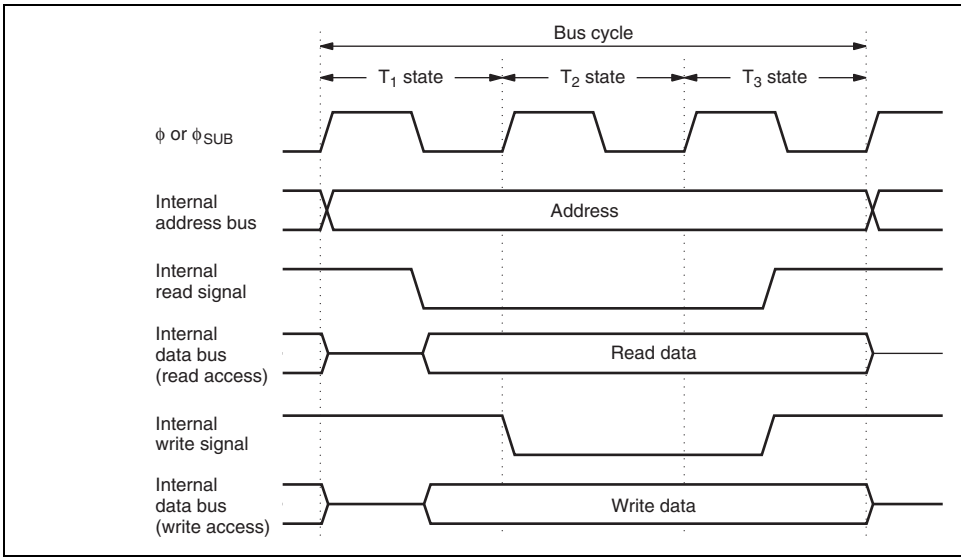


Figure 2.10 On-Chip Peripheral Module Access Cycle (3-State Access)

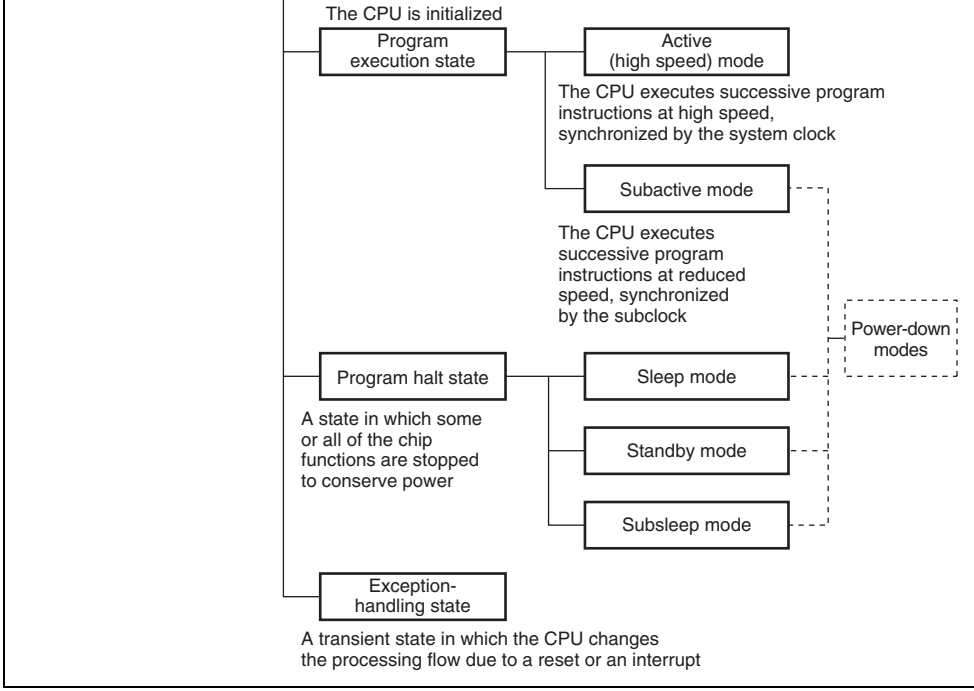


Figure 2.11 CPU Operation States

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 o
I/O registers areas available to the user. When data is transferred from CPU to empty are
transferred data will be lost. This action may also cause the CPU to malfunction. When
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
which starts from the address indicated by R5, to the address indicated by R6. Set R4L a
that the end address of the destination address (value of R6 + R4L) does not exceed H'F
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 add
byte units, manipulate the data of the target bit, and write data to the same address again
units. Special care is required when using these instructions in cases where two registers
assigned to the same address, or when a bit is directly manipulated for a port or a registe
containing a write-only bit, because this may rewrite data of a bit other than the bit to be
manipulated.

C. 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 load register. As a result, bits other than the intended bit in the timer counter may be m and the modified value may be written to the timer load register.

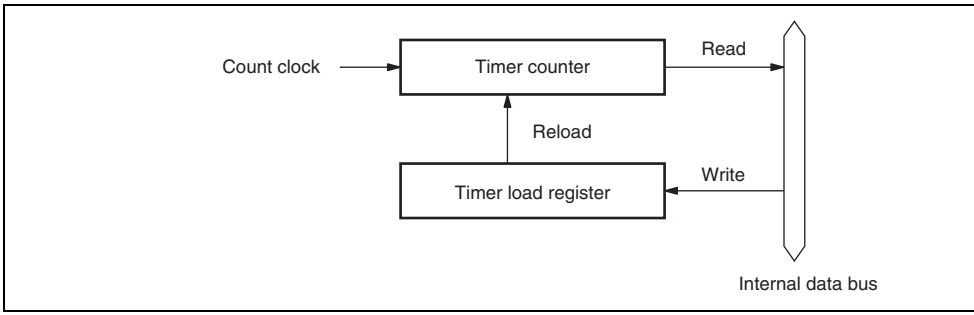


Figure 2.13 Example of Timer Configuration with Two Registers Allocated to S Address

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

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

The BSET instruction is executed for port 5.

— After executing BSET instruction

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

— Description on operation

1. 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 level input).

   P55 to P50 are output pins, so the CPU reads the value in PDR5. In this exam has a value of H'80, but the value read by the CPU is H'40.
2. Next, the CPU sets bit 0 of the read data to 1, changing the PDR5 data to H'4
3. Finally, the CPU writes H'41 to PDR5, completing execution of BSET instru

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

PDR5	1	0	0	0	0	0	0
RAM0	1	0	0	0	0	0	0

— BSET instruction executed

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

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

— After executing BSET instruction

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

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

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

Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level
PCR5	0	0	1	1	1	1	1
PDR5	1	0	0	0	0	0	0

— BCLR instruction executed

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

The BCLR instruction is executed for PCR5.

— After executing BCLR instruction

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

— Description on operation

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

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

C. 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 p
However, bits 7 and 6 in PCR5 change to 1, so that P57 and P56 change from
to output pins. To prevent this problem, store a copy of the PDR5 data in a w
memory and manipulate data of the bit in the work area, then write this data t

PDR5	1	0	0	0	0	0	0
RAM0	0	0	1	1	1	1	1

— BCLR instruction executed

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

The BCLR instructions executed for the PCR5 wo
(RAM0).

— After executing BCLR instruction

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

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

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

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, specified in the instruction code. Exception handling can be executed at all times in t program execution state, regardless of the setting of the I bit in CCR.

- Interrupts

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

—	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
RTC	Overflow	19	H'0026 to H'0027
—	Reserved for system use	20	H'0028 to H'0029
Timer V	Timer V compare match A Timer V compare match B Timer V overflow	22	H'002C to H'002D
SCI3	SCI3 receive data full SCI3 transmit data empty SCI3 transmit end SCI3 receive error	23	H'002E to H'002F

	Timer Z overflow		
	Compare match/input capture A1 to D1 Timer Z overflow Timer Z underflow	27	H'0036 to H'0037
Timer B1	Timer B1 overflow	29	H'003A to H'003B
SCI3_2	Receive data full Transmit data empty Transmit end Receive error	32	H'0040 to H'0041

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 enable register 2 (IENR2)
- Interrupt flag register 1 (IRR1)
- Interrupt flag register 2 (IRR2)
- Wakeup interrupt flag register (IWPR)

6 to 4	—	All 1	—	Reserved These bits are always read as 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

				0: Falling edge of WKP5(ADTRG) pin input is d
				1: Rising edge of WKP5(ADTRG) pin input is d
4	WPEG4	0	R/W	WKP4 Edge Select
				0: Falling edge of WKP4 pin input is detected
				1: Rising edge of WKP4 pin input is detected
3	WPEG3	0	R/W	WKP3 Edge Select
				0: Falling edge of WKP3 pin input is detected
				1: Rising edge of WKP3 pin input is detected
2	WPEG2	0	R/W	WKP2 Edge Select
				0: Falling edge of WKP2 pin input is detected
				1: Rising edge of WKP2 pin input is detected
1	WPEG1	0	R/W	WKP1Edge Select
				0: Falling edge of WKP1 pin input is detected
				1: Rising edge of WKP1 pin input is detected
0	WPEG0	0	R/W	WKP0 Edge Select
				0: Falling edge of WKP0 pin input is detected
				1: Rising edge of WKP0 pin input is detected

				enabled.
5	IENWP	0	R/W	Wakeup Interrupt Enable This bit is an enable bit, which is common to the $\overline{\text{WKP5}}$ to $\overline{\text{WKP0}}$. When the bit is set to 1, interru 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 are enabled.
2	IEN2	0	R/W	IRQ2 Interrupt Enable When this bit is set to 1, interrupt requests of the are enabled.
1	IEN1	0	R/W	IRQ1 Interrupt Enable When this bit is set to 1, interrupt requests of the are enabled.
0	IEN0	0	R/W	IRQ0 Interrupt Enable When this bit is set to 1, interrupt requests of the are enabled.

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

4 to 0	—	All 1	—	Reserved These bits are always read as 1.

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

3.2.5 Interrupt Flag Register 1 (IRR1)

IRR1 is a status flag register for direct transition interrupts, RTC interrupts, and $\overline{\text{IRQ3}}$ to 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 S instruction while DTON in SYSCR2 is set to 1. [Clearing condition] When IRRDT is cleared by writing 0

				[Setting condition] When $\overline{\text{IRQ3}}$ pin is designated for interrupt input a 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 a 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 a designated signal edge is detected. [Clearing condition] When IRRI1 is cleared by writing 0
0	IRRI0	0	R/W	IRQ0 Interrupt Request Flag [Setting condition] When $\overline{\text{IRQ0}}$ pin is designated for interrupt input a designated signal edge is detected. [Clearing condition] When IRRI0 is cleared by writing 0

				When the timer B1 counter value overflows [Clearing condition] When IRRTB1 is cleared by writing 0
4 to 0	—	All 1	—	Reserved These bits are always read as 1.

3.2.7 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, 6	—	All 1	—	Reserved 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 inpu 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 inpu designated signal edge is detected. [Clearing condition] When IWPF4 is cleared by writing 0.

				When $\overline{\text{WKP2}}$ pin is designated for interrupt input 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 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 designated signal edge is detected. [Clearing condition] When IWPF0 is cleared by writing 0.

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'000 data in that address is sent to the program counter (PC) as the start address, and prog execution starts from that address.

3.4 Interrupt Exception Handling

3.4.1 External Interrupts

As the external interrupts, there are NMI, IRQ3 to IRQ0, and WKP5 to WKP0 interrupt

(1) NMI Interrupt

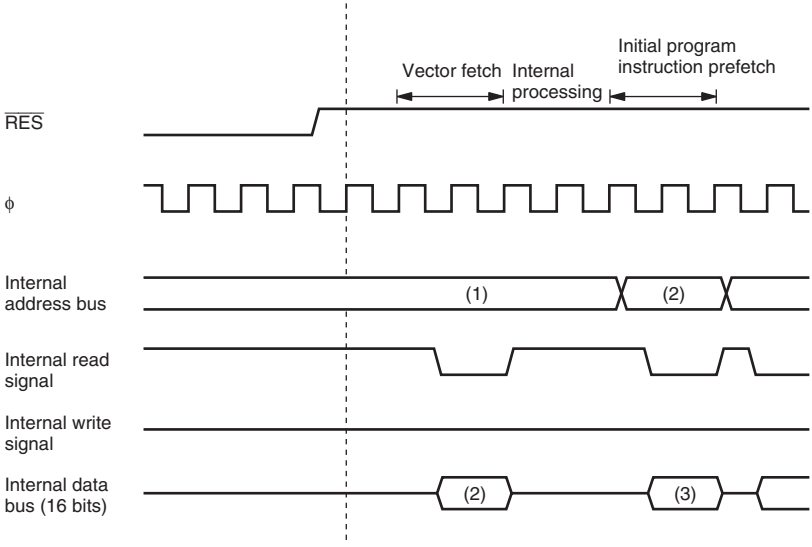
NMI interrupt is requested by input signal edge to pin $\overline{\text{NMI}}$. This interrupt is detected by rising edge sensing or falling edge sensing, depending on the setting of bit NMIEG in IE

NMI is the highest-priority interrupt, and can always be accepted without depending on 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 fou interrupts are given different vector addresses, and are detected individually by either ris sensing or falling edge sensing, depending on the settings of bits IEG3 to IEG0 in IEGR

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



(1) Reset exception handling vector address (H'0000)
(2) Program start address
(3) Initial program instruction

Figure 3.1 Reset Sequence

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 re signal is sent to the interrupt controller.
2. When multiple interrupt requests are generated, the interrupt controller requests to th the interrupt handling with the highest priority at that time according to table 3.1. Ot interrupt requests are held pending.
3. The CPU accepts the NMI and address break without depending on the I bit value. C interrupt requests are accepted, if the I bit is cleared to 0 in CCR; if the I bit is set to interrupt request is held pending.
4. If the CPU accepts the interrupt after processing of the current instruction is complet interrupt exception handling will begin. First, both PC and CCR are pushed onto the state of the stack at this time is shown in figure 3.2. The PC value pushed onto the st 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 break. Upon return from interrupt handling, the values of I bit and other bits in CCR 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, transfers the address to PC as a start address of the interrupt handling-routine. Then 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 F 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 fir 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

3.5.2 Notes on Stack Area Use

When word data is accessed, the least significant bit of the address is regarded as 0. Acce
stack always takes place in word size, so the stack pointer (SP: R7) should never indicate
address. Use PUSH Rn (MOV.W Rn, @–SP) or POP Rn (MOV.W @SP+, Rn) to save or
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, $\overline{\text{I}}$
$\overline{\text{IRQ0}}$, and $\overline{\text{WKP5}}$ to $\overline{\text{WKP0}}$, the interrupt request flag may be set to 1.

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

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

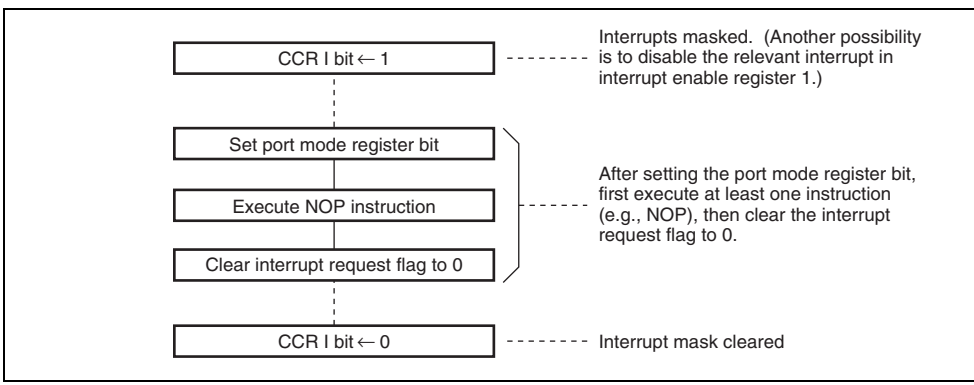
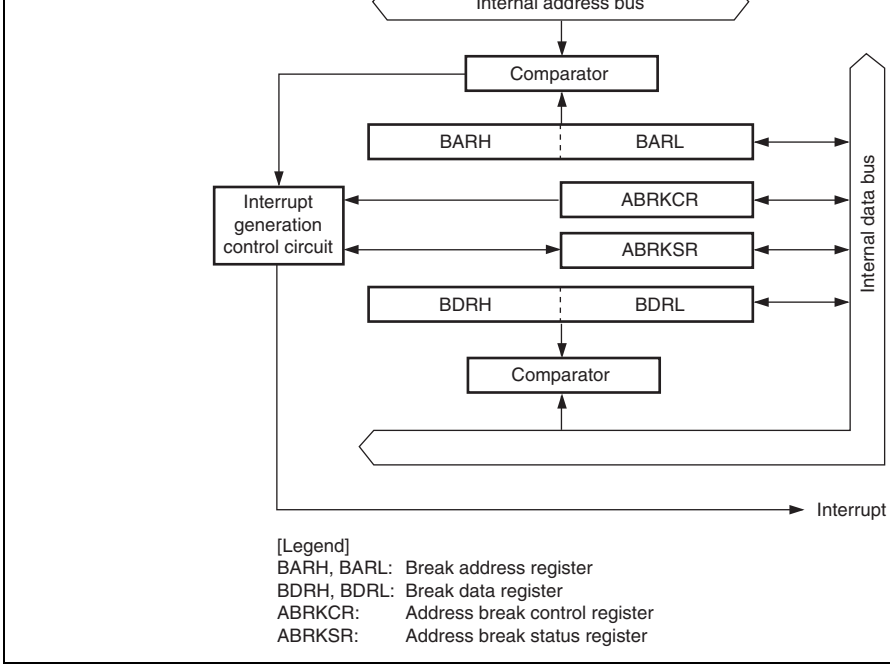


Figure 3.4 Port Mode Register Setting and Interrupt Request Flag Clearing Pro



[Legend]
BARH, BARL: Break address register
BDRH, BDRL: Break data register
ABRKCR: Address break control register
ABRKSR: Address break status register

Figure 4.1 Block Diagram of Address Break

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 instructi be executed. When this bit is 1, the interrupt is n 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 set the comparison condition between 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	

[Legend]
X: Don't care.

When an address break is set in the data read cycle or data write cycle, the data bus used depend on the combination of the byte/word access and address. Table 4.1 shows the ac data bus used. When an I/O register space with an 8-bit data bus width is accessed in wo byte access is generated twice. For details on data widths of each register, see section 19 Register Addresses (Address Order).

Table 4.1 Access and Data Bus Used

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

				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 rec 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 ad break interrupt. When setting the address break condition to the instruction execution cy 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 addres interrupt. BDRH is compared with the upper 8-bit data bus. BDRL is compared with the bit data bus. When memory or registers are accessed by byte, the upper 8-bit data bus is even and odd addresses in the data transmission. Therefore, comparison data must be set BDRH for byte access. For word access, the data bus used depends on the address. See s 4.1.1, Address Break Control Register (ABRKCR), for details. The initial value of this re undefined.

When the address break is specified in instruction execution cycle

Register setting
- ABRKCR = H'80
- 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.

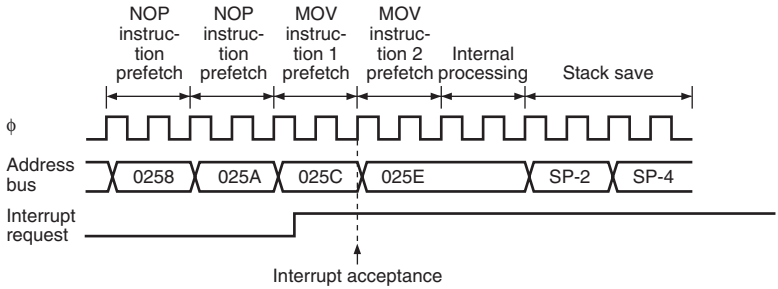


Figure 4.2 Address Break Interrupt Operation Example (1)

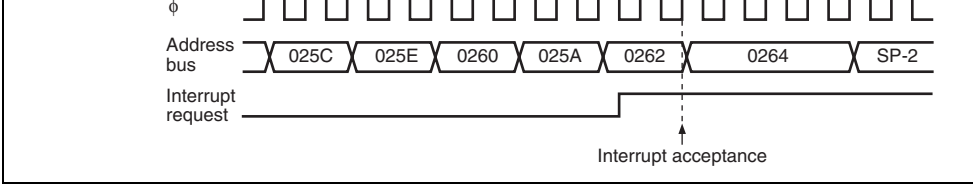


Figure 4.2 Address Break Interrupt Operation Example (2)

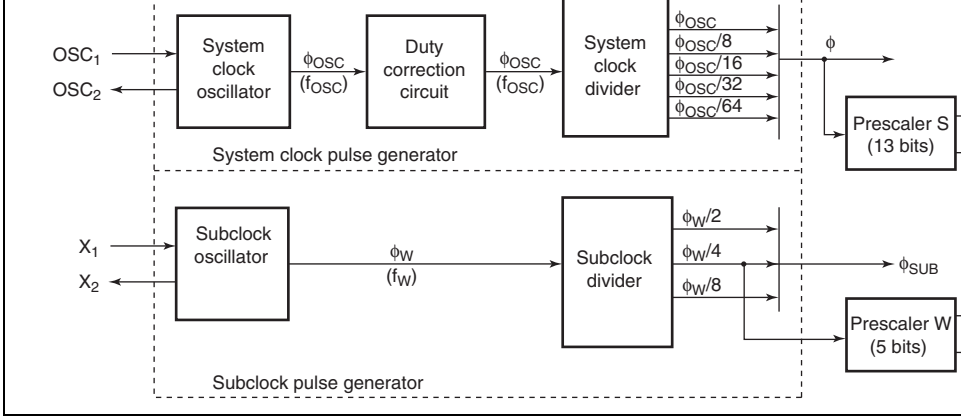


Figure 5.1 Block Diagram of Clock Pulse Generators

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

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-r
crystal resonator should be used. Figure 5.4 shows the equivalent circuit of a crystal reson
resonator having the characteristics given in table 5.1 should be used.

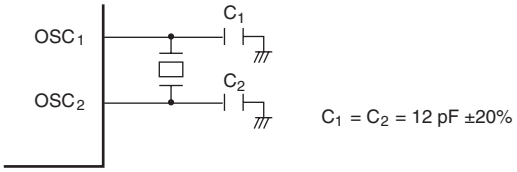


Figure 5.3 Typical Connection to Crystal Resonator

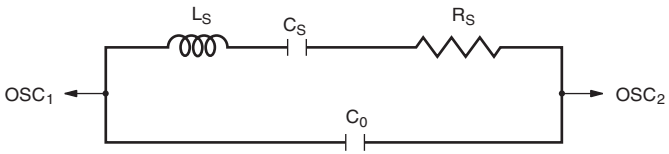


Figure 5.4 Equivalent Circuit of Crystal Resonator

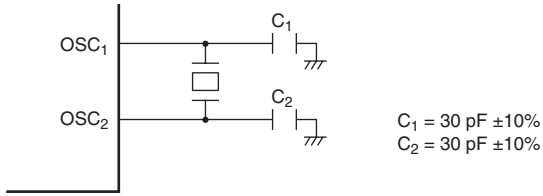


Figure 5.5 Typical Connection to Ceramic Resonator

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 show
connection. The duty cycle of the external clock signal must be 45 to 55%.

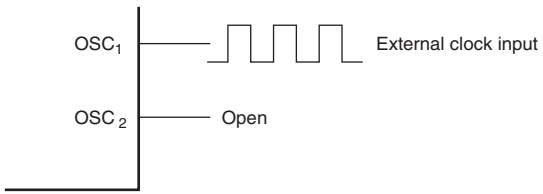


Figure 5.6 Example of External Clock Input

Note : Registance 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.

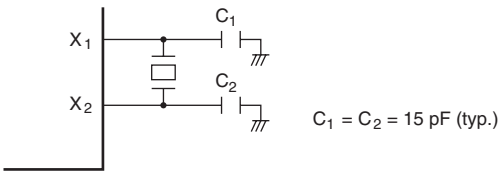
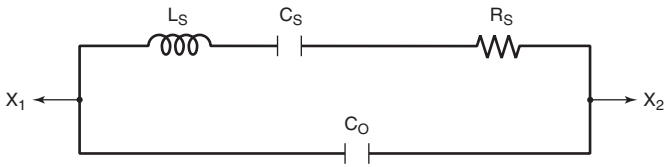


Figure 5.8 Typical Connection to 32.768-kHz Crystal Resonator



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

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 (ø) as its input clock. It is increme
per clock period. Prescaler S is initialized to H'0000 by a reset, and starts counting on ex
the reset state. In standby mode, subactive mode, and subsleep mode, the system clock p
generator stops. Prescaler S also stops and is initialized to H'0000. The CPU cannot read
prescaler S. The output from prescaler S is shared by the on-chip peripheral modules. Th
ratio can be set separately for each on-chip peripheral function. In active mode and sleep
the clock input to prescaler S is determined by the division factor designated by MA2 to
SYSCR2.

5.3.2 Prescaler W

Prescaler W is a 5-bit counter using a 32.768 kHz signal divided by 4 ($ø_W/4$) as its input
divided output is used for clock time base operation of timer A. Prescaler W is initialize
by a reset, and starts counting on exit from the reset state. Even in standby mode, subact
or subsleep mode, prescaler W continues functioning so long as clock signals are suppli
X_1 and X_2.

5.4.2 Notes on Board Design

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

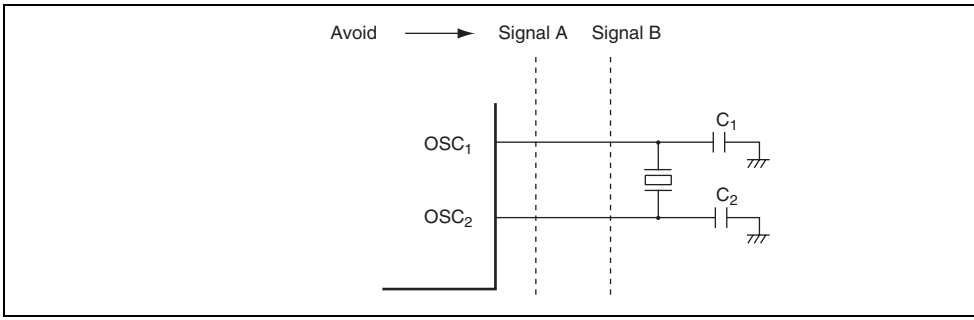


Figure 5.11 Example of Incorrect Board Design

The CPU and all on-chip peripheral modules are operable on the subclock. The subc
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 functio
  selected, the RTC is operable.
- Module standby mode

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

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 exec the SLEEP instruction. 0: Enters sleep mode or subsleep mode. 1: Enters 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 perip modules wait for stable clock operation after exit standby mode, subactive mode, or subsleep mo active mode or sleep mode due to an interrupt. T designation should be made according to the clo frequency so that the waiting time is at least 6.5 relationship between the specified value and the of wait states is shown in table 6.1. When an ext clock is to be used, the minimum value (STS2 = STS0 =1) is recommended.
5	STS1	0	R/W	
4	STS0	0	R/W	

2 to 0	—	All 0	—	Reserved These bits are always read as 0.

Table 6.1 Operating Frequency and Waiting Time

Bit Name				Operating Frequency			
STS2	STS1	STS0	Waiting Time	18 MHz	16 MHz	10 MHz	8 MHz
0	0	0	8,192 states	0.4	0.5	0.8	1.0
		1	16,384 states	0.9	1.0	1.6	2.0
	1	0	32,768 states	1.8	2.0	3.3	4.1
		1	65,536 states	3.6	4.1	6.6	8.2
1	0	0	131,072 states	7.2	8.2	13.1	16.4
		1	1,024 states	0.05	0.06	0.10	0.13
	1	0	128 states	0.00	0.00	0.01	0.02
		1	16 states	0.00	0.00	0.00	0.00

Note: Time unit is ms.

				a SLEEP instruction, as well as bit SSBY of SYS 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 i
2	MA0	0	R/W	and sleep modes. The operating clock frequency changes to the set frequency after the SLEEP in is executed. 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 i subactive and subsleep modes. The operating cl frequency changes to the set frequency after the instruction is executed. 00: $\phi_w/8$ 01: $\phi_w/4$ 1X: $\phi_w/2$

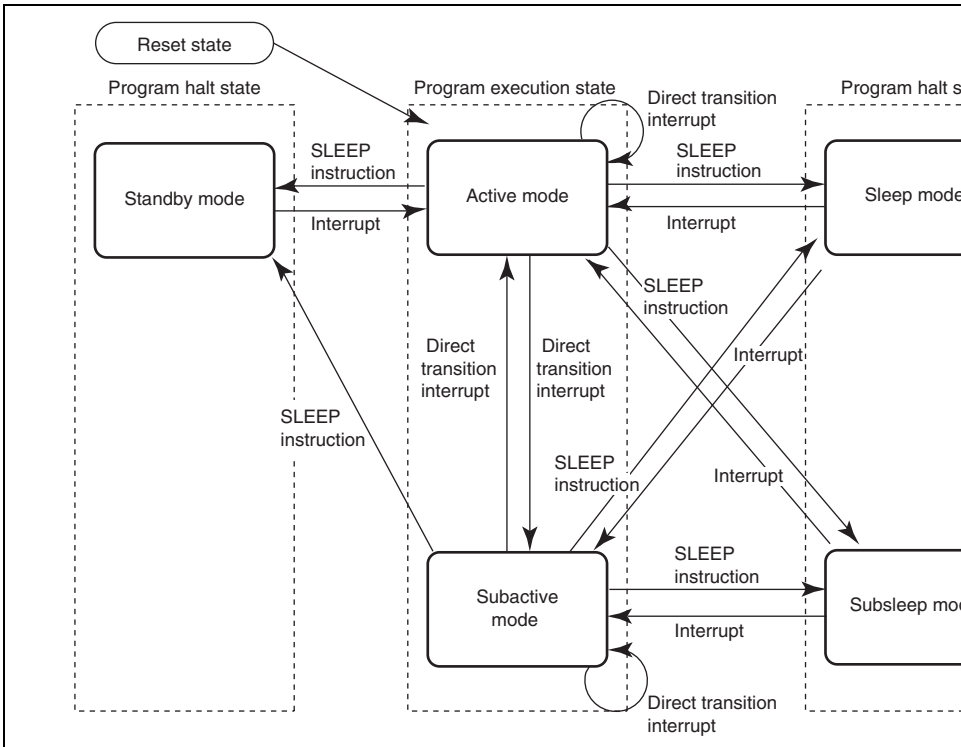
[Legend]

X: Don't care.

5	MSTS3	0	R/W	SCI3 Module Standby SCI3 enters standby mode when this bit is set t
4	MSTAD	0	R/W	A/D Converter Module Standby A/D converter enters standby mode when this b 1
3	MSTWD	0	R/W	Watchdog Timer Module Standby Watchdog timer enters standby mode when this to 1.When the internal oscillator is selected for watchdog timer clock, the watchdog timer opera regardless of the setting of this bit
2	—	0	—	Reserved This bit is always read as 0.
1	MSTTV	0	R/W	Timer V Module Standby Timer V enters standby mode when this bit is s
0	MSTTA	0	R/W	RTC Module Standby RTC enters standby mode when this bit is set t

4	MSTTB1	0	R/W	Timer B1 Module Standby Timer B1 enters standby mode when this bit is s
3, 2	—	All 0	—	Reserved These bits are always read as 0.
1	MSTTZ	0	R/W	Timer Z Module Standby Timer Z enters standby mode when this bit is se
0	MSTPWM	0	R/W	PWM Module Standby PWM enters standby mode when this bit is set to

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 in
is accepted.
2. Details on the mode transition conditions are given in table 6.2.

Figure 6.1 Mode Transition Diagram

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, SCI3_2 A/D converter are reset, and all registers are set to their initial values. To use t functions after entering active mode, reset the registers.

						are ret output impeda
External interrupts	IRQ3 to IRQ0	Functioning	Functioning	Functioning	Functioning	Functio
	WKP5 to WKP0	Functioning	Functioning	Functioning	Functioning	Functio
Peripheral functions	RTC	Functioning	Functioning	Functioning if the timekeeping time-ba is selected, and retained if not selecte		
	Timer V	Functioning	Functioning	Reset	Reset	Reset
	Watchdog timer	Functioning	Functioning	Retained (functioning if the internal os selected as a count clock*)		
	SCI3, SCI3_2	Functioning	Functioning	Reset	Reset	Reset
	IIC2	Functioning	Functioning	Retained*	Retained	Retain
	Timer B1	Functioning	Functioning	Retained*	Retained	Retain
	Timer Z	Functioning	Functioning	Retained (the counter increments acc subclocks if the internal clock (ϕ) is sel count clock*)		
	A/D converter	Functioning	Functioning	Reset	Reset	Reset

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

6.2.2 Standby Mode

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

Standby mode is cleared by an interrupt. When an interrupt is requested, the system clock generator starts. After the time set in bits STS2 to STS0 in SYSCR1 has elapsed, and inte exception handling starts. Standby mode is not cleared if the I bit of CCR is set to 1 or th 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 are supplied to the entire chip as soon as the system clock pulse generator starts functioni $\overline{\text{RES}}$ pin must be kept low until the pulse generator output stabilizes. After the pulse gene output has stabilized, the CPU starts reset exception handling if the $\overline{\text{RES}}$ pin is driven hig

6.2.3 Subsleep Mode

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

Subsleep mode is cleared by an interrupt. When an interrupt is requested, subsleep mode and interrupt exception handling starts. Subsleep mode is not cleared if the I bit of CCR i or the requested interrupt is disabled in the interrupt enable register. After subsleep mode

The operating frequency of subactive mode is selected from $\phi_w/2$, $\phi_w/4$, and $\phi_w/8$ by the SA0 bits in SYSCR2. After the SLEEP instruction is executed, the operating frequency the frequency which is set before the execution. When the SLEEP instruction is execute subactive mode, a transition to sleep mode, subsleep mode, standby mode, active mode, subactive mode is made, depending on the combination of SYSCR1 and SYSCR2. Whe pin goes low, the system clock pulse generator starts. Since system clock signals are sup the entire chip as soon as the system clock pulse generator starts functioning, the $\overline{\text{RES}}$ p kept low until the pulse generator output stabilizes. After the pulse generator output has 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 in SYSCR2. The operating frequency changes to the set frequency after SLEEP instruct execution.

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 h
(the direct transition time) is calculated by equation (1).

Direct transition time = {(number of SLEEP instruction execution states) + (number of in
processing states)}× (tcyc before transition) + (number of interrupt exception handling sta
(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 (•) cycle time
tsubcyc: Subclock (•SUB) cycle time

(when the CPU operating clock of $\phi_w/8 \rightarrow \phi_{osc}$ and a waiting time of 8192 states are s

[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 m
clock supply to modules stops to enter the power-down mode. Module standby mode en
on-chip peripheral module to enter the standby state by setting a bit that corresponds to
module to 1 and cancels the mode by clearing the bit to 0.

- 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 progra into the chip is started to erase or program of the entire flash memory. In normal us mode, individual blocks can be erased or programmed.
- Programmer mode

  Flash memory can be programmed/erased in programmer mode using a PROM prog 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 m 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 r memory can be read with low power consumption.

7.1 Block Configuration

Figure 7.1 shows the block configuration of flash memory. The thick lines indicate eras the narrow lines indicate programming units, and the values are addresses. The 56-kbyt memory is divided into 1 kbyte × 4 blocks, 28 kbytes × 1 block, 16 kbytes × 1 block, an × 1 block. Erasing is performed in these units. Programming is performed in 128-byte u starting from an address with lower eight bits H'00 or H'80.

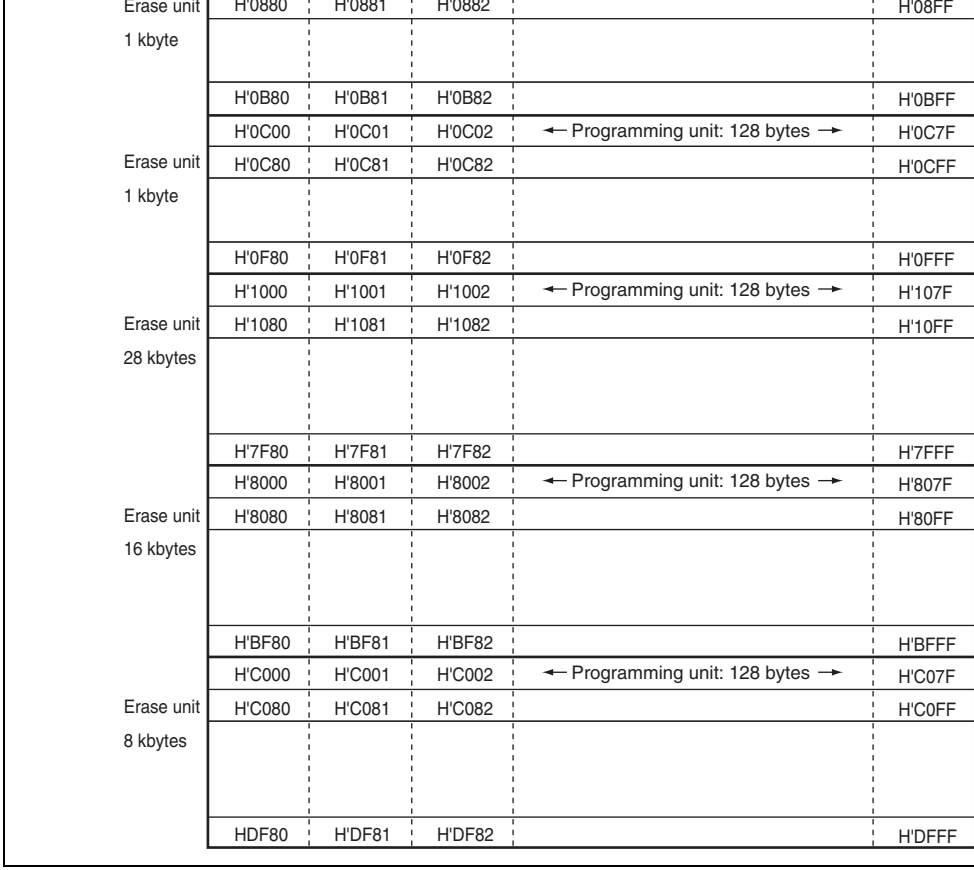


Figure 7.1 Flash Memory Block Configuration

7.2.1 Flash Memory Control Register 1 (FLMCR1)

FLMCR1 is a register that makes the flash memory change to program mode, program-v
mode, erase mode, or erase-verify mode. For details on register setting, refer to section
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 to 0, other FLMCR1 register bits and all EBR1 b be set.
5	ESU	0	R/W	Erase Setup When this bit is set to 1, the flash memory char erase setup state. When it is cleared to 0, the e setup state is cancelled. Set this bit to 1 before E bit to 1 in FLMCR1.
4	PSU	0	R/W	Program Setup When this bit is set to 1, the flash memory char program setup state. When it is cleared to 0, th setup state is cancelled. Set this bit to 1 before the P bit in FLMCR1.
3	EV	0	R/W	Erase-Verify When this bit is set to 1, the flash memory char erase-verify mode. When it is cleared to 0, era mode is cancelled.

				When this bit is set to 1 while SWE=1 and PSU= flash memory changes to program mode. When 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. FLM 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 op on flash memory (programming or erasing). Wh is set to 1, flash memory goes to the error-protec state. See section 7.5.3, Error Protection, for details.
6 to 0	—	All 0	—	Reserved These bits are always read as 0.

6	EB6	0	R/W	When this bit is set to 1, 8 bytes of H'C000 to H be erased.
5	EB5	0	R/W	When this bit is set to 1, 16 bytes of H'8000 to H will be erased.
4	EB4	0	R/W	When this bit is set to 1, 28 kbytes of H'1000 to will be erased.
3	EB3	0	R/W	When this bit is set to 1, 1 kbyte of H'0C00 to H be erased.
2	EB2	0	R/W	When this bit is set to 1, 1 kbyte of H'0800 to H be erased.
1	EB1	0	R/W	When this bit is set to 1, 1 kbyte of H'0400 to H be erased.
0	EB0	0	R/W	When this bit is set to 1, 1 kbyte of H'0000 to H be erased.

				When this bit is 0 and a transition is made to sub mode, the flash memory enters the power-down When this bit is 1, the flash memory remains in t normal mode even after a transition is made to s 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 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 this bit is set to 1. Flash memory control register be accessed when this bit is set to 0.
6 to 0	—	All 0	—	Reserved These bits are always read as 0.

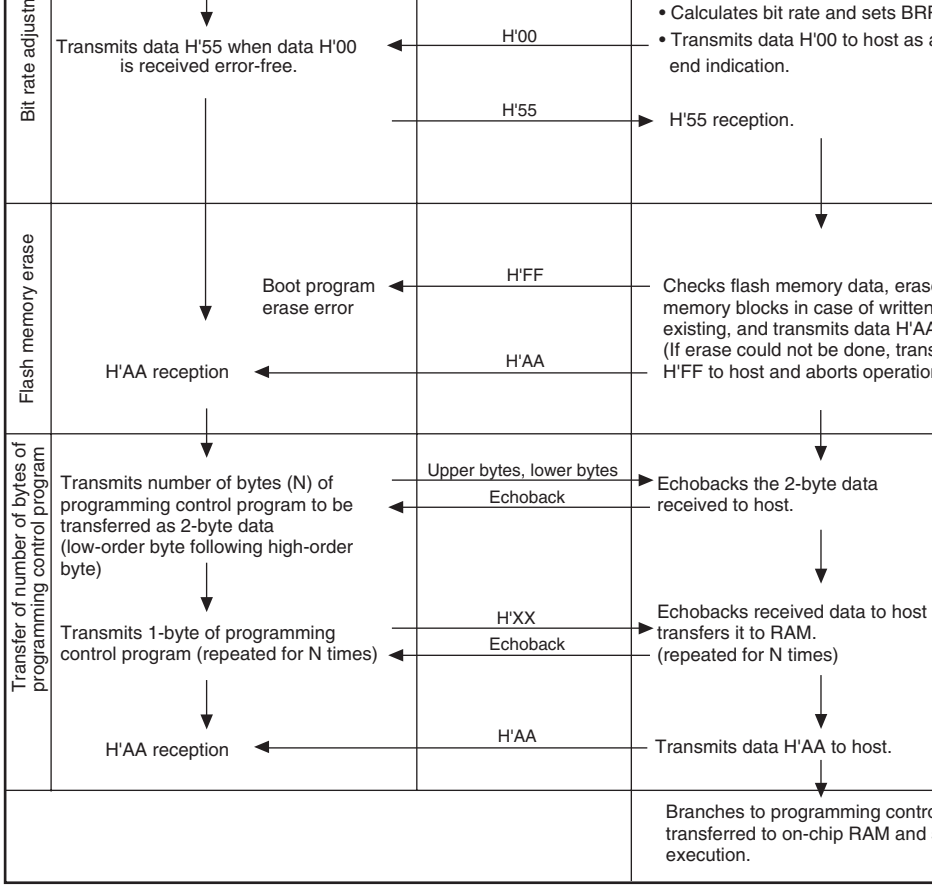
via SCI3. After erasing the entire flash memory, the programming control program is ex
This can be used for programming initial values in the on-board state or for a forcible re
programming/erasing can no longer be done in user program mode. In user program mo
individual blocks can be erased and programmed by branching to the user program/erase
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.

3. When the boot program is initiated, the chip measures the low-level period of asynchr 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 m of the host. The reset should end with the RxD pin high. The RxD and TxD pins sho pulled up on the board if necessary. After the reset is complete, it takes approximately 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 th completion of bit rate adjustment. The host should confirm that this adjustment end i (H'00) has been received normally, and transmit one H'55 byte to the chip. If receptio not be performed normally, initiate boot mode again by a reset. Depending on the hos transfer bit rate and system clock frequency of this LSI, there will be a discrepancy be the bit rates of the host and the chip. To operate the SCI properly, set the host's transf 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 I H'FEEF is the area to which the programming control program is transferred from the The boot program area cannot be used until the execution state in boot mode switches programming control program.
6. Before branching to the programming control program, the chip terminates transfer op by SCI3 (by clearing the RE and TE bits in SCR to 0), however the adjusted bit rate v remains set in BRR. Therefore, the programming control program can still use it for t of program data or verify data with the host. The TxD pin is high (PCR22 = 1, P22 = contents of the CPU general registers are undefined immediately after branching to th programming control program. These registers must be initialized at the beginning of programming control program, as the stack pointer (SP), in particular, is used implicit subroutine calls, etc.
7. Boot mode can be cleared by a reset. End the reset after driving the reset pin low, wa least 20 states, and then setting the $\overline{\text{NMI}}$ 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.


Bit rate adjustm
Transmits data H'55 when data H'00 is received error-free.
H'00
H'55
• Calculates bit rate and sets BRF
• Transmits data H'00 to host as a end indication.
H'55 reception.
Flash memory erase
Boot program erase error
H'FF
Checks flash memory data, erase memory blocks in case of written existing, and transmits data H'AA (If erase could not be done, trans H'FF to host and aborts operatio
H'AA reception
H'AA
Transfer of number of bytes of programming control program
Transmits number of bytes (N) of programming control program to be transferred as 2-byte data (low-order byte following high-order byte)
Upper bytes, lower bytes
Echoback
Echobacks the 2-byte data received to host.
Transmits 1-byte of programming control program (repeated for N times)
H'XX
Echoback
Echobacks received data to host transfers it to RAM. (repeated for N times)
H'AA reception
H'AA
Transmits data H'AA to host.
Branches to programming contro transferred to on-chip RAM and execution.

7.3.2 Programming/Erasing in User Program Mode

On-board programming/erasing of an individual flash memory block can also be perform program mode by branching to a user program/erase control program. The user must set b conditions and provide on-board means of supplying programming data. The flash memo contain the user program/erase control program or a program that provides the user progr 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, mode. Figure 7.2 shows a sample procedure for programming/erasing in user program mo Prepare a user program/erase control program in accordance with the description in sectio Flash Memory Programming/Erasing.

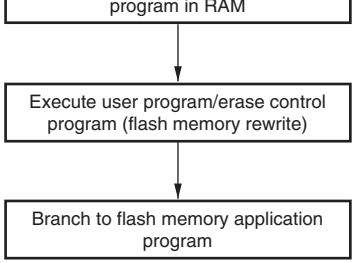
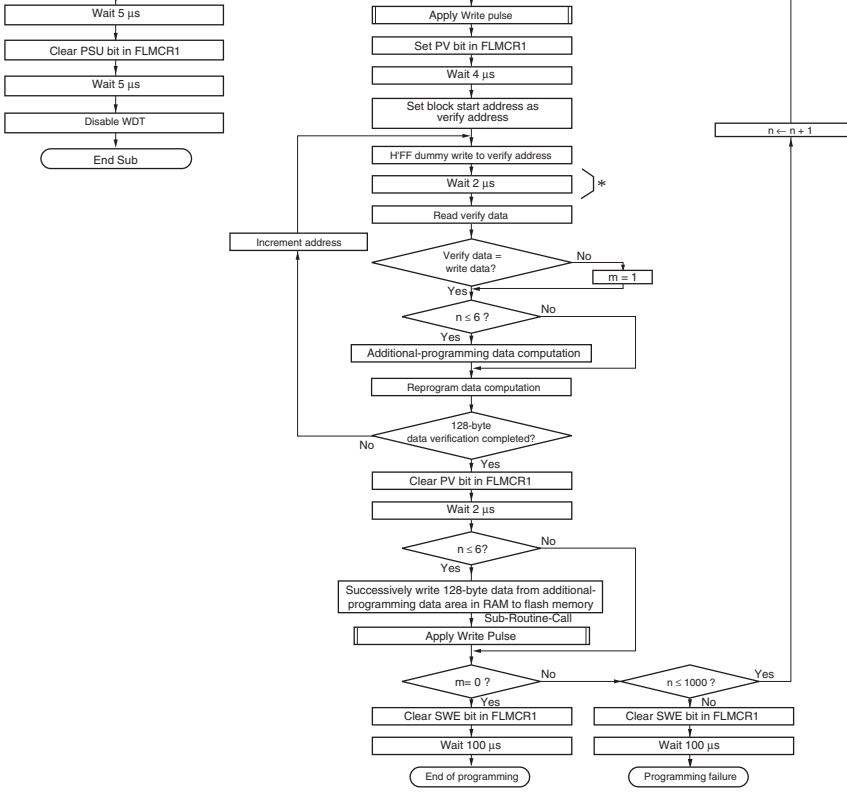


Figure 7.2 Programming/Erasing Flowchart Example in User Program Mo

7.4.1 Program/Program-Verify

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

1. Programming must be done to an empty address. Do not reprogram an address to wh programming has already been performed.
2. Programming should be carried out 128 bytes at a time. A 128-byte data transfer mus performed even if writing fewer than 128 bytes. In this case, H'FF data must be writt extra addresses.
3. Prepare the following data storage areas in RAM: A 128-byte programming data area byte reprogramming data area, and a 128-byte additional-programming data area. Per reprogramming data computation according to table 7.4, and additional programming computation according to table 7.5.
4. Consecutively transfer 128 bytes of data in byte units from the reprogramming data ar additional-programming data area to the flash memory. The program address and 128 data are latched in the flash memory. The lower 8 bits of the start address in the flash 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 allowable programming times.
6. The watchdog timer (WDT) is set to prevent overprogramming due to program runaw 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 lo are B'00. Verify data can be read in words or in longwords from the address to which dummy write was performed.



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

Reprogram Data	Verify Data	Additional-Program Data	Comments
0	0	0	Additional-program l
0	1	1	No additional progra
1	0	1	No additional progra
1	1	1	No additional progra

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.

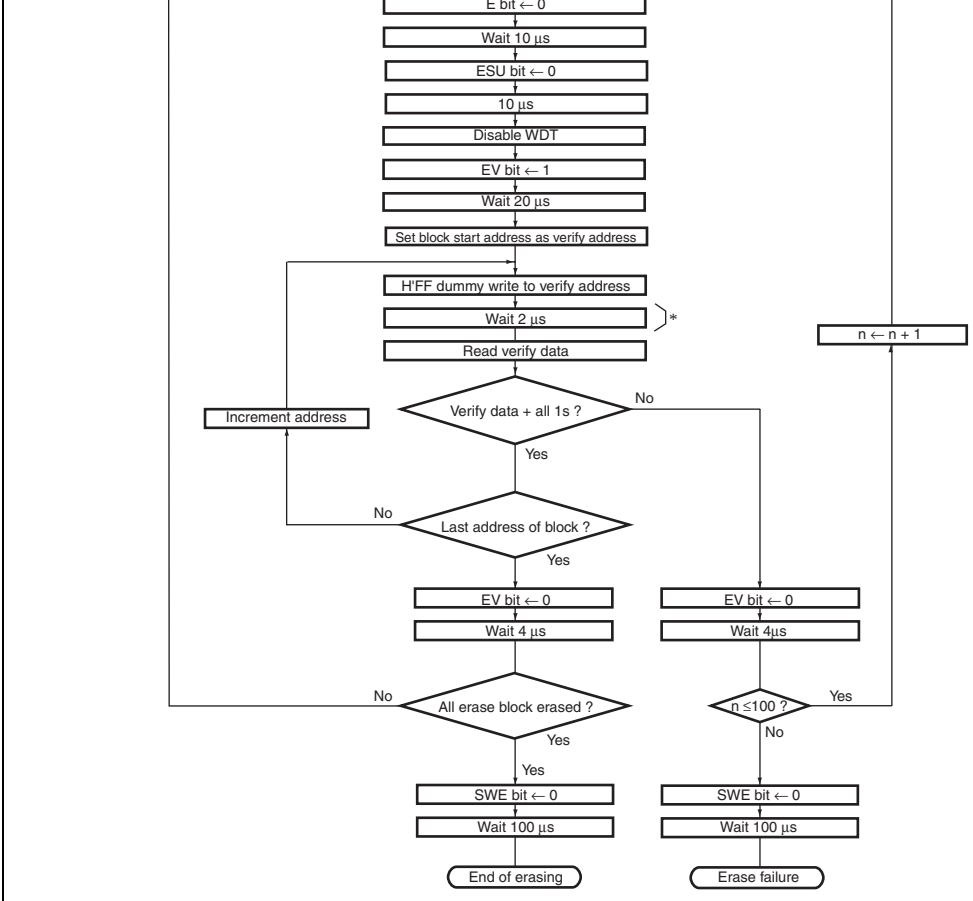
overflow cycle of approximately 19.8 ms is allowed.

5. For a dummy write to a verify address, write 1-byte data H'FF to an address whose l bits are B'00. Verify data can be read in longwords from the address to which a dum was performed.
6. If the read data is not erased successfully, set erase mode again, and repeat the erase/ verify sequence as before. The maximum number of repetitions of the erase/erase-v 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 pr 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 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 ca 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 readin

Figure 7.4 Erase/Erase-Verify Flowchart

entered unless the $\overline{\text{RES}}$ pin is held low until oscillation stabilizes after powering on. In t
a reset during operation, hold the $\overline{\text{RES}}$ pin low for the $\overline{\text{RES}}$ pulse width specified in the
Characteristics section.

7.5.2 Software Protection

Software protection can be implemented against programming/erasing of all flash memo
by clearing the SWE bit in FLMCR1. When software protection is in effect, setting the
in FLMCR1 does not cause a transition to program mode or erase mode. By setting the
block register 1 (EBR1), erase protection can be set for individual blocks. When EBR1
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/era
algorithm, and the program/erase operation is forcibly aborted. Aborting the program/er
operation prevents damage to the flash memory due to overprogramming or overerasing

When the following errors are detected during programming/erasing of flash memory, t
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/era
(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

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 mem
  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 flas
memory. In subactive mode, the flash memory can be set to operate in power-down mode
PDWND bit in FLPWCR. When the flash memory returns to its normal operating state fr
power-down mode or standby mode, a period to stabilize operation of the power supply c
that were stopped is needed. When the flash memory returns to its normal operating state
STS2 to STS0 in SYSCR1 must be set to provide a wait time of at least 20 µs, even when
external clock is being used.

