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

H8S/2626 Group, H8S/2623 Group, H8S/2626F-ZTAT™, H8S/2623F-ZTAT™

Hardware Manual

Renesas 16-Bit Single-Chip Microcomputer H8S Family/H8S/2600 Series

H8S/2626	HD6432626
	HD64F2626
H8S/2625	HD6432625
H8S/2623	HD6432623
	HD64F2623
H8S/2622	HD6432622
H8S/2621	HD64F2621

Renesas Electronics
www.renesas.com

Rev.5.00 2006.01

1. These materials are intended as a reference to assist our customers in the selection of the R
   Technology Corp. product best suited to the customer's application; they do not convey any
   under any intellectual property rights, or any other rights, belonging to Renesas Technology
   a third party.
2. Renesas Technology Corp. assumes no responsibility for any damage, or infringement of ar
   party's rights, originating in the use of any product data, diagrams, charts, programs, algorith
   circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, progra
   algorithms represents information on products at the time of publication of these materials, a
   subject to change by Renesas Technology Corp. without notice due to product improvement
   other reasons. It is therefore recommended that customers contact Renesas Technology C
   an authorized Renesas Technology Corp. product distributor for the latest product informatio
   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.co
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
   system before making a final decision on the applicability of the information and products. R
   Technology Corp. assumes no responsibility for any damage, liability or other loss resulting
   information contained herein.
5. Renesas Technology Corp. semiconductors are not designed or manufactured for use in a d
   system that is used under circumstances in which human life is potentially at stake. Please
   Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor
   considering the use of a product contained herein for any specific purposes, such as appara
   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 c
   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 produ
   contained therein.

RENESAS

2. Treatment of Unused Input Pins

Note: Fix all unused input pins to high or low level.
Generally, the input pins of CMOS products are high-impedance input pins. If
are in their open states, intermediate levels are induced by noise in the vicinity
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 in
circuits are undefined until full power is supplied throughout the chip and a lo
input on the reset pin. During the period where the states are undefined, the re
settings and the output state of each pin are also undefined. Design your syste
does not malfunction because of processing while it is in this undefined state.
products which have a reset function, reset the LSI immediately after the pow
been turned on.

4. Prohibition of Access to Undefined or Reserved Address

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

The CPU can handle a 16-Mbyte linear address space (architecturally 4 Gbytes). Prog
on the high-level language C can also be run efficiently.

The address space is divided into eight areas. The data bus width and access states can
for each of these areas, and various kinds of memory can be connected fast and easily.

Single-power-supply flash memory (F-ZTAT™*), and mask ROM versions are availa
providing a quick and flexible response to conditions from ramp-up through full-scale
production, even for applications with frequently changing specifications.

On-chip supporting functions include a 16-bit timer pulse unit (TPU), programmable p
generator (PPG), watchdog timer (WDT), serial communication interface (SCI), contr
network (HCAN), A/D converter, D/A converter (H8S/2626 Group only), and I/O por

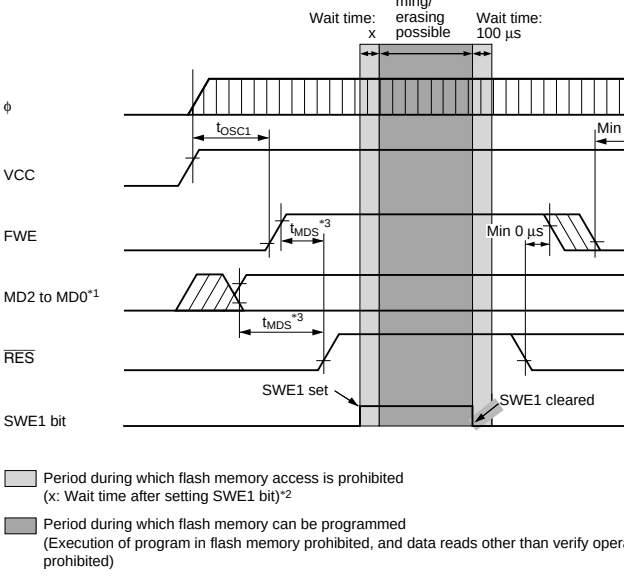
In addition, data transfer controller (DTC) is provided, enabling high-speed data trans
CPU intervention.

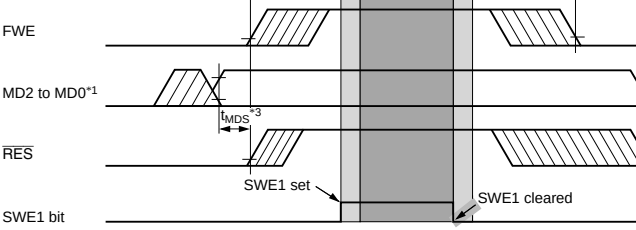
Use of the H8S/2626 Group or H8S/2623 Group enables easy implementation of com
performance systems capable of processing large volumes of data.

This manual describes the hardware of the H8S/2626 Group and H8S/2623 Group. Re
H8S/2600 Series and H8S/2000 Series Programming Manual for a detailed description
instruction set.

Note: * F-ZTAT (Flexible-ZTAT) is a trademark of Renesas Technology Corp.

Figure 19.26 Power-On/Off Timing (Boot Mode)





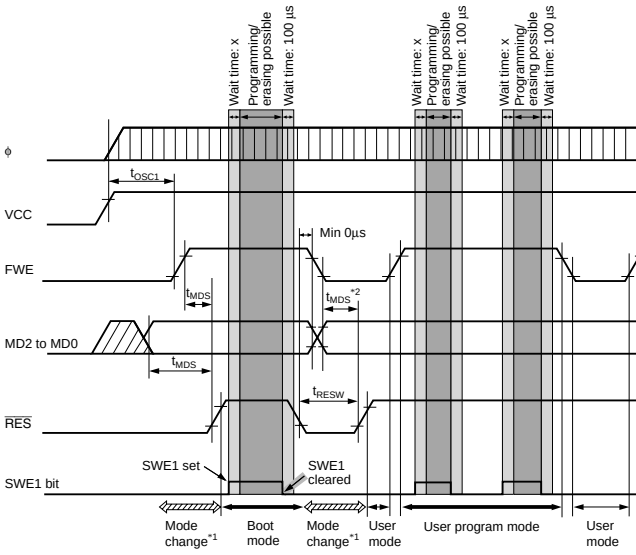
Period during which flash memory access is prohibited
(x: Wait time after setting SWE1 bit)*2

Period during which flash memory can be programmed
(Execution of program in flash memory prohibited, and data reads other than verify operati
prohibited)

Figure 19.28 Mode Transition Timing (Example: Boot Mode → User Mode ↔ User Program Mode)

684

Figure 19.28 amended



Period during which flash memory access is prohibited
(x: Wait time after setting SWE1 bit)*3

Period during which flash memory can be programmed
(Execution of program in flash memory prohibited, and data reads other than verify operations prohibited

1.3.2 Pin Functions in Each Operating Mode
1.3.3 Pin Functions

Section 2 CPU
2.1 Overview
2.1.1 Features
2.1.2 Differences between H8S/2600 CPU and H8S/2000 CPU
2.1.3 Differences from H8/300 CPU
2.1.4 Differences from H8/300H CPU
2.2 CPU Operating Modes
2.3 Address Space
2.4 Register Configuration
2.4.1 Overview
2.4.2 General Registers
2.4.3 Control Registers
2.4.4 Initial Register Values
2.5 Data Formats
2.5.1 General Register Data Formats
2.5.2 Memory Data Formats
2.6 Instruction Set
2.6.1 Overview
2.6.2 Instructions and Addressing Modes
2.6.3 Table of Instructions Classified by Function
2.6.4 Basic Instruction Formats
2.7 Addressing Modes and Effective Address Calculation
2.7.1 Addressing Mode
2.7.2 Effective Address Calculation
2.8 Processing States
2.8.1 Overview
2.8.2 Reset State
2.8.3 Exception-Handling State
2.8.4 Program Execution State

2.10.1 TAS Instruction

Section 3 MCU Operating Modes
3.1 Overview
3.1.1 Operating Mode Selection
3.1.2 Register Configuration
3.2 Register Descriptions
3.2.1 Mode Control Register (MDCR)
3.2.2 System Control Register (SYSCR)
3.2.3 Pin Function Control Register (PFCR)
3.3 Operating Mode Descriptions
3.3.1 Mode 4
3.3.2 Mode 5
3.3.3 Mode 6
3.3.4 Mode 7
3.4 Pin Functions in Each Operating Mode
3.5 Address Map in Each Operating Mode

Section 4 Exception Handling
4.1 Overview
4.1.1 Exception Handling Types and Priority
4.1.2 Exception Handling Operation
4.1.3 Exception Vector Table
4.2 Reset
4.2.1 Overview
4.2.2 Reset Sequence
4.2.3 Interrupts after Reset
4.2.4 State of On-Chip Supporting Modules after Reset Release
4.3 Traces
4.4 Interrupts
4.5 Trap Instruction
4.6 Stack Status after Exception Handling

5.2.1 System Control Register (SYSCR)
5.2.2 Interrupt Priority Registers A to K, M (IPRA to IPRK, IPRM)
5.2.3 IRQ Enable Register (IER)
5.2.4 IRQ Sense Control Registers H and L (ISCRH, ISCRL)
5.2.5 IRQ Status Register (ISR)
5.3 Interrupt Sources
5.3.1 External Interrupts
5.3.2 Internal Interrupts
5.3.3 Interrupt Exception Handling Vector Table
5.4 Interrupt Operation
5.4.1 Interrupt Control Modes and Interrupt Operation
5.4.2 Interrupt Control Mode 0
5.4.3 Interrupt Control Mode 2
5.4.4 Interrupt Exception Handling Sequence
5.4.5 Interrupt Response Times
5.5 Usage Notes
5.5.1 Contention between Interrupt Generation and Disabling
5.5.2 Instructions that Disable Interrupts
5.5.3 Times when Interrupts are Disabled
5.5.4 Interrupts during Execution of EEPMOV Instruction
5.6 DTC Activation by Interrupt
5.6.1 Overview
5.6.2 Block Diagram
5.6.3 Operation

Section 6 PC Break Controller (PBC)
6.1 Overview
6.1.1 Features
6.1.2 Block Diagram
6.1.3 Register Configuration
6.2 Register Descriptions
6.2.1 Break Address Register A (BARA)

6.3.5 PC Break Operation in Continuous Data Transfer
6.3.6 When Instruction Execution is Delayed by One State
6.3.7 Additional Notes

Section 7 Bus Controller
7.1 Overview
7.1.1 Features
7.1.2 Block Diagram
7.1.3 Pin Configuration
7.1.4 Register Configuration
7.2 Register Descriptions
7.2.1 Bus Width Control Register (ABWCR)
7.2.2 Access State Control Register (ASTCR)
7.2.3 Wait Control Registers H and L (WCRH, WCRL)
7.2.4 Bus Control Register H (BCRH)
7.2.5 Bus Control Register L (BCRL)
7.2.6 Pin Function Control Register (PFCR)
7.3 Overview of Bus Control
7.3.1 Area Partitioning
7.3.2 Bus Specifications
7.3.3 Memory Interfaces
7.3.4 Interface Specifications for Each Area
7.4 Basic Bus Interface
7.4.1 Overview
7.4.2 Data Size and Data Alignment
7.4.3 Valid Strobes
7.4.4 Basic Timing
7.4.5 Wait Control
7.5 Burst ROM Interface
7.5.1 Overview
7.5.2 Basic Timing
7.5.3 Wait Control

7.8.5 Usage Note
7.9 Bus Arbitration
7.9.1 Overview
7.9.2 Operation
7.9.3 Bus Transfer Timing
7.10 Resets and the Bus Controller

Section 8 Data Transfer Controller (DTC)
8.1 Overview
8.1.1 Features
8.1.2 Block Diagram
8.1.3 Register Configuration
8.2 Register Descriptions
8.2.1 DTC Mode Register A (MRA)
8.2.2 DTC Mode Register B (MRB)
8.2.3 DTC Source Address Register (SAR)
8.2.4 DTC Destination Address Register (DAR)
8.2.5 DTC Transfer Count Register A (CRA)
8.2.6 DTC Transfer Count Register B (CRB)
8.2.7 DTC Enable Registers (DTCER)
8.2.8 DTC Vector Register (DTVECR)
8.2.9 Module Stop Control Register A (MSTPCRA)
8.3 Operation
8.3.1 Overview
8.3.2 Activation Sources
8.3.3 DTC Vector Table
8.3.4 Location of Register Information in Address Space
8.3.5 Normal Mode
8.3.6 Repeat Mode
8.3.7 Block Transfer Mode
8.3.8 Chain Transfer
8.3.9 Operation Timing

9.2.1 Overview
9.2.2 Register Configuration
9.2.3 Pin Functions
9.3 Port 4
9.3.1 Overview
9.3.2 Register Configuration
9.3.3 Pin Functions
9.4 Port 9
9.4.1 Overview
9.4.2 Register Configuration
9.4.3 Pin Functions
9.5 Port A
9.5.1 Overview
9.5.2 Register Configuration
9.5.3 Pin Functions
9.5.4 MOS Input Pull-Up Function
9.6 Port B
9.6.1 Overview
9.6.2 Register Configuration
9.6.3 Pin Functions
9.6.4 MOS Input Pull-Up Function
9.7 Port C
9.7.1 Overview
9.7.2 Register Configuration
9.7.3 Pin Functions
9.7.4 MOS Input Pull-Up Function
9.8 Port D
9.8.1 Overview
9.8.2 Register Configuration
9.8.3 Pin Functions
9.8.4 MOS Input Pull-Up Function
9.9 Port E

Section 10 16-Bit Timer Pulse Unit (TPU)
10.1 Overview
10.1.1 Features
10.1.2 Block Diagram
10.1.3 Pin Configuration
10.1.4 Register Configuration
10.2 Register Descriptions
10.2.1 Timer Control Register (TCR)
10.2.2 Timer Mode Register (TMDR)
10.2.3 Timer I/O Control Register (TIOR)
10.2.4 Timer Interrupt Enable Register (TIER)
10.2.5 Timer Status Register (TSR)
10.2.6 Timer Counter (TCNT)
10.2.7 Timer General Register (TGR)
10.2.8 Timer Start Register (TSTR)
10.2.9 Timer Synchro Register (TSYR)
10.2.10 Module Stop Control Register A (MSTPCRA)
10.3 Interface to Bus Master
10.3.1 16-Bit Registers
10.3.2 8-Bit Registers
10.4 Operation
10.4.1 Overview
10.4.2 Basic Functions
10.4.3 Synchronous Operation
10.4.4 Buffer Operation
10.4.5 Cascaded Operation
10.4.6 PWM Modes
10.4.7 Phase Counting Mode
10.5 Interrupts
10.5.1 Interrupt Sources and Priorities
10.5.2 DTC Activation
10.5.3 A/D Converter Activation

11.1.3 Pin Configuration
11.1.4 Registers
11.2 Register Descriptions
11.2.1 Next Data Enable Registers H and L (NDERH, NDERL)
11.2.2 Output Data Registers H and L (PODRH, PODRL)
11.2.3 Next Data Registers H and L (NDRH, NDRL)
11.2.4 Notes on NDR Access
11.2.5 PPG Output Control Register (PCR)
11.2.6 PPG Output Mode Register (PMR)
11.2.7 Port 1 Data Direction Register (P1DDR)
11.2.8 Module Stop Control Register A (MSTPCRA)
11.3 Operation
11.3.1 Overview
11.3.2 Output Timing
11.3.3 Normal Pulse Output
11.3.4 Non-Overlapping Pulse Output
11.3.5 Inverted Pulse Output
11.3.6 Pulse Output Triggered by Input Capture
11.4 Usage Notes

Section 12 Watchdog Timer
12.1 Overview
12.1.1 Features
12.1.2 Block Diagram
12.1.3 Pin Configuration
12.1.4 Register Configuration
12.2 Register Descriptions
12.2.1 Timer Counter (TCNT)
12.2.2 Timer Control/Status Register (TCSR)
12.2.3 Reset Control/Status Register (RSTCSR)
12.2.4 Pin Function Control Register (PFCR)
12.2.5 Notes on Register Access

12.5.3 Switching between Watchdog Timer Mode and Interval Timer Mode...
12.5.4 System Reset by $\overline{\text{WDTOVF}}$ Signal...
12.5.5 Internal Reset in Watchdog Timer Mode...
12.5.6 OVF Flag Clearing in Interval Timer Mode ...

Section 13 Serial Communication Interface (SCI) ...
13.1 Overview...
13.1.1 Features...
13.1.2 Block Diagram...
13.1.3 Pin Configuration...
13.1.4 Register Configuration...
13.2 Register Descriptions ...
13.2.1 Receive Shift Register (RSR) ...
13.2.2 Receive Data Register (RDR) ...
13.2.3 Transmit Shift Register (TSR) ...
13.2.4 Transmit Data Register (TDR)...
13.2.5 Serial Mode Register (SMR)...
13.2.6 Serial Control Register (SCR)...
13.2.7 Serial Status Register (SSR) ...
13.2.8 Bit Rate Register (BRR) ...
13.2.9 Smart Card Mode Register (SCMR) ...
13.2.10 Module Stop Control Register B (MSTPCRB)...
13.3 Operation ...
13.3.1 Overview...
13.3.2 Operation in Asynchronous Mode ...
13.3.3 Multiprocessor Communication Function...
13.3.4 Operation in Clocked Synchronous Mode ...
13.4 SCI Interrupts...
13.5 Usage Notes ...

Section 14 Smart Card Interface ...
14.1 Overview...

14.3 Operation
14.3.1 Overview
14.3.2 Pin Connections
14.3.3 Data Format
14.3.4 Register Settings
14.3.5 Clock
14.3.6 Data Transfer Operations
14.3.7 Operation in GSM Mode
14.3.8 Operation in Block Transfer Mode
14.4 Usage Notes

Section 15 Controller Area Network (HCAN)
15.1 Overview
15.1.1 Features
15.1.2 Block Diagram
15.1.3 Pin Configuration
15.1.4 Register Configuration
15.2 Register Descriptions
15.2.1 Master Control Register (MCR)
15.2.2 General Status Register (GSR)
15.2.3 Bit Configuration Register (BCR)
15.2.4 Mailbox Configuration Register (MBCR)
15.2.5 Transmit Wait Register (TXPR)
15.2.6 Transmit Wait Cancel Register (TXCR)
15.2.7 Transmit Acknowledge Register (TXACK)
15.2.8 Abort Acknowledge Register (ABACK)
15.2.9 Receive Complete Register (RXPR)
15.2.10 Remote Request Register (RFPR)
15.2.11 Interrupt Register (IRR)
15.2.12 Mailbox Interrupt Mask Register (MBIMR)
15.2.13 Interrupt Mask Register (IMR)
15.2.14 Receive Error Counter (REC)

15.3.3 Transmit Mode
15.3.4 Receive Mode
15.3.5 HCAN Sleep Mode
15.3.6 HCAN Halt Mode
15.3.7 Interrupt Interface
15.3.8 DTC Interface
15.4 CAN Bus Interface
15.5 Usage Notes

Section 16 A/D Converter
16.1 Overview
16.1.1 Features
16.1.2 Block Diagram
16.1.3 Pin Configuration
16.1.4 Register Configuration
16.2 Register Descriptions
16.2.1 A/D Data Registers A to D (ADDRA to ADDRD)
16.2.2 A/D Control/Status Register (ADCSR)
16.2.3 A/D Control Register (ADCR)
16.2.4 Module Stop Control Register A (MSTPCRA)
16.3 Interface to Bus Master
16.4 Operation
16.4.1 Single Mode (SCAN = 0)
16.4.2 Scan Mode (SCAN = 1)
16.4.3 Input Sampling and A/D Conversion Time
16.4.4 External Trigger Input Timing
16.5 Interrupts
16.6 Usage Notes

Section 17 D/A Converter [Provided in the H8S/2626 Group only]
17.1 Overview
17.1.1 Features

Section 18 RAM
18.1 Overview
18.1.1 Block Diagram
18.1.2 Register Configuration
18.2 Register Descriptions
18.2.1 System Control Register (SYSCR)
18.3 Operation
18.4 Usage Notes

Section 19 ROM (Preliminary)
19.1 Features
19.2 Overview
19.2.1 Block Diagram
19.2.2 Mode Transitions
19.2.3 On-Board Programming Modes
19.2.4 Flash Memory Emulation in RAM
19.2.5 Differences between Boot Mode and User Program Mode
19.2.6 Block Configuration
19.3 Pin Configuration
19.4 Register Configuration
19.5 Register Descriptions
19.5.1 Flash Memory Control Register 1 (FLMCR1)
19.5.2 Flash Memory Control Register 2 (FLMCR2)
19.5.3 Erase Block Register 1 (EBR1)
19.5.4 Erase Block Register 2 (EBR2)
19.5.5 RAM Emulation Register (RAMER)
19.5.6 Flash Memory Power Control Register (FLPWCR)
19.5.7 Serial Control Register X (SCRX)
19.6 On-Board Programming Modes
19.6.1 Boot Mode
19.6.2 User Program Mode
19.7 Flash Memory Programming/Erasing

19.10 Interrupt Handling when Programming/Erasing Flash Memory
19.11 Flash Memory Programmer Mode
19.11.1 Socket Adapter Pin Correspondence Diagram
19.11.2 Programmer Mode Operation
19.11.3 Memory Read Mode
19.11.4 Auto-Program Mode
19.11.5 Auto-Erase Mode
19.11.6 Status Read Mode
19.11.7 Status Polling
19.11.8 Programmer Mode Transition Time
19.11.9 Notes on Memory Programming
19.12 Flash Memory and Power-Down States
19.12.1 Note on Power-Down States
19.13 Flash Memory Programming and Erasing Precautions
19.14 Note on Switching from F-ZTAT Version to Mask ROM Version

Section 20 Clock Pulse Generator
20.1 Overview
20.1.1 Block Diagram
20.1.2 Register Configuration
20.2 Register Descriptions
20.2.1 System Clock Control Register (SCKCR)
20.2.2 Low-Power Control Register (LPWRCR)
20.3 Oscillator
20.3.1 Connecting a Crystal Resonator
20.3.2 External Clock Input
20.4 PLL Circuit
20.5 Medium-Speed Clock Divider
20.6 Bus Master Clock Selection Circuit
20.7 Subclock Oscillator (H8S/2626 Group Only)
20.8 Subclock Waveform Shaping Circuit (H8S/2626 Group Only)
20.9 Note on Crystal Resonator

21A.4 Sleep Mode
21A.4.1 Sleep Mode
21A.4.2 Exiting Sleep Mode
21A.5 Module Stop Mode
21A.5.1 Module Stop Mode
21A.5.2 Usage Notes
21A.6 Software Standby Mode
21A.6.1 Software Standby Mode
21A.6.2 Clearing Software Standby Mode
21A.6.3 Setting Oscillation Stabilization Time after Clearing Software Standby Mode
21A.6.4 Software Standby Mode Application Example
21A.6.5 Usage Notes
21A.7 Hardware Standby Mode
21A.7.1 Hardware Standby Mode
21A.7.2 Hardware Standby Mode Timing
21A.8 φ Clock Output Disabling Function

Section 21B Power-Down Modes [H8S/2626 Group]
21B.1 Overview
21B.1.1 Register Configuration
21B.2 Register Descriptions
21B.2.1 Standby Control Register (SBYCR)
21B.2.2 System Clock Control Register (SCKCR)
21B.2.3 Low-Power Control Register (LPWRCR)
21B.2.4 Timer Control/Status Register (TCSR)
21B.2.5 Module Stop Control Register (MSTPCR)
21B.3 Medium-Speed Mode
21B.4 Sleep Mode
21B.4.1 Sleep Mode
21B.4.2 Exiting Sleep Mode
21B.5 Module Stop Mode

21B.7 Hardware Standby Mode
21B.7.1 Hardware Standby Mode
21B.7.2 Hardware Standby Mode Timing
21B.8 Watch Mode
21B.8.1 Watch Mode
21B.8.2 Exiting Watch Mode
21B.8.3 Notes
21B.9 Sub-Sleep Mode
21B.9.1 Sub-Sleep Mode
21B.9.2 Exiting Sub-Sleep Mode
21B.10 Sub-Active Mode
21B.10.1 Sub-Active Mode
21B.10.2 Exiting Sub-Active Mode
21B.11 Direct Transitions
21B.11.1 Overview of Direct Transitions
21B.12 φ Clock Output Disabling Function
21B.13 Usage Notes

Section 22 Electrical Characteristics (Preliminary)
22.1 Absolute Maximum Ratings
22.2 DC Characteristics
22.3 AC Characteristics
22.3.1 Clock Timing
22.3.2 Control Signal Timing
22.3.3 Bus Timing
22.3.4 Timing of On-Chip Supporting Modules
22.4 A/D Conversion Characteristics
22.5 D/A Conversion Characteristics
22.6 Flash Memory Characteristics
22.7 Usage Note

B.1 Address
B.2 Functions

Appendix C I/O Port Block Diagrams
C.1 Port 1 Block Diagrams
C.2 Port 4 Block Diagram
C.3 Port 9 Block Diagram
C.4 Port A Block Diagrams
C.5 Port B Block Diagram
C.6 Port C Block Diagrams
C.7 Port D Block Diagram
C.8 Port E Block Diagram
C.9 Port F Block Diagrams

Appendix D Pin States
D.1 Port States in Each Mode

Appendix E Timing of Transition to and Recovery from Hardware Standby Mode

Appendix F Product Code Lineup

Appendix G Package Dimensions

The H8S/2600 CPU has an internal 32-bit architecture, is provided with sixteen 16-bit registers and a concise, optimized instruction set designed for high-speed operation, a address a 16-Mbyte linear address space. The instruction set is upward-compatible wit and H8/300H CPU instructions at the object-code level, facilitating migration from th H8/300L, or H8/300H Series.

On-chip peripheral functions required for system configuration include a data transfer (DTC) bus master, ROM and RAM, a16-bit timer-pulse unit (TPU), programmable pu generator (PPG), watchdog timer (WDT), serial communication interface (SCI), contr network (HCAN), A/D converter, D/A converter (H8S/2626 Group only), and I/O por

The on-chip ROM is 256-kbyte flash memory (F-ZTAT™)* or 256-, 128-, or 64-kbyt ROM. The ROM is connected to the CPU by a 16-bit data bus, enabling both byte and to be accessed in one state. Instruction fetching has been speeded up, and processing s increased.

Four operating modes, modes 4 to 7, are provided, and there is a choice of single-chip external expansion mode.

The features of the H8S/2626 Group and H8S/2623 Group are shown in table 1.1.

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

	— High-speed arithmetic operations 8/16/32-bit register-register add/subtract: 50 ns 16 × 16-bit register-register multiply: 200 ns 16 × 16 + 42-bit multiply and accumulate: 200 ns 32 ÷ 16-bit register-register divide: 1000 ns • Instruction set suitable for high-speed operation — 69 basic instructions — 8/16/32-bit move/arithmetic and logic instructions — Unsigned/signed multiply and divide instructions — Multiply-and accumulate instruction — Powerful bit-manipulation instructions • Two CPU operating modes — Normal mode: 64-kbyte address space (Not available in the H8S/2626 Group or H8S/2623 Group) — Advanced mode: 16-Mbyte address space
Bus controller	• Address space divided into 8 areas, with bus specifications se independently for each area • Choice of 8-bit or 16-bit access space for each area • 2-state or 3-state access space can be designated for each ar • Number of program wait states can be set for each area • Burst ROM directly connectable • External bus release function
PC break controller	• Supports debugging functions by means of PC break interrupt • Two break channels

	• Pulse input/output processing capability for up to 16 pins • Automatic 2-phase encoder count capability
Programmable pulse generator (PPG)	• Maximum 8-bit pulse output possible with TPU as time base • Output trigger selectable in 4-bit groups • Non-overlap margin can be set • Direct output or inverse output setting
Watchdog timer (WDT), 2 channels (H8S/2626 Group)	• Watchdog timer or interval timer selectable • Subclock operation possible (one channel only)
Watchdog timer (WDT), 1 channel (H8S/2623 Group)	• Watchdog timer or interval timer selectable
Serial communication interface (SCI), 3 channels (SCI0 to SCI2)	• Asynchronous mode or synchronous mode selectable • Multiprocessor communication function • Smart card interface function
Controller area network (HCAN), 1 channel	• CAN: Ver. 2.0B compliant • Buffer size: 15 transmit/receive buffers, one transmit-only buf • Receive message filtering
A/D converter	• Resolution: 10 bits • Input: 16 channels • 13.3 µs minimum conversion time (at 20 MHz operation) • Single or scan mode selectable • Sample-and-hold function • A/D conversion can be activated by external trigger or timer t

Memory	• Flash memory or masked ROM • High-speed static RAM

Product Name	ROM	RAM
H8S/2626, H8S/2623	256 kbytes	12 kbytes
H8S/2625, H8S/2622	128 kbytes	8 kbytes
H8S/2624, H8S/2621	64 kbytes	4 kbytes

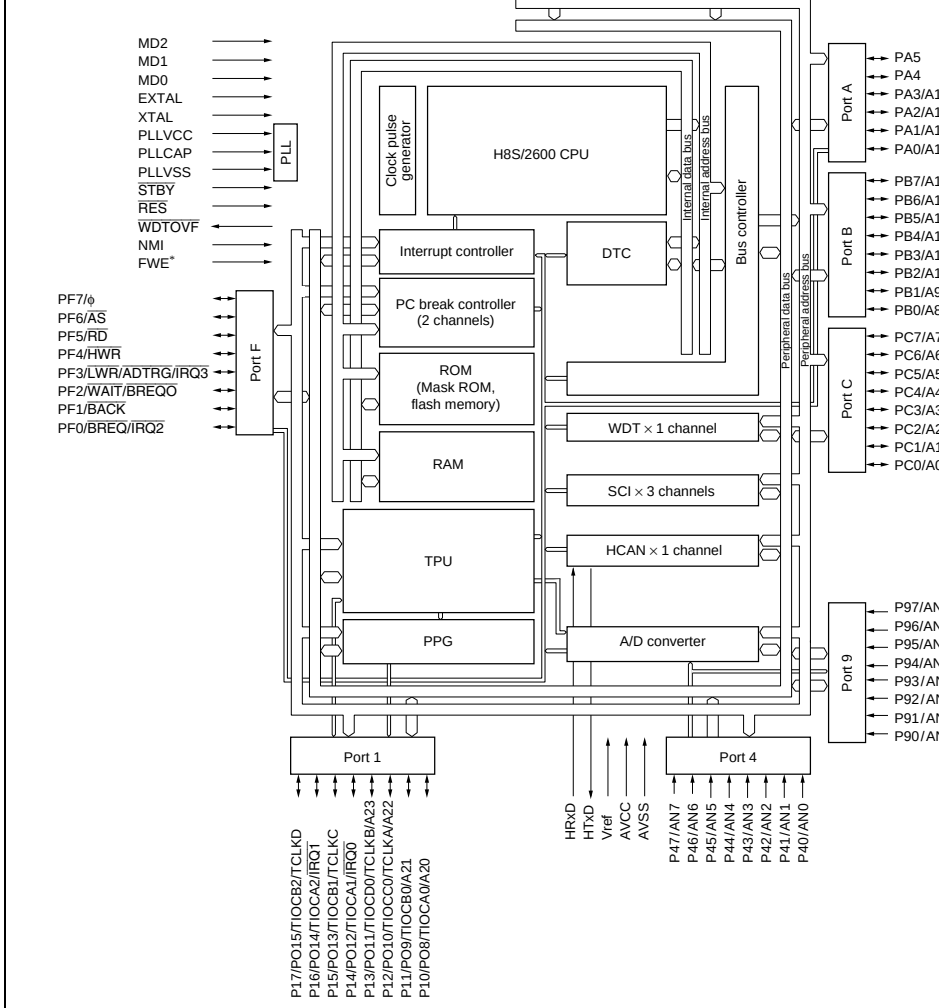
Interrupt controller	• Seven external interrupt pins (NMI, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ5}}$) • Internal interrupt sources H8S/2626: 48 H8S/2623: 47 • Eight priority levels settable
Power-down state	• Medium-speed mode • Sleep mode • Module stop mode • Software standby mode • Hardware standby mode • Subclock operation (H8S/2626 Group only)

		expansion mode		
6		On-chip ROM enabled expansion mode	Enabled	8 bits
7		Single-chip mode	Enabled	—

Clock pulse generator	• Built-in PLL circuit (×1, ×2, ×4) • Input clock frequency: 2 to 20 MHz
Package	• 100-pin plastic QFP (FP-100B)

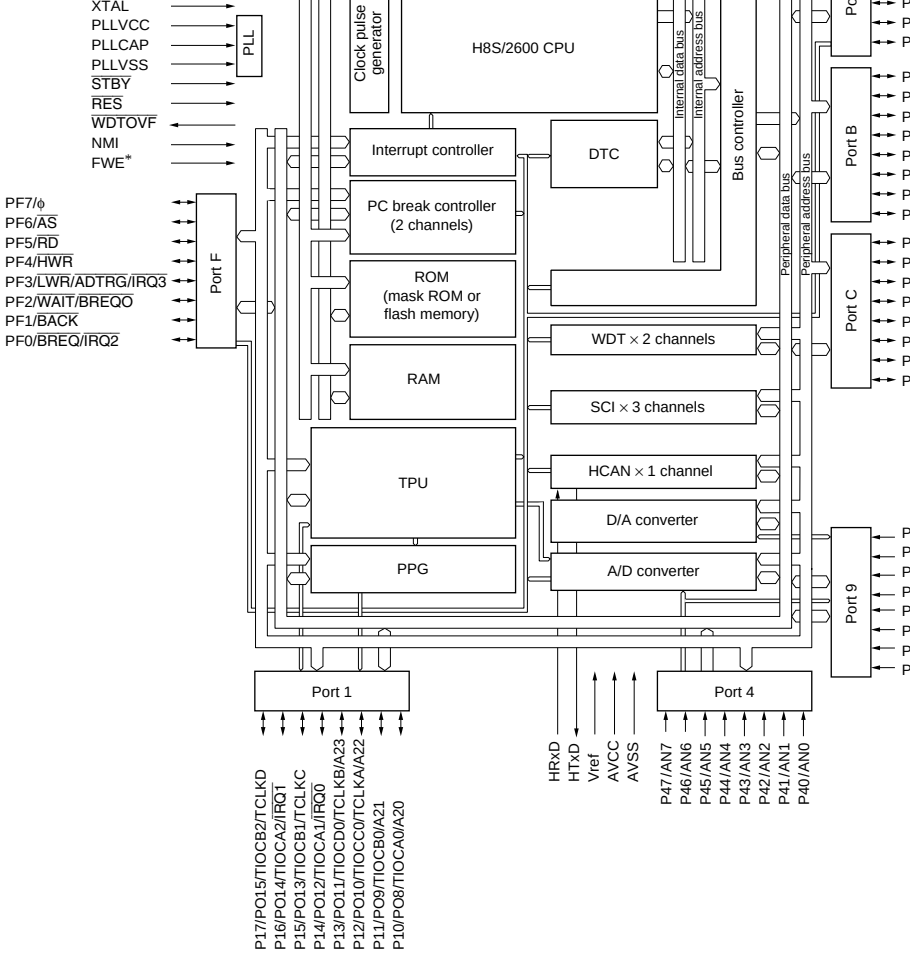
Product lineup

Model		
Mask ROM Version	**F-ZTAT Version**	**ROM/RAM (Bytes)**
HD6432626 HD6432623	HD64F2626 HD64F2623	256 k/12 k
HD6432625 HD6432622	—	128 k/8 k
HD6432624 HD6432621	—	64 k/4 k



Note: * The FWE pin is used only in the flash memory version.

Figure 1.1 Internal Block Diagram (H8S/2623 Group)



Note: * The FWE pin is provided in the flash memory version only.

Figure 1.2 Internal Block Diagram (H8S/2626 Group)

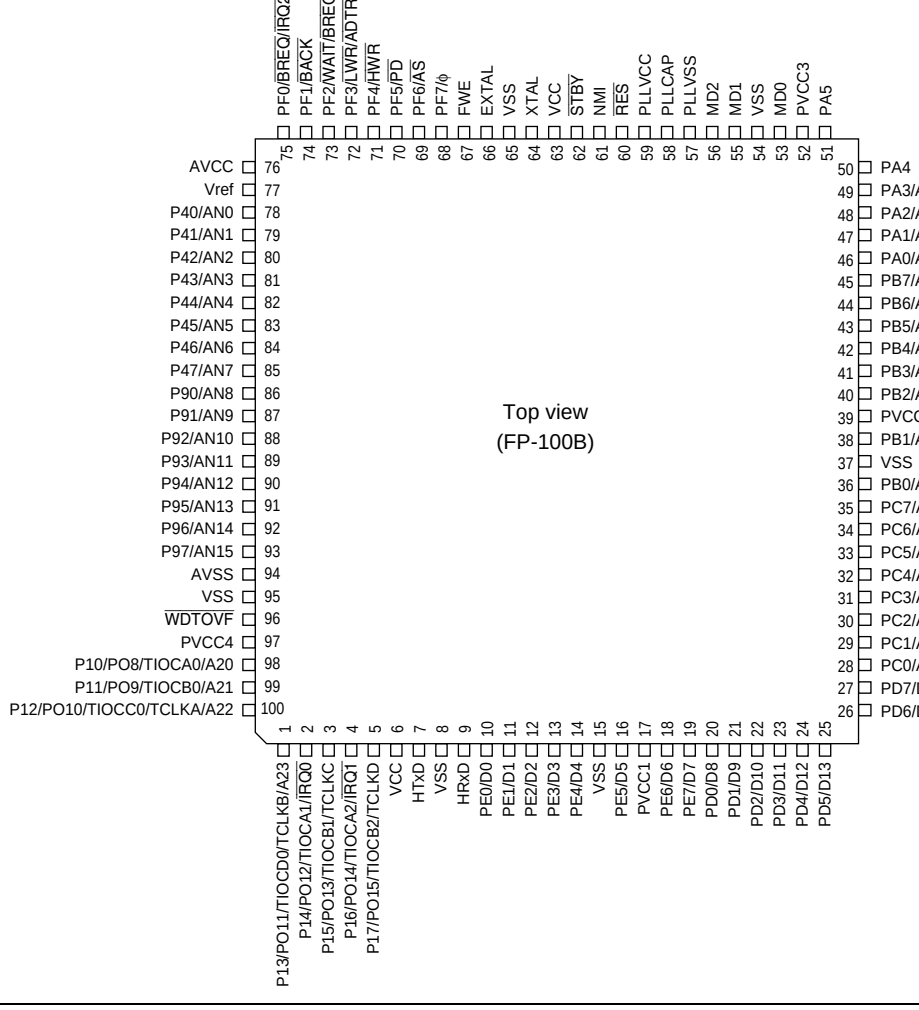


Figure 1.3 Pin Arrangement (FP-100B: Top View) (H8S/2623 Group)

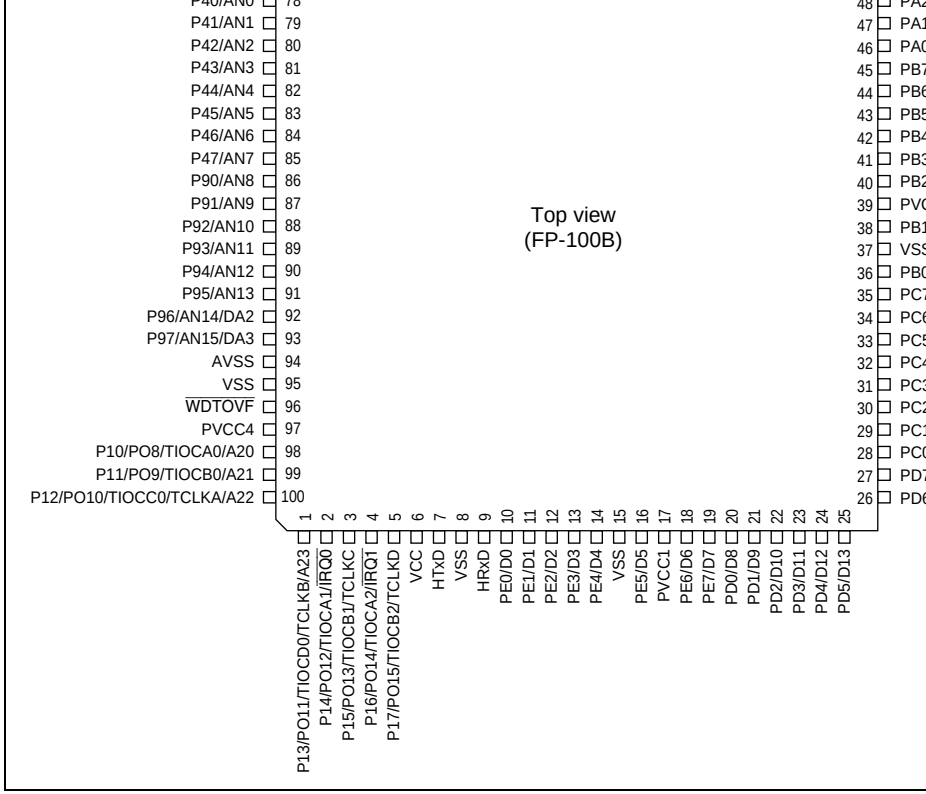


Figure 1.4 Pin Arrangement (FP-100B: Top View) (H8S/2626 Group

1	P13/PO11/TIOCD0/ TCLKB/A23	P13/PO11/TIOCD0/ TCLKB/A23	P13/PO11/TIOCD0/ TCLKB/A23	P13/PO1 TCLKB
2	P14/PO12/TIOCA1/ $\overline{\text{IRQ0}}$	P14/PO12/TIOCA1/ $\overline{\text{IRQ0}}$	P14/PO12/TIOCA1/ $\overline{\text{IRQ0}}$	P14/PO1 $\overline{\text{IRQ0}}$
3	P15/PO13/TIOCB1/ TCLKC	P15/PO13/TIOCB1/ TCLKC	P15/PO13/TIOCB1/ TCLKC	P15/PO1 TCLKC
4	P16/PO14/TIOCA2/ $\overline{\text{IRQ1}}$	P16/PO14/TIOCA2/ $\overline{\text{IRQ1}}$	P16/PO14/TIOCA2/ $\overline{\text{IRQ1}}$	P16/PO1 $\overline{\text{IRQ1}}$
5	P17/PO15/TIOCB2/ TCLKD	P17/PO15/TIOCB2/ TCLKD	P17/PO15/TIOCB2/ TCLKD	P17/PO1 TCLKD
6	VCC	VCC	VCC	VCC
7	HTxD	HTxD	HTxD	HTxD
8	VSS	VSS	VSS	VSS
9	HRxD	HRxD	HRxD	HRxD
10	PE0/D0	PE0/D0	PE0/D0	PE0
11	PE1/D1	PE1/D1	PE1/D1	PE1
12	PE2/D2	PE2/D2	PE2/D2	PE2
13	PE3/D3	PE3/D3	PE3/D3	PE3
14	PE4/D4	PE4/D4	PE4/D4	PE4
15	VSS	VSS	VSS	VSS
16	PE5/D5	PE5/D5	PE5/D5	PE5
17	PVCC1	PVCC1	PVCC1	PVCC1
18	PE6/D6	PE6/D6	PE6/D6	PE6
19	PE7/D7	PE7/D7	PE7/D7	PE7
20	D8	D8	D8	PD0
21	D9	D9	D9	PD1
22	D10	D10	D10	PD2

28	A0	A0	PC0/A0/TxD0	PC0/TxI
29	A1	A1	PC1/A1/RxD0	PC1/Rx
30	A2	A2	PC2/A2/SCK0/$\overline{\text{IRQ4}}$	PC2/SC
31	A3	A3	PC3/A3/TxD1	PC3/TxI
32	A4	A4	PC4/A4/RxD1	PC4/Rx
33	A5	A5	PC5/A5/SCK1/$\overline{\text{IRQ5}}$	PC5/SC
34	A6	A6	PC6/A6	PC6
35	A7	A7	PC7/A7	PC7
36	PB0/A8/TIOCA3	PB0/A8/TIOCA3	PB0/A8/TIOCA3	PB0/TIO
37	VSS	VSS	VSS	VSS
38	PB1/A9/TIOCB3	PB1/A9/TIOCB3	PB1/A9/TIOCB3	PB1/TIO
39	PVCC2	PVCC2	PVCC2	PVCC2
40	PB2/A10/TIOCC3	PB2/A10/TIOCC3	PB2/A10/TIOCC3	PB2/TIO
41	PB3/A11/TIOCD3	PB3/A11/TIOCD3	PB3/A11/TIOCD3	PB3/TIO
42	PB4/A12/TIOCA4	PB4/A12/TIOCA4	PB4/A12/TIOCA4	PB4/TIO
43	PB5/A13/TIOCB4	PB5/A13/TIOCB4	PB5/A13/TIOCB4	PB5/TIO
44	PB6/A14/TIOCA5	PB6/A14/TIOCA5	PB6/A14/TIOCA5	PB6/TIO
45	PB7/A15/TIOCB5	PB7/A15/TIOCB5	PB7/A15/TIOCB5	PB7/TIO
46	PA0/A16	PA0/A16	PA0/A16	PA0
47	PA1/A17/TxD2	PA1/A17/TxD2	PA1/A17/TxD2	PA1/TxD
48	PA2/A18/RxD2	PA2/A18/RxD2	PA2/A18/RxD2	PA2/RxI
49	PA3/A19/SCK2	PA3/A19/SCK2	PA3/A19/SCK2	PA3/SC
50	PA4	PA4	PA4	PA4
51	PA5	PA5	PA5	PA5
52	PVCC3	PVCC3	PVCC3	PVCC3
53	MD0	MD0	MD0	MD0

59	PLLVCC	PLLVCC	PLLVCC	PLLVCC
60	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$
61	NMI	NMI	NMI	NMI
62	$\overline{\text{STBY}}$	$\overline{\text{STBY}}$	$\overline{\text{STBY}}$	$\overline{\text{STBY}}$
63	VCC	VCC	VCC	VCC
64	XTAL	XTAL	XTAL	XTAL
65	VSS	VSS	VSS	VSS
66	EXTAL	EXTAL	EXTAL	EXTAL
67	FWE	FWE	FWE	FWE
68	PF7/ϕ	PF7/ϕ	PF7/ϕ	PF7/ϕ
69	$\overline{\text{AS}}$	$\overline{\text{AS}}$	$\overline{\text{AS}}$	PF6
70	$\overline{\text{RD}}$	$\overline{\text{RD}}$	$\overline{\text{RD}}$	PF5
71	$\overline{\text{HWR}}$	$\overline{\text{HWR}}$	$\overline{\text{HWR}}$	PF4
72	PF3/$\overline{\text{LWR}}$/ADTRG/ $\overline{\text{IRQ3}}$	PF3/$\overline{\text{LWR}}$/ADTRG/ $\overline{\text{IRQ3}}$	PF3/$\overline{\text{LWR}}$/ADTRG/ $\overline{\text{IRQ3}}$	PF3/ADT $\overline{\text{IRQ3}}$
73	PF2/$\overline{\text{WAIT}}$/$\overline{\text{BREQO}}$	PF2/$\overline{\text{WAIT}}$/$\overline{\text{BREQO}}$	PF2/$\overline{\text{WAIT}}$/$\overline{\text{BREQO}}$	PF2
74	PF1/$\overline{\text{BACK}}$	PF1/$\overline{\text{BACK}}$	PF1/$\overline{\text{BACK}}$	PF1
75	PF0/$\overline{\text{BREQ}}$/$\overline{\text{IRQ2}}$	PF0/$\overline{\text{BREQ}}$/$\overline{\text{IIRQ2}}$	PF0/$\overline{\text{BREQ}}$/$\overline{\text{IIRQ2}}$	PF0/$\overline{\text{IRQ2}}$
76	AVCC	AVCC	AVCC	AVCC
77	Vref	Vref	Vref	Vref
78	P40/AN0	P40/AN0	P40/AN0	P40/AN0
79	P41/AN1	P41/AN1	P41/AN1	P41/AN1
80	P42/AN2	P42/AN2	P42/AN2	P42/AN2
81	P43/AN3	P43/AN3	P43/AN3	P43/AN3
82	P44/AN4	P44/AN4	P44/AN4	P44/AN4
83	P45/AN5	P45/AN5	P45/AN5	P45/AN5

89	P93/AN11	P93/AN11	P93/AN11	P93/AN
90	P94/AN12	P94/AN12	P94/AN12	P94/AN
91	P95/AN13	P95/AN13	P95/AN13	P95/AN
92	P96/AN14	P96/AN14	P96/AN14	P96/AN
93	P97/AN15	P97/AN15	P97/AN15	P97/AN
94	AVSS	AVSS	AVSS	AVSS
95	VSS	VSS	VSS	VSS
96	$\overline{\text{WDTOVF}}$	$\overline{\text{WDTOVF}}$	$\overline{\text{WDTOVF}}$	$\overline{\text{WDTOV}}$
97	PVCC4	PVCC4	PVCC4	PVCC4
98	P10/PO8/TIOCA0/ A20	P10/PO8/TIOCA0/ A20	P10/PO8/TIOCA0/ A20	P10/PO
99	P11/PO9/TIOCB0/ A21	P11/PO9/TIOCB0/ A21	P11/PO9/TIOCB0/ A21	P11/PO
100	P12/PO10/TIOCC0/ TCLKA/A22	P12/PO10/TIOCC0/ TCLKA/A22	P12/PO10/TIOCC0/ TCLKA/A22	P12/PO TCLKA

Note: NC pins should be connected to VSS or left open.