H8/36085	3 kbytes	H'E800 to H'EFFF, H'FB80 to H'F
H8/36084	3 kbytes	H'E800 to H'EFFF, H'FB80 to H'F
H8/36083	3 kbytes	H'E800 to H'EFFF, H'FB80 to H'F
H8/36082	3 kbytes	H'E800 to H'EFFF, H'FB80 to H'F

Note: * When the E7 or E8 is used, area H'F780 to H'FB7F must not be accessed.

For functions in each port, see Appendix B.1, I/O Port Block Diagrams. For the executio
manipulation instructions to the port control register and port data register, see section 2
Manipulation Instruction.

9.1 Port 1

Port 1 is a general I/O port also functioning as IRQ interrupt input pins, an RTC output p
bit PWM output pin, a timer B1 input pin, and a timer V input pin. Figure 9.1 shows its
configuration.

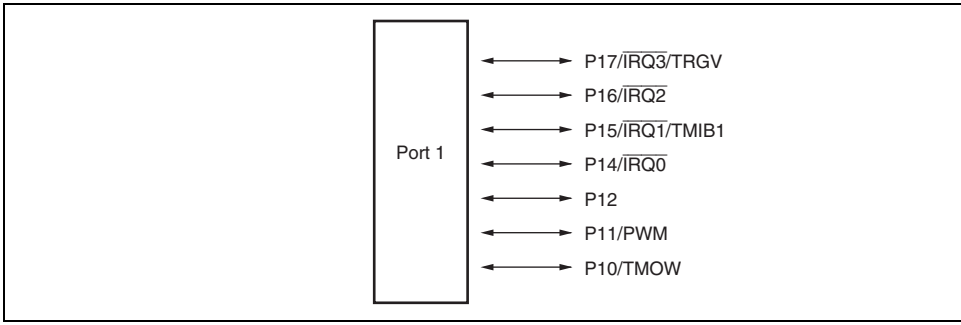


Figure 9.1 Port 1 Pin Configuration

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)

				0: General I/O port 1: $\overline{\text{IRQ2}}$ input pin
5	IRQ1	0	R/W	This bit selects the function of pin P15/$\overline{\text{IRQ1}}$/TMI 0: General I/O port 1: $\overline{\text{IRQ1}}$/TMIB1 input pin
4	IRQ0	0	R/W	This bit selects the function of pin P14/$\overline{\text{IRQ0}}$. 0: General I/O port 1: $\overline{\text{IRQ0}}$ input pin
3	TXD2	0	R/W	This bit selects the function of pin P72/TXD_2. 0: General I/O port 1: TXD_2 output pin
2	PWM	0	R/W	This bit selects the function of pin P11/PWM. 0: General I/O port 1: PWM output pin
1	TXD	0	R/W	This bit selects the function of pin P22/TXD. 0: General I/O port 1: TXD output pin
0	TMOW	0	R/W	This bit selects the function of pin P10/TMOW. 0: General I/O port 1: TMOW output pin

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, th
5	P15	0	R/W	stored in PDR1 are read. If PDR1 is read while
4	P14	0	R/W	are cleared to 0, the pin states are read regardl value stored in PDR1.
3	—	1	—	Bit 3 is a reserved bit. This bit is always read as
2	P12	0	R/W	
1	P11	0	R/W	
0	P10	0	R/W	

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 be

- P17/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.

- P13/IRQ1/TMIB1 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/TMIB1 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

- 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.

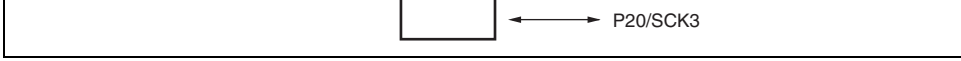


Figure 9.2 Port 2 Pin Configuration

Port 2 has the following registers.

- Port control register 2 (PCR2)
- Port data register 2 (PDR2)
- Port mode register 3 (PMR3)

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 to 5	—	—	—	Reserved
4	PCR24	0	W	When each of the port 2 pins P24 to P20 functi
3	PCR23	0	W	general I/O port, setting a PCR2 bit to 1 makes
2	PCR22	0	W	corresponding pin an output port, while clearing
1	PCR21	0	W	0 makes the pin an input port.
0	PCR20	0	W	

2	P22	0	R/W	stored in PDR2 is read. If PDR2 is read while PC are cleared to 0, the pin states are read regardle value stored in PDR2.
1	P21	0	R/W	
0	P20	0	R/W	

9.2.3 Port Mode Register 3 (PMR3)

PMR3 selects the CMOS output or NMOS open-drain output for port 2.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	—	Reserved These bits are always read as 0.
4	POF24	0	R/W	When the bit is set to 1, the corresponding pin is by PMOS and it functions as the NMOS open-dr output. When cleared to 0, the pin functions as t output.
3	POF23	0	R/W	
2 to 0	—	All 1	—	Reserved These bits are always read as 1.

- P23 pin

Register	PCR2	
Bit Name	**PCR23**	**Pin Function**
Setting Value	0	P23 input pin
	1	P23 output pin

- 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.

X: Don't care.

9.3 Port 3

Port 3 is a general I/O port. Each pin of the port 3 is shown in figure 9.3.

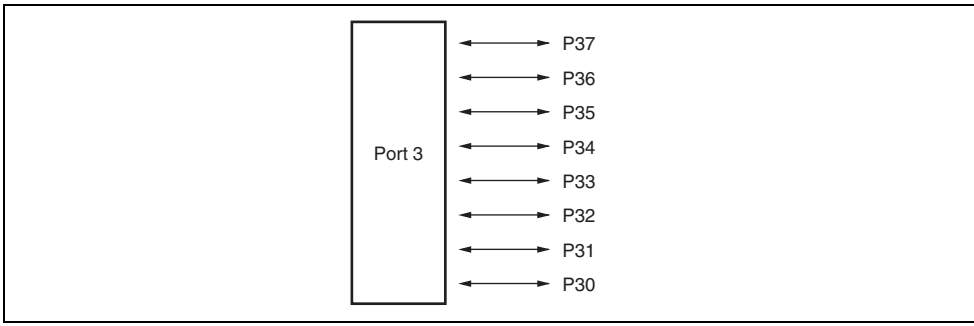


Figure 9.3 Port 3 Pin Configuration

Port 3 has the following registers.

- Port control register 3 (PCR3)
- Port data register 3 (PDR3)

3	PCR33	0	W
2	PCR32	0	W
1	PCR31	0	W
0	PCR30	0	W

9.3.2 Port Data Register 3 (PDR3)

PDR3 is a general I/O port data register of port 3.

Bit	Bit Name	Initial Value	R/W	Description
7	P37	0	R/W	PDR3 stores output data for port 3 pins.
6	P36	0	R/W	If PDR3 is read while PCR3 bits are set to 1, th
5	P35	0	R/W	stored in PDR3 is read. If PDR3 is read while P
4	P34	0	R/W	are cleared to 0, the pin states are read regardl
3	P33	0	R/W	value stored in PDR3.
2	P32	0	R/W	
1	P31	0	R/W	
0	P30	0	R/W	

- P36 pin

Register	PCR3	
Bit Name	**PCR36**	**Pin Function**
Setting Value	0	P36 input pin
	1	P36 output pin

- P35 pin

Register	PCR3	
Bit Name	**PCR35**	**Pin Function**
Setting Value	0	P35 input pin
	1	P35 output pin

- P34 pin

Register	PCR3	
Bit Name	**PCR34**	**Pin Function**
Setting Value	0	P34 input pin
	1	P34 output pin

- P33 pin

Register	PCR3	
Bit Name	**PCR33**	**Pin Function**
Setting Value	0	P33 input pin
	1	P33 output pin

Setting Value	0	P31 input pin
	1	P31 output pin

- P30 pin

Register	**PCR3**	
Bit Name	**PCR30**	**Pin Function**
Setting Value	0	P30 input pin
	1	P30 output pin

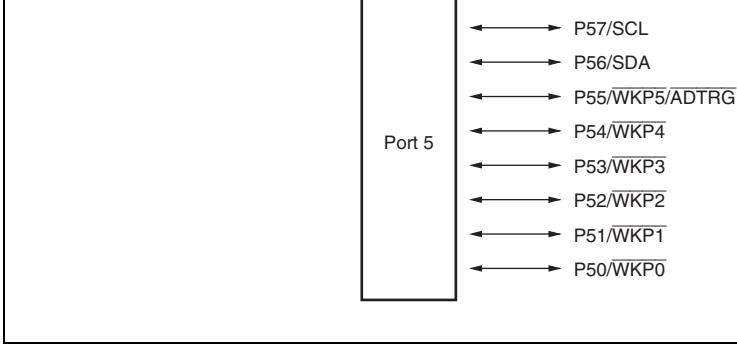


Figure 9.4 Port 5 Pin Configuration

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)

				0: General I/O port 1: $\overline{\text{WKP5}}$/$\overline{\text{ADTRG}}$ input pin
4	WKP4	0	R/W	This bit selects the function of pin P54/$\overline{\text{WKP4}}$. 0: General I/O port 1: $\overline{\text{WKP4}}$ input pin
3	WKP3	0	R/W	This bit selects the function of pin P53/$\overline{\text{WKP3}}$. 0: General I/O port 1: $\overline{\text{WKP3}}$ input pin
2	WKP2	0	R/W	This bit selects the function of pin P52/$\overline{\text{WKP2}}$. 0: General I/O port 1: $\overline{\text{WKP2}}$ input pin
1	WKP1	0	R/W	This bit selects the function of pin P51/$\overline{\text{WKP1}}$. 0: General I/O port 1: $\overline{\text{WKP1}}$ input pin
0	WKP0	0	R/W	This bit selects the function of pin P50/$\overline{\text{WKP0}}$. 0: General I/O port 1: $\overline{\text{WKP0}}$ input pin

3	PCR53	0	W
2	PCR52	0	W
1	PCR51	0	W
0	PCR50	0	W

9.4.3 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 stored in PDR5 are read. If PDR5 is read while P are cleared to 0, the pin states are read regardle value stored in PDR5.
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	

3	PUCR53	0	R/W	these bits are cleared to 0.
2	PUCR52	0	R/W	
1	PUCR51	0	R/W	
0	PUCR50	0	R/W	

9.4.5 Pin Functions

The correspondence between the register specification and the port functions is shown b

- P57/SCL pin

Register	ICCR1	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.

- 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.

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.

9.5 Port 6

Port 6 is a general I/O port also functioning as a timer Z I/O pin. Each pin of the port 6 is
figure 9.5. The register setting of the timer Z has priority for functions of the pins for botl

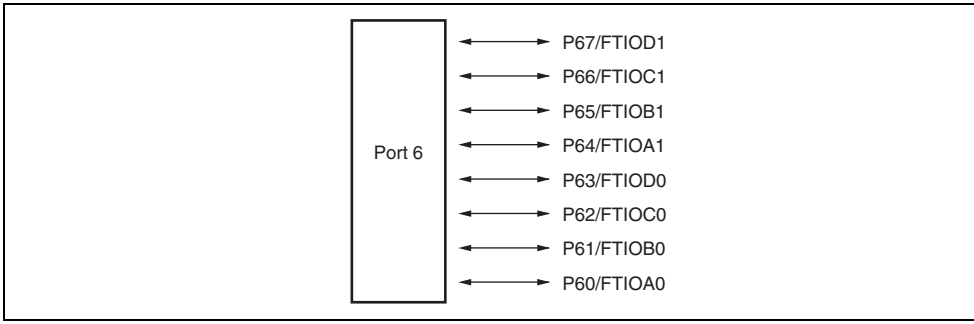


Figure 9.5 Port 6 Pin Configuration

Port 6 has the following registers.

- Port control register 6 (PCR6)
- Port data register 6 (PDR6)

3	PCR63	0	W	
2	PCR62	0	W	
1	PCR61	0	W	
0	PCR60	0	W	

9.5.2 Port Data Register 6 (PDR6)

PDR6 is a general I/O port data register of port 6.

Bit	Bit Name	Initial Value	R/W	Description
7	P67	0	R/W	Stores output data for port 6 pins.
6	P66	0	R/W	If PDR6 is read while PCR6 bits are set to 1, th
5	P65	0	R/W	stored in PDR6 are read. If PDR6 is read while
4	P64	0	R/W	are cleared to 0, the pin states are read regardl
3	P63	0	R/W	value stored in PDR6.
2	P62	0	R/W	
1	P61	0	R/W	
0	P60	0	R/W	

				1	P67 output pin
0	00	0	001 or 01X	X	FTIOD1 output pin
		1	XXX		
	Other than 00	X	XXX		

[Legend]
X: Don't care.

- P66/FTIOC1 pin

Register	TOER	TFCR	TPMR	TIORC1	PCR6	
Bit Name	**EC1**	**CMD1 and CMD0**	**PWMC1**	**IOC2 to IOC0**	**PCR66**	**Pin Function**
Setting Value	1	00	0	000 or 1XX	0	P66 input/FTIOC1 i
					1	P66 output pin
	0	00	0	001 or 01X	X	FTIOC1 output pin
			1	XXX		
		Other than 00	X	XXX		

[Legend]
X: Don't care.

			than 00		

[Legend]
X: Don't care.

- P64/FTIOA1 pin

Register	TOER	TFCR	TIORA1	PCR6	
Bit Name	EB1	CMD1 to CMD0	IOA2 to IOA0	PCR64	Pin Function
Setting Value	1	XX	000 or 1XX	0	P64 input/FTIOA1
				1	P64 output pin
	0	00	001 or 01X	X	FTIOA1 output pin

[Legend]
X: Don't care.

		than 00				

[Legend]
X: Don't care.

- P62/FTIOC0 pin

Register	TOER	TFCR	TPMR	TIORC0	PCR6	
Bit Name	**EC0**	**CMD1 to CMD0**	**PWMC0**	**IOC2 to IOC0**	**PCR62**	**Pin Function**
Setting Value	1	00	0	000 or 1XX	0	P62 input/FTIOC0
					1	P62 output pin
	0	00	0	001 or 01X	X	FTIOC0 output pin
			1	XXX		
		Other than 00	X	XXX		

[Legend]
X: Don't care.

				than 00		

[Legend]
X: Don't care.

- P60/FTIOA0 pin

Register	TOER	TFCR	TFCR	TIORA0	PCR6	
Bit Name	EA0	CMD1 to CMD0	STCLK	IOA2 to IOA0	PCR60	Pin Function
Setting Value	1	XX	X	000 or 1XX	0	P60 input/FTIOA0 in
					1	P60 output pin
	0	00	0	001 or 01X	X	FTIOA0 output pin

[Legend]
X: Don't care.

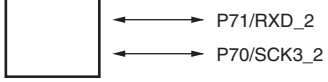


Figure 9.6 Port 7 Pin Configuration

Port 7 has the following registers.

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

9.6.1 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	—	—	—	When each of the port 7 pins P76 to P74 and P7 functions as a general I/O port, setting a PCR7 b makes the corresponding pin an output port, whi clearing the bit to 0 makes the pin an input port.
6	PCR76	0	W	
5	PCR75	0	W	
4	PCR74	0	W	Bits 7 and 3 are reserved bits.
3	—	—	—	
2	PCR72	0	W	
1	PCR71	0	W	
0	PCR70	0	W	

3	—	1	—	Bits 7 and 3 are reserved bits. These bits are al
2	P72	0	R/W	as 1.
1	P71	0	R/W	
0	P70	0	R/W	

9.6.3 Pin Functions

The correspondence between the register specification and the port functions is shown b

- 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

Setting Value	0	0	P72 input pin
		1	P72 output pin
	1	X	TXD_2 output pin

[Legend]
X: Don't care.

- P71/RXD_2 pin

Register	SCR3_2	PCR7	
Bit Name	**RE**	**PCR71**	**Pin Function**
Setting Value	0	0	P71 input pin
		1	P71 output pin
	1	X	RXD_2 input pin

[Legend]
X: Don't care.

- P70/SCK3_2 pin

Register	SCR3_2		SMR2	PCR7	
Bit Name	**CKE1**	**CKE0**	**COM**	**PCR70**	**Pin Function**
Setting Value	0	0	0	0	P70 input pin
				1	P70 output pin
	0	0	1	X	SCK3_2 output pin
	0	1	X	X	SCK3_2 output pin
	1	X	X	X	SCK3_2 input pin

[Legend]
X: Don't care.

Port 8 has the following registers.

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

9.7.1 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 P85 functi
6	PCR86	0	W	general I/O port, setting a PCR8 bit to 1 makes corresponding pin an output port, while clearing
5	PCR85	0	W	0 makes the pin an input port.
4 to 0	—	—	—	Reserved

4 to 0	—	All 1	—	Reserved These bits are always read as 1.

9.7.3 Pin Functions

The correspondence between the register specification and the port functions is shown be

- 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

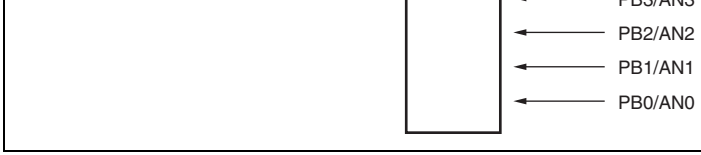


Figure 9.8 Port B Pin Configuration

Port B has the following register.

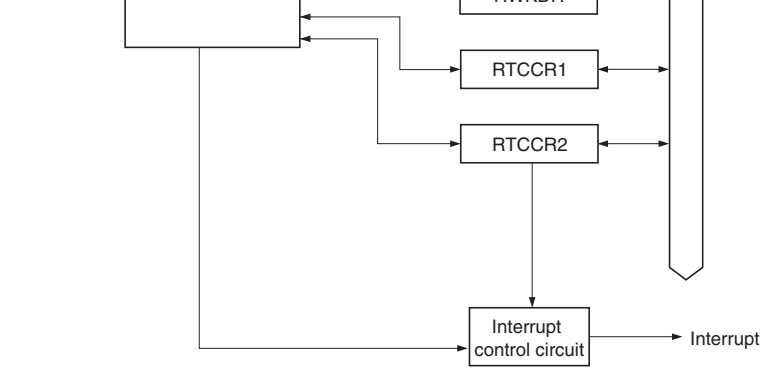
- Port data register B (PDRB)

9.8.1 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 t register.
6	PB6	—	R	
5	PB5	—	R	However, if a port B pin is designated as an ana channel by ADCSR in A/D converter, 0 is read.
4	PB4	—	R	
3	PB3	—	R	
2	PB2	—	R	
1	PB1	—	R	
0	PB0	—	R	

- Readable/writable counter of seconds, minutes, hours, and day-of-week with BCD c
- Periodic (seconds, minutes, hours, days, and weeks) interrupts
- 8-bit free running counter
- Selection of clock source



[Legend]

RTCCSR: Clock source select register
RSECDR: Second date register/free running counter data register
RMINDR: Minute date register
RHRDR: Hour date register
RWKDR: Day-of-week date register
RTCCR1: RTC control register 1
RTCCR2: RTC control register 2
PSS: Prescaler S

Figure 10.1 Block Diagram of RTC

The RTC has the following registers.

- Second data register/free running counter data register (RSECDR)
- Minute data register (RMINDR)
- Hour data register (RHRDR)
- Day-of-week data register (RWKDR)
- RTC control register 1 (RTCCR1)
- RTC control register 2 (RTCCR2)
- Clock source select register (RTCCSR)

10.3.1 Second Data Register/Free Running Counter Data Register (RSECDR)

RSECDR counts the BCD-coded second value. The setting range is decimal 00 to 59. It read register used as a counter, when it operates as a free running counter. For more info on reading seconds, minutes, hours, and day-of-week, see section 10.4.3, Data Reading

Bit	Bit Name	Initial Value	R/W	Description
7	BSY	—	R	RTC busy This bit is set to 1 when the RTC is updating (o the values of second, minute, hour, and day-of- registers. When this bit is 0, the values of seco hour, and day-of-week data registers must be a
6	SC12	—	R/W	Counting ten's position of seconds
5	SC11	—	R/W	Counts on 0 to 5 for 60-second counting.
4	SC10	—	R/W	

RMINDR counts the BCD-coded minute value on the carry generated once per minute by RSECDR counting. The setting range is decimal 00 to 59.

Bit	Bit Name	Initial Value	R/W	Description
7	BSY	—	R	RTC busy This bit is set to 1 when the RTC is updating (op the values of second, minute, hour, and day-of-w registers. When this bit is 0, the values of secon hour, and day-of-week data registers must be ac
6	MN12	—	R/W	Counting ten's position of minutes
5	MN11	—	R/W	Counts on 0 to 5 for 60-minute counting.
4	MN10	—	R/W	
3	MN03	—	R/W	Counting one's position of minutes
2	MN02	—	R/W	Counts on 0 to 9 once per minute. When a carry
1	MN01	—	R/W	generated, 1 is added to the ten's position.
0	MN00	—	R/W	

				registers. When this bit is 0, the values of secon hour, and day-of-week data registers must be a
6	—	0	—	Reserved This bit is always read as 0.
5	HR11	—	R/W	Counting ten's position of hours
4	HR10	—	R/W	Counts on 0 to 2 for ten's position of hours.
3	HR03	—	R/W	Counting one's position of hours
2	HR02	—	R/W	Counts on 0 to 9 once per hour. When a carry
1	HR01	—	R/W	generated, 1 is added to the ten's position.
0	HR00	—	R/W	

				hour, and day-of-week data registers must be ac
6 to 3	—	All 0	—	Reserved These bits are always read as 0.
2	WK2	—	R/W	Day-of-week counting
1	WK1	—	R/W	Day-of-week is indicated with a binary code
0	WK0	—	R/W	000: Sunday 001: Monday 010: Tuesday 011: Wednesday 100: Thursday 101: Friday 110: Saturday 111: Reserved (setting prohibited)

6	12/24	—	R/W	Operating mode 0: RTC operates in 12-hour mode. RHRDR co to 11. 1: RTC operates in 24-hour mode. RHRDR co to 23.
5	PM	—	R/W	A.m./p.m. 0: Indicates a.m. when RTC is in the 12-hour m 1: Indicates p.m. when RTC is in the 12-hour m
4	RST	0	R/W	Reset 0: Normal operation 1: Resets registers and control circuits except R and this bit. Clear this bit to 0 after having be 1.
3 to 0	—	All 0	—	Reserved These bits are always read as 0.

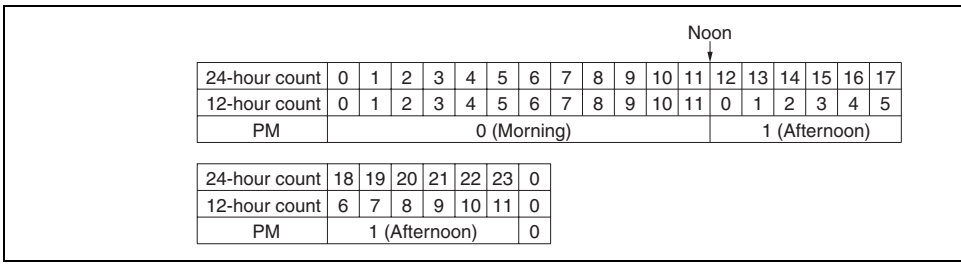


Figure 10.2 Definition of Time Expression

				These bits are always read as 0.
5	FOIE	—	R/W	Free Running Counter Overflow Interrupt Enable 0: Disables an overflow interrupt 1: Enables an overflow interrupt
4	WKIE	—	R/W	Week Periodic Interrupt Enable 0: Disables a week periodic interrupt 1: Enables a week periodic interrupt
3	DYIE	—	R/W	Day Periodic Interrupt Enable 0: Disables a day periodic interrupt 1: Enables a day periodic interrupt
2	HRIE	—	R/W	Hour Periodic Interrupt Enable 0: Disables an hour periodic interrupt 1: Enables an hour periodic interrupt
1	MNIE	—	R/W	Minute Periodic Interrupt Enable 0: Disables a minute periodic interrupt 1: Enables a minute periodic interrupt
0	SEIE	—	R/W	Second Periodic Interrupt Enable 0: Disables a second periodic interrupt 1: Enables a second periodic interrupt

7	—	0	—	Reserved This bit is always read as 0.
6	RCS6	0	R/W	Clock output selection
5	RCS5	0	R/W	Selects a clock output from the TMOW pin whe TMOW in PMR1 to 1. 00: φ/4 01: φ/8 10: φ/16 11: φ/32
4	—	0	—	Reserved This bit is always read as 0.
3	RCS3	1	R/W	Clock source selection
2	RCS2	0	R/W	0000: φ/8····· Free running counter opera
1	RCS1	0	R/W	0001: φ/32····· Free running counter opera
0	RCS0	0	R/W	0010: φ/128····· Free running counter opera 0011: φ/256····· Free running counter opera 0100: φ/512····· Free running counter opera 0101: φ/2048····· Free running counter opera 0110: φ/4096····· Free running counter opera 0111: φ/8192····· Free running counter opera 1XXX: 32.768 kHz·····RTC operation

[Legend]

X: Don't care.

Figure 10.3 shows the procedure for the initial setting of the RTC. To set the RTC again, follow this procedure.

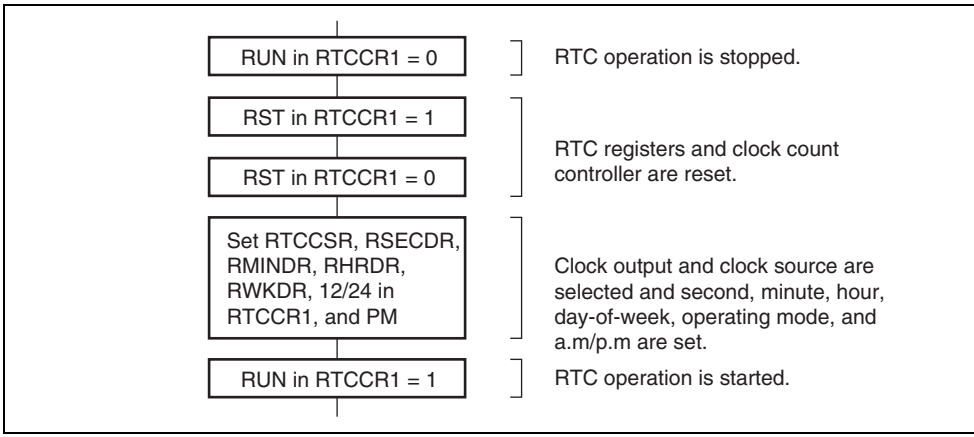


Figure 10.3 Initial Setting Procedure

bit is set to 1, the registers are updated, and the BSY bit is cleared to 0.

2. Making use of interrupts, read from the second, minute, hour, and day-of week regis the IRRTA flag in IRR1 is set to 1 and the BSY bit is confirmed to be 0.
3. Read from the second, minute, hour, and day-of week registers twice in a row, and if no change in the read data, the read data is used.

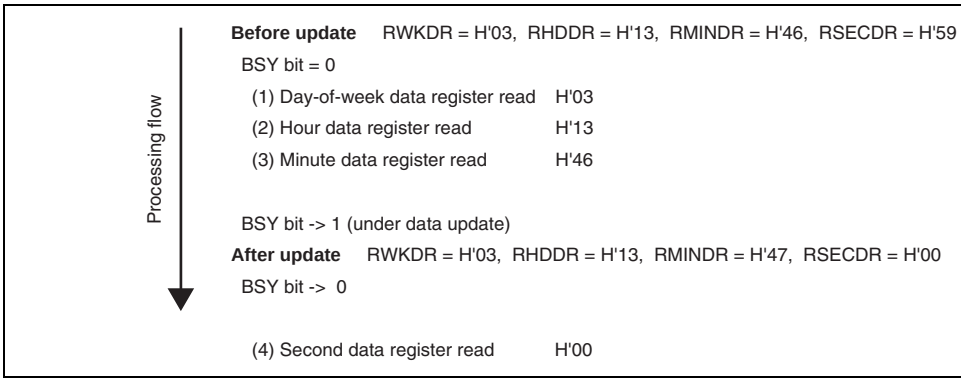


Figure 10.4 Example: Reading of Inaccurate Time Data

Table 10.2 Interrupt Source

Interrupt Name	Interrupt Source	Interrupt En
Overflow interrupt	Occurs when the free running counter is overflown.	FOIE
Week periodic interrupt	Occurs every week when the day-of-week date register value becomes 0.	WKIE
Day periodic interrupt	Occurs every day when the day-of-week date register is counted.	DYIE
Hour periodic interrupt	Occurs every hour when the hour date register is counted.	HRIE
Minute periodic interrupt	Occurs every minute when the minute date register is counted.	MNIE
Second periodic interrupt	Occurs every second when the second date register is counted.	SCIE

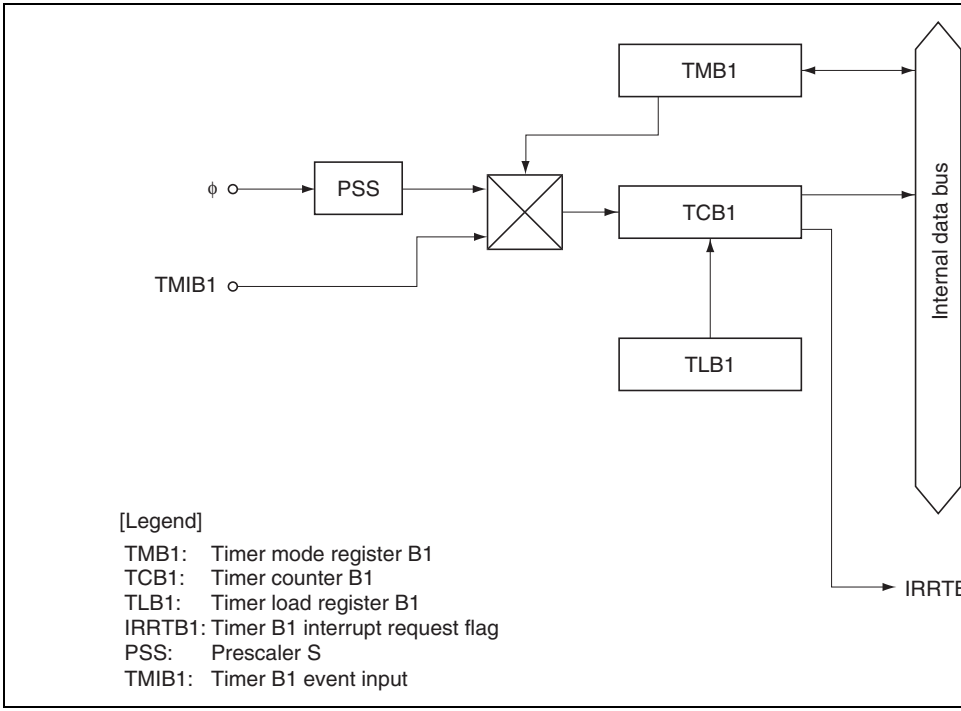


Figure 11.1 Block Diagram of Timer B1

The timer B1 has the following registers.

- Timer mode register B1 (TMB1)
- Timer counter B1 (TCB1)
- Timer load register B1 (TLB1)

11.3.1 Timer Mode Register B1 (TMB1)

TMB1 selects the auto-reload function and input clock.

Bit	Bit Name	Initial Value	R/W	Description
7	TMB17	0	R/W	Auto-reload function select 0: Interval timer function selected 1: Auto-reload function selected
6 to 3	—	All 1	—	Reserved These bits are always read as 1.

111: External event (TMIB1): rising or falling ed

Note: * The edge of the external event signal
by bit IEG1 in the interrupt edge selec
(IEGR1). See section 3.2.1, Interrupt E
Select Register 1 (IEGR1), for details.
setting TMB12 to TMB10 to 1, IRQ1 in
mode register 1 (PMR1) should be set

11.3.2 Timer Counter B1 (TCB1)

TCB1 is an 8-bit read-only up-counter, which is incremented by internal clock input. Th
source for input to this counter is selected by bits TMB12 to TMB10 in TMB1. TCB1 v
be read by the CPU at any time. When TCB1 overflows from H'FF to H'00 or to the valu
TLB1, the IRRTB1 flag in IRR2 is set to 1. TCB1 is allocated to the same address as TL
is initialized to H'00.

11.3.3 Timer Load Register B1 (TLB1)

TLB1 is an 8-bit write-only register for setting the reload value of TCB1. When a reload
set in TLB1, the same value is loaded into TCB1 as well, and TCB1 starts counting up f
value. When TCB1 overflows during operation in auto-reload mode, the TLB1 value is
into TCB1. Accordingly, overflow periods can be set within the range of 1 to 256 input
TLB1 is allocated to the same address as TCB1. TLB1 is initialized to H'00.

overflow, setting flag IRRTB1 in IRR2 to 1. If IENTB1 in IENR2 is 1, an interrupt is req
the CPU.

At overflow, TCB1 returns to H'00 and starts counting up again. During interval timer op
(TMB17 = 0), when a value is set in TLB1, the same value is set in TCB1.

11.4.2 Auto-Reload Timer Operation

Setting bit TMB17 in TMB1 to 1 causes timer B1 to function as an 8-bit auto-reload time
a reload value is set in TLB1, the same value is loaded into TCB1, becoming the value fr
TCB1 starts its count. After the count value in TCB1 reaches H'FF, the next clock signal
causes timer B1 to overflow. The TLB1 value is then loaded into TCB1, and the count c
from that value. The overflow period can be set within a range from 1 to 256 input clock
depending on the TLB1 value.

The clock sources and interrupts in auto-reload mode are the same as in interval mode. In
reload mode (TMB17 = 1), when a new value is set in TLB1, the TLB1 value is also load
TCB1.

11.4.3 Event Counter Operation

Timer B1 can operate as an event counter in which TMIB1 is set to an event input pin. Ex
event counting is selected by setting bits TMB12 to TMB10 in TMB1 to 1. TCB1 counts
rising or falling edge of an external event signal input at pin TMB1.

When timer B1 is used to count external event input, bit IRQ1 in PMR1 should be set to 1
IEN1 in IENR1 should be cleared to 0 to disable IRQ1 interrupt requests.

- 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
- Counter can be cleared by compare match A or B, or by an external reset signal. If th
  stop function is selected, the counter can be halted when cleared.
- Timer output is controlled by two independent compare match signals, enabling puls
  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 e
  both edges of the TRGV input can be selected.

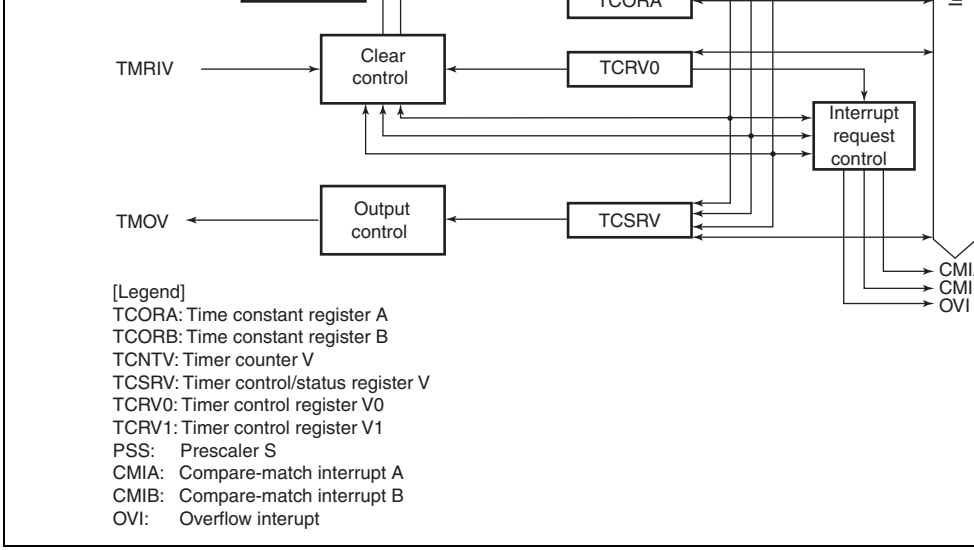


Figure 12.1 Block Diagram of Timer V

12.3 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)

12.3.1 Timer Counter V (TCNTV)

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

and the settings of bits OS3 to OS0 in TCSRV.

TCORA and TCORB are initialized to H'FF.

12.3.3 Timer Control Register V0 (TCRV0)

TCRV0 selects the input clock signals of TCNTV, specifies the clearing conditions of TC 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 th 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 th bit in TCSRV is enabled.
5	OVIE	0	R/W	Timer Overflow Interrupt Enable When this bit is set to 1, interrupt request from th bit in TCSRV is enabled.

2	CKS2	0	R/W	Clock Select 2 to 0
1	CKS1	0	R/W	These bits select clock signals to input to TCNT counting condition in combination with ICKS0 in
0	CKS0	0	R/W	Refer to table 12.2.

Table 12.2 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 e
			1	Internal clock: counts on φ/8, falling e
	1	0	0	Internal clock: counts on φ/16, falling
			1	Internal clock: counts on φ/32, falling
		1	0	Internal clock: counts on φ/64, falling
			1	Internal clock: counts on φ/128, fallin
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 f edge

				After reading CMFB = 1, cleared by writing 0 to C
6	CMFA	0	R/W	Compare Match Flag A Setting condition: When the TCNTV value matches the TCORA va Clearing condition: After reading CMFA = 1, cleared by writing 0 to C
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 OV
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 the compare match of TCORB 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 outp
for compare match A. The two output levels can be controlled independently. After a re
timer output is 0 until the first compare match.

12.3.5 Timer Control Register V1 (TCRV1)

TCRV1 selects the edge at the TRGV pin, enables TRGV input, and selects the clock in
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	TCNT starts counting up by the input of the edg selected by TVEG1 and TVEG0. 0: Disables starting counting-up TCNTV by the the TRGV pin and halting counting-up TCNT TCNTV is cleared by a compare match. 1: Enables starting counting-up TCNTV by the i the TRGV pin and halting counting-up TCNT TCNTV is cleared by a compare match.

12.4 Operation

12.4.1 Timer V Operation

1. According to table 12.2, six internal/external clock signals output by prescaler S can b selected as the timer V operating clock signals. When the operating clock signal is sel TCNTV starts counting-up. Figure 12.2 shows the count timing with an internal clock selected, and figure 12.3 shows the count timing with both edges of an external clock selected.
2. When TCNTV overflows (changes from H'FF to H'00), the overflow flag (OVF) in T will be set. The timing at this time is shown in figure 12.4. An interrupt request is sen CPU when OVIE in TCRV0 is 1.
3. TCNTV is constantly compared with TCORA and TCORB. Compare match flag A o (CMFA or CMFB) is set to 1 when TCNTV matches TCORA or TCORB, respectivel compare-match signal is generated in the last state in which the values match. Figure shows the timing. An interrupt request is generated for the CPU when CMIEA or CM TCRV0 is 1.
4. When a compare match A or B is generated, the TMOV responds with the output valu selected by bits OS3 to OS0 in TCSRV. Figure 12.6 shows the timing when the outpu toggled by compare match A.
5. When CCLR1 or CCLR0 in TCRV0 is 01 or 10, TCNTV can be cleared by the corres compare match. Figure 12.7 shows the timing.
6. When CCLR1 or CCLR0 in TCRV0 is 11, TCNTV can be cleared by the rising edge input of TMRIV pin. A TMRIV input pulse-width of at least 1.5 system clocks is nec Figure 12.8 shows the timing.
7. When a counter-clearing source is generated with TRGE in TCRV1 set to 1, the coun halted as soon as TCNTV is cleared. TCNTV resumes counting-up when the edge sel TVEG1 or TVEG0 in TCRV1 is input from the TGRV pin.

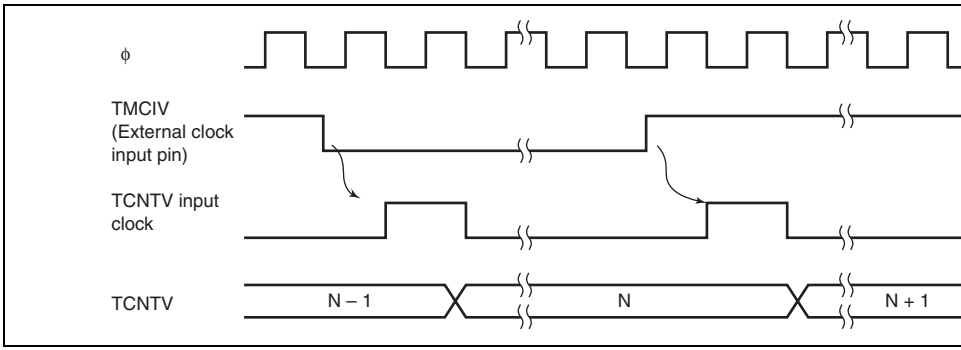


Figure 12.3 Increment Timing with External Clock

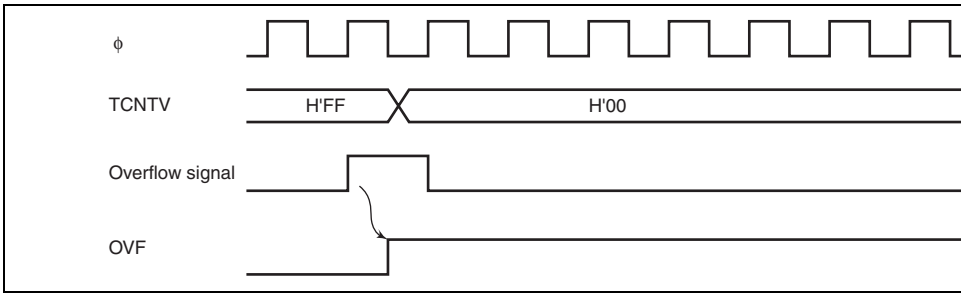


Figure 12.4 OVF Set Timing

Figure 12.5 CMFA and CMFB Set Timing

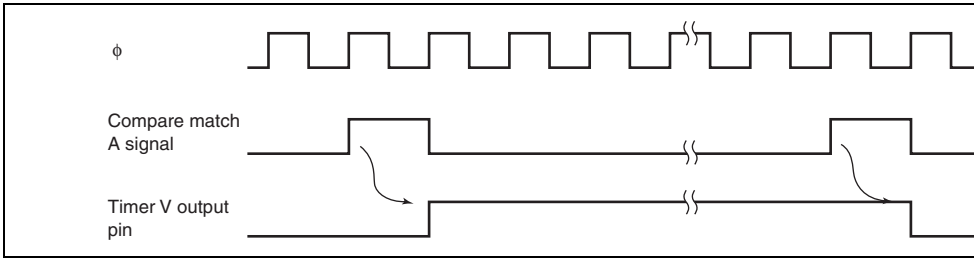


Figure 12.6 TMOV Output Timing

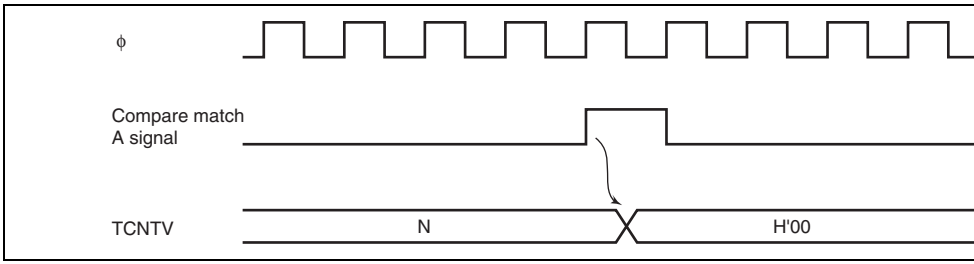


Figure 12.7 Clear Timing by Compare Match

12.5 Timer V Application Examples

12.5.1 Pulse Output with Arbitrary Duty Cycle

Figure 12.9 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 m
   TCORA.
2. Set bits OS3 to OS0 in TCSRV so that the output will go to 1 at compare match with
   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 clo
4. With these settings, a waveform is output without further software intervention, with
   determined by TCORA and a pulse width determined by TCORB.

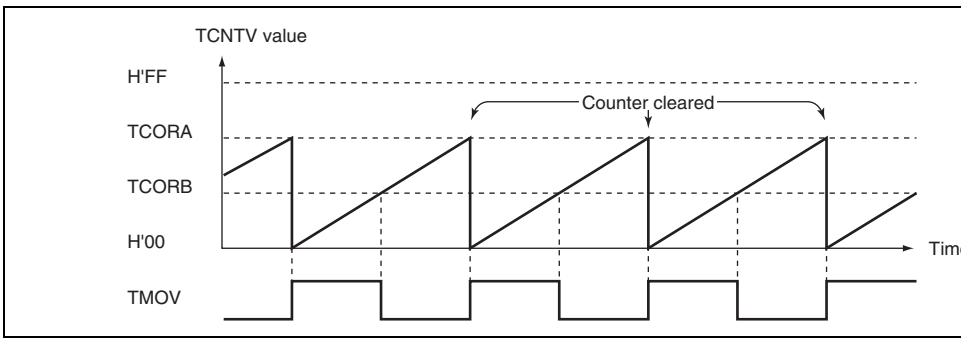


Figure 12.9 Pulse Output Example

input.

4. Set bits CKS2 to CKS0 in TCRV0 and bit ICKS0 in TCRV1 to select the desired cloc
5. After these settings, a pulse waveform will be output without further software interve with a delay determined by TCORA from the TRGV input, and a pulse width determi (TCORB – TCORA).

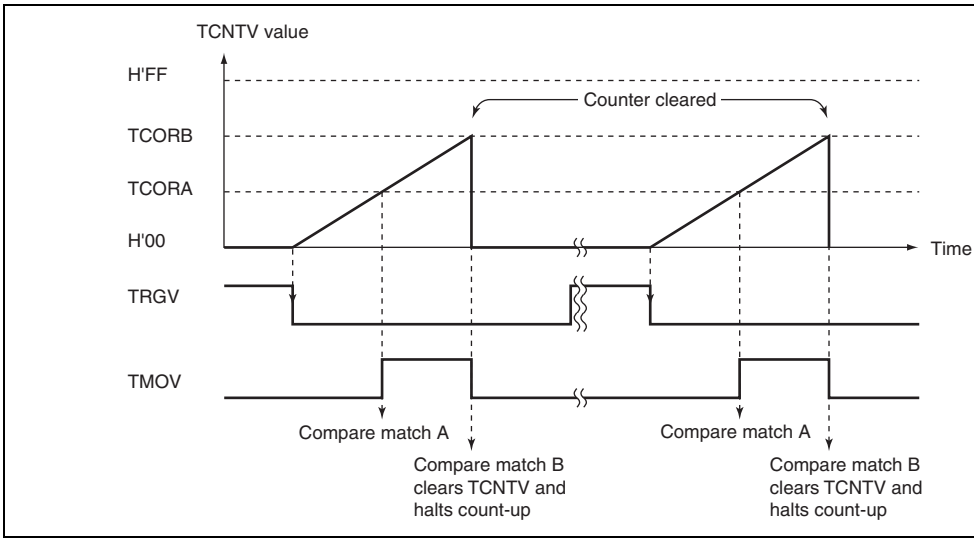


Figure 12.10 Example of Pulse Output Synchronized to TRGV Input

12.12 shows the timing.

3. If compare matches A and B occur simultaneously, any conflict between the output for compare match A and compare match B is resolved by the following priority: to output > output 1 > output 0.
4. Depending on the timing, TCNTV may be incremented by a switch between differe clock sources. When TCNTV is internally clocked, an increment pulse is generated falling edge of an internal clock signal, that is divided system clock (ϕ). Therefore, in figure 12.3 the switch is from a high clock signal to a low clock signal, the switcl seen as a falling edge, causing TCNTV to increment. TCNTV can also be incremen switch between internal and external clocks.

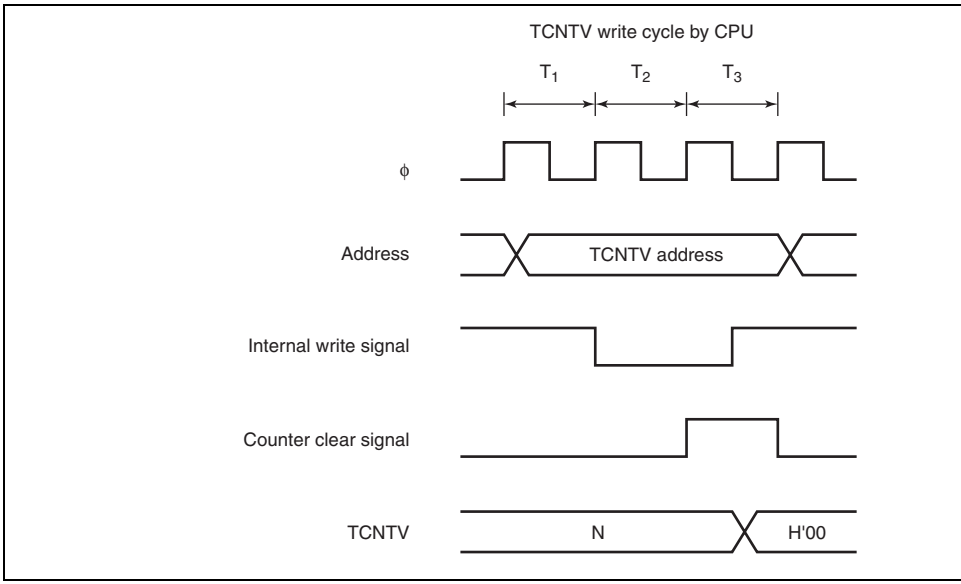


Figure 12.11 Contention between TCNTV Write and Clear

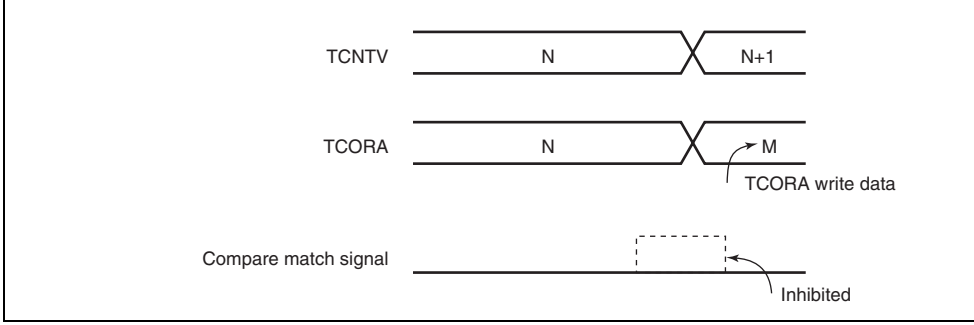


Figure 12.12 Contention between TCORA Write and Compare Match

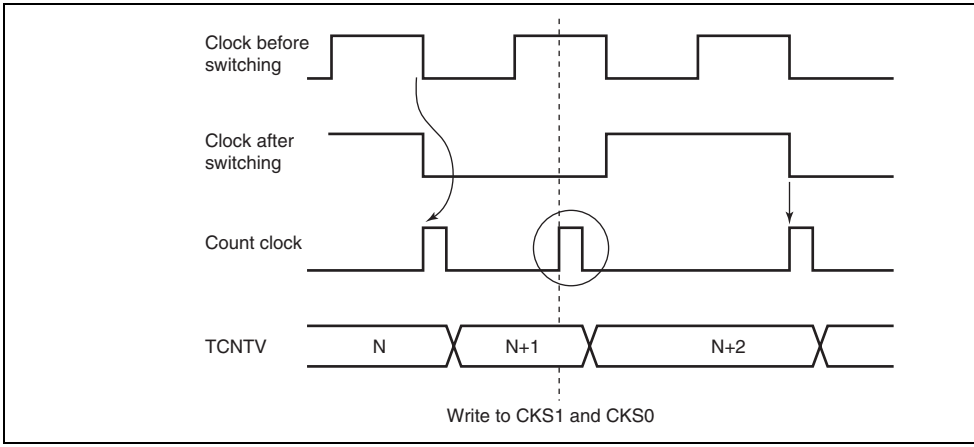


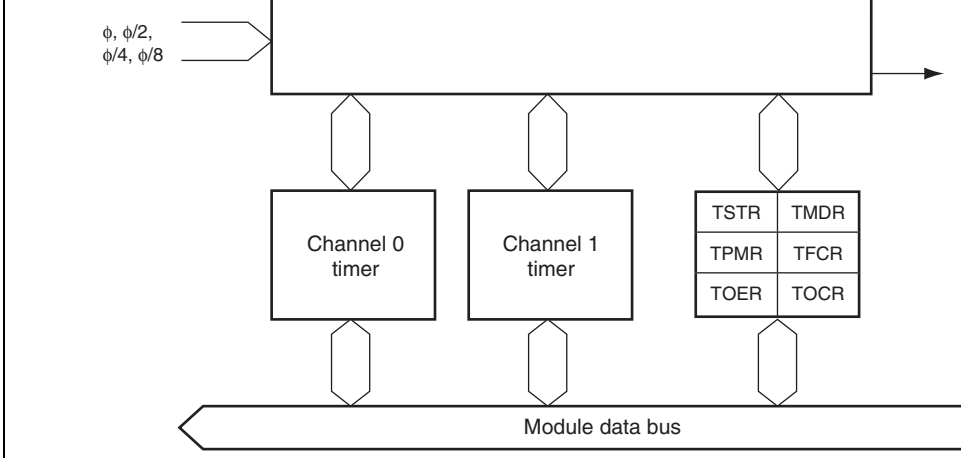
Figure 12.13 Internal Clock Switching and TCNTV Operation

— Independently assignable output compare or input capture functions

- Selection of five counter clock sources: four internal clocks (φ, φ/2, φ/4, and φ/8) and external clock
- Seven selectable operating modes
  - — Output compare function
    Selection of 0 output, 1 output, or toggle output
  - — Input capture function
    Rising edge, falling edge, or both edges
  - — Synchronous operation
    Timer counters_0 and _1 (TCNT_0 and TCNT_1) can be written simultaneously
    Simultaneous clearing by compare match or input capture is possible.
  - — PWM mode
    Up to six-phase PWM output can be provided with desired duty ratio.
  - — Reset synchronous PWM mode
    Three-phase PWM output for normal and counter phases
  - — Complementary PWM mode
    Three-phase PWM output for non-overlapped normal and counter phases
    The A/D conversion start trigger can be set for PWM cycles.
  - — Buffer operation
    The input capture register can be consisted of double buffers.
    The output compare register can automatically be modified.
- High-speed access by the internal 16-bit bus
  - — 16-bit TCNT and GR registers can be accessed in high speed by a 16-bit bus inte
- Any initial timer output value can be set
- Output of the timer is disabled by external trigger

capture registers)

Buffer register		GRC_0, GRD_0	GRC_1, GRD_1
I/O pins		FTIOA0, FTIOB0, FTIOC0, FTIOD0	FTIOA1, FTIOB1, FTIOC1 FTIOD1
Counter clearing function		Compare match/input capture of GRA_0, GRB_0, GRC_0, or GRD_0	Compare match/input cap GRA_1, GRB_1, GRC_1, GRD_1
Compare match output	0 output	Yes	Yes
	1 output	Yes	Yes
	output	Yes	Yes
Input capture function		Yes	Yes
Synchronous operation		Yes	Yes
PWM mode		Yes	Yes
Reset synchronous PWM mode		Yes	Yes
Complementary PWM mode		Yes	Yes
Buffer function		Yes	Yes
Interrupt sources		Compare match/input capture A0 to D0 Overflow	Compare match/input cap to D1 Overflow Underflow



[Legend]

TSTR: Timer start register (8 bits)

TMDR: Timer mode register (8 bits)

TPMR: Timer PWM mode register (8 bits)

TFCR: Timer function control register (8 bits)

TOER: Timer output master enable register (8 bits)

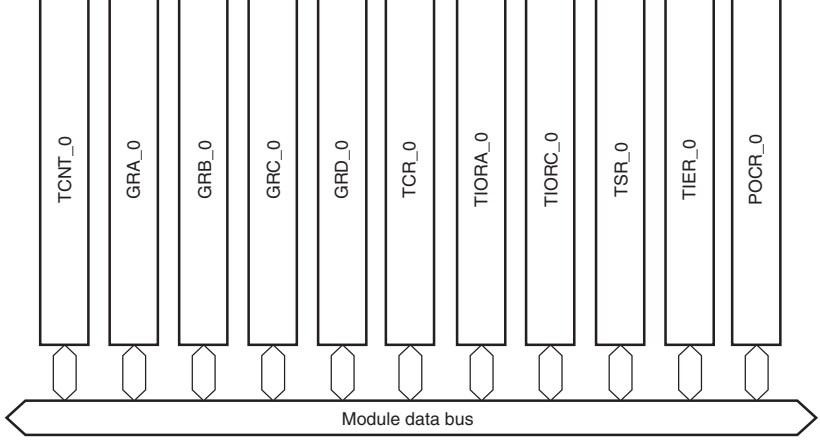
TOCR: Timer output control register (8 bits)

$\overline{\text{ADTRG}}$: A/D conversion start trigger output signal

ITMZ0: Channel 0 interrupt

ITMZ1: Channel 1 interrupt

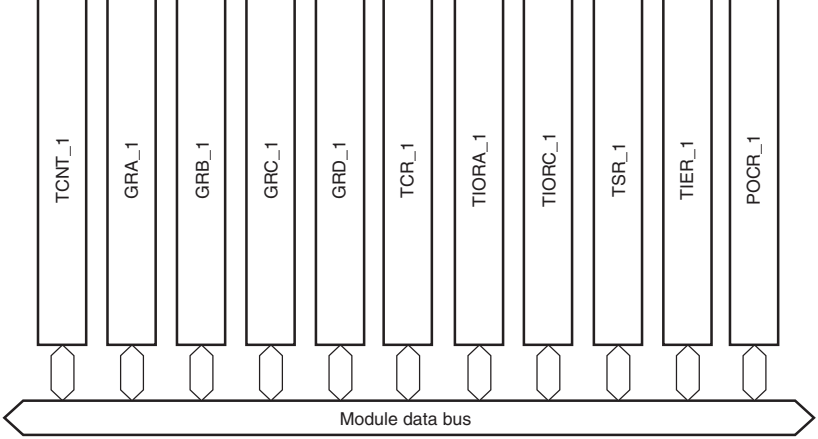
Figure 13.1 Timer Z Block Diagram



[Legend]

TCNT_0:	Timer counter_0 (16 bits)
GRA_0, GRB_0: GRC_0, GRD_0 :	General registers A_0, B_0, C_0, and D_0 (input capture/output compare registers: 16 bits × 4)
TCR_0:	Timer control register_0 (8 bits)
TIORA_0:	Timer I/O control register A_0 (8 bits)
TIORC_0:	Timer I/O control register C_0 (8 bits)
TSR_0:	Timer status register_0 (8 bits)
TIER_0:	Timer interrupt enable register_0 (8 bits)
POCR_0:	PWM mode output level control register_0 (8 bits)
ITMZ0:	Channel 0 interrupt

Figure 13.2 Timer Z (Channel 0) Block Diagram



[Legend]

TCNT_1:	Timer counter_1 (16 bits)
GRA_1, GRB_1:	General registers A_1, B_1, C_1, and D_1 (input capture/output compare registers:
GRC_1, GRD_1:	16 bits × 4)
TCR_1:	Timer control register_1 (8 bits)
TIORA_1:	Timer I/O control register A_1 (8 bits)
TIORC_1:	Timer I/O control register C_1 (8 bits)
TSR_1:	Timer status register_1 (8 bits)
TIER_1:	Timer interrupt enable register_1 (8 bits)
POCR_1:	PWM mode output level control register_1 (8 bits)
ITMZ1:	Channel 1 interrupt

Figure 13.3 Timer Z (Channel 1) Block Diagram

compare B0			input capture input, or PWM ou
Input capture/output compare C0	FTIOC0	Input/output	GRC_0 output compare output input capture input, or PWM synchronous output (in reset synchronous PWM and comple PWM modes)
Input capture/output compare D0	FTIOD0	Input/output	GRD_0 output compare output input capture input, or PWM ou
Input capture/output compare A1	FTIOA1	Input/output	GRA_1 output compare output input capture input, or PWM ou reset synchronous PWM and complementary PWM modes)
Input capture/output compare B1	FTIOB1	Input/output	GRB_1 output compare output input capture input, or PWM ou
Input capture/output compare C1	FTIOC1	Input/output	GRC_1 output compare output input capture input, or PWM ou
Input capture/output compare D1	FTIOD1	Input/output	GRD_1 output compare output input capture input, or PWM ou

- Timer output control register (TOCR)

Channel 0

- Timer control register_0 (TCR_0)
- Timer I/O control register A_0 (TIORA_0)
- Timer I/O control register C_0 (TIORC_0)
- Timer status register_0 (TSR_0)
- Timer interrupt enable register_0 (TIER_0)
- PWM mode output level control register_0 (POCR_0)
- Timer counter_0 (TCNT_0)
- General register A_0 (GRA_0)
- General register B_0 (GRB_0)
- General register C_0 (GRC_0)
- General register D_0 (GRD_0)

Channel 1

- Timer control register_1 (TCR_1)
- Timer I/O control register A_1 (TIORA_1)
- Timer I/O control register C_1 (TIORC_1)
- Timer status register_1 (TSR_1)
- Timer interrupt enable register_1 (TIER_1)
- PWM mode output level control register_1 (POCR_1)
- Timer counter_1 (TCNT_1)
- General register A_1 (GRA_1)
- General register B_1 (GRB_1)

				These bits are always read as 1, and cannot be
1	STR1	0	R/W	Channel 1 Counter Start 0: TCNT_1 halts counting 1: TCNT_1 starts counting
0	STR0	0	R/W	Channel 0 Counter Start 0: TCNT_0 halts counting 1: TCNT_0 starts counting

13.3.2 Timer Mode Register (TMDR)

TMDR selects buffer operation settings and synchronized operation.

Bit	Bit Name	Initial Value	R/W	Description
7	BFD1	0	R/W	Buffer Operation D1 0: GRD_1 operates normally 1: GRB_1 and GRD_1 are used together for buff operation
6	BFC1	0	R/W	Buffer Operation C1 0: GRC_1 operates normally 1: GRA_1 and GRD_1 are used together for buff operation
5	BFD0	0	R/W	Buffer Operation D0 0: GRD_0 operates normally 1: GRB_0 and GRD_0 are used together for buff operation

				1: TCNT_1 and TCNT_0 are synchronized TCNT_1 and TCNT_0 can be pre-set or clea synchronously

13.3.3 Timer PWM Mode Register (TPMR)

TPMR sets the pin to enter PWM mode.

Bit	Bit Name	Initial Value	R/W	Description
7	—	1	—	Reserved This bit is always read as 1, and cannot be mod
6	PWMD1	0	R/W	PWM Mode D1 0: FTIOD1 operates normally 1: FTIOD1 operates in PWM mode
5	PWMC1	0	R/W	PWM Mode C1 0: FTIOC1 operates normally 1: FTIOC1 operates in PWM mode
4	PWMB1	0	R/W	PWM Mode B1 0: FTIOB1 operates normally 1: FTIOB1 operates in PWM mode
3	—	1	—	Reserved This bit is always read as 1, and cannot be mod

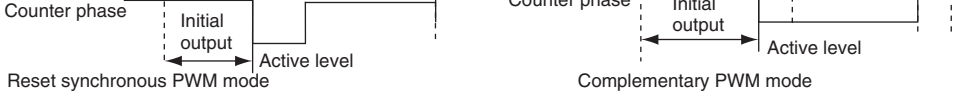
1: FTIOB0 operates in PWM mode

13.3.4 Timer Function Control Register (TFCR)

TFCR selects the settings and output levels for each operating mode.

Bit	Bit Name	Initial Value	R/W	Description
7	—	1	—	Reserved This bit is always read as 1.
6	STCLK	0	R/W	External Clock Input Select 0: External clock input is disabled 1: External clock input is enabled
5	ADEG	0	R/W	A/D Trigger Edge Select A/D module should be set to start an A/D conver the external trigger 0: A/D trigger at the crest in complementary PWI 1: A/D trigger at the trough in complementary PW
4	ADTRG	0	R/W	External Trigger Disable 0: A/D trigger for PWM cycles is disabled in complementary PWM mode 1: A/D trigger for PWM cycles is enabled in complementary PWM mode

				0: Initial output is high and the active level is lo
				1: Initial output is low and the active level is hig
				Figure 13.4 shows an example of outputs in re synchronous PWM mode and complementary P mode when OLS1 = 0 and OLS0 = 0.
1	CMD1	0	R/W	Combination Mode 1 and 0
0	CMD0	0	R/W	00: Channel 0 and channel 1 operate normally
				01: Channel 0 and channel 1 are used together operate in reset synchronous PWM mode
				10: Channel 0 and channel 1 are used together operate in complementary PWM mode (tra the trough)
				11: Channel 0 and channel 1 are used together operate in complementary PWM mode (tra the crest)
				Note: When reset synchronous PWM mode or complementary PWM mode is selected b bits, this setting has the priority to the set PWM mode by each bit in TPMR. Stop T and TCNT_1 before making settings for r synchronous PWM mode or complement mode.



Note: Write H'00 to TOCR to start initial outputs after stopping the counter.

Figure 13.4 Example of Outputs in Reset Synchronous PWM Mode and Complementary PWM Mode

				1: FTIOD1 pin output is disabled regardless of t TFCR, and TIORC_1 settings (FTIOD1 pin is as an I/O port).
6	EC1	1	R/W	Master Enable C1 0: FTIOC1 pin output is enabled according to th TFCR, and TIORC_1 settings 1: FTIOC1 pin output is disabled regardless of t TFCR, and TIORC_1 settings (FTIOC1 pin is as an I/O port).
5	EB1	1	R/W	Master Enable B1 0: FTIOB1 pin output is enabled according to th TFCR, and TIORA_1 settings 1: FTIOB1 pin output is disabled regardless of t TFCR, and TIORA_1 settings (FTIOB1 pin is as an I/O port).
4	EA1	1	R/W	Master Enable A1 0: FTIOA1 pin output is enabled according to th TFCR, and TIORA_1 settings 1: FTIOA1 pin output is disabled regardless of t TFCR, and TIORA_1 settings (FTIOA1 pin is as an I/O port).
3	ED0	1	R/W	Master Enable D0 0: FTIOD0 pin output is enabled according to th TFCR, and TIORC_0 settings 1: FTIOD0 pin output is disabled regardless of t TFCR, and TIORC_0 settings (FTIOD0 pin is as an I/O port).

				1: FTIOB0 pin output is disabled regardless of th TFCR, and TIORA_0 settings (FTIOB0 pin is as an I/O port).
0	EA0	1	R/W	Master Enable A0 0: FTIOA0 pin output is enabled according to the TFCR, and TIORA_0 settings 1: FTIOA0 pin output is disabled regardless of th TFCR, and TIORA_0 settings (FTIOA0 pin is as an I/O port).

13.3.6 Timer Output Control Register (TOCR)

TOCR selects the initial outputs before the first occurrence of a compare match. Note tha OLS1 and OLS0 in TFCR set these initial outputs in reset synchronous PWM mode and complementary PWM mode.

Bit	Bit Name	Initial Value	R/W	Description
7	TOD1	0	R/W	Output Level Select D1 0: 0 output at the FTIOD1 pin* 1: 1 output at the FTIOD1 pin*
6	TOC1	0	R/W	Output Level Select C1 0: 0 output at the FTIOC1 pin* 1: 1 output at the FTIOC1 pin*

				1: 1 output at the FTIOD0 pin*
2	TOC0	0	R/W	Output Level Select C0 0: 0 output at the FTIOC0 pin* 1: 1 output at the FTIOC0 pin*
1	TOB0	0	R/W	Output Level Select B0 0: 0 output at the FTIOB0 pin* 1: 1 output at the FTIOB0 pin*
0	TOA0	0	R/W	Output Level Select A0 0: 0 output at the FTIOA0 pin* 1: 1 output at the FTIOA0 pin*

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

13.3.7 Timer Counter (TCNT)

The timer Z has two TCNT counters (TCNT_0 and TCNT_1), one for each channel. Th counters are 16-bit readable/writable registers that increment/decrement according to inp Input clocks can be selected by bits TPSC2 to TPSC0 in TCR. TCNT0 and TCNT 1 increment/decrement in complementary PWM mode, while they only increment in othe

The TCNT counters are initialized to H'0000 by compare matches with corresponding G GRC, or GRD, or input captures to GRA, GRB, GRC, or GRD (counter clearing functio the TCNT counters overflow, an OVF flag in TSR for the corresponding channel is set t TCNT_1 underflows, an UDF flag in TSR is set to 1. The TCNT counters cannot be acc bit units; they must always be accessed as a 16-bit unit.

external signals. At this point, IMFA to IMFD flags in the corresponding TSR are set to 1
Detection edges for input capture signals can be selected by TIORA and TIORC.

When PWM mode, complementary PWM mode, or reset synchronous PWM mode is sele
values in TIORA and TIORC are ignored. Upon reset, the GR registers are set as output c
registers (no output) and initialized to H'FFFF. The GR registers cannot be accessed in 8-
they must always be accessed as a 16-bit unit.

				010: Clears TCNT by GRB compare match/inpu capture*1
				011: Synchronization clear; Clears TCNT in syn with counter clearing of the other channel'
				100: Disables TCNT clearing
				101: Clears TCNT by GRC compare match/inpu capture*1
				110: Clears TCNT by GRD compare match/inpu capture*1
				111: Synchronization clear; Clears TCNT in syn with counter clearing of the other channel'
4	CKEG1	0	R/W	Clock Edge 1 and 0
3	CKEG0	0	R/W	00: Count at rising edge
				01: Count at falling edge
				1X: Count at both edges
2	TPSC2	0	R/W	Time Prescaler 2 to 0
1	TPSC1	0	R/W	000: Internal clock: count by ϕ
0	TPSC0	0	R/W	001: Internal clock: count by $\phi/2$
				010: Internal clock: count by $\phi/4$
				011: Internal clock: count by $\phi/8$
				1XX: External clock: count by FTIOA0 (TCLK) p

Notes: 1. When GR functions as an output compare register, TCNT is cleared by comp
When GR functions as input capture, TCNT is cleared by input capture.
2. Synchronous operation is set by TMDR.
3. X: Don't care

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 to B0
5	IOB1	0	R/W	GRB is an output compare register:
4	IOB0	0	R/W	000: Disables pin output by compare match 001: 0 output by GRB compare match 010: 1 output by GRB compare match 011: Toggle output by GRB compare match GRB is an input capture register: 100: Input capture to GRB at the rising edge 101: Input capture to GRB at the falling edge 11X: Input capture to GRB at both rising and falli
3	—	1	—	Reserved This bit is always read as 1.

101: Input capture to GRA at the falling edge

11X: Input capture to GRA at both rising and fa

[Legend]

X: Don't care

- TIORC

  TIORC selects whether GRC or GRD is used as an output compare register or an inp register. When an output compare register is selected, the output setting is selected. input capture register is selected, an input edge of an input capture signal is selected. also selects the function of FTIOC or FTIOD pin.

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 to D0
5	IOD1	0	R/W	GRD is an output compare register:
4	IOD0	0	R/W	000: Disables pin output by compare match 001: 0 output by GRD compare match 010: 1 output by GRD compare match 011: Toggle output by GRD compare match GRD is an input capture register: 100: Input capture to GRD at the rising edge 101: Input capture to GRD at the falling edge 11X: Input capture to GRD at both rising and fa edges

				GRC is an input capture register: 100: Input capture to GRC at the rising edge 101: Input capture to GRC at the falling edge 11X: Input capture to GRC at both rising and fall edges

[Legend]
X: Don't care

13.3.11 Timer Status Register (TSR)

TSR indicates generation of an overflow/underflow of TCNT and a compare match/input of GRA, GRB, GRC, and GRD. These flags are interrupt sources. If an interrupt is enab corresponding bit in TIER, TSR requests an interrupt for the CPU. Timer Z has two TSR one for each channel.

Bit	Bit Name	Initial value	R/W	Description
7, 6	—	All 1	—	Reserved These bits are always read as 1.
5	UDF*	0	R/W	Underflow Flag [Setting condition] • When TCNT_1 underflows [Clearing condition] • When 0 is written to UDF after reading UDF =

				compare register • When TCNT value is transferred to GRD by capture signal and GRD is functioning as in capture register [Clearing condition] • When 0 is written to IMFD after reading IMF
2	IMFC	0	R/W	Input Capture/Compare Match Flag C [Setting conditions] • When TCNT = GRC and GRC is functioning compare register • When TCNT value is transferred to GRC by capture signal and GRC is functioning as in capture register [Clearing condition] • When 0 is written to IMFC after reading IMF
1	IMFB	0	R/W	Input Capture/Compare Match Flag B [Setting conditions] • When TCNT = GRB and GRB is functioning compare register • When TCNT value is transferred to GRB by capture signal and GRB is functioning as in capture register [Clearing condition] • When 0 is written to IMFB after reading IMF

- When 0 is written to IMFA after reading IMFA

Note: Bit 5 is not the UDF flag in TSR_0. It is a reserved bit. It is always read as 1.

13.3.12 Timer Interrupt Enable Register (TIER)

TIER enables or disables interrupt requests for overflow or GR compare match/input capt
Timer Z has two TIER registers, one for each channel.

Bit	Bit Name	Initial value	R/W	Description
7 to 5	—	All 1	—	Reserved These bits are always read as 1.
4	OVIE	0	R/W	Overflow Interrupt Enable 0: Interrupt requests (OVI) by OVF or UDF flag a disabled 1: Interrupt requests (OVI) by OVF or UDF flag a enabled
3	IMIED	0	R/W	Input Capture/Compare Match Interrupt Enable D 0: Interrupt requests (IMID) by IMFD flag are disa 1: Interrupt requests (IMID) by IMFD flag are ena
2	IMIEC	0	R/W	Input Capture/Compare Match Interrupt Enable C 0: Interrupt requests (IMIC) by IMFC flag are disa 1: Interrupt requests (IMIC) by IMFC flag are ena

13.3.13 PWM Mode Output Level Control Register (POCR)

POCR control the active level in PWM mode. Timer Z has two POCR registers, one for channel.

Bit	Bit Name	Initial value	R/W	Description
7 to 3	—	All 1	—	Reserved These bits are always read as 1.
2	POLD	0	R/W	PWM Mode Output Level Control D 0: The output level of FTIOD is low-active 1: The output level of FTIOD is high-active
1	POLC	0	R/W	PWM Mode Output Level Control C 0: The output level of FTIOC is low-active 1: The output level of FTIOC is high-active
0	POLB	0	R/W	PWM Mode Output Level Control B 0: The output level of FTIOB is low-active 1: The output level of FTIOB is high-active

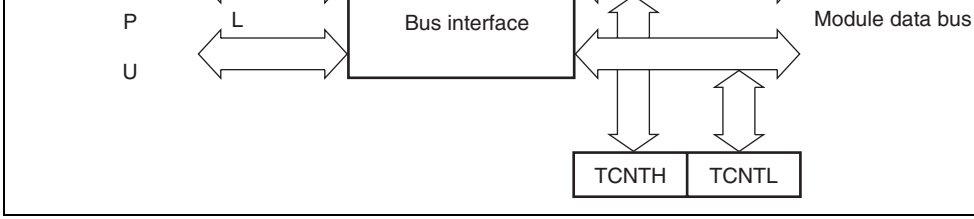


Figure 13.5 Accessing Operation of 16-Bit Register (between CPU and TCNT (16

2. 8-bit register

Registers other than TCNT and GR are 8-bit registers that are connected internally wi
CPU in an 8-bit width. Figure 13.6 shows an example of accessing the 8-bit registers.

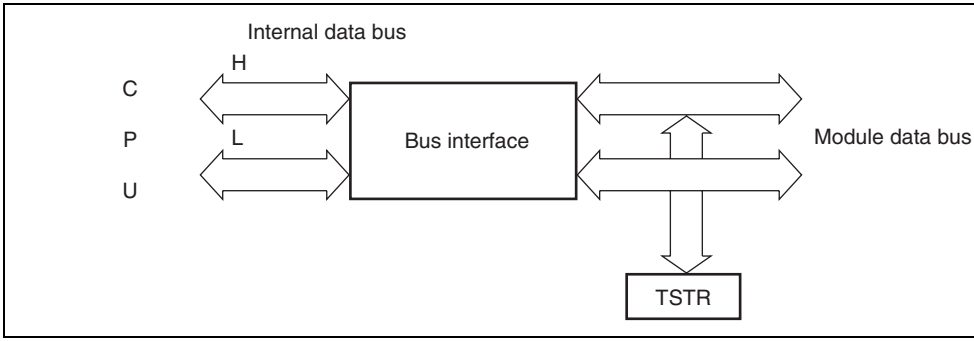
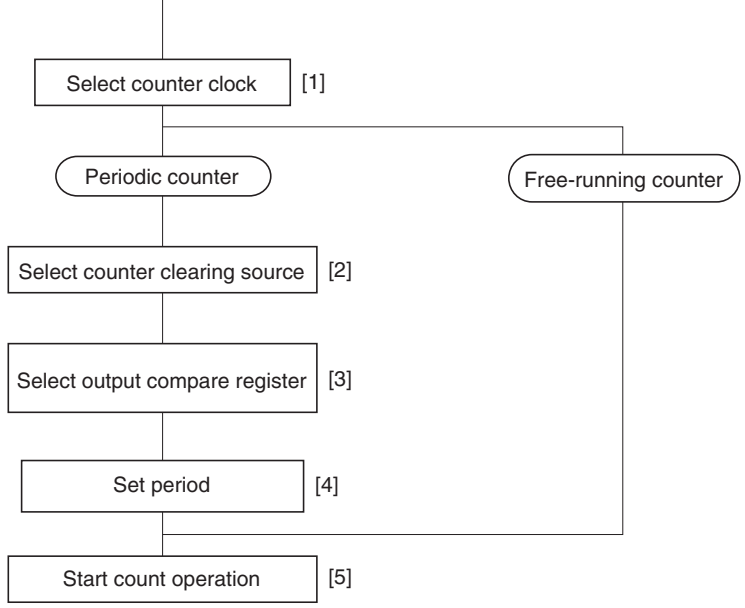


Figure 13.6 Accessing Operation of 8-Bit Register (between CPU and TSTR (8



TPSC2 to TPS
TCR. When an
clock is select
the external cl
with bits CKEG
and CKEG0 in
[2] For periodic co
operation, sele
TCNT clearing
source with bit
CCLR2 to CCL
TCR.
[3] Designate the
register select
as an output c
register by me
TIOR.
[4] Set the period
cycle in the ge
register select
in [2].
[5] Set the STR b
to 1 to start the
operation.

Figure 13.7 Example of Counter Operation Setting Procedure

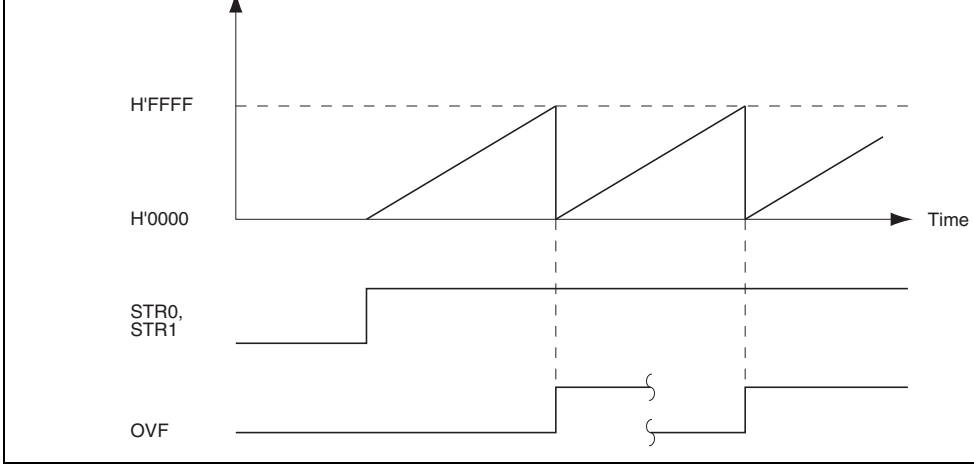


Figure 13.8 Free-Running Counter Operation

When compare match is selected as the TCNT clearing source, the TCNT counter for the
channel performs periodic count operation. The GR registers for setting the period are de
as output compare registers, and counter clearing by compare match is selected by means
CCLR1 and CCLR0 in TCR. After the settings have been made, TCNT starts an increme
operation as a periodic counter when the corresponding bit in TSTR is set to 1. When the
value matches the value in GR, the IMFA, IMFB, IMFC, or IMFD flag in TSR is set to 1
TCNT is cleared to H'0000.

If the value of the corresponding IMIEA, IMIEB, IMIEC, or IMIED bit in TIER is 1 at th
the timer Z requests an interrupt. After a compare match, TCNT starts an increment opera
again from H'0000.

Figure 13.9 illustrates periodic counter operation.

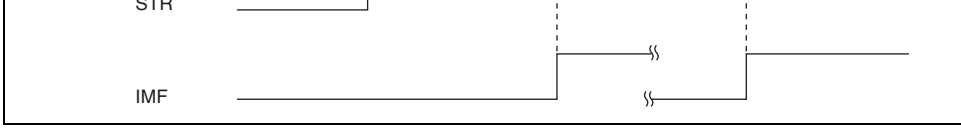


Figure 13.9 Periodic Counter Operation

2. TCNT count timing

   A. Internal clock operation

   A system clock (ϕ) or three types of clocks (ϕ/2, ϕ/4, or ϕ/8) that divides the syst
   can be selected by bits TPSC2 to TPSC0 in TCR.

   Figure 13.10 illustrates this timing.

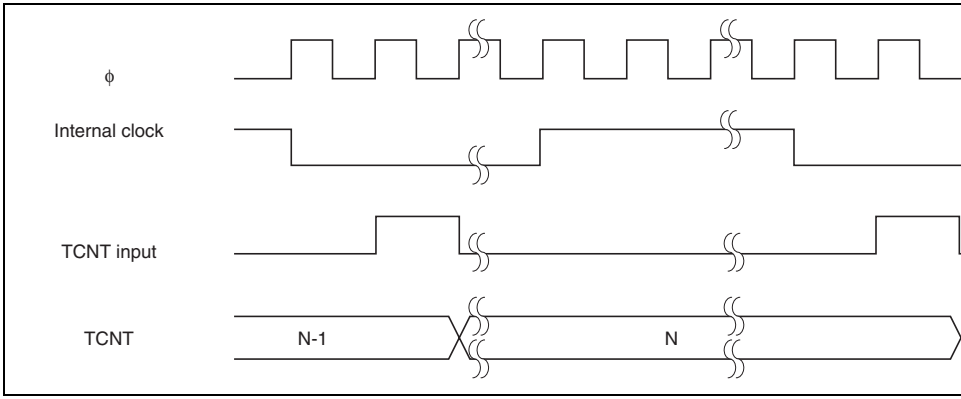


Figure 13.10 Count Timing at Internal Clock Operation

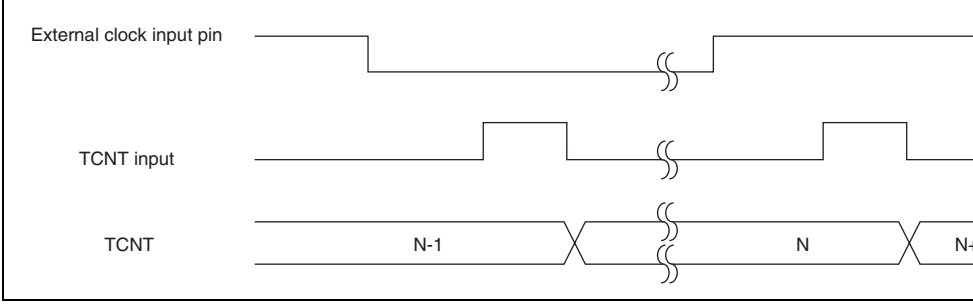
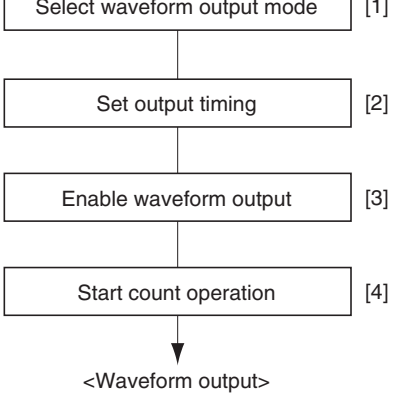


Figure 13.11 Count Timing at External Clock Operation (Both Edges Detect



[1] Select the waveform output mode in TIOR. Waveform output unit the first compare match occurs.
[2] Set the timing for compare match generation in GRA/GRB/GRC/GRD.
[3] Enable or disable the timer output by TOER.
[4] Set the STR bit in TSTR to 1 to start the TCNT count operation.

Figure 13.12 Example of Setting Procedure for Waveform Output by Compare

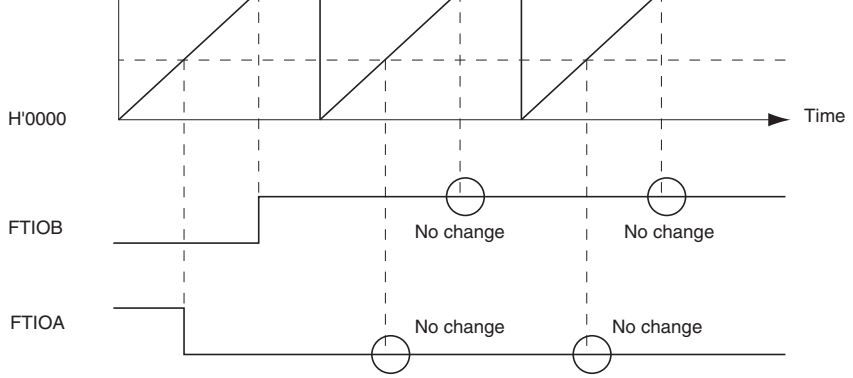


Figure 13.13 Example of 0 Output/1 Output Operation

Figure 13.14 shows an example of toggle output.

In this example, TCNT has been designated as a periodic counter (with counter clearin
compare match B), and settings have been made such that the output is toggled by bot
compare match A and compare match B.

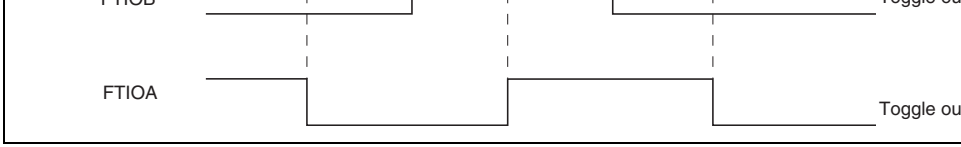


Figure 13.14 Example of Toggle Output Operation

2. Output compare timing

   The compare match signal is generated in the last state in which TCNT and GR matc
   TCNT changes from the matching value to the next value). When the compare matc
   generated, the output value selected in TIOR is output at the compare match output p
   (FTIOA, FTIOB, FTIOC, or FTIOD). When TCNT matches GR, the compare match
   generated only after the next TCNT input clock pulse is input.

   Figure 13.15 shows an example of the output compare timing.

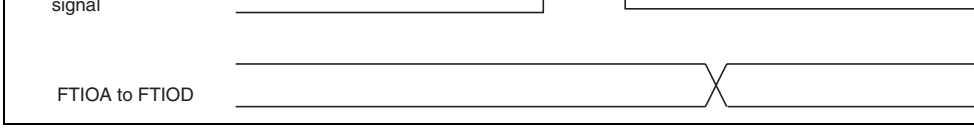


Figure 13.15 Output Compare Timing

13.4.3 Input Capture Function

The TCNT value can be transferred to GR on detection of the input edge of the input capture/output compare pin (FTIOA, FTIOB, FTIOC, or FTIOD). Rising edge, falling ed both edges can be selected as the detected edge. When the input capture function is used, width or period can be measured.

Figure 13.16 shows an example of the input capture operation setting procedure.

<Input capture operation>

Figure 13.16 Example of Input Capture Operation Setting Procedure

1. Example of input capture operation

   Figure 13.17 shows an example of input capture operation.

   In this example, both rising and falling edges have been selected as the FTIOA pin i
   capture input edge, the falling edge has been selected as the FTIOB pin input capture
   edge, and counter clearing by GRB input capture has been designated for TCNT.

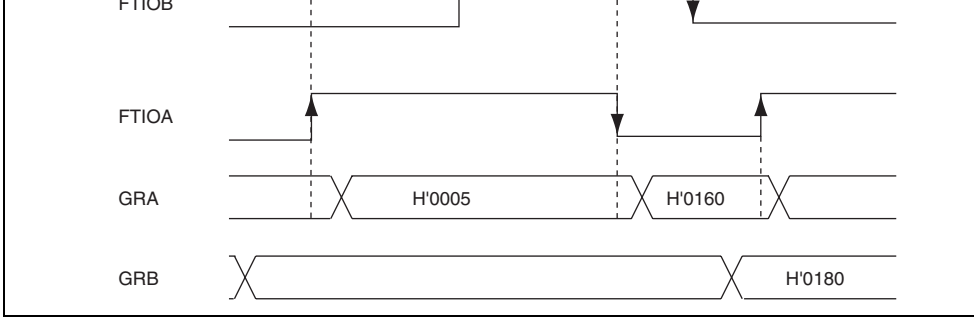


Figure 13.17 Example of Input Capture Operation

2. Input capture signal timing

   Input capture on the rising edge, falling edge, or both edges can be selected through s
   TIOR. Figure 13.18 shows the timing when the rising edge is selected. The pulse widt
   input capture signal must be at least two system clock (ϕ) cycles.

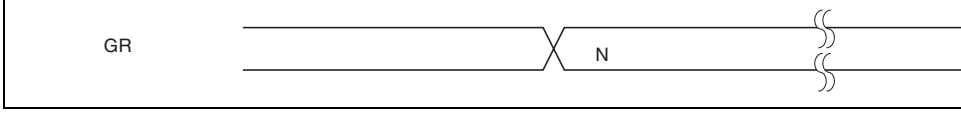
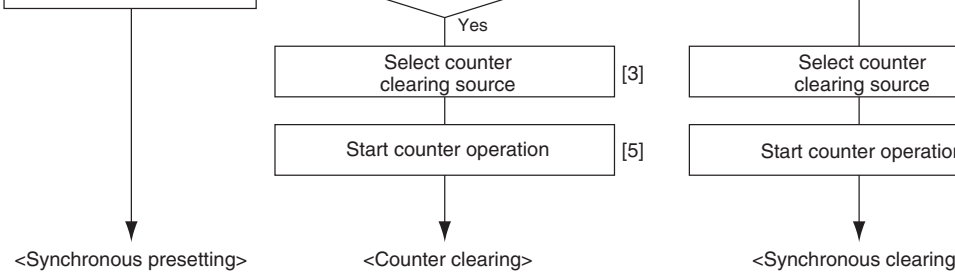


Figure 13.18 Input Capture Signal Timing

13.4.4 Synchronous Operation

In synchronous operation, the values in a number of TCNT counters can be rewritten simultaneously (synchronous presetting). Also, a number of TCNT counters can be clea simultaneously by making the appropriate setting in TCR (synchronous clearing). Synch operation enables GR to be increased with respect to a single time base.

Figure 13.19 shows an example of the synchronous operation setting procedure.



[1] Set the SYNC bits in TMDR to 1.
[2] When a value is written to either of the TCNT counters, the same value is simultaneously written to the other TCNT counter.
[3] Set bits CCLR1 and CCLR0 in TCR to specify counter clearing by compare match/input capture.
[4] Set bits CCLR1 and CCLR0 in TCR to designate synchronous clearing for the counter clearing source
[5] Set the STR bit in TSTR to 1 to start the count operation.

Figure 13.19 Example of Synchronous Operation Setting Procedure

Figure 13.20 shows an example of synchronous operation. In this example, synchronous has been selected, FTIOB0 and FTIOB1 have been designated for PWM mode, GRA_0 c match has been set as the channel 0 counter clearing source, and synchronous clearing ha for the channel 1 counter clearing source. In addition, the same input clock has been set a counter input clock for channel 0 and channel 1. Two-phase PWM waveforms are output pins FTIOB0 and FTIOB1. At this time, synchronous presetting and synchronous operati GRA_0 compare match are performed by TCNT counters.

For details on PWM mode, see section 13.4.5, PWM Mode.

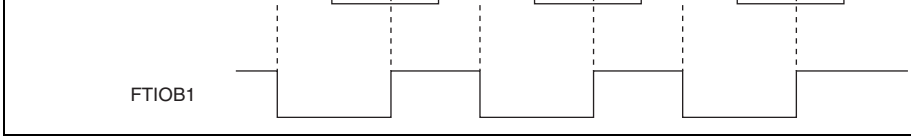


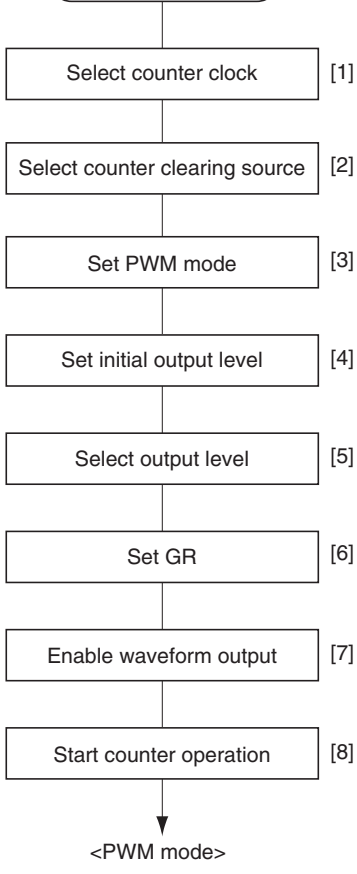
Figure 13.20 Example of Synchronous Operation

13.4.5 PWM Mode

In PWM mode, PWM waveforms are output from the FTIOB, FTIOC, and FTIOD outp
with GRA as a cycle register and GRB, GRC, and GRD as duty registers. The initial ou
of the corresponding pin depends on the setting values of TOCR and POCR. Table 13.3
example of the initial output level of the FTIOB0 pin.

The output level is determined by the POLB to POLD bits corresponding to POCR. Wh
is 0, the FTIOB output pin is set to 0 by compare match B and set to 1 by compare matc
When POLB is 1, the FTIOB output pin is set to 1 by compare match B and cleared to 0
compare match A. In PWM mode, maximum 6-phase PWM outputs are possible.

Figure 13.21 shows an example of the PWM mode setting procedure.



[1] Select the counter clock with bits TPSC2 to TOSC0 in TCR. When an external clock is selected, select the external clock edge with bits CKEG1 and CKEG0 in TCR.
[2] Use bits CCLR1 and CCLR0 in TCR to select the counter clearing source.
[3] Select the PWM mode with bits PWMB0 to PWMD0 and PWMB1 to PWMD1 in TPMR.
[4] Set the initial output value with bits TOB0 to TOD0 and TOB1 to TOD1 in TOCR.
[5] Set the output level with bits POLB to POLD in POCR.
[6] Set the cycle in GRA, and set the duty in the other GR.
[7] Enable or disable the timer output by TOER.
[8] Set the STR bit in TSTR to 1 and start the counter operation.

Figure 13.21 Example of PWM Mode Setting Procedure

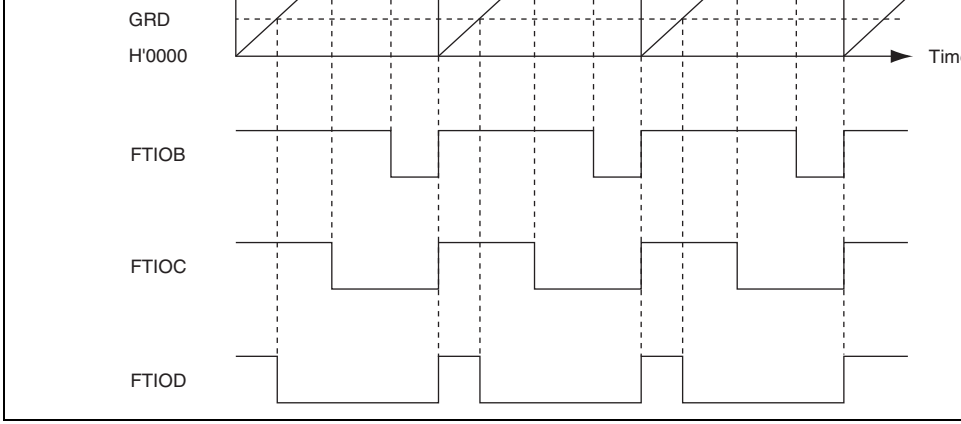


Figure 13.22 Example of PWM Mode Operation (1)

Figure 13.23 shows another example of operation in PWM mode. The output signals go
TCNT is reset at compare match A, and the output signals go to 1 at compare match B, (
(TOB, TOC, and TOD = 0, POLB, POLC, and POLD = 1).

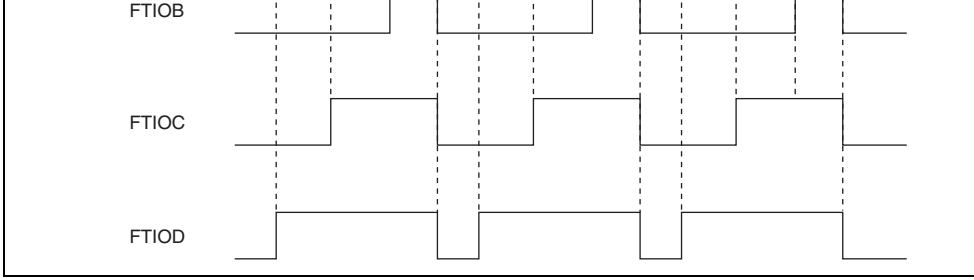


Figure 13.23 Example of PWM Mode Operation (2)

Figures 13.24 (when TOB, TOC, and TOD = 0, POLB, POLC, and POLD = 0) and 13.25
TOB, TOC, and TOD = 0, POLB, POLC, and POLD = 1) show examples of the output o
waveforms with duty cycles of 0% and 100% in PWM mode.

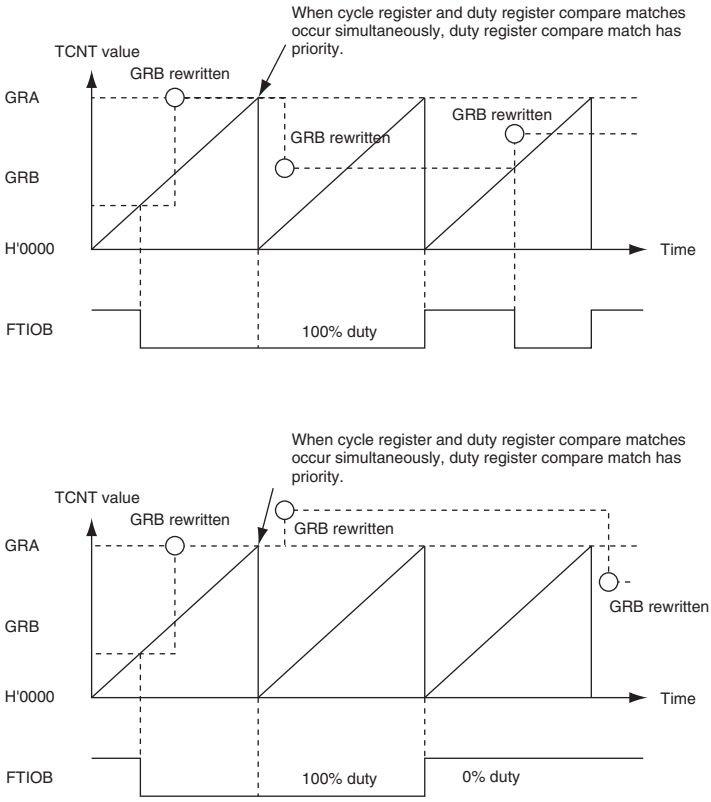


Figure 13.24 Example of PWM Mode Operation (3)

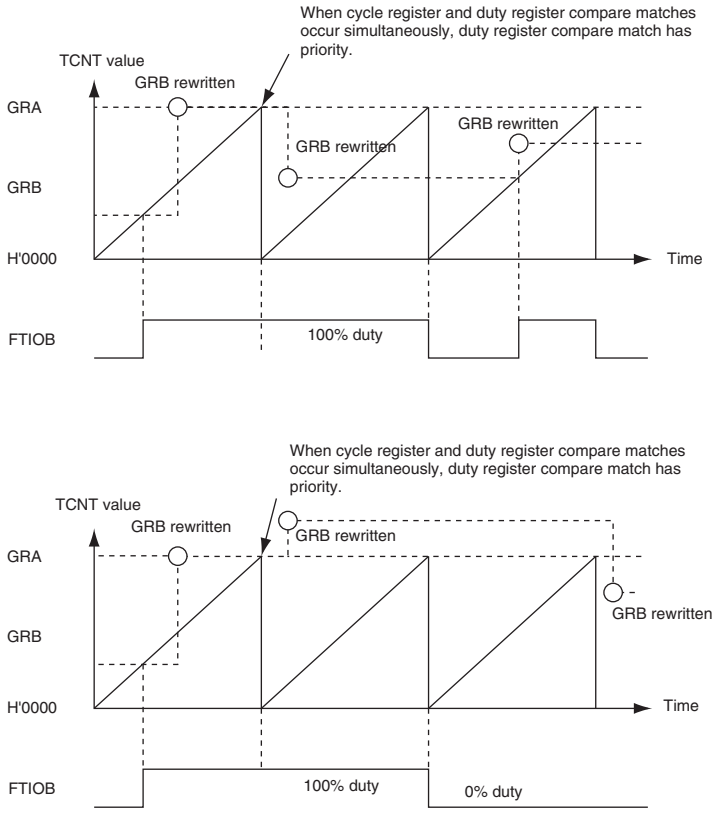
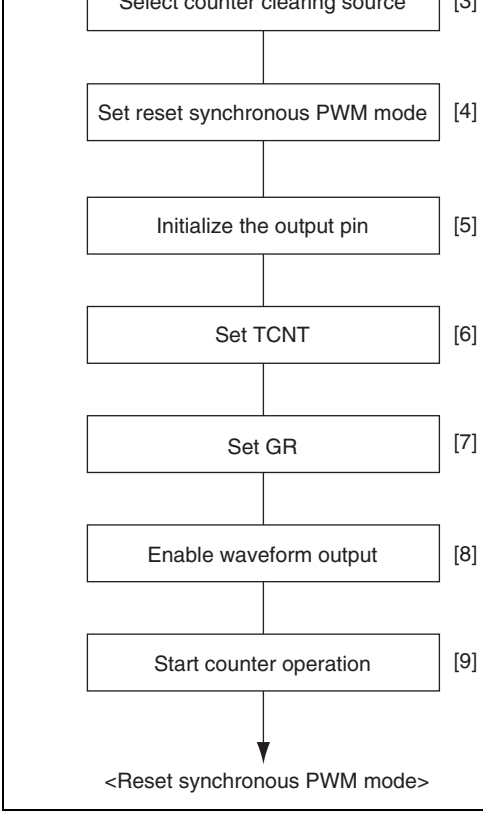


Figure 13.25 Example of PWM Mode Operation (4)

Channel	Pin Name	Input/Output	Pin Function
0	FTIOC0	Output	Toggle output in synchronous with PWM cyc
0	FTIOB0	Output	PWM output 1
0	FTIOD0	Output	PWM output 1 (counter-phase waveform of F output 1)
1	FTIOA1	Output	PWM output 2
1	FTIOC1	Output	PWM output 2 (counter-phase waveform of F output 2)
1	FTIOB1	Output	PWM output 3
1	FTIOD1	Output	PWM output 3 (counter-phase waveform of F output 3)

Table 13.5 Register Settings in Reset Synchronous PWM Mode

Register	Description
TCNT_0	Initial setting of H'0000
TCNT_1	Not used (independently operates)
GRA_0	Sets counter cycle of TCNT_0
GRB_0	Set a changing point of the PWM waveform output from pins FTIOB0 a FTIOD0.
GRA_1	Set a changing point of the PWM waveform output from pins FTIOA1 a FTIOC1.
GRB_1	Set a changing point of the PWM waveform output from pins FTIOB1 a FTIOD1.



[4] Select the reset synchronous PWM mo with bits CMD1 and CMD0 in TFCR. FTIOB0 to FTIOD0 and FTIOA1 to FTIOD1 become PWM output pins automatically.
[5] Set H'00 to TOCR.
[6] Set TCNT_0 as H'0000. TCNT1 does need to be set.
[7] GRA_0 is a cycle register. Set a cycle GRA_0. Set the changing point timing the PWM output waveform for GRB_0, GRA_1, and GRB_1.
[8] Enable or disable the timer output by TOER.
[9] Set the STR bit in TSTR to 1 and start counter operation.

Figure 13.26 Example of Reset Synchronous PWM Mode Setting Procedur

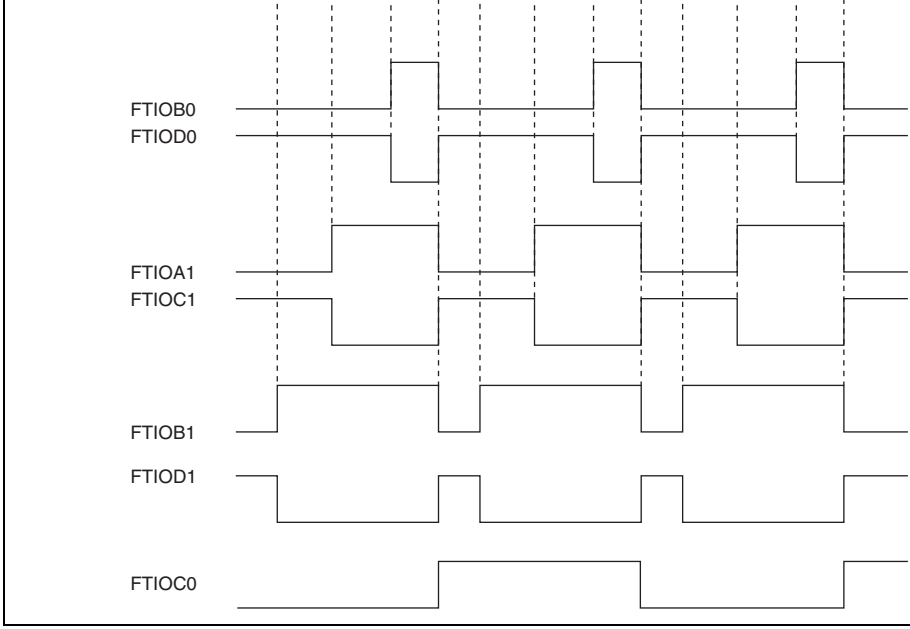


Figure 13.27 Example of Reset Synchronous PWM Mode Operation (OLS0 = O

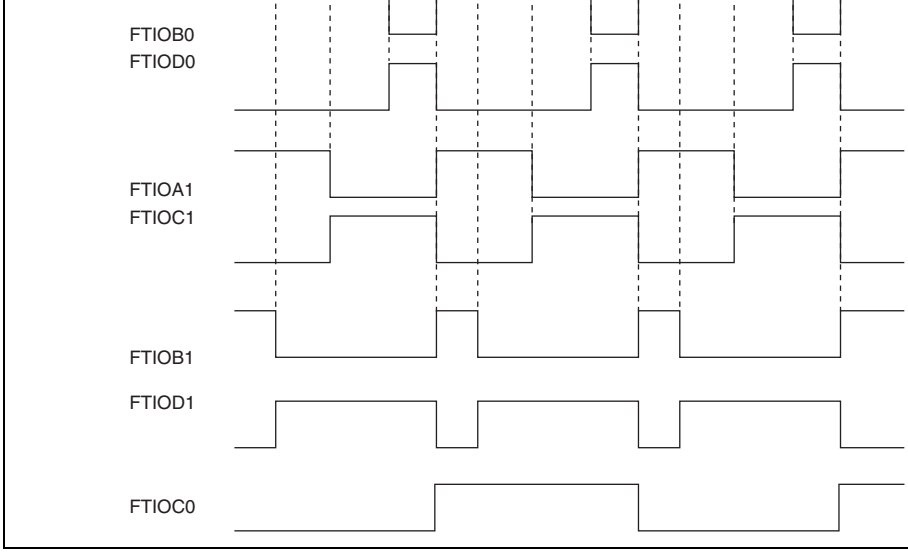


Figure 13.28 Example of Reset Synchronous PWM Mode Operation (OLS0 = OL

In reset synchronous PWM mode, TCNT_0 and TCNT_1 perform increment and indepen operations, respectively. However, GRA_1 and GRB_1 are separated from TCNT_1. W compare match occurs between TCNT_0 and GRA_0, a counter is cleared and an increm operation is restarted from H'0000.