3	P15/PO13/TIOCB1/ TCLKC	P15/PO13/TIOCB1/ TCLKC	P15/PO13/TIOCB1/ TCLKC	P15/PO1 TCLKC
4	P16/PO14/TIOCA2/ $\overline{\text{IRQ1}}$	P16/PO14/TIOCA2/ $\overline{\text{IRQ1}}$	P16/PO14/TIOCA2/ $\overline{\text{IRQ1}}$	P16/PO1 $\overline{\text{IRQ1}}$
5	P17/PO15/TIOCB2/ TCLKD	P17/PO15/TIOCB2/ TCLKD	P17/PO15/TIOCB2/ TCLKD	P17/PO1 TCLKD
6	VCC	VCC	VCC	VCC
7	HTxD	HTxD	HTxD	HTxD
8	VSS	VSS	VSS	VSS
9	HRxD	HRxD	HRxD	HRxD
10	PE0/D0	PE0/D0	PE0/D0	PE0
11	PE1/D1	PE1/D1	PE1/D1	PE1
12	PE2/D2	PE2/D2	PE2/D2	PE2
13	PE3/D3	PE3/D3	PE3/D3	PE3
14	PE4/D4	PE4/D4	PE4/D4	PE4
15	VSS	VSS	VSS	VSS
16	PE5/D5	PE5/D5	PE5/D5	PE5
17	PVCC1	PVCC1	PVCC1	PVCC1
18	PE6/D6	PE6/D6	PE6/D6	PE6
19	PE7/D7	PE7/D7	PE7/D7	PE7
20	D8	D8	D8	PD0
21	D9	D9	D9	PD1
22	D10	D10	D10	PD2
23	D11	D11	D11	PD3
24	D12	D12	D12	PD4
25	D13	D13	D13	PD5
26	D14	D14	D14	PD6

32	A4	A4	PC4/A4/RxD1	PC4/Rx
33	A5	A5	PC5/A5/SCK1/$\overline{\text{IRQ5}}$	PC5/SC
34	A6	A6	PC6/A6	PC6
35	A7	A7	PC7/A7	PC7
36	PB0/A8/TIOCA3	PB0/A8/TIOCA3	PB0/A8/TIOCA3	PB0/TIC
37	VSS	VSS	VSS	VSS
38	PB1/A9/TIOCB3	PB1/A9/TIOCB3	PB1/A9/TIOCB3	PB1/TIC
39	PVCC2	PVCC2	PVCC2	PVCC2
40	PB2/A10/TIOCC3	PB2/A10/TIOCC3	PB2/A10/TIOCC3	PB2/TIC
41	PB3/A11/TIOCD3	PB3/A11/TIOCD3	PB3/A11/TIOCD3	PB3/TIC
42	PB4/A12/TIOCA4	PB4/A12/TIOCA4	PB4/A12/TIOCA4	PB4/TIC
43	PB5/A13/TIOCB4	PB5/A13/TIOCB4	PB5/A13/TIOCB4	PB5/TIC
44	PB6/A14/TIOCA5	PB6/A14/TIOCA5	PB6/A14/TIOCA5	PB6/TIC
45	PB7/A15/TIOCB5	PB7/A15/TIOCB5	PB7/A15/TIOCB5	PB7/TIC
46	PA0/A16	PA0/A16	PA0/A16	PA0
47	PA1/A17/TxD2	PA1/A17/TxD2	PA1/A17/TxD2	PA1/TxD
48	PA2/A18/RxD2	PA2/A18/RxD2	PA2/A18/RxD2	PA2/RxI
49	PA3/A19/SCK2	PA3/A19/SCK2	PA3/A19/SCK2	PA3/SC
50	OSC1	OSC1	OSC1	OSC1
51	OSC2	OSC2	OSC2	OSC2
52	PVCC3	PVCC3	PVCC3	PVCC3
53	MD0	MD0	MD0	MD0
54	VSS	VSS	VSS	VSS
55	MD1	MD1	MD1	MD1
56	MD2	MD2	MD2	MD2
57	PLLVSS	PLLVSS	PLLVSS	PLLVSS

63	VCC	VCC	VCC	VCC
64	XTAL	XTAL	XTAL	XTAL
65	VSS	VSS	VSS	VSS
66	EXTAL	EXTAL	EXTAL	EXTAL
67	FWE	FWE	FWE	FWE
68	PF7/φ	PF7/φ	PF7/φ	PF7/φ
69	$\overline{\text{AS}}$	$\overline{\text{AS}}$	$\overline{\text{AS}}$	PF6
70	$\overline{\text{RD}}$	$\overline{\text{RD}}$	$\overline{\text{RD}}$	PF5
71	$\overline{\text{HWR}}$	$\overline{\text{HWR}}$	$\overline{\text{HWR}}$	PF4
72	PF3/$\overline{\text{LWR}}$/ADTRG/ $\overline{\text{IRQ3}}$	PF3/$\overline{\text{LWR}}$/ADTRG/ $\overline{\text{IRQ3}}$	PF3/$\overline{\text{LWR}}$/ADTRG/ $\overline{\text{IRQ3}}$	PF3/ADT $\overline{\text{IRQ3}}$
73	PF2/$\overline{\text{WAIT}}$/$\overline{\text{BREQO}}$	PF2/$\overline{\text{WAIT}}$/$\overline{\text{BREQO}}$	PF2/$\overline{\text{WAIT}}$/$\overline{\text{BREQO}}$	PF2
74	PF1/$\overline{\text{BACK}}$/BUZZ	PF1/$\overline{\text{BACK}}$/BUZZ	PF1/$\overline{\text{BACK}}$/BUZZ	PF1/BUZ
75	PF0/$\overline{\text{BREQ}}$/$\overline{\text{IRQ2}}$	PF0/$\overline{\text{BREQ}}$/$\overline{\text{IIRQ2}}$	PF0/$\overline{\text{BREQ}}$/$\overline{\text{IIRQ2}}$	PF0/$\overline{\text{IRQ}}$
76	AVCC	AVCC	AVCC	AVCC
77	Vref	Vref	Vref	Vref
78	P40/AN0	P40/AN0	P40/AN0	P40/AN0
79	P41/AN1	P41/AN1	P41/AN1	P41/AN1
80	P42/AN2	P42/AN2	P42/AN2	P42/AN2
81	P43/AN3	P43/AN3	P43/AN3	P43/AN3
82	P44/AN4	P44/AN4	P44/AN4	P44/AN4
83	P45/AN5	P45/AN5	P45/AN5	P45/AN5
84	P46/AN6	P46/AN6	P46/AN6	P46/AN6
85	P47/AN7	P47/AN7	P47/AN7	P47/AN7
86	P90/AN8	P90/AN8	P90/AN8	P90/AN8
87	P91/AN9	P91/AN9	P91/AN9	P91/AN9

93	P97/AN15/DA3	P97/AN15/DA3	P97/AN15/DA3	P97/AN
94	AVSS	AVSS	AVSS	AVSS
95	VSS	VSS	VSS	VSS
96	$\overline{\text{WDTOVF}}$	$\overline{\text{WDTOVF}}$	$\overline{\text{WDTOVF}}$	$\overline{\text{WDTOV}}$
97	PVCC4	PVCC4	PVCC4	PVCC4
98	P10/PO8/TIOCA0/ A20	P10/PO8/TIOCA0/ A20	P10/PO8/TIOCA0/ A20	P10/PO
99	P11/PO9/TIOCB0/ A21	P11/PO9/TIOCB0/ A21	P11/PO9/TIOCB0/ A21	P11/PO
100	P12/PO10/TIOCC0/ TCLKA/A22	P12/PO10/TIOCC0/ TCLKA/A22	P12/PO10/TIOCC0/ TCLKA/A22	P12/PO TCLKA

Note: NC pins should be connected to VSS or left open.

				power supply.
	PVCC1	Input	Port power supply	Port power supply pins. Co
	PVCC2	Input	Port power supply	these pins to the same pow
	PVCC3	Input	Port power supply	
	PVCC4	Input	Port power supply	
	VSS	Input	Ground	For connection to the powe (0 V). Connect all V_{SS} pins t system power supply (0 V).
Clock	PLLVCC	Input	PLL power supply	On-chip PLL oscillator pow
	PLLVSS	Input	PLL ground	On-chip PLL oscillator grou
	PLLCAP	Input	PLL capacitance	On-chip PLL oscillator exte capacitance pin
	XTAL	Input	Crystal	For connection to a crystal For examples of crystal res connection and external clo see section 20, Clock Pulse Generator.
	EXTAL	Input	External clock	For connection to a crystal For examples of crystal res connection and external clo see section 20, Clock Pulse Generator.
	OSC1*1	Input	Subclock	For connection to a recomm 32.768 kHz resonator. For of crystal resonator connec section 20, Clock Pulse Ge
	OSC2*1	Input	Subclock	For connection to a recomm 32.768 kHz resonator. For of crystal resonator connec section 20, Clock Pulse Ge
	ϕ	Output	System clock	Supplies the system clock t devices.

			1	—
	1		0	—
			1	—
1	0		0	Mod
			1	Mod
	1		0	Mod
			1	Mod

System control	$\overline{\text{RES}}$	Input	Reset input	When this pin is driven lov reset.
	$\overline{\text{STBY}}$	Input	Standby	When this pin is driven lov transition is made to hardv mode.
	$\overline{\text{BREQ}}$	Input	Bus request	Used by an external bus m issue a bus request to the
	$\overline{\text{BREQO}}$	Output	Bus request output	External bus request signa an internal bus master acc external space in the exter released state.
	$\overline{\text{BACK}}$	Output	Bus request acknowledge	Indicates that the bus has released to an external bu
	FWE	Input	Flash write enable	Pin for use by flash memo
Interrupts	NMI	Input	Nonmaskable interrupt	Requests a nonmaskable this pin is not used, it shou high.
	$\overline{\text{IRQ5}}$ to $\overline{\text{IRQ0}}$	Input	Interrupt request 5 to 0	These pins request a mas interrupt.
Address bus	A23 to A0	Output	Address bus	These pins output address

				external address space; ind data on the upper data bus D8).
	$\overline{\text{LWR}}$	Output	Low write	Strobe signal indicating writ external address space; ind data on the lower data bus
	$\overline{\text{WAIT}}$	Input	Wait	Requests insertion of wait s bus cycles during access to external address space.
16-bit timer-pulse unit (TPU)	TCLKD to TCLKA	Input	Clock input D to A	These pins input an externa
	TIOCA0, TIOCB0, TIOCC0, TIOCD0	Input/ output	Input capture/ output compare match A0 to D0	The TGR0A to TGR0D inpu input/output compare outpu output pins
	TIOCA1, TIOCB1	Input/ output	Input capture/ output compare match A1 and B1	The TGR1A and TGR1B inp input/output compare outpu output pins
	TIOCA2, TIOCB2	Input/ output	Input capture/ output compare match A2 and B2	The TGR2A and TGR2B inp input/output compare outpu output pins
	TIOCA3, TIOCB3, TIOCC3, TIOCD3	Input/ output	Input capture/ output compare match A3 to D3	The TGR3A to TGR3D inpu input/output compare outpu output pins
	TIOCA4, TIOCB4	Input/ output	Input capture/ output compare match A4 and B4	The TGR4A and TGR4B inp input/output compare outpu output pins
	TIOCA5, TIOCB5	Input/ output	Input capture/ output compare match A5 and B5	The TGR5A and TGR5B inp input/output compare outpu output pins

communication interface (SCI)/ smart card interface	TxD0			
	RxD2, RxD1, RxD0	Input	Receive data	Data input pins
	SCK2, SCK1, SCK0	Input/ output	Serial clock	Clock input/output pins
Controller area network (HCAN)	HTxD	Output	HCAN transmit data	The CAN bus transmissio
	HRxD	Input	HCAN receive data	The CAN bus reception pi
A/D converter	AN15 to AN0	Input	Analog 15 to 0	Analog input pins
	$\overline{\text{ADTRG}}$	Input	A/D conversion external trigger input	Pin for input of an externa start A/D conversion
D/A converter pin	DA3, DA2	Output	Analog output	D/A converter analog outp
A/D converter/ D/A converter	AVCC	Input	Analog power supply	The power supply pin for t D/A converters. When the D/A converters are not us this pin to the system pow (+5 V).
	AVSS	Input	Analog ground	The ground pin and refere for the A/D and D/A conve Connect this pin to the sys supply (0 V).
	Vref	Input	Analog reference power supply	The reference voltage inpu A/D and D/A converters. V A/D and D/A converters ar connect this pin to the sys supply (+5 V).

PA5 to PA0[*2]	Input/ output	Port A	Six input/output pins. Input can be selected for each pi port A data direction registe (PADDR).
PB7 to PB0	Input/ output	Port B	Eight input/output pins. Inpu can be selected for each pi port B data direction registe (PBDDR).
PC7 to PC0	Input/ output	Port C	Eight input/output pins. Inpu can be selected for each pi port C data direction registe (PCDDR).
PD7 to PD0	Input/ output	Port D	Eight input/output pins. Inpu can be selected for each pi port D data direction registe (PDDDR).
PE7 to PE0	Input/ output	Port E	Eight input/output pins. Inpu can be selected for each pi port E data direction registe (PEDDR).
PF7 to PF0	Input/ output	Port F	Eight input/output pins. Inpu can be selected for each pi port F data direction registe

Notes: 1. Applies to the H8S/2626 Group only.
2. PA3 to PA0 in the H8S/2626 Group.

ideal for realtime control.

2.1.1 Features

The H8S/2600 CPU has the following features.

- Upward-compatible with H8/300 and H8/300H CPUs
  - Can execute H8/300 and H8/300H object programs
- General-register architecture
  - Sixteen 16-bit general registers (also usable as sixteen 8-bit registers or eight 3 registers)
- Sixty-nine basic instructions
  - 8/16/32-bit arithmetic and logic instructions
  - Multiply and divide instructions
  - Powerful bit-manipulation instructions
  - Multiply-and-accumulate instruction
- Eight addressing modes
  - Register direct [Rn]
  - Register indirect [@ERn]
  - Register indirect with displacement [@(d:16,ERn) or @(d:32,ERn)]
  - Register indirect with post-increment or pre-decrement [@ERn+ or @–ERn]
  - Absolute address [@aa:8, @aa:16, @aa:24, or @aa:32]
  - Immediate [#xx:8, #xx:16, or #xx:32]
  - Program-counter relative [@(d:8,PC) or @(d:16,PC)]
  - Memory indirect [@@aa:8]
- 16-Mbyte address space
  - Program: 16 Mbytes
  - Data: 16 Mbytes (4 Gbytes architecturally)

— 32 ÷ 16-bit register-register divide: 1000 ns

- Two CPU operating modes
  — Normal mode*
  — Advanced mode

Note: * Not available in the H8S/2626 Group or H8S/2623 Group.

- Power-down state
  — Transition to power-down state by SLEEP instruction
  — CPU clock speed selection

2.1.2 Differences between H8S/2600 CPU and H8S/2000 CPU

The differences between the H8S/2600 CPU and the H8S/2000 CPU are as shown belo

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

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

— Eight 16-bit expanded registers, and one 8-bit and two 32-bit control registers, added.

- Expanded address space
  - — Normal mode* supports the same 64-kbyte address space as the H8/300 CPU.
  - — Advanced mode supports a maximum 16-Mbyte address space.

Note: * Not available in the H8S/2626 Group or H8S/2623 Group.

- Enhanced addressing
  - — The addressing modes have been enhanced to make effective use of the 16-Mb space.
- Enhanced instructions
  - — Addressing modes of bit-manipulation instructions have been enhanced.
  - — Signed multiply and divide instructions have been added.
  - — A multiply-and-accumulate instruction has been added.
  - — Two-bit shift instructions have been added.
  - — Instructions for saving and restoring multiple registers have been added.
  - — A test and set instruction has been added.
- Higher speed
  - — Basic instructions execute twice as fast.

2.1.4 Differences from H8/300H CPU

In comparison to the H8/300H CPU, the H8S/2600 CPU has the following enhanceme

- Additional control register
  - — One 8-bit and two 32-bit control registers have been added.
- Enhanced instructions
  - — Addressing modes of bit-manipulation instructions have been enhanced.
  - — A multiply-and-accumulate instruction has been added.
  - — Two-bit shift instructions have been added.

maximum 64-kbyte address space. Advanced mode supports a maximum 16-Mbyte tot
space (architecturally a maximum 16-Mbyte program area and a maximum of 4 Gbytes
program and data areas combined). The mode is selected by the mode pins of the micro

Note: * Not available in the H8S/2626 Group or H8S/2623 Group.

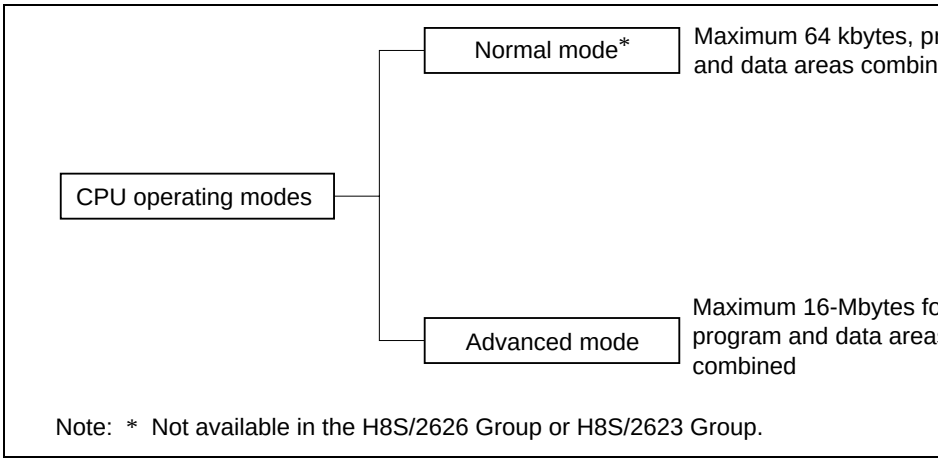


Figure 2.1 CPU Operating Modes

(1) Normal Mode (Not Available in the H8S/2626 Group or H8S/2623 Group)

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

Address Space: A maximum address space of 64 kbytes can be accessed.

Extended Registers (En): The extended registers (E0 to E7) can be used as 16-bit regi
the upper 16-bit segments of 32-bit registers. When En is used as a 16-bit register it can
any value, even when the corresponding general register (Rn) is used as an address regi
general register is referenced in the register indirect addressing mode with pre-decreme

of the exception vector table, see section 4, Exception Handling.

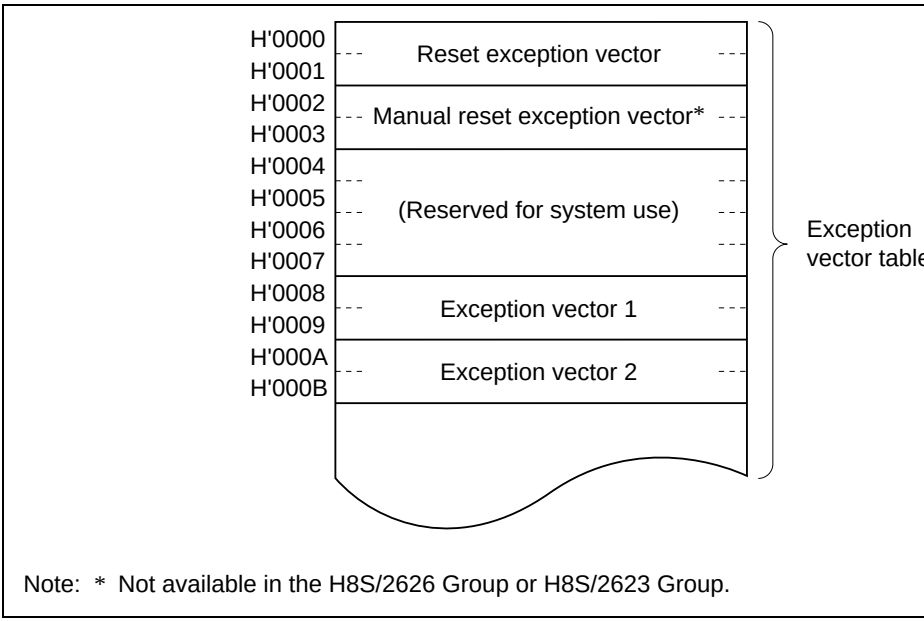
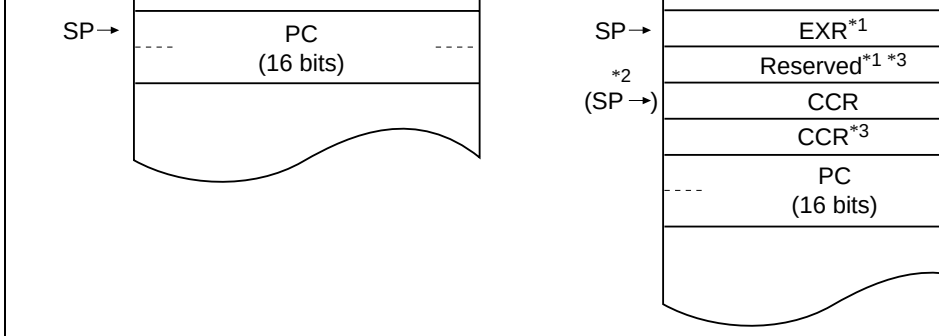


Figure 2.2 Exception Vector Table (Normal Mode)

The memory indirect addressing mode (@@aa:8) employed in the JMP and JSR instru an 8-bit absolute address included in the instruction code to specify a memory operand contains a branch address. In normal mode the operand is a 16-bit word operand, prov bit branch address. Branch addresses can be stored in the top area from H'0000 to H'0 that this area is also used for the exception vector table.



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

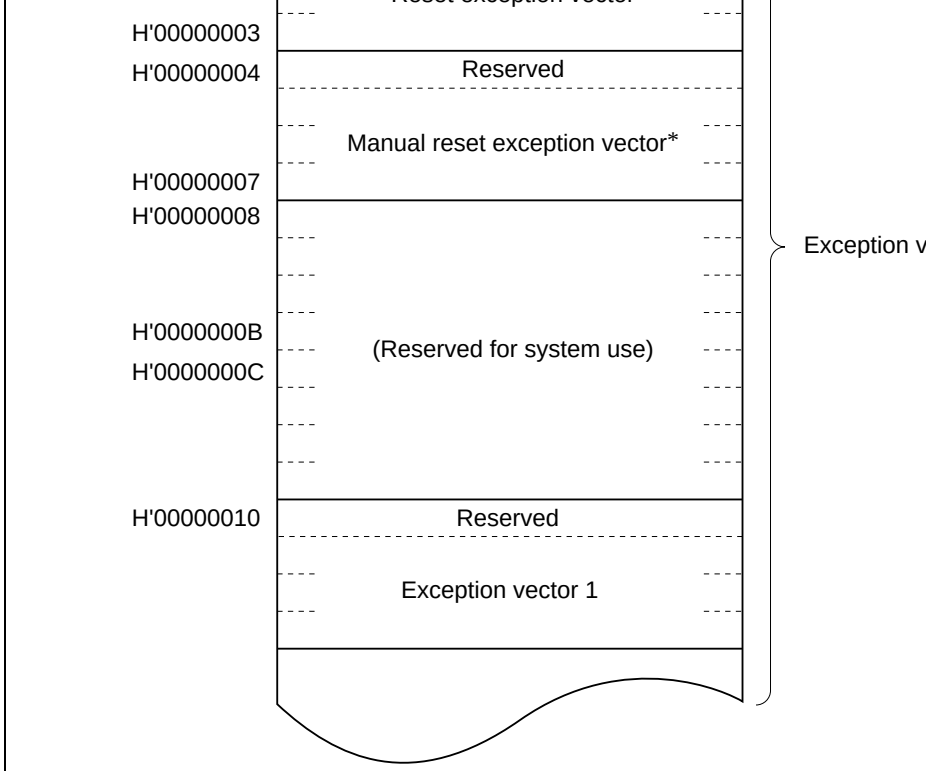
Figure 2.3 Stack Structure in Normal Mode

(2) Advanced Mode

Address Space: Linear access is provided to a 16-Mbyte maximum address space (arch a maximum 16-Mbyte program area and a maximum 4-Gbyte data area, with a maximu Gbytes for program and data areas combined).

Extended Registers (En): The extended registers (E0 to E7) can be used as 16-bit regi the upper 16-bit segments of 32-bit registers or address registers.

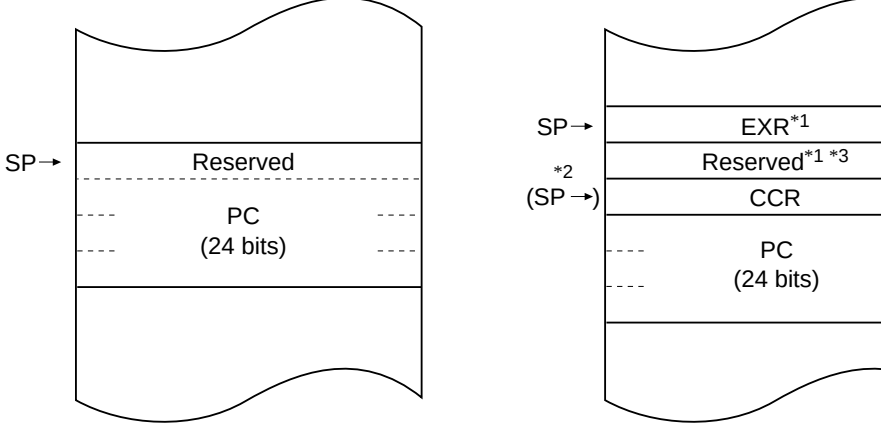
Instruction Set: All instructions and addressing modes can be used.



Note: * Not available in the H8S/2626 Group or H8S/2623 Group.

Figure 2.4 Exception Vector Table (Advanced Mode)

The memory indirect addressing mode (@@aa:8) employed in the JMP and JSR instru
an 8-bit absolute address included in the instruction code to specify a memory operand
contains a branch address. In advanced mode the operand is a 32-bit longword operan
a 32-bit branch address. The upper 8 bits of these 32 bits are a reserved area that is reg



(a) Subroutine Branch **(b) Exception Handlin**

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

Figure 2.5 Stack Structure in Advanced Mode

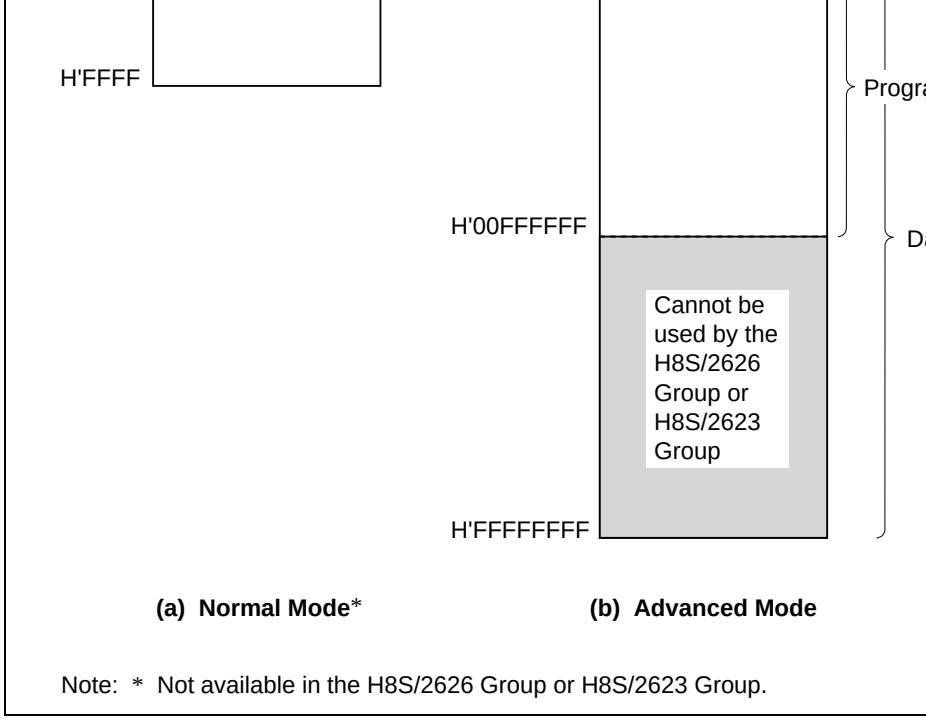
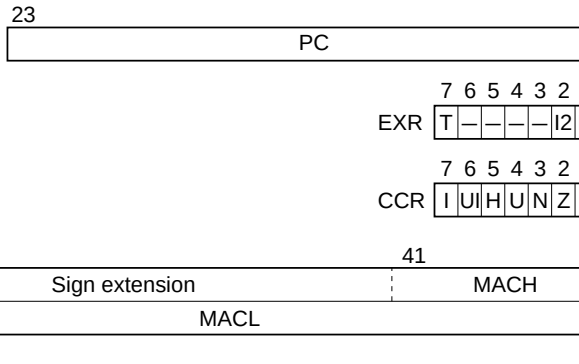


Figure 2.6 Memory Map

General Registers (Rn) and Extended Registers (En)

	15	07	07
ER0	E0	R0H	R0L
ER1	E1	R1H	R1L
ER2	E2	R2H	R2L
ER3	E3	R3H	R3L
ER4	E4	R4H	R4L
ER5	E5	R5H	R5L
ER6	E6	R6H	R6L
ER7 (SP)	E7	R7H	R7L

Control Registers (CR)



Legend:

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

Note: * Cannot be used as an interrupt mask bit in the H8S/2626 Group or H8S/2623 G

Figure 2.7 CPU Registers

registers. The E registers (E0 to E7) are also referred to as extended registers.

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

Figure 2.8 illustrates the usage of the general registers. The usage of each register can
independently.

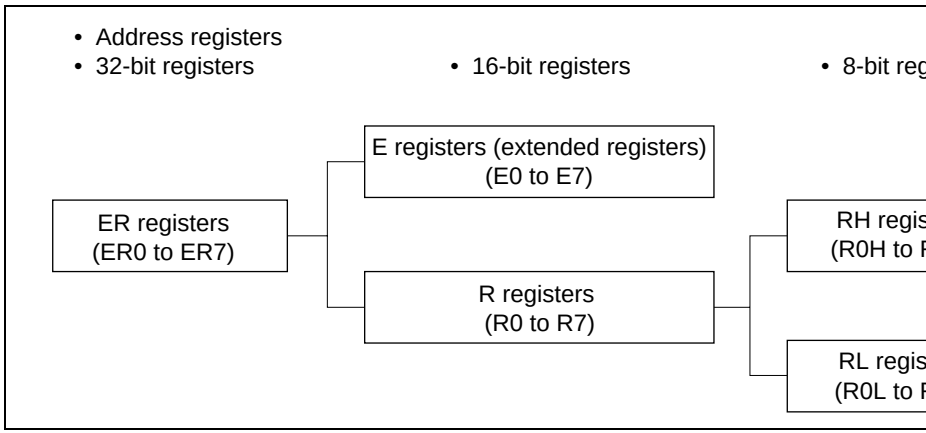


Figure 2.8 Usage of General Registers

General register ER7 has the function of stack pointer (SP) in addition to its general-r
function, and is used implicitly in exception handling and subroutine calls. Figure 2.9
stack.

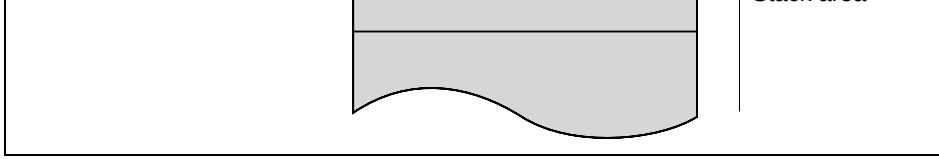


Figure 2.9 Stack

2.4.3 Control Registers

The control registers are the 24-bit program counter (PC), 8-bit extended control registe
8-bit condition-code register (CCR), and 64-bit multiply-accumulate register (MAC).

(1) Program Counter (PC)

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

(2) Extended Control Register (EXR)

This 8-bit register contains the trace bit (T) and three interrupt mask bits (I2 to I0).

Bit 7—Trace Bit (T): Selects trace mode. When this bit is cleared to 0, instructions are
in sequence. When this bit is set to 1, a trace exception is generated each time an instru
executed.

Bits 6 to 3—Reserved: They are always read as 1.

Bits 2 to 0—Interrupt Mask Bits (I2 to I0): These bits designate the interrupt mask le
7). For details, refer to section 5, Interrupt Controller.

Operations can be performed on the EXR bits by the LDC, STC, ANDC, ORC, and XC
instructions. All interrupts, including NMI, are disabled for three states after one of the
instructions is executed, except for STC.

Bit 6—User Bit or Interrupt Mask Bit (UI): Can be written and read by software us
LDC, STC, ANDC, ORC, and XORC instructions. This bit can also be used as an inte
bit. For details, refer to section 5, Interrupt Controller.

Bit 5—Half-Carry Flag (H): When the ADD.B, ADDX.B, SUB.B, SUBX.B, CMP.
instruction is executed, this flag is set to 1 if there is a carry or borrow at bit 3, and cle
otherwise. When the ADD.W, SUB.W, CMP.W, or NEG.W instruction is executed, t
set to 1 if there is a carry or borrow at bit 11, and cleared to 0 otherwise. When the AD
SUB.L, CMP.L, or NEG.L instruction is executed, the H flag is set to 1 if there is a ca
borrow at bit 27, and cleared to 0 otherwise.

Bit 4—User Bit (U): Can be written and read by software using the LDC, STC, AND
XORC instructions.

Bit 3—Negative Flag (N): Stores the value of the most significant bit (sign bit) of dat

Bit 2—Zero Flag (Z): Set to 1 to indicate zero data, and cleared to 0 to indicate non-z

Bit 1—Overflow Flag (V): Set to 1 when an arithmetic overflow occurs, and cleared
times.

Bit 0—Carry Flag (C): Set to 1 when a carry occurs, and cleared to 0 otherwise. Use

- Add instructions, to indicate a carry
- Subtract instructions, to indicate a borrow
- Shift and rotate instructions, to store the value shifted out of the end bit

The carry flag is also used as a bit accumulator by bit manipulation instructions.

Some instructions leave some or all of the flag bits unchanged. For the action of each
on the flag bits, refer to appendix A.1, Instruction List.

2.4.4 Initial Register Values

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

Figure 2.10 shows the data formats in general registers.

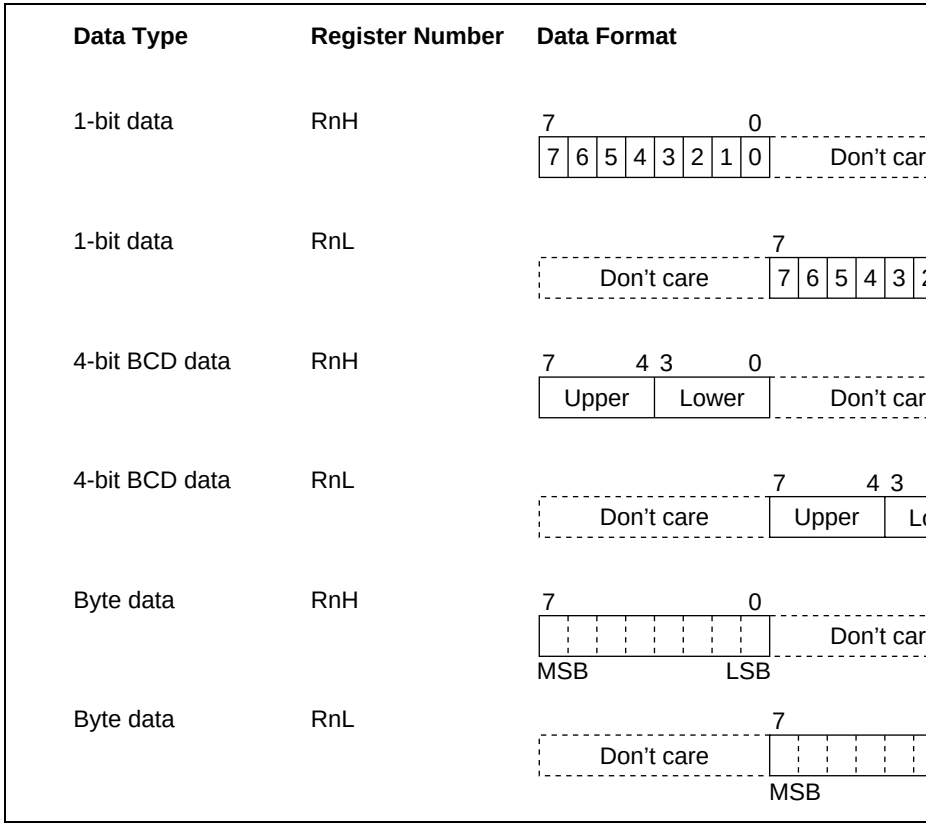
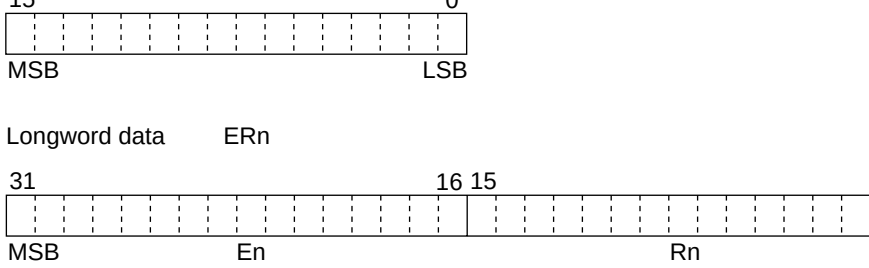


Figure 2.10 General Register Data Formats



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 2.10 General Register Data Formats (cont)

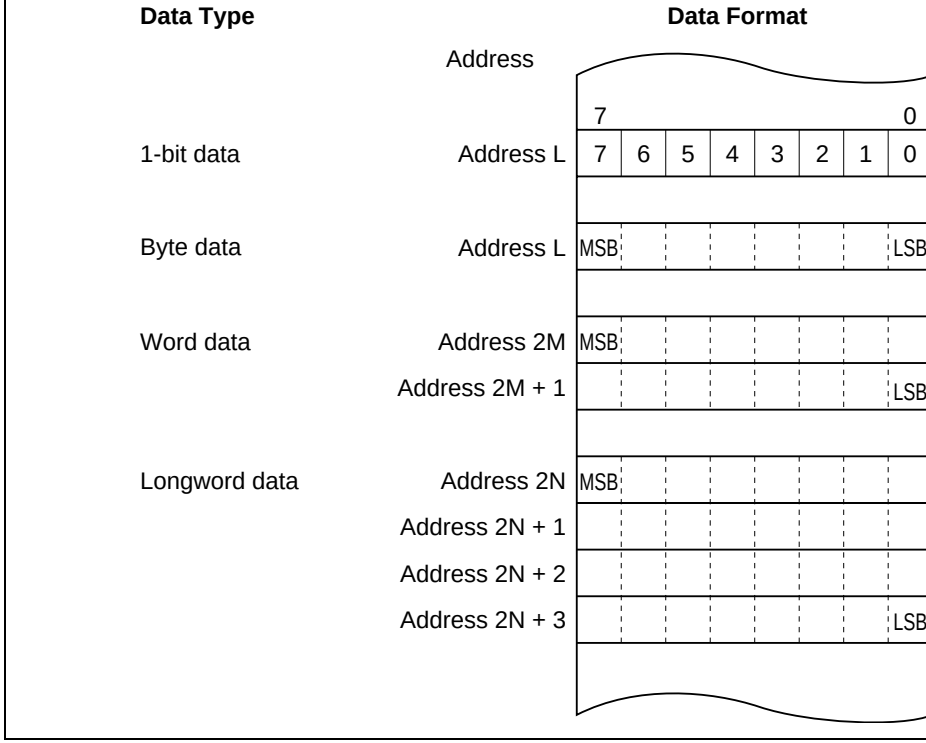


Figure 2.11 Memory Data Formats

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

Function	Instructions	Siz
Data transfer	MOV	BW
	POP*1, PUSH*1	WL
	LDM, STM	L
	MOVFPE*3, MOVTPE*3	B
Arithmetic operations	ADD, SUB, CMP, NEG	BW
	ADDX, SUBX, DAA, DAS	B
	INC, DEC	BW
	ADDS, SUBS	L
	MULXU, DIVXU, MULXS, DIVXS	BW
	EXTU, EXTS	WL
	TAS*4	B
	MAC, LDMAC, STMAC, CLRMAC	—
Logic operations	AND, OR, XOR, NOT	BW
Shift	SHAL, SHAR, SHLL, SHLR, ROTL, ROTR, ROTXL, ROTXR	BW
Bit manipulation	BSET, BCLR, BNOT, BTST, BLD, BILD, BST, BIST, BAND, BIAND, BOR, BIOR, BXOR, BIXOR	B
Branch	Bcc*2, JMP, BSR, JSR, RTS	—
System control	TRAPA, RTE, SLEEP, LDC, STC, ANDC, ORC, XORC, NOP	—
Block data transfer	EEPMOV	—

T

Legend: B: Byte
W: Word
L: Longword

Notes: 1. POP.W Rn and PUSH.W Rn are identical to MOV.W @SP+, Rn and MOV.W @-SP. POP.L ERn and PUSH.L ERn are identical to MOV.L @SP+, ERn an ERn, @-SP.
2. Bcc is the general name for conditional branch instructions.
3. Not available in the H8S/2626 Group or H8S/2623 Group.
4. Only register ER0, ER1, ER4, or ER5 should be used when using the TAS in

Function	Instruction	#xx	Rn	@ERn	@(d:16,ER	@(d:32,ER	@–ERn/@	@aa:8	@aa:16	@aa:24	@aa:32	@(d:8,PC)	@(d:16,PC
Data transfer	MOV	BWL	BWL	BWL	BWL	BWL	BWL	B	BWL	—	BWL	—	—
	POP, PUSH	—	—	—	—	—	—	—	—	—	—	—	—
	LDM, STM	—	—	—	—	—	—	—	—	—	—	—	—
	MOVEPE[*1], MOVTPE[*1]	—	—	—	—	—	—	—	B	—	—	—	—
Arithmetic operations	ADD, CMP	BWL	BWL	—	—	—	—	—	—	—	—	—	—
	SUB	WL	BWL	—	—	—	—	—	—	—	—	—	—
	ADDX, SUBX	B	B	—	—	—	—	—	—	—	—	—	—
	ADDS, SUBS	—	L	—	—	—	—	—	—	—	—	—	—
	INC, DEC	—	BWL	—	—	—	—	—	—	—	—	—	—
	DAA, DAS	—	B	—	—	—	—	—	—	—	—	—	—
	MULXU, DIVXU	—	BW	—	—	—	—	—	—	—	—	—	—
	MULXS, DIVXS	—	BW	—	—	—	—	—	—	—	—	—	—
	NEG	—	BWL	—	—	—	—	—	—	—	—	—	—
	EXTU, EXTS	—	WL	—	—	—	—	—	—	—	—	—	—
	TAS[*2]	—	—	B	—	—	—	—	—	—	—	—	—
	MAC	—	—	—	—	—	○	—	—	—	—	—	—
	CLRMAC	—	—	—	—	—	—	—	—	—	—	—	—
	LDMAC, STMAC	—	L	—	—	—	—	—	—	—	—	—	—
Logic operations	AND, OR, XOR	BWL	BWL	—	—	—	—	—	—	—	—	—	—
	NOT	—	BWL	—	—	—	—	—	—	—	—	—	—
Shift		—	BWL	—	—	—	—	—	—	—	—	—	—
Bit manipulation		—	B	B	—	—	—	B	B	—	B	—	—
Branch	Bcc, BSR	—	—	—	—	—	—	—	—	—	—	○	○
	JMP, JSR	—	—	—	—	—	—	—	—	○	—	—	—
	RTS	—	—	—	—	—	—	—	—	—	—	—	—
System control	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		—	—	—	—	—	—	—	—	—	—	—	—

Legend: B: Byte
W: Word
L: Longword

Notes: 1. Not available in the H8S/2626 Group or H8S/2623 Group.
2. Only register ER0, ER1, ER4, or ER5 should be used when using the TAS instruction.

RENESAS

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

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

MOVTPE	B	Cannot be used in the H8S/2626 Group or H Group.
POP	W/L	@SP+ → Rn Pops a register from the stack. POP.W Rn is MOV.W @SP+, Rn. POP.L ERn is identical @SP+, ERn.
PUSH	W/L	Rn → @–SP Pushes a register onto the stack. PUSH.W R identical to MOV.W Rn, @–SP. PUSH.L ER to MOV.L ERn, @–SP.
LDM	L	@SP+ → Rn (register list) Pops two or more general registers from the
STM	L	Rn (register list) → @–SP Pushes two or more general registers onto t

		Performs addition or subtraction with carry or byte data in two general registers, or on imme and data in a general register.
INC DEC	B/W/L	Rd ± 1 → Rd, Rd ± 2 → Rd Increments or decrements a general register (Byte operands can be incremented or decrer 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 fro 32-bit register.
DAA DAS	B	Rd decimal adjust → Rd Decimal-adjusts an addition or subtraction res general register by referring to the CCR to pro BCD data.
MULXU	B/W	Rd × Rs → Rd Performs unsigned multiplication on data in tv registers: either 8 bits × 8 bits → 16 bits or 16 16 bits → 32 bits.
MULXS	B/W	Rd × Rs → Rd Performs signed multiplication on data in two registers: either 8 bits × 8 bits → 16 bits or 16 16 bits → 32 bits.
DIVXU	B/W	Rd ÷ Rs → Rd Performs unsigned division on data in two ge registers: either 16 bits ÷ 8 bits → 8-bit quotie remainder or 32 bits ÷ 16 bits → 16-bit quotie bit remainder.

		general register or with immediate data, and bits according to the result.
NEG	B/W/L	0 – Rd → Rd Takes the two's complement (arithmetic com data in a general register.
EXTU	W/L	Rd (zero extension) → Rd Extends the lower 8 bits of a 16-bit register t or the lower 16 bits of a 32-bit register to lon by padding with zeros on the left.
EXTS	W/L	Rd (sign extension) → Rd Extends the lower 8 bits of a 16-bit register t or the lower 16 bits of a 32-bit register to lon by extending the sign bit.
TAS	B	@ERd – 0, 1 → (<bit 7> of @ERd)*2 Tests memory contents, and sets the most s (bit 7) to 1.
MAC	—	(EAs) × (EAd) + MAC → MAC Performs signed multiplication on memory c adds the result to the multiply-accumulate re following operations can be performed: 16 bits × 16 bits + 32 bits → 32 bits, saturati 16 bits × 16 bits + 42 bits → 42 bits, non-sat
CLRMAC	—	0 → MAC Clears the multiply-accumulate register to ze
LDMAC STMAC	L	Rs → MAC, MAC → Rd Transfers data between a general register a multiply-accumulate register.

			Performs a logical exclusive OR operation on register and another general register or imme
	NOT	B/W/L	¬ (Rd) → (Rd) Takes the one's complement of general regis contents.
Shift operations	SHAL SHAR	B/W/L	Rd (shift) → Rd Performs an arithmetic shift on general regist 1-bit or 2-bit shift is possible.
	SHLL SHLR	B/W/L	Rd (shift) → Rd Performs a logical shift on general register co 1-bit or 2-bit shift is possible.
	ROTL ROTR	B/W/L	Rd (rotate) → Rd Rotates general register contents. 1-bit or 2-bit rotation is possible.
	ROTXL ROTXR	B/W/L	Rd (rotate) → Rd Rotates general register contents through the 1-bit or 2-bit rotation is possible.

		operand to 0. The bit number is specified by immediate data or the lower three bits of a g register.
BNOT	B	¬ (<bit-No.> of <EAd>) → (<bit-No.> of <EA Inverts a specified bit in a general register o operand. The bit number is specified by 3-bi data or the lower three bits of a general regi
BTST	B	¬ (<bit-No.> of <EAd>) → Z Tests a specified bit in a general register or operand and sets or clears the Z flag accord bit number is specified by 3-bit immediate da 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 g register or memory operand and stores the r carry flag.
BIAND	B	C ∧ ¬ (<bit-No.> of <EAd>) → C ANDs the carry flag with the inverse of a spe a general register or memory operand and s result in the carry flag. The bit number is specified by 3-bit immedia
BOR	B	C ∨ (<bit-No.> of <EAd>) → C ORs the carry flag with a specified bit in a ge register or memory operand and stores the r carry flag.
BIOR	B	C ∨ ¬ (<bit-No.> of <EAd>) → C ORs the carry flag with the inverse of a spec general register or memory operand and sto result in the carry flag. The bit number is specified by 3-bit immedia

		and stores the result in the carry flag.
		The bit number is specified by 3-bit immediat
BLD	B	(<bit-No.> of <EAd>) → C
		Transfers a specified bit in a general register operand to the carry flag.
BILD	B	¬ (<bit-No.> of <EAd>) → C
		Transfers the inverse of a specified bit in a ge register or memory operand to the carry flag.
		The bit number is specified by 3-bit immediat
BST	B	C → (<bit-No.> of <EAd>)
		Transfers the carry flag value to a specified b general register or memory operand.
BIST	B	¬ C → (<bit-No.> of <EAd>)
		Transfers the inverse of the carry flag value t specified bit in a general register or memory o
		The bit number is specified by 3-bit immediat

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

		them and memory. The upper 8 bits are valid.
STC	B/W	CCR → (EAd), EXR → (EAd) Transfers CCR or EXR contents to a general memory. Although CCR and EXR are 8-bit re word-size transfers are performed between th memory. The upper 8 bits are valid.
ANDC	B	CCR ∧ #IMM → CCR, EXR ∧ #IMM → EXR Logically ANDs the CCR or EXR contents wit immediate data.
ORC	B	CCR ∨ #IMM → CCR, EXR ∨ #IMM → EXR Logically ORs the CCR or EXR contents with data.
XORC	B	CCR ⊕ #IMM → CCR, EXR ⊕ #IMM → EXR Logically exclusive-ORs the CCR or EXR con immediate data.
NOP	—	PC + 2 → PC Only increments the program counter.

		Until R4 = 0 else next;
		Transfers a data block according to paramet general registers R4L or R4, ER5, and ER6. R4L or R4: size of block (bytes) ER5: starting source address ER6: starting destination address Execution of the next instruction begins as s transfer is completed.

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

2.6.4 Basic Instruction Formats

The H8S/2626 Group and H8S/2623 Group instructions consist of 2-byte (1-word) un
instruction consists of an operation field (op field), a register field (r field), an effectiv
extension (EA field), and a condition field (cc).

(1) Operation Field: Indicates the function of the instruction, the addressing mode, a
operation to be carried out on the operand. The operation field always includes the firs
the instruction. Some instructions have two operation fields.

(2) Register Field: Specifies a general register. Address registers are specified by 3 b
registers by 3 bits or 4 bits. Some instructions have two register fields. Some have no
field.

(3) Effective Address Extension: Eight, 16, or 32 bits specifying immediate data, an
address, or a displacement.

(4) Condition Field: Specifies the branching condition of Bcc instructions.

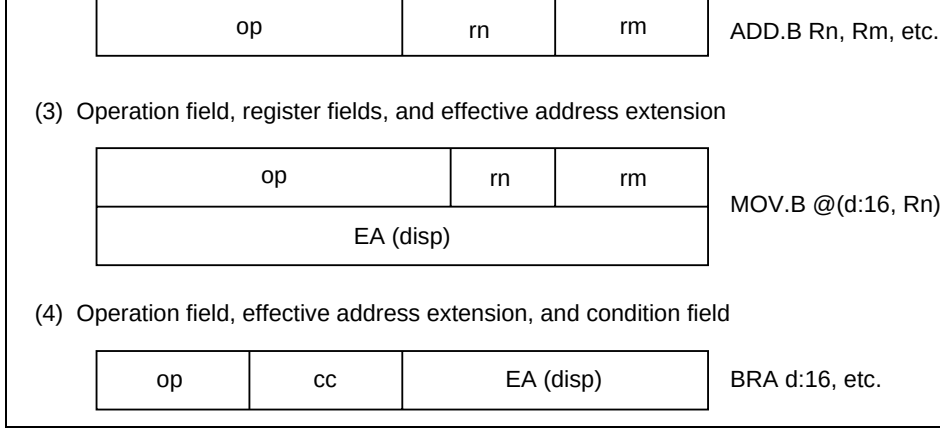


Figure 2.12 Instruction Formats (Examples)

absolute addressing mode to specify an operand, and register direct (BSET, BCLR, BI
BTST instructions) or immediate (3-bit) addressing mode to specify a bit number in th

Table 2.4 Addressing Modes

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

(1) Register Direct—Rn

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

(2) Register Indirect—@ERn

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

The register field of the instruction code specifies an address register (ERn) which c
address of a memory operand. After the operand is accessed, 1, 2, or 4 is added to th
register contents and the sum is stored in the address register. The value added is 1 f
access, 2 for word transfer instruction, or 4 for longword transfer instruction. For w
longword transfer instruction, 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 result becomes the address of a memory operand. Th
  also stored in the address register. The value subtracted is 1 for byte access, 2 for w
  instruction, or 4 for longword transfer instruction. For word or longword transfer in
  the register value should be even.

(5) Absolute Address—@aa:8, @aa:16, @aa:24, or @aa:32

The instruction code contains the absolute address of a memory operand. The absolute
may be 8 bits long (@aa:8), 16 bits long (@aa:16), 24 bits long (@aa:24), or 32 bits lon
(@aa:32).

To access data, the absolute address should be 8 bits (@aa:8), 16 bits (@aa:16), or 32 b
(@aa:32) long. For an 8-bit absolute address, the upper 24 bits are all assumed to be 1 (
For a 16-bit absolute address the upper 16 bits are a sign extension. A 32-bit absolute a
access the entire address space.

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

Table 2.5 indicates the accessible absolute address ranges.

address

Note: * Not available in the H8S/2626 Group or H8S/2623 Group.

(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 d operand.

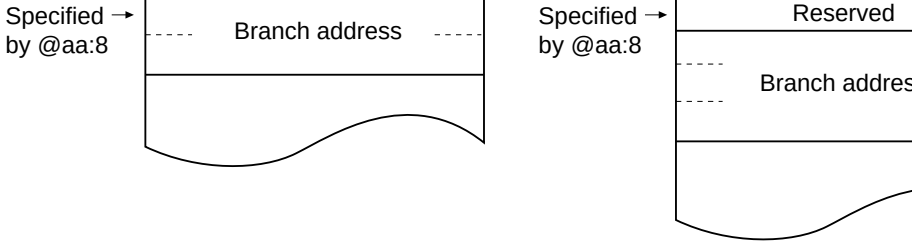
The ADDS, SUBS, INC, and DEC instructions contain immediate data implicitly. Son manipulation instructions contain 3-bit immediate data in the instruction code, specify number. The TRAPA instruction contains 2-bit immediate data in its instruction code, vector address.

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

This mode is used in the Bcc and BSR instructions. An 8-bit or 16-bit displacement co the instruction is sign-extended and added to the 24-bit PC contents to generate a bran Only the lower 24 bits of this branch address are valid; the upper 8 bits are all assume (H'00). The PC value to which the displacement is added is the address of the first byt instruction, so the possible branching range is –126 to +128 bytes (–63 to +64 words) +32768 bytes (–16383 to +16384 words) from the branch instruction. The resulting va be an even number.

(8) Memory Indirect—@@aa:8

This mode can be used by the JMP and JSR instructions. The instruction code contain absolute address specifying a memory operand. This memory operand contains a bran The upper bits of the absolute address are all assumed to be 0, so the address range is (H'0000 to H'00FF in normal mode, H'000000 to H'0000FF in advanced mode). In nor the memory operand is a word operand and the branch address is 16 bits long. In adva the memory operand is a longword operand, the first byte of which is assumed to be a



(a) Normal Mode* **(b) Advanced M**

Note: * Not available in the H8S/2626 Group or H8S/2623 Group.

Figure 2.13 Branch Address Specification in Memory Indirect Mode

If an odd address is specified in word or longword memory access, or as a branch addre least significant bit is regarded as 0, causing data to be accessed or instruction code to b at the address preceding the specified address. (For further information, see section 2.5 Data Formats.)

2.7.2 Effective Address Calculation

Table 2.6 indicates how effective addresses are calculated in each addressing mode. In mode* the upper 8 bits of the effective address are ignored in order to generate a 16-bit

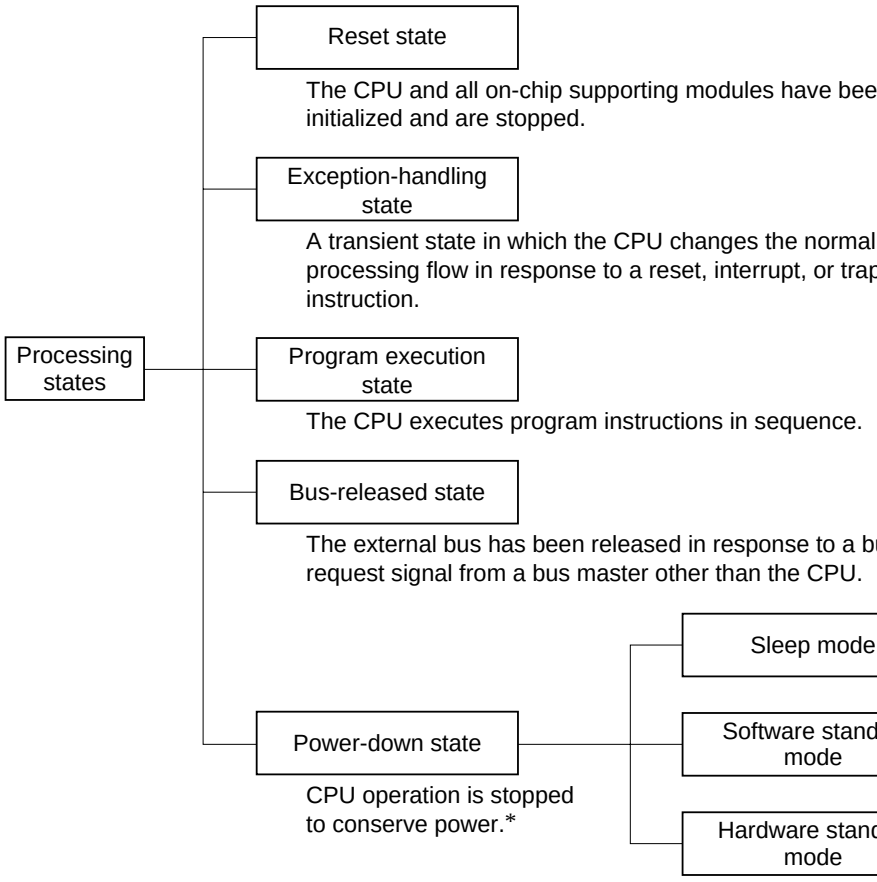
Note: * Cannot be set in the H8S/2626 Group or H8S/2623 Group.

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

RENESAS

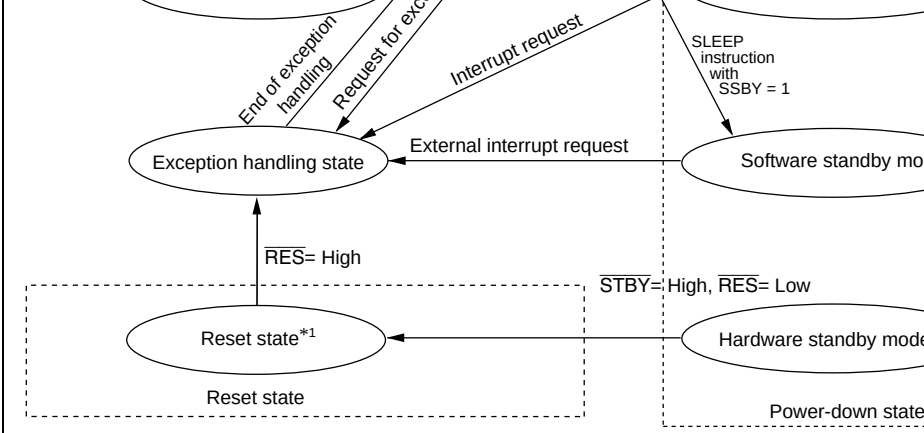
No.	Addressing Mode and Instruction Format	Effective Address Calculation	Ef
5	Absolute address @aa:8 op | abs @aa:16 op | abs @aa:24 op | abs @aa:32 op | abs		31 24 Don't care 31 24 Don't care 31 24 Don't care 31 24 Don't care
6	Immediate #xx:8/#xx:16/#xx:32 op | IMM		Operand is

No.	Addressing Mode and Instruction Format	Effective Address Calculation	Ef
7	Program-counter relative @(d:8, PC)/@(d:16, PC) op | disp	23 0 PC contents 23 0 Sign extension | disp 31 24 Don't care	
8	Memory indirect @@aa:8 • Normal mode* op | abs	31 8 7 0 H'000000 | abs 15 0 Memory contents 31 24 Don't care	
	• Advanced mode op | abs	31 8 7 0 H'000000 | abs 31 0 Memory contents 31 24 Don't care	



Note: * The power-down state also includes a medium-speed mode, module stop m
subactive mode, subsleep mode, and watch mode. Subclock functions (sub
mode, subsleep mode, and watch mode) are not available in the H8S/2623
but are available in the H8S/2626 Group.

Figure 2.14 Processing States



Notes: 1. From any state except hardware standby mode, a transition to the reset state occurs wh
goes low. A transition can also be made to the reset state when the watchdog timer over
2. From any state, a transition to hardware standby mode occurs when $\overline{\text{STBY}}$ goes low.
3. Apart from these states, there are also the watch mode, subactive mode, and the subsle
H8S/2626 Group. See section 21B, Power-Down Modes [H8S/2626 Group].

Figure 2.15 State Transitions

2.8.2 Reset State

When the $\overline{\text{RES}}$ input goes low all current processing stops and the CPU enters the rese
interrupts are masked in the reset state. Reset exception handling starts when the $\overline{\text{RES}}$
changes from low to high.

The reset state can also be entered by a watchdog timer overflow. For details, refer to
Watchdog Timer.

indicates the types of exception handling and their priority. Trap instruction exception h
always accepted, in the program execution state.

Exception handling and the stack structure depend on the interrupt control mode set in

Table 2.7 Exception Handling Types and Priority

Priority	Type of Exception	Detection Timing	Start of Exception H
High	Reset	Synchronized with clock	Exception handling st immediately after a lo transition at the $\overline{\text{RES}}$ when the watchdog ti overflows.
	Trace	End of instruction execution or end of exception-handling sequence*1	When the trace (T) bi 1, the trace starts at t the current instruction exception-handling se
	Interrupt	End of instruction execution or end of exception-handling sequence*2	When an interrupt is exception handling st end of the current ins current exception-han sequence
	Trap instruction	When TRAPA instruction is executed	Exception handling st a trap (TRAPA) instru executed*3
Low			

Notes: 1. Traces are enabled only in interrupt control mode 2. Trace exception-handlin executed at the end of the RTE instruction.
2. Interrupts are not detected at the end of the ANDC, ORC, XORC, and LDC i or immediately after reset exception handling.
3. Trap instruction exception handling is always accepted, in the program exec

Traces are enabled only in interrupt control mode 2. Trace mode is entered when the T is set to 1. When trace mode is established, trace exception handling starts at the end o instruction.

At the end of a trace exception-handling sequence, the T bit of EXR is cleared to 0 an is cleared. Interrupt masks are not affected.

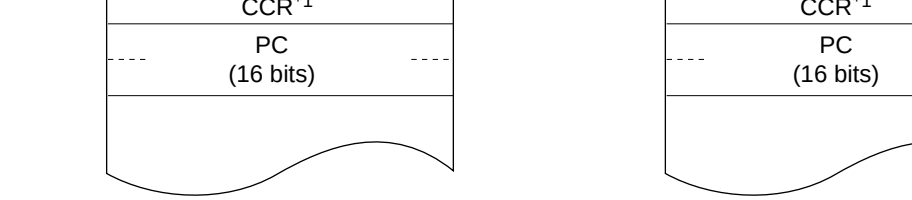
The T bit saved on the stack retains its value of 1, and when the RTE instruction is ex return from the trace exception-handling routine, trace mode is entered again. Trace ex handling is not executed at the end of the RTE instruction.

Trace mode is not entered in interrupt control mode 0, regardless of the state of the T b

(4) Interrupt Exception Handling and Trap Instruction Exception Handling

When interrupt or trap-instruction exception handling begins, the CPU references the (ER7) and pushes the program counter and other control registers onto the stack. Next alters the settings of the interrupt mask bits in the control registers. Then the CPU fetc address (vector) from the exception vector table and program execution starts from tha address.

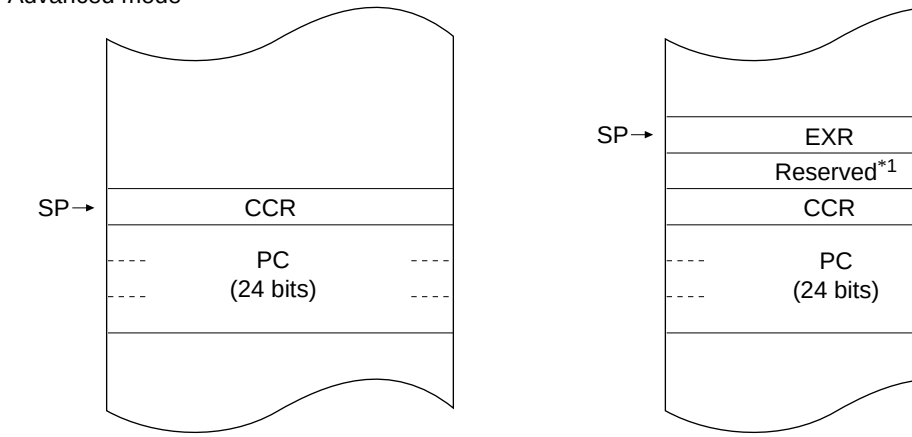
Figure 2.16 shows the stack after exception handling ends.



(a) Interrupt control mode 0

(b) Interrupt control m

Advanced mode



(c) Interrupt control mode 0

(d) Interrupt control m

Notes: 1. Ignored when returning.
2. Not available in the H8S/2626 Group or H8S/2623 Group.

Figure 2.16 Stack Structure after Exception Handling (Examples)

Bus masters other than the CPU are data transfer controller (DTC).

For further details, refer to section 7, Bus Controller.

2.8.6 Power-Down State

The power-down state includes both modes in which the CPU stops operating and mo
the CPU does not stop. There are five modes in which the CPU stops operating: sleep
software standby mode, hardware standby mode, subsleep mode*, and watch mode*.
also three other power-down modes: medium-speed mode, module stop mode, and sul
mode*. In medium-speed mode the CPU and other bus masters operate on a medium-
Module stop mode permits halting of the operation of individual modules, other than t
Subactive mode*, subsleep mode*, and watch mode* are power-down states using sul
For details, refer to section 21B, Power-Down Modes [H8S/2626 Group].

Note: * Supported only in the H8S/2626 Group; not available in the H8S/2623 Gro

2.9.2 On-Chip Memory (ROM, RAM)

On-chip memory is accessed in one state. The data bus is 16 bits wide, permitting both word transfer instruction. Figure 2.17 shows the on-chip memory access cycle. Figure 2 the pin states.

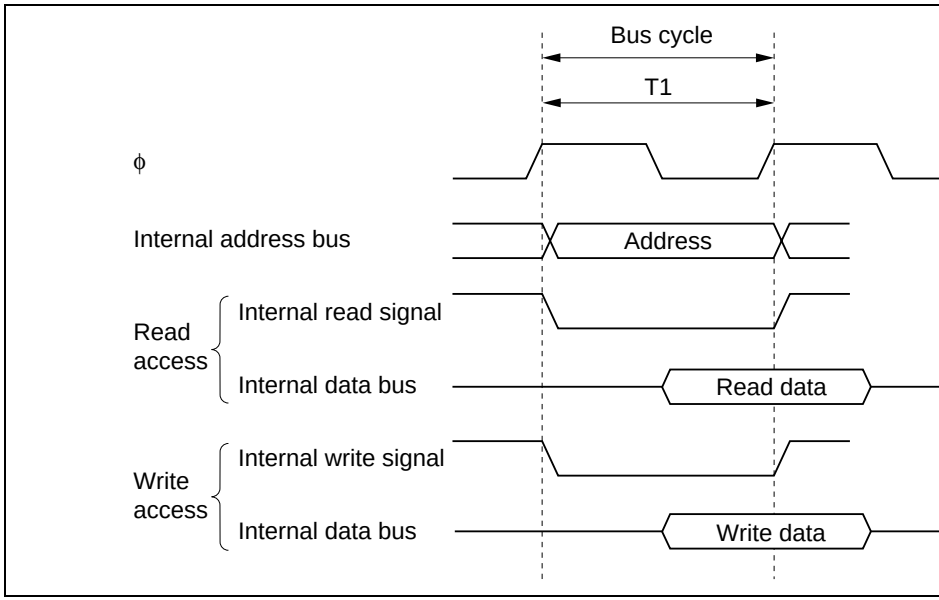


Figure 2.17 On-Chip Memory Access Cycle

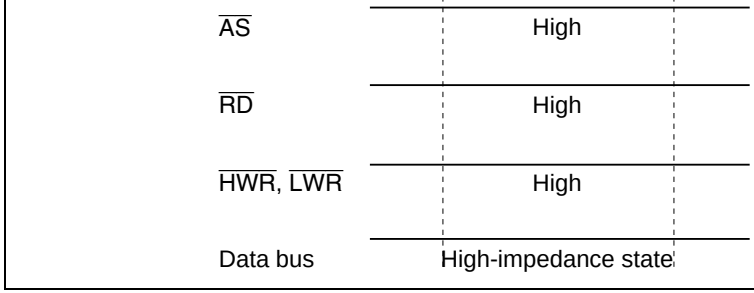


Figure 2.18 Pin States during On-Chip Memory Access

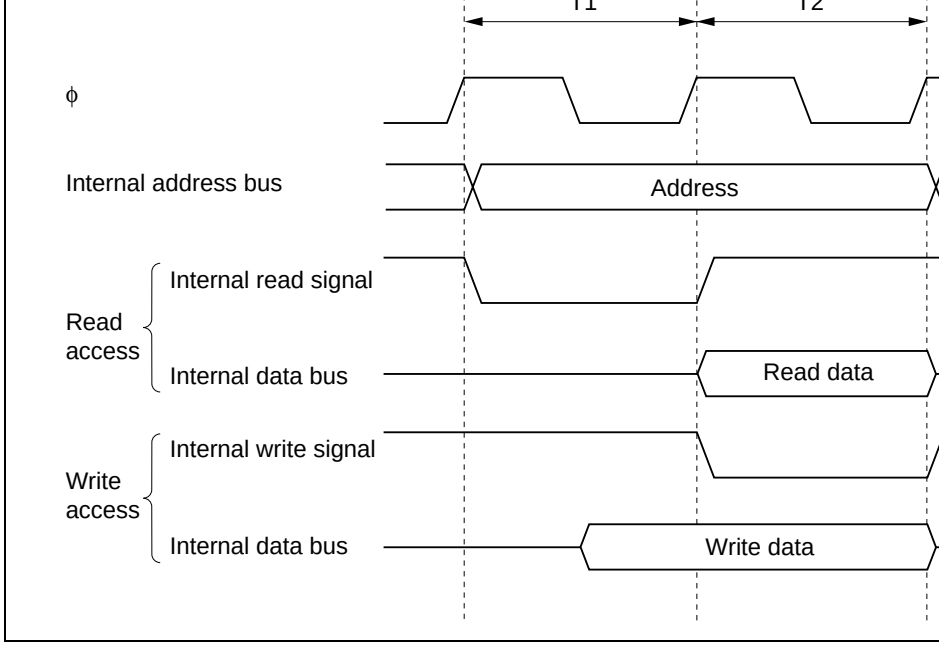


Figure 2.19 On-Chip Supporting Module Access Cycle

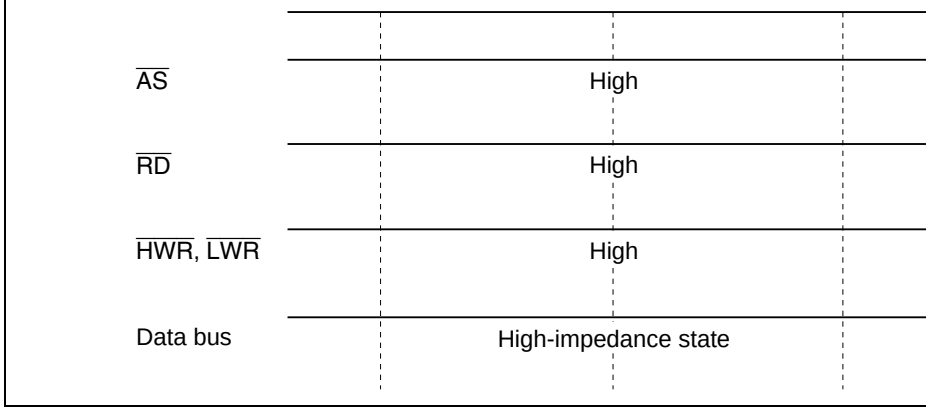


Figure 2.20 Pin States during On-Chip Supporting Module Access Cy

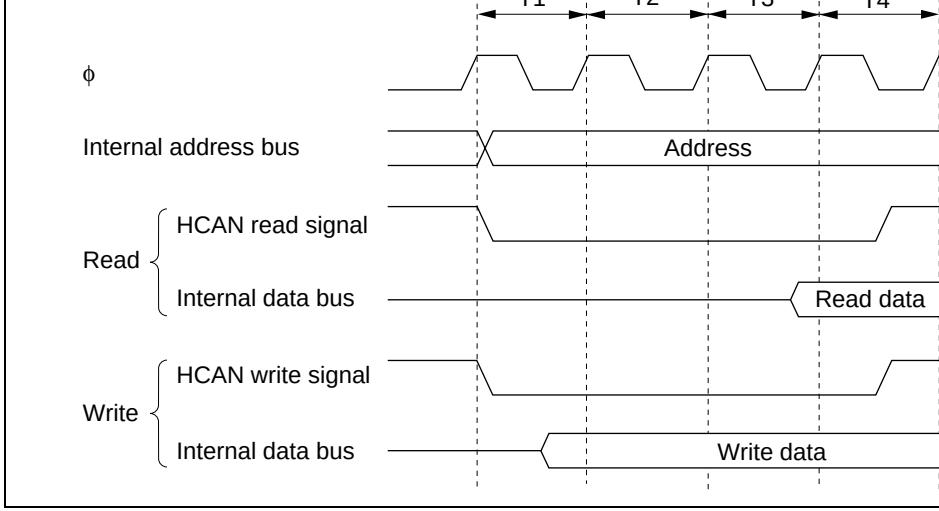


Figure 2.21 On-Chip HCAN Module Access Cycle (No Wait State)

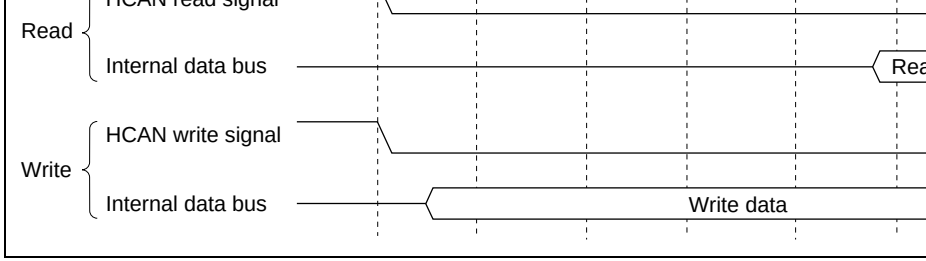


Figure 2.22 On-Chip HCAN Module Access Cycle (Wait States Inserte

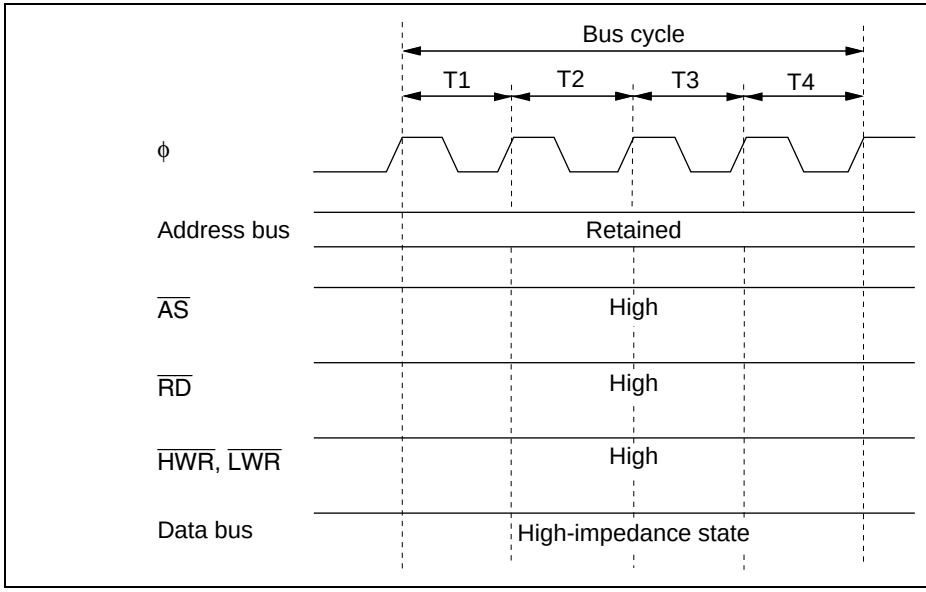


Figure 2.23 Pin States in On-Chip HCAN Module Access

2.10.1 TAS Instruction

Only register ER0, ER1, ER4, or ER5 should be used when using the TAS instruction. instruction is not generated by the Renesas Technology H8S and H8/300 series C/C++ If the TAS instruction is used as a user-defined intrinsic function, ensure that only regis ER1, ER4, or ER5 is used.

modes enable selection of the CPU operating mode, enabling/disabling of on-chip RO initial bus width setting, by setting the mode pins (MD2 to MD0).

Table 3.1 lists the MCU operating modes.

Table 3.1 MCU Operating Mode Selection

MCU Operating Mode	MD2	MD1	MD0	CPU Operating Mode	Description	On-Chip ROM	Extern Initial Width
0*	0	0	0	—	—	—	—
1*			1	—			
2*		1	0				
3*			1				
4	1	0	0	Advanced	On-chip ROM disabled, expanded mode	Disabled	16 bits
5			1				8 bits
6		1	0		On-chip ROM enabled, expanded mode	Enabled	8 bits
7			1		Single-chip mode		—

Note: * Not available in the H8S/2626 Group or H8S/2623 Group.

The CPU's architecture allows for 4 Gbytes of address space, but the H8S/2626 Grou H8S/2623 Group actually access a maximum of 16 Mbytes.

Modes 4 to 6 are externally expanded modes that allow access to external memory an devices.

The external expansion modes allow switching between 8-bit and 16-bit bus modes. A execution starts, an 8-bit or 16-bit address space can be set for each area, depending o controller setting. If 16-bit access is selected for any one area, 16-bit bus mode is set; access is selected for all areas, 8-bit bus mode is set.

The H8S/2626 Group and H8S/2623 Group have a mode control register (MDCR) that the inputs at the mode pins (MD2 to MD0), and a system control register (SYSCR) that the operation of the H8S/2626 Group or H8S/2623 Group chip. Table 3.2 summarizes t registers.

Table 3.2 MCU Registers

Name	Abbreviation	R/W	Initial Value	Add
Mode control register	MDCR	R/W	Undetermined	H'FD
System control register	SYSCR	R/W	H'01	H'FD
Pin function control register	PFCR	R/W	H'0D/H'00	H'FD

Note: * Lower 16 bits of the address.

R/W	:	R/W	—	—	—	—	R	R

Note: * Determined by pins MD2 to MD0.

MDCR is an 8-bit register that indicates the current operating mode of the H8S/2626 G
H8S/2623 Group chip.

Bit 7—Reserved: Only 1 should be written to this bit.

Bits 6 to 3—Reserved: These bits are always read as 0 and cannot be modified.

Bits 2 to 0—Mode Select 2 to 0 (MDS2 to MDS0): These bits indicate the input leve
MD2 to MD0 (the current operating mode). Bits MDS2 to MDS0 correspond to MD2
MDS2 to MDS0 are read-only bits-they cannot be written to. The mode pin (MD2 to I
levels are latched into these bits when MDCR is read. These latches are canceled by a

3.2.2 System Control Register (SYSCR)

Bit	:	7	6	5	4	3	2	1
		MACS	—	INTM1	INTM0	NMIEG	—	—
Initial value	:	0	0	0	0	0	0	0
R/W	:	R/W	—	R/W	R/W	R/W	R/W	—

SYSCR is an 8-bit readable/writable register that selects saturating or non-saturating c
for the MAC instruction, selects the interrupt control mode and the detected edge for I
enables or disables on-chip RAM.

SYSCR is initialized to H'01 by a reset and in hardware standby mode. SYSCR is not
software standby mode.

Bit 6—Reserved: This bit is always read as 0 and cannot be modified.

Bits 5 and 4—Interrupt Control Mode 1 and 0 (INTM1, INTM0): These bits select mode of the interrupt controller. For details of the interrupt control modes, see section 5 Interrupt Control Modes and Interrupt Operation.

Bit 5	Bit 4	Interrupt	
INTM1	INTM0	Control Mode	Description
0	0	0	Control of interrupts by I bit (I
	1	—	Setting prohibited
1	0	2	Control of interrupts by I2 to I0 bits and IPI
	1	—	Setting prohibited

Bit 3—NMI Edge Select (NMIEG): Selects the valid edge of the NMI interrupt input.

Bit 3	
NMIEG	Description
0	An interrupt is requested at the falling edge of NMI input (I
1	An interrupt is requested at the rising edge of NMI input

Bit 2—Reserved: Only 0 should be written to this bit.

Bit 1—Reserved: This bit is always read as 0 and cannot be modified.

3.2.3 Pin Function Control Register (PFCR)

Bit	:	7	6	5	4	3	2	1
		—	—	BUZZE	—	AE3	AE2	AE1
Initial value	:	0	0	0	0	1/0	1/0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PFCR is an 8-bit readable/writable register that performs address output control in ext
expanded mode.

PFCR is initialized to H'0D/H'00 by a reset and in hardware standby mode. It retains i
state in software standby mode.

Bits 7 to 4—Reserved: Only 0 should be written to these bits.

Bit 5—BUZZE Output Enable (BUZZE): This bit is for use only in the H8S/2626. (
should be writtn to this bit.

Bits 3 to 0—Address Output Enable 3 to 0 (AE3 to AE0): These bits select enablin
disabling of address outputs A8 to A23 in ROMless expanded mode and modes with E
a pin is enabled for address output, the address is output regardless of the correspondi
setting. When a pin is disabled for address output, it becomes an output port when the
corresponding DDR bit is set to 1.

	1	0	0	A8–A11 address output enabled; A12–A23 address disabled
			1	A8–A12 address output enabled; A13–A23 address disabled
		1	0	A8–A13 address output enabled; A14–A23 address disabled
			1	A8–A14 address output enabled; A15–A23 address disabled
1	0	0	0	A8–A15 address output enabled; A16–A23 address disabled
			1	A8–A16 address output enabled; A17–A23 address disabled
		1	0	A8–A17 address output enabled; A18–A23 address disabled
			1	A8–A18 address output enabled; A19–A23 address disabled
	1	0	0	A8–A19 address output enabled; A20–A23 address disabled
			1	A8–A20 address output enabled; A21–A23 address disabled (Init
		1	0	A8–A21 address output enabled; A22, A23 address disabled
			1	A8–A23 address output enabled

Note: * In expanded mode with ROM, bits AE3 to AE0 are initialized to B'0000.
In ROMless expanded mode, bits AE3 to AE0 are initialized to B'1101.
Address pins A0 to A7 are made address outputs by setting the correspondi bits to 1.

The initial bus mode after a reset is 16 bits, with 16-bit access to all areas. However, n
bit access is designated by the bus controller for all areas, the bus mode switches to 8

3.3.2 Mode 5

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM

Ports 1, A, B, and C, function as an address bus, ports D and E function as a data bus,
port F carries bus control signals.

The initial bus mode after a reset is 8 bits, with 8-bit access to all areas. However, not
bit access is designated by the bus controller for any area, the bus mode switches to 16
port E becomes a data bus.

3.3.3 Mode 6

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM

Ports 1, A, B, and C, function as input port pins immediately after a reset. Address out
performed by setting the corresponding DDR (data direction register) bits to 1.

Port D function as a data bus, and part of port F carries data bus signals.

The initial bus mode after a reset is 8 bits, with 8-bit access to all areas. However, not
bit access is designated by the bus controller for any area, the bus mode switches to 16
port E becomes a data bus.

3.3.4 Mode 7

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM
but external addresses cannot be accessed.

Port		Mode 4	Mode 5	Mode 6	Mode
Port 1	P10	A	A	P*/A	P
	P11 to P13	P*/A	P*/A	P*/A	P
Port A	PA4 to PA0	A	A	P*/A	P
Port B		A	A	P*/A	P
Port C		A	A	P*/A	P
Port D		D	D	D	P
Port E		P/D*	P*/D	P*/D	P
Port F	PF7	P/C*	P/C*	P/C*	P*/C
	PF6 to PF4	C	C	C	P
	PF3	P/C*	P*/C	P*/C	
	PF2 to PF0	P*/C	P*/C	P*/C	

Legend:

P: I/O port

A: Address bus output

D: Data bus I/O

C: Control signals, clock I/O

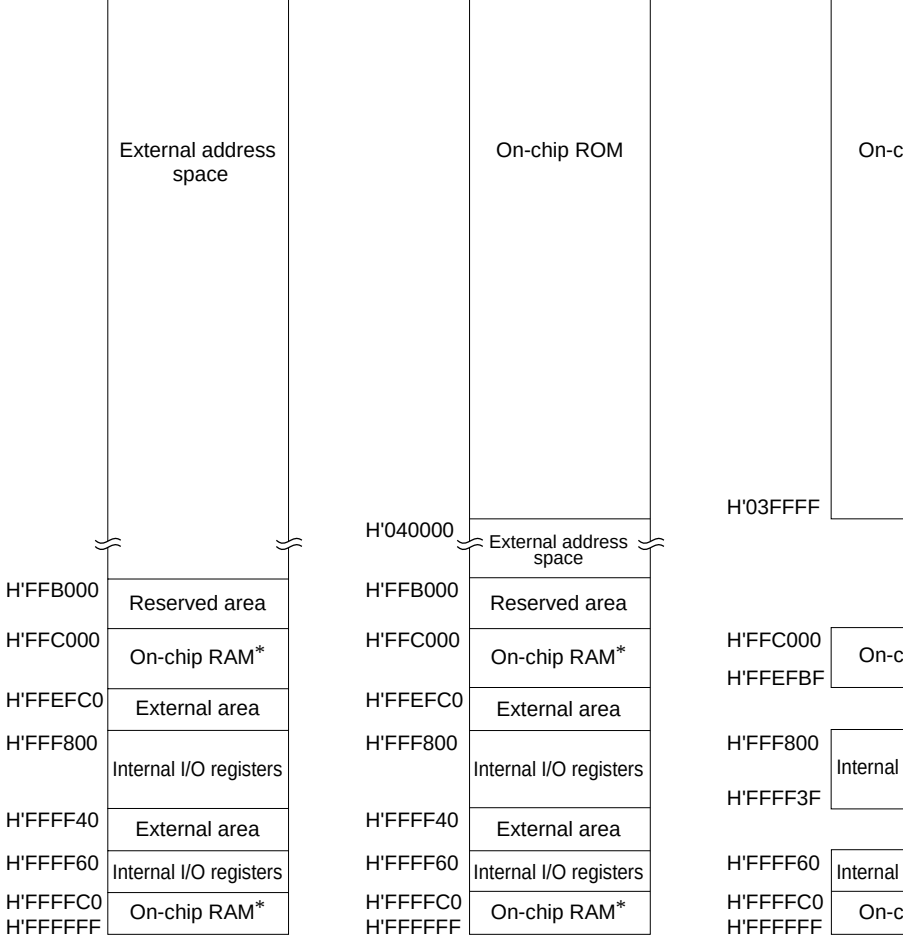
Note: * After reset

3.5 Address Map in Each Operating Mode

An address map of the H8S/2623 and H8S/2626 is shown in figure 3.1, and an address
H8S/2622, and H8S/2625 in figure 3.2, and an address map of the H8S/2621 and H8S/
figure 3.3.