The PWM pin outputs 0 or 1 whenever a compare match between GRB_0, GRA_1, GRB TCNT_0 or counter clearing occur.

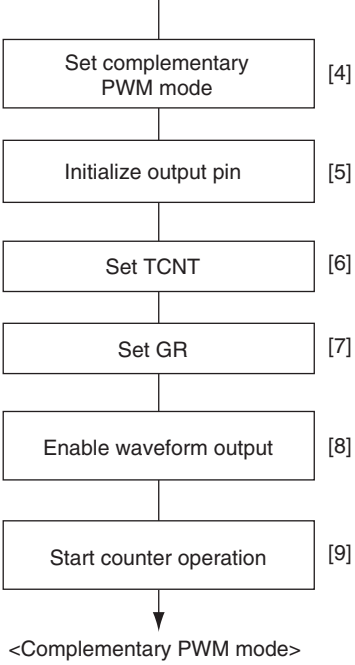
For details on operations when reset synchronous PWM mode and buffer operation are simultaneously set, refer to section 13.4.8, Buffer Operation.

Table 13.6 Output Pins in Complementary PWM Mode

Channel	Pin Name	Input/Output	Pin Function
0	FTIOC0	Output	Toggle output in synchronous with PWM cyc
0	FTIOB0	Output	PWM output 1
0	FTIOD0	Output	PWM output 1 (counter-phase waveform nor overlapped with PWM output 1)
1	FTIOA1	Output	PWM output 2
1	FTIOC1	Output	PWM output 2 (counter-phase waveform nor overlapped with PWM output 2)
1	FTIOB1	Output	PWM output 3
1	FTIOD1	Output	PWM output 3 (counter-phase waveform nor overlapped with PWM output 3)

Table 13.7 Register Settings in Complementary PWM Mode

Register	Description
TCNT_0	Initial setting of non-overlapped periods (non-overlapped periods are di with TCNT_1)
TCNT_1	Initial setting of H'0000
GRA_0	Sets (upper limit value – 1) of TCNT_0
GRB_0	Set a changing point of the PWM waveform output from pins FTIOB0 a FTIOD0.
GRA_1	Set a changing point of the PWM waveform output from pins FTIOA1 a FTIOC1.
GRB_1	Set a changing point of the PWM waveform output from pins FTIOB1 a FTIOD1.



[4] Use bits CMD1 and CMD0 in TFCR to set complementary PWM mode. FTIOB0 to FTIOD0 and FTIOA1 to FTIOD1 automatically become PWM output pins.
[5] Set H'00 to TOCR.
[6] TCNT_1 must be H'0000. Set a non-overlapped period to TCNT_0.
[7] GRA_0 is a cycle register. Set the cycle to GRA_0. Set the timing to change the PWM output waveform to GRB_0, GRA_1, and GRB_1. Note that the timing must be set within the range of compare match carried out for TCNT_0 and TCNT_1. For GR settings, see 3. Setting GR Value in Complementary PWM Mode in section 13.4.7.
[8] Use TOER to enable or disable the timer output.
[9] Set the STR0 and STR1 bits in TSTR to 1 to start the count operation.

Note: To re-enter complementary PWM mode, first, enter a mode other than the complement PWM mode. After that, repeat the setting procedures from step [1].
For settings of waveform outputs with a duty cycle of 0% and 100%, see the settings sh in 2. Examples of Complementary PWM Mode Operation and 3. Setting GR Value in Complementary PWM Mode in section 13.4.7.

Figure 13.29 Example of Complementar y PWM Mode Setting Procedure

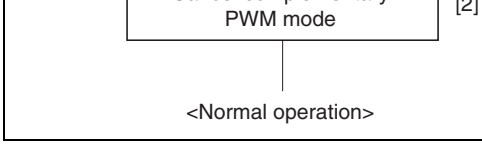


Figure 13.30 Canceling Procedure of Complementary PWM Mode

2. Examples of Complementary PWM Mode Operation: Figure 13.31 shows an exampl
complementary PWM mode operation. In complementary PWM mode, TCNT_0 and
perform an increment or decrement operation. When TCNT_0 and GRA_0 are comp
their contents match, the counter is decremented, and when TCNT_1 underflows, the
is incremented. In GRA_0, GRA_1, and GRB_1, compare match is carried out in the
TCNT_0 → TCNT_1 → TCNT_1 → TCNT_0 and PWM waveform is output, durin
cycle of a up/down counter. In this mode, the initial setting will be TCNT_0 > TCNT

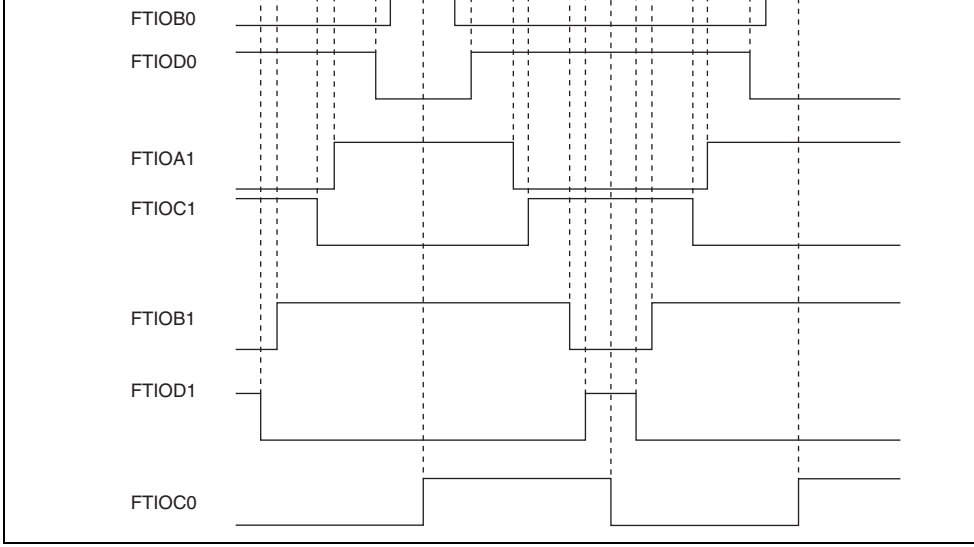


Figure 13.31 Example of Complementary PWM Mode Operation (1)

cycle waveform output, see 3. C., Outputting a waveform with a duty cycle of 0% an
section 13.4.7.

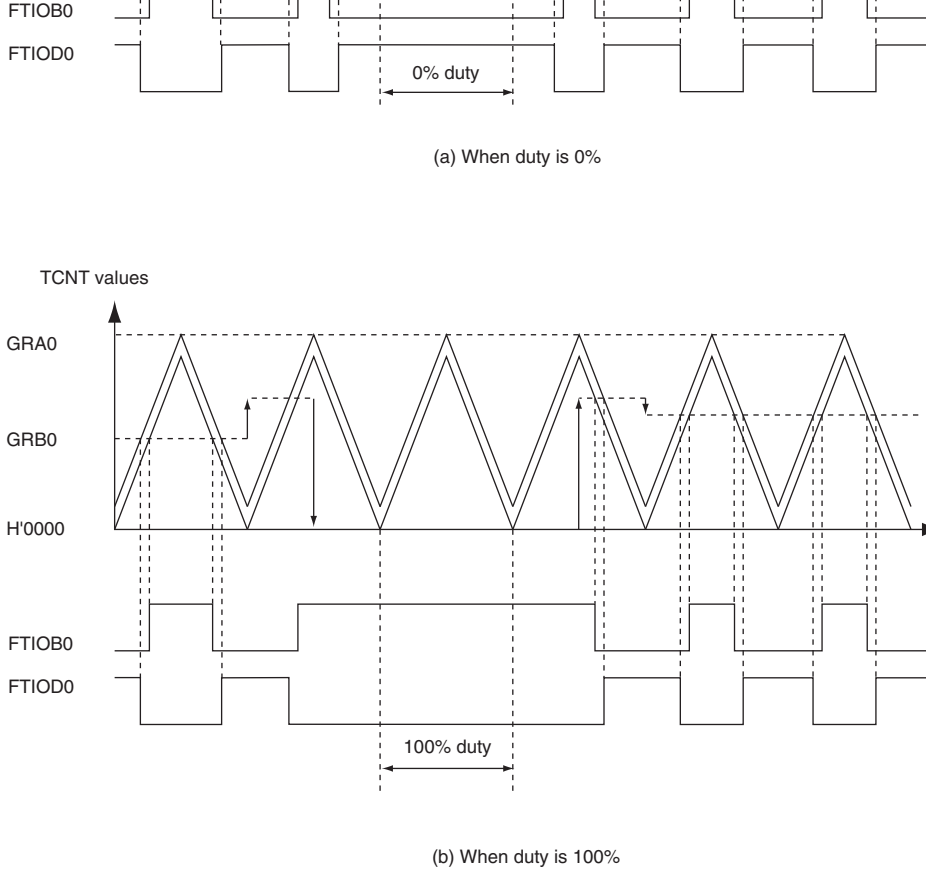
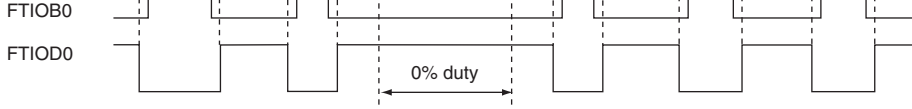
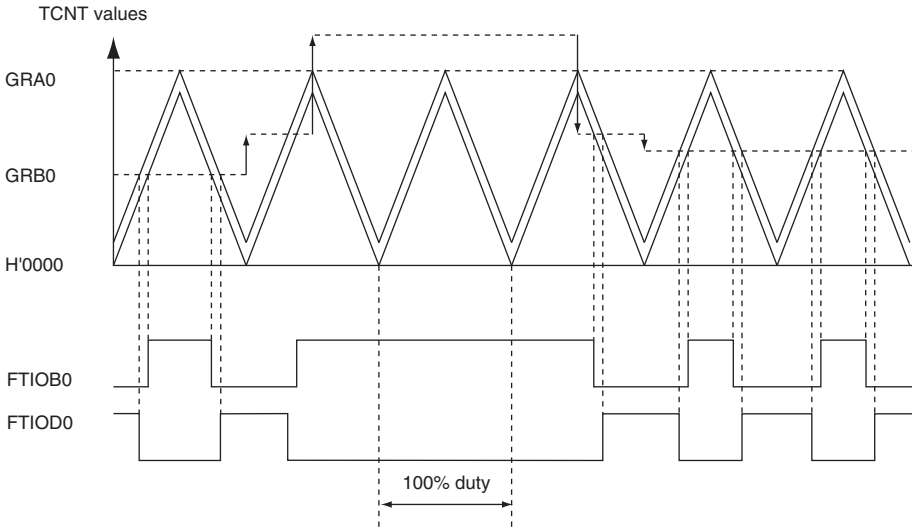


Figure 13.32 (1) Example of Complementary PWM Mode Operation (TPSC2 = TPSC1 = TPSC0 = 0) (2)



(a) When duty is 0%



(b) When duty is 100%

Figure 13.32 (2) Example of Complementary PWM Mode Operation (TPSC2 = TPSC1 = TPSC0 ≠ 0) (3)

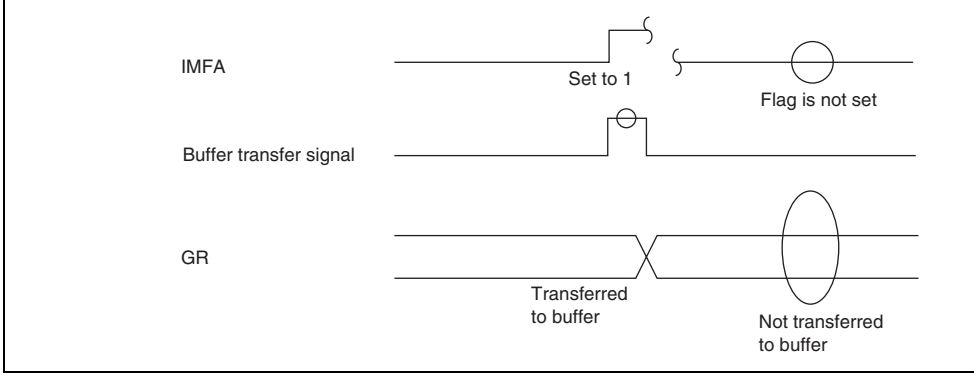


Figure 13.33 Timing of Overshooting

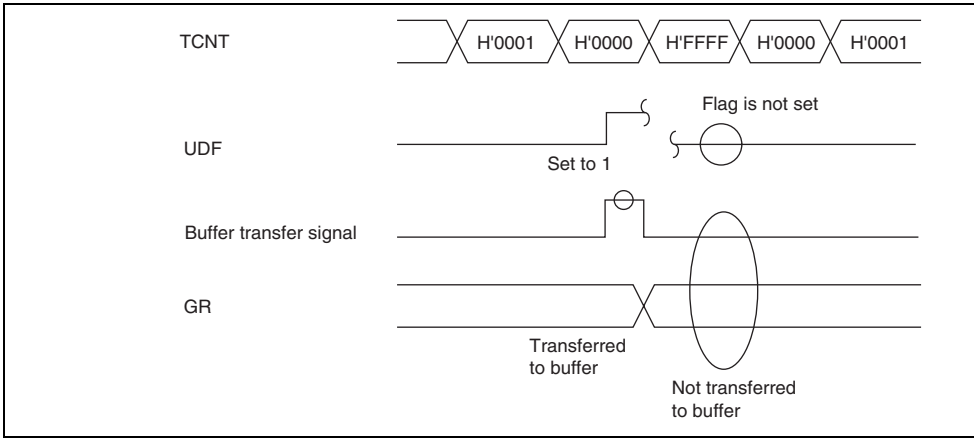


Figure 13.34 Timing of Undershooting

H'FFFC or less. When TPSC2 = TPSC1 = TPSC0 = 0, the GRA_0 value can
H'FFFF or less.

b. H'0000 to T – 1 (T: Initial value of TCNT0) must not be set for the initial val

c. GRA_0 – (T – 1) or more must not be set for the initial value.

d. When using buffer operation, the same values must be set in the buffer regist
corresponding general registers.

B. Modifying the setting value

a. Writing to GR directly must be performed while the TCNT_1 and TCNT_0 v
should satisfy the following expression: H'0000 ≤ TCNT_1 < previous GR va
previous GR value < TCNT_0 ≤ GRA_0. Otherwise, a waveform is not outpu
correctly. For details on outputting a waveform with a duty cycle of 0% and 1
C., Outputting a waveform with a duty cycle of 0% and 100%.

b. Do not write the following values to GR directly. When writing the values, a
is not output correctly.

H'0000 ≤ GR ≤ T – 1 and GRA_0 – (T – 1) ≤ GR < GRA_0 when TPSC2 =
TPSC0 = 0

H'0000 < GR ≤ T – 1 and GRA_0 – (T – 1) ≤ GR < GRA_0 + 1 when TPSC
= TPSC0 = 0

c. Do not change settings of GRA_0 during operation.

C. Outputting a waveform with a duty cycle of 0% and 100%

a. Buffer operation is not used and TPSC2 = TPSC1 = TPSC0 = 0

Write H'0000 or a value equal to or more than the GRA_0 value to GR direct
timing shown below.

- To output a 0%-duty cycle waveform, write a value equal to or more than the
value while H'0000 ≤ TCNT_1 < previous GR value
- To output a 100%-duty cycle waveform, write H'0000 while previous GR val
TCNT_0 ≤ GRA_0

- To output a 0%-duty cycle waveform, write a value equal to or more than the C value to the buffer register
- To output a 100%-duty cycle waveform, write H'0000 to the buffer register

For details on buffer operation, see section 13.4.8, Buffer Operation.

c. Buffer operation is not used and other than TPSC2 = TPSC1 = TPSC0 = 0

Write a value which satisfies GRA_0 + 1 < GR < H'FFFF to GR directly at the shown below.

- To output a 0%-duty cycle waveform, write the value while H'0000 ≤ TCNT_1 previous GR value
- To output a 100%-duty cycle waveform, write the value while previous GR va TCNT_0 ≤ GRA_0

To change duty cycles while a waveform with a duty cycle of 0% and 100% is output, the following procedure must be followed.

- To change duty cycles while a 0%-duty cycle waveform is being output, write while H'0000 ≤ TCNT_1 < previous GR value
- To change duty cycles while a 100%-duty cycle waveform is being output, wri while previous GR value< TCNT_0 ≤ GRA_0

Note that changing from a 0%-duty cycle waveform to a 100%-duty cycle wav and vice versa is not possible.

d. Buffer operation is used and other than TPSC2 = TPSC1 = TPSC0 = 0

Write a value which satisfies GRA_0 + 1 < GR < H'FFFF to the buffer register waveform with a duty cycle of 0% can be output. However, a waveform with a cycle of 100% cannot be output using the buffer operation. Also, the buffer op cannot be used to change duty cycles while a waveform with a duty cycle of 1 being output. For details on buffer operation, see section 13.4.8, Buffer Operat

GRA	GRC
GRB	GRD

1. When GR is an output compare register

   When a compare match occurs, the value in the buffer register of the corresponding
   transferred to the general register.

   This operation is illustrated in figure 13.35.

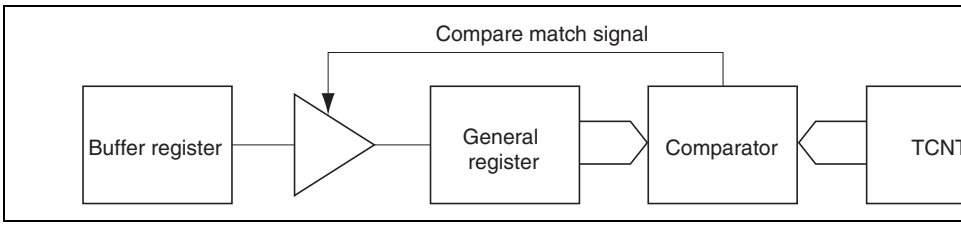


Figure 13.35 Compare Match Buffer Operation

2. When GR is an input capture register

   When an input capture occurs, the value in TCNT is transferred to the general regist
   value previously stored in the general register is transferred to the buffer register.

   This operation is illustrated in figure 13.36.

buffer register is transferred to the general register. Here, the value of the buffer regis
transferred to the general register in the following timing:

A. When TCNT_0 and GRA_0 are compared and their contents match

B. When TCNT_1 underflows

4. Reset Synchronous PWM Mode

   The value of the buffer register is transferred from compare match A0 to the general r

5. Example of Buffer Operation Setting Procedure

   Figure 13.37 shows an example of the buffer operation setting procedure.

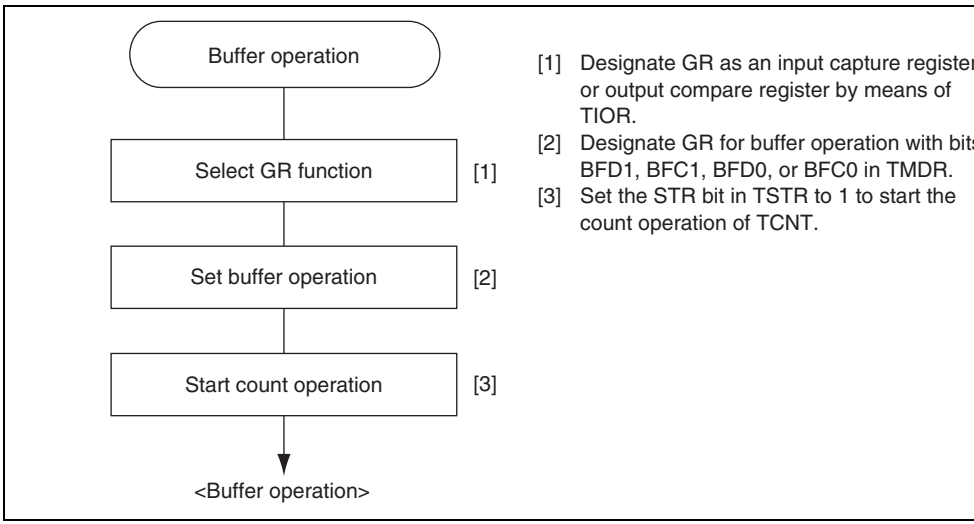


Figure 13.37 Example of Buffer Operation Setting Procedure

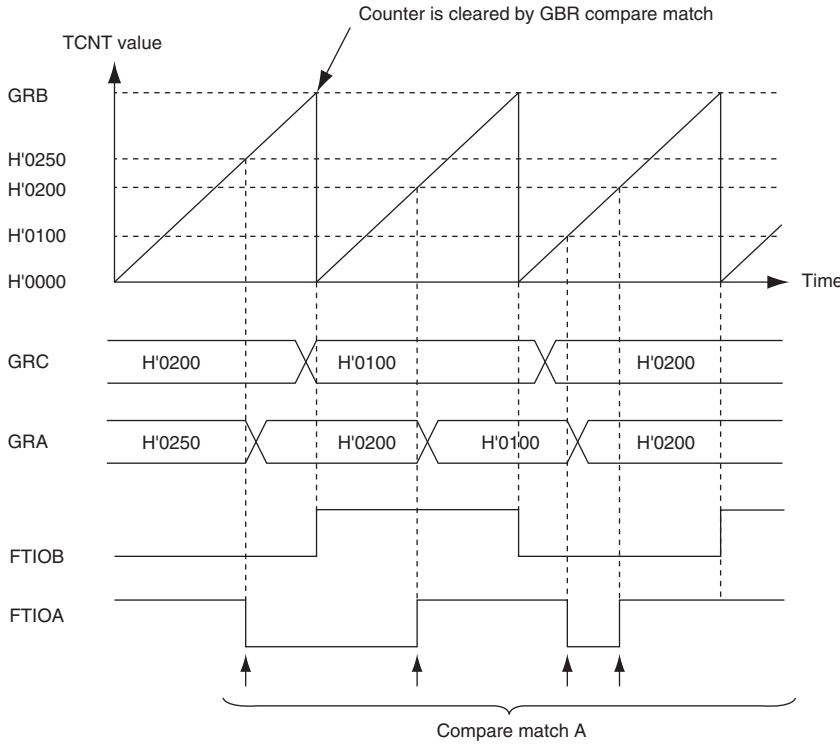


Figure 13.38 Example of Buffer Operation (1)
(Buffer Operation for Output Compare Register)

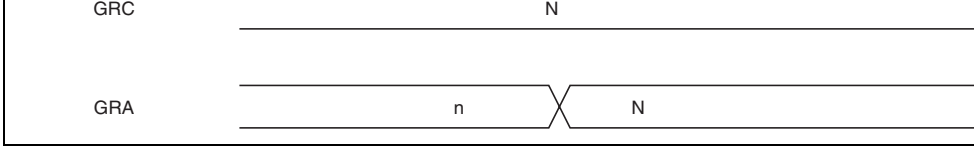


Figure 13.39 Example of Compare Match Timing for Buffer Operation

Figure 13.40 shows an operation example in which GRA has been designated as an input
register, and buffer operation has been designated for GRA and GRC.

Counter clearing by input capture B has been set for TCNT, and falling edges have been s
as the FIOCB pin input capture input edge. And both rising and falling edges have been s
as the FIOCA pin input capture input edge.

As buffer operation has been set, when the TCNT value is stored in GRA upon the occurr
input capture A, the value previously stored in GRA is simultaneously transferred to GRC
transfer timing is shown in figure 13.41.

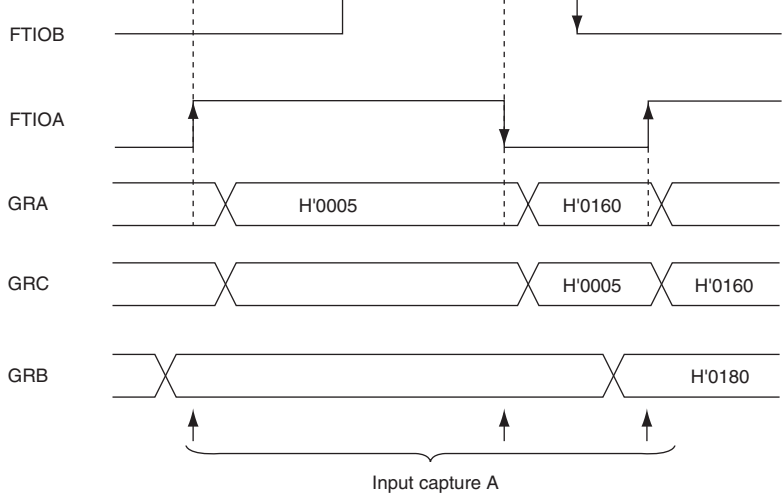


Figure 13.40 Example of Buffer Operation (2)
(Buffer Operation for Input Capture Register)

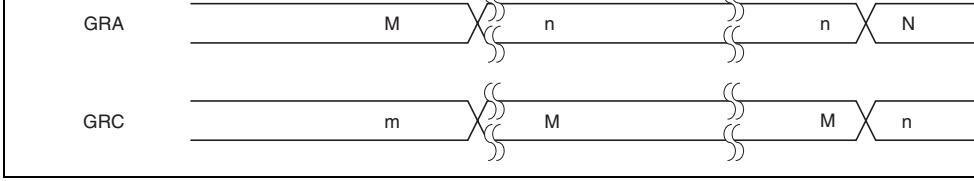


Figure 13.41 Input Capture Timing of Buffer Operation

Figures 13.42 and 13.43 show the operation examples when buffer operation has been de for GRB_0 and GRD_0 in complementary PWM mode. These are examples when a PWM waveform of 0% duty is created by using the buffer operation and performing GRD_0 ≥ Data is transferred from GRD_0 to GRB_0 according to the settings of CMD_0 and CMD TCNT_0 and GRA_0 are compared and their contents match or when TCNT_1 underflow However, when GRD_0 ≥ GRA_0, data is transferred from GRD_0 to GRB_0 when TCN underflows regardless of the setting of CMD_0 and CMD_1. When GRD_0 = H'0000, da transferred from GRD_0 to GRB_0 when TCNT_0 and GRA_0 are compared and their c match regardless of the settings of CMD_0 and CMD_1.

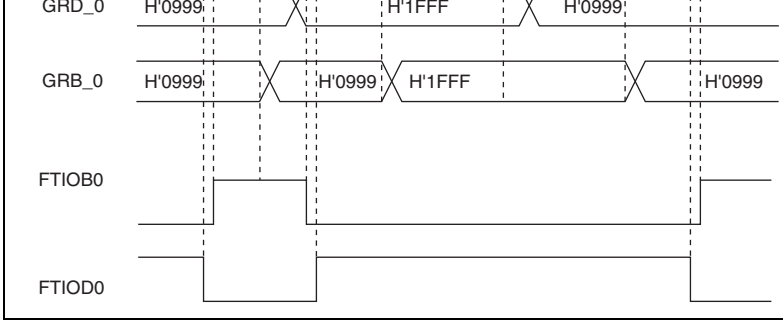


Figure 13.42 Buffer Operation (3)
(Buffer Operation in Complementary PWM Mode CMD1 = CMD0 = 1)

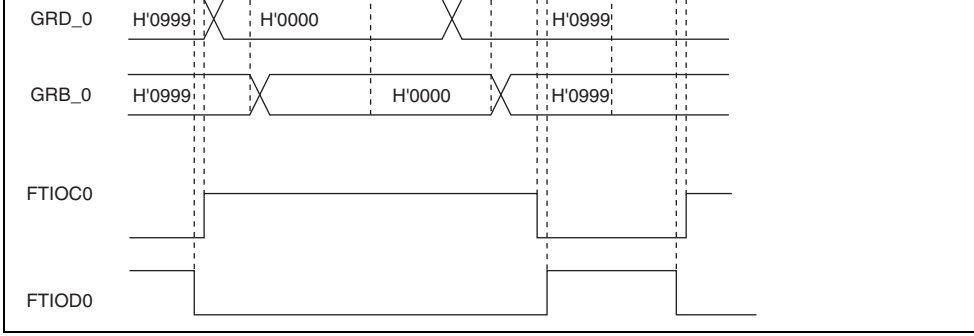


Figure 13.43 Buffer Operation (4)
(Buffer Operation in Complementary PWM Mode CMD1 = CMD0 = 1)

13.4.9 Timer Z Output Timing

The outputs of channels 0 and 1 can be disabled or inverted by the settings of TOER and and the external level.

1. Output Disable/Enable Timing of Timer Z by TOER: Setting the master enable bit in 1 disables the output of timer Z. By setting the PCR and PDR of the corresponding I/O beforehand, any value can be output. Figure 13.44 shows the timing to enable or disab output of timer Z by TOER.

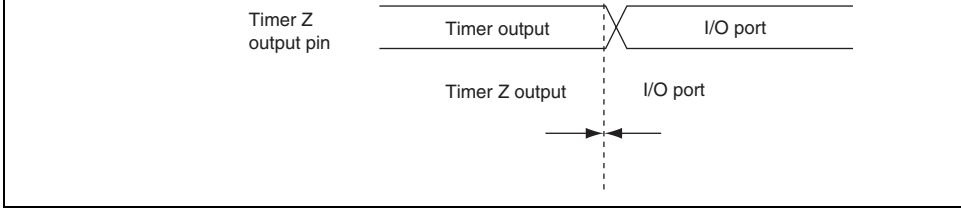


Figure 13.44 Example of Output Disable Timing of Timer Z by Writing to T

2. Output Disable Timing of Timer Z by External Trigger: When P54/$\overline{\text{WKP4}}$ is set as input pin, and low level is input to $\overline{\text{WKP4}}$, the master enable bit in TOER is set to 1 output of timer Z will be disabled.

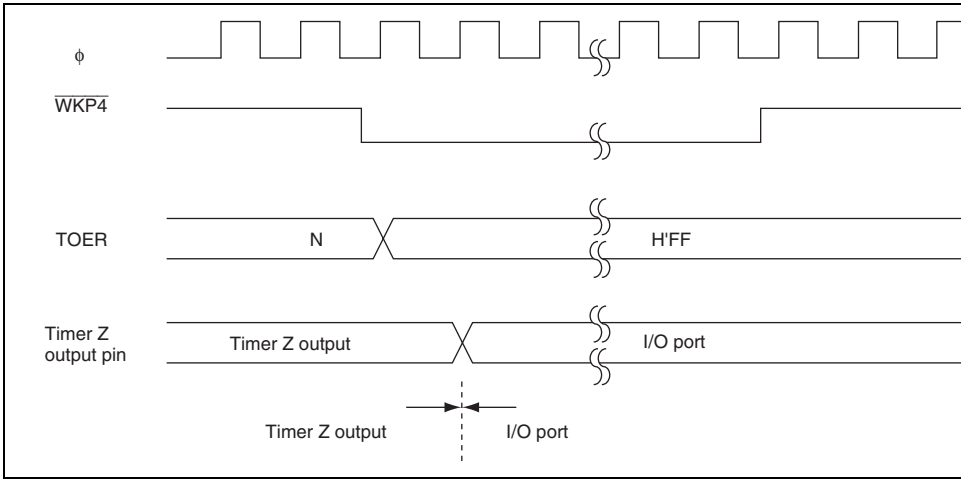


Figure 13.45 Example of Output Disable Timing of Timer Z by External Tri

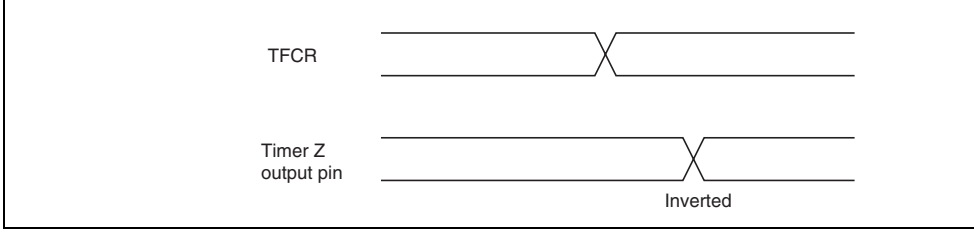


Figure 13.46 Example of Output Inverse Timing of Timer Z by Writing to TF

4. Output Inverse Timing by POCR: The output level can be inverted by inverting the P POLC, and POLB bits in POCR in PWM mode. Figure 13.47 shows the timing.

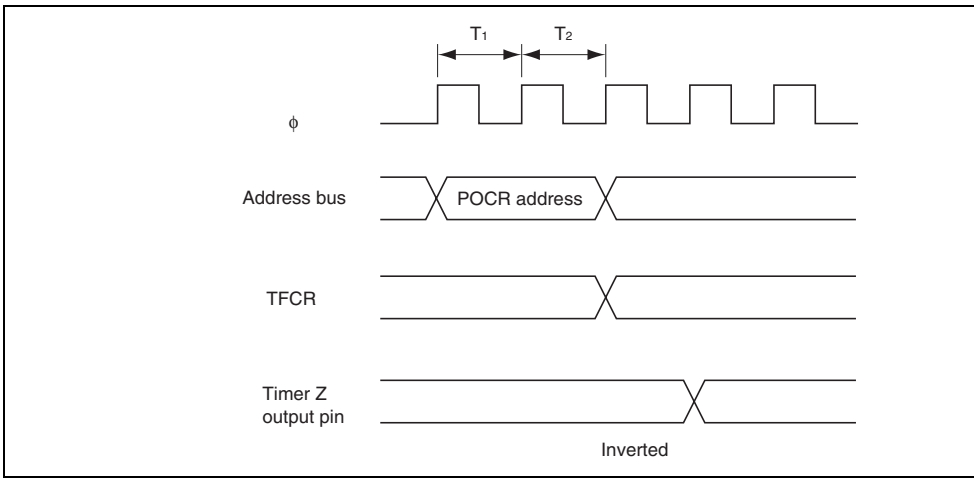


Figure 13.47 Example of Output Inverse Timing of Timer Z by Writing to PO

when the TCNT and GR matches, the compare match signal will not be generated un
TCNT input clock is generated. Figure 13.48 shows the timing to set the IMF flag.

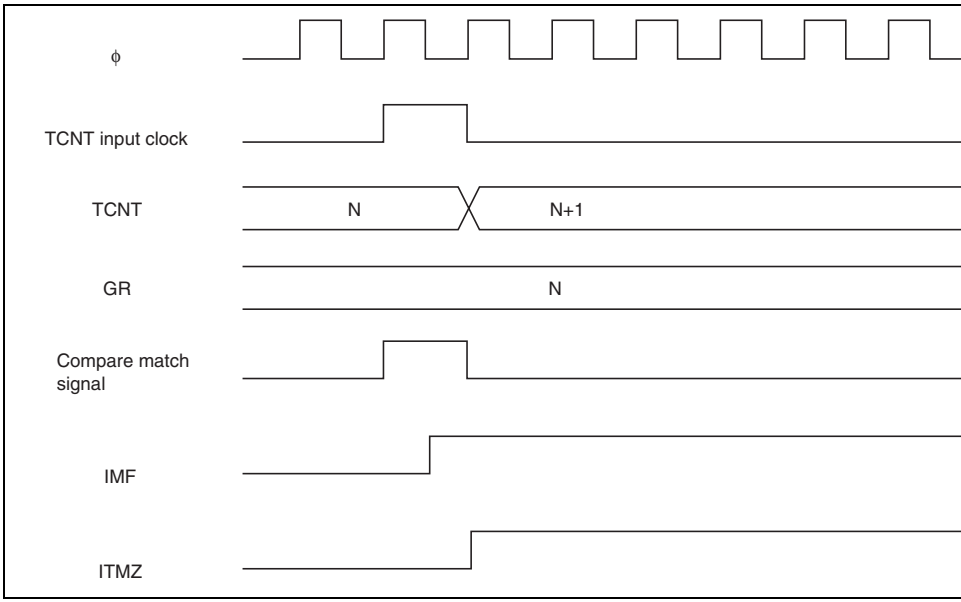


Figure 13.48 IMF Flag Set Timing when Compare Match Occurs

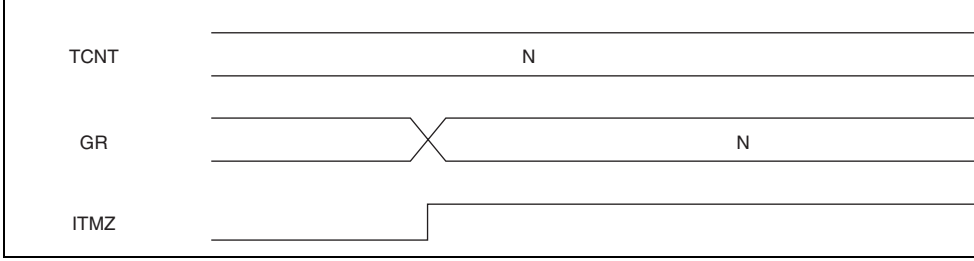


Figure 13.49 IMF Flag Set Timing at Input Capture

3. Overflow Flag (OVF) Set Timing: The overflow flag is set to 1 when the TCNT overf
Figure 13.50 shows the timing.

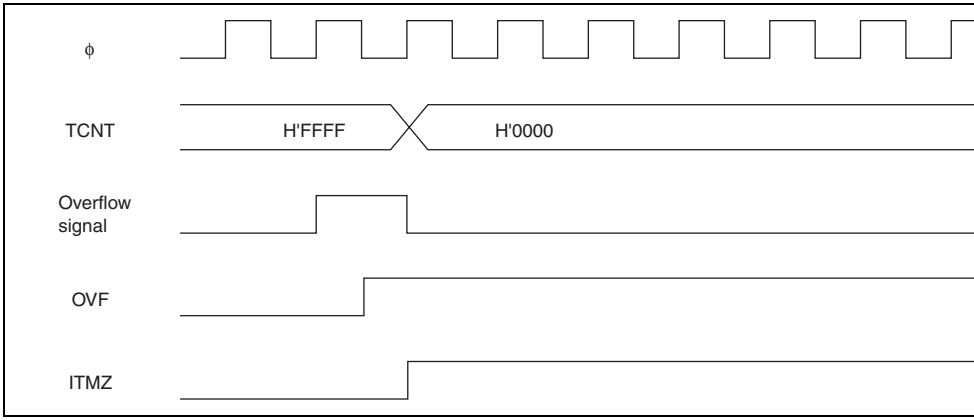


Figure 13.50 OVF Flag Set Timing

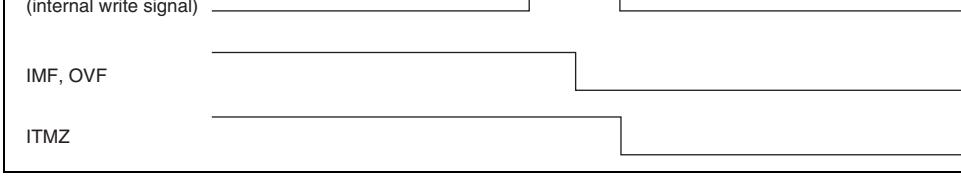


Figure 13.51 Status Flag Clearing Timing

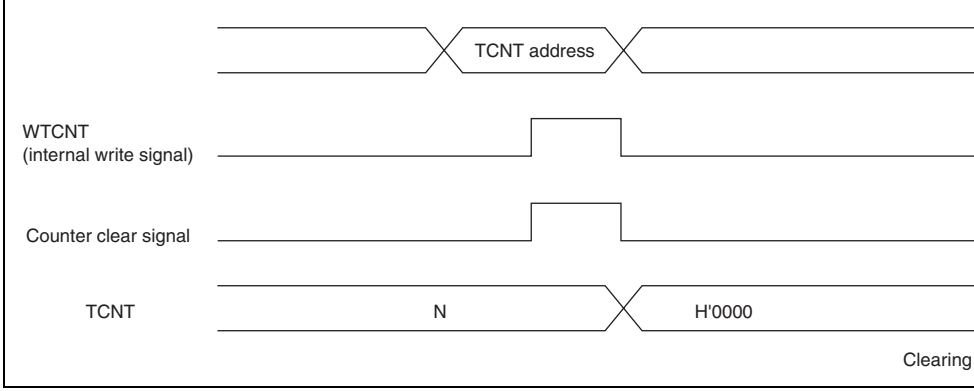


Figure 13.52 Contention between TCNT Write and Clear Operations

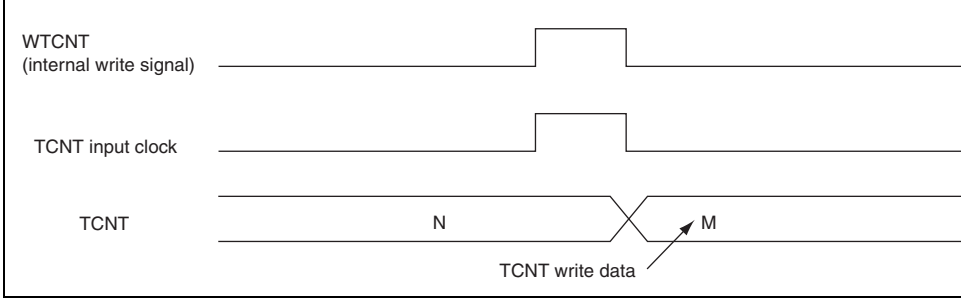


Figure 13.53 Contention between TCNT Write and Increment Operation

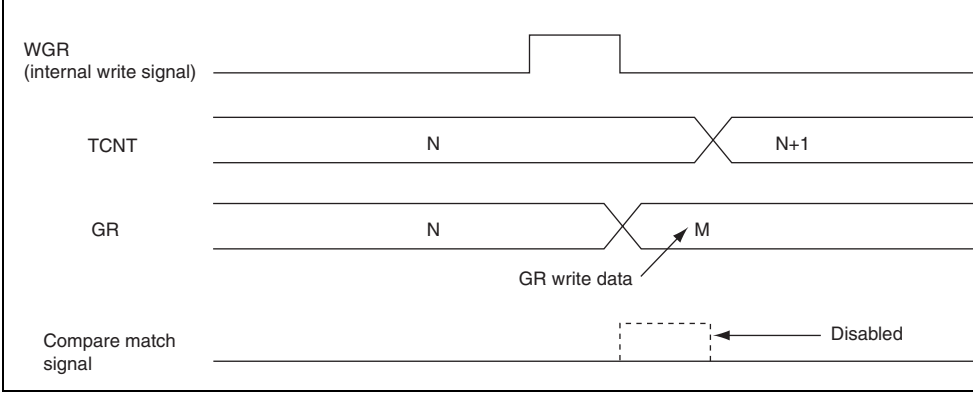


Figure 13.54 Contention between GR Write and Compare Match

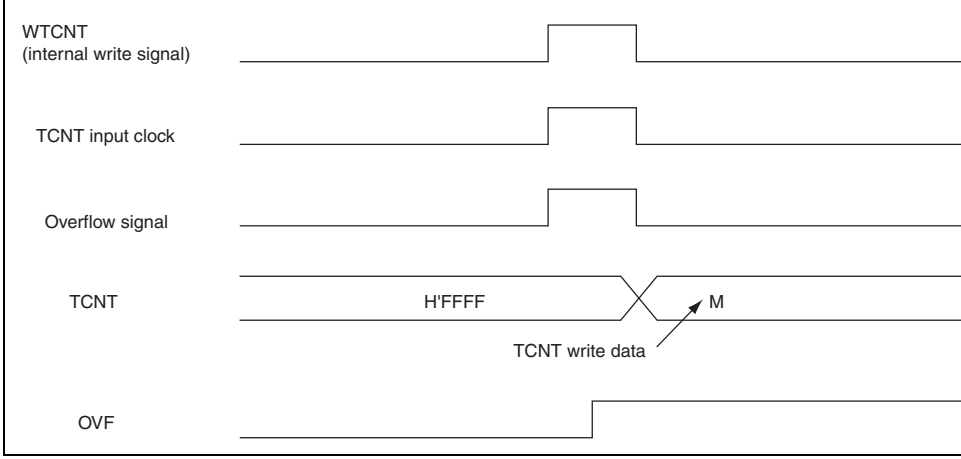


Figure 13.55 Contention between TCNT Write and Overflow

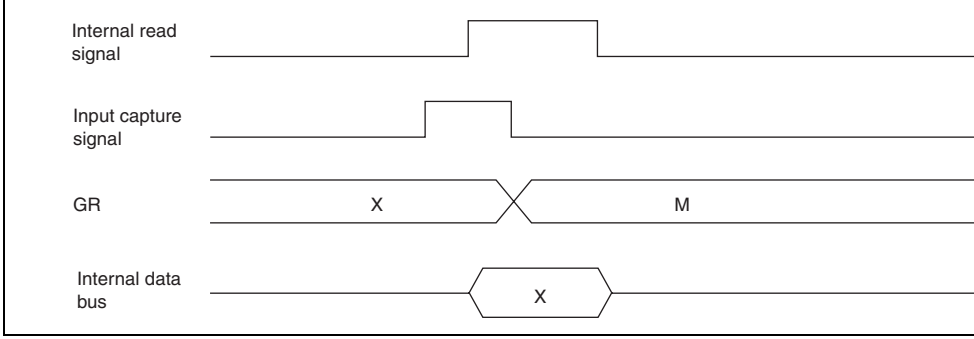


Figure 13.56 Contention between GR Read and Input Capture

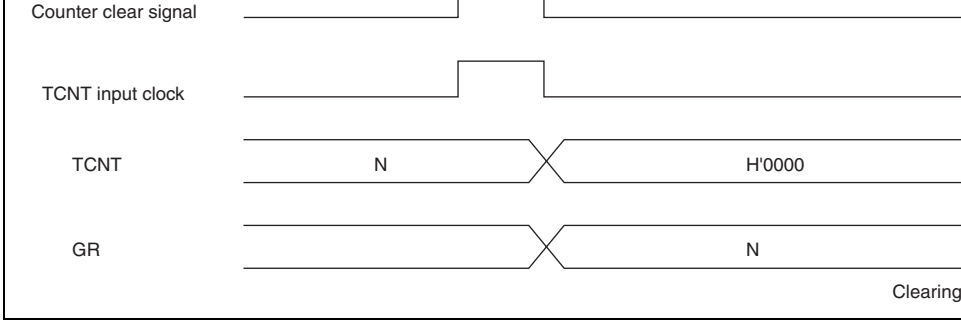


Figure 13.57 Contention between Count Clearing and Increment Operatio by Input Capture

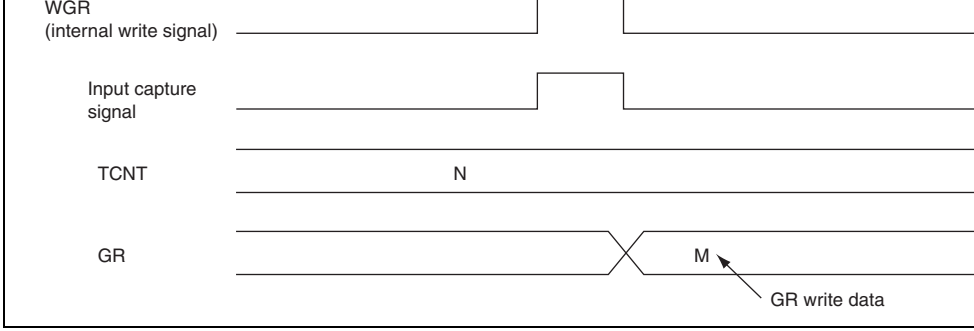


Figure 13.58 Contention between GR Write and Input Capture

8. Notes on Setting Reset Synchronous PWM Mode/Complementary PWM Mode: Whe
   CMD1 and CMD0 in TFCR are set, note the following:
   A. Write bits CMD1 and CMD0 while TCNT_1 and TCNT_0 are halted.
   B. Changing the settings of reset synchronous PWM mode to complementary PWM
      vice versa is disabled. Set reset synchronous PWM mode or complementary PWM
      after the normal operation (bits CMD1 and CMD0 are cleared to 0) has been set.

9. Note on Clearing TSR Flag: When a specific flag in TSR is cleared, a combination o
   BCLR or MOV instructions is used to read 1 from the flag and then write 0 to the fla
   However, if another bit is set during this processing, the bit may also be cleared
   simultaneously. To avoid this, the following processing that does not use the BCLR
   instruction must be executed. Note that this note is only applied to the F-ZTAT versi
   problem has already been solved in the mask ROM version.

   Example: When clearing bit 4 (OVF) in TSR
   MOV.B @TSR,R0L

same timing, the writing to TOCR has the priority. Thus, output change due to the co
match is not reflected to the FTIOA0 to FTIOD0 and FTIOA1 to FTIOD1 pins. Ther
when bit manipulation instruction is used to write to TOCR, the values of the FTIOA
FTIOD0 and FTIOA1 to FTIOD1 pin output may result in an unexpected result. Wh
is to be written to while compare match is operating, stop the counter once before ac
TOCR, read the port 6 state to reflect the values of FTIOA0 to FTIOD0 and FTIOA1
FTIOD1 output, to TOA0 to TOD0 and TOA1 to TOD1, and then restart the counter
13.59 shows an example when the compare match and the bit manipulation instructio
TOCR occur at the same timing.

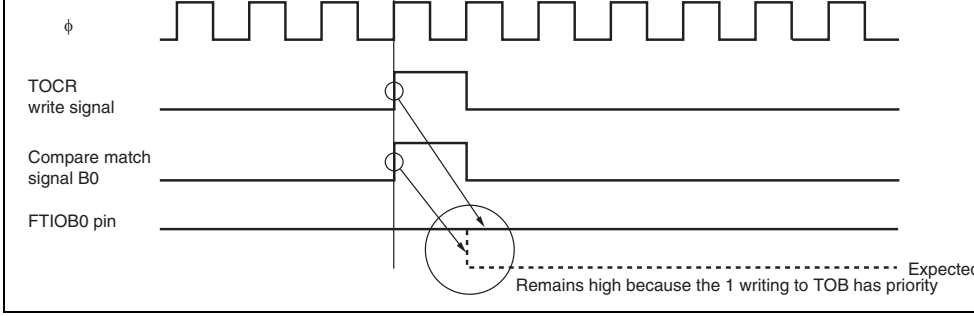


Figure 13.59 When Compare Match and Bit Manipulation Instruction to TO Occur at the Same Timing

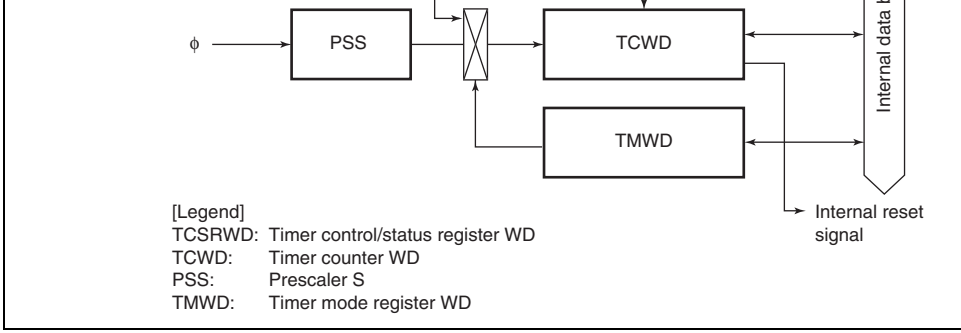


Figure 14.1 Block Diagram of Watchdog Timer

14.1 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) internal oscillator can be selected as the timer-counter clock. When the internal oscil 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.

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

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 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 When writing data to this bit, the value for bit 7 m
5	B4WI	1	R/W	Bit 4 Write Inhibit The TCSRWE bit can be written only when the w value of the B4WI bit is 0. This bit is always read
4	TCSRWE	0	R/W	Timer Control/Status Register WD Write Enable The WDON and WRST bits can be written when TCSRWE bit is set to 1. When writing data to this bit, the value for bit 5 m
3	B2WI	1	R/W	Bit 2 Write Inhibit This bit can be written to the WDON bit only whe write value of the B2WI bit is 0. This bit is always read as 1.

				• When 0 is written to the WDON bit while wri 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 wh write value of the B0WI bit is 0. This bit is alway 1.
0	WRST	0	R/W	Watchdog Timer Reset [Setting condition] When TCWD overflows and an internal reset si generated [Clearing condition] • Reset by $\overline{\text{RES}}$ pin • When 0 is written to the WRST bit while wri the B0WI bit when the TCSRWE bit=1

14.2.2 Timer Counter WD (TCWD)

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

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 se Electrical Characteristics.

[Legend]

X: Don't care.