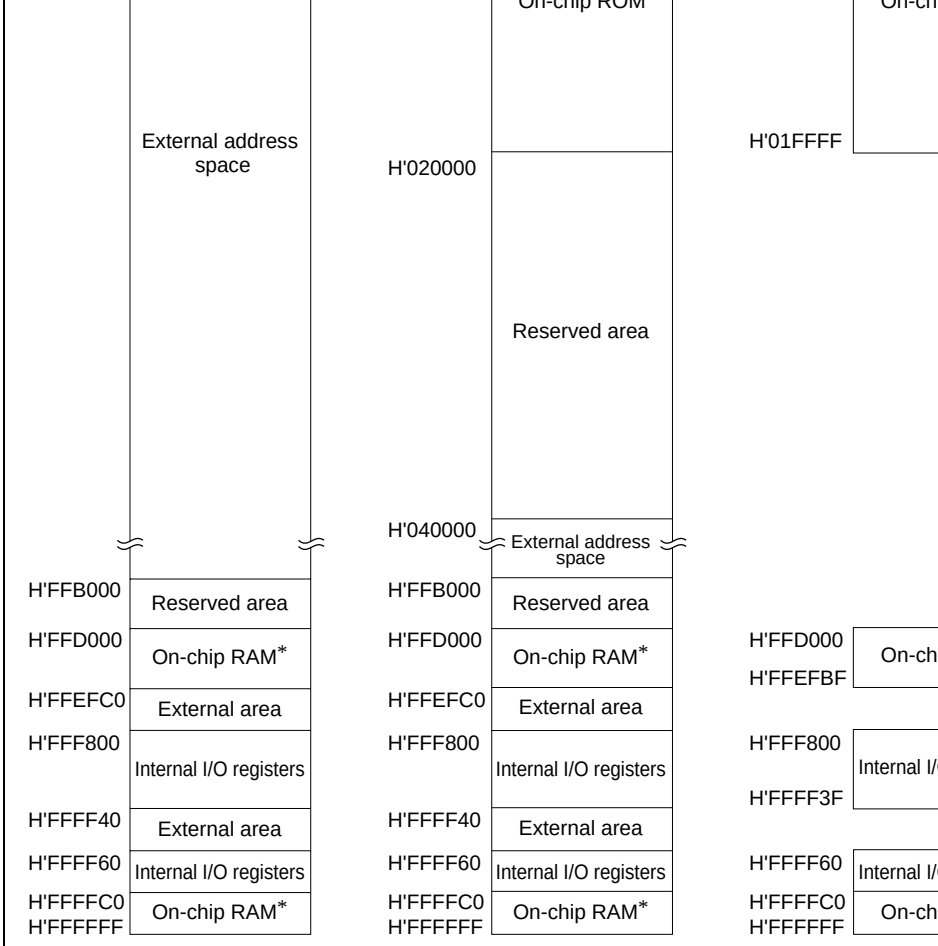
The address space is 16 Mbytes in modes 4 to 7 (advanced modes).

The address space is divided into eight areas for modes 4 to 7. For details, see section 7
Controller.



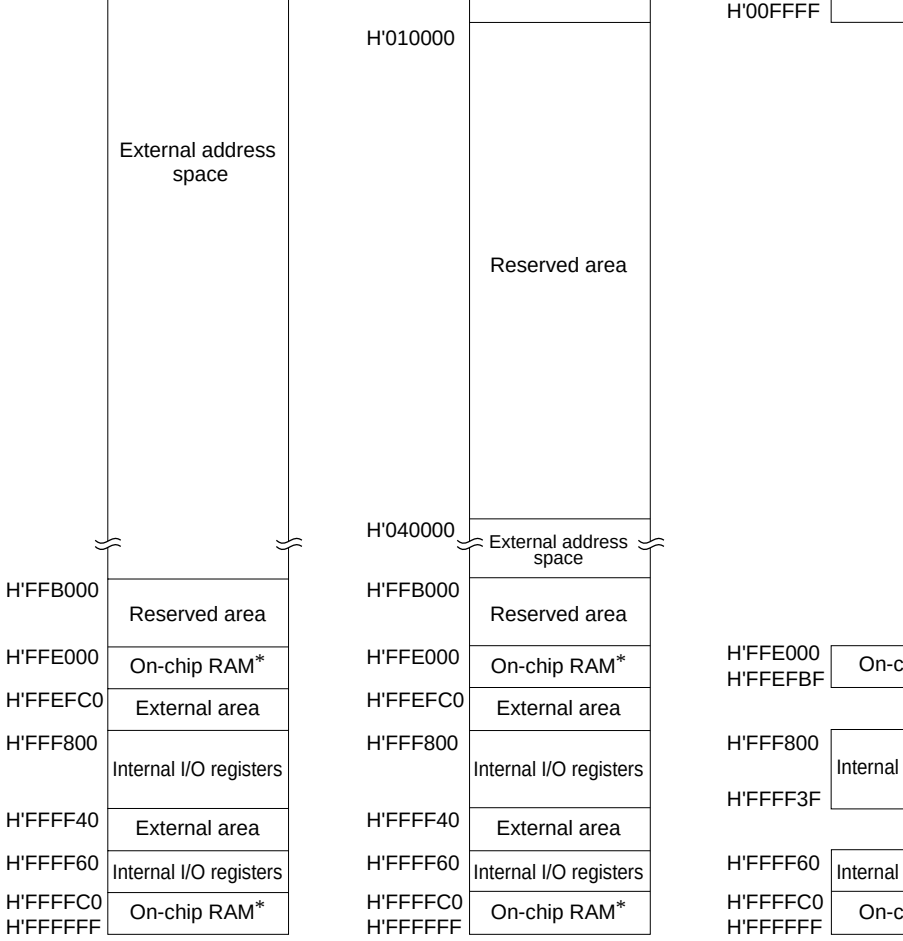
Note: * External addresses can be accessed by clearing the RAME bit in SYSCR to 0.

Figure 3.1 Memory Map in Each Operating Mode in the H8S/2623 and H8



Note: * External addresses can be accessed by clearing the RAME bit in SYSCR to 0.

Figure 3.2 Memory Map in Each Operating Mode in the H8S/2622 and H8S



Note: * External addresses can be accessed by clearing the RAME bit in SYSCR to 0.

Figure 3.3 Memory Map in Each Operating Mode in the H8S/2621 and H8

Exception handling is prioritized as shown in table 4.1. If two or more exceptions occu
simultaneously, they are accepted and processed in order of priority. Trap instruction e
are accepted at all times, in the program execution state.

Exception handling sources, the stack structure, and the operation of the CPU vary dep
the interrupt control mode set by the INTM0 and INTM1 bits of SYSCR.

Table 4.1 Exception Types and Priority

Priority	Exception Type	Start of Exception Handling
High ▲	Reset	Starts immediately after a low-to-high transition at the R when the watchdog timer overflows. The CPU enters th state when the $\overline{\text{RES}}$ pin is low.
	Trace*1	Starts when execution of the current instruction or exce handling ends, if the trace (T) bit is set to 1
	Direct transition	Starts when a direction transition occurs as the result o instruction execution.
	Interrupt	Starts when execution of the current instruction or exce handling ends, if an interrupt request has been issued*
Low ▼	Trap instruction (TRAPA)*3	Started by execution of a trap instruction (TRAPA)

Notes: 1. Traces are enabled only in interrupt control mode 2. Trace exception handl executed after execution of an RTE instruction.
2. Interrupt detection is not performed on completion of ANDC, ORC, XORC, instruction execution, or on completion of reset exception handling.
3. Trap instruction exception handling requests are accepted at all times in pr execution state.

starts from that address.

For a reset exception, steps 2 and 3 above are carried out.

4.1.3 Exception Vector Table

The exception sources are classified as shown in figure 4.1. Different vector addresses
assigned to different exception sources.

Table 4.2 lists the exception sources and their vector addresses.

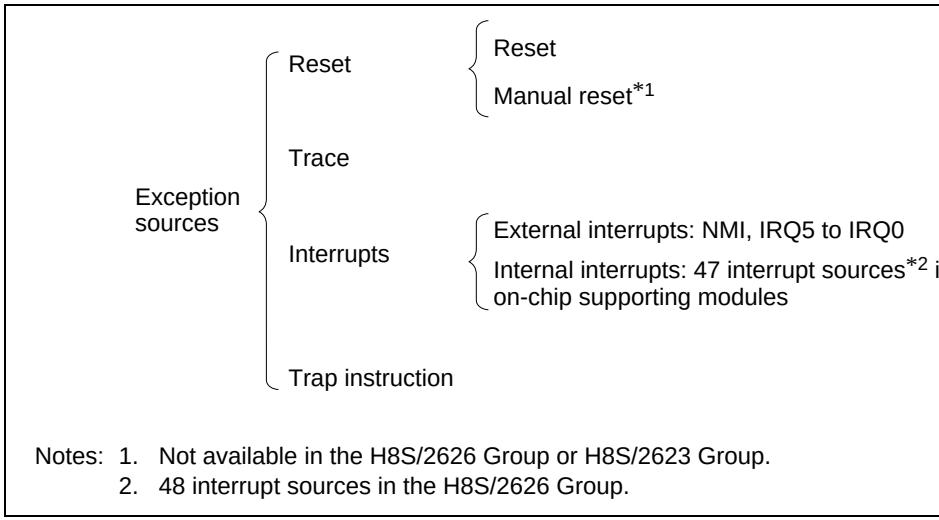


Figure 4.1 Exception Sources

		4	H'0010 to H'0013
Trace		5	H'0014 to H'0017
Direct transitions*4 (H8S/2626 only)		6	H'0018 to H'001B
External interrupt	NMI	7	H'001C to H'001F
Trap instruction (4 sources)		8	H'0020 to H'0023
		9	H'0024 to H'0027
		10	H'0028 to H'002B
		11	H'002C to H'002F
Reserved		12	H'0030 to H'0033
		13	H'0034 to H'0037
		14	H'0038 to H'003B
		15	H'003C to H'003F
External interrupt	IRQ0	16	H'0040 to H'0043
	IRQ1	17	H'0044 to H'0047
	IRQ2	18	H'0048 to H'004B
	IRQ3	19	H'004C to H'004F
	IRQ4	20	H'0050 to H'0053
	IRQ5	21	H'0054 to H'0057
Reserved		22	H'0058 to H'005B
		23	H'005C to H'005F
Internal interrupt*2		24 ⎪ 127	H'0060 to H'0063 ⎪ H'01FC to H'01FF

Notes: 1. Lower 16 bits of the address.

2. For details of internal interrupt vectors, see section 5.3.3, Interrupt Exceptio Vector Table.
3. Not available in the H8S/2626 Group or H8S/2623 Group.
4. See section 21B.11, Direct Transitions, for details.

supporting modules. Immediately after a reset, interrupt control mode 0 is set.

Reset exception handling begins when the $\overline{\text{RES}}$ pin changes from low to high.

The chip can also be reset by overflow of the watchdog timer. For details see section 12
Watchdog Timer.

4.2.2 Reset Sequence

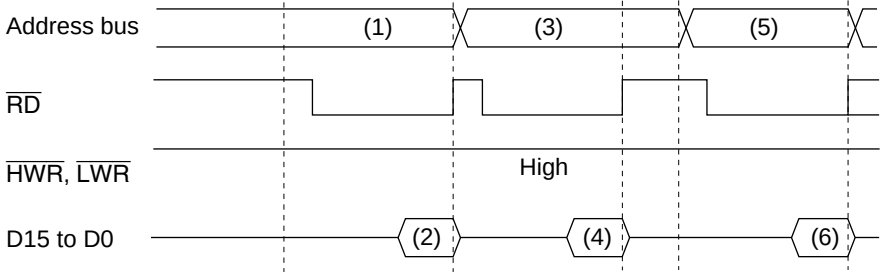
The chip enters the reset state when the $\overline{\text{RES}}$ pin goes low.

To ensure that the chip is reset, hold the $\overline{\text{RES}}$ pin low for at least 20 ms at power-up. To
chip during operation, hold the $\overline{\text{RES}}$ pin low for at least 20 states.

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

1. The internal state of the CPU and the registers of the on-chip supporting modules ar
   initialized, the T bit is cleared to 0 in EXR, and the I bit is set to 1 in EXR and CCR
2. The reset exception handling vector address is read and transferred to the PC, and p
   execution starts from the address indicated by the PC.

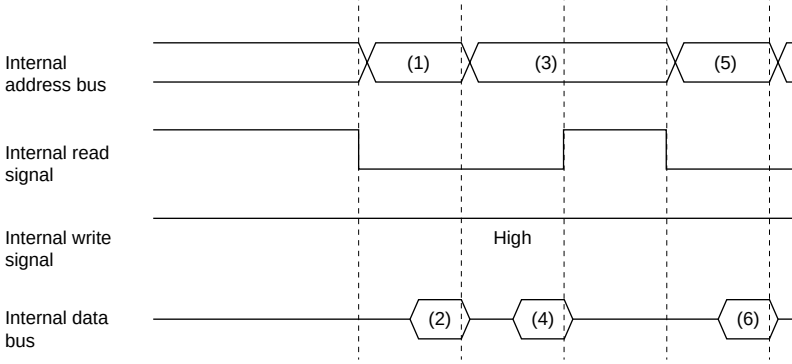
Figures 4.2 and 4.3 show examples of the reset sequence.



(1) (3) Reset exception handling vector address (when reset, (1) = H'000000, (3) = H
(2) (4) Start address (contents of reset exception handling vector address)
(5) Start address ((5) = (2) (4))
(6) First program instruction

Note: * Three program wait states are inserted.

Figure 4.2 Reset Sequence (Modes 4 and 5)



(1) (3) Reset exception handling vector address (when reset, (1) = H'000000, (3) = H'000002)
(2) (4) Start address (contents of reset exception handling vector address)
(5) Start address ((5) = (2) (4))
(6) First program instruction

Figure 4.3 Reset Sequence (Modes 6 and 7)

4.2.3 Interrupts after Reset

If an interrupt is accepted after a reset but before the stack pointer (SP) is initialized, th CCR will not be saved correctly, leading to a program crash. To prevent this, all interru including NMI, are disabled immediately after a reset. Since the first instruction of a pr always executed immediately after the reset state ends, make sure that this instruction i the stack pointer (example: MOV.L #xx: 32, SP).

4.2.4 State of On-Chip Supporting Modules after Reset Release

After reset release, MSTPCRA to MSTPCRC are initialized to H'3F, H'FF, and H'FF, r and all modules except the DTC enter module stop mode. Consequently, on-chip suppo

If the T bit in EXR is set to 1, trace mode is activated. In trace mode, a trace exception completion of each instruction.

Trace mode is canceled by clearing the T bit in EXR to 0. It is not affected by interrup

Table 4.3 shows the state of CCR and EXR after execution of trace exception handling

Interrupts are accepted even within the trace exception handling routine.

The T bit saved on the stack retains its value of 1, and when control is returned from t exception handling routine by the RTE instruction, trace mode resumes.

Trace exception handling is not carried out after execution of the RTE instruction.

Table 4.3 Status of CCR and EXR after Trace Exception Handling

Interrupt Control Mode	CCR		EXR	
	I	UI	I2 to I0	
0	Trace exception handling cannot be used			
2	1	—	—	

Legend:
1: Set to 1
0: Cleared to 0
—: Retains value prior to execution.

interrupt source has a separate vector address.

NMI is the highest-priority interrupt. Interrupts are controlled by the interrupt controlle
interrupt controller has two interrupt control modes and can assign interrupts other than
eight priority/mask levels to enable multiplexed interrupt control.

For details of interrupts, see section 5, Interrupt Controller.

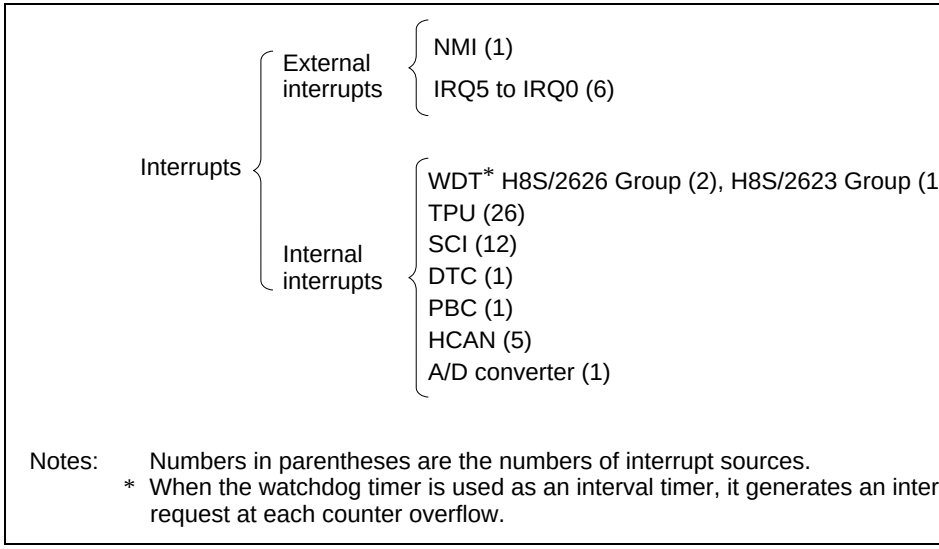


Figure 4.4 Interrupt Sources and Number of Interrupts

Table 4.4 Status of CCR and EXR after Trap Instruction Exception Handling

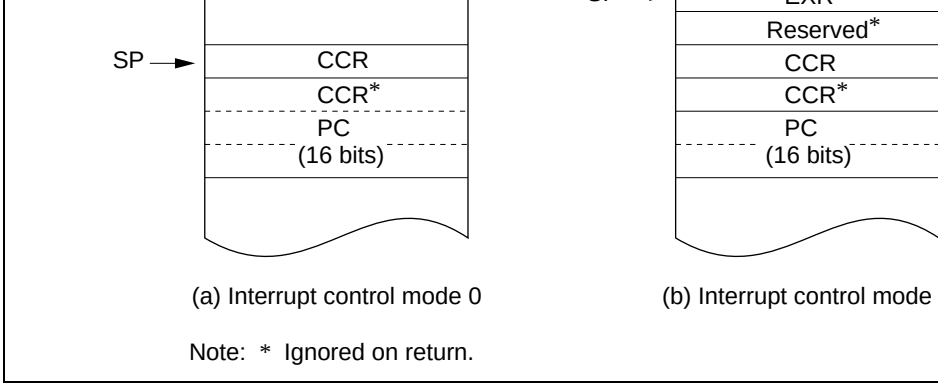
Interrupt Control Mode	CCR		EXR	
	I	UI	I2 to I0	
0	1	—	—	
2	1	—	—	

Legend:

1: Set to 1

0: Cleared to 0

—: Retains value prior to execution.

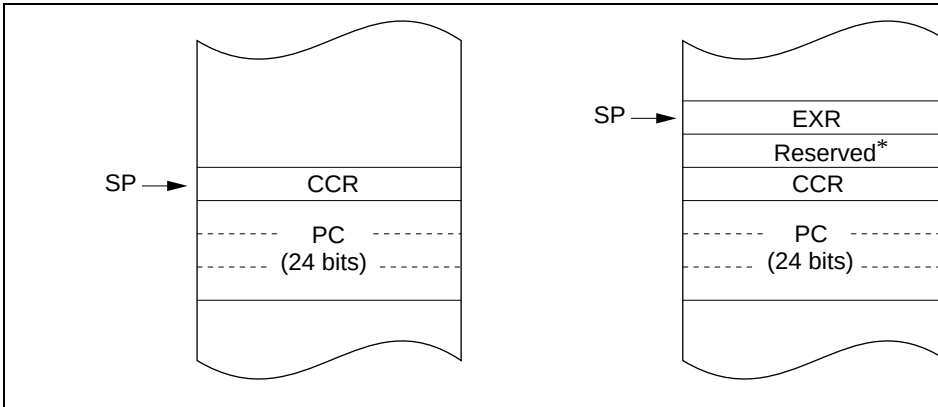


(a) Interrupt control mode 0

(b) Interrupt control mode

Note: * Ignored on return.

Figure 4.5 (1) Stack Status after Exception Handling
(Normal Modes: Not Available in the H8S/2626 Group or H8S/2623 Grou



(a) Interrupt control mode 0

(b) Interrupt control mode

Note: * Ignored on return.

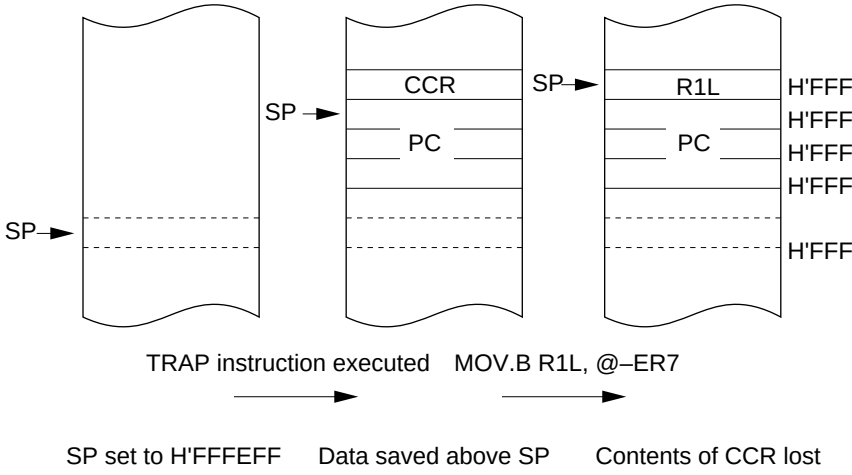
Figure 4.5 (2) Stack Status after Exception Handling
(Advanced Modes)

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

Use the following instructions to restore registers:

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

Setting SP to an odd value may lead to a malfunction. Figure 4.6 shows an example of
happens when the SP value is odd.



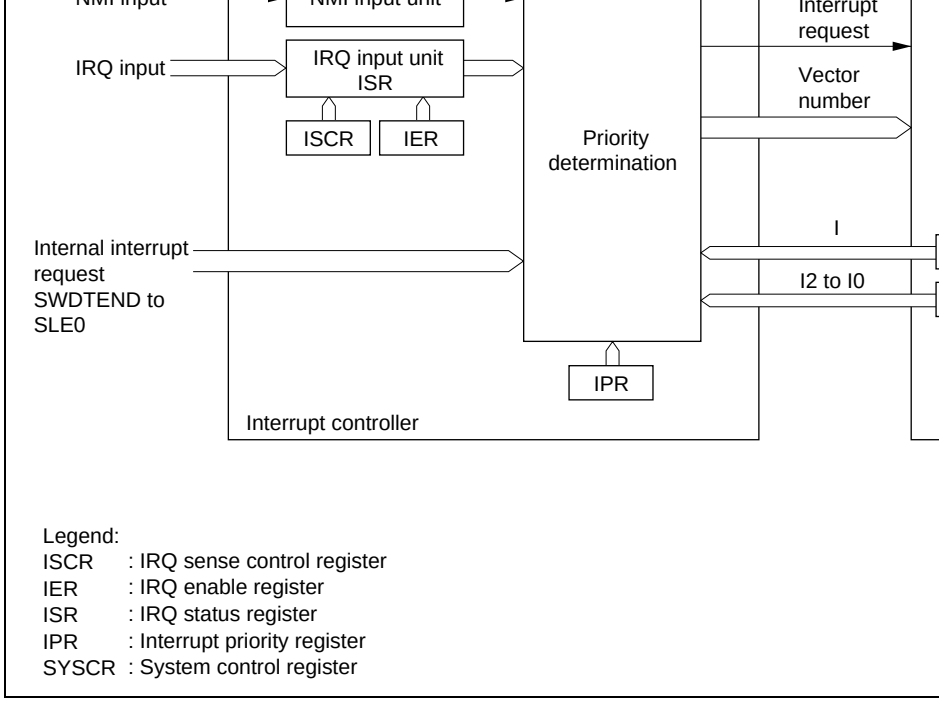
Legend: CCR: Condition code register
PC: Program counter
R1L: General register R1L
SP: Stack pointer

Note: This diagram illustrates an example in which the interrupt control mode is 0, in advanced mode.

Figure 4.6 Operation when SP Value is Odd

The interrupt controller has the following features:

- Two interrupt control modes
  - Any of two interrupt control modes can be set by means of the INTM1 and INT
    the system control register (SYSCR).
- Priorities settable with IPR
  - An interrupt priority register (IPR) is provided for setting interrupt priorities. E
    levels can be set for each module for all interrupts except NMI.
  - NMI is assigned the highest priority level of 8, and can be accepted at all times
- Independent vector addresses
  - All interrupt sources are assigned independent vector addresses, making it unn
    the source to be identified in the interrupt handling routine.
- Seven external interrupts
  - NMI is the highest-priority interrupt, and is accepted at all times. Rising edge o
    edge can be selected for NMI.
  - Falling edge, rising edge, or both edge detection, or level sensing, can be selec
    to IRQ0.
- DTC control
  - DTC activation is performed by means of interrupts.



Legend:

ISCR	: IRQ sense control register
IER	: IRQ enable register
ISR	: IRQ status register
IPR	: Interrupt priority register
SYSCR	: System control register

Figure 5.1 Block Diagram of Interrupt Controller

External interrupt requests 5 to 0	$\overline{IRQ5}$ to $\overline{IRQ0}$	Input	Maskable external interrupts; risin both edges, or level sensing, can

5.1.4 Register Configuration

Table 5.2 summarizes the registers of the interrupt controller.

Table 5.2 Interrupt Controller Registers

Name	Abbreviation	R/W	Initial Value	A
System control register	SYSCR	R/W	H'01	H
IRQ sense control register H	ISCRH	R/W	H'00	H
IRQ sense control register L	ISCRL	R/W	H'00	H
IRQ enable register	IER	R/W	H'00	H
IRQ status register	ISR	R/(W)*2	H'00	H
Interrupt priority register A	IPRA	R/W	H'77	H
Interrupt priority register B	IPRB	R/W	H'77	H
Interrupt priority register C	IPRC	R/W	H'77	H
Interrupt priority register D	IPRD	R/W	H'77	H
Interrupt priority register E	IPRE	R/W	H'77	H
Interrupt priority register F	IPRF	R/W	H'77	H
Interrupt priority register G	IPRG	R/W	H'77	H
Interrupt priority register H	IPRH	R/W	H'77	H
Interrupt priority register I	IPRI	R/W	H'77	H
Interrupt priority register J	IPRJ	R/W	H'77	H
Interrupt priority register K	IPRK	R/W	H'77	H
Interrupt priority register M	IPRM	R/W	H'77	H

Notes: 1. Lower 16 bits of the address.
2. Only 0 can be written, for flag clearing.

R/W	.	R/W	—	R/W	R/W	R/W	R/W	—

SYSCR is an 8-bit readable/writable register that selects the interrupt control mode, and detected edge for NMI.

Only bits 5 to 3 are described here; for details of the other bits, see section 3.2.2, Syster Register (SYSCR).

SYSCR is initialized to H'01 by a reset and in hardware standby mode. SYSCR is not i software standby mode.

Bits 5 and 4—Interrupt Control Mode 1 and 0 (INTM1, INTM0): These bits select interrupt control modes for the interrupt controller.

Bit 5	Bit 4	Interrupt		
INTM1	INTM0	Control Mode	Description	
0	0	0	Interrupts are controlled by I bit	(
	1	—	Setting prohibited	
1	0	2	Interrupts are controlled by bits I2 to I0, and IPR	
	1	—	Setting prohibited	

Bit 3—NMI Edge Select (NMIEG): Selects the input edge for the NMI pin.

Bit 3		
NMIEG	Description	
0	Interrupt request generated at falling edge of NMI input	(
1	Interrupt request generated at rising edge of NMI input	

interrupts other than NMI.

The correspondence between IPR settings and interrupt sources is shown in table 5.3.

The IPR registers set a priority (level 7 to 0) for each interrupt source other than NMI.

The IPR registers are initialized to H'77 by a reset and in hardware standby mode.

They are not initialized in software standby mode.

Bits 7 and 3—Reserved: These bits are always read as 0 and cannot be modified.

IPRD	WDT0	—*1
IPRE	PC break	A/D converter, WDT1*
IPRF	TPU channel 0	TPU channel 1
IPRG	TPU channel 2	TPU channel 3
IPRH	TPU channel 4	TPU channel 5
IPRI	—*1	—*1
IPRJ	—*1	SCI channel 0
IPRK	SCI channel 1	SCI channel 2
IPRM	HCAN	—*1

Notes: 1. Reserved bits. These bits are always read as 1 and cannot be modified.
2. Valid only in the H8S/2626 Group.

As shown in table 5.3, multiple interrupts are assigned to one IPR. Setting a value in th
from H'0 to H'7 in the 3-bit groups of bits 6 to 4 and 2 to 0 sets the priority of the corre
interrupt. The lowest priority level, level 0, is assigned by setting H'0, and the highest p
level, level 7, by setting H'7.

When interrupt requests are generated, the highest-priority interrupt according to the pr
levels set in the IPR registers is selected. This interrupt level is then compared with the
mask level set by the interrupt mask bits (I2 to I0) in the extend register (EXR) in the C
the priority level of the interrupt is higher than the set mask level, an interrupt request i
the CPU.

IRQ5 to IRQ0.

IER is initialized to H'00 by a reset and in hardware standby mode.

They are not initialized in software standby mode.

Bits 7 and 6—Reserved: Only 0 should be written to these bits.

Bits 5 to 0—IRQ5 to IRQ0 Enable (IRQ7E to IRQ0E): These bits select whether I
IRQ0 are enabled or disabled.

Bit n

IRQnE	Description
0	IRQn interrupts disabled
1	IRQn interrupts enabled

ISCRL

Bit	:	7	6	5	4	3	2	1
		IRQ3SCB	IRQ3SCA	IRQ2SCB	IRQ2SCA	IRQ1SCB	IRQ1SCA	IRQ0SCB
Initial value	:	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The ISCR registers are 16-bit readable/writable registers that select rising edge, falling both edge detection, or level sensing, for the input at pins $\overline{\text{IRQ5}}$ to $\overline{\text{IRQ0}}$.

The ISCR registers are initialized to H'0000 by a reset and in hardware standby mode.

They are not initialized in software standby mode.

Bits 15 to 12—Reserved: Only 0 should be written to these bits.

Bits 11 to 0: IRQ7 Sense Control A and B (IRQ5SCA, IRQ5SCB) to IRQ0 Sense Cont B (IRQ0SCA, IRQ0SCB)

Bits 11 to 0

IRQ5SCB to IRQ0SCB	IRQ5SCA to IRQ0SCA	Description
0	0	Interrupt request generated at $\overline{\text{IRQ5}}$ to $\overline{\text{IRQ0}}$ input low l (I
	1	Interrupt request generated at falling edge of $\overline{\text{IRQ5}}$ to $\overline{\text{I}}$
1	0	Interrupt request generated at rising edge of $\overline{\text{IRQ5}}$ to $\overline{\text{IF}}$
	1	Interrupt request generated at both falling and rising ed $\overline{\text{IRQ5}}$ to $\overline{\text{IRQ0}}$ input

ISR is an 8-bit readable/writable register that indicates the status of IRQ5 to IRQ0 inte
requests.

ISR is initialized to H'00 by a reset and in hardware standby mode.

They are not initialized in software standby mode.

Bits 7 and 6—Reserved: Only 0 should be written to these bits.

Bits 5 to 0—IRQ5 to IRQ0 flags (IRQ5F to IRQ0F): These bits indicate the status o
IRQ0 interrupt requests.

	edge detection is set (IRQnSCB = 1 or IRQnSCA = 1)
	• When the DTC is activated by an IRQn interrupt, and the DISEL bit in MRI DTC is cleared to 0
1	[Setting conditions]
	• When $\overline{\text{IRQn}}$ input goes low when low-level detection is set (IRQnSCB = IR 0)
	• When a falling edge occurs in $\overline{\text{IRQn}}$ input when falling edge detection is se (IRQnSCB = 0, IRQnSCA = 1)
	• When a rising edge occurs in $\overline{\text{IRQn}}$ input when rising edge detection is set (IRQnSCB = 1, IRQnSCA = 0)
	• When a falling or rising edge occurs in $\overline{\text{IRQn}}$ input when both-edge detecti (IRQnSCB = IRQnSCA = 1)

5.3 Interrupt Sources

Interrupt sources comprise external interrupts (NMI and IRQ5 to IRQ0) and internal in sources: H8S/2626 Group, 47 sources: H8S/2623 Group).

5.3.1 External Interrupts

There are seven external interrupts: NMI and IRQ5 to IRQ0. These interrupts can be us restore the H8S/2626 Group or H8S/2623 Group chip from software standby mode.

NMI Interrupt: NMI is the highest-priority interrupt, and is always accepted by the C regardless of the interrupt control mode or the status of the CPU interrupt mask bits. Th bit in SYSCR can be used to select whether an interrupt is requested at a rising edge or edge on the NMI pin.

The vector number for NMI interrupt exception handling is 7.

by software.

A block diagram of interrupts IRQ5 to IRQ0 is shown in figure 5.2.

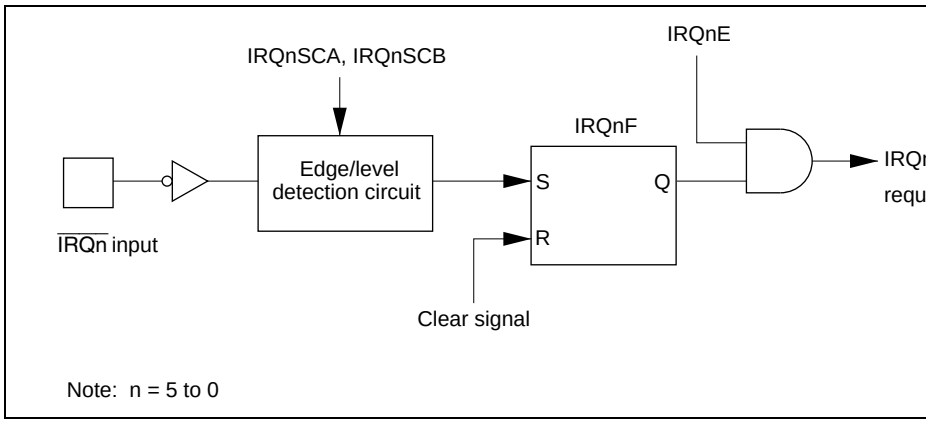


Figure 5.2 Block Diagram of Interrupts IRQ5 to IRQ0

Figure 5.3 shows the timing of setting IRQnF.

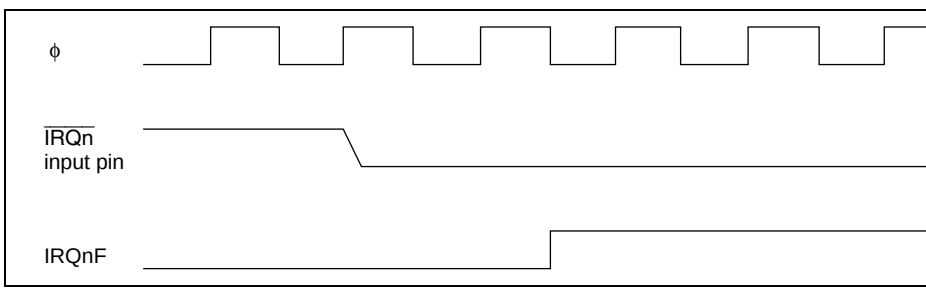


Figure 5.3 Timing of Setting IRQnF

The vector numbers for IRQ5 to IRQ0 interrupt exception handling are 21 to 16.

- For each on-chip supporting module there are flags that indicate the interrupt reque and enable bits that select enabling or disabling of these interrupts. If both of these a for a particular interrupt source, an interrupt request is issued to the interrupt contro
- The interrupt priority level can be set by means of IPR.
- The DTC can be activated by a TPU, 8-bit timer, SCI, or other interrupt request. Wh DTC is activated by an interrupt, the interrupt control mode and interrupt mask bits affected.

5.3.3 Interrupt Exception Handling Vector Table

Table 5.4 shows interrupt exception handling sources, vector addresses, and interrupt p For default priorities, the lower the vector number, the higher the priority.

Priorities among modules can be set by means of the IPR. The situation when two or m modules are set to the same priority, and priorities within a module, are fixed as shown table 5.4.

IRQ1		17	H'0044	IPRA2 to 0
IRQ2		18	H'0048	IPRB6 to 4
IRQ3		19	H'004C	
IRQ4		20	H'0050	IPRB2 to 0
IRQ5		21	H'0054	
Reserved	—	22	H'0058	IPRC6 to 4
		23	H'005C	
SWDTEND (software activation interrupt end)	DTC	24	H'0060	IPRC2 to 0
WOVI0 (interval timer)	Watchdog timer 0	25	H'0064	IPRD6 to 4
Reserved	—	26	H'0068	IPRD2 to 0
PC break	PC break	27	H'006C	IPRE6 to 4
ADI (A/D conversion end)	A/D	28	H'0070	IPRE2 to 0
WOVI1 (interval timer) (H8S/2626 Group only)	Watchdog timer 1	29	H'0074	
Reserved	—	30	H'0078	
		31	H'007C	
TGI0A (TGR0A input capture/ compare match)	TPU channel 0	32	H'0080	IPRF6 to 4
TGI0B (TGR0B input capture/ compare match)		33	H'0084	
TGI0C (TGR0C input capture/ compare match)		34	H'0088	
TGI0D (TGR0D input capture/ compare match)		35	H'008C	
TCI0V (overflow 0)		36	H'0090	
Reserved	—	37	H'0094	
		38	H'0098	
		39	H'009C	

TCI1V (overflow 1)		42	H'00A8	
TCI1U (underflow 1)		43	H'00AC	
TGI2A (TGR2A input capture/ compare match)	TPU channel 2	44	H'00B0	IPRG6 to 4
TGI2B (TGR2B input capture/ compare match)		45	H'00B4	
TCI2V (overflow 2)		46	H'00B8	
TCI2U (underflow 2)		47	H'00BC	
TGI3A (TGR3A input capture/ compare match)	TPU channel 3	48	H'00C0	IPRG2 to 0
TGI3B (TGR3B input capture/ compare match)		49	H'00C4	
TGI3C (TGR3C input capture/ compare match)		50	H'00C8	
TGI3D (TGR3D input capture/ compare match)		51	H'00CC	
TCI3V (overflow 3)		52	H'00D0	
Reserved	—	53 54 55	H'00D4 H'00D8 H'00DC	
TGI4A (TGR4A input capture/ compare match)	TPU channel 4	56	H'00E0	IPRH6 to 4
TGI4B (TGR4B input capture/ compare match)		57	H'00E4	
TCI4V (overflow 4)		58	H'00E8	
TCI4U (underflow 4)		59	H'00EC	

TCI5V (overflow 5)		62	H'00F8	
TCI5U (underflow 5)		63	H'00FC	
Reserved	—	64	H'0100	IPRI6 to 4
		65	H'0104	
		66	H'0108	
		67	H'010C	
		68	H'0110	IPRI2 to 0
		69	H'0114	
		70	H'0118	
		71	H'011C	
		72	H'0120	IPRJ6 to 4
		73	H'0124	
		74	H'0128	
		75	H'012C	
		76	H'0130	
		77	H'0134	
		78	H'0138	
		79	H'013C	
ERI0 (receive error 0)	SCI channel 0	80	H'0140	IPRJ2 to 0
RXI0 (reception completed 0)		81	H'0144	
TXI0 (transmit data empty 0)		82	H'0148	
TEI0 (transmission end 0)		83	H'014C	
ERI1 (receive error 1)	SCI channel 1	84	H'0150	IPRK6 to 4
RXI1 (reception completed 1)		85	H'0154	
TXI1 (transmit data empty 1)		86	H'0158	
TEI1 (transmission end 1)		87	H'015C	
ERI2 (receive error 2)	SCI channel 2	88	H'0160	IPRK2 to 0
RXI2 (reception completed 2)		89	H'0164	
TXI2 (transmit data empty 2)		90	H'0168	
TEI2 (transmission end 2)		91	H'016C	

SLE0	108	H'01B0	IPRM2 to 0

Note: * Lower 16 bits of the start address.

5.4 Interrupt Operation

5.4.1 Interrupt Control Modes and Interrupt Operation

Interrupt operations in the H8S/2626 Group and H8S/2623 Group differ depending on t
interrupt control mode.

NMI interrupts are accepted at all times except in the reset state and the hardware stand
the case of IRQ interrupts and on-chip supporting module interrupts, an enable bit is pr
each interrupt. Clearing an enable bit to 0 disables the corresponding interrupt request.
sources for which the enable bits are set to 1 are controlled by the interrupt controller.

Table 5.5 shows the interrupt control modes.

The interrupt controller performs interrupt control according to the interrupt control mo
the INTM1 and INTM0 bits in SYSCR, the priorities set in IPR, and the masking state
by the I and UI bits in the CPU's CCR, and bits I2 to I0 in EXR.

				is performed by b 8 priority levels ca with IPR.
—	1	—	—	Setting prohibited

Figure 5.4 shows a block diagram of the priority decision circuit.

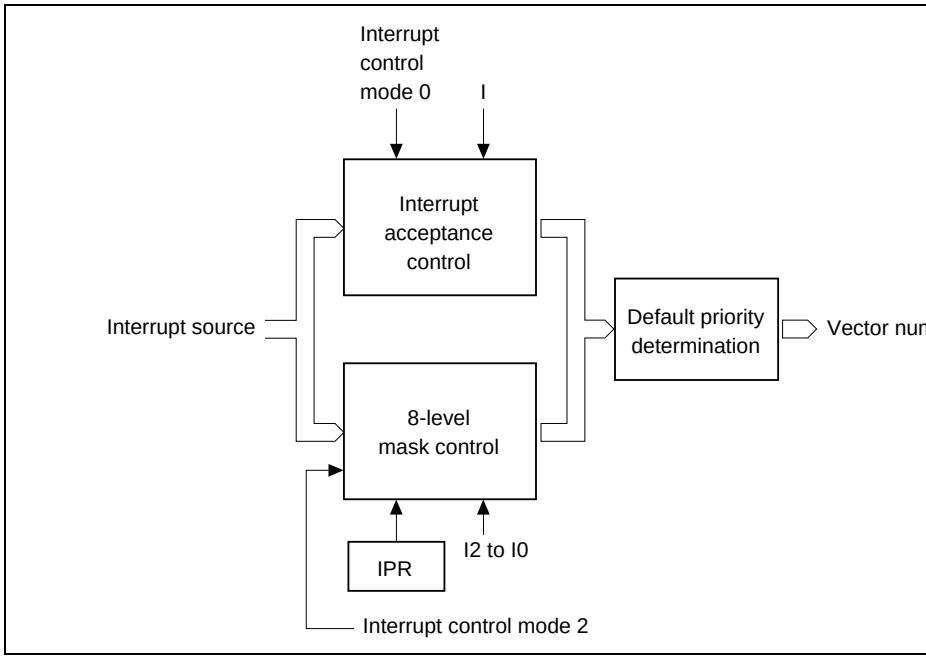


Figure 5.4 Block Diagram of Interrupt Control Operation

Interrupt Control Mode	I	Selected Interrupts
0	0	All interrupts
	1	NMI interrupts
2	*	All interrupts

*

(2) 8-Level Control

In interrupt control mode 2, 8-level mask level determination is performed for the selec
interrupts in interrupt acceptance control according to the interrupt priority level (IPR).

The interrupt source selected is the interrupt with the highest priority level, and whose
level set in IPR is higher than the mask level.

Table 5.7 Interrupts Selected in Each Interrupt Control Mode (2)

Interrupt Control Mode	Selected Interrupts
0	All interrupts
2	Highest-priority-level (IPR) interrupt whose priority level is than the mask level (IPR > I2 to I0).

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

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

Table 5.8 Operations and Control Signal Functions in Each Interrupt Control

Interrupt Control Mode	Setting		Interrupt Acceptance Control		8-Level Control			Default Prior Determinati
	INTM1	INTM0		I		I2 to I0	IPR	
0	0	0	○	IM	X	—	—*2	○
2	1	0	X	—*1	○	IM	PR	○

Legend:

○ : Interrupt operation control performed

X : No operation. (All interrupts enabled)

IM : Used as interrupt mask bit

PR: Sets priority.

— : Not used.

Notes: 1. Set to 1 when interrupt is accepted.

2. Keep the initial setting.

[1] If an interrupt source occurs when the corresponding interrupt enable bit is set to 1, interrupt request is sent to the interrupt controller.

[2] The I bit is then referenced. If the I bit is cleared to 0, the interrupt request is accept bit is set to 1, only an NMI interrupt is accepted, and other interrupt requests are hel

[3] Interrupt requests are sent to the interrupt controller, the highest-ranked interrupt ac the priority system is accepted, and other interrupt requests are held pending.

[4] When an interrupt request is accepted, interrupt exception handling starts after exec current instruction has been completed.

[5] The PC and CCR are saved to the stack area by interrupt exception handling. The P the stack shows the address of the first instruction to be executed after returning fro interrupt handling routine.

[6] Next, the I bit in CCR is set to 1. This masks all interrupts except NMI.