Figure 14.2 shows an example of watchdog timer operation.

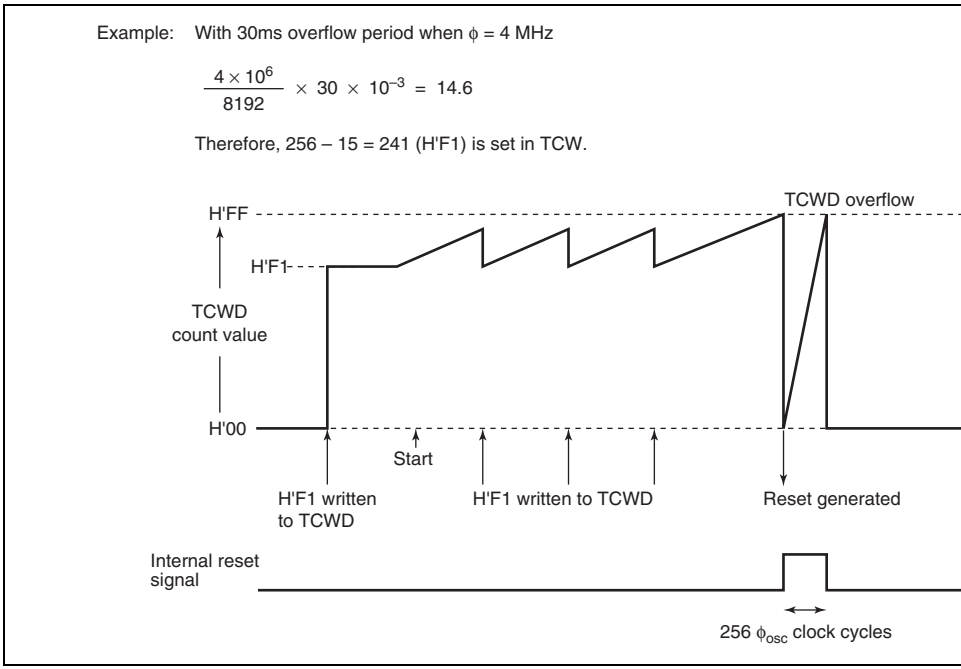


Figure 14.2 Watchdog Timer Operation Example

- Pulse division method for less ripple

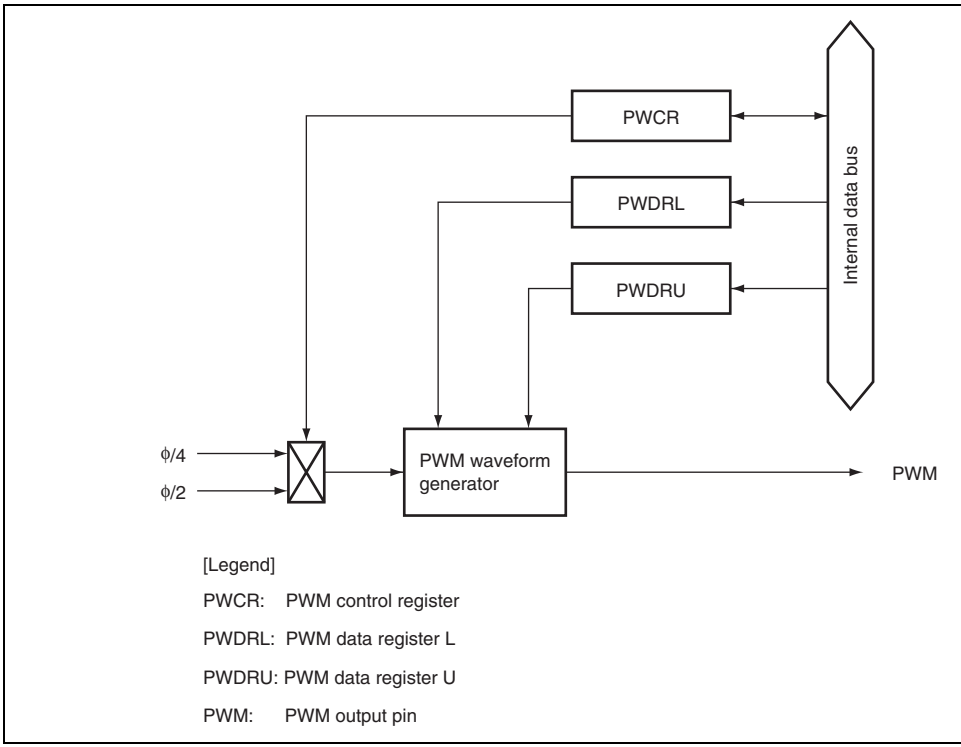


Figure 15.1 Block Diagram of 14-Bit PWM

The 14-bit PWM has the following registers.

- PWM control register (PWCR)
- PWM data register U (PWDRU)
- PWM data register L (PWDRL)

15.3.1 PWM Control Register (PWCR)

PWCR selects the conversion period.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 1	—	Reserved These bits are always read as 1, and cannot be
0	PWCR0	0	R/W	Clock Select 0: The input clock is $\phi/2$ ($t\phi = 2/\phi$) — The conversion period is $16384/\phi$, with a modulation width of $1/\phi$ 1: The input clock is $\phi/4$ ($t\phi = 4/\phi$) — The conversion period is $32768/\phi$, with a modulation width of $2/\phi$

[Legend]
$t\phi$: Period of PWM clock input

PWDRU and PWDRL are initialized to H'C000.

15.4 Operation

When using the 14-bit PWM, set the registers in this sequence:

1. Set the PWM bit in the port mode register 1 (PMR1) to set the P11/PWM pin to func PWM output pin.
2. Set the PWCR0 bit in PWCR to select a conversion period of either.
3. Set the output waveform data in PWDRU and PWDRL. Be sure to write byte data f PWDRL and then to PWDRU. When the data is written in PWDRU, the contents of registers are latched in the PWM waveform generator, and the PWM waveform gene data is updated in synchronization in internal signals.

One conversion period consists of 64 pulses, as shown in figure 15.2. The total high-lev during this period (T_H) corresponds to the data in PWDRU and PWDRL. This relation c expressed as follows:

$$T_H = (\text{data value in PWDRU and PWDRL} + 64) \times t\phi/2$$

where $t\phi$ is the period of PWM clock input: $2/\phi$ (bit PWCR0 = 0) or $4/\phi$ (bit PWCR0 = 1 If the data value in PWDRU and PWDRL is from H'FFC0 to H'FFFF, the PWM output When the data value is H'C000, T_H is calculated as follows:

$$T_H = 64 \times t\phi/2 = 32\ t\phi$$

t f1 = t f2 = t f3 = ··· = t f64

Figure 15.2 Waveform Output by 14-Bit PWM

SCI3. Since pin functions are identical for each of the two channels (SCI3 and SCI3_2), explanations are not given in this section.

16.1 Features

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

  The transmitter and receiver are mutually independent, enabling transmission and re be executed simultaneously.

  Double-buffering is used in both the transmitter and the receiver, enabling continuou 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 sou
- Six interrupt sources

  Transmit-end, transmit-data-empty, receive-data-full, overrun error, framing error, a 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 c framing error

			TDR	H'FFAB
			SSR	H'FFAC
			RDR	H'FFAD
			RSR	—
			TSR	—
Channel 2	SCI3_2	SCK3_2	SMR_2	H'F740
		RXD_2	BRR_2	H'F741
		TXD_2	SCR3_2	H'F742
			TDR_2	H'F743
			SSR_2	H'F744
			RDR_2	H'F745
			RSR_2	—
			TSR_2	—

Note: * The channel 1 of the SCI3 is used in on-board programming mode by boot mo

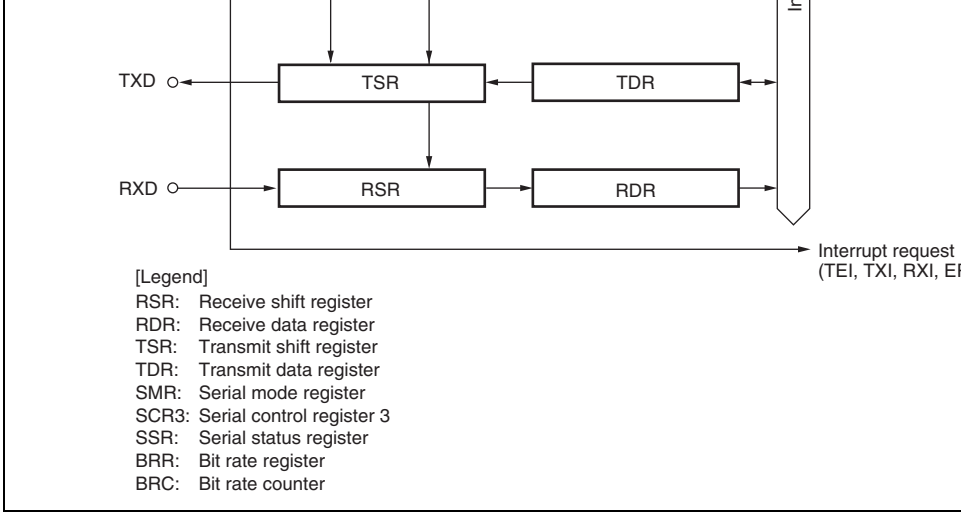


Figure 16.1 Block Diagram of SCI3

16.3 Register Descriptions

The SCI3 has the following registers for each channel.

- 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)

receive-enabled. As RSR and RDR function as a double buffer in this way, continuous r
operations are possible. After confirming that the RDRF bit in SSR is set to 1, read RDR
once. RDR cannot be written to by the CPU. RDR is initialized to H'00.

16.3.3 Transmit Shift Register TSR (SCI3)

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

16.3.4 Transmit Data Register (TDR)

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

6	CHR	0	R/W	Character Length (enabled only in asynchronous 0: Selects 8 bits as the data length. 1: Selects 7 bits as the data length.
5	PE	0	R/W	Parity Enable (enabled only in asynchronous mo When this bit is set to 1, the parity bit is added to data before transmission, and the parity bit is ch reception.
4	PM	0	R/W	Parity Mode (enabled only when the PE bit is 1 i asynchronous mode) 0: Selects even parity. 1: Selects odd parity.
3	STOP	0	R/W	Stop Bit Length (enabled only in asynchronous m Selects the stop bit length in transmission. 0: 1 stop bit 1: 2 stop bits For reception, only the first stop bit is checked, r of the value in the bit. If the second stop bit is 0, treated as the start bit of the next transmit chara
2	MP	0	R/W	Multiprocessor Mode When this bit is set to 1, the multiprocessor communication function is enabled. The PE bit a bit settings are invalid in multiprocessor mode. In synchronous mode, clear this bit to 0.

and the baud rate, see section 16.3.8, Bit Rate
(BRR). n is the decimal representation of the va
BRR (see section 16.3.8, Bit Rate Register (BR

16.3.6 Serial Control Register 3 (SCR3)

SCR3 is a register that enables or disables SCI3 transfer operations and interrupt request
also used to select the transfer clock source. For details on interrupt requests, refer to sec
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 reque enabled.
6	RIE	0	R/W	Receive Interrupt Enable When this bit is set to 1, RXI and ERI interrupt are enabled.
5	TE	0	R/W	Transmit Enable When this bit s set to 1, transmission is enable
4	RE	0	R/W	Receive Enable When this bit is set to 1, reception is enabled.

2	TEIE	0	R/W	Transmit End Interrupt Enable When this bit is set to 1, TEI interrupt request is
1	CKE1	0	R/W	Clock Enable 0 and 1
0	CKE0	0	R/W	Selects the clock source. • Asynchronous mode 00: On-chip baud rate generator 01: On-chip baud rate generator Outputs a clock of the same frequency as th from the SCK3 pin. 10: External clock Inputs a clock with a frequency 16 times the from the SCK3 pin. 11:Reserved • Clocked synchronous mode 00: On-chip clock (SCK3 pin functions as clock o 01: Reserved 10: External clock (SCK3 pin functions as clock i 11: Reserved

				• 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 TD • 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 RD [Setting condition] • When serial reception ends normally and re is transferred from RSR to RDR [Clearing conditions] • When 0 is written to RDRF after reading RD • 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
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

				• When TDRE = 1 at transmission of the last b frame serial transmit character [Clearing conditions] • When 0 is written to TDRE after reading TDF • When the transmit data is written to TDR
1	MPBR	0	R	Multiprocessor Bit Receive MPBR stores the multiprocessor bit in the receiv character data. When the RE bit in SCR3 is clea its state is retained.
0	MPBT	0	R/W	Multiprocessor Bit Transfer MPBT stores the multiprocessor bit to be added transmit character data.

[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: CSK1 and CSK0 settings in SMR ($0 \leq n \leq 3$)

1200	0	103	0.16	0	127	0.00	0	129
2400	0	51	0.16	0	63	0.00	0	64
4800	0	25	0.16	0	31	0.00	0	32
9600	0	12	0.16	0	15	0.00	0	15
19200	0	6	–6.99	0	7	0.00	0	7
31250	0	3	0.00	0	4	–1.70	0	4
38400	0	2	8.51	0	3	0.00	0	3

[Legend]

—: A setting is available but error occurs

1200	0	155	0.16	0	159	0.00	0	191
2400	0	77	0.16	0	79	0.00	0	95
4800	0	38	0.16	0	39	0.00	0	47
9600	0	19	–2.34	0	19	0.00	0	23
19200	0	9	–2.34	0	9	0.00	0	11
31250	0	5	0.00	0	5	2.40	0	6
38400	0	4	–2.34	0	4	0.00	0	5

2400	0	103	0.16	0	127	0.00	0	129	0.16	0	155
4800	0	51	0.16	0	63	0.00	0	64	0.16	0	77
9600	0	25	0.16	0	31	0.00	0	32	–1.36	0	38
19200	0	12	0.16	0	15	0.00	0	15	1.73	0	19
31250	0	7	0.00	0	9	–1.70	0	9	0.00	0	11
38400	0	6	-6.99	0	7	0.00	0	7	1.73	0	9

[Legend]

—: A setting is available but error occurs.

1200	1	79	0.00	1	90	0.16	1	95	0.00	1	10
2400	0	159	0.00	0	181	0.16	0	191	0.00	0	20
4800	0	79	0.00	0	90	0.16	0	95	0.00	0	10
9600	0	39	0.00	0	45	–0.93	0	47	0.00	0	51
19200	0	19	0.00	0	22	–0.93	0	23	0.00	0	25
31250	0	11	2.40	0	13	0.00	0	14	–1.70	0	15
38400	0	9	0.00	—	—	—	0	11	0.00	0	12

2400	0	233	0.16
4800	0	116	0.16
9600	0	58	–0.96
19200	0	28	1.02
31250	0	17	0.00
38400	0	14	–2.34

[Legend]

—: A setting is available but error occurs.

9.8304	307200	0	0
10	312500	0	0
12	375000	0	0
12.288	384000	0	0
14	437500	0	0
14.7456	460800	0	0
16	500000	0	0
17.2032	537600	0	0
18	562500	0	0

2.5k	1	99	1	199	1	249	2
5k	0	199	1	99	1	124	1
10k	0	99	0	199	0	249	1
25k	0	39	0	79	0	99	0
50k	0	19	0	39	0	49	0
100k	0	9	0	19	0	24	0
250k	0	3	0	7	0	9	0
500k	0	1	0	3	0	4	0
1M	0	0*	0	1	—	—	0
2M			0	0*	—	—	0
2.5M					0	0*	—
4M							0

[Legend]

Blank: No setting is available.

—: A setting is available but error occurs.

*: Continuous transfer is not possible.

2.5k	2	112
5k	1	224
10k	1	112
25k	0	179
50k	0	89
100k	0	44
250k	0	17
500k	0	8
1M	0	4
2M	—	—
2.5M	—	—
4M	—	—

[Legend]

Blank: No setting is available.

—: A setting is available but error occurs.

*: Continuous transfer is not possible.

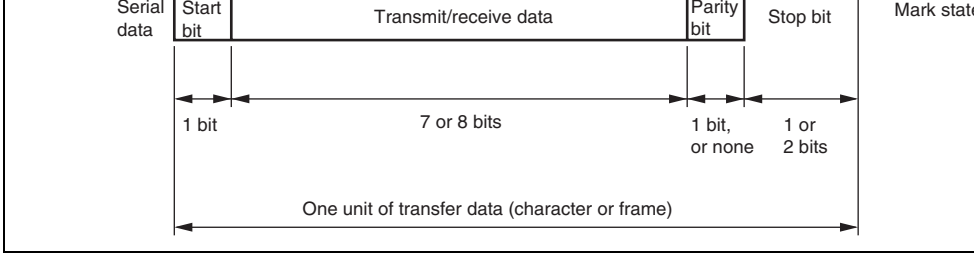


Figure 16.2 Data Format in Asynchronous Communication

16.4.1 Clock

Either an internal clock generated by the on-chip baud rate generator or an external clock
the SCK3 pin can be selected as the SCI3's serial clock, according to the setting of the C
SMR and the CKE0 and CKE1 bits in SCR3. When an external clock is input at the SCK
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 p
frequency of the clock output in this case is equal to the bit rate, and the phase is such tha
rising edge of the clock is in the middle of the transmit data, as shown in figure 16.3.

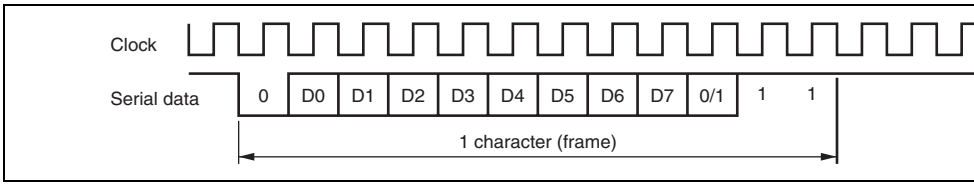
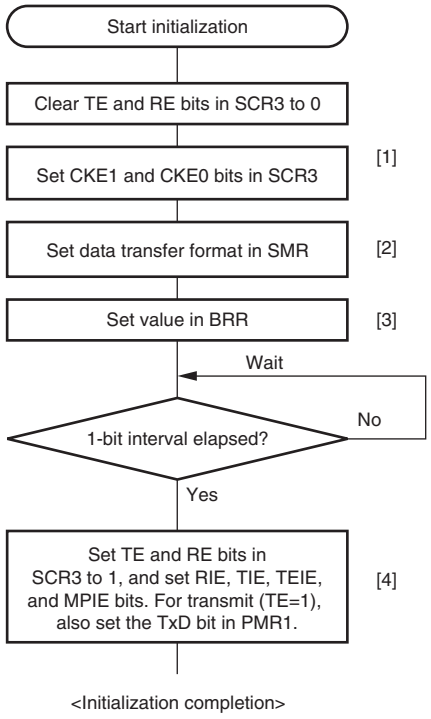


Figure 16.3 Relationship between Output Clock and Transfer Data Phase (Asynchronous Mode)(Example with 8-Bit Data, Parity, Two Stop Bits)



[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.

In asynchronous transmission mode, after the TE bit is set to 1, a frame of 1s is output, and transmission is enabled.

Figure 16.4 Sample SCI3 Initialization Flowchart

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, an 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 t state" is entered, in which 1 is output. If the TEIE bit in SCR3 is set to 1 at this time, a interrupt request is generated.
6. Figure 16.6 shows a sample flowchart for transmission in asynchronous mode.

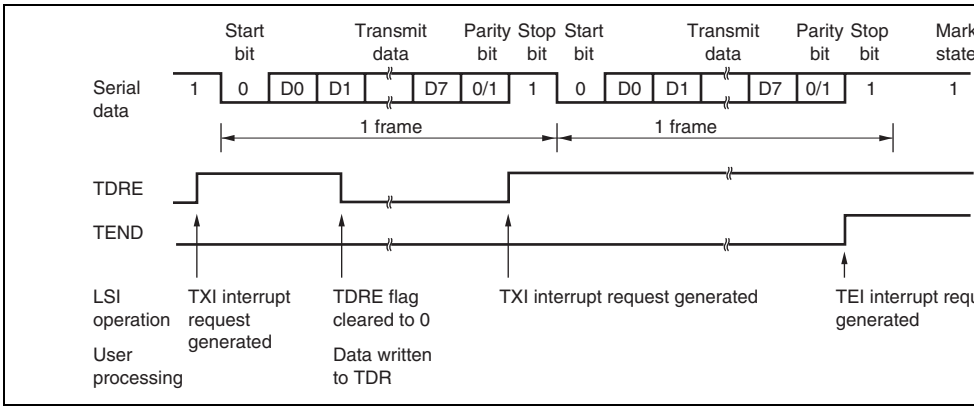
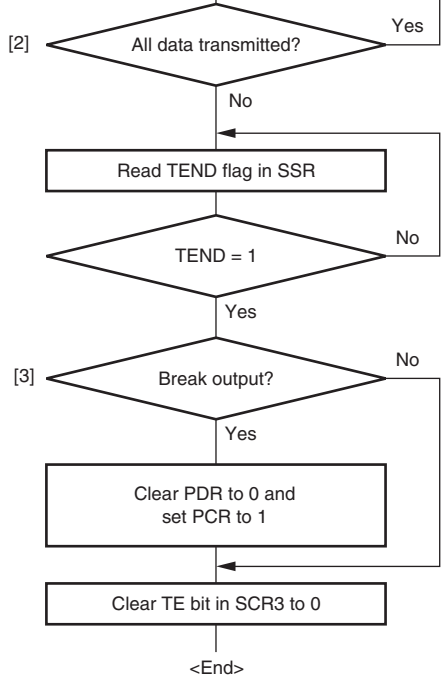


Figure 16.5 Example of SCI3 Transmission in Asynchronous Mode (8-Bit Data, Parity, One Stop Bit)



and PDR to 0, clear TxD in PMR
to 0, then clear the TE bit in SCR
to 0.

Figure 16.6 Sample Serial Transmission Data Flowchart (Asynchronous Mo

3. If a parity error is detected, the PER bit in SSR is set to 1 and receive data is transferr RDR. If the RIE bit in SCR3 is set to 1 at this time, an ERI interrupt request is genera
4. If a framing error is detected (when the stop bit is 0), the FER bit in SSR is set to 1 an data is transferred to RDR. If the RIE bit in SCR3 is set to 1 at this time, an ERI inter request is generated.
5. If reception is completed successfully, the RDRF bit in SSR is set to 1, and receive da transferred to RDR. If the RIE bit in SCR3 is set to 1 at this time, an RXI interrupt re generated. Continuous reception is possible because the RXI interrupt routine reads th data transferred to RDR before reception of the next receive data has been completed.

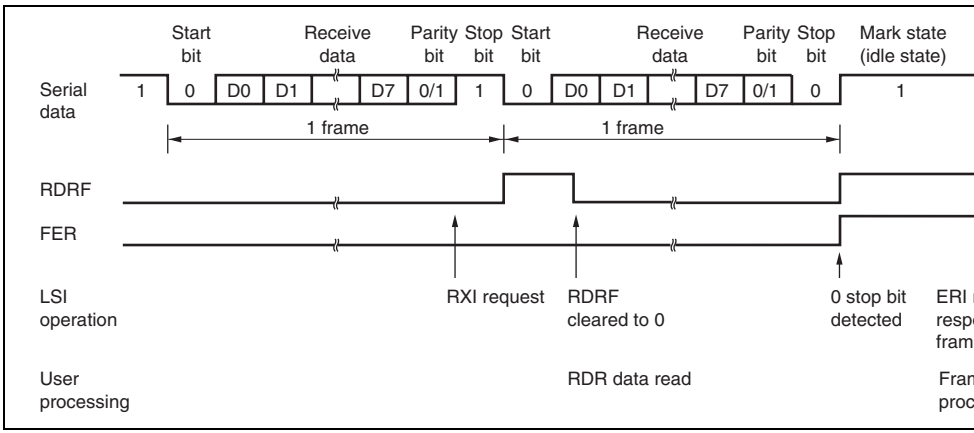
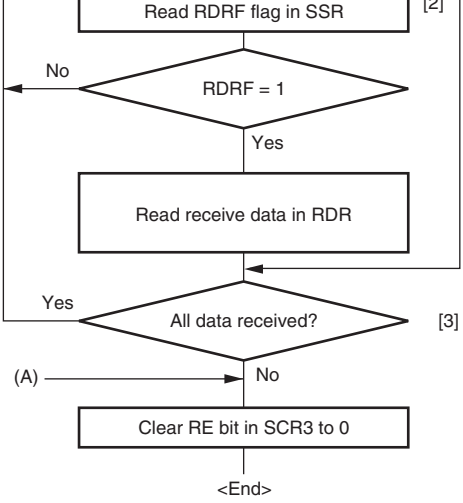


Figure 16.7 Example of SCI3 Reception in Asynchronous Mode (8-Bit Data, Parity, One Stop Bit)

0	0	1	0	Transferred to RDR	Framing error
0	0	0	1	Transferred to RDR	Parity error
1	1	1	0	Lost	Overrun error + fram
1	1	0	1	Lost	Overrun error + parit
0	0	1	1	Transferred to RDR	Framing error + parit
1	1	1	1	Lost	Overrun error + fram parity error

Note: * The RDRF flag retains the state it had before data reception.



the error. After performing the
appropriate error processing, en
that the OER, PER, and FER fla
all cleared to 0. Reception cann
resumed if any of these flags are
1. In the case of a framing error
break can be detected by readin
value of the input port correspon
the RxD pin.

Figure 16.8 Sample Serial Reception Data Flowchart (Asynchronous Mode)

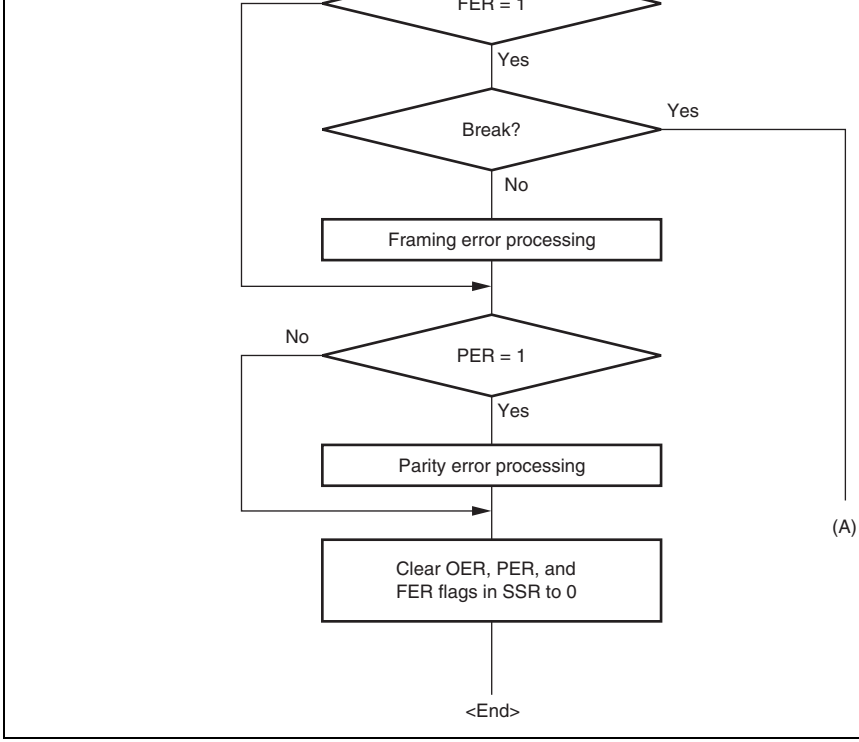


Figure 16.8 Sample Serial Reception Data Flowchart (Asynchronous Mode

duplex communication through the use of a common clock. Both the transmitter and the r
also have a double-buffered structure, so data can be read or written during transmission o
reception, enabling continuous data transfer.

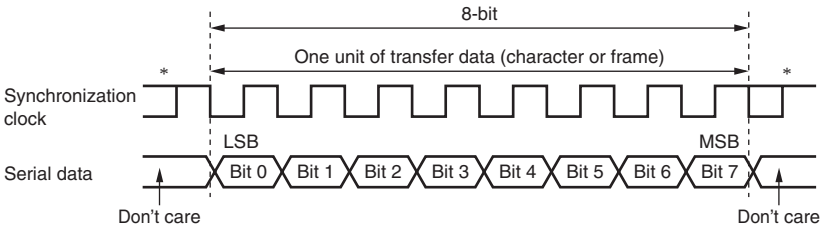


Figure 16.9 Data Format in Clocked Synchronous Communication

16.5.1 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 th bit in SMR and CKE0 and CKE1 bits in SCR3. When the SCI3 is operated on an internal the synchronization clock is output from the SCK3 pin. Eight synchronization clock pulse output in the transfer of one character, and when no transfer is performed the clock is fixe

16.5.2 SCI3 Initialization

Before transmitting and receiving data, the SCI3 should be initialized as described in a sa flowchart in figure 16.4.

mode has been specified, and synchronized with the input clock when use of an exte
has been specified. Serial data is transmitted sequentially from the LSB (bit 0), from
pin.

4. The SCI3 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 tra
of the next frame is started.
6. If the TDRE flag is set to 1, the TEND flag in SSR is set to 1, and the TDRE flag ma
output state of the last bit. If the TEIE bit in SCR3 is set to 1 at this time, a TEI inter
request is generated.
7. The SCK3 pin is fixed high at the end of transmission.

Figure 16.11 shows a sample flow chart for serial data transmission. Even if the TDRE f
cleared to 0, transmission will not start while a receive error flag (OER, FER, or PER) is
Make sure that the receive error flags are cleared to 0 before starting transmission.

Figure 16.10 Example of SCI3 Transmission in Clocked Synchronous Mod

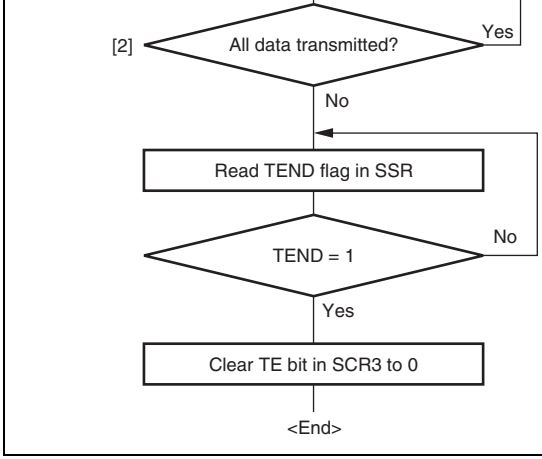


Figure 16.11 Sample Serial Transmission Flowchart (Clocked Synchronous M

time, an ERI interrupt request is generated, receive data is not transferred to RDR, and
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 da
transferred to RDR. If the RIE bit in SCR3 is set to 1 at this time, an RXI interrupt rec
generated.

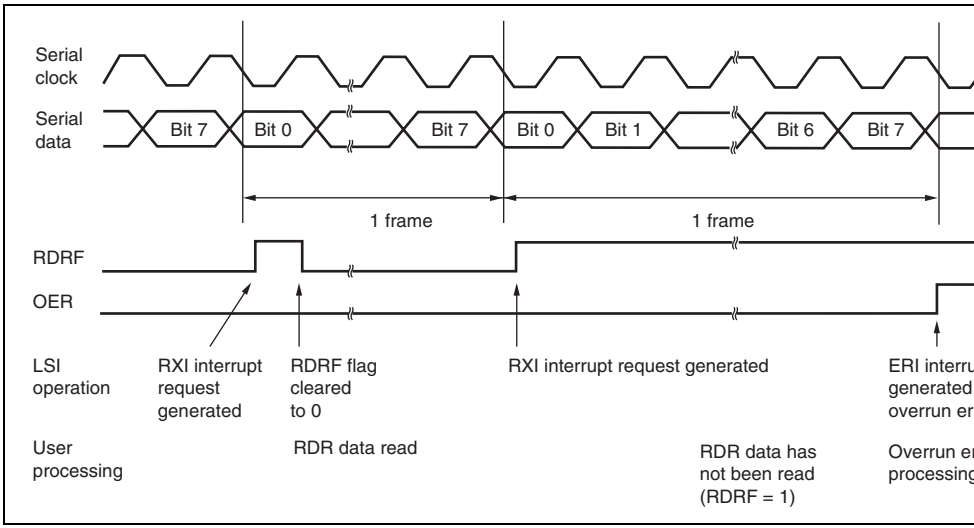
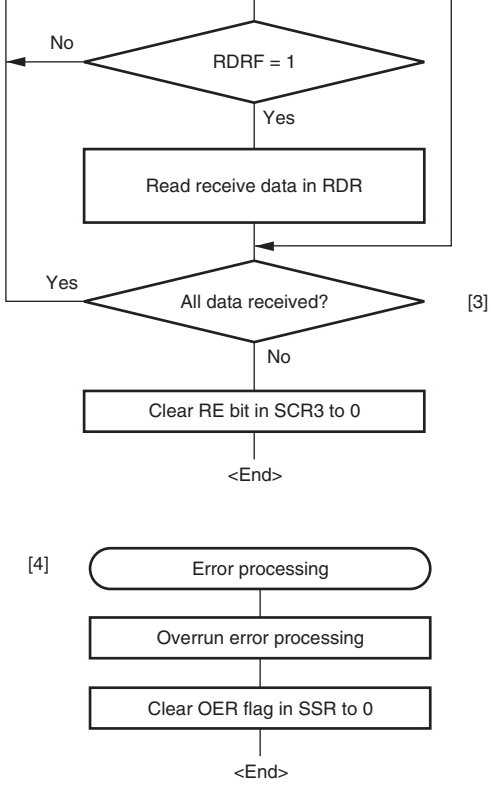


Figure 16.12 Example of SCI3 Reception in Clocked Synchronous Mode

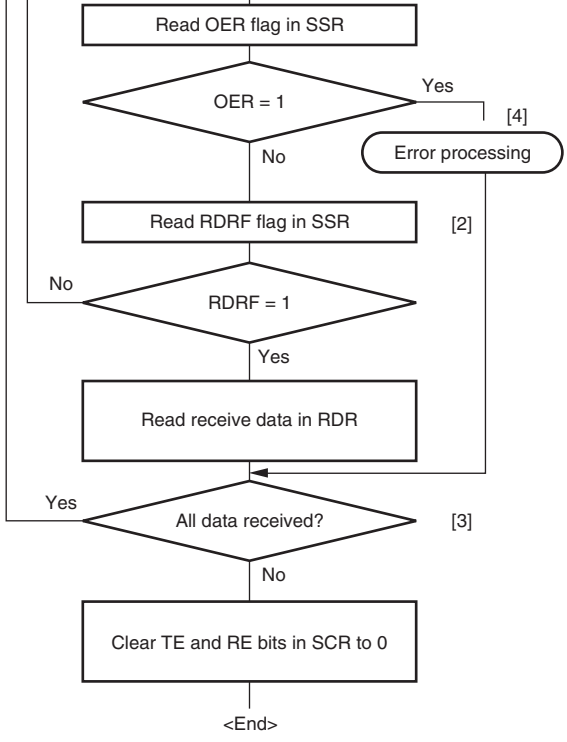
Reception cannot be resumed while a receive error flag is set to 1. Accordingly, clear the
FER, PER, and RDRF bits to 0 before resuming reception. Figure 16.13 shows a sample f
chart for serial data reception.



cleared to 0.

[4] If an overrun error occurs, read the
flag in SSR, and after performing th
appropriate error processing, clear t
flag to 0. Reception cannot be resu
the OER flag is set to 1.

Figure 16.13 Sample Serial Reception Flowchart (Clocked Synchronous Mo



reading the RDRF flag, reading
Also, before the MSB (bit 7) of
current frame is transmitted, re
from the TDRE flag to confirm
writing is possible. Then write
TDR.

When data is written to TDR, t
TDRE flag is automatically clea
0. When data is read from RDR
RDRF flag is automatically clea
0.

[4] If an overrun error occurs, read
OER flag in SSR, and after
performing the appropriate erro
processing, clear the OER flag
Transmission/reception cannot
resumed if the OER flag is set
For overrun error processing, s
figure 16.13.

**Figure 16.14 Sample Flowchart of Simultaneous Serial Transmit and Receive Op
(Clocked Synchronous Mode)**

cycle is a data transmission cycle. Figure 16.13 shows an example of inter-processor
communication using the multiprocessor format. The transmitting station first sends the I
of the receiving station with which it wants to perform serial communication as data with
multiprocessor bit added. It then sends transmit data as data with a 0 multiprocessor bit a
When data with a 1 multiprocessor bit is received, the receiving station compares that dat
own ID. The station whose ID matches then receives the data sent next. Stations whose II
match continue to skip data until data with a 1 multiprocessor bit is again received.

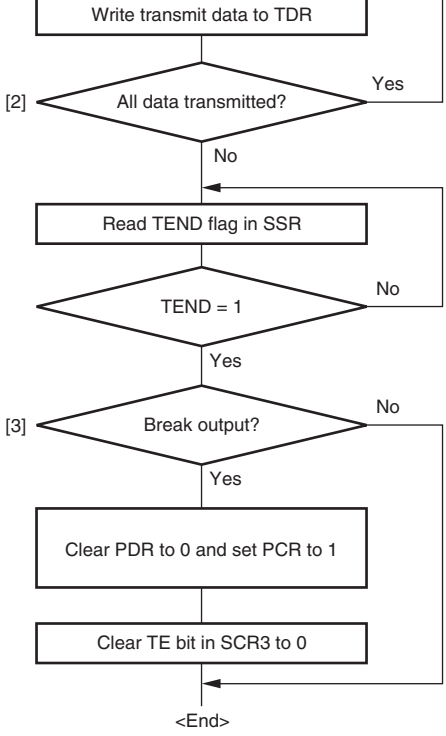
The SCI3 uses the MPIE bit in SCR3 to implement this function. When the MPIE bit is s
transfer of receive data from RSR to RDR, error flag detection, and setting the SSR status
RDRF, FER, and OER, to 1, are inhibited until data with a 1 multiprocessor bit is receive
reception of a receive character with a 1 multiprocessor bit, the MPBR bit in SSR is set to
the MPIE bit is automatically cleared, thus normal reception is resumed. If the RIE bit in
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
settings are the same as those in normal asynchronous mode. The clock used for multipro
communication is the same as that in normal asynchronous mode.

specification | receiving station specified by ID

Legend
MPB: Multiprocessor bit

Figure 16.15 Example of Inter-Processor Communication Using Multiprocessor (Transmission of Data H'AA to Receiving Station A)



[3] To output a break in serial transmission, set the port PCR to 1, clear PDR to 0, then clear the TE bit in SCR3 to 0.

Figure 16.16 Sample Multiprocessor Serial Transmission Flowchart

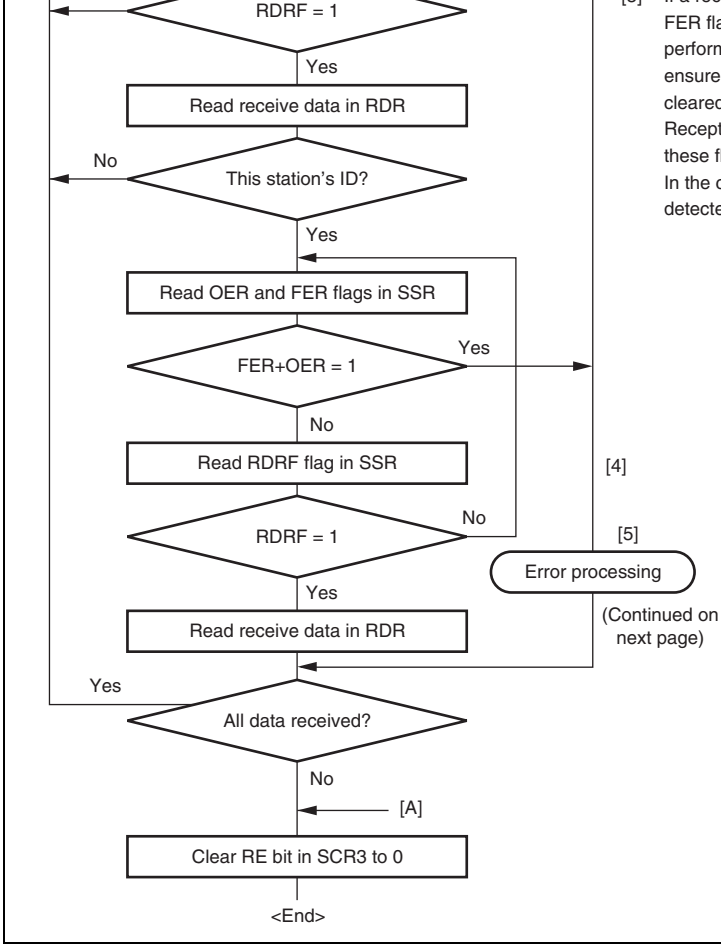


Figure 16.17 Sample Multiprocessor Serial Reception Flowchart (1)

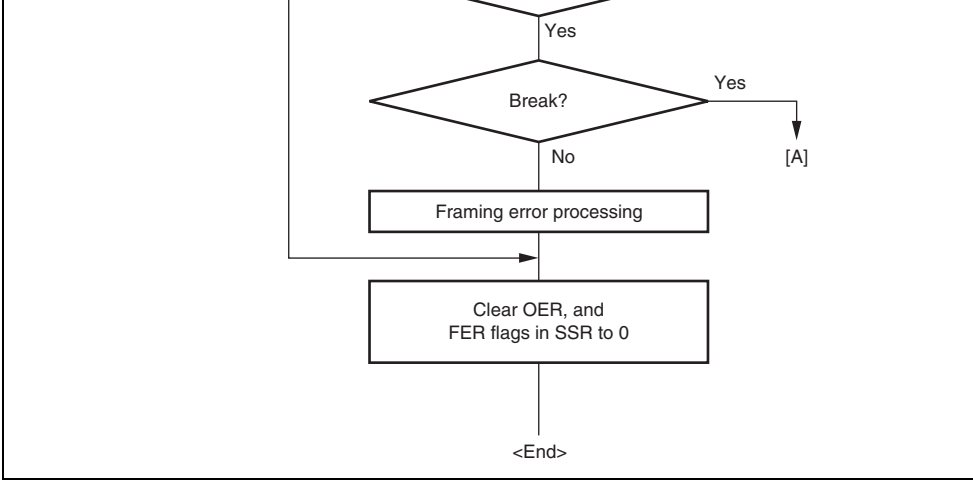
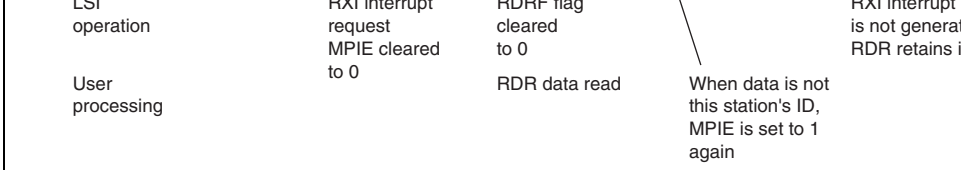
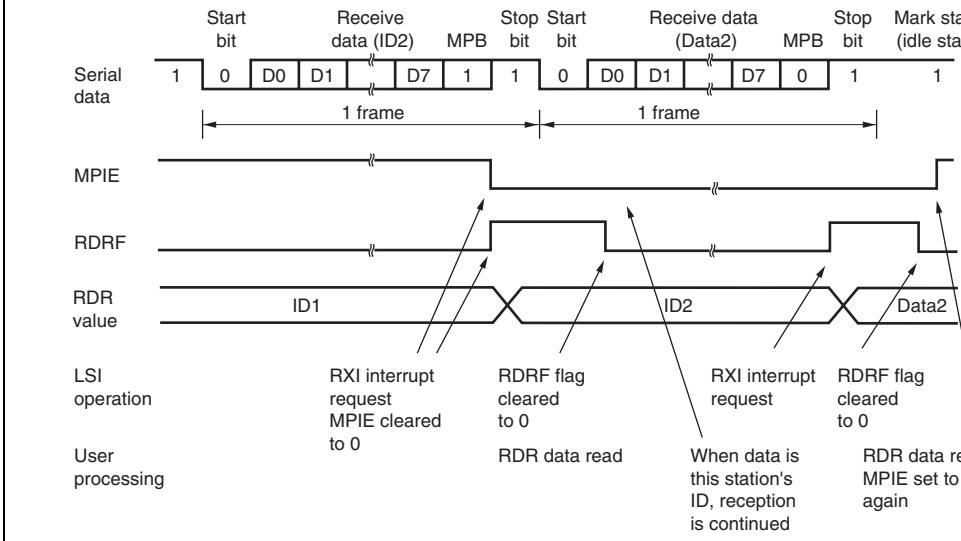


Figure 16.17 Sample Multiprocessor Serial Reception Flowchart (2)



(a) When data does not match this receiver's ID



(b) When data matches this receiver's ID

Figure 16.18 Example of SCI3 Reception Using Multiprocessor Format (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)

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

When TE is 0, the TxD pin is used as an I/O port whose direction (input or output) and l
determined by PCR and PDR. This can be used to set the TxD pin to mark state (high le
send a break during serial data transmission. To maintain the communication line at mar
until TE is set to 1, set both PCR and PDR to 1. As TE is cleared to 0 at this point, the T
becomes an I/O port, and 1 is output from the TxD pin. To send a break during serial tra
first set PCR to 1 and clear PDR to 0, and then clear TE to 0. When TE is cleared to 0, t
transmitter is initialized regardless of the current transmission state, the TxD pin becom
port, and 0 is output from the TxD pin.

16.8.3 Receive Error Flags and Transmit Operations (Clocked Synchronous Mo

Transmission cannot be started when a receive error flag (OER, PER, or FER) is set to 1
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
to 0.

[Legend]
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.
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 allowe
system design.

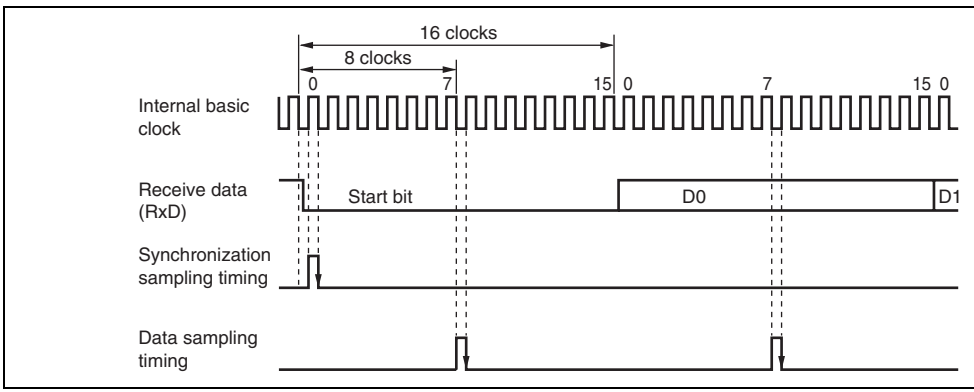


Figure 16.19 Receive Data Sampling Timing in Asynchronous Mode

17.1 Features

- Selection of I^2C format or clocked synchronous serial format
- Continuous transmission/reception

  Since the shift register, transmit data register, and receive data register are independe each other, the continuous transmission/reception can be performed.

I^2C bus format

- Start and stop conditions generated automatically in master mode
- Selection of acknowledge output levels when receiving
- Automatic loading of acknowledge bit when transmitting
- Bit synchronization/wait function

  In master mode, the state of SCL is monitored per bit, and the timing is synchronized automatically.

  If transmission/reception is not yet possible, set the SCL to low until preparations ar completed.
- Six interrupt sources

  Transmit data empty (including slave-address match), transmit end, receive data full slave-address match), arbitration lost, NACK detection, and stop condition detection
- Direct bus drive

  Two pins, SCL and SDA pins, function as NMOS open-drain outputs when the bus d function is selected.

Clocked synchronous format

- Four interrupt sources

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

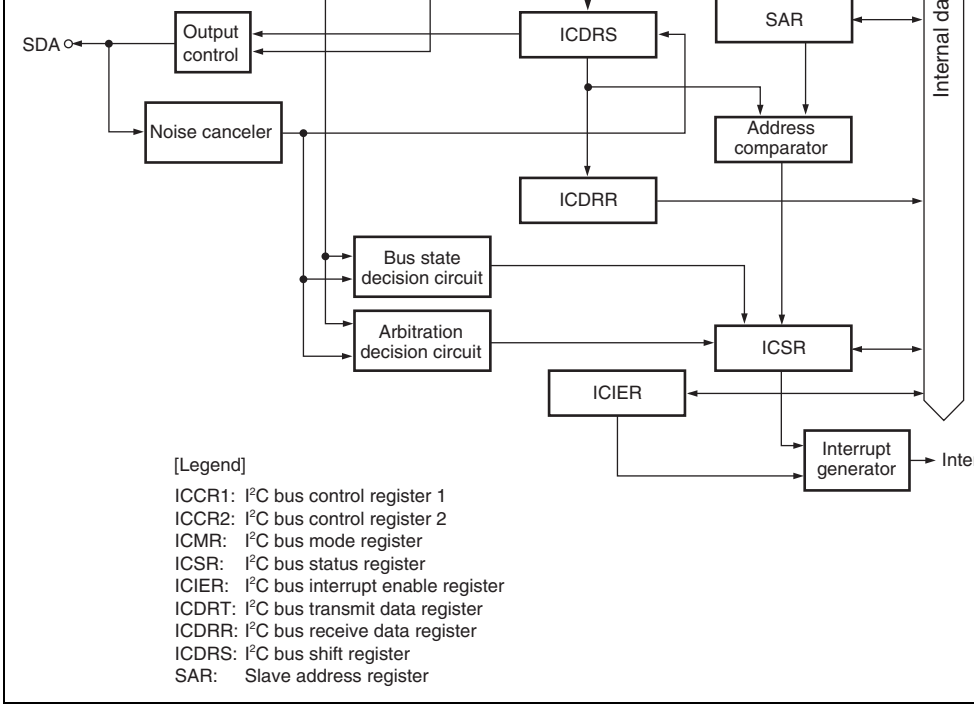


Figure 17.1 Block Diagram of I²C Bus Interface 2

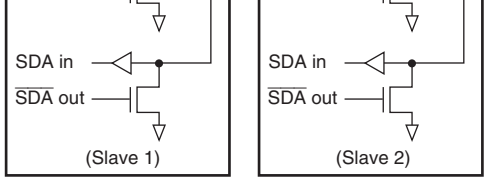


Figure 17.2 External Circuit Connections of I/O Pins

power supply (V_{cc}) of this LSI.

17.3 Register Descriptions

The I^2C bus interface 2 has the following registers:

- I^2C bus control register 1 (ICCR1)
- I^2C bus control register 2 (ICCR2)
- I^2C bus mode register (ICMR)
- I^2C bus interrupt enable register (ICIER)
- I^2C bus status register (ICSR)
- I^2C bus slave address register (SAR)
- I^2C bus transmit data register (ICDRT)
- I^2C bus receive data register (ICDRR)
- I^2C bus shift register (ICDRS)

				1: This bit is enabled for transfer operations. (S SDA pins are bus drive state.)
6	RCVD	0	R/W	Reception Disable This bit enables or disables the next operation is 0 and ICDRR is read. 0: Enables next reception 1: Disables next reception
5	MST	0	R/W	Master/Slave Select
4	TRS	0	R/W	Transmit/Receive Select In master mode with the I^2C bus format, when a is lost, MST and TRS are both reset by hardwa causing a transition to slave receive mode. Mod of the TRS bit should be made between transfe After data receive has been started in slave rec mode, when the first seven bits of the receive d with the slave address that is set to SAR and th bit is 1, TRS is automatically set to 1. If an ove occurs in master mode with the clock synchron format, MST is cleared to 0 and slave receive m entered. Operating modes are described below accordin and TRS combination. When clocked synchron format is selected and MST is 1, clock is output 00: Slave receive mode 01: Slave transmit mode 10: Master receive mode 11: Master transmit mode

Bit 3	Bit 2	Bit 1	Bit 0		Transfer Rate				
CKS3	CKS2	CKS1	CKS0	Clock	φ = 5 MHz	φ = 8 MHz	φ = 10 MHz	φ = 16 MHz	φ
0	0	0	0	φ/28	179 kHz	286 kHz	357 kHz	571 kHz	64
			1	φ/40	125 kHz	200 kHz	250 kHz	400 kHz	45
		1	0	φ/48	104 kHz	167 kHz	208 kHz	333 kHz	37
			1	φ/64	78.1 kHz	125 kHz	156 kHz	250 kHz	28
	1	0	0	φ/80	62.5 kHz	100 kHz	125 kHz	200 kHz	22
			1	φ/100	50.0 kHz	80.0 kHz	100 kHz	160 kHz	18
		1	0	φ/112	44.6 kHz	71.4 kHz	89.3 kHz	143 kHz	16
			1	φ/128	39.1 kHz	62.5 kHz	78.1 kHz	125 kHz	14
1	0	0	0	φ/56	89.3 kHz	143 kHz	179 kHz	286 kHz	32
			1	φ/80	62.5 kHz	100 kHz	125 kHz	200 kHz	22
		1	0	φ/96	52.1 kHz	83.3 kHz	104 kHz	167 kHz	18
			1	φ/128	39.1 kHz	62.5 kHz	78.1 kHz	125 kHz	14
	1	0	0	φ/160	31.3 kHz	50.0 kHz	62.5 kHz	100 kHz	11
			1	φ/200	25.0 kHz	40.0 kHz	50.0 kHz	80.0 kHz	90
		1	0	φ/224	22.3 kHz	35.7 kHz	44.6 kHz	71.4 kHz	80
			1	φ/256	19.5 kHz	31.3 kHz	39.1 kHz	62.5 kHz	70

				format, this bit has no meaning. With the I^2C bu this bit is set to 1 when the SDA level changes to low under the condition of SCL = high, assur the start condition has been issued. This bit is c 0 when the SDA level changes from low to high condition of SCL = high, assuming that the stop has been issued. Write 1 to BBSY and 0 to SCI a start condition. Follow this procedure when al transmitting a start condition. Write 0 in BBSY a SCP to issue a stop condition. To issue start/st conditions, use the MOV instruction.
6	SCP	1	W	Start/Stop Issue Condition Disable The SCP bit controls the issue of start/stop con master mode. To issue a start condition, write 1 in BBSY and A retransmit start condition is issued in the sam issue a stop condition, write 0 in BBSY and 0 in This bit is always read as 1. If 1 is written, the c stored.
5	SDAO	1	R/W	SDA Output Value Control This bit is used with SDAOP when modifying ou of SDA. This bit should not be manipulated duri transfer. 0: When reading, SDA pin outputs low. When writing, SDA pin is changed to output 1: When reading, SDA pin outputs high. When writing, SDA pin is changed to output (outputs high by external pull-up resistance)

				This bit is always read as 1, and cannot be modi
1	IICRST	0	R/W	IIC Control Part Reset This bit resets the control part except for I^2C regi this bit is set to 1 when hang-up occurs because communication failure during I^2C operation, I^2C c part can be reset without setting ports and initiali registers.
0	—	1	—	Reserved This bit is always read as 1, and cannot be modi

				Set this bit to 0 when the I²C bus format is used
6	WAIT	0	R/W	Wait Insertion Bit In master mode with the I²C bus format, this bit whether to insert a wait after data transfer exce acknowledge bit. When WAIT is set to 1, after t the clock for the final data bit, low period is exte two transfer clocks. If WAIT is cleared to 0, data acknowledge bits are transferred consecutively wait inserted. The setting of this bit is invalid in slave mode w bus format or with the clocked synchronous ser
5, 4	—	All 1	—	Reserved These bits are always read as 1, and cannot be
3	BCWP	1	R/W	BC Write Protect This bit controls the BC2 to BC0 modifications. modifying BC2 to BC0, this bit should be cleare use the MOV instruction. In clock synchronous mode, BC should not be modified. 0: When writing, values of BC2 to BC0 are set. 1: When reading, 1 is always read. When writing, settings of BC2 to BC0 are inv

synchronous serial format, these bits should not modified.