[7] A vector address is generated for the accepted interrupt, and execution of the interru handling routine starts at the address indicated by the contents of that vector addres

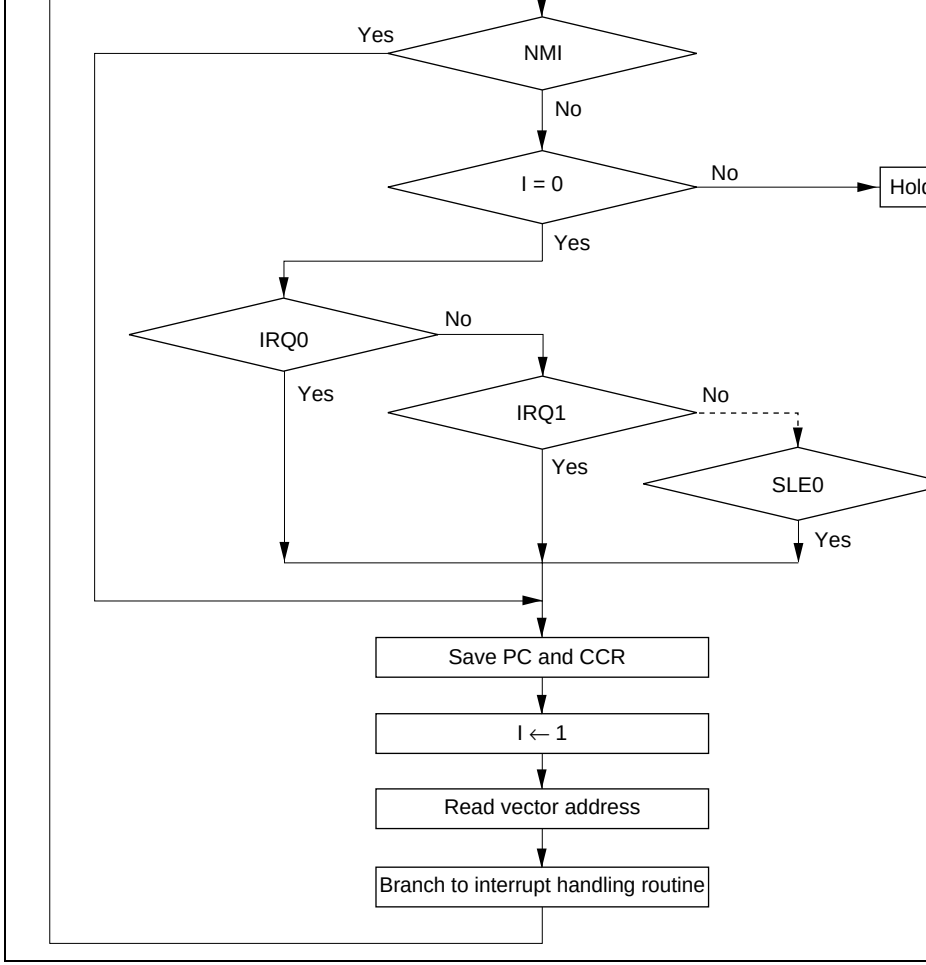


Figure 5.5 Flowchart of Procedure Up to Interrupt Acceptance in Interrupt Control Mode 0

[2] When interrupt requests are sent to the interrupt controller, the interrupt with the hig
priority according to the interrupt priority levels set in IPR is selected, and lower-pr
interrupt requests are held pending. If a number of interrupt requests with the same
generated at the same time, the interrupt request with the highest priority according
priority system shown in table 5.4 is selected.

[3] Next, the priority of the selected interrupt request is compared with the interrupt ma
in EXR. An interrupt request with a priority no higher than the mask level set at tha
held pending, and only an interrupt request with a priority higher than the interrupt
is accepted.

[4] When an interrupt request is accepted, interrupt exception handling starts after exec
current instruction has been completed.

[5] The PC, CCR, and EXR are saved to the stack area by interrupt exception handling.
saved on the stack shows the address of the first instruction to be executed after retu
the interrupt handling routine.

[6] The T bit in EXR is cleared to 0. The interrupt mask level is rewritten with the prio
the accepted interrupt.

If the accepted interrupt is NMI, the interrupt mask level is set to H'7.

[7] A vector address is generated for the accepted interrupt, and execution of the interru
handling routine starts at the address indicated by the contents of that vector addres

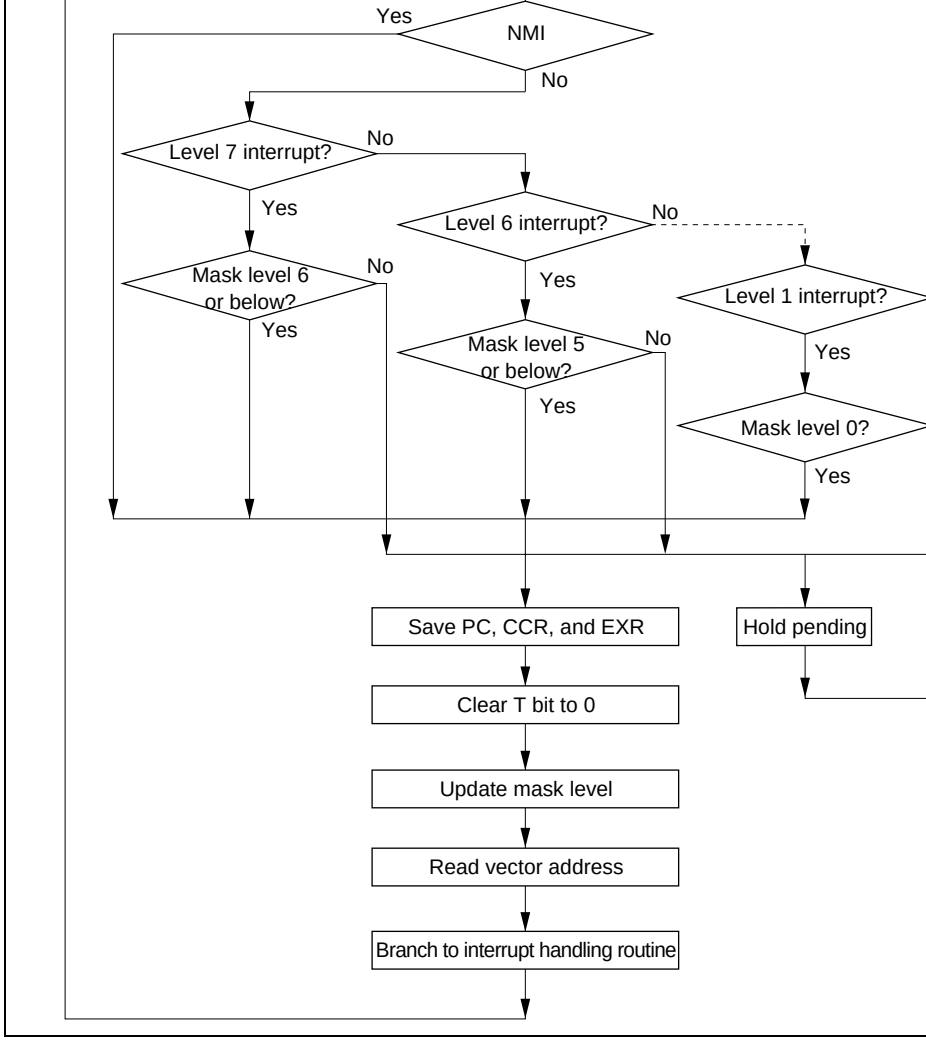


Figure 5.6 Flowchart of Procedure Up to Interrupt Acceptance in Interrupt Control Mode 2

Figure 5.7 Interrupt Exception Handling

Table 5.9 Interrupt Response Times

No.	Execution Status	Normal Mode*5		Advanc
		INTM1 = 0	INTM1 = 1	INTM1 = 0
1	Interrupt priority determination*1	3	3	3
2	Number of wait states until executing instruction ends*2	1 to $(19+2\cdot S_I)$	1 to $(19+2\cdot S_I)$	1 to $(19+2\cdot S_I)$
3	PC, CCR, EXR stack save	$2\cdot S_K$	$3\cdot S_K$	$2\cdot S_K$
4	Vector fetch	S_I	S_I	$2\cdot S_I$
5	Instruction fetch*3	$2\cdot S_I$	$2\cdot S_I$	$2\cdot S_I$
6	Internal processing*4	2	2	2
Total (using on-chip memory)		11 to 31	12 to 32	12 to 32

Notes: 1. Two states in case of internal interrupt.
2. Refers to MULXS and DIVXS instructions.
3. Prefetch after interrupt acceptance and interrupt handling routine prefetch.
4. Internal processing after interrupt acceptance and internal processing after
5. Not available in the H8S/2626 Group or H8S/2623 Group.

Branch address read	S_J
Stack manipulation	S_K

Legend:

m: Number of wait states in an external device access.

5.5 Usage Notes

5.5.1 Contention between Interrupt Generation and Disabling

When an interrupt enable bit is cleared to 0 to disable interrupts, the disabling becomes
after execution of the instruction.

In other words, when an interrupt enable bit is cleared to 0 by an instruction such as BC
MOV, if an interrupt is generated during execution of the instruction, the interrupt conc
still be enabled on completion of the instruction, and so interrupt exception handling fo
interrupt will be executed on completion of the instruction. However, if there is an inte
request of higher priority than that interrupt, interrupt exception handling will be execu
higher-priority interrupt, and the lower-priority interrupt will be ignored.

The same also applies when an interrupt source flag is cleared to 0.

Figure 5.8 shows an example in which the TCIEV bit in the TPU's TIER0 register is cl

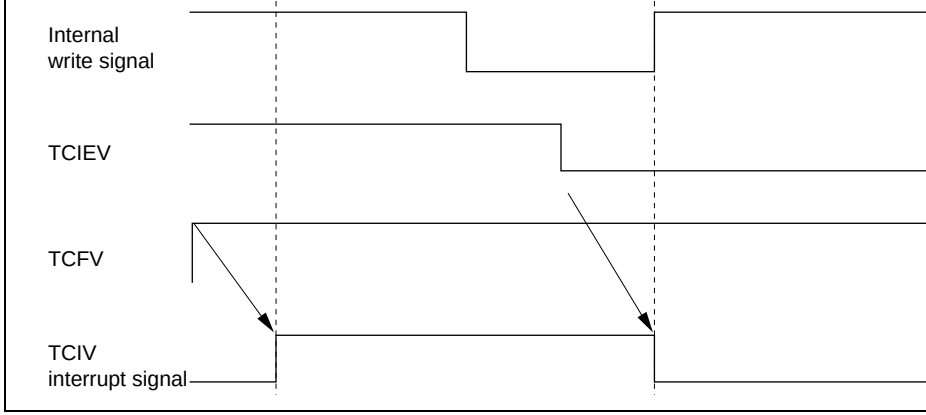


Figure 5.8 Contention between Interrupt Generation and Disabling

The above contention will not occur if an enable bit or interrupt source flag is cleared the interrupt is masked.

5.5.2 Instructions that Disable Interrupts

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

5.5.3 Times when Interrupts are Disabled

There are times when interrupt acceptance is disabled by the interrupt controller.

The interrupt controller disables interrupt acceptance for a 3-state period after the CPU updated the mask level with an LDC, ANDC, ORC, or XORC instruction.

case is the address of the next instruction.

Therefore, if an interrupt is generated during execution of an EEPMOV.W instruction, following coding should be used.

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

5.6 DTC Activation by Interrupt

5.6.1 Overview

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

- Interrupt request to CPU
- Activation request to DTC
- Selection of a number of the above

For details of interrupt requests that can be used with to activate the DTC, see section 8 Transfer Controller (DTC).

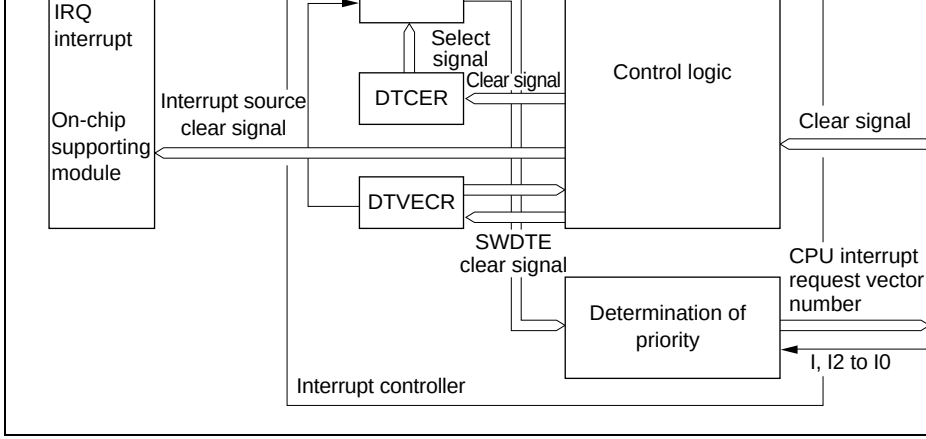


Figure 5.9 Interrupt Control for DTC

5.6.3 Operation

The interrupt controller has three main functions in DTC control.

(1) Selection of Interrupt Source

Interrupt sources can be specified as DTC activation requests or CPU interrupt reques
of the DTCE bit of DTCERA to DTCERG in the DTC.

After a DTC data transfer, the DTCE bit can be cleared to 0 and an interrupt request s
CPU in accordance with the specification of the DISEL bit of MRB in the DTC.

When the DTC has performed the specified number of data transfers and the transfer c
is zero, the DTCE bit is cleared to 0 and an interrupt request is sent to the CPU after t
transfer.

data transfer is performed first, followed by CPU interrupt exception handling.

Table 5.11 summarizes interrupt source selection and interrupt source clearance control to the settings of the DTCE bit of DTCERA to DTCERG in the DTC, and the DISEL b in the DTC.

Table 5.11 Interrupt Source Selection and Clearing Control

Settings			
DTC		Interrupt Source Selection/Clearing C	
DTCE	DISEL	DTC	CPU
0	*	X	Δ
1	0	Δ	X
	1	O	Δ

Legend:

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

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

X: The relevant bit cannot be used.

*: Don't care

(4) Notes on Use

SCI and A/D converter interrupt sources are cleared when the DTC reads or writes to th prescribed register.

instruction fetch, data read, data write, and data read/write.

6.1.1 Features

The PC break controller has the following features:

- Two break channels (A and B)
- The following can be set as break compare conditions:
  - 24 address bits
    Bit masking possible
  - Bus cycle
    Instruction fetch
    Data access: data read, data write, data read/write
  - Bus master
    Either CPU or CPU/DTC can be selected
- The timing of PC break exception handling after the occurrence of a break conditi follows:
  - Immediately before execution of the instruction fetched at the set address (inst fetch)
  - Immediately after execution of the instruction that accesses data at the set addr access)
- Module stop mode can be set
  - The initial setting is for PBC operation to be halted. Register access is enabled module stop mode.

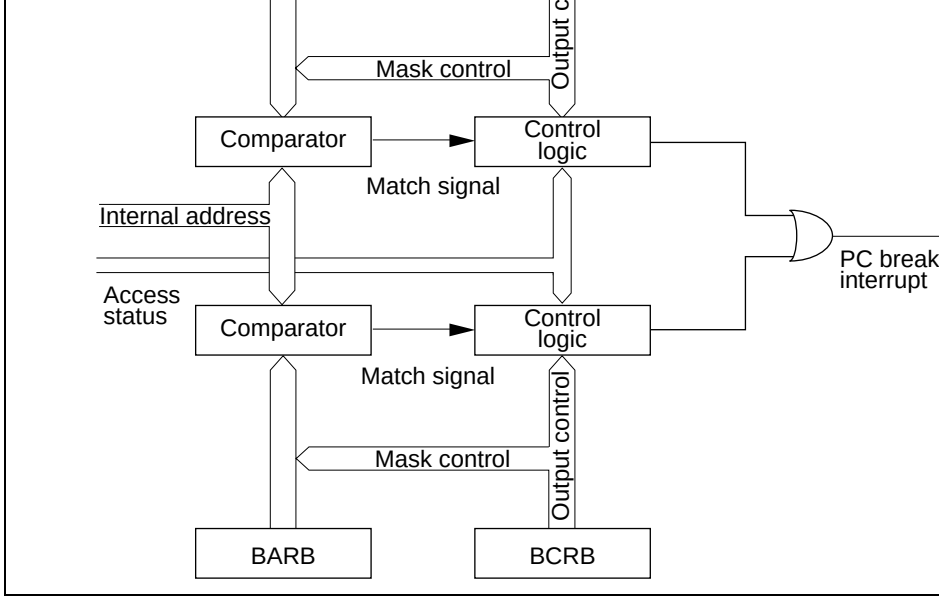


Figure 6.1 Block Diagram of PC Break Controller

Break address register A	BARA	R/W	H'000000	Retained
Break address register B	BARB	R/W	H'000000	Retained
Break control register A	BCRA	R/(W)*2	H'00	Retained
Break control register B	BCRB	R/(W)*2	H'00	Retained
Module stop control register C	MSTPCRC	R/W	H'FF	Retained

Notes: 1. Lower 16 bits of the address.
2. Only 0 can be written, for flag clearing.
3. Not available in the H8S/2626 Group or H8S/2623 Group.

6.2 Register Descriptions

6.2.1 Break Address Register A (BARA)

Bit	31	…	24	23	22	21	20	19	18	17	16	…	7	6	5	4	3	2
	—	…	—	BAA23	BAA22	BAA21	BAA20	BAA19	BAA18	BAA17	BAA16	…	BAA7	BAA6	BAA5	BAA4	BAA3	BA2
Initial value	Undefined	…	Undefined	0	0	0	0	0	0	0	0	…	0	0	0	0	0	0
R/W	—	…	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	…	R/W	R/W	R/W	R/W	R/W	R

BARA is a 32-bit readable/writable register that specifies the channel A break address

BAA23 to BAA0 are initialized to H'000000 by a reset and in hardware standby mode

Bits 31 to 24—Reserved: These bits return an undefined value if read, and cannot be

Bits 23 to 0—Break Address A23 to A0 (BAA23 to BAA0): These bits hold the cha
break address.

	CMFA	CDA	BAMRA2	BAMRA1	BAMRA0	CSELA1	CSELA0	
Initial value	0	0	0	0	0	0	0	
R/W	R/(W)*	R/W	R/W	R/W	R/W	R/W	R/W	

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

BCRA is an 8-bit readable/writable register that controls channel A PC breaks. BCRA (
the break condition bus master, (2) specifies bits subject to address comparison maskin
specifies whether the break condition is applied to an instruction fetch or a data access.
contains a condition match flag.

BCRA is initialized to H'00 by a reset and in hardware standby mode.

Bit 7—Condition Match Flag A (CMFA): Set to 1 when a break condition set for cha
satisfied. This flag is not cleared to 0.

Bit 7

CMFA	Description	
0	[Clearing condition]	
	When 0 is written to CMFA after reading CMFA = 1	(In
1	[Setting condition]	
	When a condition set for channel A is satisfied	

Bit 6—CPU Cycle/DTC Cycle Select A (CDA): Selects the channel A break conditio
master.

Bit 6

CDA	Description	
0	PC break is performed when CPU is bus master	(In
1	PC break is performed when CPU or DTC is bus master	

			conditions
	1	0	BAA1–0 (lower 2 bits) are masked, and not included conditions
		1	BAA2–0 (lower 3 bits) are masked, and not included conditions
1	0	0	BAA3–0 (lower 4 bits) are masked, and not included conditions
		1	BAA7–0 (lower 8 bits) are masked, and not included conditions
	1	0	BAA11–0 (lower 12 bits) are masked, and not includ conditions
		1	BAA15–0 (lower 16 bits) are masked, and not includ conditions

Bits 2 and 1—Break Condition Select A (CSELA1, CSELA0): These bits select an fetch, data read, data write, or data read/write cycle as the channel A break condition.

Bit 2	Bit 1	
CSELA1	**CSELA0**	**Description**
0	0	Instruction fetch is used as break condition (I
	1	Data read cycle is used as break condition
1	0	Data write cycle is used as break condition
	1	Data read/write cycle is used as break condition

6.2.4 Break Control Register B (BCRB)

BCRB is the channel B break control register. The bit configuration is the same as for B

6.2.5 Module Stop Control Register C (MSTPCRC)

Bit	7	6	5	4	3	2	1
	MSTPC7	MSTPC6	MSTPC5	MSTPC4	MSTPC3	MSTPC2	MSTPC1
Initial value	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MSTPCRC is an 8-bit readable/writable register that performs module stop mode contr

When the MSTPC4 bit is set to 1, PC break controller operation is stopped at the end o cycle, and module stop mode is entered. Register read/write accesses are not possible i stop mode. For details, see sections 21A.5, 21B.5, Module Stop Mode.

MSTPCRC is initialized to H'FF by a power on reset and in hardware standby mode. It initialized in software standby mode.

Bit 4—Module Stop (MSTPC4): Specifies the PC break controller module stop mode

Bit 4

MSTPC4	Description
0	PC break controller module stop mode is cleared
1	PC break controller module stop mode is set (In

(1) Initial settings

— Set the break address in BARA. For a PC break caused by an instruction fetch, address of the first instruction byte as the break address.

— Set the break conditions in BCRA.

BCRA bit 6 (CDA): With a PC break caused by an instruction fetch, the bus m be the CPU. Set 0 to select the CPU.

BCRA bits 5–3 (BAMA2–0): Set the address bits to be masked.

BCRA bits 2, 1 (CSELA1, 0): Set 00 to specify an instruction fetch as the bre

BCRA bit 0 (BIEA): Set to 1 to enable break interrupts.

(2) Satisfaction of break condition

— When the instruction at the set address is fetched, a PC break request is genera immediately before execution of the fetched instruction, and the condition mat (CMFA) is set.

(3) Interrupt handling

— After priority determination by the interrupt controller, PC break interrupt exce handling is started.

BCRA bit 0 (CDA): Select the bus master.

BCRA bits 5–3 (BAMA2–0): Set the address bits to be masked.

BCRA bits 2, 1 (CSELA1, 0): Set 01, 10, or 11 to specify data access as the br condition.

BCRA bit 0 (BIEA): Set to 1 to enable break interrupts.

(2) Satisfaction of break condition

— After execution of the instruction that performs a data access on the set address, request is generated and the condition match flag (CMFA) is set.

(3) Interrupt handling

— After priority determination by the interrupt controller, PC break interrupt excep handling is started.

6.3.3 Notes on PC Break Interrupt Handling

(1) The PC break interrupt is shared by channels A and B. The channel from which the was issued must be determined by the interrupt handler.

(2) The CMFA and CMFB flags are not cleared to 0, so 0 must be written to CMFA or after first reading the flag while it is set to 1. If the flag is left set to 1, another interr requested after interrupt handling ends.

(3) A PC break interrupt generated when the DTC is the bus master is accepted after th been transferred to the CPU by the bus controller.

mode, and PC break interrupt handling is executed. After execution of PC break in handling, the instruction at the address after the SLEEP instruction is executed (fig

(2) When the SLEEP instruction causes a transition from high-speed (medium-speed) subactive mode:

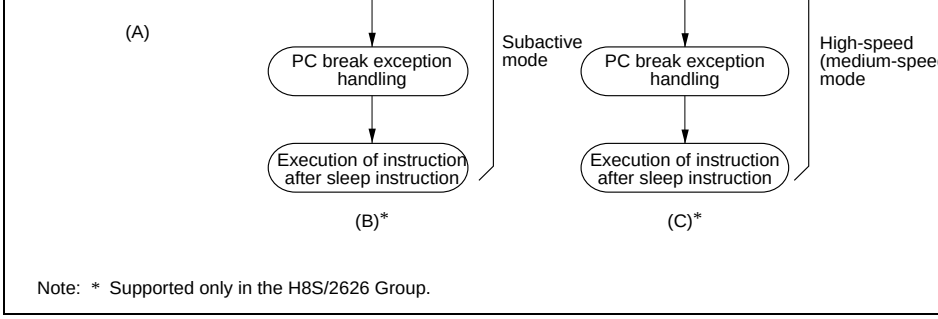
After execution of the SLEEP instruction, a transition is made to subactive mode v transition exception handling. After the transition, PC break interrupt handling is e then the instruction at the address after the SLEEP instruction is executed (figure 6 (Supported only in the H8S/2626 Group).

(3) When the SLEEP instruction causes a transition from subactive mode to high-spee speed) mode:

After execution of the SLEEP instruction, and following the clock oscillation settl transition is made to high-speed (medium-speed) mode via direct transition except handling. After the transition, PC break interrupt handling is executed, then the ins the address after the SLEEP instruction is executed (figure 6.2 (C)) (Supported onl H8S/2626 Group).

(4) When the SLEEP instruction causes a transition to software standby mode or watc

After execution of the SLEEP instruction, a transition is made to the respective mo break interrupt handling is not executed. However, the CMFA or CMFB flag is set (D)).



Note: * Supported only in the H8S/2626 Group.

Figure 6.2 Operation in Power-Down Mode Transitions

6.3.5 PC Break Operation in Continuous Data Transfer

If a PC break interrupt is generated when the following operations are being performed handling is executed on completion of the specified transfer.

(1) When a PC break interrupt is generated at the transfer address of an EEPMOV.B in
PC break exception handling is executed after all data transfers have been complete EEPMOV.B instruction has ended.

(2) When a PC break interrupt is generated at a DTC transfer address:
PC break exception handling is executed after the DTC has completed the specified data transfers, or after data for which the DISEL bit is set to 1 has been transferred.

(2) When break interruption by instruction fetch is set, the set address indicates on-chi
RAM space, and that address is used for data access, the instruction that executes t
access is one state later than in normal operation.

(3) When break interruption by instruction fetch is set and a break interrupt is generat
executing instruction immediately preceding the set instruction has one of the addr
modes shown below, and that address indicates on-chip ROM or RAM, and that ac
used for data access, the instruction will be one state later than in normal operation

@ERn, @(d:16,ERn), @(d:32,ERn), @-ERn/ERn+, @aa:8, @aa:24, @aa:32, @(d
@(d:16,PC), @@aa:8

(4) When break interruption by instruction fetch is set and a break interrupt is generat
executing instruction immediately preceding the set instruction is NOP or SLEEP,
#xx,Rn as its addressing mode, and that instruction is located in on-chip ROM or R
instruction will be one state later than in normal operation.

(2) When the I bit is set by an LDC, ANDC, ORC, or XORC instruction, a PC break in
becomes valid two states after the end of the executing instruction. If a PC break int
set for the instruction following one of these instructions, since interrupts, including
disabled for a 3-state period in the case of LDC, ANDC, ORC, and XORC, the next
is always executed. For details, see section 5, Interrupt Controller.

(3) When a PC break is set for an instruction fetch at the address following a Bcc instru

A PC break interrupt is generated if the instruction at the next address is executed in
accordance with the branch condition, but is not generated if the instruction at the n
is not executed.

(4) When a PC break is set for an instruction fetch at the branch destination address of
instruction:

A PC break interrupt is generated if the instruction at the branch destination is exec
accordance with the branch condition, but is not generated if the instruction at the b
destination is not executed.

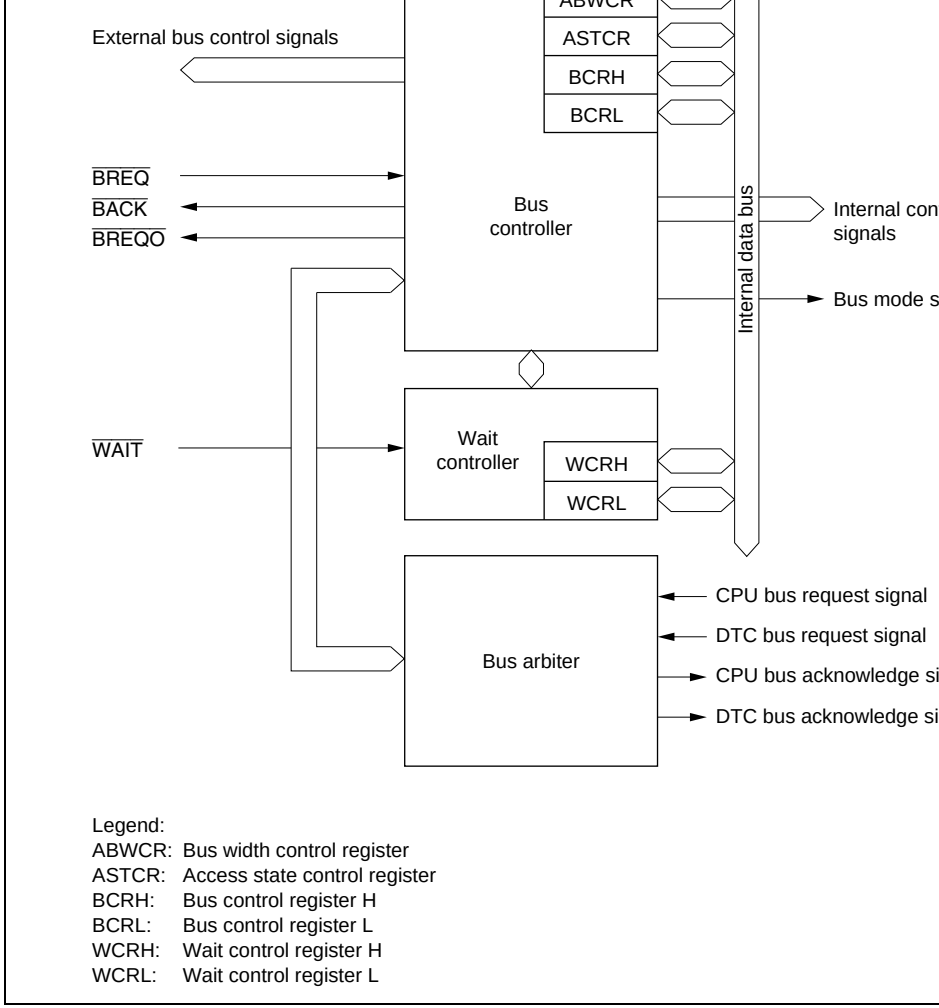
connected easily.

The bus controller also has a bus arbitration function, and controls the operation of the masters: the CPU and data transfer controller (DTC).

7.1.1 Features

The features of the bus controller are listed below.

- Manages external address space in area units
  - Manages the external space as 8 areas of 2 Mbytes
  - Bus specifications can be set independently for each area
  - Burst ROM interface can be set
- Basic bus interface
  - 8-bit access or 16-bit access can be selected for each area
  - 2-state access or 3-state access can be selected for each area
  - Program wait states can be inserted for each area
- Burst ROM interface
  - Burst ROM interface can be set for area 0
  - Choice of 1- or 2-state burst access
- Idle cycle insertion
  - An idle cycle can be inserted in case of an external read cycle between differen
  - An idle cycle can be inserted in case of an external write cycle immediately aft external read cycle
- Write buffer functions
  - External write cycle and internal access can be executed in parallel
- Bus arbitration function
  - Includes a bus arbiter that arbitrates bus mastership among the CPU and DTC
- Other features
  - External bus release function



Legend:
ABWCR: Bus width control register
ASTCR: Access state control register
BCRH: Bus control register H
BCRL: Bus control register L
WCRH: Wait control register H
WCRL: Wait control register L

Figure 7.1 Block Diagram of Bus Controller

Read	$\overline{RD}$	Output	Strobe signal indicating that external space read.
High write	$\overline{HWR}$	Output	Strobe signal indicating that external space written, and upper half (D15 to D8) of data enabled.
Low write	$\overline{LWR}$	Output	Strobe signal indicating that external space written, and lower half (D7 to D0) of data b enabled.
Wait	$\overline{WAIT}$	Input	Wait request signal when accessing extern access space.
Bus request	$\overline{BREQ}$	Input	Request signal that releases bus to extern
Bus request acknowledge	$\overline{BACK}$	Output	Acknowledge signal indicating that bus has released.
Bus request output	$\overline{BREQO}$	Output	External bus request signal used when inte master accesses external space when exte released.

Access state control register	ASTCR	R/W	H'FF	Retained
Wait control register H	WCRH	R/W	H'FF	Retained
Wait control register L	WCRL	R/W	H'FF	Retained
Bus control register H	BCRH	R/W	H'D0	Retained
Bus control register L	BCRL	R/W	H'08	Retained
Pin function control register	PFCR	R/W	H'0D/H'00	Retained

Notes: 1. Lower 16 bits of the address.
2. Determined by the MCU operating mode.
3. Not available in the H8S/2623 Group.

Initial value	:	1	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Mode 4								
Initial value	:	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ABWCR is an 8-bit readable/writable register that designates each area for either 8-bi
16-bit access.

ABWCR sets the data bus width for the external memory space. The bus width for on-
memory and internal I/O registers is fixed regardless of the settings in ABWCR.

After a reset and in hardware standby mode, ABWCR is initialized to H'FF in modes 5
H'00 in mode 4. It is not initialized in software standby mode.

Bits 7 to 0—Area 7 to 0 Bus Width Control (ABW7 to ABW0): These bits select w
corresponding area is to be designated for 8-bit access or 16-bit access.

Bit n

ABWn	Description
0	Area n is designated for 16-bit access
1	Area n is designated for 8-bit access

space or a 3-state access space.

ASTCR sets the number of access states for the external memory space. The number of states for on-chip memory and internal I/O registers is fixed regardless of the settings in

ASTCR is initialized to H'FF by a reset and in hardware standby mode. It is not initializ software standby mode.

Bits 7 to 0—Area 7 to 0 Access State Control (AST7 to AST0): These bits select wh corresponding area is to be designated as a 2-state access space or a 3-state access space

Wait state insertion is enabled or disabled at the same time.

Bit n

ASTn	Description
0	Area n is designated for 2-state access Wait state insertion in area n external space is disabled
1	Area n is designated for 3-state access Wait state insertion in area n external space is enabled

(1) WCRH

Bit	:	7	6	5	4	3	2	1
		W71	W70	W61	W60	W51	W50	W41
Initial value	:	1	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 7 and 6—Area 7 Wait Control 1 and 0 (W71, W70): These bits select the num
program wait states when area 7 in external space is accessed while the AST7 bit in A
to 1.

Bit 7	Bit 6	
W71	**W70**	**Description**
0	0	Program wait not inserted when external space area 7 is acce
	1	1 program wait state inserted when external space area 7 is a
1	0	2 program wait states inserted when external space area 7 is
	1	3 program wait states inserted when external space area 7 is

1	0	2 program wait states inserted when external space area 6 is a
	1	3 program wait states inserted when external space area 6 is a (I

Bits 3 and 2—Area 5 Wait Control 1 and 0 (W51, W50): These bits select the numb program wait states when area 5 in external space is accessed while the AST5 bit in AS to 1.

Bit 3	Bit 2	
W51	**W50**	**Description**
0	0	Program wait not inserted when external space area 5 is acces
	1	1 program wait state inserted when external space area 5 is ac
1	0	2 program wait states inserted when external space area 5 is a
	1	3 program wait states inserted when external space area 5 is a (I

Bits 1 and 0—Area 4 Wait Control 1 and 0 (W41, W40): These bits select the numb program wait states when area 4 in external space is accessed while the AST4 bit in AS to 1.

Bit 1	Bit 0	
W41	**W40**	**Description**
0	0	Program wait not inserted when external space area 4 is acces
	1	1 program wait state inserted when external space area 4 is ac
1	0	2 program wait states inserted when external space area 4 is a
	1	3 program wait states inserted when external space area 4 is a (I

program wait states when area 3 in external space is accessed while the AST3 bit in A
to 1.

Bit 7	Bit 6	
W31	**W30**	**Description**
0	0	Program wait not inserted when external space area 3 is acce
	1	1 program wait state inserted when external space area 3 is a
1	0	2 program wait states inserted when external space area 3 is
	1	3 program wait states inserted when external space area 3 is

Bits 5 and 4—Area 2 Wait Control 1 and 0 (W21, W20): These bits select the num
program wait states when area 2 in external space is accessed while the AST2 bit in A
to 1.

Bit 5	Bit 4	
W21	**W20**	**Description**
0	0	Program wait not inserted when external space area 2 is acce
	1	1 program wait state inserted when external space area 2 is a
1	0	2 program wait states inserted when external space area 2 is
	1	3 program wait states inserted when external space area 2 is

1	0	2 program wait states inserted when external space area 1 is a
	1	3 program wait states inserted when external space area 1 is a (I

Bits 1 and 0—Area 0 Wait Control 1 and 0 (W01, W00): These bits select the numb program wait states when area 0 in external space is accessed while the AST0 bit in AS to 1.

Bit 1	Bit 0	
W01	**W00**	**Description**
0	0	Program wait not inserted when external space area 0 is acces
	1	1 program wait state inserted when external space area 0 is ac
1	0	2 program wait states inserted when external space area 0 is a
	1	3 program wait states inserted when external space area 0 is a (I

insertion, and the memory interface for area 0.

BCRH is initialized to H'D0 by a reset and in hardware standby mode. It is not initiali
software standby mode.

Bit 7—Idle Cycle Insert 1 (ICIS1): Selects whether or not one idle cycle state is to b
between bus cycles when successive external read cycles are performed in different ar

Bit 7

ICIS1	Description
0	Idle cycle not inserted in case of successive external read cycles in differe
1	Idle cycle inserted in case of successive external read cycles in different a

Bit 6—Idle Cycle Insert 0 (ICIS0): Selects whether or not one idle cycle state is to b
between bus cycles when successive external read and external write cycles are perfor

Bit 6

ICIS0	Description
0	Idle cycle not inserted in case of successive external read and external wr
1	Idle cycle inserted in case of successive external read and external write c

Bit 4—Burst Cycle Select 1 (BRSTS1): Selects the number of burst cycles for the bur
interface.

Bit 4

BRSTS1	Description
0	Burst cycle comprises 1 state
1	Burst cycle comprises 2 states (I

Bit 3—Burst Cycle Select 0 (BRSTS0): Selects the number of words that can be acce
burst ROM interface burst access.

Bit 3

BRSTS0	Description
0	Max. 4 words in burst access (I
1	Max. 8 words in burst access

Bits 2 to 0—Reserved: Only 0 should be written to these bits.

state protocol, enabling or disabling of the write data buffer function, and enabling or $\overline{\text{WAIT}}$ pin input.

BCRL is initialized to H'08 by a reset and in hardware standby mode. It is not initializ software standby mode.

Bit 7—Bus Release Enable (BRLE): Enables or disables external bus release.

Bit 7

BRLE	Description
0	External bus release is disabled. $\overline{\text{BREQ}}$, $\overline{\text{BACK}}$, and $\overline{\text{BREQO}}$ can be used ports.
1	External bus release is enabled.

Bit 6—BREQO Pin Enable (BREQOE): Outputs a signal that requests the external to drop the bus request signal ($\overline{\text{BREQ}}$) in the external bus release state, when an intern master performs an external space access, or when a refresh request is generated.

Bit 6

BREQOE	Description
0	$\overline{\text{BREQO}}$ output disabled. $\overline{\text{BREQO}}$ can be used as I/O port.
1	$\overline{\text{BREQO}}$ output enabled.

Bit 5—Reserved: This bit cannot be modified and is always read as 0.

Bit 4—Reserved: Only 0 should be written to this bit.

Bit 3—Reserved: Only 1 should be written to this bit.

Bit 2—Reserved: Only 0 should be written to this bit.

Bit 0—WAIT Pin Enable (WAITE): Selects enabling or disabling of wait input by th
pin.

Bit 0	
WAITE	**Description**
0	Wait input by $\overline{\text{WAIT}}$ pin disabled. $\overline{\text{WAIT}}$ pin can be used as I/O port. (
1	Wait input by $\overline{\text{WAIT}}$ pin enabled

7.2.6 Pin Function Control Register (PFCR)

Bit	:	7	6	5	4	3	2	1
		—	—	BUZZE	—	AE3	AE2	AE1
Initial value	:	0	0	0	0	1/0	1/0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PFCR is an 8-bit readable/writable register that performs address output control in exte
expanded mode.

PFCR is initialized to H'0D/H'00 by a reset and in hardware standby mode. It retains its
state in software standby mode.

Bits 7 and 6—Reserved: Only 0 should be written to these bits.

Bit 5—BUZZ Output Enable (BUZZE): Enables or disables BUZZ output from the P
details, see section 12.2.4, Pin Function Control Register (PFCR).

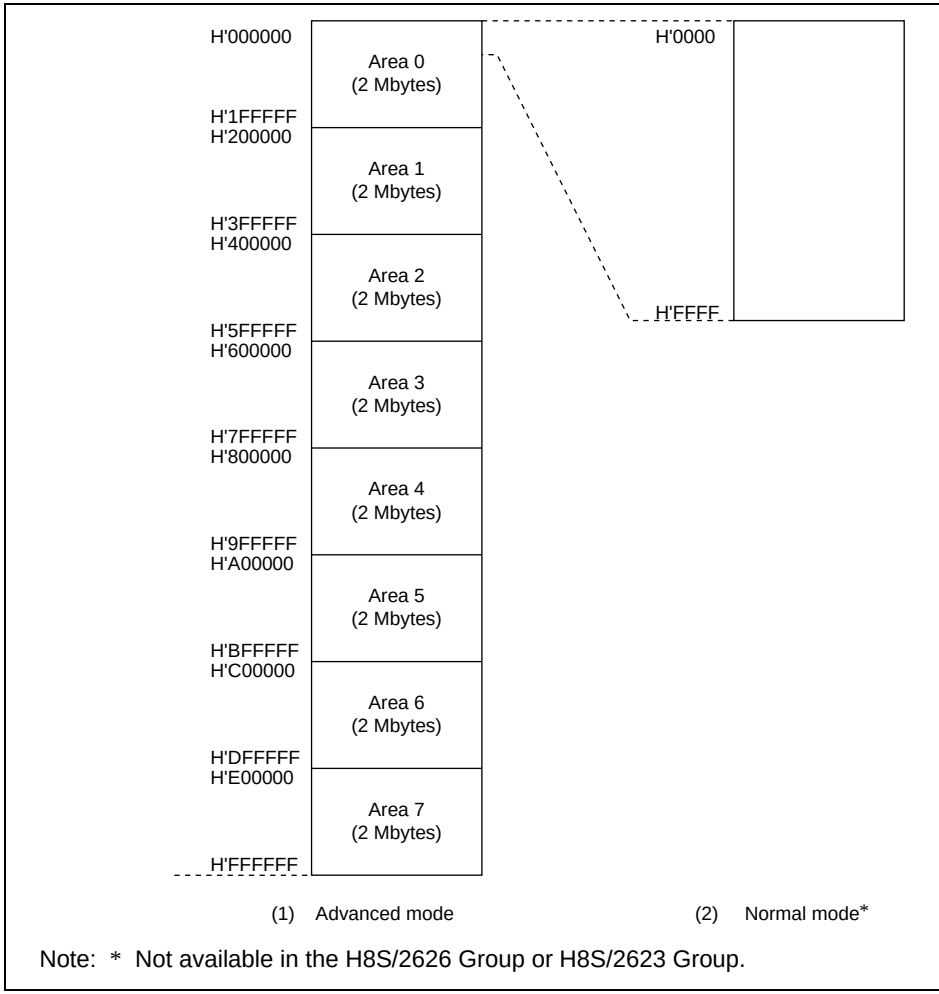
Bit 4—Reserved: Only 0 should be written to this bit.

Bits 3 to 0—Address Output Enable 3 to 0 (AE3 to AE0): These bits select enabling
disabling of address outputs A8 to A23 in ROMless expanded mode and modes with R
a pin is enabled for address output, the address is output regardless of the correspondin

				disabled
			1	A8–A10 address output enabled; A11–A23 address disabled
	1	0	0	A8–A11 address output enabled; A12–A23 address disabled
			1	A8–A12 address output enabled; A13–A23 address disabled
		1	0	A8–A13 address output enabled; A14–A23 address disabled
			1	A8–A14 address output enabled; A15–A23 address disabled
1	0	0	0	A8–A15 address output enabled; A16–A23 address disabled
			1	A8–A16 address output enabled; A17–A23 address disabled
		1	0	A8–A17 address output enabled; A18–A23 address disabled
			1	A8–A18 address output enabled; A19–A23 address disabled
	1	0	0	A8–A19 address output enabled; A20–A23 address disabled
			1	A8–A20 address output enabled; A21–A23 address disabled (In
		1	0	A8–A21 address output enabled; A22, A23 address disabled
			1	A8–A23 address output enabled

Note: * In expanded mode with ROM, bits AE3 to AE0 are initialized to B'0000.
In ROMless expanded mode, bits AE3 to AE0 are initialized to B'1101.
Address pins A0 to A7 are made address outputs by setting the correspond bits to 1.

Note: * Not available in the H8S/2626 Group or H8S/2623 Group.



Note: * Not available in the H8S/2626 Group or H8S/2623 Group.

Figure 7.2 Overview of Area Partitioning

8-bit bus is selected functions as an 8-bit access space, and an area for which a 16-bit
selected functions as a16-bit access space.

If all areas are designated for 8-bit access, 8-bit bus mode is set; if any area is designa
access, 16-bit bus mode is set. When the burst ROM interface is designated, 16-bit bu
always set.

(2) Number of Access States: Two or three access states can be selected with ASTC
for which 2-state access is selected functions as a 2-state access space, and an area for
state access is selected functions as a 3-state access space.

With the burst ROM interface, the number of access states may be determined withou
ASTCR.

When 2-state access space is designated, wait insertion is disabled.

(3) Number of Program Wait States: When 3-state access space is designated by A
number of program wait states to be inserted automatically is selected with WCRH an
From 0 to 3 program wait states can be selected.

Table 7.3 shows the bus specifications for each basic bus interface area.

		1	0			2
			1			3
1	0	—	—	8	2	0
	1	0	0		3	0
			1			1
		1	0			2
			1			3

7.3.3 Memory Interfaces

The H8S/2626 Group and H8S/2623 Group memory interfaces comprise a basic bus in
allows direct connection or ROM, SRAM, and so on, and a burst ROM interface that al
connection of burst ROM. The memory interface can be selected independently for eac

An area for which the basic bus interface is designated functions as normal space, and a
which the burst ROM interface is designated functions as burst ROM space.

space.

Either basic bus interface or burst ROM interface can be selected for area 0.

Areas 1 to 6: In external expansion mode, all of areas 1 to 6 is external space.

Only the basic bus interface can be used for areas 1 to 6.

Area 7: Area 7 includes the on-chip RAM and internal I/O registers. In external expan the space excluding the on-chip RAM and internal I/O registers is external space. The RAM is enabled when the RAME bit in the system control register (SYSCR) is set to RAME bit is cleared to 0, the on-chip RAM is disabled and the corresponding space b external space.

Only the basic bus interface can be used for the area 7.

7.4.2 Data Size and Data Alignment

Data sizes for the CPU and other internal bus masters are byte, word, and longword. Th
controller has a data alignment function, and when accessing external space, controls w
upper data bus (D15 to D8) or lower data bus (D7 to D0) is used according to the bus
specifications for the area being accessed (8-bit access space or 16-bit access space) an
size.

8-Bit Access Space: Figure 7.3 illustrates data alignment control for the 8-bit access sp
the 8-bit access space, the upper data bus (D15 to D8) is always used for accesses. The
data that can be accessed at one time is one byte: a word transfer instruction is perform
byte accesses, and a longword transfer instruction, as four byte accesses.