I²C Bus Format	Clock Synchronous Serial
000: 9 bits	000: 8 bits
001: 2 bits	001: 1 bits
010: 3 bits	010: 2 bits
011: 4 bits	011: 3 bits
100: 5 bits	100: 4 bits
101: 6 bits	101: 5 bits
110: 7 bits	110: 6 bits
111: 8 bits	111: 7 bits

				1: Transmit data empty interrupt request (TXI) i
6	TEIE	0	R/W	Transmit End Interrupt Enable This bit enables or disables the transmit end int (TEI) at the rising of the ninth clock while the TD ICSR is 1. TEI can be canceled by clearing the or the TEIE bit to 0. 0: Transmit end interrupt request (TEI) is disabl 1: Transmit end interrupt request (TEI) is enabl
5	RIE	0	R/W	Receive Interrupt Enable This bit enables or disables the receive data ful request (RXI) and the overrun error interrupt re (ERI) with the clocked synchronous format, whe receive data is transferred from ICDRS to ICDR RDRF bit in ICSR is set to 1. RXI can be cance clearing the RDRF or RIE bit to 0. 0: Receive data full interrupt request (RXI) and error interrupt request (ERI) with the clocked synchronous format are disabled. 1: Receive data full interrupt request (RXI) and error interrupt request (ERI) with the clocked synchronous format are enabled.

3	STIE	0	R/W	Stop Condition Detection Interrupt Enable 0: Stop condition detection interrupt request (ST disabled. 1: Stop condition detection interrupt request (ST enabled.
2	ACKE	0	R/W	Acknowledge Bit Judgement Select 0: The value of the receive acknowledge bit is ig and continuous transfer is performed. 1: If the receive acknowledge bit is 1, continuous is halted.
1	ACKBR	0	R	Receive Acknowledge In transmit mode, this bit stores the acknowledge that are returned by the receive device. This bit modified. 0: Receive acknowledge = 0 1: Receive acknowledge = 1
0	ACKBT	0	R/W	Transmit Acknowledge In receive mode, this bit specifies the bit to be se acknowledge timing. 0: 0 is sent at the acknowledge timing. 1: 1 is sent at the acknowledge timing.

				• When TRS is set • When a start condition (including re-transfe been issued • When transmit mode is entered from receiv slave mode [Clearing conditions] • When 0 is written in TDRE after reading TD • When data is written to ICDRT with an instr
6	TEND	0	R/W	Transmit End [Setting conditions] • When the ninth clock of SCL rises with the I format while the TDRE flag is 1 • When the final bit of transmit frame is sent v clock synchronous serial format [Clearing conditions] • When 0 is written in TEND after reading TE • When data is written to ICDRT with an instr
5	RDRF	0	R/W	Receive Data Register Full [Setting condition] • When a receive data is transferred from ICD ICDRR [Clearing conditions] • When 0 is written in RDRF after reading RD • When ICDRR is read with an instruction

[Setting Conditions]

- In master mode, when a stop condition is det
  after frame transfer
- In slave mode, when a stop condition is dete
  the general call address or the first byte slav
  address, next to detection of start condition,
  with the address set in SAR

[Clearing Condition]

- When 0 is written in STOP after reading STC

				[Setting conditions] • If the internal SDA and SDA pin disagree at SCL in master transmit mode • When the SDA pin outputs high in master m a start condition is detected • When the final bit is received with the clock synchronous format while RDRF = 1 [Clearing condition] • When 0 is written in AL/OVE after reading A
1	AAS	0	R/W	Slave Address Recognition Flag In slave receive mode, this flag is set to 1 if the following a start condition matches bits SVA6 to SAR. [Setting conditions] • When the slave address is detected in slave mode • When the general call address is detected i receive mode. [Clearing condition] • When 0 is written in AAS after reading AAS

17.3.6 Slave Address Register (SAR)

SAR selects the communication format and sets the slave address. When the chip is in sla
with the I^2C bus format, if the upper 7 bits of SAR match the upper 7 bits of the first fram
received after a start condition, the chip operates as the slave device.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	SVA6 to SVA0	All 0	R/W	Slave Address 6 to 0 These bits set a unique address in bits SVA6 to differing form the addresses of other slave devic connected to the I^2C bus.
0	FS	0	R/W	Format Select 0: I^2C bus format is selected. 1: Clocked synchronous serial format is selected

ICDRR is an 8-bit register that stores the receive data. When data of one byte is received transfers the receive data from ICDRS to ICDRR and the next data can be received. ICD receive-only register, therefore the CPU cannot write to this register. The initial value o is H'FF.

17.3.9 I^2C Bus Shift Register (ICDRS)

ICDRS is a register that is used to transfer/receive data. In transmission, data is transferr ICDRT to ICDRS and the data is sent from the SDA pin. In reception, data is transferred ICDRS to ICDRR after data of one byte is received. This register cannot be read directly CPU.

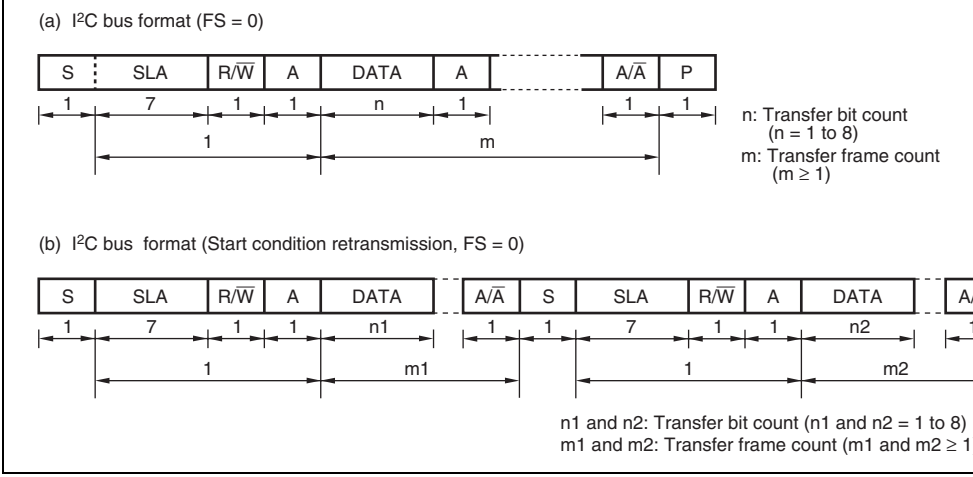


Figure 17.3 I²C Bus Formats

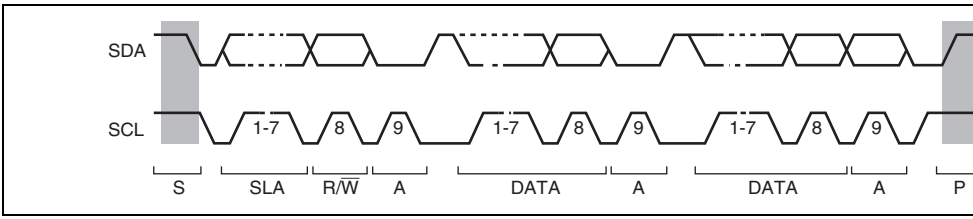


Figure 17.4 I²C Bus Timing

In master transmit mode, the master device outputs the transmit clock and transmit data, slave device returns an acknowledge signal. For master transmit mode operation timing, figures 17.5 and 17.6. The transmission procedure and operations in master transmit mo described below.

1. Set the ICE bit in ICCR1 to 1. Set the MLS and WAIT bits in ICMR and the CKS3 t bits in ICCR1 to 1. (Initial setting)
2. Read the BBSY flag in ICCR2 to confirm that the bus is free. Set the MST and TRS ICCR1 to select master transmit mode. Then, write 1 to BBSY and 0 to SCP using M instruction. (Start condition issued) This generates the start condition.
3. After confirming that TDRE in ICSR has been set, write the transmit data (the first b show the slave address and R/$\overline{W}$) to ICDRT. At this time, TDRE is automatically cle and data is transferred from ICDRT to ICDRS. TDRE is set again.
4. When transmission of one byte data is completed while TDRE is 1, TEND in ICSR i at the rise of the 9th transmit clock pulse. Read the ACKBR bit in ICIER, and confir slave device has been selected. Then, write second byte data to ICDRT. When ACK the slave device has not been acknowledged, so issue the stop condition. To issue the condition, write 0 to BBSY and SCP using MOV instruction. SCL is fixed low until transmit data is prepared or the stop condition is issued.
5. The transmit data after the second byte is written to ICDRT every time TDRE is set.
6. Write the number of bytes to be transmitted to ICDRT. Wait until TEND is set (the e byte data transmission) while TDRE is 1, or wait for NACK (NACKF in ICSR = 1) receive device while ACKE in ICIER is 1. Then, issue the stop condition to clear TE NACKF.
7. When the STOP bit in ICSR is set to 1, the operation returns to the slave receive mo

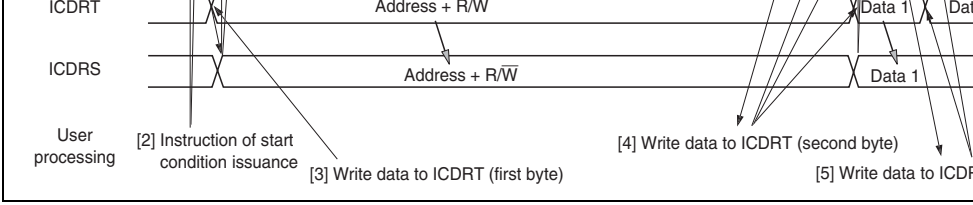


Figure 17.5 Master Transmit Mode Operation Timing (1)

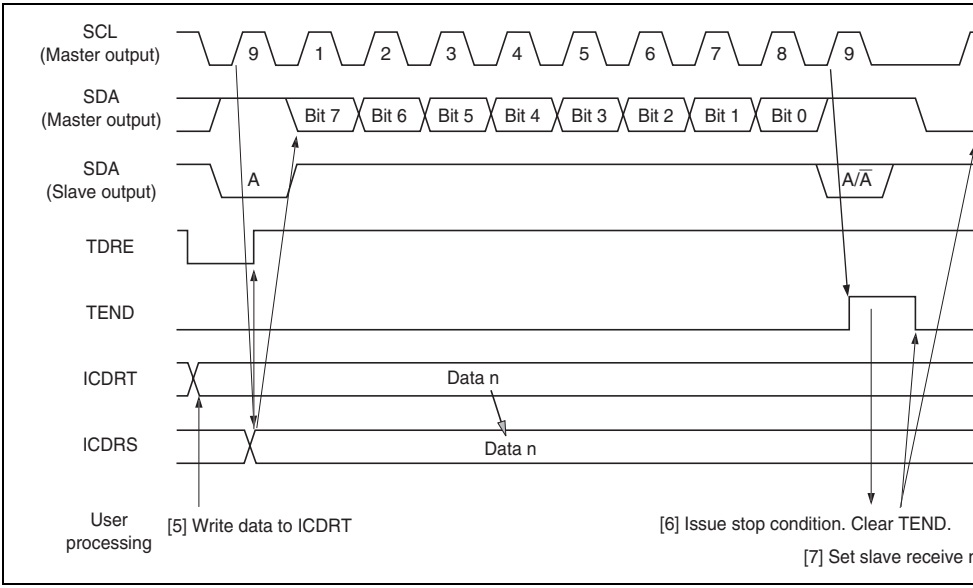


Figure 17.6 Master Transmit Mode Operation Timing (2)

level specified by ACKBT in ICIER to SDA, at the 9th receive clock pulse.

3. After the reception of first frame data is completed, the RDRF bit in ICST is set to 1 of 9th receive clock pulse. At this time, the receive data is read by reading ICDRR, a is cleared to 0.
4. The continuous reception is performed by reading ICDRR every time RDRF is set. I receive clock pulse falls after reading ICDRR by the other processing while RDRF i fixed low until ICDRR is read.
5. If next frame is the last receive data, set the RCVD bit in ICCR1 to 1 before reading This enables the issuance of the stop condition after the next reception.
6. When the RDRF bit is set to 1 at rise of the 9th receive clock pulse, issue the stage c
7. When the STOP bit in ICSR is set to 1, read ICDRR. Then clear the RCVD bit to 0.
8. The operation returns to the slave receive mode.

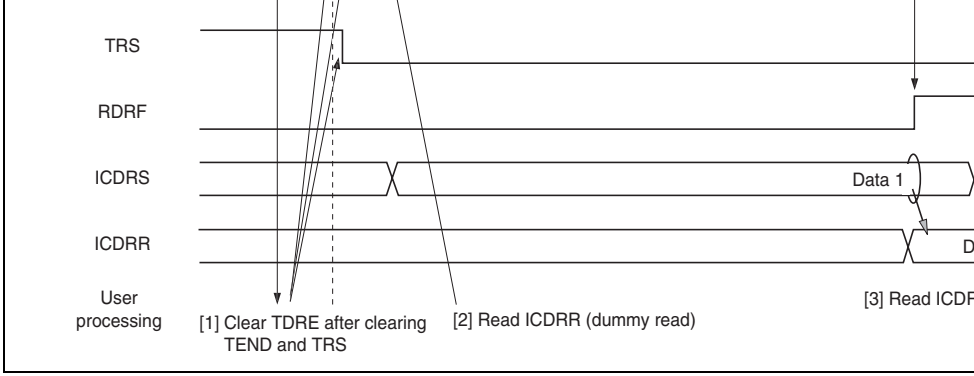


Figure 17.7 Master Receive Mode Operation Timing (1)

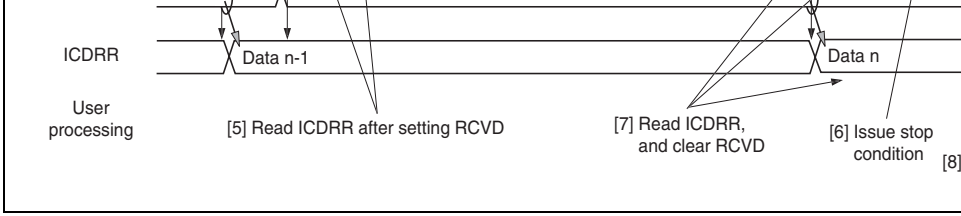


Figure 17.8 Master Receive Mode Operation Timing (2)

17.4.4 Slave Transmit Operation

In slave transmit mode, the slave device outputs the transmit data, while the master devi the receive clock and returns an acknowledge signal. For slave transmit mode operation refer to figures 17.9 and 17.10.

The transmission procedure and operations in slave transmit mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set the MLS and WAIT bits in ICMR and the CKS3 t bits in ICCR1 to 1. (Initial setting) Set the MST and TRS bits in ICCR1 to select sla mode, and wait until the slave address matches.
2. When the slave address matches in the first frame following detection of the start co the slave device outputs the level specified by ACKBT in ICIER to SDA, at the rise clock pulse. At this time, if the 8th bit data (R/$\overline{W}$) is 1, the TRS and ICSR bits in IC set to 1, and the mode changes to slave transmit mode automatically. The continuou transmission is performed by writing transmit data to ICDRT every time TDRE is se
3. If TDRE is set after writing last transmit data to ICDRT, wait until TEND in ICSR is with TDRE = 1. When TEND is set, clear TEND.
4. Clear TRS for the end processing, and read ICDRR (dummy read). SCL is free.
5. Clear TDRE.

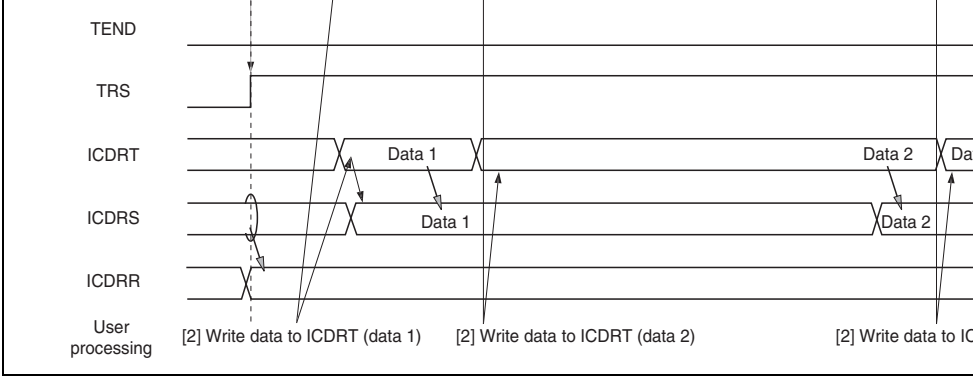


Figure 17.9 Slave Transmit Mode Operation Timing (1)

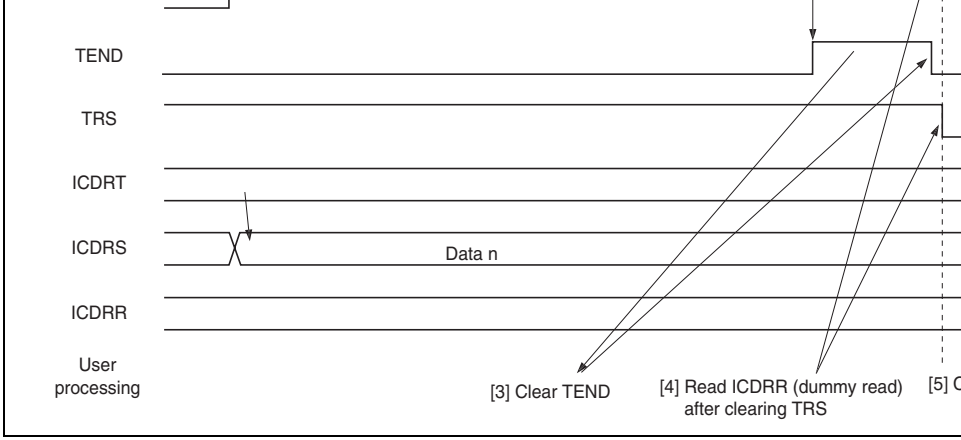


Figure 17.10 Slave Transmit Mode Operation Timing (2)

2. When the slave address matches in the first frame following detection of the start cond the slave device outputs the level specified by ACKBT in ICIER to SDA, at the rise o clock pulse. At the same time, RDRF in ICSR is set to read ICDRR (dummy read). (S read data show the slave address and R/$\overline{W}$, it is not used.)
3. Read ICDRR every time RDRF is set. If 8th receive clock pulse falls while RDRF is fixed low until ICDRR is read. The change of the acknowledge before reading ICDRR returned to the master device, is reflected to the next transmit frame.
4. The last byte data is read by reading ICDRR.

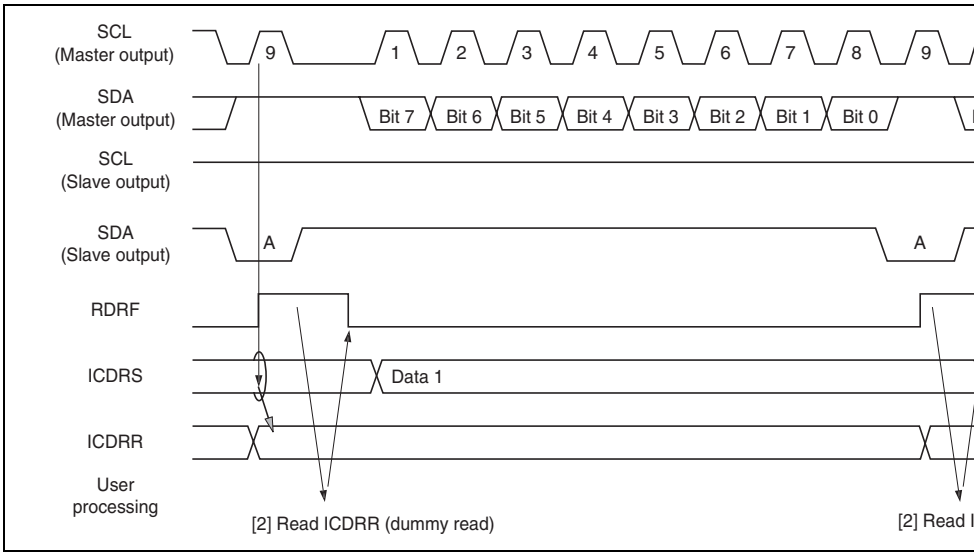


Figure 17.11 Slave Receive Mode Operation Timing (1)

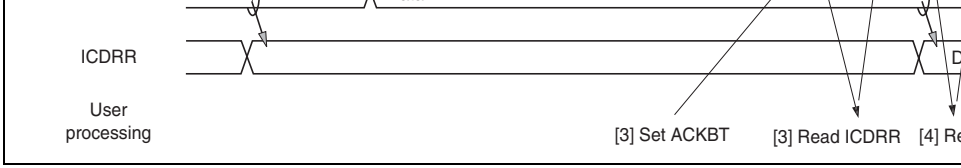


Figure 17.12 Slave Receive Mode Operation Timing (2)

17.4.6 Clocked Synchronous Serial Format

This module can be operated with the clocked synchronous serial format, by setting the SAR to 1. When the MST bit in ICCR1 is 1, the transfer clock output from SCL is selec MST is 0, the external clock input is selected.

(1) Data Transfer Format

Figure 17.13 shows the clocked synchronous serial transfer format.

The transfer data is output from the rise to the fall of the SCL clock, and the data at the r of the SCL clock is guaranteed. The MLS bit in ICMR sets the order of data transfer, in MSB first or LSB first. The output level of SDA can be changed during the transfer wai SDAO bit in ICCR2.

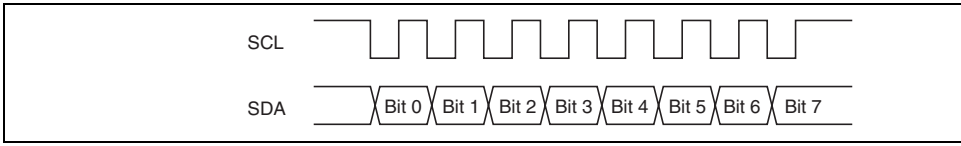


Figure 17.13 Clocked Synchronous Serial Transfer Format

transferred from ICDRT to ICDRS, and TDRE is set automatically. The continuous transmission is performed by writing data to ICDRT every time TDRE is set. When c from transmit mode to receive mode, clear TRS while TDRE is 1.

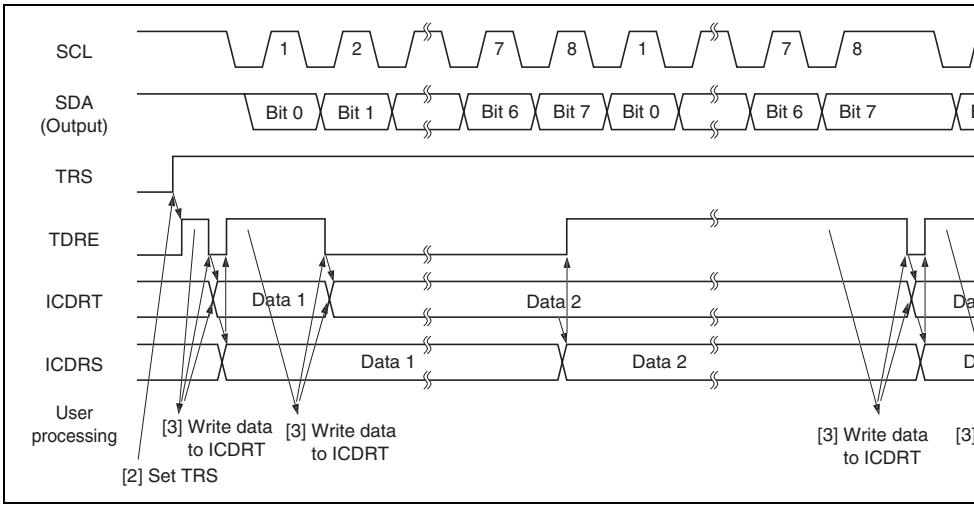


Figure 17.14 Transmit Mode Operation Timing

continually output. The continuous reception is performed by reading ICDRR every
RDRF is set. When the 8th clock is risen while RDRF is 1, the overrun is detected a
AL/OVE in ICSR is set. At this time, the previous reception data is retained in ICDR

4. To stop receiving when MST = 1, set RCVD in ICCR1 to 1, then read ICDRR. Then
fixed high after receiving the next byte data.

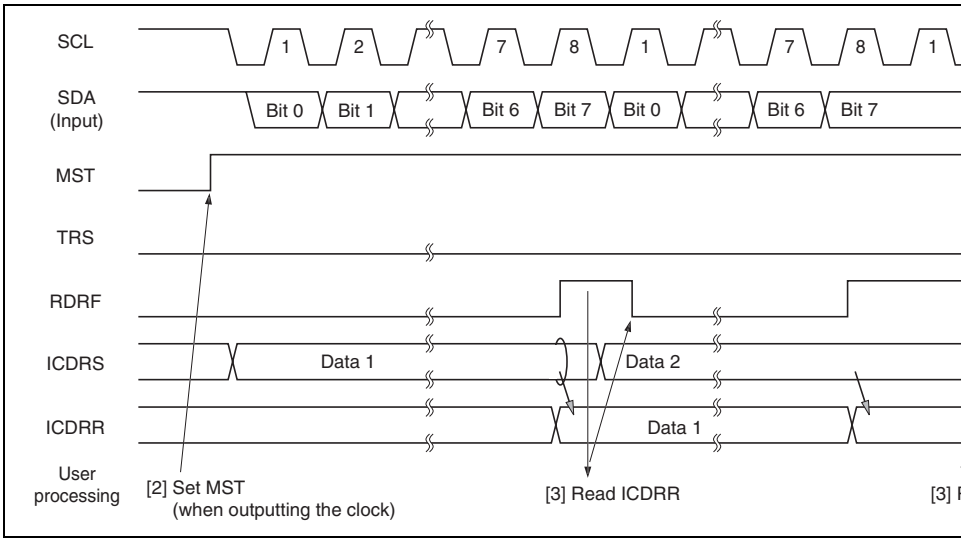


Figure 17.15 Receive Mode Operation Timing

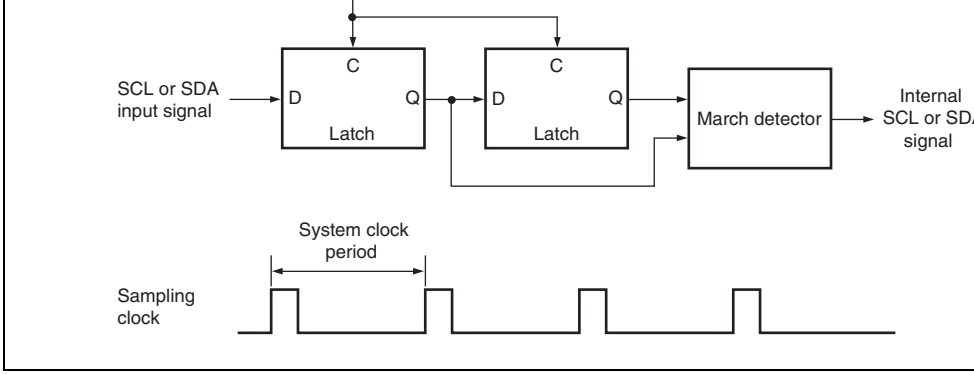


Figure 17.16 Block Diagram of Noise Conceler

17.4.8 Example of Use

Flowcharts in respective modes that use the I^2C bus interface are shown in figures 17.17 t

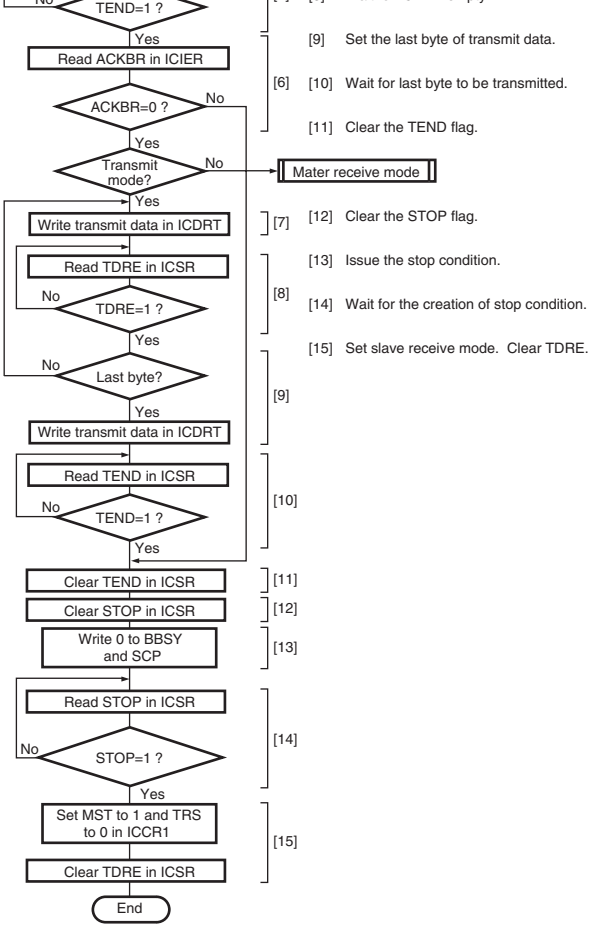
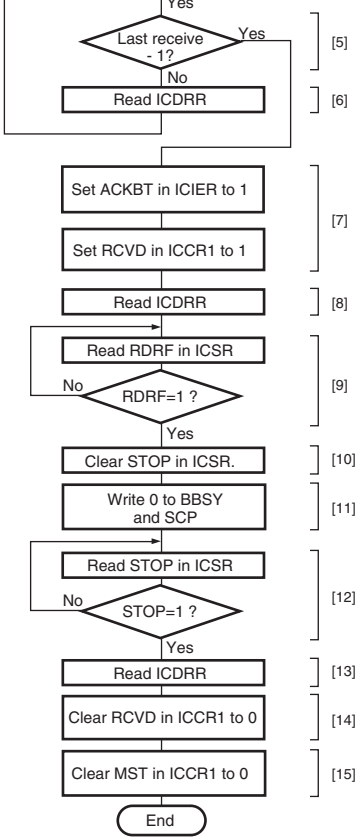


Figure 17.17 Sample Flowchart for Master Transmit Mode



[9] Wait for the last byte to be receive.

[10] Clear the STOP flag.

[11] Issue the stop condition.

[12] Wait for the creation of stop condition.

[13] Read the last byte of receive data.

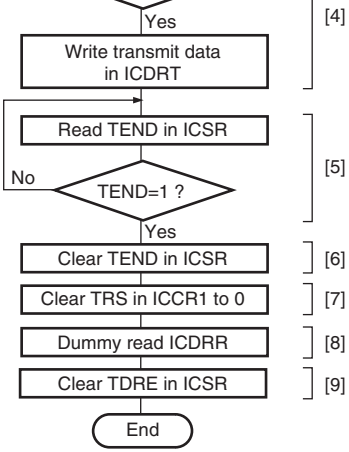
[14] Clear RCVD.

[15] Set slave receive mode.

Note: Do not activate an interrupt during the execution of steps [1] to [3].

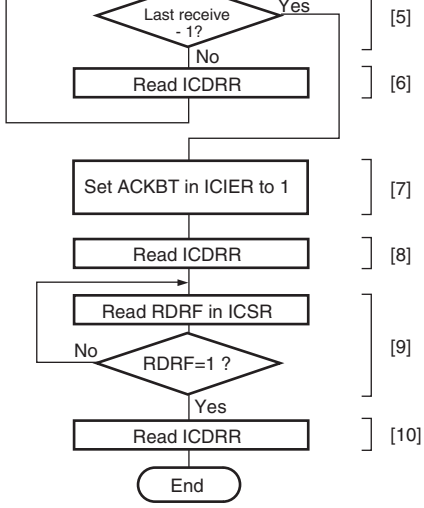
Supplementary explanation: When one byte is received, steps [2] to [6] a
skipped after step [1], before jumping to step
The step [8] is dummy-read in ICDRR.

Figure 17.18 Sample Flowchart for Master Receive Mode



[9] Clear the TDRE flag.

Figure 17.19 Sample Flowchart for Slave Transmit Mode



[8] Read the (last byte - 1) of receive data.

[9] Wait the last byte to be received.

[10] Read for the last byte of receive data.

Supplementary explanation: When one byte is received, steps [2] to [6] are skipped after step [1], before jumping to step [7]. The step [8] is dummy-read in ICDRR.

Figure 17.20 Sample Flowchart for Slave Receive Mode

Transmit End	TEI	(TEND=1) • (TEIE=1)	○	○
Receive Data Full	RXI	(RDRF=1) • (RIE=1)	○	○
STOP Recognition	STPI	(STOP=1) • (STIE=1)	○	×
NACK Receive	NAKI	{(NACKF=1)+(AL=1)} • (NAKIE=1)	○	×
Arbitration Lost/Overrun Error			○	○

When interrupt conditions described in table 17.3 are 1 and the I bit in CCR is 0, the CP executes an interrupt exception processing. Interrupt sources should be cleared in the ex processing. TDRE and TEND are automatically cleared to 0 by writing the transmit data ICDRT. RDRF are automatically cleared to 0 by reading ICDRR. TDRE is set to 1 agai same time when transmit data is written to ICDRT. When TDRE is cleared to 0, then an data of one byte may be transmitted.

Figure 17.21 shows the timing of the bit synchronous circuit and table 17.4 shows the tim
SCL output changes from low to Hi-Z then SCL is monitored.

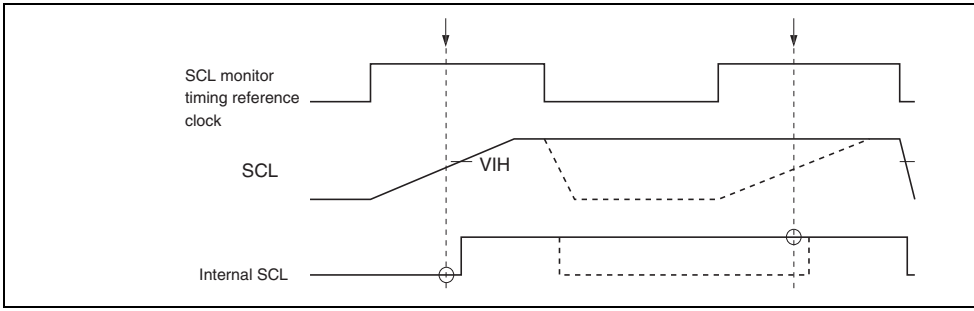


Figure 17.21 The Timing of the Bit Synchronous Circuit

Table 17.4 Time for Monitoring SCL

CKS3	CKS2	Time for Monitoring SCL
0	0	7.5 tcyc
	1	19.5 tcyc
1	0	17.5 tcyc
	1	41.5 tcyc

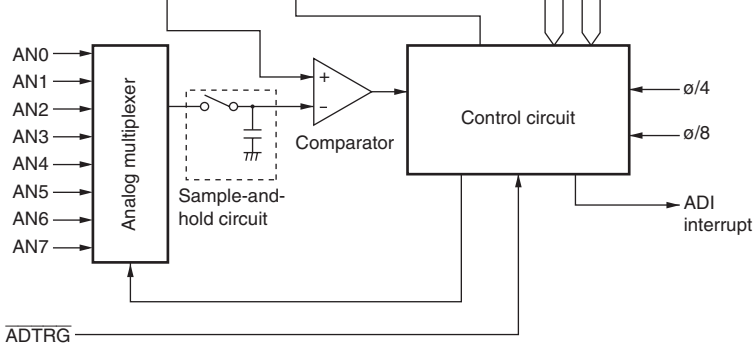
Circuit, by the load of the SCL bus (load capacitance or pull-up resistance)

2. When the bit synchronous circuit is activated by extending the low period of eighth a clocks, that is driven by the slave device

17.7.2 WAIT Setting in I²C Bus Mode Register (ICMR)

If the WAIT bit is set to 1, and the SCL signal is driven low for two or more transfer clo slave device at the eighth and ninth clocks, the high period of ninth clock may be shorte avoid this, set the WAIT bit in ICMR to 0.

- Conversion time: at least 3.9 µs per channel (at 18-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 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

Figure 18.1 Block Diagram of A/D Converter

Analog input pin 0	AN0	Input	Group 0 analog input
Analog input pin 1	AN1	Input	
Analog input pin 2	AN2	Input	
Analog input pin 3	AN3	Input	
Analog input pin 4	AN4	Input	Group 1 analog input
Analog input pin 5	AN5	Input	
Analog input pin 6	AN6	Input	
Analog input pin 7	AN7	Input	
A/D external trigger input pin	$\overline{\text{ADTRG}}$	Input	External trigger input for A/D conversion

18.3.1 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 r
A/D conversion. The ADDR registers, which store a conversion result for each analog in
channel, are shown in table 18.2.

The converted 10-bit data is stored in bits 15 to 6. The lower 6 bits are always read as 0.

The data bus width between the CPU and the A/D converter is 8 bits. The upper byte can
directly from the CPU, however the lower byte should be read via a temporary register. T
temporary register contents are transferred from the ADDR when the upper byte data is r
Therefore byte access to ADDR should be done by reading the upper byte first then the lo
Word access is also possible. ADDR is initialized to H'0000.

Table 18.2 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 Conversi**
AN0	AN4	ADDRA
AN1	AN5	ADDRB
AN2	AN6	ADDRC
AN3	AN7	ADDRD

				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 request (ADI) is en ADF when this bit is set to 1
5	ADST	0	R/W	A/D Start Setting this bit to 1 starts A/D conversion. In sin this bit is cleared to 0 automatically when conve the specified channel is complete. In scan mode conversion continues sequentially on the specifi channels until this bit is cleared to 0 by software a transition to standby mode.
4	SCAN	0	R/W	Scan Mode Selects single mode or scan mode as the A/D c 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 cor time.

	101: AN5	101: AN4 and AN5
	110: AN6	110: AN4 to AN6
	111: AN7	111: AN4 to AN7

18.3.3 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 t edge of the external trigger signal ($\overline{\text{ADTRG}}$) wher set to 1. The selection between the falling edge and rising the external trigger pin ($\overline{\text{ADTRG}}$) conforms to the 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/

channel as follows:

1. A/D conversion is started when the ADST bit in ADCSR is set to 1, according to sof external trigger input.
2. When A/D conversion is completed, the result is transferred to the corresponding A/ register of the channel.
3. On completion of conversion, the ADF bit in ADCSR is set to 1. If the ADIE bit is s 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, bit is automatically cleared to 0 and the A/D converter enters the wait state.

18.4.2 Scan Mode

In scan mode, A/D conversion is performed sequentially for the analog input of the spec channels (four channels maximum) as follows:

1. When the ADST bit in ADCSR 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 C
2. When A/D conversion for each channel is completed, the result is sequentially transf the A/D data register corresponding to each channel.
3. When conversion of all the selected channels is completed, the ADF flag in ADCSR If the ADIE bit is set to 1 at this time, an ADI interrupt requested is generated. A/D c starts again on the first channel in the group.
4. The ADST bit is not automatically cleared to 0. Steps [2] and [3] are repeated as lon ADST bit remains set to 1. When the ADST bit is cleared to 0, A/D conversion stops

In scan mode, the values given in table 18.3 apply to the first conversion time. In the secc
subsequent conversions, the conversion time is 128 states (fixed) when CKS = 0 and 66 s
(fixed) when CKS = 1.

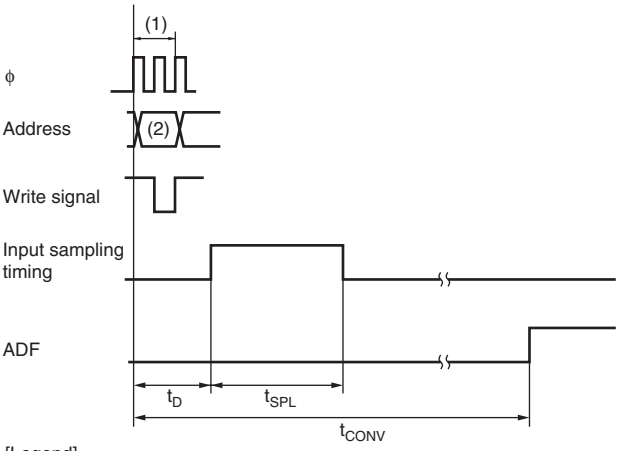


Figure 18.2 A/D Conversion Timing

18.4.4 External Trigger Input Timing

A/D conversion can also be started by an external trigger input. When the TRGE bit in A set to 1, external trigger input is enabled at the $\overline{\text{ADTRG}}$ pin. A falling edge at the $\overline{\text{ADTR}}$ pin sets the ADST bit in ADCSR to 1, starting A/D conversion. Other operations, in bo and scan modes, are the same as when the bit ADST has been set to 1 by software. Figu shows the timing.

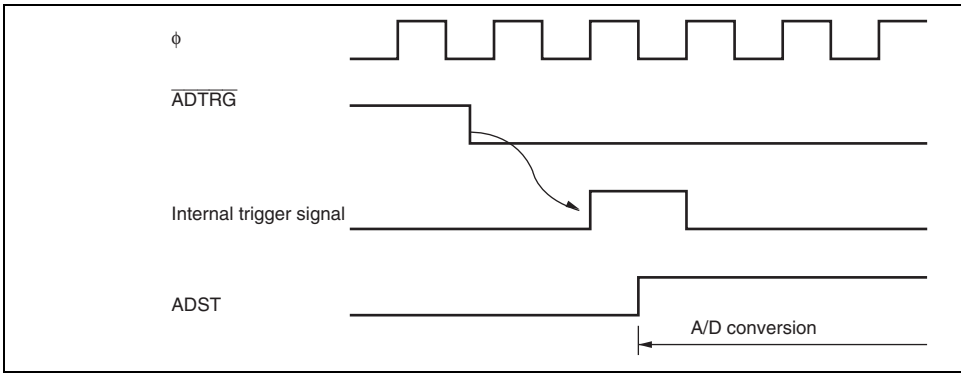


Figure 18.3 External Trigger Input Timing

when the digital output changes from the minimum voltage value 0000000000 to 000
(see figure 18.5).

- Full-scale error

  The deviation of the analog input voltage value from the ideal A/D conversion charac
  when the digital output changes from 1111111110 to 1111111111 (see figure 18.5).

- Nonlinearity error

  The deviation from the ideal A/D conversion characteristic as the voltage changes fro
  full scale. This does not include the offset error, full-scale error, or quantization error

- Absolute accuracy

  The deviation between the digital value and the analog input value. Includes offset er
  scale error, quantization error, and nonlinearity error.

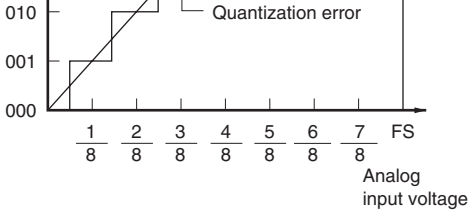


Figure 18.4 A/D Conversion Accuracy Definitions (1)

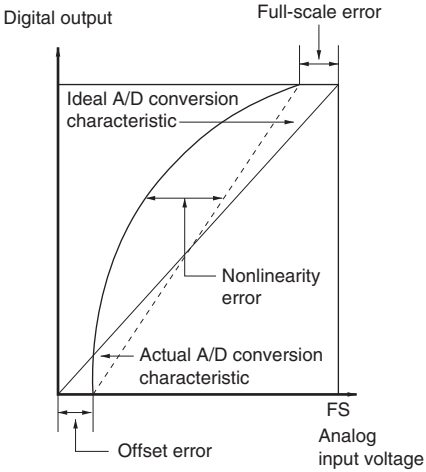


Figure 18.5 A/D Conversion Accuracy Definitions (2)

input resistance of 10 kΩ, and the signal source impedance is ignored. However, as a low filter effect is obtained in this case, it may not be possible to follow an analog signal with differential coefficient (e.g., 5 mV/µs or greater) (see figure 18.6). When converting a hi analog signal or converting in scan mode, a low-impedance buffer should be inserted.

18.6.2 Influences on Absolute Accuracy

Adding capacitance results in coupling with GND, and therefore noise in GND may adve 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 ac antennas on the mounting board.

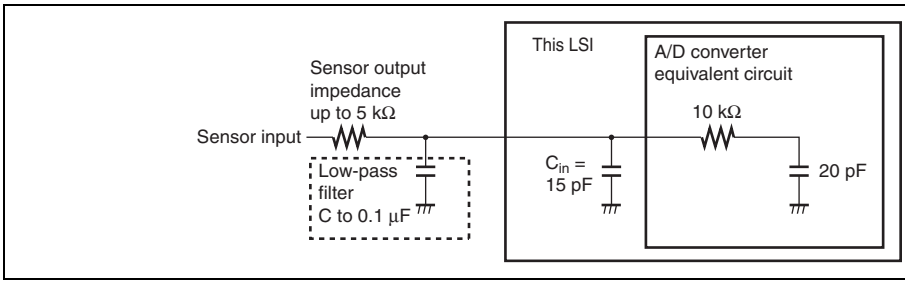


Figure 18.6 Analog Input Circuit Example

Do not attempt to access reserved addresses.