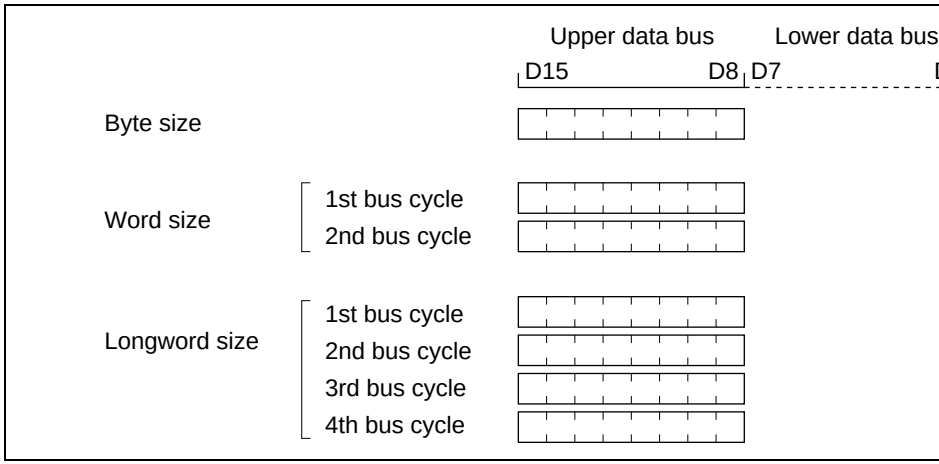


Figure 7.3 Access Sizes and Data Alignment Control (8-Bit Access Spac

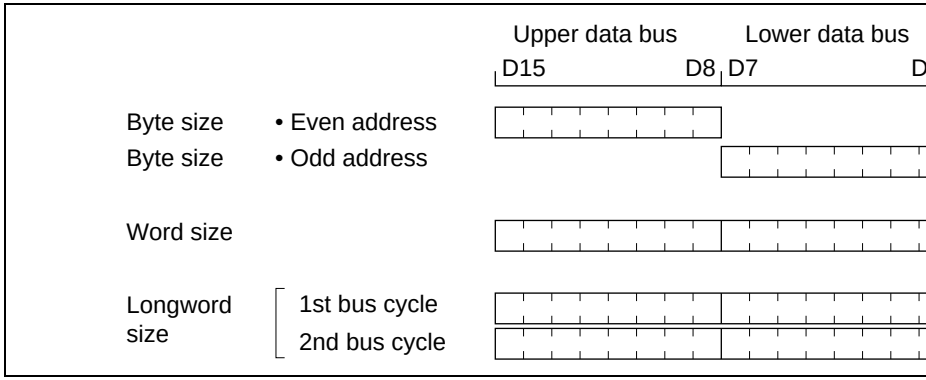


Figure 7.4 Access Sizes and Data Alignment Control (16-Bit Access Spa

Table 7.4 Data Buses Used and Valid Strobes

Area	Access Size	Read/ Write	Address	Valid Strobe	Upper Data Bus (D15 to D8)	Lowe (D7 to
8-bit access space	Byte	Read	—	$\overline{RD}$	Valid	Invalid
		Write	—	$\overline{HWR}$		Hi-Z
16-bit access space	Byte	Read	Even	$\overline{RD}$	Valid	Invalid
			Odd		Invalid	Valid
		Write	Even	$\overline{HWR}$	Valid	Hi-Z
			Odd	$\overline{LWR}$	Hi-Z	Valid
	Word	Read	—	$\overline{RD}$	Valid	Valid
		Write	—	$\overline{HWR}$, $\overline{LWR}$	Valid	Valid

Notes: Hi-Z: High impedance.
Invalid: Input state; input value is ignored.

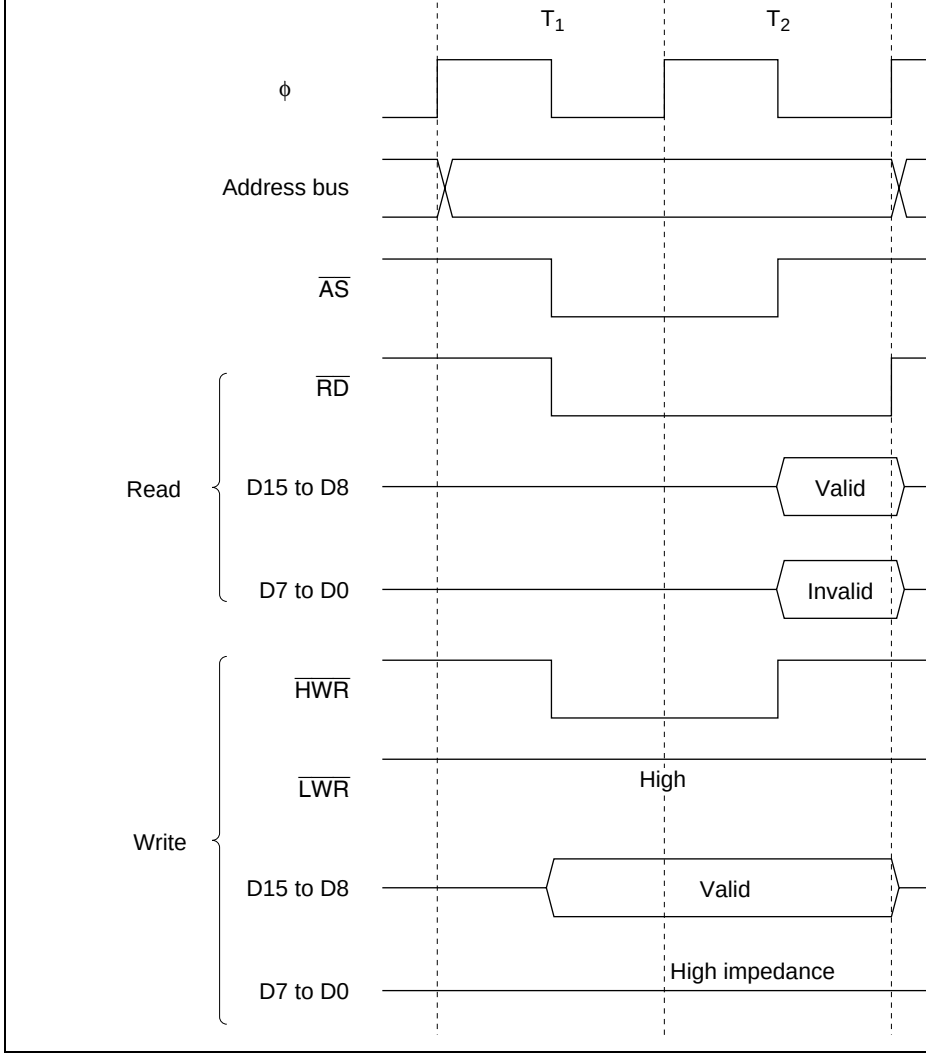


Figure 7.5 Bus Timing for 8-Bit 2-State Access Space

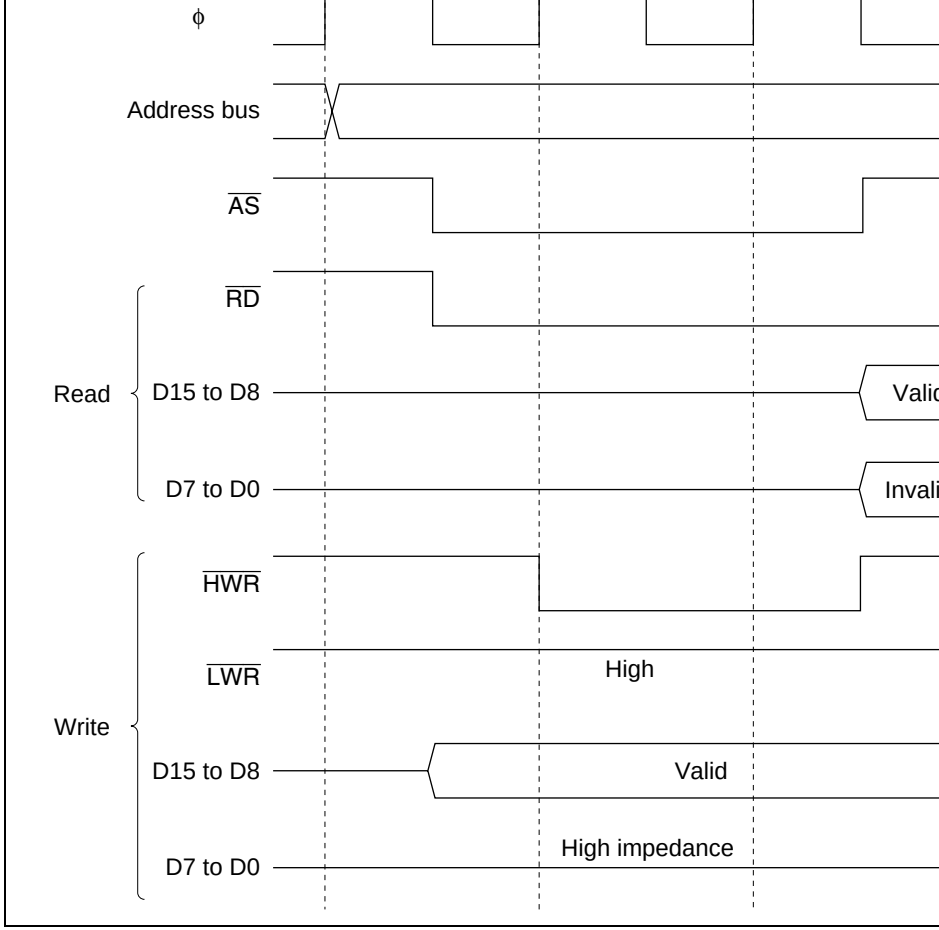


Figure 7.6 Bus Timing for 8-Bit 3-State Access Space

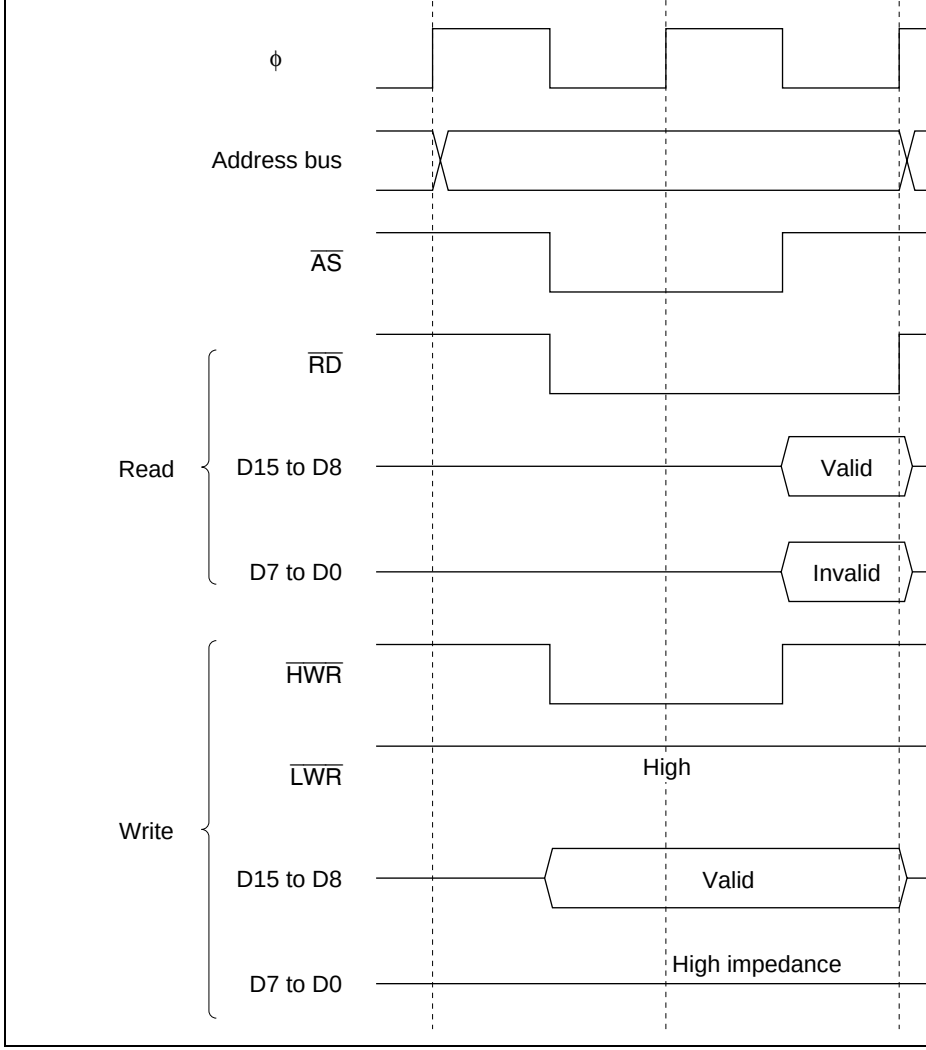


Figure 7.7 Bus Timing for 16-Bit 2-State Access Space (1) (Even Address Byt

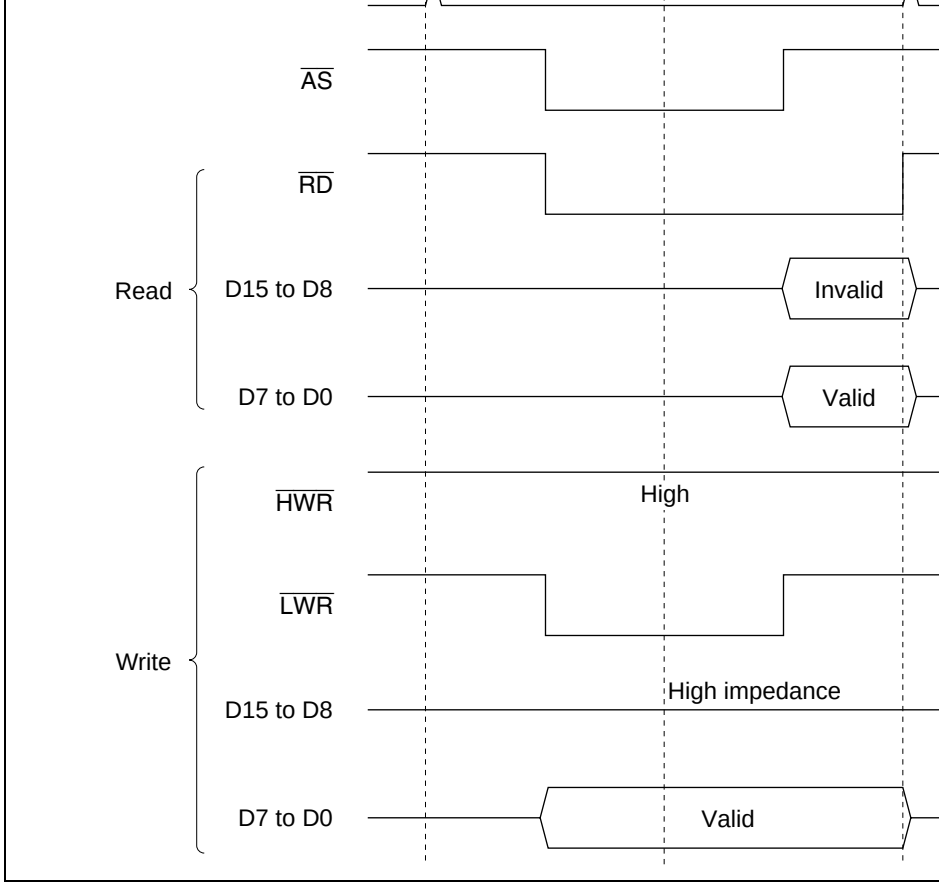


Figure 7.8 Bus Timing for 16-Bit 2-State Access Space (2) (Odd Address Byte

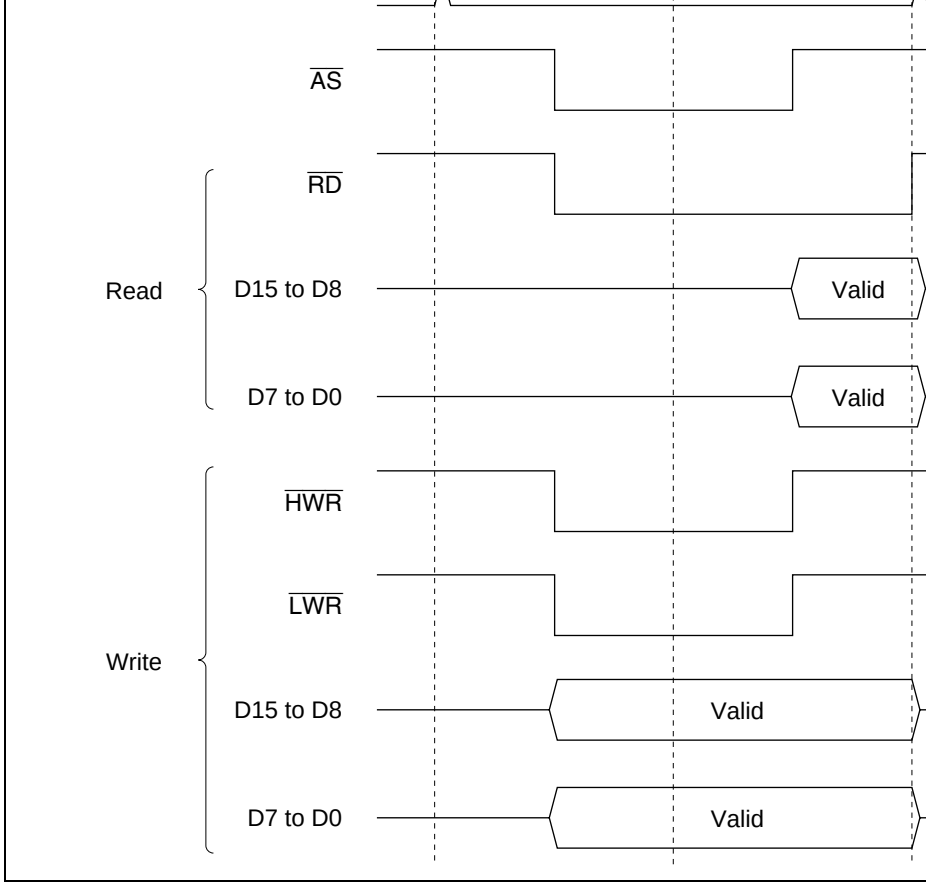


Figure 7.9 Bus Timing for 16-Bit 2-State Access Space (3) (Word Acce

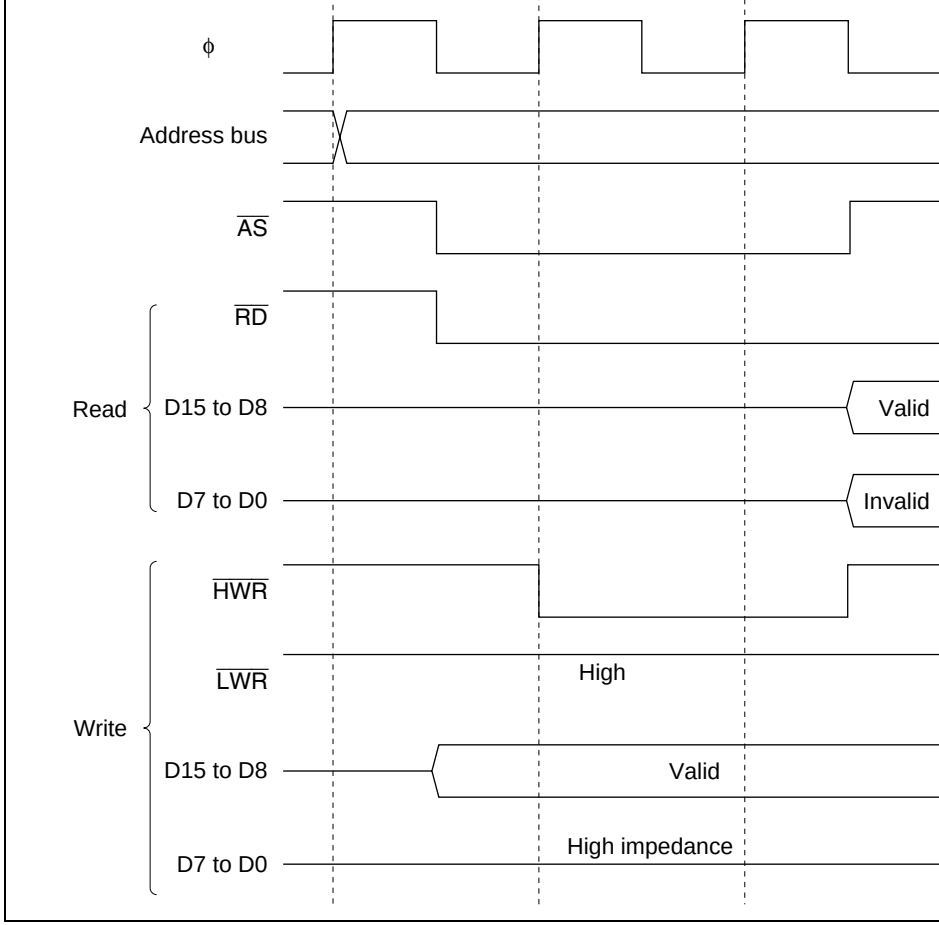


Figure 7.10 Bus Timing for 16-Bit 3-State Access Space (1) (Even Address Byte

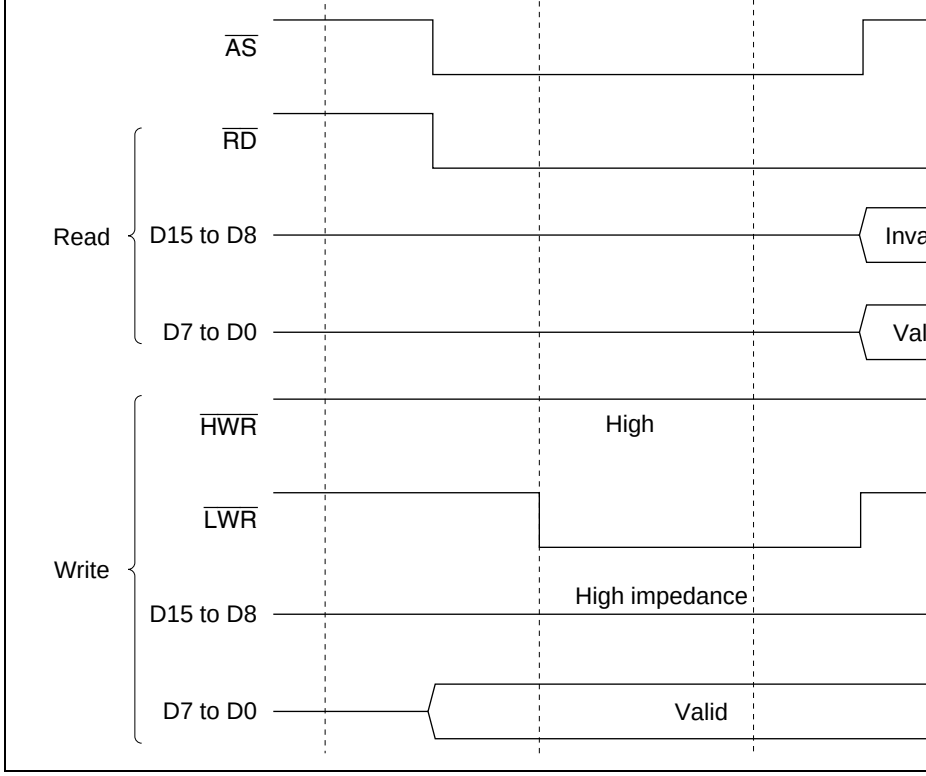


Figure 7.11 Bus Timing for 16-Bit 3-State Access Space (2) (Odd Address Byt

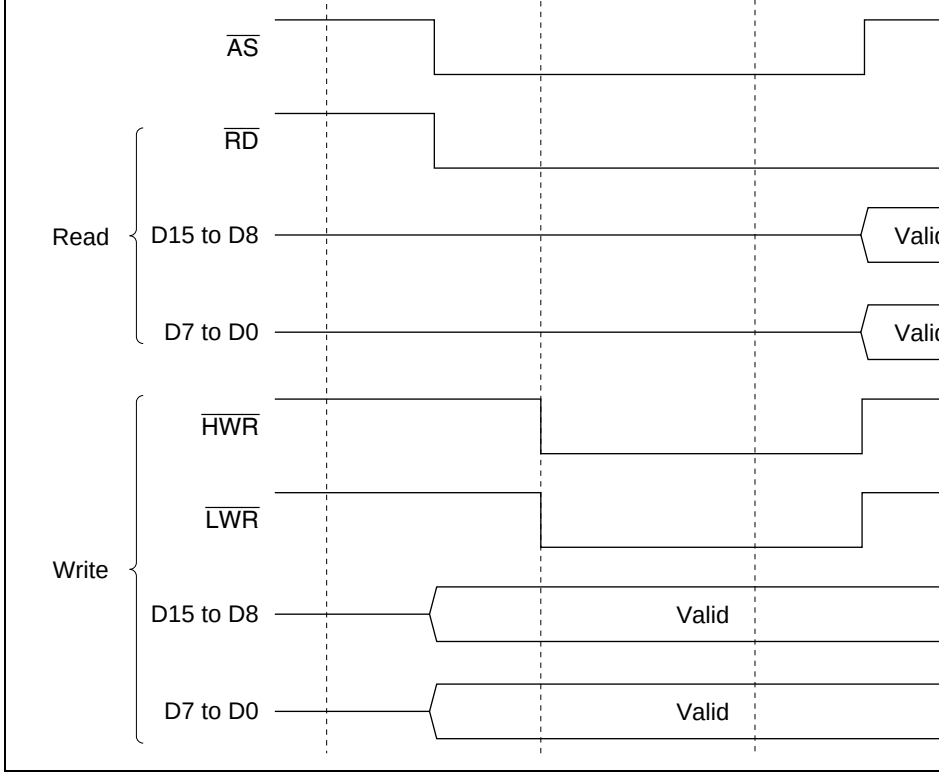


Figure 7.12 Bus Timing for 16-Bit 3-State Access Space (3) (Word Acces

individual area basis in 3-state access space, according to the settings of WCRH and W

Pin Wait Insertion

Setting the WAITE bit in BCRL to 1 enables wait insertion by means of the $\overline{\text{WAIT}}$ pi wait insertion is first carried out according to the settings in WCRH and WCRL. Then $\overline{\text{WAIT}}$ pin is low at the falling edge of ϕ in the last T_2 or T_w state, a T_w state is inserted $\overline{\text{WAIT}}$ pin is held low, T_w states are inserted until it goes high.

This is useful when inserting four or more T_w states, or when changing the number of different external devices.

The WAITE bit setting applies to all areas.

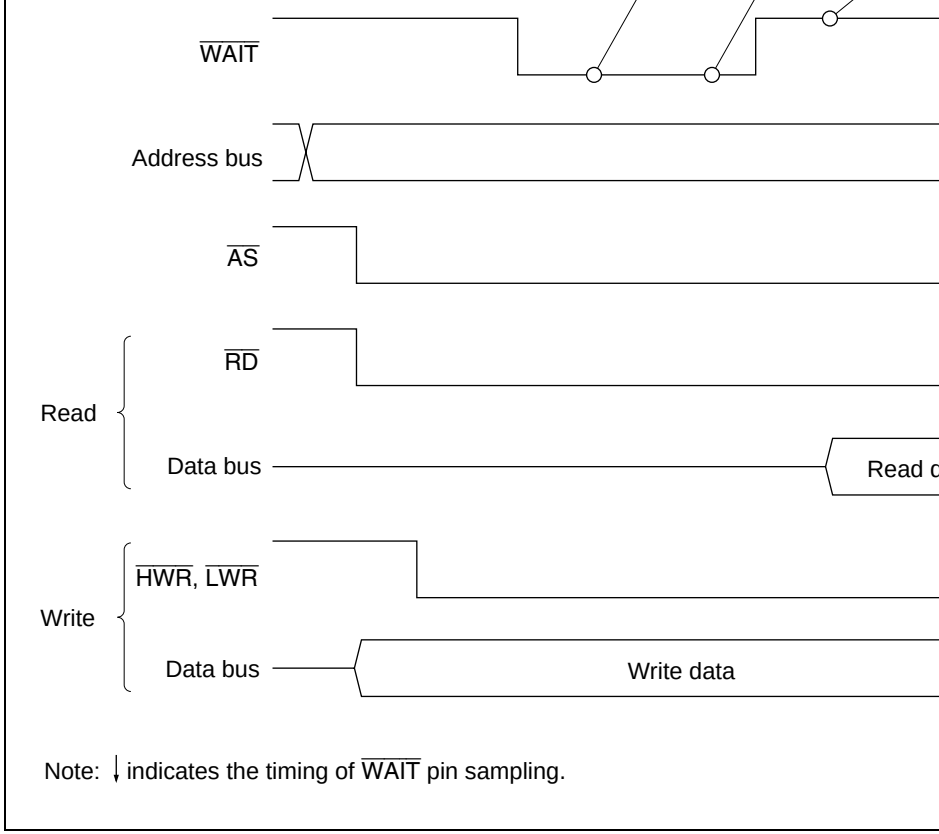


Figure 7.13 Example of Wait State Insertion Timing

The settings after a reset are: 3-state access, 3 program wait state insertion, and WAIT i
disabled.

Area 0 can be designated as burst ROM space by means of the BRSTRM bit in BCRH. Consecutive burst accesses of a maximum of 4 words or 8 words can be performed for instruction fetches only. One or two states can be selected for burst access.

7.5.2 Basic Timing

The number of states in the initial cycle (full access) of the burst ROM interface is in accordance with the setting of the AST0 bit in ASTCR. Also, when the AST0 bit is set to 1, wait state insertion is possible. One or two states can be selected for the burst cycle, according to the setting of the BRSTS1 bit in BCRH. Wait states cannot be inserted. When area 0 is designated as burst ROM space, it becomes 16-bit access space regardless of the setting of the ABW0 bit in ABWCR.

When the BRSTS0 bit in BCRH is cleared to 0, burst access of up to 4 words is performed. When the BRSTS0 bit is set to 1, burst access of up to 8 words is performed.

The basic access timing for burst ROM space is shown in figures 7.14 (a) and (b). The timing shown in figure 7.14 (a) is for the case where the AST0 and BRSTS1 bits are both set to 1, and that in figure 7.14 (b) is for the case where both these bits are cleared to 0.

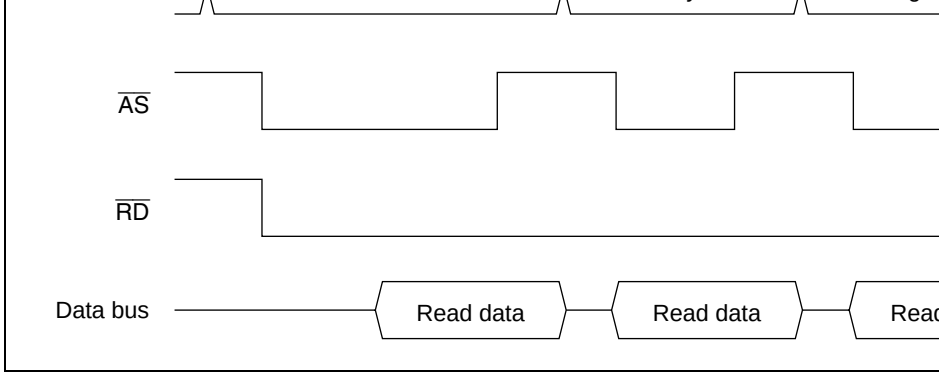


Figure 7.14 (a) Example of Burst ROM Access Timing (When AST0 = BRST

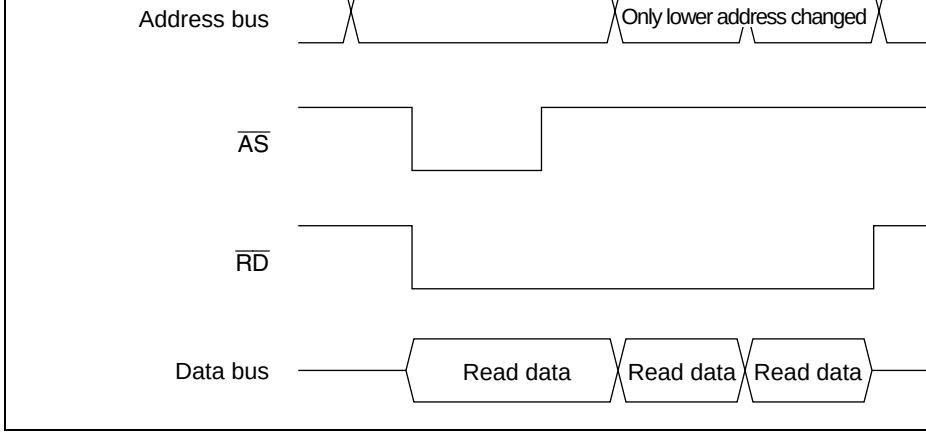


Figure 7.14 (b) Example of Burst ROM Access Timing (When AST0 = BRS

7.5.3 Wait Control

As with the basic bus interface, either program wait insertion or pin wait insertion usi
pin can be used in the initial cycle (full access) of the burst ROM interface. See sectio
Control.

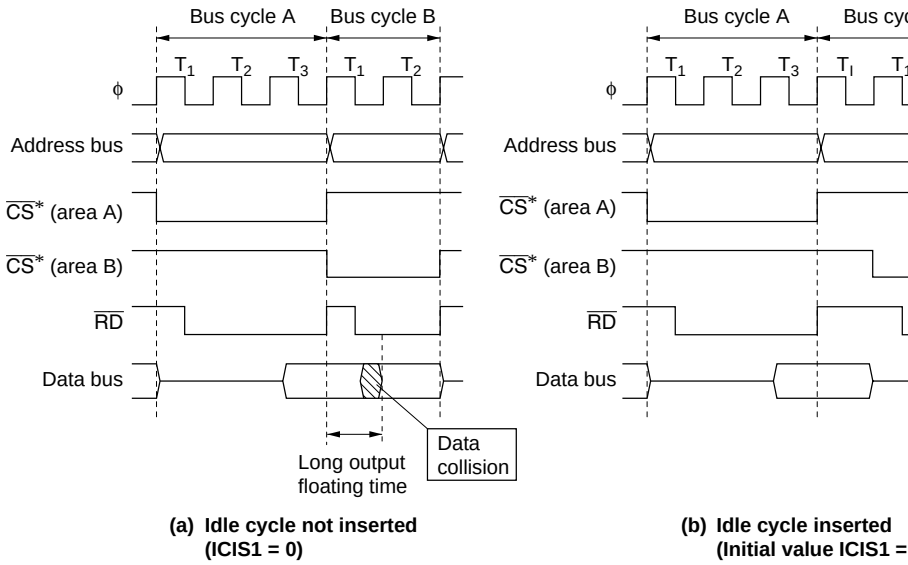
Wait states cannot be inserted in a burst cycle.

long output floating time, and high-speed memory, I/O interfaces, and so on.

(1) Consecutive Reads between Different Areas

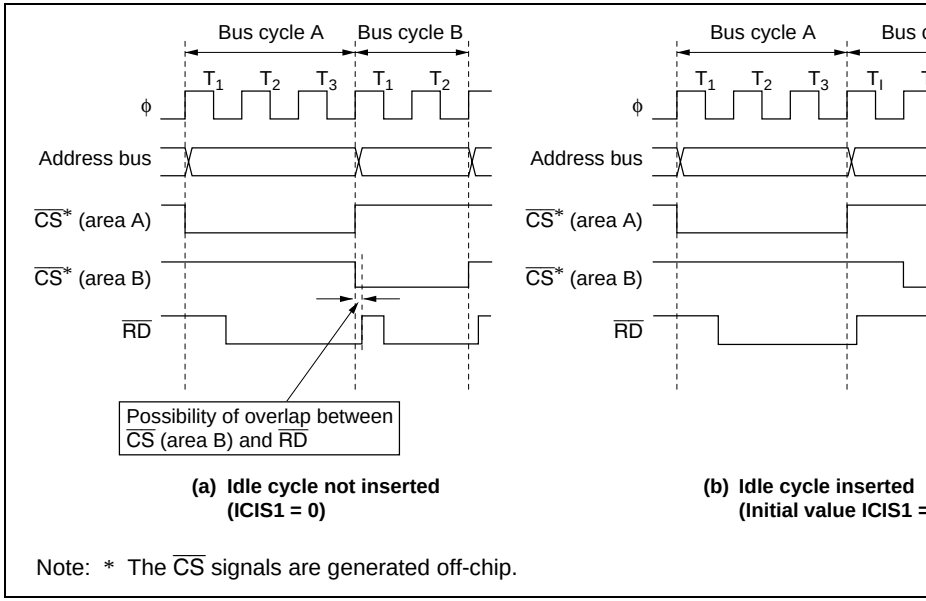
If consecutive reads between different areas occur while the ICIS1 bit in BCRH is set t
cycle is inserted at the start of the second read cycle.

Figure 7.15 shows an example of the operation in this case. In this example, bus cycle A
cycle from ROM with a long output floating time, and bus cycle B is a read cycle from
each being located in a different area. In (a), an idle cycle is not inserted, and a collisio
cycle B between the read data from ROM and that from SRAM. In (b), an idle cycle is
and a data collision is prevented.



Note: * The $\overline{CS}$ signals are generated off-chip.

Figure 7.15 Example of Idle Cycle Operation (1)



Note: * The $\overline{CS}$ signals are generated off-chip.

Figure 7.16 Example of Idle Cycle Operation (2)

$\overline{AS}$	High
$\overline{RD}$	High
$\overline{HWR}$	High
$\overline{LWR}$	High

next, only an external write is executed in the first state, but from the next state onwar
access (on-chip memory or internal I/O register read/write) is executed in parallel with
write rather than waiting until it ends.

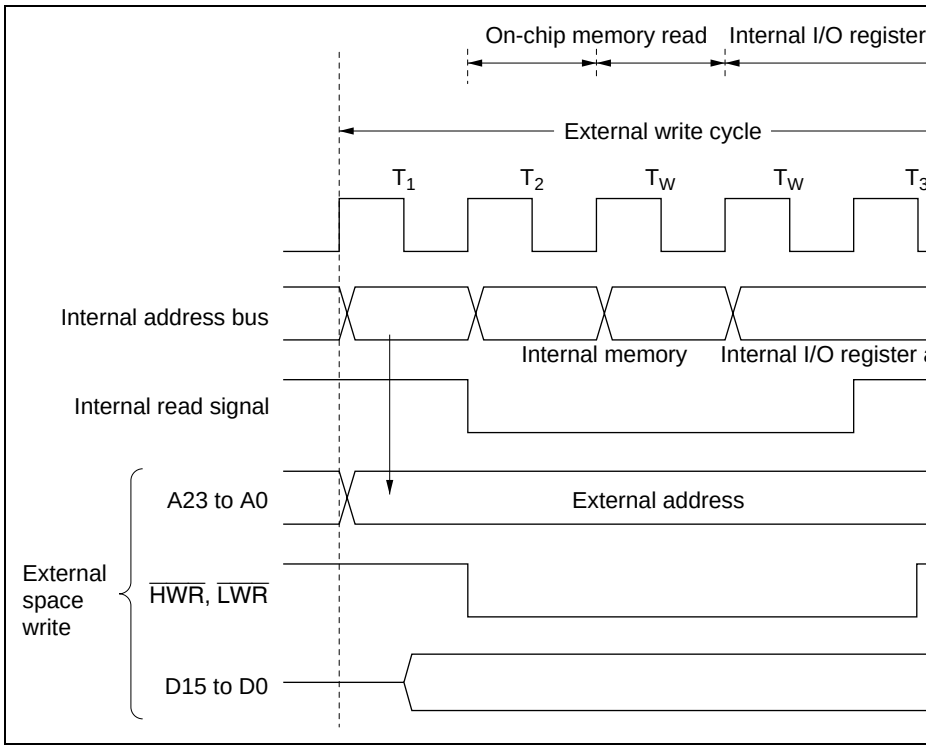


Figure 7.17 Example of Timing when Write Data Buffer Function is U

If an internal bus master wants to make an external access in the external bus released s
issue a bus request off-chip.

7.8.2 Operation

In external expansion mode, the bus can be released to an external device by setting the
in BCRL to 1. Driving the $\overline{\text{BREQ}}$ pin low issues an external bus request to the H8S/262
H8S/2623 Group. When the $\overline{\text{BREQ}}$ pin is sampled, at the prescribed timing the $\overline{\text{BACK}}$
driven low, and the address bus, data bus, and bus control signals are placed in the high
impedance state, establishing the external bus-released state.

In the external bus released state, an internal bus master can perform accesses using the
bus. When an internal bus master wants to make an external access, it temporarily defe
activation of the bus cycle, and waits for the bus request from the external bus master to
dropped.

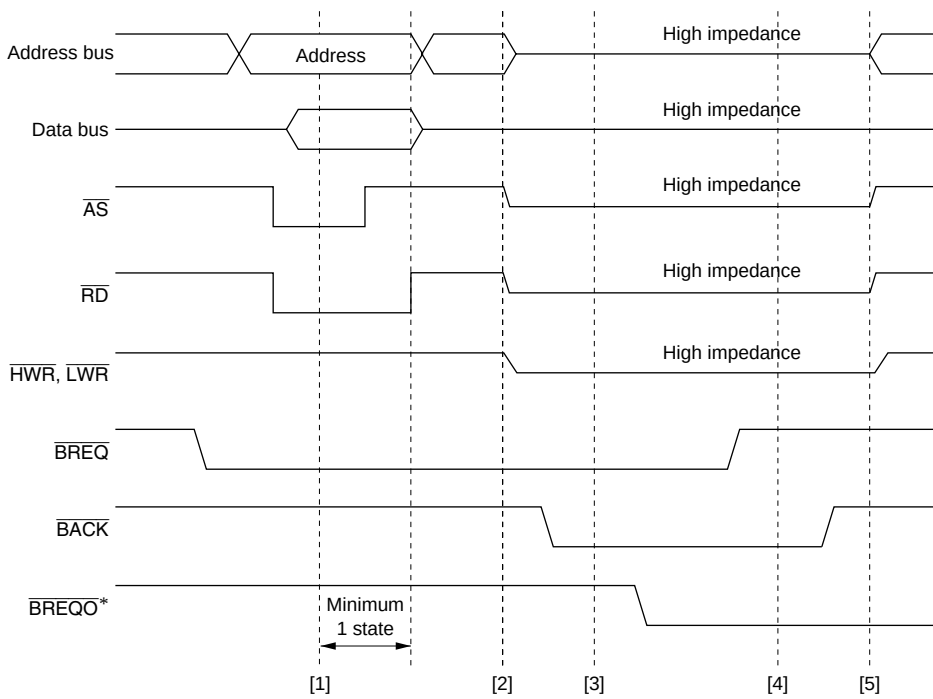
If the BREQOE bit in BCRL is set to 1, when an internal bus master wants to make an
access in the external bus released state, the $\overline{\text{BREQO}}$ pin is driven low and a request ca
off-chip to drop the bus request.

When the $\overline{\text{BREQ}}$ pin is driven high, the $\overline{\text{BACK}}$ pin is driven high at the prescribed timi
external bus released state is terminated.

If an external bus release request and internal bus master external access occur simultan
order of priority is as follows:

(High) External bus release > Internal bus master external access (Low)

$\overline{AS}$	High impedance
$\overline{RD}$	High impedance
$\overline{HWR}$	High impedance
$\overline{LWR}$	High impedance



[1] Low level of $\overline{\text{BREQ}}$ pin is sampled at rise of T_2 state.
[2] $\overline{\text{BACK}}$ pin is driven low at end of CPU read cycle, releasing bus to external bus master.
[3] $\overline{\text{BREQ}}$ pin state is still sampled in external bus released state.
[4] High level of $\overline{\text{BREQ}}$ pin is sampled.
[5] $\overline{\text{BACK}}$ pin is driven high, ending bus release cycle.
[6] $\overline{\text{BREQO}}$ signal goes high 1.5 clocks after $\overline{\text{BACK}}$ signal goes high.

Note: * Output only when BREQOE is set to 1.

Figure 7.18 Bus-Released State Transition Timing

7.9.1 Overview

The H8S/2626 Group and H8S/2623 Group have a bus arbiter that arbitrates bus mast operations.

There are two bus masters, the CPU and DTC, which perform read/write operations w have possession of the bus. Each bus master requests the bus by means of a bus reque bus arbiter determines priorities at the prescribed timing, and permits use of the bus by bus request acknowledge signal. The selected bus master then takes possession of the begins its operation.

7.9.2 Operation

The bus arbiter detects the bus masters' bus request signals, and if the bus is requested request acknowledge signal to the bus master making the request. If there are bus requ more than one bus master, the bus request acknowledge signal is sent to the one with t priority. When a bus master receives the bus request acknowledge signal, it takes poss bus until that signal is canceled.

The order of priority of the bus masters is as follows:

(High) DTC > CPU (Low)

An internal bus access by an internal bus master, and external bus release, can be exec parallel.

In the event of simultaneous external bus release request, and internal bus master exte request generation, the order of priority is as follows:

(High) External bus release > Internal bus master external access (Low)

- The bus is transferred at a break between bus cycles. However, if a bus cycle is exe discrete operations, as in the case of a longword-size access, the bus is not transferr the operations. See appendix A.5, Bus States During Instruction Execution, for timi which the bus is not transferred.
- If the CPU is in sleep mode, it transfers the bus immediately.

DTC: The DTC sends the bus arbiter a request for the bus when an activation request i

The DTC can release the bus after a vector read, a register information read (3 states), a transfer, or a register information write (3 states). It does not release the bus during a re information read (3 states), a single data transfer, or a register information write (3 state

7.10 Resets and the Bus Controller

In a reset, the H8S/2626 Group or H8S/2623 Group, including the bus controller, enters state at that point, and an executing bus cycle is discontinued.

8.1.1 Features

The features of the DTC are:

- Transfer possible over any number of channels
  - Transfer information is stored in memory
  - One activation source can trigger a number of data transfers (chain transfer)
- Wide range of transfer modes
  - Normal, repeat, and block transfer modes available
  - Incrementing, decrementing, and fixing of source and destination addresses ca
- Direct specification of 16-Mbyte address space possible
  - 24-bit transfer source and destination addresses can be specified
- Transfer can be set in byte or word units
- A CPU interrupt can be requested for the interrupt that activated the DTC
  - An interrupt request can be issued to the CPU after one data transfer ends
  - An interrupt request can be issued to the CPU after the specified data transfers completely ended
- Activation by software is possible
- Module stop mode can be set
  - The initial setting enables DTC registers to be accessed. DTC operation is halt module stop mode.

Note: When the DTC is used, the RAME bit in SYSCR must be set to 1.

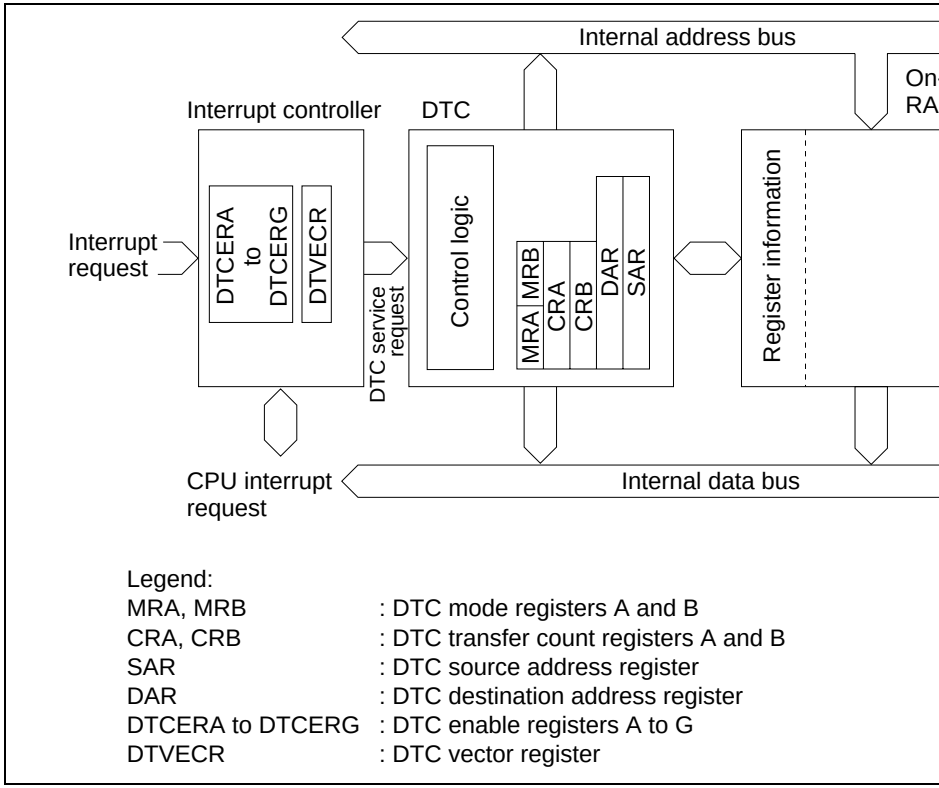


Figure 8.1 Block Diagram of DTC

DTC source address register	SAR	—*2	Undefined	—*3
DTC destination address register	DAR	—*2	Undefined	—*3
DTC transfer count register A	CRA	—*2	Undefined	—*3
DTC transfer count register B	CRB	—*2	Undefined	—*3
DTC enable registers	DTCER	R/W	H'00	H'FE1
DTC vector register	DTVECR	R/W	H'00	H'FE1
Module stop control register	MSTPCRA	R/W	H'3F	H'FDE

Notes: 1. Lower 16 bits of the address.

2. Registers within the DTC cannot be read or written to directly.

3. Register information is located in on-chip RAM addresses H'EBC0 to H'EF be located in external memory space. When the DTC is used, the RAME bi must be set to 1.

R/W	:	—	—	—	—	—	—	—

MRA is an 8-bit register that controls the DTC operating mode.

Bits 7 and 6—Source Address Mode 1 and 0 (SM1, SM0): These bits specify whethe to be incremented, decremented, or left fixed after a data transfer.

Bit 7	Bit 6	
SM1	**SM0**	**Description**
0	—	SAR is fixed
1	0	SAR is incremented after a transfer (by +1 when Sz = 0; by +2 when Sz = 1)
	1	SAR is decremented after a transfer (by –1 when Sz = 0; by –2 when Sz = 1)

Bits 5 and 4—Destination Address Mode 1 and 0 (DM1, DM0): These bits specify v DAR is to be incremented, decremented, or left fixed after a data transfer.

Bit 5	Bit 4	
DM1	**DM0**	**Description**
0	—	DAR is fixed
1	0	DAR is incremented after a transfer (by +1 when Sz = 0; by +2 when Sz = 1)
	1	DAR is decremented after a transfer (by –1 when Sz = 0; by –2 when Sz = 1)

Bit 1—DTC Transfer Mode Select (DTS): Specifies whether the source side or the d
side is set to be a repeat area or block area, in repeat mode or block transfer mode.

Bit 1

DTS	Description
0	Destination side is repeat area or block area
1	Source side is repeat area or block area

Bit 0—DTC Data Transfer Size (Sz): Specifies the size of data to be transferred.

Bit 0

Sz	Description
0	Byte-size transfer
1	Word-size transfer

MRB is an 8-bit register that controls the DTC operating mode.

Bit 7—DTC Chain Transfer Enable (CHNE): Specifies chain transfer. With chain tr
number of data transfers can be performed consecutively in response to a single transfe

In data transfer with CHNE set to 1, determination of the end of the specified number o
clearing of the interrupt source flag, and clearing of DTCER is not performed.

Bit 7

CHNE	Description
0	End of DTC data transfer (activation waiting state is entered)
1	DTC chain transfer (new register information is read, then data is transferre

Bit 6—DTC Interrupt Select (DISEL): Specifies whether interrupt requests to the CP
disabled or enabled after a data transfer.

Bit 6

DISEL	Description
0	After a data transfer ends, the CPU interrupt is disabled unless the transfer 0 (the DTC clears the interrupt source flag of the activating interrupt to 0)
1	After a data transfer ends, the CPU interrupt is enabled (the DTC does not interrupt source flag of the activating interrupt to 0)

Bits 5 to 0—Reserved: These bits have no effect on DTC operation in the H8S/2626 G
H8S/2623 Group, and should always be written with 0.

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

8.2.4 DTC Destination Address Register (DAR)

Bit	:	23	22	21	20	19	- - -	4	3	2
							- - -			
Initial value	:	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	- - -	Unde-fined	Unde-fined	Und fin
R/W	:	—	—	—	—	—	- - -	—	—	–

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

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

In normal mode, the entire CRA functions as a 16-bit transfer counter (1 to 65,536). It i
decremented by 1 every time data is transferred, and transfer ends when the count reach

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

8.2.6 DTC Transfer Count Register B (CRB)

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
Initial value:		Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined
R/W	:	—	—	—	—	—	—	—	—	—	—	—	—	—	—

CRB is a 16-bit register that designates the number of times data is to be transferred by
block transfer mode. It functions as a 16-bit transfer counter (1 to 65,536) that is decrem
every time data is transferred, and transfer ends when the count reaches H'0000.

DTCERG, with bits corresponding to the interrupt sources that can control enabling a
of DTC activation. These bits enable or disable DTC service for the corresponding int
sources.

The DTC enable registers are initialized to H'00 by a reset and in hardware standby m

Bit n—DTC Activation Enable (DTCEn)

Bit n	
DTCEn	**Description**
0	DTC activation by this interrupt is disabled [Clearing conditions] • When the DISEL bit is 1 and the data transfer has ended • When the specified number of transfers have ended
1	DTC activation by this interrupt is enabled [Holding condition] When the DISEL bit is 0 and the specified number of transfers have not e

A DTCE bit can be set for each interrupt source that can activate the DTC. The corres
between interrupt sources and DTCE bits is shown in table 8.4, together with the vect
generated for each interrupt controller.

For DTCE bit setting, use bit manipulation instructions such as BSET and BCLR for
writing. If all interrupts are masked, multiple activation sources can be set at one time
data after executing a dummy read on the relevant register.

Notes: 1. Only 1 can be written to the SWDTE bit.
2. Bits DTVEC6 to DTVEC0 can be written to when SWDTE = 0.

DTVECR is an 8-bit readable/writable register that enables or disables DTC activation software, and sets a vector number for the software activation interrupt.

DTVECR is initialized to H'00 by a reset and in hardware standby mode.

Bit 7—DTC Software Activation Enable (SWDTE): Enables or disables DTC activa software.

Bit 7

SWDTE	Description
0	DTC software activation is disabled ([Clearing conditions] • When the DISEL bit is 0 and the specified number of transfers have not • When 0 s written to the DISEL bit after a software-activated data transfe interrupt (SWDTEND) request has been sent to the CPU
1	DTC software activation is enabled [Holding conditions] • When the DISEL bit is 1 and data transfer has ended • When the specified number of transfers have ended • During data transfer due to software activation

Bits 6 to 0—DTC Software Activation Vectors 6 to 0 (DTVEC6 to DTVEC0): Thes specify a vector number for DTC software activation.

The vector address is expressed as H'0400 + ((vector number) << 1). <<1 indicates a or shift. For example, when DTVEC6 to DTVEC0 = H'10, the vector address is H'0420.

When the MSTPA6 bit in MSTPCRA is set to 1, the DTC operation stops at the end o
cycle and a transition is made to module stop mode. However, 1 cannot be written in t
bit while the DTC is operating. For details, see sections 21A.5, 21B.5, Module Stop M

MSTPCRA is initialized to H'3F by a reset and in hardware standby mode. It is not in
software standby mode.

Bit 6—Module Stop (MSTPA6): Specifies the DTC module stop mode.

Bit 6

MSTPA6	Description
0	DTC module stop mode cleared
1	DTC module stop mode set

8.3 Operation

8.3.1 Overview

When activated, the DTC reads register information that is already stored in memory a
data on the basis of that register information. After the data transfer, it writes updated
information back to memory. Pre-storage of register information in memory makes it
transfer data over any required number of channels. Setting the CHNE bit to 1 makes
perform a number of transfers with a single activation.

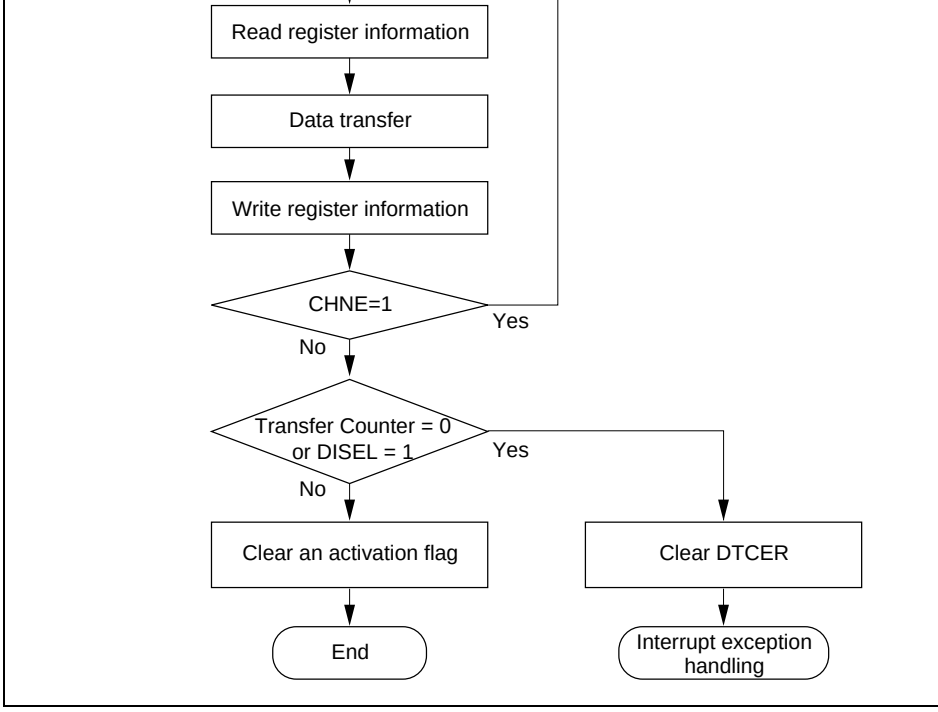


Figure 8.2 Flowchart of DTC Operation

The DTC transfer mode can be normal mode, repeat mode, or block transfer mode.

The 24-bit SAR designates the DTC transfer source address and the 24-bit DAR design transfer destination address. After each transfer, SAR and DAR are independently incre decremented, or left fixed.

Table 8.2 outlines the functions of the DTC.

or decremented by 1 or 2
  - Up to 65,536 transfers possible
- Repeat mode
  - One transfer request transfers one byte or one word
  - Memory addresses are incremented or decremented by 1 or 2
  - After the specified number of transfers (1 to 256), the initial state resumes and operation continues
- Block transfer mode
  - One transfer request transfers a block of the specified size
  - Block size is from 1 to 256 bytes or words
  - Up to 65,536 transfers possible
  - A block area can be designated at either the source or destination

- A/D converter ADI
- Software
- HCAN RM0

8.3.2 Activation Sources

The DTC operates when activated by an interrupt or by a write to DTVECR by softwa
interrupt request can be directed to the CPU or DTC, as designated by the correspondi
bit. An interrupt becomes a DTC activation source when the corresponding bit is set t
CPU interrupt source when the bit is cleared to 0.

At the end of a data transfer (or the last consecutive transfer in the case of chain transf
activation source or corresponding DTCER bit is cleared. Table 8.3 shows activation s
DTCER clearance. The activation source flag, in the case of RXI0, for example, is the
of SCI0.

The activation source flag is cleared to 0	The activation source flag rema A request is issued to the CPU activation source interrupt

Figure 8.3 shows a block diagram of activation source control. For details see section 5 Controller.

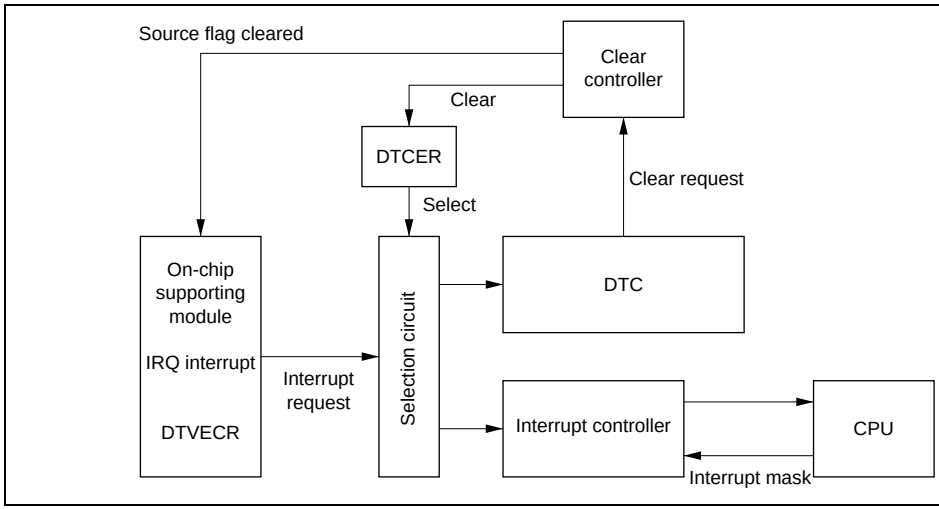


Figure 8.3 Block Diagram of DTC Activation Source Control

When an interrupt has been designated a DTC activation source, existing CPU mask lev interrupt controller priorities have no effect. If there is more than one activation source time, the DTC operates in accordance with the default priorities.

The DTC reads the start address of the register information from the vector address se
activation source, and then reads the register information from that start address. The r
information can be placed at predetermined addresses in the on-chip RAM. The start a
the register information should be an integral multiple of four.

The configuration of the vector address is the same in both normal* and advanced mo
unit being used in both cases. These two bytes specify the lower bits of the address in
RAM.

Note: * Not available in the H8S/2626 Group or H8S/2623 Group.

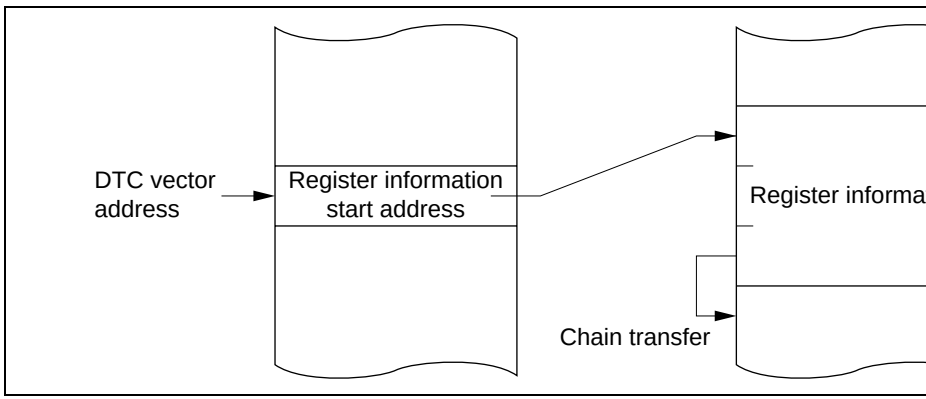


Figure 8.4 Correspondence between DTC Vector Address and Register Info

IRQ1		17	H'0422	DTCEA6
IRQ2		18	H'0424	DTCEA5
IRQ3		19	H'0426	DTCEA4
IRQ4		20	H'0428	DTCEA3
IRQ5		21	H'042A	DTCEA2
Reserved		22	H'042C	DTCEA1
		23	H'042E	DTCEA0
ADI (A/D conversion end)	A/D	28	H'0438	DTCEB6
TGI0A (GR0A compare match/ input capture)	TPU channel 0	32	H'0440	DTCEB5
TGI0B (GR0B compare match/ input capture)		33	H'0442	DTCEB4
TGI0C (GR0C compare match/ input capture)		34	H'0444	DTCEB3
TGI0D (GR0D compare match/ input capture)		35	H'0446	DTCEB2
TGI1A (GR1A compare match/ input capture)	TPU channel 1	40	H'0450	DTCEB1
TGI1B (GR1B compare match/ input capture)		41	H'0452	DTCEB0
TGI2A (GR2A compare match/ input capture)	TPU channel 2	44	H'0458	DTCEC7
TGI2B (GR2B compare match/ input capture)		45	H'045A	DTCEC6

TGI3D (GR3D compare match/ input capture)		51	H'0466	DTCEC2
TGI4A (GR4A compare match/ input capture)	TPU channel 4	56	H'0470	DTCEC1
TGI4B (GR4B compare match/ input capture)		57	H'0472	DTCEC0
TGI5A (GR5A compare match/ input capture)	TPU channel 5	60	H'0478	DTCED5
TGI5B (GR5B compare match/ input capture)		61	H'047A	DTCED4
Reserved	—	64	H'0480	DTCED3
		65	H'0482	DTCED2
		68	H'0488	DTCED1
		69	H'048A	DTCED0
		72	H'0490	DTCEE7
		73	H'0492	DTCEE6
		74	H'0494	DTCEE5
		75	H'0496	DTCEE4
RXI0 (reception complete 0)	SCI channel 0	81	H'04A2	DTCEE3
TXI0 (transmit data empty 0)		82	H'04A4	DTCEE2
RXI1 (reception complete 1)	SCI channel 1	85	H'04AA	DTCEE1
TXI1 (transmit data empty 1)		86	H'04AC	DTCEE0
RXI2 (reception complete 2)	SCI channel 2	89	H'04B2	DTCEF7
TXI2 (transmit data empty 2)		90	H'04B4	DTCEF6
RM0	HCAN	106	H'04D4	DTCEG5

Note: * DTCE bits with no corresponding interrupt are reserved, and should be writ

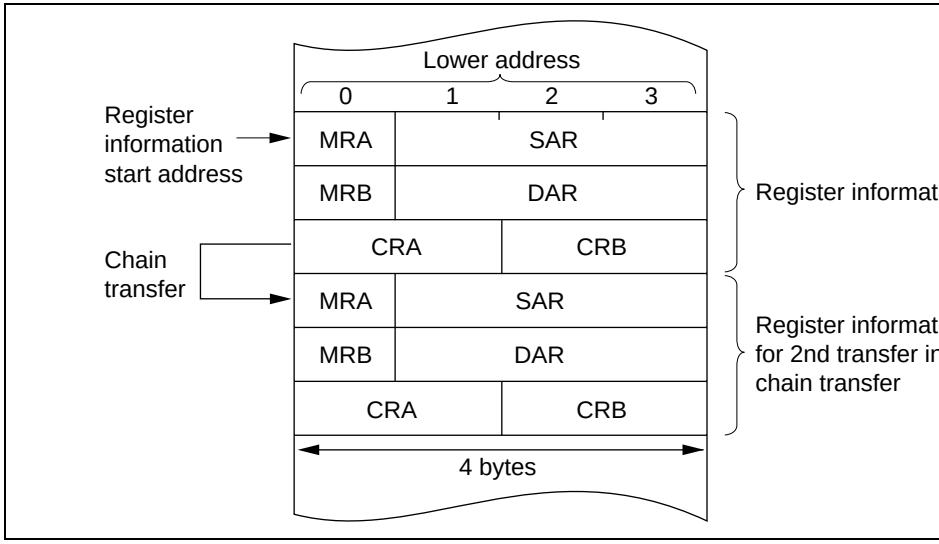


Figure 8.5 Location of Register Information in Address Space

Table 8.5 Register Information in Normal Mode

Name	Abbreviation	Function
DTC source address register	SAR	Designates source addre
DTC destination address register	DAR	Designates destination a
DTC transfer count register A	CRA	Designates transfer coun
DTC transfer count register B	CRB	Not used

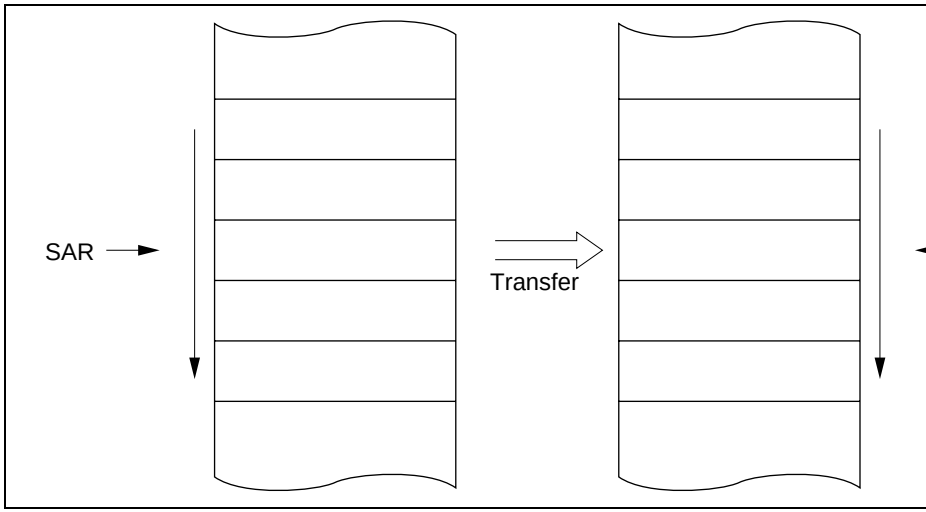


Figure 8.6 Memory Mapping in Normal Mode

Table 8.6 lists the register information in repeat mode and figure 8.7 shows memory ma
repeat mode.

Table 8.6 Register Information in Repeat Mode

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

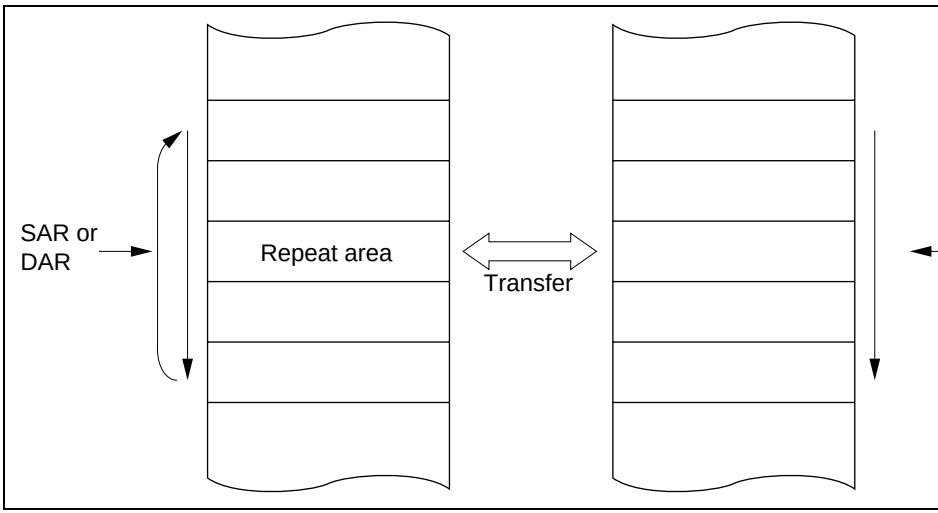


Figure 8.7 Memory Mapping in Repeat Mode

From 1 to 65,536 transfers can be specified. Once the specified number of transfers ha
CPU interrupt is requested.

Table 8.7 lists the register information in block transfer mode and figure 8.8 shows me
mapping in block transfer mode.

Table 8.7 Register Information in Block Transfer Mode

Name	Abbreviation	Function
DTC source address register	SAR	Designates source addre
DTC destination address register	DAR	Designates destination a
DTC transfer count register AH	CRAH	Holds block size
DTC transfer count register AL	CRAL	Designates block size co
DTC transfer count register B	CRB	Transfer count

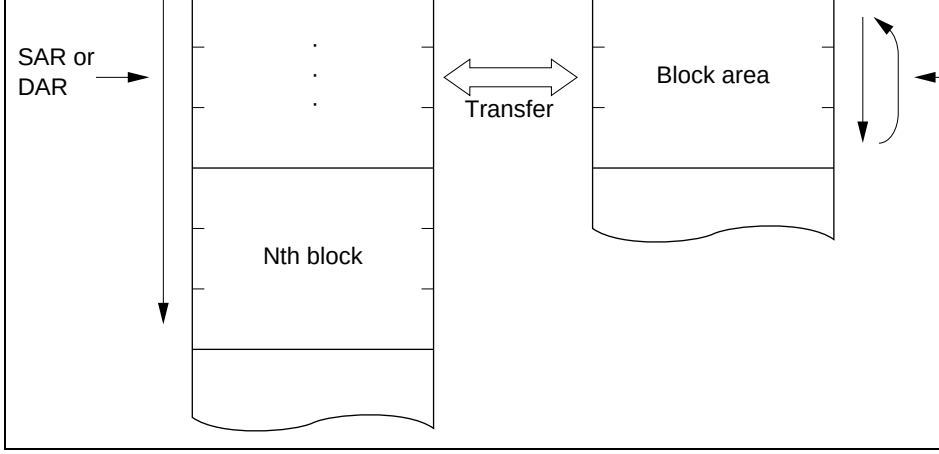


Figure 8.8 Memory Mapping in Block Transfer Mode

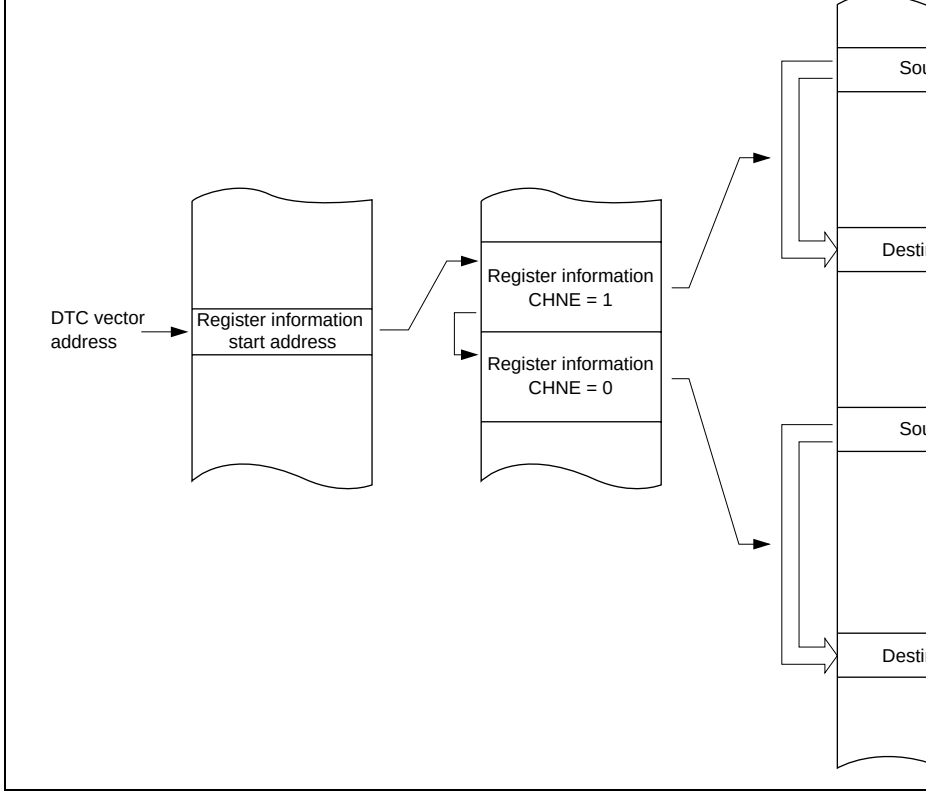


Figure 8.9 Chain Transfer Memory Map

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

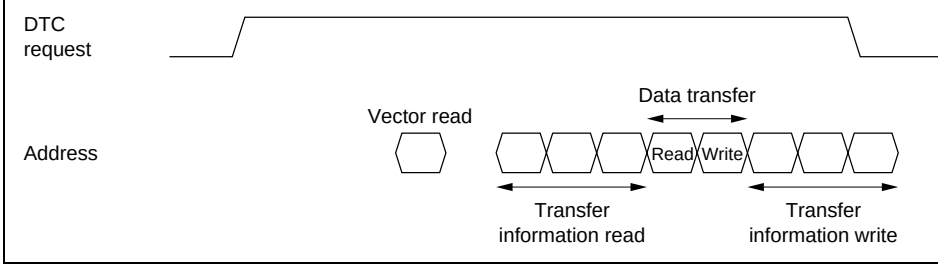


Figure 8.10 DTC Operation Timing (Example in Normal Mode or Repeat M

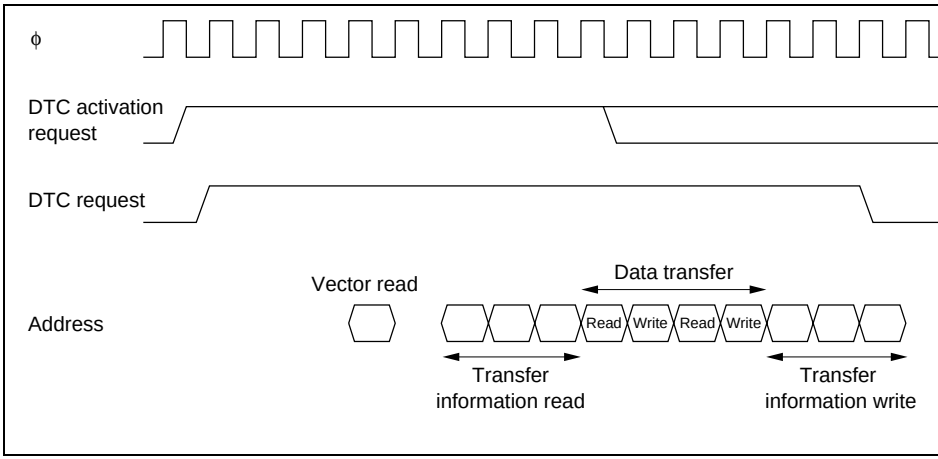


Figure 8.11 DTC Operation Timing (Example of Block Transfer Mode with Block Size of 2)

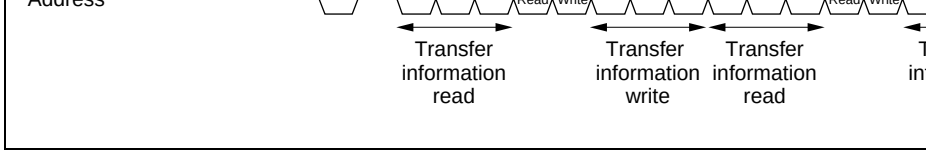


Figure 8.12 DTC Operation Timing (Example of Chain Transfer)

8.3.10 Number of DTC Execution States

Table 8.8 lists execution statuses for a single DTC data transfer, and table 8.9 shows t states required for each execution status.

Table 8.8 DTC Execution Statuses

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

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

	Register information read/write	S_J	1	—	—	—	—	—	
	Byte data read	S_K	1	1	2	2	2	3+m	
	Word data read	S_K	1	1	4	2	4	6+2m	
	Byte data write	S_L	1	1	2	2	2	3+m	
	Word data write	S_L	1	1	4	2	4	6+2m	
	Internal operation	S_M	1						

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

$$\text{Number of execution states} = I \cdot (S_I + 1) + \Sigma (J \cdot S_J + K \cdot S_K + L \cdot S_L) + M \cdot S_M$$

For example, when the DTC vector address table is located in on-chip ROM, normal m
and data is transferred from the on-chip ROM to an internal I/O register, the time requi
DTC operation is 14 states. The time from activation to the end of the data write is 11 s

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

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

Activation by Software: The procedure for using the DTC with software activation is

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

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

[3] Check that the SWDTE bit is 0.

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

[5] Check the vector number written to DTVECR.

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

any value. Set MRB for one data transfer by one interrupt (CHNE = 0, DISEL = 0).
SCI RDR address in SAR, the start address of the RAM area where the data will be
DAR, and 128 (H'0080) in CRA. CRB can be set to any value.

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

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

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

[5] Each time reception of one byte of data ends on the SCI, the RDRF flag in SSR is s
RXI interrupt is generated, and the DTC is activated. The receive data is transferred
to RAM by the DTC. DAR is incremented and CRA is decremented. The RDRF fla
automatically cleared to 0.

[6] When CRA becomes 0 after the 128 data transfers have ended, the RDRF flag is he
DTCE bit is cleared to 0, and an RXI interrupt request is sent to the CPU. The inter
handling routine should perform wrap-up processing.

[1] Perform settings for transfer to the PPG's NDR. Set MRA to source address incr
(SM1 = 1, SM0 = 0), fixed destination address (DM1 = DM0 = 0), repeat mode (
MD0 = 1), and word size (Sz = 1). Set the source side as a repeat area (DTS = 1)
chain mode (CHNE = 1, DISEL = 0). Set the data table start address in SAR, the
address in DAR, and the data table size in CRAH and CRAL. CRB can be set to

[2] Perform settings for transfer to the TPU's TGR. Set MRA to source address incr
(SM1 = 1, SM0 = 0), fixed destination address (DM1 = DM0 = 0), normal mode
MD0 = 0), and word size (Sz = 1). Set the data table start address in SAR, the TC
in DAR, and the data table size in CRA. CRB can be set to any value.

[3] Locate the TPU transfer register information consecutively after the NDR transfe
information.

[4] Set the start address of the NDR transfer register information to the DTC vector a

[5] Set the bit corresponding to TGIA in DTCER to 1.

[6] Set TGRA as an output compare register (output disabled) with TIOR, and enabl
interrupt with TIER.

[7] Set the initial output value in PODR, and the next output value in NDR. Set bits i
NDER for which output is to be performed to 1. Using PCR, select the TPU com
to be used as the output trigger.

[8] Set the CST bit in TSTR to 1, and start the TCNT count operation.

[9] Each time a TGRA compare match occurs, the next output value is transferred to
the set value of the next output trigger period is transferred to TGRA. The activat
TGFA flag is cleared.

[10] When the specified number of transfers are completed (the TPU transfer CRA va
TGFA flag is held at 1, the DTCE bit is cleared to 0, and a TGIA interrupt reques
the CPU. Termination processing should be performed in the interrupt handling r

0). Set the transfer source address (H'1000) in SAR, the destination address (H'2000
and 128 (H'8080) in CRA. Set 1 (H'0001) in CRB.

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

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

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

[5] Read DTVECR again and check that it is set to the vector number (H'60). If it is no
indicates that the write failed. This is presumably because an interrupt occurred betv
3 and 4 and led to a different software activation. To activate this transfer, go back t

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

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

When the DISEL bit is 1 and one data transfer has ended, or the specified number of t
have ended, after data transfer ends, the SWDTE bit is held at 1 and an SWDTEND in
generated. The interrupt handling routine should clear the SWDTE bit to 0.

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

8.5 Usage Notes

Module Stop: When the MSTPA6 bit in MSTPCRA is set to 1, the DTC clock stops,
DTC enters the module stop state. However, 1 cannot be written in the MSTPA6 bit w
DTC is operating.

On-Chip RAM: The MRA, MRB, SAR, DAR, CRA, and CRB registers are all locate
RAM. When the DTC is used, the RAME bit in SYSCR must not be cleared to 0.

DTCE Bit Setting: For DTCE bit setting, use bit manipulation instructions such as B
BCLR. If all interrupts are masked, multiple activation sources can be set at one time
data after executing a dummy read on the relevant register.

Table 9.1 summarizes the port functions. The pins of each port also have other functio

Each I/O port includes a data direction register (DDR) that controls input/output, a dat
(DR) that stores output data, and a port register (PORT) used to read the pin states. Th
ports do not have a DR or DDR register.

Ports A to E have a built-in pull-up MOS function, and in addition to DR and DDR, h
input pull-up control register (PCR) to control the on/off state of MOS input pull-up.

Ports A to C include an open-drain control register (ODR) that controls the on/off stat
output buffer PMOS.

Ports 10 to 13, A0 to A3, and B to E can drive a single TTL load and 50 pF capacitive
used as expansion bus control signal output pins. In other cases these ports, together w
to 17 and 3, can drive a single TTL load and 30 pF capacitive load. All the I/O ports c
Darlington transistor when in output mode. Ports 1, A, B, and C can drive an LED (10
current).

See appendix C, I/O Port Block Diagrams, for a block diagram of each port.

		P12/PO10/TIOCC0/ TCLKA/A22 P11/PO9/TIOCB0/A21 P10/PO8/TIOCA0/A20	T T T T T T T T P pi P in pi IR
Port 4	• 8-bit input port	P47/AN7 P46/AN6 P45/AN5 P44/AN4 P43/AN3 P42/AN2 P41/AN1 P40/AN0	8-bit input port also functioning as A/D converte inputs (AN7 to AN0)
Port 9	• 8-bit input port	P97/AN15/DA3*1 P96/AN14/DA2*1 P95/AN13 P94/AN12 P93/AN11 P92/AN10 P91/AN9 P90/AN8	8-bit input port also functioning as A/D converte inputs (AN15 to AN8) and D/A converter analog (DA3, DA2)

Port B	• 8-bit I/O port • Built-in MOS input pull-up • Open-drain output capability	PB7/A15/TIOCB5 PB6/A14/TIOCA5 PB5/A13/TIOCB4 PB4/A12/TIOCA4 PB3/A11/TIOCD3 PB2/A10/TIOCC3 PB1/A9/TIOCB3 PB0/A8/TIOCA3	8-bit I/O port also functioning as HTPU I/O pins (TIOCB5, TIOCA5, TIOCB4, TIOCA4, TIOCD3, TIOCC3, TIOCB3, TIOCA3) and address outputs (A15 to A8)	
Port C	• 8-bit I/O port • Built-in MOS input pull-up • Open-drain output capability	PC7/A7 PC6/A6 PC5/A5/SCK1/$\overline{\text{IRQ5}}$ PC4/A4/RxD1 PC3/A3/TxD1 PC2/A2/SCK0/$\overline{\text{IRQ4}}$ PC1/A1/RxD0 PC0/A0/TxD0	Address output (A7 to A0)	8-bit I/O port also functioning as SCI (channel 0, 1) I/O pins (TxD0, RxD0, SCK0, TxD1, RxD1, SCK1), interrupt input pins ($\overline{\text{IRQ4}}$, $\overline{\text{IRQ5}}$), and address outputs (A7 to A0)
Port D	• 8-bit I/O port • Built-in MOS input pull-up	PD7/D15 PD6/D14 PD5/D13 PD4/D12 PD3/D11 PD2/D10 PD1/D9 PD0/D8	Data bus input/output	

		PE0/D0		
Port F	• 8-bit I/O port	PF7/φ	When DDR = 0: input port When DDR = 1 (after reset): φ output	W 0 in W 1
		PF6/$\overline{AS}$ PF5/$\overline{RD}$ PF4/$\overline{HWR}$ PF3/$\overline{LWR}$/$\overline{ADTRG}$/$\overline{IRQ3}$	$\overline{AS}$, $\overline{RD}$, $\overline{HWR}$, $\overline{LWR}$ output $\overline{ADTRG}$, $\overline{IRQ3}$ input	I/ $\overline{A}$ $\overline{IR}$
		PF2/$\overline{WAIT}$/$\overline{BREQO}$	When WAITE = 0 and BREQOE = 0 (after reset): I/O port When WAITE = 1 and BREQOE = 0: $\overline{WAIT}$ input When WAITE = 0 and BREQOE = 1: $\overline{BREQO}$ input	I/
		PF1/$\overline{BACK}$/BUZZ*3 PF0/$\overline{BREQ}$/$\overline{IRQ2}$	When BRLE = 0 (after reset): I/O port When BRLE = 1: $\overline{BREQ}$ input, $\overline{BACK}$ output, BUZZ output, $\overline{IRQ2}$ input	I/ B $\overline{IR}$

Notes: 1. DA3 and DA2 are outputs in the H8S/2626 Group only.
2. In the H8S/2626 Group, PA5 and PA4 are OSC2 and OSC1, respectively.
3. BUZZ output pin in the H8S/2626 Group only.

Figure 9.1 shows the port 1 pin configuration.

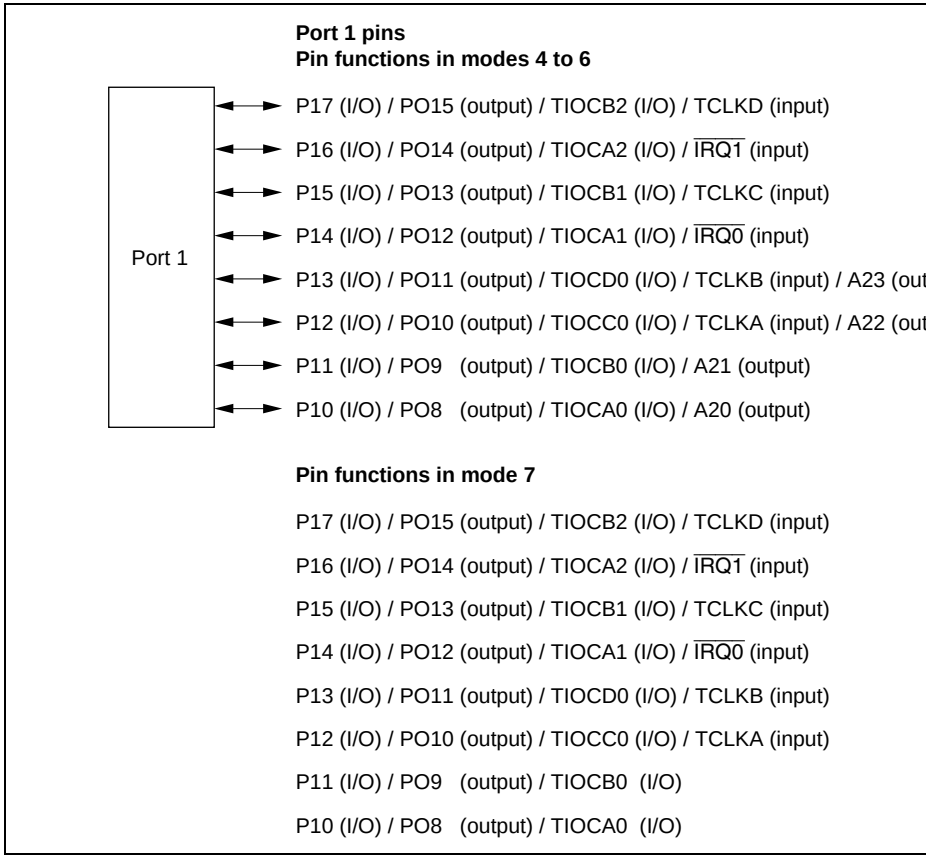


Figure 9.1 Port 1 Pin Functions

Port 1 register	PORT1	R	Undefined	H'F

Note: * Lower 16 bits of the address.

Port 1 Data Direction Register (P1DDR)

Bit	:	7	6	5	4	3	2	1
		P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR
Initial value	:	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W

P1DDR is an 8-bit write-only register, the individual bits of which specify input or outp pins of port 1. P1DDR cannot be read; if it is, an undefined value will be read.

Setting a P1DDR bit to 1 makes the corresponding port 1 pin an output pin, while clear to 0 makes the pin an input pin.

P1DDR is initialized to H'00 by a reset, and in hardware standby mode. It retains its pri software standby mode.

P1DR is initialized to H'00 by a reset, and in hardware standby mode. It retains its pri
software standby mode.

Port 1 Register (PORT1)

Bit	:	7	6	5	4	3	2	1
		P17	P16	P15	P14	P13	P12	P11
Initial value	:	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R

Note: * Determined by state of pins P17 to P10.

PORT1 is an 8-bit read-only register that shows the pin states. It cannot be written to.
output data for the port 1 pins (P17 to P10) must always be performed on P1DR.

If a port 1 read is performed while P1DDR bits are set to 1, the P1DR values are read.
read is performed while P1DDR bits are cleared to 0, the pin states are read.

After a reset and in hardware standby mode, PORT1 contents are determined by the p
P1DDR and P1DR are initialized. PORT1 retains its prior state in software standby m

Pin	Selection Method and Pin Functions
P17/PO15/ TIOCB2/TCLKD	The pin function is switched as shown below according to the comb the TPU channel 2 setting (by bits MD3 to MD0 in TMDR2, bits IOB in TIOR2, and bits CCLR1 and CCLR0 in TCR2), bits TPSC2 to TP TCR0 and TCR5, bit NDER15 in NDERH, and bit P17DDR.