- When the address is 16-bit wide, the address of the upper byte is given in the list.
- 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 addr
- 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 spec for an on-chip peripheral module, refer to the section on that on-chip peripheral mod

—	—	—	H'F000 to H'F6FF	—	—
Timer control register_0	TCR_0	8	H'F700	Timer Z	8
Timer I/O control register A_0	TIORA_0	8	H'F701	Timer Z	8
Timer I/O control register C_0	TIORC_0	8	H'F702	Timer Z	8
Timer status register_0	TSR_0	8	H'F703	Timer Z	8
Timer interrupt enable register_0	TIER_0	8	H'F704	Timer Z	8
PWM mode output level control register_0	POCR_0	8	H'F705	Timer Z	8
Timer counter_0	TCNT_0	16	H'F706	Timer Z	16
General register A_0	GRA_0	16	H'F708	Timer Z	16
General register B_0	GRB_0	16	H'F70A	Timer Z	16
General register C_0	GRC_0	16	H'F70C	Timer Z	16
General register D_0	GRD_0	16	H'F70E	Timer Z	16
Timer control register_1	TCR_1	8	H'F710	Timer Z	8
Timer I/O control register A_1	TIORA_1	8	H'F711	Timer Z	8
Timer I/O control register C_1	TIORC_1	8	H'F712	Timer Z	8
Timer status register_1	TSR_1	8	H'F713	Timer Z	8
Timer interrupt enable register_1	TIER_1	8	H'F714	Timer Z	8
PWM mode output level control register_1	POCR_1	8	H'F715	Timer Z	8
Timer counter_1	TCNT_1	16	H'F716	Timer Z	16

Timer Z, for common use	TFCR	8	H'F723	Timer Z	8
Timer output master enable register	TOER	8	H'F724	Timer Z	8
Timer output control register	TOCR	8	H'F725	Timer Z	8
—	—	—	H'F726, H'F727	—	—
Second data register/free running counter data register	RSECDR	8	H'F728	RTC	8
Minute data register	RMINDR	8	H'F729	RTC	8
Hour data register	RHRDR	8	H'F72A	RTC	8
Day-of-week data register	RWKDR	8	H'F72B	RTC	8
RTC control register 1	RTCCR1	8	H'F72C	RTC	8
RTC control register 2	RTCCR2	8	H'F72D	RTC	8
—	—	—	H'F72E	RTC	—
Clock source select register	RTCCSR	8	H'F72F	RTC	8
—	—	—	H'F730 to H'F73F	—	—
Serial mode register_2	SMR_2	8	H'F740	SCI3_2	8
Bit rate register_2	BRR_2	8	H'F741	SCI3_2	8
Serial control register 3_2	SCR3_2	8	H'F742	SCI3_2	8
Transmit data register_2	TDR_2	8	H'F743	SCI3_2	8
Serial status register_2	SSR_2	8	H'F744	SCI3_2	8
Receive data register_2	RDR_2	8	H'F745	SCI3_2	8
—	—	—	H'F746, H'F747	—	—

I2C bus receive data register	ICDRR	8	H'F74F	IIC2	8
—	—	—	H'F750 to H'F75F	—	—
Timer mode register B1	TMB1	8	H'F760	Timer B1	8
Timer counter B1	TCB1	8	H'F761	Timer B1	8
Timer load register B1	TLB1	8	H'F761	Timer B1	8
—	—	—	H'F762 to H'FF8F	—	—
Flash memory control register 1	FLMCR1	8	H'FF90	ROM	8
Flash memory control register 2	FLMCR2	8	H'FF91	ROM	8
Flash memory power control register	FLPWCR	8	H'FF92	ROM	8
Erase block register 1	EBR1	8	H'FF93	ROM	8
—	—	—	H'FF94 to H'FF9A	—	—
Flash memory enable register	FENR	8	H'FF9B	ROM	8
—	—	—	H'FF9C to H'FF9F	—	—
Timer control register V0	TCRV0	8	H'FFA0	Timer V	8
Timer control/status register V	TCSRV	8	H'FFA1	Timer V	8
Time constant register A	TCORA	8	H'FFA2	Timer V	8
Time constant register B	TCORB	8	H'FFA3	Timer V	8
Timer counter V	TCNTV	8	H'FFA4	Timer V	8
Timer control register V1	TCRV1	8	H'FFA5	Timer V	8
—	—	—	H'FFA6, H'FFA7	—	—

			H'FFAF		
A/D data register	ADDRA	16	H'FFB0	A/D converter	8
A/D data register	ADDRB	16	H'FFB2	A/D converter	8
A/D data register	ADDRC	16	H'FFB4	A/D converter	8
A/D data register	ADDRD	16	H'FFB6	A/D converter	8
A/D control/status register	ADCSR	8	H'FFB8	A/D converter	8
A/D control register	ADCR	8	H'FFB9	A/D converter	8
—	—	—	H'FFBA, H'FFBB	—	—
PWM data register L	PWDRL	8	H'FFBC	14-bit PWM	8
PWM data register U	PWDRU	8	H'FFBD	14-bit PWM	8
PWM control register	PWCR	8	H'FFBE	14-bit PWM	8
—	—	—	H'FFBF	14-bit PWM	—
Timer control/status register WD	TCSRWD	8	H'FFC0	WDT*	8
Timer counter WD	TCWD	8	H'FFC1	WDT*	8
Timer mode register WD	TMWD	8	H'FFC2	WDT*	8
—	—	—	H'FFC3	WDT*	—
—	—	—	H'FFC4 to H'FFC7	—	—
Address break control register	ABRKCR	8	H'FFC8	Address break	8
Address break status register	ABRKSR	8	H'FFC9	Address break	8
Break address register H	BARH	8	H'FFCA	Address break	8
Break address register L	BARL	8	H'FFCB	Address break	8
Break data register H	BDRH	8	H'FFCC	Address break	8

Port data register 3	PDR3	8	H'FFD6	I/O port	8
—	—	—	H'FFD7	I/O port	—
Port data register 5	PDR5	8	H'FFD8	I/O port	8
Port data register 6	PDR6	8	H'FFD9	I/O port	8
Port data register 7	PDR7	8	H'FFDA	I/O port	8
Port data register 8	PDR8	8	H'FFDB	I/O port	8
—	—	—	H'FFDC	I/O port	—
Port data register B	PDRB	8	H'FFDD	I/O port	8
—	—	—	H'FFDE, H'FFDF	—	—
Port mode register 1	PMR1	8	H'FFE0	I/O port	8
Port mode register 5	PMR5	8	H'FFE1	I/O port	8
Port mode register 3	PMR3	8	H'FFE2	I/O port	8
—	—	—	H'FFD3	I/O port	—
Port control register 1	PCR1	8	H'FFE4	I/O port	8
Port control register 2	PCR2	8	H'FFE5	I/O port	8
Port control register 3	PCR3	8	H'FFE6	I/O port	8
—	—	—	H'FFE7	I/O port	—
Port control register 5	PCR5	8	H'FFE8	I/O port	8
Port control register 6	PCR6	8	H'FFE9	I/O port	8
Port control register 7	PCR7	8	H'FFEA	I/O port	8
Port control register 8	PCR8	8	H'FFEB	I/O port	8
—	—	—	H'FFEC to H'FFEF	—	—

Interrupt flag register 2	IRR2	8	H'FFF7	Interrupt	8
Wakeup interrupt flag register	IWPR	8	H'FFF8	Interrupt	8
Module standby control register 1	MSTCR1	8	H'FFF9	Low power	8
Module standby control register 2	MSTCR2	8	H'FFFA	Low power	8
—	—	—	H'FFFB to H'FFFF	—	—

Note: * WDT: Watchdog timer

TIORC_0	—	IOD2	IOD1	IOD0	—	IOC2	IOC1	IOC0
TSR_0	—	—	—	OVF	IMFD	IMFC	IMFB	IMFA
TIER_0	—	—	—	OVIE	IMIED	IMIEC	IMIEB	IMIEA
POCR_0	—	—	—	—	—	POLD	POLC	POLB
TCNT_0	TCNT0H7	TCNT0H6	TCNT0H5	TCNT0H4	TCNT0H3	TCNT0H2	TCNT0H1	TCNT0H0
	TCNT0L7	TCNT0L6	TCNT0L5	TCNT0L4	TCNT0L3	TCNT0L2	TCNT0L1	TCNT0L0
GRA_0	GRA0H7	GRA0H6	GRA0H5	GRA0H4	GRA0H3	GRA0H2	GRA0H1	GRA0H0
	GRA0L7	GRA0L6	GRA0L5	GRA0L4	GRA0L3	GRA0L2	GRA0L1	GRA0L0
GRB_0	GRB0H7	GRB0H6	GRB0H5	GRB0H4	GRB0H3	GRB0H2	GRB0H1	GRB0H0
	GRB0L7	GRB0L6	GRB0L5	GRB0L4	GRB0L3	GRB0L2	GRB0L1	GRB0L0
GRC_0	GRC0H7	GRC0H6	GRC0H5	GRC0H4	GRC0H3	GRC0H2	GRC0H1	GRC0H0
	GRC0L7	GRC0L6	GRC0L5	GRC0L4	GRC0L3	GRC0L2	GRC0L1	GRC0L0
GRD_0	GRD0H7	GRD0H6	GRD0H5	GRD0H4	GRD0H3	GRD0H2	GRD0H1	GRD0H0
	GRD0L7	GRD0L6	GRD0L5	GRD0L4	GRD0L3	GRD0L2	GRD0L1	GRD0L0
TCR_1	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
TIORA_1	—	IOB2	IOB1	IOB0	—	IOA2	IOA1	IOA0
TIORC_1	—	IOD2	IOD1	IOD0	—	IOC2	IOC1	IOC0
TSR_1	—	—	UDF	OVF	IMFD	IMFC	IMFB	IMFA
TIER_1	—	—	—	OVIE	IMIED	IMIEC	IMIEB	IMIEA
POCR_1	—	—	—	—	—	POLD	POLC	POLB
TCNT_1	TCNT1H7	TCNT1H6	TCNT1H5	TCNT1H4	TCNT1H3	TCNT1H2	TCNT1H1	TCNT1H0
	TCNT1L7	TCNT1L6	TCNT1L5	TCNT1L4	TCNT1L3	TCNT1L2	TCNT1L1	TCNT1L0
GRA_1	GRA1H7	GRA1H6	GRA1H5	GRA1H4	GRA1H3	GRA1H2	GRA1H1	GRA1H0
	GRA1L7	GRA1L6	GRA1L5	GRA1L4	GRA1L3	GRA1L2	GRA1L1	GRA1L0

TMDR	BFD1	BFC1	BFD0	BFC0				STNO
TPMR	—	PWMD1	PWMC1	PWMB1	—	PWMD0	PWMC0	PWMB0
TFCR	—	STCLK	ADEG	ADTRG	OLS1	OLS0	CMD1	CMD0
TOER	ED1	EC1	EB1	EA1	ED0	EC0	EB0	EA0
TOCR	TOD1	TOC1	TOB1	TOA1	TOD0	TOC0	TOB0	TOA0
RSECDR	BSY	SC12	SC11	SC10	SC03	SC02	SC01	SC00
RMINDR	BSY	MN12	MN11	MN10	MN03	MN02	MN01	MN00
RHRDR	BSY	—	HR11	HR10	HR03	HR02	HR01	HR00
RWKDR	BSY	—	—	—	—	WK2	WK1	WK0
RTCCR1	RUN	12/24	PM	RST	—	—	—	—
RTCCR2	—	—	FOIE	WKIE	DYIE	HRIE	MNIE	SEIE
RTCCSR	—	RCS6	RCS5	—	RCS3	RCS2	RCS1	RCS0
—	—	—	—	—	—	—	—	—
SMR_2	COM	CHR	PE	PM	STOP	MP	CKS1	CKS0
BRR_2	BRR7	BRR6	BRR5	BRR4	BRR3	BRR2	BRR1	BRR0
SCR3_2	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
TDR_2	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1	TDR0
SSR_2	TDRE	RDRF	OER	FER	PER	TEND	MPBR	MPBT
RDR_2	RDR7	RDR6	RDR5	RDR4	RDR3	RDR2	RDR1	RDR0
ICCR1	ICE	RCVD	MST	TRS	CKS3	CKS2	CKS1	CKS0
ICCR2	BBSY	SCP	SDAO	SDAOP	SCLO	—	IICRST	—
ICMR	MLS	WAIT	—	—	BCWP	BC2	BC1	BC0
ICIER	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT
ICSR	TDRE	TEND	RDRF	NACKF	STOP	AL/OVE	AAS	ADZ

FLMCR1	—	SWE	ESU	PSU	EV	PV	E	P	P
FLMCR2	FLER	—	—	—	—	—	—	—	
FLPWCR	PDWND	—	—	—	—	—	—	—	
EBR1	—	EB6	EB5	EB4	EB3	EB2	EB1	EB0	
FENR	FLSHE	—	—	—	—	—	—	—	
TCRV0	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	T
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	
—	—	—	—	—	—	—	—	—	—
SMR	COM	CHR	PE	PM	STOP	MP	CKS1	CKS0	S
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
	AD1	AD0	—	—	—	—	—	—	
ADDRB	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	

PWDRL	PWDRL7	PWDRL6	PWDRL5	PWDRL4	PWDRL3	PWDRL2	PWDRL1	PWDRL0
PWDRU	—	—	PWDRU5	PWDRU4	PWDRU3	PWDRU2	PWDRU1	PWDRU0
PWCR	—	—	—	—	—	—	—	PWCR0
TCSRWD	B6WI	TCWE	B4WI	TCSRWE	B2WI	WDON	B0WI	WRST
TCWD	TCWD7	TCWD6	TCWD5	TCWD4	TCWD3	TCWD2	TCWD1	TCWD0
TMWD	—	—	—	—	CKS3	CKS2	CKS1	CKS0
—	—	—	—	—	—	—	—	—
ABRKCR	RTINTE	CSEL1	CSEL0	ACMP2	ACMP1	ACMP0	DCMP1	DCMP0
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
PUCR5	—	—	PUCR55	PUCR54	PUCR53	PUCR52	PUCR51	PUCR50
PDR1	P17	P16	P15	P14	—	P12	P11	P10
PDR2	—	—	—	P24	P23	P22	P21	P20
PDR3	P37	P36	P35	P34	P33	P32	P31	P30
PDR5	P57	P56	P55	P54	P53	P52	P51	P50
PDR6	P67	P66	P65	P64	P63	P62	P61	P60
PDR7	—	P76	P75	P74	—	P72	P71	P70
PDR8	P87	P86	P85	—	—	—	—	—
PDRB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0

PCR7	—	PCR76	PCR75	PCR74	—	PCR72	PCR71	PCR70
PCR8	PCR87	PCR86	PCR85	—	—	—	—	—
SYSCR1	SSBY	STS2	STS1	STS0	NESEL	—	—	—
SYSCR2	SMSEL	LSON	DTON	MA2	MA1	MA0	SA1	SA0
IEGR1	NMIEG	—	—	—	IEG3	IEG2	IEG1	IEG0
IEGR2	—	—	WPEG5	WPEG4	WPEG3	WPEG2	WPEG1	WPEG0
IENR1	IENDT	IENTA	IENWP	—	IEN3	IEN2	IEN1	IEN0
IENR2	—	—	IENTB1	—	—	—	—	—
IRR1	IRRDT	IRRTA	—	—	IRRI3	IRRI2	IRRI1	IRRI0
IRR2	—	—	IRRTB1	—	—	—	—	—
IWPR	—	—	IWPF5	IWPF4	IWPF3	IWPF2	IWPF1	IWPF0
MSTCR1	—	MSTIIC	MSTS3	MSTAD	MSTWD	—	MSTTV	MSTTA
MSTCR2	MSTS3_2	—	—	MSTTB1	—	—	MSTTZ	MSTPWM
—	—	—	—	—	—	—	—	—

Note: * WDT: Watchdog timer

POCR_0	Initialized					
TCNT_0	Initialized	—	—	—	—	—
GRA_0	Initialized	—	—	—	—	—
GRB_0	Initialized	—	—	—	—	—
GRC_0	Initialized	—	—	—	—	—
GRD_0	Initialized	—	—	—	—	—
TCR_1	Initialized	—	—	—	—	—
TIORA_1	Initialized	—	—	—	—	—
TIORC_1	Initialized	—	—	—	—	—
TSR_1	Initialized	—	—	—	—	—
TIER_1	Initialized	—	—	—	—	—
POCR_1	Initialized	—	—	—	—	—
TCNT_1	Initialized	—	—	—	—	—
GRA_1	Initialized	—	—	—	—	—
GRB_1	Initialized	—	—	—	—	—
GRC_1	Initialized	—	—	—	—	—
GRD_1	Initialized	—	—	—	—	—
TSTR	Initialized	—	—	—	—	—
TMDR	Initialized	—	—	—	—	—
TPMR	Initialized	—	—	—	—	—
TFCR	Initialized	—	—	—	—	—
TOER	Initialized	—	—	—	—	—
TOCR	Initialized	—	—	—	—	—

SMR_2	Initialized	—	—	Initialized	Initialized	Initialized	SCI3_
BRR_2	Initialized	—	—	Initialized	Initialized	Initialized	
SCR3_2	Initialized	—	—	Initialized	Initialized	Initialized	
TDR_2	Initialized	—	—	Initialized	Initialized	Initialized	
SSR_2	Initialized	—	—	Initialized	Initialized	Initialized	
RDR_2	Initialized	—	—	Initialized	Initialized	Initialized	
ICCR1	Initialized	—	—	—	—	—	IIC2
ICCR2	Initialized	—	—	—	—	—	
ICMR	Initialized	—	—	—	—	—	
ICIER	Initialized	—	—	—	—	—	
ICSR	Initialized	—	—	—	—	—	
SAR	Initialized	—	—	—	—	—	
ICDRT	Initialized	—	—	—	—	—	
ICDRR	Initialized	—	—	—	—	—	
TMB1	Initialized	—	—	—	—	—	Timer
TCB1	Initialized	—	—	—	—	—	
TLB1	Initialized	—	—	—	—	—	
FLMCR1	Initialized	—	—	Initialized	Initialized	Initialized	ROM
FLMCR2	Initialized	—	—	—	—	—	
FLPWCR	Initialized	—	—	—	—	—	
EBR1	Initialized	—	—	Initialized	Initialized	Initialized	
FENR	Initialized	—	—	—	—	—	

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	
ADDRA	Initialized	—	—	Initialized	Initialized	Initialized	A/D
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	
PWDRL	Initialized	—	—	—	—	—	14bi
PWDRU	Initialized	—	—	—	—	—	
PWCR	Initialized	—	—	—	—	—	
TCSRWD	Initialized	—	—	—	—	—	WDT
TCWD	Initialized	—	—	—	—	—	
TMWD	Initialized	—	—	—	—	—	
ABRKCR	Initialized	—	—	—	—	—	Add
ABRKSR	Initialized	—	—	—	—	—	
BARH	Initialized	—	—	—	—	—	
BARL	Initialized	—	—	—	—	—	
BDRH	Initialized	—	—	—	—	—	
BDRL	Initialized	—	—	—	—	—	

PDR8	Initialized	—	—	—	—	—	
PDRB	Initialized	—	—	—	—	—	
PMR1	Initialized	—	—	—	—	—	
PMR5	Initialized	—	—	—	—	—	
PMR3	Initialized	—	—	—	—	—	
PCR1	Initialized	—	—	—	—	—	
PCR2	Initialized	—	—	—	—	—	
PCR3	Initialized	—	—	—	—	—	
PCR5	Initialized	—	—	—	—	—	
PCR6	Initialized	—	—	—	—	—	
PCR7	Initialized	—	—	—	—	—	
PCR8	Initialized	—	—	—	—	—	
SYSCR1	Initialized	—	—	—	—	—	Low p
SYSCR2	Initialized	—	—	—	—	—	
IEGR1	Initialized	—	—	—	—	—	Interr
IEGR2	Initialized	—	—	—	—	—	
IENR1	Initialized	—	—	—	—	—	
IENR2	Initialized	—	—	—	—	—	
IRR1	Initialized	—	—	—	—	—	
IRR2	Initialized	—	—	—	—	—	
IWPR	Initialized	—	—	—	—	—	
MSTCR1	Initialized	—	—	—	—	—	Low p
MSTCR2	Initialized	—	—	—	—	—	

Note: * WDT: Watchdog timer

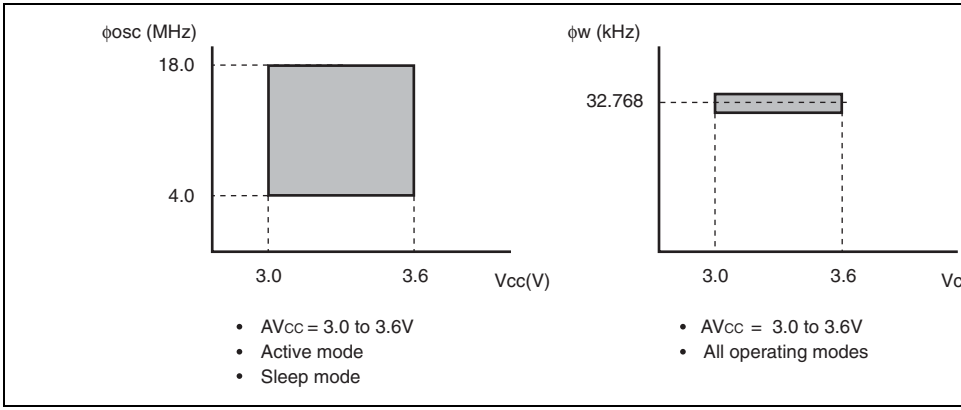
	and X1			
	Port B		–0.3 to AV_{CC} +0.3	V
	X1		–0.3 to 4.3	V
Operating temperature		T_{opr}	–20 to +75	°C
Storage temperature		T_{stg}	–55 to +125	°C

Note: * Permanent damage may result if maximum ratings are exceeded. Normal ope should be under the conditions specified in Electrical Characteristics. Exceed values can result in incorrect operation and reduced reliability.

20.2 Electrical Characteristics (F-ZTAT™ Version)

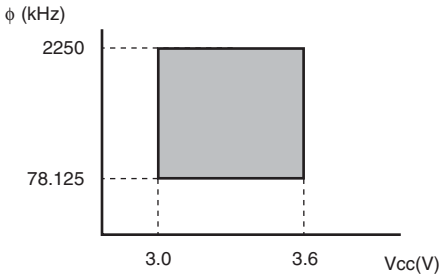
20.2.1 Power Supply Voltage and Operating Ranges

(1) Power Supply Voltage and Oscillation Frequency Range



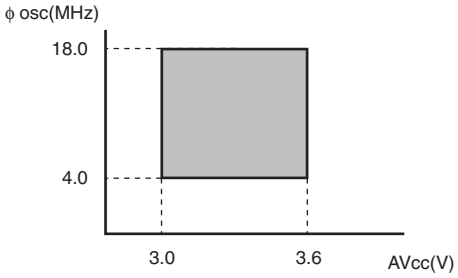
- AV_{CC} = 3.0 to 3.6V
- Active mode
- Sleep mode
  (When MA2 is SYSCR2 = 0)

- AV_{CC} = 3.0 to 3.6V
- Subactive mode
- Subsleep mode



- AV_{CC} = 3.0 to 3.6V
- Active mode
- Sleep mode
  (When MA2 is SYSCR2 = 1)

(3) Analog Power Supply Voltage and A/D Converter Accuracy Guarantee Range



- V_{CC} = 3.0 to 3.6V
- Active mode
- Sleep mode

		TMRIV, TMCIV, FTIOA0 to FTIOD0, FTIOA1 to FTIOD1,SCK3, SCK3_2, TRGV				
		RXD, RXD_2, SCL, SDA, P10 to P12, P14 to P17, P20 to P24, P30 to P37, P50 to P57, P60 to P67, P70 to P72 P74 to P76, P85 to P87	$V_{CC} \times 0.7$	—	$V_{CC} + 0.3$	V
		PB0 to PB7	$V_{CC} \times 0.7$	—	$AV_{CC} + 0.3$	V
		OSC1	$V_{CC} - 0.5$	—	$V_{CC} + 0.3$	V
Input low voltage	V_{IL}	$\overline{RES}$, $\overline{NMI}$, $\overline{WKP0}$ to $\overline{WKP5}$, $\overline{IRQ0}$ to $\overline{IRQ3}$, $\overline{ADTRG}$, TMIB1, TMRIV, TMCIV, FTIOA0 to FTIOD0, FTIOA1 to FTIOD1, SCK3, SCK3_2, TRGV	–0.3	—	$V_{CC} \times 0.2$	V

Note: Connect the TEST pin to Vss.

		PB0 to PB7		–0.3	—	$V_{CC} \times 0.3$	V
		OSC1		–0.3	—	0.5	V
Output high voltage	V_{OH}	P10 to P12, P14 to P17, P20 to P24, P30 to P37, P50 to P55, P60 to P67, P70 to P72, P74 to P76, P85 to P87,	$-I_{OH}$ = 2.0 mA	V_{CC} – 0.5	—	—	V
		P56, P57	$-I_{OH}$ = 0.1 mA	V_{CC} – 2.0	—	—	V
Output low voltage	V_{OL}	P10 to P12, P14 to P17, P20 to P24, P30 to P37, P50 to P57, P70 to P72, P74 to P76, P85 to P87	I_{OL} = 1.6 mA	—	—	0.6	V
			I_{OL} = 0.4 mA	—	—	0.4	
		P60 to P67	I_{OL} = 10.0 mA	—	—	1.0	V
			I_{OL} = 1.6 mA	—	—	0.4	
		SCL, SDA	I_{OL} = 6.0 mA	—	—	0.6	V
			I_{OL} = 3.0 mA	—	—	0.4	

		SDA					
		P10 to P12, P14 to P17, P20 to P24, P30 to P37, P50 to P57, P60 to P67, P70 to P72, P74 to P76, P85 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} = 3.3 V, V_{IN} = 0.0 V	33.0	—	165.0	µA
Pull-up register	R_p	$\overline{RES}$		60.0	150	—	kΩ
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 current consumption	I_{OPE1}	V_{CC}	Active mode 1 V_{CC} = 3.3 V, f_{OSC} = 18 MHz	—	21.0	28.0	mA
			Active mode 1 V_{CC} = 3.3 V, f_{OSC} = 10 MHz	—	11.6	—	
	I_{OPE2}	V_{CC}	Active mode 2 V_{CC} = 3.3 V, f_{OSC} = 18 MHz	—	1.4	2.8	mA
			Active mode 2 V_{CC} = 3.3 V, f_{OSC} = 10 MHz	—	1.2	—	

			V_{CC} = 3.3 V, f_{OSC} = 10 MHz				
Subactive mode current consumption	I_{SUB}	V_{CC}	V_{CC} = 3.3 V 32-kHz crystal resonator ($\phi_{SUB} = \phi_W/2$)	—	35.0	60.0	µA
			V_{CC} = 3.3 V 32-kHz crystal resonator ($\phi_{SUB} = \phi_W/8$)	—	20.0	—	
Subsleep mode current consumption	I_{SUBSP1}	V_{CC}	Subsleep mode 1 V_{CC} = 3.3 V 32-kHz crystal resonator ($\phi_{SUB} = \phi_W/2$)	—	20.0	40.0	µA
	I_{SUBSP2}	V_{CC}	Subsleep mode 2 V_{CC} = 3.3 V 32-kHz crystal resonator not used	—	—	5.0	µA
Standby mode current consumption	I_{STBY}	V_{CC}	32-kHz crystal resonator not used	—	—	5.0	µA
RAM data retaining voltage	V_{RAM}	V_{CC}		2.0	—	—	V

				ceramic or cryst
Subsleep mode 1	V_{CC}	Only timers operate	V_{CC}	Subclock: crystal resonato
Subsleep mode 2		CPU and timers both stop		Main clock: ceramic or cryst Subclock: Pin X1 = V_{SS}
Standby mode	V_{CC}	CPU and timers both stop	V_{CC}	Main clock: ceramic or cryst Subclock: Pin X1 = V_{SS}

Allowable output low current (total)	$\sum I_{OL}$	Output pins except port 6, SCL, and SDA	—	—	20.0
		Port 6, SCL, and SDA	—	—	40.0
Allowable output high current (per pin)	$\lvert -I_{OH} \rvert$	All output pins	—	—	2.0
Allowable output high current (total)	$\lvert -\sum I_{OH} \rvert$	All output pins	—	—	20.0

System clock (ϕ) cycle time	t_{cyc}		1	—	64	t_{OSC}	F
			—	—	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	*
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	25.0	—	—	ns	F
External clock low width	t_{CPL}	OSC1	25.0	—	—	ns	
External clock rise time	t_{CPr}	OSC1	—	—	10.0	ns	
External clock fall time	t_{CPf}	OSC1	—	—	10.0	ns	

		WKP5, TMCIV, TMRIV, TRGV, $\overline{\text{ADTRG}}$, FTIOA0 to FTIOD0, FTIOA1 to FTIOD1				
Input pin low width	t_{IL}	$\overline{\text{NMI}}$, TMIB1, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ3}}$, $\overline{\text{WKP0}}$ to $\overline{\text{WKP5}}$, TMCIV, TMRIV, TRGV, $\overline{\text{ADTRG}}$, FTIOA0 to FTIOD0, FTIOA1 to FTIOD1	2	—	—	t_{cyc} t_{subcyc}

Notes: * Determined by MA2, MA1, MA0, SA1, and SA0 of system control register 2 (S

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}	—	—	300	ns

Input clock pulse width	t_{SCKW}	SCK3	0.4	—	0.6	t_{Scyc}	
Transmit data delay time (clocked synchronous)	t_{TXD}	TXD	—	—	1	t_{cyc}	Fi
Receive data setup time (clocked synchronous)	t_{RXS}	RXD	55.5	—	—	ns	
Receive data hold time (clocked synchronous)	t_{RXH}	RXD	55.5	—	—	ns	

Analog power supply current	AI_{OPE}	AV_{CC}	AV_{CC} = 3.3 V f_{OSC} = 18 MHz	—	—	2.0	mA
	AI_{STOP1}	AV_{CC}		—	50	—	µA
	AI_{STOP2}	AV_{CC}		—	—	5.0	µA
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 to 3.6 V	134	—	—	t_{cyc}
Nonlinearity error				—	—	±5.5	LSB
Offset error				—	—	±5.5	LSB
Full-scale error				—	—	±5.5	LSB
Quantization error				—	—	±0.5	LSB
Absolute accuracy				—	—	±6.0	LSB
Conversion time (single mode)			AV_{CC} = 3.0 to 3.6 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			Unit	R F
				Min.	Typ.	Max.		
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 gene
when the internal oscillator is selected.

Programming	Wait time after SWE bit setting*[1]	x		1	—	—
	Wait time after PSU bit setting*[1]	y		50	—	—
	Wait time after P bit setting *[1]*[4]	z1	1 ≤ n ≤ 6	28	30	32
		z2	7 ≤ n ≤ 1000	198	200	202
		z3	Additional-programming	8	10	12
	Wait time after P bit clear*[1]	α		5	—	—
	Wait time after PSU bit clear*[1]	β		5	—	—
	Wait time after PV bit setting*[1]	γ		4	—	—
	Wait time after dummy write*[1]	ε		2	—	—
	Wait time after PV bit clear*[1]	η		2	—	—
	Wait time after SWE bit clear*[1]	θ		100	—	—
	Maximum programming count *[1]*[4]*[5]	N		—	—	1000

Wait time after EV bit setting*[1]	γ	20	—	—
Wait time after dummy write*[1]	ε	2	—	—
Wait time after EV bit clear*[1]	η	4	—	—
Wait time after SWE bit clear*[1]	θ	100	—	—
Maximum erase count *[1]*[6]*[7]	N	—	—	120

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 b memory control register 1 (FLMCR1) is set. The program-verify time is not incl

3. The time required to erase one block. (Indicates the time for which the E bit in memory control register 1 (FLMCR1) is set. The erase-verify time is not include

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 and z3, so that it does not exceed the programming time maximum value (t_P(m The wait time after P bit setting (z1, z2) should be changed as follows accordin value of the programming count (n).

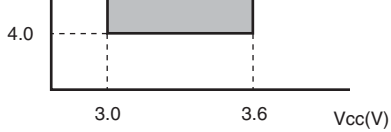
   Programming count (n)

   $1 \leq n \leq 6$ z1 = 30 µs

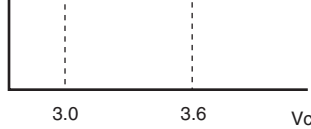
   $7 \leq n \leq 1000$ z2 = 200 µs

6. Erase time maximum value (t_E(max.)) = wait time after E bit setting (z) × maxim erase count (N)

7. Set the maximum erase count (N) according to the actual set value of (z), so th does not exceed the erase time maximum value (t_E(max.)).



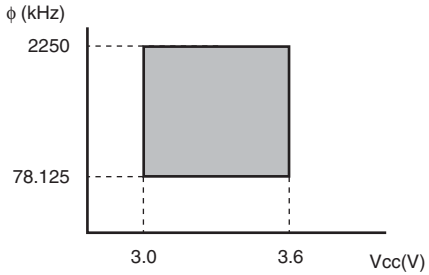
- AV_{CC} = 3.0 to 3.6V
- Active mode
- Sleep mode



- AV_{CC} = 3.0 to 3.6V
- All operating modes

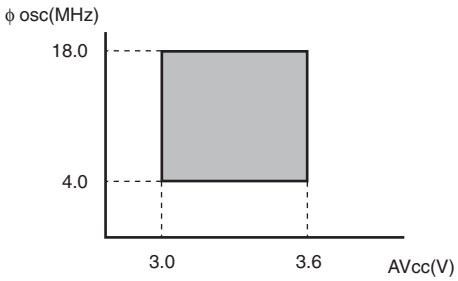
- AV_{CC} = 3.0 to 3.6V
- Active mode
- Sleep mode
  (When MA2 in SYSCR2 = 0)

- AV_{CC} = 3.0 to 3.6V
- Subactive mode
- Subsleep mode



- AV_{CC} = 3.0 to 3.6V
- Active mode
- Sleep mode
  (When MA2 in SYSCR2 = 1)

(3) Analog Power Supply Voltage and A/D Converter Accuracy Guarantee Range



- V_{CC} = 3.0 to 3.6V
- Active mode
- Sleep mode

	TMRIV, TMCIV, FTIOA0 to FTIOD0, FTIOA1 to FTIOD1, SCK3, SCK3_2, TRGV				
	RXD, RXD_2 SCL, SDA, P10 to P12, P14 to P17, P20 to P24, P30 to P37 P50 to P57, P60 to P67, P70 to P72, P74 to P76, P85 to P87	$V_{cc} \times 0.7$	—	$V_{cc} + 0.3$	V
	PB0 to PB7	$V_{cc} \times 0.7$	—	$AV_{cc} + 0.3$	V

Note: Connect the TEST pin to Vss.

		FTIOD1, SCK3, SCK3_2, TRGV					
		RXD, RXD_2, SCL, SDA, P10 to P12, P14 to P17, P20 to P24, P30 to P37, P50 to P57, P60 to P67,. P70 to P72, P74 to P76, P85 to P87		–0.3	—	$V_{CC} \times 0.3$	V
		PB0 to PB7		–0.3	—	$V_{CC} \times 0.3$	V
		OSC1		–0.3	—	0.5	V
Output high voltage	V_{OH}	P10 to P12, P14 to P17, P20 to P24, P30 to P37, P50 to P55, P60 to P67, P70 to P72, P74 to P76, P85 to P87	$-I_{OH}$ = 2.0 mA	V_{CC} – 0.5	—	—	V
		P56, P57	$-I_{OH}$ = 0.1 mA	V_{CC} – 2.0	—	—	V

		SCL, SDA	I_{OL} = 6.0 mA	—	—	0.6	V
			I_{OL} = 3.0 mA	—	—	0.4	
Input/ output leakage current	$\lvert I_{IL} \rvert$	OSC1, TMIB1, $\overline{\text{NMI}}$, $\overline{\text{WKP0}}$ to $\overline{\text{WKP5}}$, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ3}}$, $\overline{\text{ADTRG}}$, TRGV, TMRIV, TMCIV, FTIOA0 to FTIOD0, FTIOA1 to FTIOD1, RXD, SCK3, RXD_2, SCK3_2, SCL, SDA	V_{IN} = 0.5 V to (V_{CC} – 0.5 V)	—	—	1.0	µA
		P10 to P12, P14 to P17, P20 to P24, P30 to P37, P50 to P57, P60 to P67, P70 to P72, P74 to P76, P85 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 P55	V_{CC} = 3.3 V, V_{IN} = 0.0 V	33.0	—	165.0	µA
Pull-up resistor	R_P	$\overline{\text{RES}}$		60.0	150.0	—	kΩ
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 2 V_{CC} = 3.3 V, f_{OSC} = 10 MHz		1.2			
Sleep mode current consumption	I_{SLEEP1}	V_{CC}	Sleep mode 1 V_{CC} = 3.3 V, f_{OSC} = 18 MHz	—	16.5	21.0	mA	*
			Sleep mode 1 V_{CC} = 3.3 V, f_{OSC} = 10 MHz	—	9.0	—		*
	I_{SLEEP2}	V_{CC}	Sleep mode 2 V_{CC} = 3.3 V, f_{OSC} = 18 MHz	—	1.3	2.5	mA	*
			Sleep mode 2 V_{CC} = 3.3V, f_{OSC} = 10 MHz	—	1.1	—		*
Subactive mode current consumption	I_{SUB}	V_{CC}	V_{CC} = 3.3 V 32-kHz crystal resonator ($\phi_{SUB} = \phi_W/2$)	—	35.0	60.0	µA	*
			V_{CC} = 3.3 V 32-kHz crystal resonator ($\phi_{SUB} = \phi_W/8$)	—	20.0	—		*
Subsleep mode current consumption	I_{SUBSP1}	V_{CC}	Subsleep mode 1 V_{CC} = 3.3 V 32-kHz crystal resonator ($\phi_{SUB} = \phi_W/2$)	—	20.0	40.0	µA	*
	I_{SUBSP2}	V_{CC}	Subsleep mode 2 V_{CC} = 3.3 V 32-kHz crystal resonator not used	—	—	5.0	µA	*

Mode	$\overline{RES}$ Pin	Internal State	Other Pins	Oscillator Pi
Active mode 1	V_{CC}	Operates	V_{CC}	Main clock: ceramic or cry resonator
Active mode 2		Operates (ϕOSC/64)		Subclock: Pin X1 = V_{SS}
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 cry resonator
Subsleep mode 1	V_{CC}	Only timers operate	V_{CC}	Subclock reso crystal
Subsleep mode 2		CPU and timers both stop		Main clock: ceramic or cry resonator
				Subclock: Pin X1 = V_{SS}
Standby mode	V_{CC}	CPU and timers both stop	V_{CC}	Main clock: ceramic or cry resonator
				Subclock: Pin X1 = V_{SS}

Allowable output low current (total)	$\sum I_{OL}$	Output pins except port 6, SCL, and SDA	—	—	20.0
		Port 6, SCL, and SDA	—	—	40.0
Allowable output high current (per pin)	$\lvert -I_{OH} \rvert$	All output pins	—	—	2.0
Allowable output high current (total)	$\lvert -\sum I_{OH} \rvert$	All output pins	—	—	20.0

System clock (ϕ) cycle time	t_{cyc}		1	—	64	t_{OSC}	Fi
			—	—	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	*
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	25.0	—	—	ns	Fi
External clock low width	t_{CPL}	OSC1	25.0	—	—	ns	
External clock rise time	t_{CPr}	OSC1	—	—	10.0	ns	
External clock fall time	t_{CPf}	OSC1	—	—	10.0	ns	

		TMCIV, TMRIV, TRGV, $\overline{\text{ADTRG}}$, FTIOA0 to FTIOD0, FTIOA1 to FTIOD1				
Input pin low width	t_{IL}	$\overline{\text{NMI}}$, TMIB1, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ3}}$, $\overline{\text{WKP0}}$ to $\overline{\text{WKP5}}$, TMCIV, TMRIV, TRGV, $\overline{\text{ADTRG}}$, FTIOA0 to FTIOD0, FTIOA1 to FTIOD1	2	—	—	t_{cyc} t_{subcyc}

Notes: * Determined by MA2, MA1, MA0, SA1, and SA0 of system control register 2 (S

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}	—	—	300	ns

Transmit data delay time (clocked sycronous)	t_{TXD}	TXD	—	—	1	t_{cyc}	Fi
Receive data setup time (clocked synchronous)	t_{RXS}	RXD	55.5	—	—	ns	
Receive data hold time (clocked synchronous)	t_{RXH}	RXD	55.5	—	—	ns	

Analog power supply current	AI_{OPE}	AV_{CC}	AV_{CC} = 3.3 V f_{OSC} = 18 MHz	—	—	2.0	mA
	AI_{STOP1}	AV_{CC}		—	50	—	µA
	AI_{STOP2}	AV_{CC}		—	—	5.0	µA
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 to 3.6 V	134	—	—	t_{cyc}
Nonlinearity error				—	—	±5.5	LSB
Offset error				—	—	±5.5	LSB
Full-scale error				—	—	±5.5	LSB
Quantization error				—	—	±0.5	LSB
Absolute accuracy				—	—	±6.0	LSB
Conversion time (single mode)			AV_{CC} = 3.0 to 3.6 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			Unit	R F
				Min.	Typ.	Max.		
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 gene when the internal oscillator is selected.

Figure 20.1 System Clock Input Timing

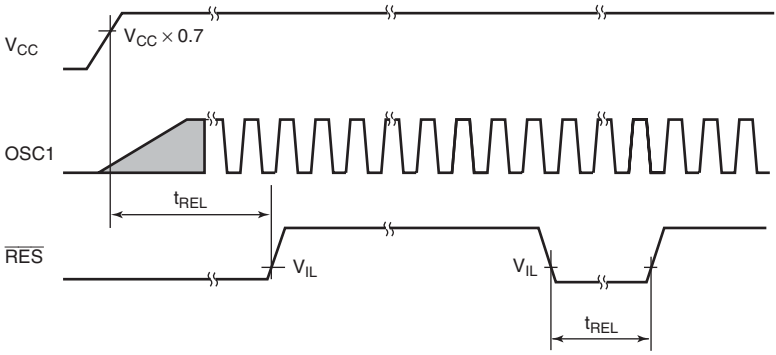


Figure 20.2 $\overline{\text{RES}}$ Low Width Timing

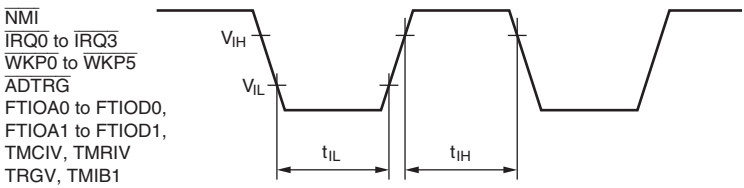


Figure 20.3 Input Timing

Note: * S, P, and Sr represent the following:
S: Start condition
P: Stop condition
Sr: Retransmission start condition

Figure 20.4 I²C Bus Interface Input/Output Timing

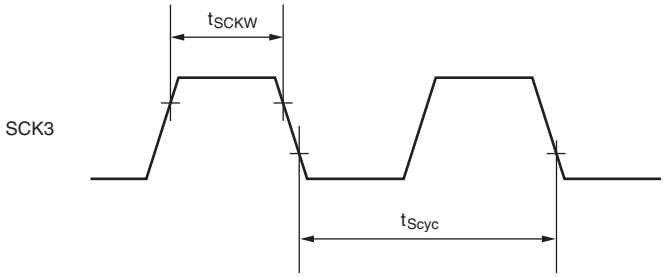
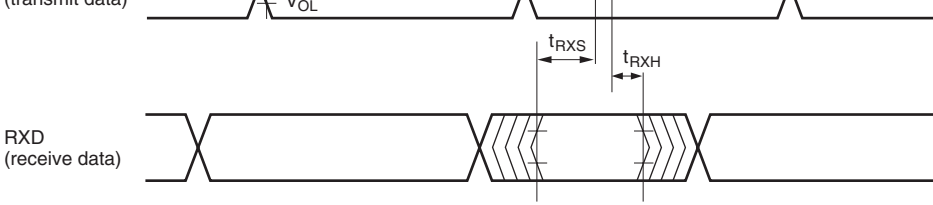


Figure 20.5 SCK3 Input Clock Timing



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 20.7.

Figure 20.6 SCI Input/Output Timing in Clocked Synchronous Mode

Figure 20.7 Output Load Circuit

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 transi 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
⊕	Logical exclusive OR of the operands on both sides

0	Cleared to 0
1	Set to 1
—	Not affected by execution of the instruction
Δ	Varies depending on conditions, described in notes

	MOV.B @ERs, Rd	B			2						@ERs → Rd8	—	—	↕	↕	0
	MOV.B @(d:16, ERs), Rd	B				4					@(d:16, ERs) → Rd8	—	—	↕	↕	0
	MOV.B @(d:24, ERs), Rd	B				8					@(d:24, ERs) → Rd8	—	—	↕	↕	0
	MOV.B @ERs+, Rd	B					2				@ERs → Rd8 ERs32+1 → ERs32	—	—	↕	↕	0
	MOV.B @aa:8, Rd	B						2			@aa:8 → Rd8	—	—	↕	↕	0
	MOV.B @aa:16, Rd	B						4			@aa:16 → Rd8	—	—	↕	↕	0
	MOV.B @aa:24, Rd	B						6			@aa:24 → Rd8	—	—	↕	↕	0
	MOV.B Rs, @ERd	B			2						Rs8 → @ERd	—	—	↕	↕	0
	MOV.B Rs, @(d:16, ERd)	B				4					Rs8 → @(d:16, ERd)	—	—	↕	↕	0
	MOV.B Rs, @(d:24, ERd)	B				8					Rs8 → @(d:24, ERd)	—	—	↕	↕	0
	MOV.B Rs, @–ERd	B					2				ERd32–1 → ERd32 Rs8 → @ERd	—	—	↕	↕	0
	MOV.B Rs, @aa:8	B						2			Rs8 → @aa:8	—	—	↕	↕	0
	MOV.B Rs, @aa:16	B						4			Rs8 → @aa:16	—	—	↕	↕	0
	MOV.B Rs, @aa:24	B						6			Rs8 → @aa:24	—	—	↕	↕	0
	MOV.W #xx:16, Rd	W	4								#xx:16 → Rd16	—	—	↕	↕	0
	MOV.W Rs, Rd	W		2							Rs16 → Rd16	—	—	↕	↕	0
	MOV.W @ERs, Rd	W			2						@ERs → Rd16	—	—	↕	↕	0
	MOV.W @(d:16, ERs), Rd	W				4					@(d:16, ERs) → Rd16	—	—	↕	↕	0
	MOV.W @(d:24, ERs), Rd	W				8					@(d:24, ERs) → Rd16	—	—	↕	↕	0
	MOV.W @ERs+, Rd	W					2				@ERs → Rd16 ERs32+2 → @ERd32	—	—	↕	↕	0
	MOV.W @aa:16, Rd	W						4			@aa:16 → Rd16	—	—	↕	↕	0
	MOV.W @aa:24, Rd	W						6			@aa:24 → Rd16	—	—	↕	↕	0
	MOV.W Rs, @ERd	W			2						Rs16 → @ERd	—	—	↕	↕	0
	MOV.W Rs, @(d:16, ERd)	W				4					Rs16 → @(d:16, ERd)	—	—	↕	↕	0
	MOV.W Rs, @(d:24, ERd)	W				8					Rs16 → @(d:24, ERd)	—	—	↕	↕	0

	MOV.L ERs, ERd	L		2								MOV.L ERs → ERd32	—	—	↕	↕	0	-
	MOV.L @ERs, ERd	L			4							@ERs → ERd32	—	—	↕	↕	0	-
	MOV.L @(d:16, ERs), ERd	L				6						@(d:16, ERs) → ERd32	—	—	↕	↕	0	-
	MOV.L @(d:24, ERs), ERd	L				10						@(d:24, ERs) → ERd32	—	—	↕	↕	0	-
	MOV.L @ERs+, ERd	L					4					@ERs → ERd32 ERs32+4 → ERs32	—	—	↕	↕	0	-
	MOV.L @aa:16, ERd	L						6				@aa:16 → ERd32	—	—	↕	↕	0	-
	MOV.L @aa:24, ERd	L						8				@aa:24 → ERd32	—	—	↕	↕	0	-
	MOV.L ERs, @ERd	L			4							ERs32 → @ERd	—	—	↕	↕	0	-
	MOV.L ERs, @(d:16, ERd)	L				6						ERs32 → @(d:16, ERd)	—	—	↕	↕	0	-
	MOV.L ERs, @(d:24, ERd)	L				10						ERs32 → @(d:24, ERd)	—	—	↕	↕	0	-
	MOV.L ERs, @–ERd	L					4					ERd32–4 → ERd32 ERs32 → @ERd	—	—	↕	↕	0	-
	MOV.L ERs, @aa:16	L						6				ERs32 → @aa:16	—	—	↕	↕	0	-
	MOV.L ERs, @aa:24	L						8				ERs32 → @aa:24	—	—	↕	↕	0	-
POP	POP.W Rn	W									2	@SP → Rn16 SP+2 → SP	—	—	↕	↕	0	-
	POP.L ERn	L									4	@SP → ERn32 SP+4 → SP	—	—	↕	↕	0	-
PUSH	PUSH.W Rn	W									2	SP–2 → SP Rn16 → @SP	—	—	↕	↕	0	-
	PUSH.L ERn	L									4	SP–4 → SP ERn32 → @SP	—	—	↕	↕	0	-
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					

	ADD.L #xx:32, ERd	L	6									ERd32+#xx:32 → ERd32	—	(2)	↕	↕	↕
	ADD.L ERs, ERd	L		2								ERd32+ERs32 → ERd32	—	(2)	↕	↕	↕
ADDX	ADDX.B #xx:8, Rd	B	2									Rd8+#xx:8 +C → Rd8	—	↕	↕	(3)	↕
	ADDX.B Rs, Rd	B		2								Rd8+Rs8 +C → Rd8	—	↕	↕	(3)	↕
ADDS	ADDS.L #1, ERd	L		2								ERd32+1 → ERd32	—	—	—	—	—
	ADDS.L #2, ERd	L		2								ERd32+2 → ERd32	—	—	—	—	—
	ADDS.L #4, ERd	L		2								ERd32+4 → ERd32	—	—	—	—	—
INC	INC.B Rd	B		2								Rd8+1 → Rd8	—	—	↕	↕	↕
	INC.W #1, Rd	W		2								Rd16+1 → Rd16	—	—	↕	↕	↕
	INC.W #2, Rd	W		2								Rd16+2 → Rd16	—	—	↕	↕	↕
	INC.L #1, ERd	L		2								ERd32+1 → ERd32	—	—	↕	↕	↕
	INC.L #2, ERd	L		2								ERd32+2 → ERd32	—	—	↕	↕	↕
DAA	DAA Rd	B		2								Rd8 decimal adjust → Rd8	—	*	↕	↕	*
SUB	SUB.B Rs, Rd	B		2								Rd8–Rs8 → Rd8	—	↕	↕	↕	↕
	SUB.W #xx:16, Rd	W	4									Rd16–#xx:16 → Rd16	—	(1)	↕	↕	↕
	SUB.W Rs, Rd	W		2								Rd16–Rs16 → Rd16	—	(1)	↕	↕	↕
	SUB.L #xx:32, ERd	L	6									ERd32–#xx:32 → ERd32	—	(2)	↕	↕	↕
	SUB.L ERs, ERd	L		2								ERd32–ERs32 → ERd32	—	(2)	↕	↕	↕
SUBX	SUBX.B #xx:8, Rd	B	2									Rd8–#xx:8–C → Rd8	—	↕	↕	(3)	↕
	SUBX.B Rs, Rd	B		2								Rd8–Rs8–C → Rd8	—	↕	↕	(3)	↕
SUBS	SUBS.L #1, ERd	L		2								ERd32–1 → ERd32	—	—	—	—	—
	SUBS.L #2, ERd	L		2								ERd32–2 → ERd32	—	—	—	—	—
	SUBS.L #4, ERd	L		2								ERd32–4 → ERd32	—	—	—	—	—
DEC	DEC.B Rd	B		2								Rd8–1 → Rd8	—	—	↕	↕	↕
	DEC.W #1, Rd	W		2								Rd16–1 → Rd16	—	—	↕	↕	↕
	DEC.W #2, Rd	W		2								Rd16–2 → Rd16	—	—	↕	↕	↕

	MULXU. W Rs, ERd	W		2								Rd16 × Rs16 → ERd32 (unsigned multiplication)	—	—	—	—	—	
MULXS	MULXS. B Rs, Rd	B		4								Rd8 × Rs8 → Rd16 (signed multiplication)	—	—	↕	↕	—	
	MULXS. W Rs, ERd	W		4								Rd16 × Rs16 → ERd32 (signed multiplication)	—	—	↕	↕	—	
DIVXU	DIVXU. B Rs, Rd	B		2								Rd16 ÷ Rs8 → Rd16 (RdH: remainder, RdL: quotient) (unsigned division)	—	—	(6)	(7)	—	
	DIVXU. W Rs, ERd	W		2								ERd32 ÷ Rs16 → ERd32 (Ed: remainder, Rd: quotient) (unsigned division)	—	—	(6)	(7)	—	
DIVXS	DIVXS. B Rs, Rd	B		4								Rd16 ÷ Rs8 → Rd16 (RdH: remainder, RdL: quotient) (signed division)	—	—	(8)	(7)	—	
	DIVXS. W Rs, ERd	W		4								ERd32 ÷ Rs16 → ERd32 (Ed: remainder, Rd: quotient) (signed division)	—	—	(8)	(7)	—	
CMP	CMP.B #xx:8, Rd	B	2									Rd8–#xx:8	—	↕	↕	↕	↕	
	CMP.B Rs, Rd	B		2								Rd8–Rs8	—	↕	↕	↕	↕	
	CMP.W #xx:16, Rd	W	4									Rd16–#xx:16	—	(1)	↕	↕	↕	
	CMP.W Rs, Rd	W		2								Rd16–Rs16	—	(1)	↕	↕	↕	
	CMP.L #xx:32, ERd	L	6									ERd32–#xx:32	—	(2)	↕	↕	↕	
	CMP.L ERs, ERd	L		2								ERd32–ERs32	—	(2)	↕	↕	↕	

											of ERd32)						
EXTS	EXTS.W Rd	W		2							(<bit 7> of Rd16) → (<bits 15 to 8> of Rd16)	—	—	↕	↕	0	
	EXTS.L ERd	L		2							(<bit 15> of ERd32) → (<bits 31 to 16> of ERd32)	—	—	↕	↕	0	

	AND.L #xx:32, ERd	L	6									ERd32∧#xx:32 → ERd32	—	—	↕	↕	0	-
	AND.L ERs, ERd	L		4								ERd32∧ERs32 → ERd32	—	—	↕	↕	0	-
OR	OR.B #xx:8, Rd	B	2									Rd8∨#xx:8 → Rd8	—	—	↕	↕	0	-
	OR.B Rs, Rd	B		2								Rd8∨Rs8 → Rd8	—	—	↕	↕	0	-
	OR.W #xx:16, Rd	W	4									Rd16∨#xx:16 → Rd16	—	—	↕	↕	0	-
	OR.W Rs, Rd	W		2								Rd16∨Rs16 → Rd16	—	—	↕	↕	0	-
	OR.L #xx:32, ERd	L	6									ERd32∨#xx:32 → ERd32	—	—	↕	↕	0	-
	OR.L ERs, ERd	L		4								ERd32∨ERs32 → ERd32	—	—	↕	↕	0	-
XOR	XOR.B #xx:8, Rd	B	2									Rd8⊕#xx:8 → Rd8	—	—	↕	↕	0	-
	XOR.B Rs, Rd	B		2								Rd8⊕Rs8 → Rd8	—	—	↕	↕	0	-
	XOR.W #xx:16, Rd	W	4									Rd16⊕#xx:16 → Rd16	—	—	↕	↕	0	-
	XOR.W Rs, Rd	W		2								Rd16⊕Rs16 → Rd16	—	—	↕	↕	0	-
	XOR.L #xx:32, ERd	L	6									ERd32⊕#xx:32 → ERd32	—	—	↕	↕	0	-
	XOR.L ERs, ERd	L		4								ERd32⊕ERs32 → ERd32	—	—	↕	↕	0	-
NOT	NOT.B Rd	B		2								¬ Rd8 → Rd8	—	—	↕	↕	0	-
	NOT.W Rd	W		2								¬ Rd16 → Rd16	—	—	↕	↕	0	-
	NOT.L ERd	L		2								¬ Rd32 → Rd32	—	—	↕	↕	0	-

	SHAR.W Rd	W		2								—	—	↕	↕	0
	SHAR.L ERd	L		2								—	—	↕	↕	0
SHLL	SHLL.B Rd	B		2								—	—	↕	↕	0
	SHLL.W Rd	W		2								—	—	↕	↕	0
	SHLL.L ERd	L		2								—	—	↕	↕	0
SHLR	SHLR.B Rd	B		2								—	—	↕	↕	0
	SHLR.W Rd	W		2								—	—	↕	↕	0
	SHLR.L ERd	L		2								—	—	↕	↕	0
ROTXL	ROTXL.B Rd	B		2								—	—	↕	↕	0
	ROTXL.W Rd	W		2								—	—	↕	↕	0
	ROTXL.L ERd	L		2								—	—	↕	↕	0
ROTXR	ROTXR.B Rd	B		2								—	—	↕	↕	0
	ROTXR.W Rd	W		2								—	—	↕	↕	0
	ROTXR.L ERd	L		2								—	—	↕	↕	0
ROTL	ROTL.B Rd	B		2								—	—	↕	↕	0
	ROTL.W Rd	W		2								—	—	↕	↕	0
	ROTL.L ERd	L		2								—	—	↕	↕	0
ROTR	ROTR.B Rd	B		2								—	—	↕	↕	0
	ROTR.W Rd	W		2								—	—	↕	↕	0
	ROTR.L ERd	L		2								—	—	↕	↕	0

	BSET Rn, @ERd	B			4							(Rn8 of @ERd) ← 1	—	—	—	—	—	
	BSET Rn, @aa:8	B						4				(Rn8 of @aa:8) ← 1	—	—	—	—	—	
BCLR	BCLR #xx:3, Rd	B		2								(#xx:3 of Rd8) ← 0	—	—	—	—	—	
	BCLR #xx:3, @ERd	B			4							(#xx:3 of @ERd) ← 0	—	—	—	—	—	
	BCLR #xx:3, @aa:8	B						4				(#xx:3 of @aa:8) ← 0	—	—	—	—	—	
	BCLR Rn, Rd	B		2								(Rn8 of Rd8) ← 0	—	—	—	—	—	
	BCLR Rn, @ERd	B			4							(Rn8 of @ERd) ← 0	—	—	—	—	—	
	BCLR Rn, @aa:8	B						4				(Rn8 of @aa:8) ← 0	—	—	—	—	—	
BNOT	BNOT #xx:3, Rd	B		2								(#xx:3 of Rd8) ← ¬ (#xx:3 of Rd8)	—	—	—	—	—	
	BNOT #xx:3, @ERd	B			4							(#xx:3 of @ERd) ← ¬ (#xx:3 of @ERd)	—	—	—	—	—	
	BNOT #xx:3, @aa:8	B						4				(#xx:3 of @aa:8) ← ¬ (#xx:3 of @aa:8)	—	—	—	—	—	
	BNOT Rn, Rd	B		2								(Rn8 of Rd8) ← ¬ (Rn8 of Rd8)	—	—	—	—	—	
	BNOT Rn, @ERd	B			4							(Rn8 of @ERd) ← ¬ (Rn8 of @ERd)	—	—	—	—	—	
	BNOT Rn, @aa:8	B						4				(Rn8 of @aa:8) ← ¬ (Rn8 of @aa:8)	—	—	—	—	—	
BTST	BTST #xx:3, Rd	B		2								¬ (#xx:3 of Rd8) → Z	—	—	—	↕	—	
	BTST #xx:3, @ERd	B			4							¬ (#xx:3 of @ERd) → Z	—	—	—	↕	—	
	BTST #xx:3, @aa:8	B						4				¬ (#xx:3 of @aa:8) → Z	—	—	—	↕	—	
	BTST Rn, Rd	B		2								¬ (Rn8 of @Rd8) → Z	—	—	—	↕	—	
	BTST Rn, @ERd	B			4							¬ (Rn8 of @ERd) → Z	—	—	—	↕	—	
	BTST Rn, @aa:8	B						4				¬ (Rn8 of @aa:8) → Z	—	—	—	↕	—	
BLD	BLD #xx:3, Rd	B		2								(#xx:3 of Rd8) → C	—	—	—	—	—	

BST	BST #xx:3, @ERd	B			4							C → (#xx:3 of @ERd24)	—	—	—	—	—
	BST #xx:3, @aa:8	B						4				C → (#xx:3 of @aa:8)	—	—	—	—	—
BIST	BIST #xx:3, Rd	B		2								¬ C → (#xx:3 of Rd8)	—	—	—	—	—
	BIST #xx:3, @ERd	B			4							¬ C → (#xx:3 of @ERd24)	—	—	—	—	—
	BIST #xx:3, @aa:8	B						4				¬ C → (#xx:3 of @aa:8)	—	—	—	—	—
BAND	BAND #xx:3, Rd	B		2								C∧(#xx:3 of Rd8) → C	—	—	—	—	—
	BAND #xx:3, @ERd	B			4							C∧(#xx:3 of @ERd24) → C	—	—	—	—	—
	BAND #xx:3, @aa:8	B						4				C∧(#xx:3 of @aa:8) → C	—	—	—	—	—
BIAND	BIAND #xx:3, Rd	B		2								C∧ ¬ (#xx:3 of Rd8) → C	—	—	—	—	—
	BIAND #xx:3, @ERd	B			4							C∧ ¬ (#xx:3 of @ERd24) → C	—	—	—	—	—
	BIAND #xx:3, @aa:8	B						4				C∧ ¬ (#xx:3 of @aa:8) → C	—	—	—	—	—
BOR	BOR #xx:3, Rd	B		2								C∨(#xx:3 of Rd8) → C	—	—	—	—	—
	BOR #xx:3, @ERd	B			4							C∨(#xx:3 of @ERd24) → C	—	—	—	—	—
	BOR #xx:3, @aa:8	B						4				C∨(#xx:3 of @aa:8) → C	—	—	—	—	—
BIOR	BIOR #xx:3, Rd	B		2								C∨ ¬ (#xx:3 of Rd8) → C	—	—	—	—	—
	BIOR #xx:3, @ERd	B			4							C∨ ¬ (#xx:3 of @ERd24) → C	—	—	—	—	—
	BIOR #xx:3, @aa:8	B						4				C∨ ¬ (#xx:3 of @aa:8) → C	—	—	—	—	—
BXOR	BXOR #xx:3, Rd	B		2								C⊕(#xx:3 of Rd8) → C	—	—	—	—	—
	BXOR #xx:3, @ERd	B			4							C⊕(#xx:3 of @ERd24) → C	—	—	—	—	—
	BXOR #xx:3, @aa:8	B						4				C⊕(#xx:3 of @aa:8) → C	—	—	—	—	—
BIXOR	BIXOR #xx:3, Rd	B		2								C⊕ ¬ (#xx:3 of Rd8) → C	—	—	—	—	—
	BIXOR #xx:3, @ERd	B			4							C⊕ ¬ (#xx:3 of @ERd24) → C	—	—	—	—	—
	BIXOR #xx:3, @aa:8	B						4				C⊕ ¬ (#xx:3 of @aa:8) → C	—	—	—	—	—

	BHI d:8	—						2			C∨ Z = 0	—	—	—	—	—				
	BHI d:16	—						4				—	—	—	—	—				
	BLS d:8	—						2			C∨ Z = 1	—	—	—	—	—				
	BLS d:16	—						4				—	—	—	—	—				
	BCC d:8 (BHS d:8)	—						2			C = 0	—	—	—	—	—				
	BCC d:16 (BHS d:16)	—						4				—	—	—	—	—				
	BCS d:8 (BLO d:8)	—						2			C = 1	—	—	—	—	—				
	BCS d:16 (BLO d:16)	—						4				—	—	—	—	—				
	BNE d:8	—						2			Z = 0	—	—	—	—	—				
	BNE d:16	—						4				—	—	—	—	—				
	BEQ d:8	—						2			Z = 1	—	—	—	—	—				
	BEQ d:16	—						4				—	—	—	—	—				
	BVC d:8	—						2			V = 0	—	—	—	—	—				
	BVC d:16	—						4				—	—	—	—	—				
	BVS d:8	—						2			V = 1	—	—	—	—	—				
	BVS d:16	—						4				—	—	—	—	—				
	BPL d:8	—						2			N = 0	—	—	—	—	—				
	BPL d:16	—						4				—	—	—	—	—				
	BMI d:8	—						2			N = 1	—	—	—	—	—				
	BMI d:16	—						4				—	—	—	—	—				
	BGE d:8	—						2			N⊕V = 0	—	—	—	—	—				
	BGE d:16	—						4				—	—	—	—	—				
	BLT d:8	—						2			N⊕V = 1	—	—	—	—	—				
	BLT d:16	—						4				—	—	—	—	—				
	BGT d:8	—						2			Z∨ (N⊕V) = 0	—	—	—	—	—				
	BGT d:16	—						4				—	—	—	—	—				
	BLE d:8	—						2			Z∨ (N⊕V) = 1	—	—	—	—	—				
	BLE d:16	—						4				—	—	—	—	—				

	JSR d:16								4			PC ← PC+d:16						
JSR	JSR @ERn	—			2							PC → @–SP PC ← ERn	—	—	—	—	—	
	JSR @aa:24	—						4				PC → @–SP PC ← aa:24	—	—	—	—	—	
	JSR @@aa:8	—								2		PC → @–SP PC ← @aa:8	—	—	—	—	—	
RTS	RTS	—									2	PC ← @SP+	—	—	—	—	—	

												PC ← @SP+						
SLEEP	SLEEP	—										Transition to power-down state	—	—	—	—	—	
LDC	LDC #xx:8, CCR	B	2									#xx:8 → CCR	↕	↕	↕	↕	↕	
	LDC Rs, CCR	B		2								Rs8 → CCR	↕	↕	↕	↕	↕	
	LDC @ERs, CCR	W			4							@ERs → CCR	↕	↕	↕	↕	↕	
	LDC @(d:16, ERs), CCR	W				6						@(d:16, ERs) → CCR	↕	↕	↕	↕	↕	
	LDC @(d:24, ERs), CCR	W				10						@(d:24, ERs) → CCR	↕	↕	↕	↕	↕	
	LDC @ERs+, CCR	W					4					@ERs → CCR ERs32+2 → ERs32	↕	↕	↕	↕	↕	
	LDC @aa:16, CCR	W						6				@aa:16 → CCR	↕	↕	↕	↕	↕	
	LDC @aa:24, CCR	W						8				@aa:24 → CCR	↕	↕	↕	↕	↕	
STC	STC CCR, Rd	B		2								CCR → Rd8	—	—	—	—	—	
	STC CCR, @ERd	W			4							CCR → @ERd	—	—	—	—	—	
	STC CCR, @(d:16, ERd)	W				6						CCR → @(d:16, ERd)	—	—	—	—	—	
	STC CCR, @(d:24, ERd)	W				10						CCR → @(d:24, ERd)	—	—	—	—	—	
	STC CCR, @–ERd	W					4					ERd32–2 → ERd32 CCR → @ERd	—	—	—	—	—	
	STC CCR, @aa:16	W						6				CCR → @aa:16	—	—	—	—	—	
	STC CCR, @aa:24	W						8				CCR → @aa:24	—	—	—	—	—	
ANDC	ANDC #xx:8, CCR	B	2									CCR∧#xx:8 → CCR	↕	↕	↕	↕	↕	
ORC	ORC #xx:8, CCR	B	2									CCR∨#xx:8 → CCR	↕	↕	↕	↕	↕	
XORC	XORC #xx:8, CCR	B	2									CCR⊕#xx:8 → CCR	↕	↕	↕	↕	↕	
NOP	NOP	—									2	PC ← PC+2	—	—	—	—	—	

												R4L–1 → R4L									
												until R4L=0									
												else next									
	EEPMOV. W	—									4	if R4 ≠ 0 then	—	—	—	—	—				
												repeat @R5 → @R6									
												R5+1 → R5									
												R6+1 → R6									
												R4–1 → R4									
												until R4=0									
												else next									

Notes: 1. The number of states in cases where the instruction code and its operands ar
in on-chip memory is shown here. For other cases see section A.3, Number o
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 prev
   (5) The number of states required for execution of an instruction that transfer
   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.

Instruction code:

1st byte	2nd byte
AH	AL	BH	BL

Instruction when most significant bit of B

Instruction when most significant bit of B

AH \ AL	0	1	2	3	4	5	6	7	8	9	A	B
0	NOP	Table A.2 (2)	STC	LDC	ORC	XORC	ANDC	LDC	ADD		Table A.2 (2)	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)
2	MOV.B											
3												
4	BRA	BRN	BHI	BLS	BCC	BCS	BNE	BEQ	BVC	BVS	BPL	BMI
5	MULXU	DIVXU	MULXU	DIVXU	RTS	BSR	RTE	TRAPA	Table A.2 (2)	JMP		
6	BSET	BNOT	BCLR	BTST	OR	XOR	AND	BST / BLD	MOV			
7					BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BIST / BILD		Table A.2 (2)	Table A.2 (2)	EEPMOV
8	ADD											
9	ADDX											
A	CMP											
B	SUBX											
C	OR											
D	XOR											
E	AND											
F	MOV											

Instruction code:

1st byte		2nd byte	
AH	AL	BH	BL

AH \ BH AL	0	1	2	3	4	5	6	7	8	9	A
01	MOV				LDC/STC				SLEEP		
0A	INC										
0B	ADDS					INC		INC	ADDS		
0F	DAA										
10	SHLL	SHLL		SHLL					SHAL		
11	SHLR	SHLR		SHLR					SHAR		
12	ROTXL	ROTXL		ROTXL					ROTL		
13	ROTXR	ROTXR		ROTXR					ROTR		
17	NOT	NOT		NOT		EXTU		EXTU	NEG		
1A	DEC										
1B	SUBS	SUBS				DEC		DEC	SUB		
1F	DAS										
58	BRA	BRN	BHI	BLS	BCC	BCS	BNE	BEQ	BVC	BVS	BPL
79	MOV	ADD	CMP	SUB	OR	XOR	AND				
7A	MOV	ADD	CMP	SUB	OR	XOR	AND				

Instruction code:

1st byte		2nd byte		3rd byte		4th byte	
AH	AL	BH	BL	CH	CL	DH	DL

Instruction wh
Instruction wh

AH / ALBH / BLCH \ CL	0	1	2	3	4	5	6	7	8	9	A	
01406										LDC / STC		L
01C05	MULXS		MULXS									
01D05		DIVXS		DIVXS								
01F06					OR	XOR	AND					
7Cr06 *1				BTST								
7Cr07 *1				BTST	BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD				
7Dr06 *1	BSET	BNOT	BCLR					BST / BIST				
7Dr07 *1	BSET	BNOT	BCLR									
7Eaa6 *2				BTST								
7Eaa7 *2				BTST	BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD				
7Faa6 *2	BSET	BNOT	BCLR					BST / BIST				
7Faa7 *2	BSET	BNOT	BCLR									

Notes: 1. r is the register designation field.
2. aa is the absolute address field.

When instruction is fetched from on-chip ROM, and an on-chip RAM is accessed.

A. 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-
and on-chip RAM is used for stack area.

B. 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 = $2 \times 2 + 1 \times 2 + 1 \times 2 = 8$

Note: * Depends on which on-chip peripheral module is accessed. See section 19.1, F
Addresses (Address Order).

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

	BEQ d:8	2
	BVC d:8	2
	BVS d:8	2
	BPL d:8	2
	BMI d:8	2
	BGE d:8	2
Bcc	BLT d:8	2
	BGT d:8	2
	BLE d:8	2
	BRA d:16(BT d:16)	2
	BRN d:16(BF d:16)	2
	BHI d:16	2
	BLS d:16	2
	BCC d:16(BHS d:16)	2
	BCS d:16(BLO d:16)	2
	BNE d:16	2
	BEQ d:16	2
	BVC d:16	2
	BVS d:16	2
	BPL d:16	2
	BMI d:16	2
	BGE d:16	2
	BLT d:16	2
	BGT d:16	2
	BLE d:16	2

	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
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

	BSET Rn, @ERd	2		2
	BSET Rn, @aa:8	2		2
BSR	BSR d:8	2	1	
	BSR d:16	2	1	
BST	BST #xx:3, Rd	1		
	BST #xx:3, @ERd	2		2
	BST #xx:3, @aa:8	2		2
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		

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					
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					
	JMP @@aa:8	2	1				
JSR	JSR @ERn	2			1		
	JSR @aa:24	2			1		
	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
	LDC@aa:16, CCR	3					1
	LDC@aa:24, CCR	4					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	
	MOV.B Rs, @aa:8	1		1	
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
	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

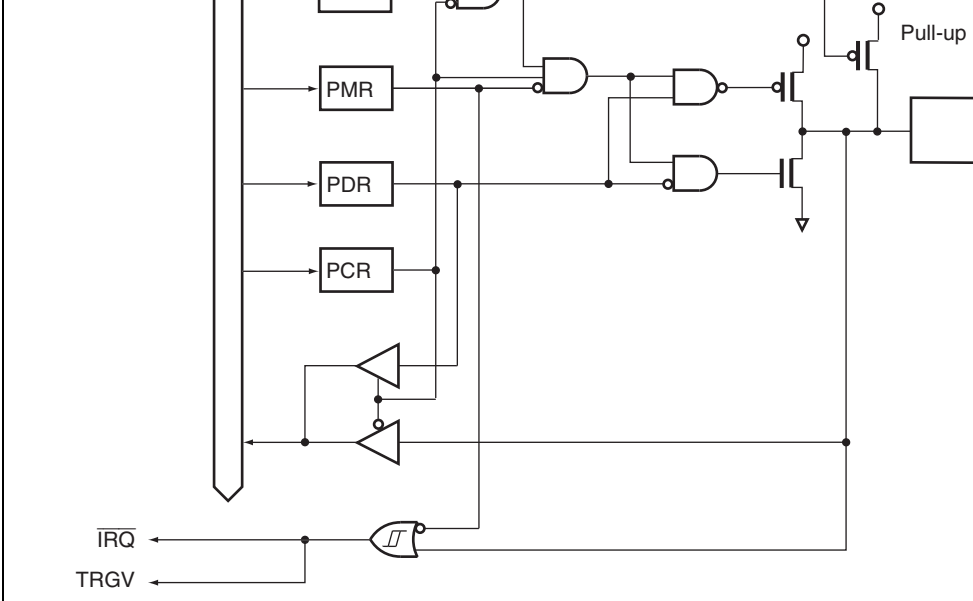
	MOV.L @(d:24,ERs), ERd	5		2
	MOV.L @ERs+, ERd	2		2
	MOV.L @aa:16, ERd	3		2
	MOV.L @aa:24, ERd	4		2
	MOV.L ERs,@ERd	2		2
	MOV.L ERs, @(d:16,ERd)	3		2
	MOV.L ERs, @(d:24,ERd)	5		2
	MOV.L ERs, @-ERd	2		2
	MOV.L ERs, @aa:16	3		2
	MOV.L ERs, @aa:24	4		2
MOVFPE	MOVFPE @aa:16, Rd*[2]	2	1	
MOVTPE	MOVTPE Rs,@aa:16*[2]	2	1	
MULXS	MULXS.B Rs, Rd	2		
	MULXS.W Rs, ERd	2		
MULXU	MULXU.B Rs, Rd	1		
	MULXU.W Rs, ERd	1		
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		

POP	POP.W Rn	1		1
	POP.L ERn	2		2
PUSH	PUSH.W Rn	1		1
	PUSH.L ERn	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		
ROTXR	ROTXR.B Rd	1		
	ROTXR.W Rd	1		
	ROTXR.L ERd	1		
RTE	RTE	2	2	
RTS	RTS	2	1	
SHAL	SHAL.B Rd	1		
	SHAL.W Rd	1		
	SHAL.L ERd	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
	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					
SUBX	SUBX #xx:8, Rd	1					
	SUBX. Rs, Rd	1					
TRAPA	TRAPA #xx:2	2	1		2		

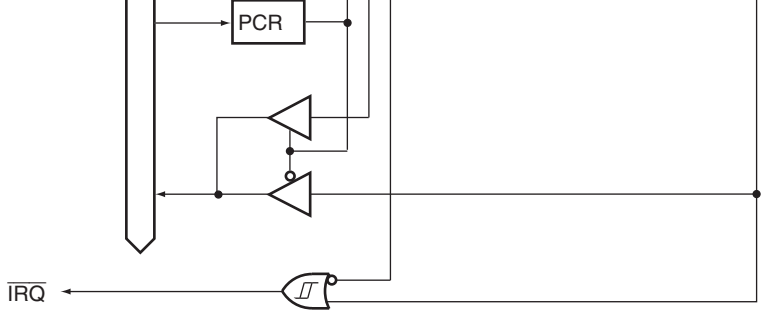
Notes: 1. n: Specified value in R4L and R4. The source and destination operands are ac
n+1 times respectively.
2. Cannot be used in this LSI.

instructions	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		—	—	—	—	—	—	—	—	—	—	—	



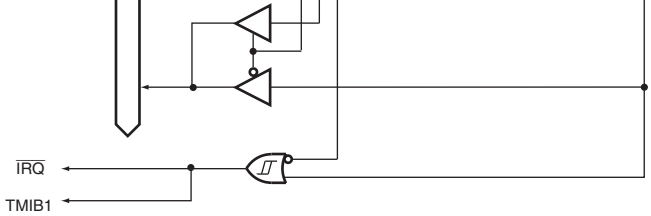
[Legend]
PUCR: Port pull-up control register
PMR: Port mode register
PDR: Port data register
PCR: Port control register

Figure B.1 Port 1 Block Diagram (P17)



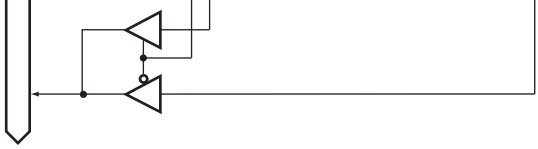
[Legend]
PUCR: Port pull-up control register
PMR: Port mode register
PDR: Port data register
PCR: Port control register

Figure B.2 Port 1 Block Diagram (P14, P16)



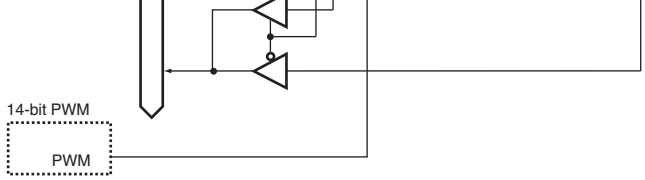
[Legend]
PUCR: Port pull-up control register
PMR: Port mode register
PDR: Port data register
PCR: Port control register

Figure B.3 Port 1 Block Diagram (P15)



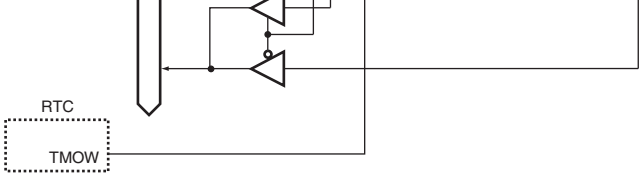
[Legend]
PUCR: Port pull-up control register
PDR: Port data register
PCR: Port control register

Figure B.4 Port 1 Block Diagram (P12)



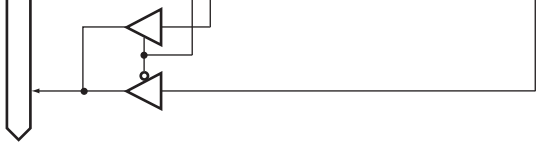
[Legend]
PUCR: Port pull-up control register
PMR: Port mode register
PDR: Port data register
PCR: Port control register

Figure B.5 Port 2 Block Diagram (P11)



[Legend]
PUCR: Port pull-up control register
PMR: Port mode register
PDR: Port data register
PCR: Port control register

Figure B.6 Port 1 Block Diagram (P10)



[Legend]
PMR: Port mode register
PDR: Port data register
PCR: Port control register

Figure B.7 Port 2 Block Diagram (P24, P23)

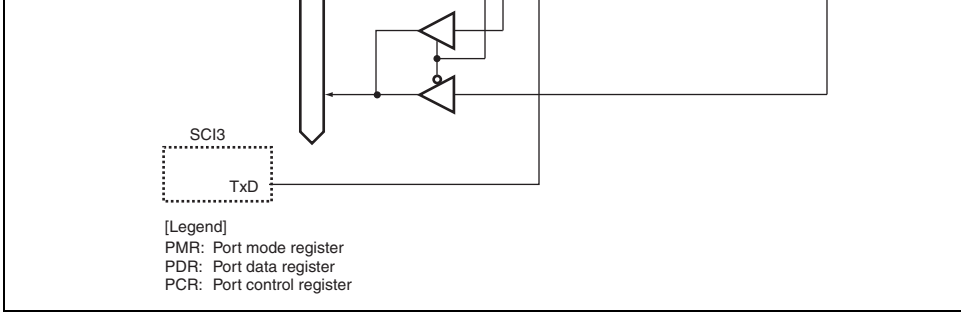


Figure B.8 Port 2 Block Diagram (P22)

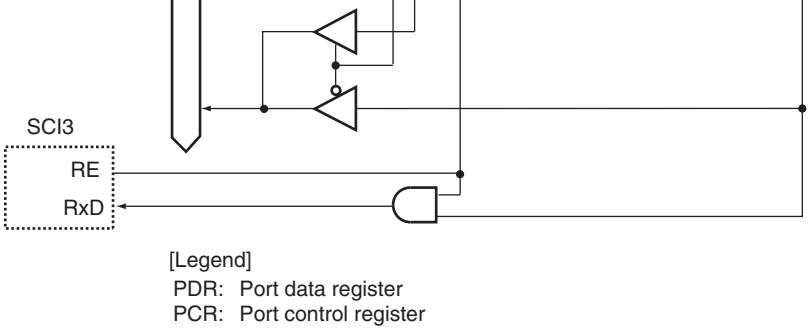
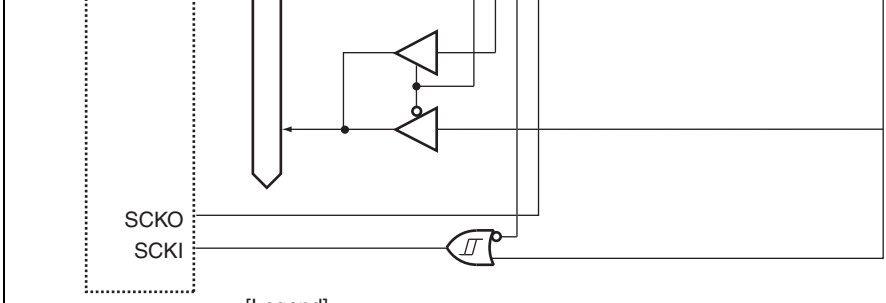
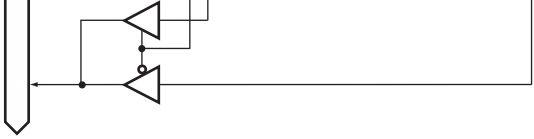


Figure B.9 Port 2 Block Diagram (P21)



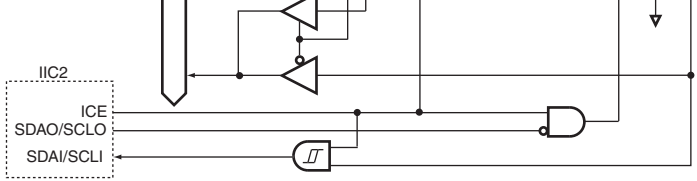
[Legend]
PDR: Port data register
PCR: Port control register

Figure B.10 Port 2 Block Diagram (P20)



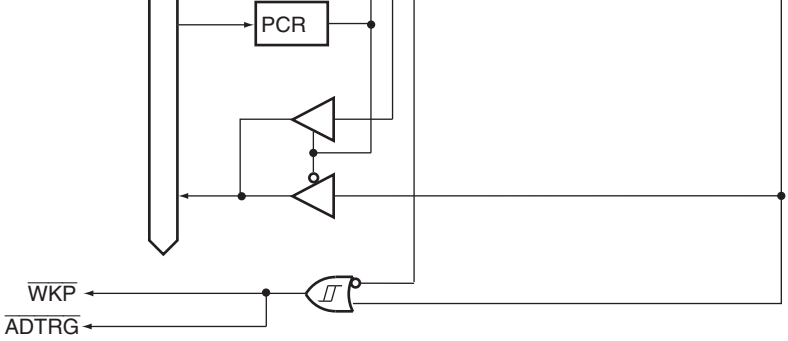
[Legend]
PDR: Port data register
PCR: Port control register

Figure B.11 Port 3 Block Diagram (P37 to P30)



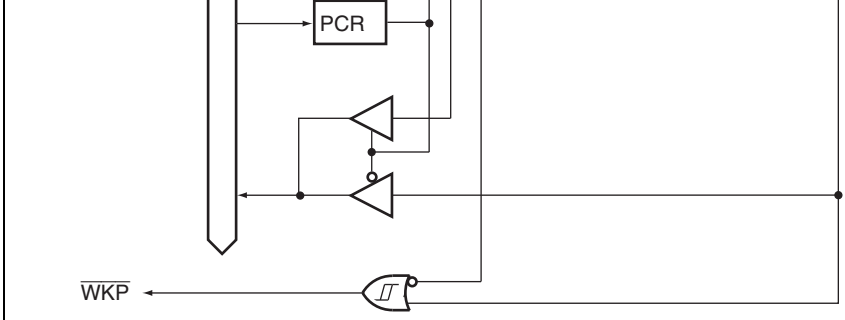
[Legend]
PMR: Port mode register
PDR: Port data register
PCR: Port control register

Figure B.12 Port 5 Block Diagram (P57, P56)



[Legend]
PUCR: Port pull-up control register
PMR: Port mode register
PDR: Port data register
PCR: Port control register

Figure B.13 Port 5 Block Diagram (P55)



[Legend]
PUCR: Port pull-up control register
PMR: Port mode register
PDR: Port data register
PCR: Port control register

Figure B.14 Port 5 Block Diagram (P54 to P50)

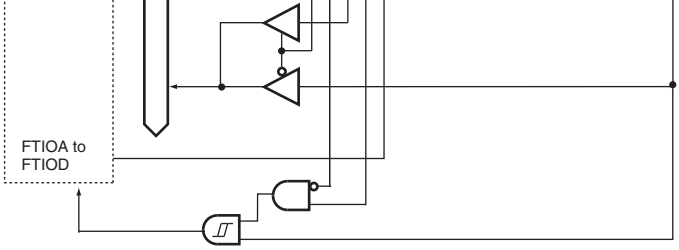


Figure B.15 Port 6 Block Diagram (P67 to P60)

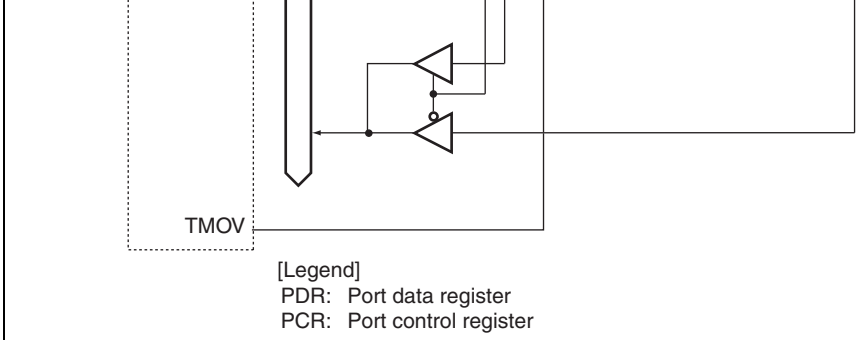


Figure B.16 Port 7 Block Diagram (P76)

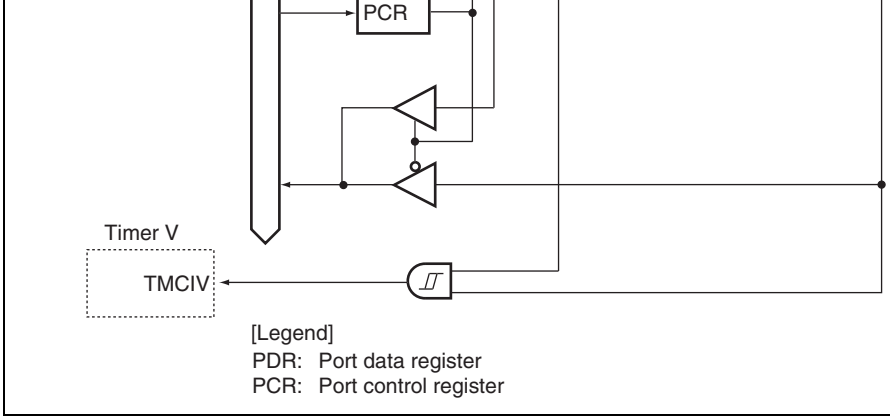


Figure B.17 Port 7 Block Diagram (P75)

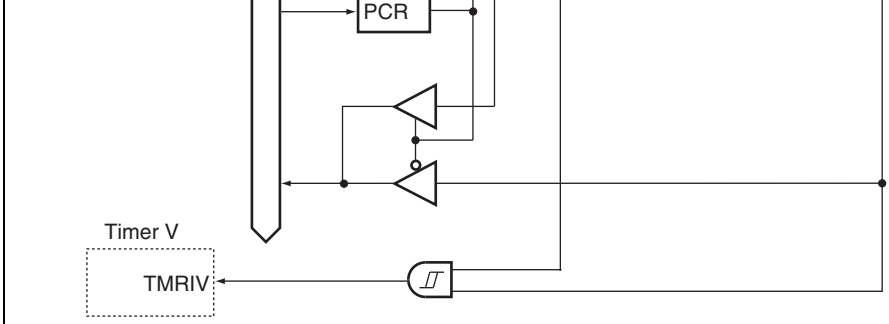


Figure B.18 Port 7 Block Diagram (P74)

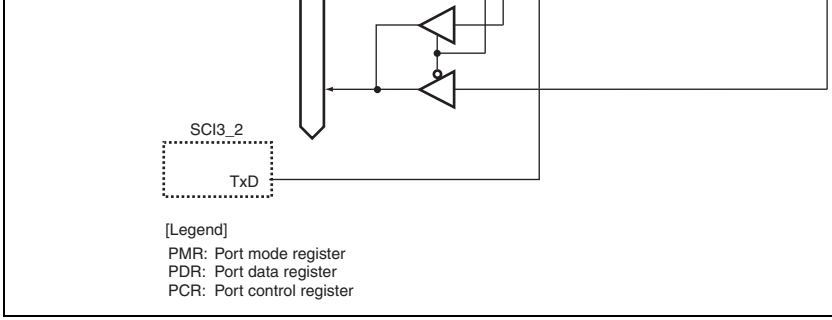


Figure B.19 Port 7 Block Diagram (P72)

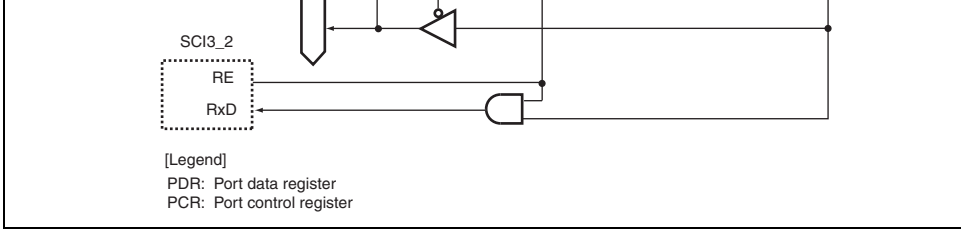


Figure B.20 Port 7 Block Diagram (P71)

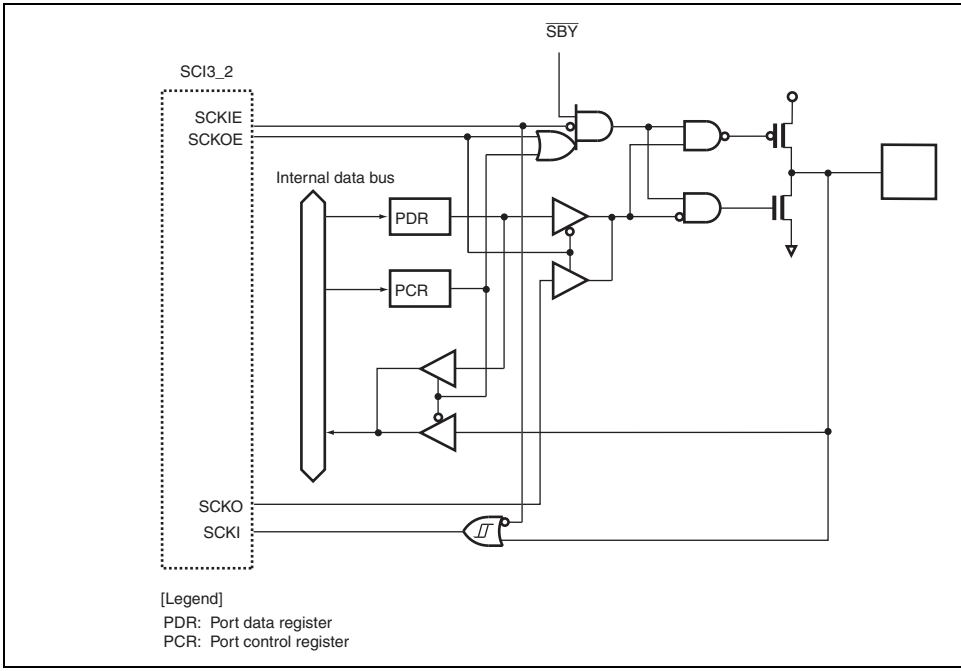
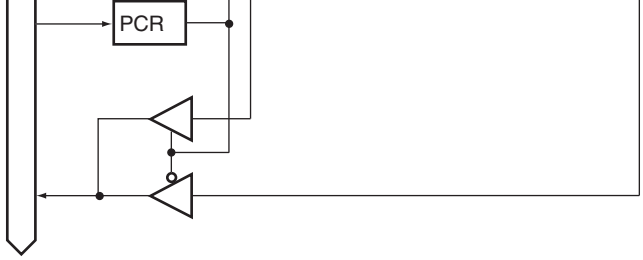


Figure B.21 Port 7 Block Diagram (P70)



[Legend]
PDR: Port data register
PCR: Port control register

Figure B.22 Port 8 Block Diagram (P87 to P85)

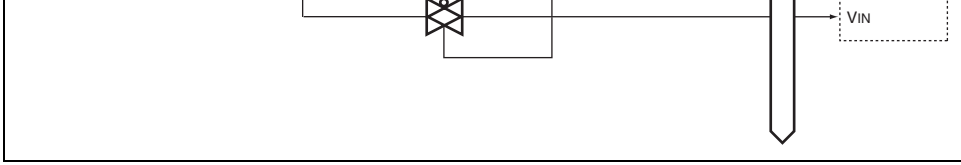


Figure B.23 Port B Block Diagram (PB7 to PB0)

P67 to P60	High impedance	Retained	Retained	High impedance	Functioning	Fun
P76 to P74, P72 to P70	High impedance	Retained	Retained	High impedance	Functioning	Fun
P87 to P85	High impedance	Retained	Retained	High impedance	Functioning	Fun
PB7 to PB0	High impedance	High impedance	High impedance	High impedance	High impedance	Hig imp

Notes: * High level output when the pull-up MOS is in on state.

H8/36085	Mask ROM version	Standard product	HD64336085H	D336085(***)H	QFP-64
			HD64336085FP	D336085(***)FP	LQFP-64
H8/36084	Mask ROM version	Standard product	HD64336084H	D336084(***)H	QFP-64
			HD64336084FP	D336084(***)FP	LQFP-64
H8/36083	Mask ROM version	Standard product	HD64336083H	D336083(***)H	QFP-64
			HD64336083FP	D336083(***)FP	LQFP-64
H8/36082	Mask ROM version	Standard product	HD64336082H	D336082(***)H	QFP-64
			HD64336082FP	D336082(***)FP	LQFP-64

[Legend]

(***): ROM code

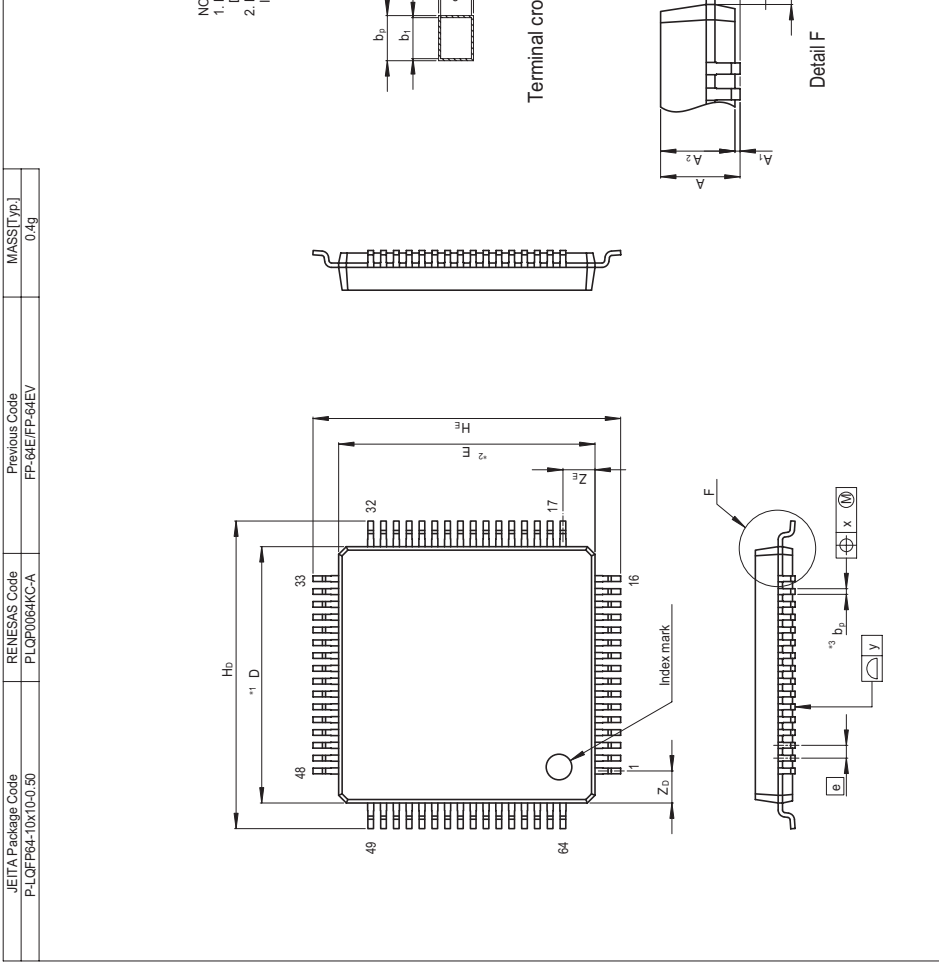


Figure D.1 FP-64E Package Dimensions

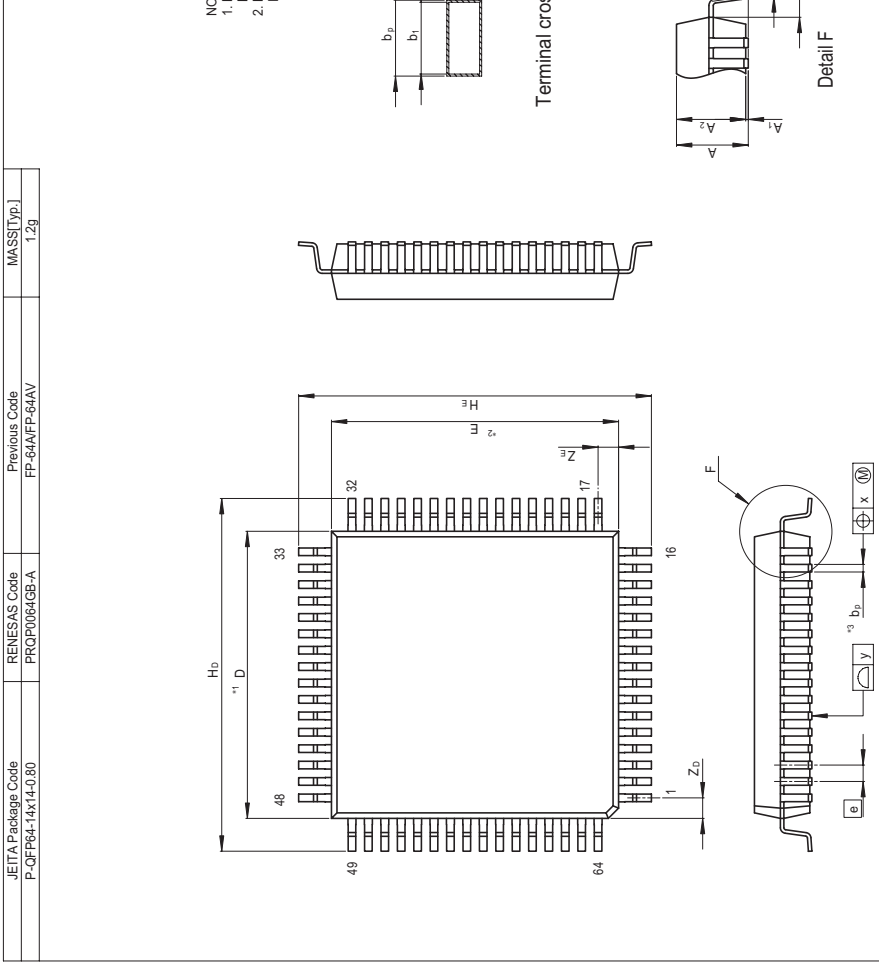


Figure D.2 FP-64A Package Dimensions

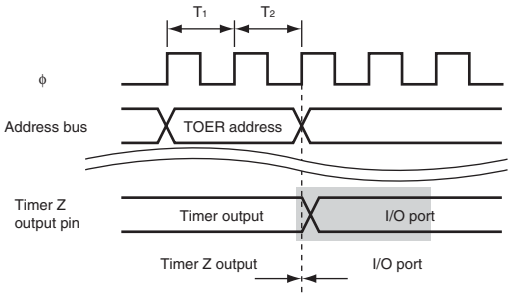
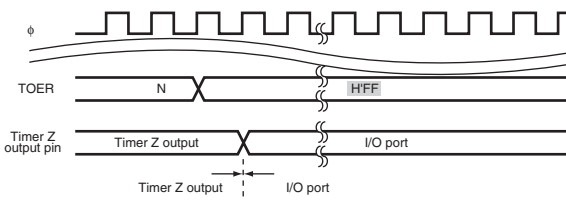
5. When the E7 or E8 is used, address breaks can
either available to the user or for use by the E7 o
address breaks are set as being used by the E7
address break control registers must not be acce
6. When the E7 or E8 is used, $\overline{\text{NMI}}$ is an input/outp
(open-drain in output mode).
7. Use channel 1 of the SCI3 (P21/RXD, P22/TXD)
board programming mode by boot mode.

Note has been deleted.

Section 6 Power-Down Modes 6.1.1 System Control Register 1 (SYSCR1)	75	

Bit	Bit Name	Description
3	NESEL	Noise Elimination Sampling Frequency The subclock pulse generator generat watch clock signal (ϕ_w) and the system pulse generator generates the oscillato (ϕ_{osc}). This bit selects the sampling fre the oscillator clock when the watch clo (ϕ_w) is sampled. When ϕ_{osc} = 4 to 18 M NESEL to 0.

Section 8 RAM	107	Note: * When the E7 or E8 is used, area H'F780 to must not be accessed.

13.4.4 Synchronous Operation	208	Figure 13.20 shows an example of synchronous oper this example, synchronous operation has been select FTIOB0 and FTIOB1 have been designated for PWM GRA_0 compare match has been set as the channel counter clearing source, and synchronous clearing ha set for the channel 1 counter clearing source. In addi same input clock has been set as the counter input cl channel 0 and channel 1. Two-phase PWM waveform output from pins FTIOB0 and FTIOB1.
13.4.9 Timer Z Output Timing Figure 13.44 Example of Output Disable Timing of Timer Z by Writing to TOER	237	
Figure 13.45 Example of Output Disable Timing of Timer Z by External Trigger	237	
Section 14 Watchdog Timer 14.2.1 Timer Control/Status Register WD (TCSRWD)	252	**Bit** **Bit Name** **Description** 4 TCSRWE Timer Control/Status Register WD Write

in SAR

[Clearing Condition]

- When 0 is written in STOP after rea
  STOP = 1

17.7 Usage Notes	343	Added
Section 18 A/D Converter 18.3.1 A/D Data Registers A to D (ADDRA to ADDRD)	348	Therefore byte access to ADDR should be done by re upper byte first then the lower one. Word access is al possible. ADDR is initialized to H'0000.
Section 20 Electrical Characteristics 20.2.2 DC Characteristics Table 20.2 DC Characteristics (1)	379	

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 operat
Sleep mode 2		Only timers operat (φOSC/64)

20.3.2 DC Characteristics 20.3.3 AC Characteristics 20.3.4 A/D Converter Characteristics 20.3.5 Watchdog Timer Characteristics	391 to 402	Preliminary has been deleted.
Appendix D.1 Package Dimensions	463, 464	Swapped with new ones.

Sample-and-hold circuit 352
Scan mode 351
Single mode 351
Address break 61
Addressing modes 30
Absolute address 31
Immediate 32
Memory indirect 32
Program-counter relative 32
Register direct 31
Register indirect 31
Register indirect with displacement 31
Register indirect with post-increment 31
Register indirect with pre-decrement 31

C

Clock pulse generators 67
Prescaler S 71
Prescaler W 71
Subclock generator 70
System clock generator 68
Condition field 29
Condition-code register (CCR) 14
CPU 9

E

Effective address 34
Flash memory .....
Boot mode .....
Boot program .....
Erase/erase-verify .....
Erasing units .....
Error protection .....
Hardware protection .....
Power-down states .....
Program/program-verify .....
Programmer mode .....
Programming units .....
Programming/erasing in user pr
mode .....
Software protection .....

G

General registers .....

I

I/O ports .....
I/O port block diagrams .....
I^2C bus format .....
I^2C bus interface 2 (IIC2) .....
Acknowledge .....
Bit synchronous circuit .....
Clock synchronous serial forma
Noise canceler .....

Shift instructions 23
System control instructions 27
Interrupt
Internal interrupts 57
Interrupt response time 57, 58
IRQ3 to IRQ0 interrupts 55
NMI interrupt 55
WKP5 to WKP0 interrupts 56
Interrupt mask bit 15

L

Large current ports 2

M

Memory map 10
Module standby function 85

O

On-board programming modes 93
Operation field 29

P

Package 2
Package dimensions 462
Realtime clock (RTC) .....
Data reading procedure .....
Initial setting procedure .....
Registers
ABRKCR 62, 361,
ABRKSR 64, 361,
ADCR 350, 361,
ADCSR 349, 361,
ADDRA 348, 361,
ADDRB 348, 361,
ADDRC 348, 361,
ADDRD 348, 361,
BARH 64, 361,
BARL 64, 361,
BDRH 64, 361,
BDRL 64, 362,
BRR 271, 361,
EBR1 ..... 91, 360,
FENR 92, 360,
FLMCR1 ..... 89, 360,
FLMCR2 ..... 90, 360,
FLPWCR 92, 360,
GRA 188, 358,
GRB 188, 358,
GRC 188, 358,
GRD 188, 358,
ICCR1 ..... 311, 360,
ICCR2 ..... 313, 360,
ICDRR 323, 360,

IWPR 53, 363, 368, 372
MSTCR1 ..... 77, 363, 368, 372
MSTCR2 ..... 78, 363, 368, 372
PCR1 ..... 111, 362, 368, 372
PCR2 ..... 115, 362, 368, 372
PCR3 ..... 119, 362, 368, 372
PCR5 ..... 124, 362, 368, 372
PCR6 ..... 129, 362, 368, 372
PCR7 ..... 134, 362, 368, 372
PCR8 ..... 137, 362, 368, 372
PDR1 ..... 111, 362, 367, 372
PDR2 ..... 116, 362, 367, 372
PDR3 ..... 119, 362, 367, 372
PDR5 ..... 124, 362, 367, 372
PDR6 ..... 129, 362, 367, 372
PDR7 ..... 135, 362, 367, 372
PDR8 ..... 138, 362, 367, 372
PDRB 139, 362, 367, 372
PMR1 ..... 110, 362, 368, 372
PMR3 ..... 116, 362, 368, 372
PMR5 ..... 123, 362, 368, 372
POCR 195, 358, 364, 369
PUCR1 ..... 112, 362, 367, 372
PUCR5 ..... 125, 362, 367, 372
PWCR 258, 361, 367, 371
PWDRL 259, 361, 367, 371
PWDRU 259, 361, 367, 371
RDR 265, 361, 366, 371
RHRDR 145, 359, 365, 370
SYSCR1 ..... 74, 36
SYSCR2 ..... 76, 36
TCB1 ..... 155, 36
TCNT 187, 35
TCNTV 161, 36
TCORA 162, 36
TCORB 162, 36
TCR 189, 35
TCRV0 ..... 162, 36
TCRV1 ..... 165, 36
TCSRV 164, 36
TCSRWD 252, 36
TCWD 253, 36
TDR 265, 36
TFCR 182, 35
TIER 194, 35
TIORA 190, 35
TIORC 191, 35
TLB1 .....
TMB1 ..... 154, 36
TMDR 180, 35
TMWD 254, 36
TOCR 186, 35
TOER 185, 35
TPMR 181, 35
TSR 192, 35
TSTR 180, 35
Register field .....

Overrun error 284
Parity error 284
Stack pointer (SP) 14

T

Timer B1 153
Auto-reload timer operation 156
Event counter operation 156

V

Vector address .....

W

Watchdog timer .....

Renesas 16-Bit Single-Chip Microcomputer
Hardware Manual
H8/36087 Group

Publication Date:	Rev.1.00, Jun. 03, 2004
	Rev.2.00, Sep. 23, 2005
Published by:	Sales Strategic Planning Div.
	Renesas Technology Corp.
Edited by:	Customer Support Department
	Global Strategic Communication Div.
	Renesas Solutions Corp.

RENESAS SALES OFFICES

Refer to "**http://www.renesas.com/en/network**" for the latest and detailed information.

Renesas Technology America, Inc.
450 Holger Way, San Jose, CA 95134-1368, U.S.A
Tel: <1> (408) 382-7500, Fax: <1> (408) 382-7501

Renesas Technology Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, United Kingdom
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

Renesas Technology Hong Kong Ltd.
7th Floor, North Tower, World Finance Centre, Harbour City, 1 Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2730-6071

Renesas Technology Taiwan Co., Ltd.
10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 2713-2999

Renesas Technology (Shanghai) Co., Ltd.
Unit2607 Ruijing Building, No.205 Maoming Road (S), Shanghai 200020, China
Tel: <86> (21) 6472-1001, Fax: <86> (21) 6415-2952

Renesas Technology Singapore Pte. Ltd.
1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

Renesas Technology Korea Co., Ltd.
Kukje Center Bldg. 18th Fl., 191, 2-ka, Hangang-ro, Yongsan-ku, Seoul 140-702, Korea
Tel: <82> 2-796-3115, Fax: <82> 2-796-2145

Renesas Technology Malaysia Sdn. Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No.18, Jalan Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan,
Tel: <603> 7955-9390, Fax: <603> 7955-9510

H8/36087 Group
Hardware Manual

Renesas Electronics Corporation
1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan

REJ09B0160-0200

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for 16-bit Microcontrollers - MCU ***category:***

Click to view products by Renesas ***manufacturer:***

Other Similar products are found below :

MB90F036APMC-GSE1 MB90F342CASPMC-GSE1 MB90F345CESPMC-GE1 MB90F349CAPFR-GSE1 MB90F428GCPFR-GSE1 MB90F462APFM-GE1 MB90F462APMC-G-SNE1 MB90F497GPF-GE1 MB90F546GSPFR-GE1 MB90F947APFR-GS-SPE1 MB96F346RSBPMC-GS-N2E2 MB96F683RBPMC-GSAE1 R5F11BGEAFB#30 DF3026XBL25V S912ZVFP64F1VLL R4F24268NVRFQV R5F107DEGSP#X0 R5F11B7EANA#U0 R5F21172DSP#U0 M30622F8PGP#U3C MB90092PF-G-BNDE1 MB90F335APMC1-G-SPE1 MB90F342CASPFR-GS-N2E1 MB90F345CAPFR-GSE1 MB90F543GPF-GE1 MB90F546GSPF-GE1 MB90F568PMCR-GE1 MB90F594APFR-GE1 MB90F882ASPMC-GE1 MB96F346RSAPQCR-GS-N2E2 MB96F387RSBPMC-GSE2 MB96F387RSBPMC-GS-N2E2 MB96F395RSAPMC-GSE2 MB96F623RBPMC1-GSE1 MB96F646RBPMC-GSE1 XE167F96F66LACFXUMA1 MB96F696RBPMC-GSAE1 MB96F018RBPMC-GSE1 MB90F962SPMCR-GE1 MB90F867ASPFR-GE1 MB90F543GPF-G-FLE1 MB90F345CESPF-GE1 M30290FCHP#U3A DF2239FA20IV HD64F3672FPV R5F104AEASP#V0 R5F100BCANA#U0 R5F100BFANA#U0 S9S12H256J2VFVER R5F100ACASP#V0