TPU Channel 2 Setting	Table Below (1)	Table Below	
P17DDR	—	0	1
NDER15	—	—	0
Pin function	TIOCB2 output	P17 input	P17 output
		TIOCB2 inpu	
	TCLKD input *2		

Notes: 1. TIOCB2 input when MD3 to MD0 = B'0000 or B'01xx, ar 1.

2. TCLKD input when the setting for either TCR0 or TCR5 to TPSC0 = B'111.
   TCLKD input when channels 2 and 4 are set to phase co mode.

TPU Channel 2 Setting	(2)	(1)	(2)	(2)	(1)
MD3 to MD0	B'0000, B'01xx		B'0010		B'0011
IOB3 to IOB0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	—	B'xx00	Other th
CCLR1, CCLR0	—	—	—	—	Other than B'10
Output function	—	Output compare output	—	—	PWM mode 2 output

x

NDER14	—	—	0
Pin function	TIOCA2 output	P16 input	P16 output
		TIOCA2 inp	
	$\overline{\text{IRQ1}}$ input		

TPU Channel 2 Setting	(2)	(1)	(2)	(1)	(1)
MD3 to MD0	B'0000, B'01xx		B'001x	B'0010	B
IOA3 to IOA0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B	
CCLR1, CCLR0	—	—	—	—	Other than B'0
Output function	—	Output compare output	—	PWM mode 1 output *2	PWM mode output

Notes: 1. TIOCA2 input when MD3 to MD0 = B'0000 or B'01xx, a
1.
2. TIOCB2 output is disabled.

NDER13	—	—	0
Pin function	TIOCB1 output	P15 input	P15 output
		TIOCB1 inpu	
	TCLKC input *2		

Notes: 1. TIOCB1 input when MD3 to MD0 = B'0000 or B'01xx, ar
to IOB0 = B'10xx.

2. TCLKC input when the setting for either TCR0 or TCR2
to TPSC0 = B'110; or when the setting for either TCR4 c
TPSC2 to TPSC0 = B'101.
TCLKC input when channels 2 and 4 are set to phase co
mode.

TPU Channel 1 Setting	(2)	(1)	(2)	(2)	(1)
MD3 to MD0	B'0000, B'01xx		B'0010		B'0011
IOB3 to IOB0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	—	B'xx00	Other th
CCLR1, CCLR0	—	—	—	—	Other than B'10
Output function	—	Output compare output	—	—	PWM mode 2 output

x

NDER12	—	—	0
Pin function	TIOCA1 output	P14 input	P14 output
		TIOCA1 inp	
	$\overline{\text{IRQ0}}$ input		

TPU Channel 1 Setting	(2)	(1)	(2)	(1)	(1)
MD3 to MD0	B'0000, B'01xx		B'001x	B'0010	B
IOA3 to IOA0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'xx00	Other t
CCLR1, CCLR0	—	—	—	—	Other than B'0
Output function	—	Output compare output	—	PWM mode 1 output*2	PWM mode 1 output

Notes: 1. TIOCA1 input when MD3 to MD0 = B'0000 or B'01xx, a IOA0 = B'10xx.

2. TIOCB1 output is disabled.

TPU Channel 0 Setting	Table Below (1)	Table Below (2)		
P13DDR	—	0	1	1
NDER11	—	—	0	1
Pin function	TIOCD0 output	P13 input	P13 output	PO11 output
		TIOCD0 input *1		
	TCLKB input *2			

Operating mode	Mode 7			
AE3 to AE0	—			
TPU Channel 0 Setting	Table Below (1)	Table Below (2)		
P13DDR	—	0	1	
NDER11	—	—	0	
Pin function	TIOCD0 output	P13 input	P13 output	PC
		TIOCD0 input *1		
	TCLKB input *2			

Notes: 1. TIOCD0 input when MD3 to MD0 = B'0000, and IOD3 to
B'10xx.

2. TCLKB input when the setting for TCR0 to TCR2 is: TPS
TPSC0 = B'101.

TCLKB input when channels 1 and 5 are set to phase co
mode.

CCLR2 to CCLR0	—	—	—	—	Other than B'110
Output function	—	Output compare output	—	—	PWM mode 2 output

TPU Channel 0 Setting	Table Below (1)	Table Below (2)		
P12DDR	—	0	1	1
NDER10	—	—	0	1
Pin function	TIOCC0 output	P12 input	P12 output	PO10 output
		TIOCC0 input *1		
	TCLKA input *2			

Operating mode	Mode 7			
AE3 to AE0	—			
TPU Channel 0 Setting	Table Below (1)	Table Below (2)		
P12DDR	—	0	1	
NDER10	—	—	0	
Pin function	TIOCC0 output	P12 input	P12 output	PC
		TIOCC0 input*1		
	TCLKA input*2			

CCLR2 to CCLR0	—	—	—	—	Other than B'101
Output function	—	Output compare output	—	PWM mode 1 output*3	PWM mode output

Notes: 1. TIOCC0 input when MD3 to MD0 = B'0000, and IOC3 t B'10xx.

2. TCLKA input when the setting for TCR0 to TCR5 is: TP TPSC0 = B'100.

   TCLKA input when channels 1 and 5 are set to phase c mode.

3. TIOCD0 output is disabled.
   When BFA = 1 or BFB = 1 in TMDR0, output is disable setting (2) applies.

TPU Channel 0 Setting	Table Below (1)	Table Below (2)			
P11DDR	—	0	1	1	
NDER9	—	—	0	1	
Pin function	TIOCB0 output	P11 input	P11 output	PO9 output	
		TIOCB0 input*			

Operating mode	Mode 7			
AE3 to AE0	—			
TPU Channel 0 Setting	Table Below (1)	Table Below (2)		
P11DDR	—	0	1	
NDER9	—	—	0	
Pin function	TIOCB0 output	P11 input	P11 output	P
		TIOCB0 input*		

Note: * TIOCB0 input when MD3 to MD0 = B'0000, and IOB3 to B'10xx.

CCLR2 to CCLR0	—	—	—	—	Other than B'010
Output function	—	Output compare output	—	—	PWM mode 2 output

TPU Channel 0 Setting	Table Below (1)	Table Below (2)			
P10DDR	—	0	1	1	
NDER8	—	—	0	1	
Pin function	TIOCA0 output	P10 input	P10 output	PO8 output	
		TIOCA0 input*1			

Operating mode	Mode 7			
AE3 to AE0	—			
TPU Channel 0 Setting	Table Below (1)	Table Below (2)		
P10DDR	—	0	1	
NDER8	—	—	0	
Pin function	TIOCA0 output	P10 input	P10 output	P
		TIOCA0 input*1		

CCLR2 to CCLR0	—	—	—	—	Other than B'001
Output function	—	Output compare output	—	PWM mode 1 output*2	PWM mode 2 output

Notes: 1. TIOCA0 input when MD3 to MD0 = B'0000, and IOA3 t B'10xx.
2. TIOCB0 output is disabled.

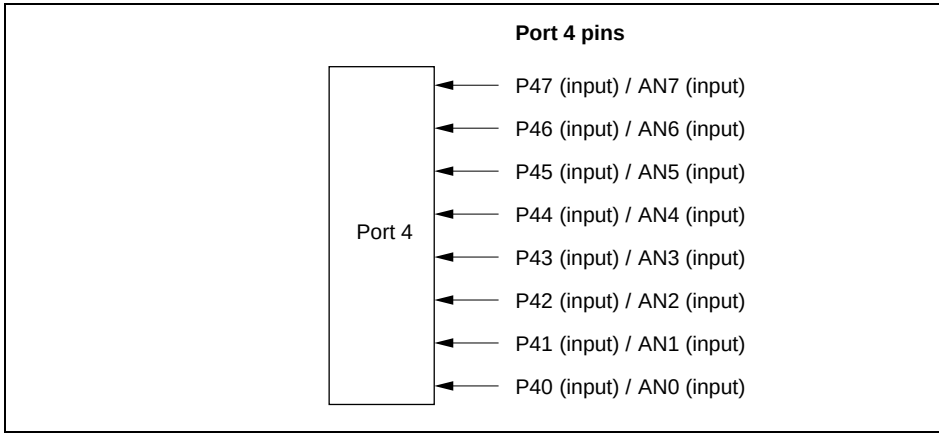


Figure 9.2 Port 4 Pin Functions

Port 4 register	PORT4	R	Undefined	

Note: * Lower 16 bits of the address.

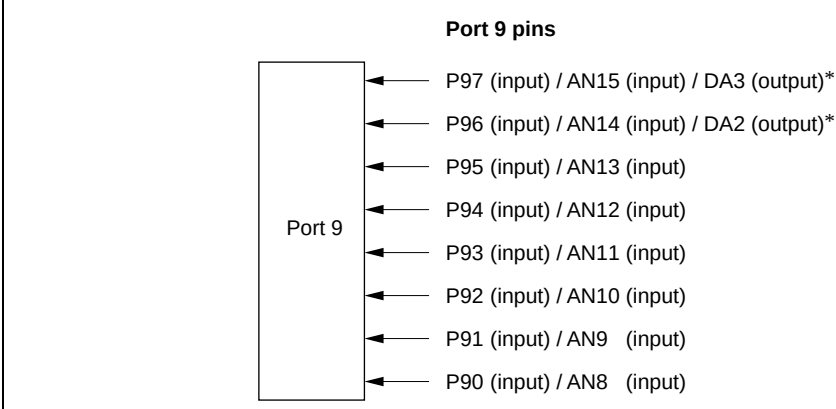
Port 4 Register (PORT4): The pin states are always read when a port 4 read is perfo

Bit	:	7	6	5	4	3	2	1
		P47	P46	P45	P44	P43	P42	P41
Initial value	:	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R

Note: * Determined by state of pins P47 to P40.

9.3.3 Pin Functions

Port 4 pins also function as A/D converter analog input pins (AN0 to AN7).



Note: * DA3 and DA2 are outputs in the H8S/2626 Group only.

Figure 9.3 Port 9 Pin Functions

Note: * Lower 16 bits of the address.

Port 9 Register (PORT9): The pin states are always read when a port 9 read is perfo

Bit	:	7	6	5	4	3	2	1
		P97	P96	P95	P94	P93	P92	P91
Initial value	:	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R

Note: * Determined by state of pins P97 to P90.

9.4.3 Pin Functions

Port 9 pins also function as A/D converter analog input pins (AN8 to AN15) and D/A analog output pins (DA3, DA2).

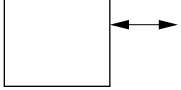
9.5 Port A

9.5.1 Overview

Port A is a 6-bit I/O port. Port A pins also function as address bus outputs and SCI2 I/ (SCK2, RxD2, and TxD2). The pin functions change according to the operating mode

Port A has a built-in MOS input pull-up function that can be controlled by software.

Figure 9.4 shows the port A pin configuration.



PA0 (I/O) / A16 (output)

Pin functions in mode 7

PA5 (I/O)

PA4 (I/O)

PA3 (I/O) / SCK2 (output)

PA2 (I/O) / RxD2 (input)

PA1 (I/O) / TxD2 (output)

PA0 (I/O)

Note: * In the H8S/2626 Series, PA5 and PA4 are OSC2 and OSC1, respectively.

Figure 9.4 Port A Pin Functions

9.5.2 Register Configuration

Table 9.6 shows the port A register configuration.

Table 9.6 Port A Registers

Name	Abbreviation	R/W	Initial Value*2	Ad
Port A data direction register	PADDR	W	H'0	H'
Port A data register	PADR	R/W	H'0	H'
Port A register	PORTA	R	Undefined	H'
Port A MOS pull-up control register	PAPCR	R/W	H'0	H'
Port A open-drain control register	PAODR	R/W	H'0	H'

Notes: 1. Lower 16 bits of the address.
2. Value of bits 3 to 0.

PADDR is an 8-bit write-only register, the individual bits of which specify input or ou
pins of port A. PADDR cannot be read; if it is, an undefined value will be read.

Bits 7 and 6 are reserved; they return an undetermined value if read.

PADDR is initialized to H'0 (bits 5 to 0) by a reset, and in hardware standby mode. It
prior state in software standby mode. The OPE bit in SBYCR is used to select whethe
output pins retain their output state or become high-impedance when a transition is ma
software standby mode.

- Modes 4 to 6

  The corresponding port A pins become address outputs in accordance with the sett
  AE3 to AE0 in PFCR, irrespective of the value of bits PA5DDR to PA0DDR. Wh
  not used as address outputs, setting a PADDR bit to 1 makes the corresponding po
  output port, while clearing the bit to 0 makes the pin an input port.

- Mode 7

  Setting a PADDR bit to 1 makes the corresponding port A pin an output port, whil
  the bit to 0 makes the pin an input port.

Port A Data Register (PADR)

Bit	:	7	6	5	4	3	2	1
		—	—	PA5DR*	PA4DR*	PA3DR	PA2DR	PA1D
Initial value	:	Undefined	Undefined	0	0	0	0	0
R/W	:	—	—	R/W	R/W	R/W	R/W	R/W

Note: * In the H8S/2626 Group bits 5 and 4 are reserved, and will return an undefi
read.

Bit	:	7	6	5	4	3	2	1
		—	—	PA5*2	PA4*2	PA3	PA2	PA1
Initial value	:	Undefined	Undefined	—*1	—*1	—*1	—*1	—*1
R/W	:	—	—	R	R	R	R	R

Notes: 1. Determined by state of pins PA5 to PA0.
2. In the H8S/2626 Group bits 5 and 4 are reserved, and will return an undefine read.

PORTA is an 8-bit read-only register that shows the pin states. It cannot be written to. V output data for the port A pins (PA5 to PA0) must always be performed on PADR.

Bits 7 and 6 are reserved; they return an undetermined value if read, and cannot be mod

If a port A read is performed while PADDR bits are set to 1, the PADR values are read read is performed while PADDR bits are cleared to 0, the pin states are read.

After a reset and in hardware standby mode, PORTA contents are determined by the pi PADDR and PADR are initialized. PORTA retains its prior state in software standby m

Port A MOS Pull-Up Control Register (PAPCR)

Bit	:	7	6	5	4	3	2	1
		—	—	PA5PCR*	PA4PCR*	PA3PCR	PA2PCR	PA1PCR
Initial value	:	Undefined	Undefined	0	0	0	0	0
R/W	:	—	—	R/W	R/W	R/W	R/W	R/W

Note: * In the H8S/2626 Group bits 5 and 4 are reserved, and will return an undefine read.

PAPCR is an 8-bit readable/writable register that controls the MOS input pull-up functi incorporated into port A on an individual bit basis.

PAPCR is initialized to H'0 (bits 5 to 0) by a reset, and in hardware standby mode. It r
prior state in software standby mode.

Port A Open Drain Control Register (PAODR)

Bit	:	7	6	5	4	3	2	1
		—	—	PA5ODR*	PA4ODR*	PA3ODR	PA2ODR	PA1OD
Initial value	:	Undefined	Undefined	0	0	0	0	0
R/W	:	—	—	R/W	R/W	R/W	R/W	R/W

Note: * In the H8S/2626 Group bits 5 and 4 are reserved, and will return an undefi
read.

PAODR is an 8-bit readable/writable register that controls whether PMOS is on or off
port A pin (PA5 to PA0).

Bits 7 and 6 are reserved; they return an undetermined value if read, and cannot be mo

When pins are not address outputs in accordance with the setting of bits AE3 to AE0 i
setting a PAODR bit makes the corresponding port A pin an NMOS open-drain outpu
clearing the bit to 0 makes the pin a CMOS output.

PAODR is initialized to H'0 (bits 3 to 0) by a reset, and in hardware standby mode. It
prior state in software standby mode.

PA5 The pin function is switched as shown below according to bit PA5DD

PA5DDR	0	1
Pin function	PA5 input	PA5 outpu

Note: * In the H8S/2626 Group, PA5 is OSC2.

PA4* The pin function is switched as shown below according to bit PA4DD

PA4DDR	0	1
Pin function	PA4 input	PA4 outpu

Note: * In the H8S/2626 Group, PA4 is OSC1.

PA3/A19/SCK2 The pin function is switched as shown below according to the operati bits AE3 to AE0 in PFCR, bit C/$\overline{A}$ in SMR and bits CKE0 and CKE1 i SCI2, and bit PA3DDR.

Operating mode	Modes 4 to 6					
AE3 to AE0	B'0000 to B'1011					B'1100
CKE1	0				1	
C/$\overline{A}$	0			1	—	
CKE0	0		1	—	—	
PA3DDR	0	1	—	—	—	
Pin function	PA3 input	PA3 output	SCK2 output	SCK2 output	SCK2 input	A19

Operating mode	Mode 7				
CKE1	0				
C/$\overline{A}$	0			1	
CKE0	0		1	—	
PA3DDR	0	1	—	—	
Pin function	PA3 input	PA3 output	SCK2 output	SCK2 output	

Pin function	PA2 input	PA2 output	RxD2 input	A

Operating mode	Mode 7		
RE	0		
PA2DDR	0	1	
Pin function	PA2 input	PA2 output	RxI

PA1/A17/TxD2 The pin function is switched as shown below according to the opera bits AE3 to AE0 in PFCR, bit TE in SCR of SCI2, and bit PA1DDR.

Operating mode	Modes 4 to 6			
AE3 to AE0	B'0000 to B'1001			B'101
TE	0		1	
PA1DDR	0	1	—	
Pin function	PA1 input	PA1 output	TxD2 output	A1

Operating mode	Mode 7		
TE	0		
PA1DDR	0	1	
Pin function	PA1 input	PA1 output	TxD

Operating mode	Mode 7	
PA0DDR	0	1
Pin function	PA0 input	PA0 outpu

In mode 7, if a pin is in the input state in accordance with the settings in the SCI's SCI and SCR, and in DDR, setting the corresponding PAPCR bit to 1 turns on the MOS in for that pin.

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

Table 9.8 summarizes the MOS input pull-up states.

Table 9.8 MOS Input Pull-Up States (Port A)

Pin States	Power-On Reset	Hardware Standby Mode	Software Standby Mode	In Oth Opera
Address output or SCI output	OFF	OFF	OFF	OFF
Other than above			ON/OFF	ON/O

Legend:

OFF : MOS input pull-up is always off.

ON/OFF : On when PADDR = 0 and PAPCR = 1; otherwise off.

Port B has a built-in MOS input pull-up function that can be controlled by software.

Figure 9.5 shows the port B pin configuration.

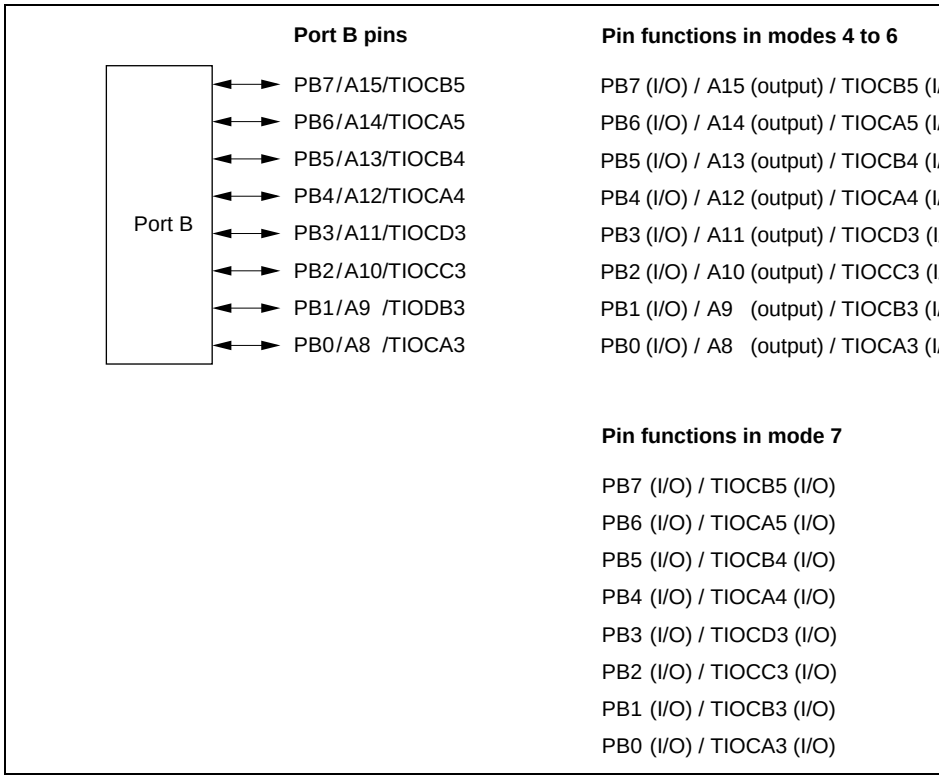


Figure 9.5 Port B Pin Functions

Port B register	PORTB	R	Undefined	H
Port B MOS pull-up control register	PBPCR	R/W	H'00	H
Port B open-drain control register	PBODR	R/W	H'00	H

Note: * Lower 16 bits of the address.

Port B Data Direction Register (PBDDR)

Bit	:	7	6	5	4	3	2	1
		PB7DDR	PB6DDR	PB5DDR	PB4DDR	PB3DDR	PB2DDR	PB1DD
Initial value	:	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W

PBDDR is an 8-bit write-only register, the individual bits of which specify input or ou pins of port B. PBDDR cannot be read; if it is, an undefined value will be read.

PBDDR is initialized to H'00 by a reset, and in hardware standby mode. It retains its p software standby mode. The OPE bit in SBYCR is used to select whether the address retain their output state or become high-impedance when a transition is made to softw mode.

- Modes 4 to 6

  The corresponding port B pins become address outputs in accordance with the sett AE3 to AE0 in PFCR, irrespective of the value of the PBDDR bits. When pins are address outputs, setting a PBDDR bit to 1 makes the corresponding port B pin an o while clearing the bit to 0 makes the pin an input port.

- Mode 7

  Setting a PBDDR bit to 1 makes the corresponding port B pin an output port, whil the bit to 0 makes the pin an input port.

PB0). PBDR is initialized to H'00 by a reset, and in hardware standby mode. It retains state in software standby mode.

Port B Register (PORTB)

Bit	:	7	6	5	4	3	2	1
		PB7	PB6	PB5	PB4	PB3	PB2	PB1
Initial value	:	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R

Note: * Determined by state of pins PB7 to PB0.

PORTB is an 8-bit read-only register that shows the pin states. It cannot be written to. V output data for the port B pins (PB7 to PB0) must always be performed on PBDR.

If a port B read is performed while PBDDR bits are set to 1, the PBDR values are read. read is performed while PBDDR bits are cleared to 0, the pin states are read.

After a reset and in hardware standby mode, PORTB contents are determined by the pi PBDDR and PBDR are initialized. PORTB retains its prior state in software standby m

Port B MOS Pull-Up Control Register (PBPCR)

Bit	:	7	6	5	4	3	2	1
		PB7PCR	PB6PCR	PB5PCR	PB4PCR	PB3PCR	PB2PCR	PB1PCR
Initial value	:	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PBPCR is an 8-bit readable/writable register that controls the MOS input pull-up functi incorporated into port B on an individual bit basis.

Port B Open Drain Control Register (PBODR)

Bit	:	7	6	5	4	3	2	1
		PB7ODR	PB6ODR	PB5ODR	PB4ODR	PB3ODR	PB2ODR	PB1OD
Initial value	:	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PBODR is an 8-bit readable/writable register that controls the PMOS on/off state for e
pin (PB7 to PB0).

When pins are not address outputs in accordance with the setting of bits AE3 to AE0 i
setting a PBODR bit makes the corresponding port B pin an NMOS open-drain output
clearing the bit to 0 makes the pin a CMOS output.

PBODR is initialized to H'00 by a reset, and in hardware standby mode. It retains its p
software standby mode.

9.6.3 Pin Functions

Port B pins also function as TPU input/output pins (TIOCA3, TIOCB3, TIOCC3, TIO
TIOCA4, TIOCB4, TIOCA5, TIOCB5) and address bus output pins (A15 to A8). Port
functions are shown in table 9.10.

TPU channel 5 settings	(1) in table below	(2) in table below		
PB7DDR	—	0	1	
Pin function	TIOCB5 output	PB7 input	PB7 output	A15
		TIOCB5 input*		

Operating mode	Mode 7		
TPU channel 5 settings	(1) in table below	(2) in table below	
PB7DDR	—	0	1
Pin function	TIOCB5 output	PB7 input	PB7 o
		TIOCB5 input*	

TPU channel 5 settings	(2)	(1)	(2)	(2)	(1)
MD3 to MD0	B'0000, B'01xx		B'0010	B'0011	
IOB3 to IOB0	B'0000, B'0100, B'1xxx	B'0001 to B'0011, B'0101 to B'0111	—	B'xx00	Not B
CCLR1, CCLR0	—	—	—	—	Not B'10
Output function	—	Output compare output	—	—	PWM mode 2 output

x

Note: * TIOCB5 input when MD3 to MD0 = B'0000 or B'01xx, and IOB3

PB6DDR	—	0	1	
Pin function	TIOCA5 output	PB6 input	PB6 output	A1
		TIOCA5 input*1		

Operating mode	Mode 7		
TPU channel 5 settings	(1) in table below	(2) in table below	
PB6DDR	—	0	
Pin function	TIOCA5 output	PB6 input	PB6
		TIOCA5 input*1	

TPU channel 5 settings	(2)	(1)	(2)	(1)	(1)
MD3 to MD0	B'0000, B'01xx		B'001x	B'0010	B'(
IOA3 to IOA0	B'0000, B'0100, B'1xxx	B'0001 to B'0011, B'0101 to B'0111	B'xx00	Not B'xx00	Not
CCLR1, CCLR0	—	—	—	—	Not B'01
Output function	—	Output compare output	—	PWM mode 1 output*2	PWM mode 2 output

Notes: 1. TIOCA5 input when MD3 to MD0 = B'0000 or B'01xx, and IOA
2. TIOCB5 is disabled for output.

PB5DDR	—	0	1	
Pin function	TIOCB4 output	PB5 input	PB5 output	A13
		TIOCB4 input*		

Operating mode	Mode 7		
TPU channel 4 settings	(1) in table below	(2) in table below	
PB5DDR	—	0	1
Pin function	TIOCB4 output	PB5 input	PB5 o
		TIOCB4 input*	

TPU channel 5 settings	(2)	(1)	(2)	(2)	(1)
MD3 to MD0	B'0000, B'01xx		B'0010	B'0011	
IOB3 to IOB0	B'0000, B'0100, B'1xxx	B'0001 to B'0011, B'0101 to B'0111	—	B'xx00	Not B
CCLR1, CCLR0	—	—	—	—	Not B'10
Output function	1	Output compare output	1	1	PWM mode 2 output

x

Note: * TIOCB4 input when MD3 to MD0 = B'0000 or B'01xx, and IOB3 B'10xx.

PB4DDR	—	0	1	
Pin function	TIOCA4 output	PB4 input	PB4 output	A1
		TIOCA4 input*1		

Operating mode	Mode 7		
TPU channel 4 settings	(1) in table below	(2) in table below	
PB4DDR	—	0	
Pin function	TIOCA4 output	PB4 input	PB4
		TIOCA4 input*1	

TPU channel 4 settings	(2)	(1)	(2)	(1)	(1)
MD3 to MD0	B'0000, B'01xx		B'001x	B'0010	B'(
IOA3 to IOA0	B'0000, B'0100, B'1xxx	B'0001 to B'0011, B'0101 to B'0111	B'xx00	Not B'xx00	Not
CCLR1, CCLR0	—	—	—	—	Not B'01
Output function	—	Output compare output	—	PWM mode 1 output*2	PWM mode 2 output

Notes: 1. TIOCA4 input when MD3 to MD0 = B'0000 or B'01xx, and IOA
B'10xx.
2. TIOCB4 is disabled for output.

PB3DDR	—	0	1	
Pin function	TIOCD3 output	PB3 input	PB3 output	A11
		TIOCD3 input*		

Operating mode	Mode 7		
TPU channel 3 settings	(1) in table below	(2) in table below	
PB3DDR	—	0	1
Pin function	TIOCD3 output	PB3 input	PB3 o
		TIOCD3 input*	

TPU channel 3 settings	(2)	(1)	(2)	(2)	(1)
MD3 to MD0	B'0000		B'0010	B'0011	
IOD3 to IOD0	B'0000, B'0100, B'1xxx	B'0001 to B'0011, B'0101 to B'0111	—	B'xx00	Not B
CCLR2 to CCLR0	—	—	—	—	Not B'110
Output function	—	Output compare output	—	—	PWM mode 2 output

x

Note: * TIOCD3 input when MD3 to MD0 = B'0000 or B'01xx, and IOD3 B'10xx.

PB2DDR	—	0	1	
Pin function	TIOCC3 output	PB2 input	PB2 output	A1
		TIOCC3 input*1		

Operating mode	Mode 7		
TPU channel 3 settings	(1) in table below	(2) in table below	
PB2DDR	—	0	
Pin function	TIOCC3 output	PB2 input	PB2
		TIOCC3 input*1	

TPU channel 3 settings	(2)	(1)	(2)	(1)	(1)
MD3 to MD0	B'0000		B'001x	B'0010	B'(
IOC3 to IOC0	B'0000, B'0100, B'1xxx	B'0001 to B'0011, B'0101 to B'0111	B'xx00	Not B'xx00	Not
CCLR2 to CCLR0	—	—	—	—	Not B'101
Output function	—	Output compare output	—	PWM mode 1 output*2	PWM mode 2 output

Notes: 1. TIOCC3 input when MD3 to MD0 = B'0000, and IOC3 to IOC0
2. TIOCD3 is disabled for output.

PB1DDR	—	0	1	
Pin function	TIOCB3 output	PB1 input	PB1 output	A9
		TIOCB3 input*		

Operating mode	Mode 7		
TPU channel 3 settings	(1) in table below	(2) in table below	
PB1DDR	—	0	1
Pin function	TIOCB3 output	PB1 input	PB1 o
		TIOCB3 input*	

TPU channel 3 settings	(2)	(1)	(2)	(2)	(1)
MD3 to MD0	B'0000		B'0010	B'0011	
IOB3 to IOB0	B'0000, B'0100, B'1xxx	B'0001 to B'0011, B'0101 to B'0111	—	B'xx00	Not B
CCLR2 to CCLR0	—	—	—	—	Not B'010
Output function	—	Output compare output	—	—	PWM mode 2 output

x

Note: * TIOCB3 input when MD3 to MD0 = B'0000, and IOB3 to IOB0 =

PB0DDR	—	0	1	
Pin function	TIOCA3 output	PB0 input	PB0 output	A
		TIOCA3 input*1		

Operating mode	Mode 7		
TPU channel 3 settings	(1) in table below	(2) in table below	
PB0DDR	—	0	
Pin function	TIOCA3 output	PB0 input	PB0
		TIOCA3 input*1	

TPU channel 3 settings	(2)	(1)	(2)	(1)	(1)
MD3 to MD0	B'0000		B'001x	B'0010	B'
IOA3 to IOA0	B'0000, B'0100, B'1xxx	B'0001 to B'0011, B'0101 to B'0111	B'xx00	Not B'xx00	Not
CCLR2 to CCLR0	—	—	—	—	Not B'001
Output function	—	Output compare output	—	PWM mode 1 output*2	PWM mode 2 output

Notes: 1. TIOCA3 input when MD3 to MD0 = B'0000, and IOA3 to IOA0
2. TIOCB3 is disabled for output.

In mode 7, if a pin is in the input state in accordance with the settings in the TPU's TIO DDR, setting the corresponding PBPCR bit to 1 turns on the MOS input pull-up for tha

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

Table 9.11 summarizes the MOS input pull-up states.

Table 9.11 MOS Input Pull-Up States (Port B)

Pin States	Power-On Reset	Hardware Standby Mode	Software Standby Mode	In Othe Operat
Address output or TPU output	OFF	OFF	OFF	OFF
Other than above			ON/OFF	ON/OF

Legend:

OFF : MOS input pull-up is always off.

ON/OFF : On when PBDDR = 0 and PBPCR = 1; otherwise off.

Port C has a built-in MOS input pull-up function that can be controlled by software.

Figure 9.6 shows the port C pin configuration.

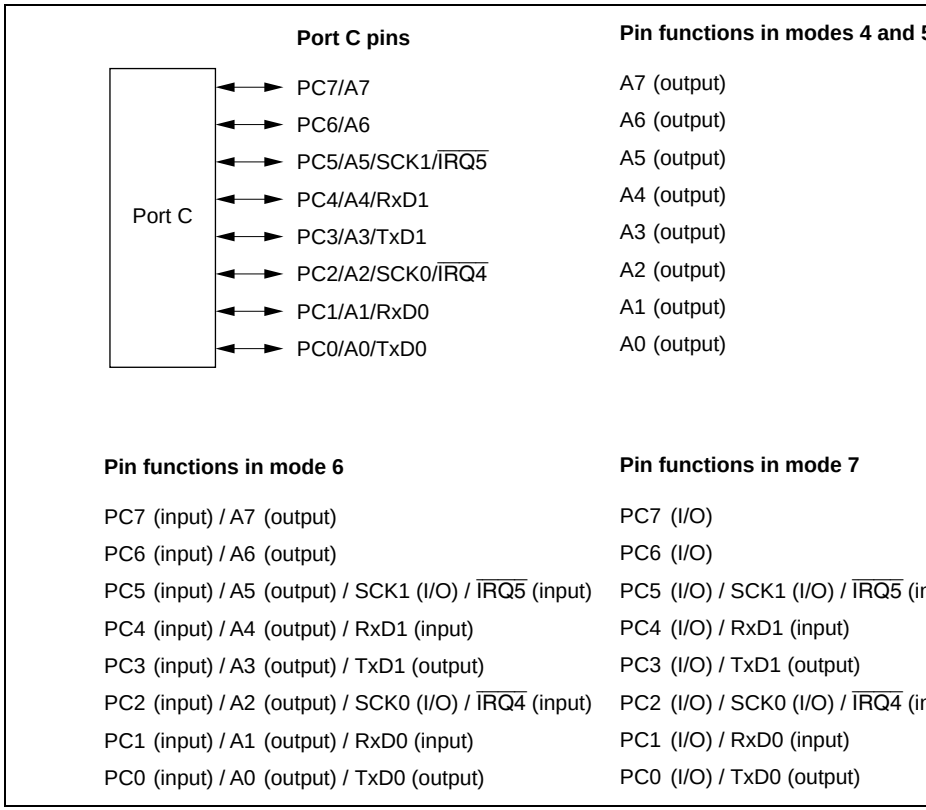


Figure 9.6 Port C Pin Functions

Port C register	PORTC	R	Undefined	H'
Port C MOS pull-up control register	PCPCR	R/W	H'00	H'
Port C open-drain control register	PCODR	R/W	H'00	H'

Note: * Lower 16 bits of the address.

Port C Data Direction Register (PCDDR)

Bit	:	7	6	5	4	3	2	1
		PC7DDR	PC6DDR	PC5DDR	PC4DDR	PC3DDR	PC2DDR	PC1DDR
Initial value	:	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W

PCDDR is an 8-bit write-only register, the individual bits of which specify input or out
pins of port C. PCDDR cannot be read; if it is, an undefined value will be read.

PCDDR is initialized to H'00 by a reset, and in hardware standby mode. It retains its pr
software standby mode. As the SCI is initialized, pin states are determined by the PCD
PCDR specifications. The OPE bit in SBYCR is used to select whether the address outp
retain their output state or become high-impedance when a transition is made to softwa
mode.

- Modes 4 and 5
  The corresponding port C pins are address outputs irrespective of the value of the P
- Mode 6
  Setting a PCDDR bit to 1 makes the corresponding port C pin an address output, w
  clearing the bit to 0 makes the pin an input port.
- Mode 7
  Setting a PCDDR bit to 1 makes the corresponding port C pin an output port, while
  the bit to 0 makes the pin an input port.

PC0).

PCDR is initialized to H'00 by a reset, and in hardware standby mode. It retains its pri
software standby mode.

Port C Register (PORTC)

Bit	:	7	6	5	4	3	2	1
		PC7	PC6	PC5	PC4	PC3	PC2	PC1
Initial value	:	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R

Note: * Determined by state of pins PC7 to PC0.

PORTC is an 8-bit read-only register that shows the pin states. It cannot be written to.
output data for the port C pins (PC7 to PC0) must always be performed on PCDR.

If a port C read is performed while PCDDR bits are set to 1, the PCDR values are read
read is performed while PCDDR bits are cleared to 0, the pin states are read.

After a reset and in hardware standby mode, PORTC contents are determined by the p
PCDDR and PCDR are initialized. PORTC retains its prior state in software standby m

incorporated into port C on an individual bit basis.

In modes 6 and 7, if a pin is in the input state in accordance with the settings in the SCI and SCR, and in PCDDR, setting the corresponding PCPCR bit to 1 turns on the MOS up for that pin.

PCPCR is initialized to H'00 by a reset, and in hardware standby mode. It retains its pri software standby mode.

Port C Open Drain Control Register (PCODR)

Bit	:	7	6	5	4	3	2	1
		PC7ODR	PC6ODR	PC5ODR	PC4ODR	PC3ODR	PC2ODR	PC1ODI
Initial value	:	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PCODR is an 8-bit readable/writable register that controls the PMOS on/off status for e pin (PC7 to PC0).

If the setting of bits AE3 to AE0 in PFCR is other than address output, setting a PCOD makes the corresponding port C pin an NMOS open-drain output, while clearing the bit the pin a CMOS output.

PCODR is initialized to H'00 by a reset, and in hardware standby mode. It retains its p after a manual reset, and in software standby mode.

PC7/A7

The pin function is switched as shown below according to the oper
and bit PC7DDR.

Operating Mode	Modes 4 and 5	Mode 6		Mo
PC7DDR	—	0	1	0
Pin function	A7 output	PC7 input	A7 output	PC7 input

PC6/A6

The pin function is switched as shown below according to the oper
and bit PC6DDR.

Operating Mode	Modes 4 and 5	Mode 6		Mo
PC6DDR	—	0	1	0
Pin function	A6 output	PC6 input	A6 output	PC6 input

CKE0	—	0	1	—	—
Pin function	A5 output	PC5 input	SCK1 output	SCK1 output	SCK1 input
	$\overline{\text{IRQ5}}$ input				

Operating Mode	Mode 7				
CKE1	0				
C/$\overline{\text{A}}$	0			1	
CKE0	0		1	—	
PC5DDR	0	1	—	—	
Pin function	PC5 input	PC5 output	SCK1 output	SCK1 output	
	$\overline{\text{IRQ5}}$ input				

Operating Mode	Mode 7		
RE	0		
PC4DDR	0	1	
Pin function	PC4 input	PC4 output	RxI

PC3/A3/TxD1 The pin function is switched as shown below according to the oper bit TE in the SCI1's SCR, and bit PC3DDR.

Operating Mode	Modes 4 and 5	Mode 6		
PC3DDR	—	0		
TE	—	0	1	
Pin function	A3 output	PC3 input	TxD1 output	

Operating Mode	Mode 7		
TE	0		
PC3DDR	0	1	
Pin function	PC3 input	PC3 output	TxD

CKE0	—	0	1	—	—
Pin function	A2 output	PC2 input	SCK0 output	SCK0 output	SCK0 input
	$\overline{\text{IRQ4}}$ input				

Operating Mode	Mode 7				
CKE1	0				
C/$\overline{\text{A}}$	0			1	
CKE0	0		1	—	
PC2DDR	0	1	—	—	
Pin function	PC2 input	PC2 output	SCK0 output	SCK0 output	
	$\overline{\text{IRQ4}}$ input				

Operating Mode	Mode 7		
RE	0		
PC1DDR	0	1	
Pin function	PC1 input	PC1 output	RxI

PC0/A0/TxD0

The pin function is switched as shown below according to the oper
bit TE in the SCI0's SCR, and bit PC0DDR.

Operating Mode	Modes 4 and 5	Mode 6		
PC0DDR	—	0		
TE	—	0	1	
Pin function	A0 output	PC0 input	TxD0 output	

Operating Mode	Mode 7		
TE	0		
PC0DDR	0	1	
Pin function	PC0 input	PC0 output	TxD

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

Table 9.14 summarizes the MOS input pull-up states.

Table 9.14 MOS Input Pull-Up States (Port C)

Pin States	Power-On Reset	Hardware Standby Mode	Software Standby Mode	In Othe Operat
Address output	OFF	OFF	OFF	OFF
Other than above			ON/OFF	ON/OF

Legend:
OFF : MOS input pull-up is always off.
ON/OFF : On when PCDDR = 0 and PCPCR = 1; otherwise off.

Figure 9.7 shows the port D pin configuration.

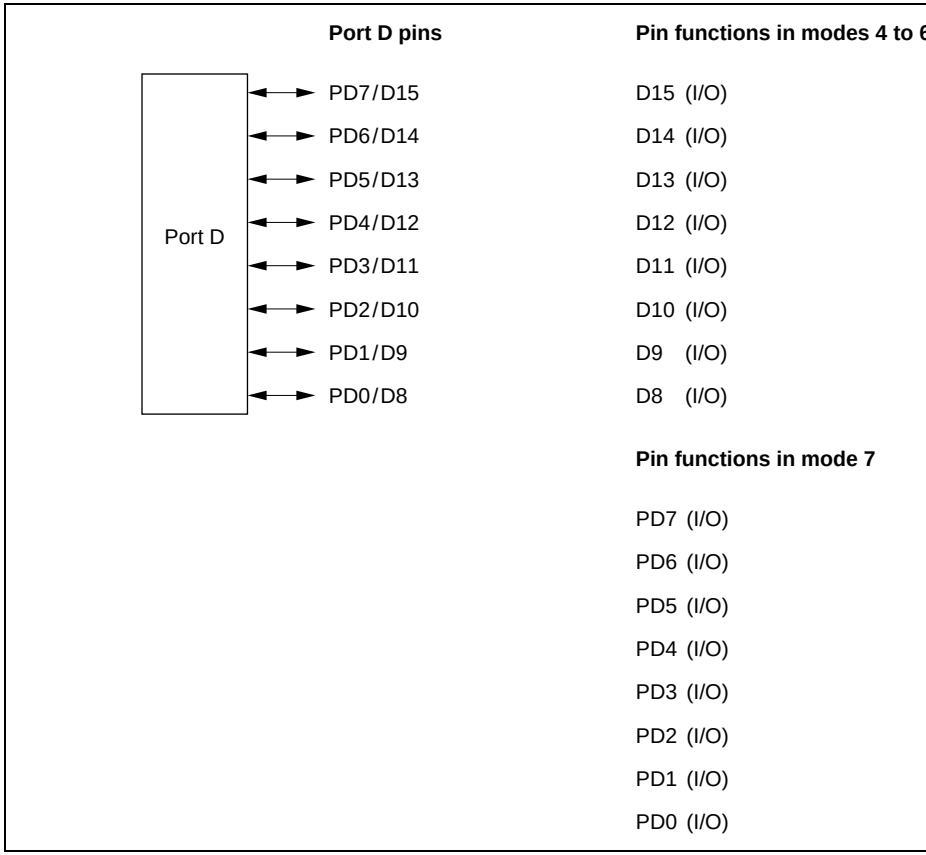


Figure 9.7 Port D Pin Functions

Port D register	PORTD	R	Undefined	H'
Port D MOS pull-up control register	PDPCR	R/W	H'00	H'

Note: * Lower 16 bits of the address.

Port D Data Direction Register (PDDDR)

Bit	:	7	6	5	4	3	2	1
		PD7DDR	PD6DDR	PD5DDR	PD4DDR	PD3DDR	PD2DDR	PD1DDI
Initial value	:	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W

PDDDR is an 8-bit write-only register, the individual bits of which specify input or out pins of port D. PDDDR cannot be read; if it is, an undefined value will be read.

PDDDR is initialized to H'00 by a reset, and in hardware standby mode. It retains its pr software standby mode.

- Modes 4 to 6

  The input/output direction specification by PDDDR is ignored, and port D is autom designated for data I/O.

- Mode 7

  Setting a PDDDR bit to 1 makes the corresponding port D pin an output port, while the bit to 0 makes the pin an input port.

PD0).

PDDR is initialized to H'00 by a reset, and in hardware standby mode. It retains its pri
software standby mode.

Port D Register (PORTD)

Bit	:	7	6	5	4	3	2	1
		PD7	PD6	PD5	PD4	PD3	PD2	PD1
Initial value	:	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R

Note: * Determined by state of pins PD7 to PD0.

PORTD is an 8-bit read-only register that shows the pin states. It cannot be written to.
output data for the port D pins (PD7 to PD0) must always be performed on PDDR.

If a port D read is performed while PDDDR bits are set to 1, the PDDR values are rea
read is performed while PDDDR bits are cleared to 0, the pin states are read.

After a reset and in hardware standby mode, PORTD contents are determined by the p
PDDDR and PDDR are initialized. PORTD retains its prior state in software standby

incorporated into port D on an individual bit basis.

When a PDDDR bit is cleared to 0 (input port setting) in mode 7, setting the correspond
PDPCR bit to 1 turns on the MOS input pull-up for the corresponding pin.

PDPCR is initialized to H'00 by a reset, and in hardware standby mode. It retains its pri
software standby mode.

9.8.3 Pin Functions

In modes 4 to 6, port D pins automatically function as data bus input/output pins (D15 t
mode 7, each pin in port D functions as an input/output port, and input or output can be
individually for each pin. Port D pin functions are shown in table 9.16.

Table 9.16 Port D Pin Functions

Pin	Selection Method and Pin Functions
PD7/D15, PD6/D14, PD5/D13, PD4/D12, PD3/D11, PD2/D10, PD1/D9, PD0/D8	The pin function is switched as shown below according to the operating m PDDDR.

Operating mode	Modes 4 to 6		Mo
PDnDDR	—	0	
Pin function	Data bus input/ output (D15 to D8)	PDn input	PDn

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

Table 9.17 summarizes the MOS input pull-up states.

Table 9.17 MOS Input Pull-Up States (Port D)

Modes	Power-On Reset	Hardware Standby Mode	Software Standby Mode	In Oth Opera
4 to 6	OFF	OFF	OFF	OFF
7			ON/OFF	ON/O

Legend:
OFF : MOS input pull-up is always off.
ON/OFF : On when PDDDR = 0 and PDPCR = 1; otherwise off.

Figure 9.8 shows the port E pin configuration.

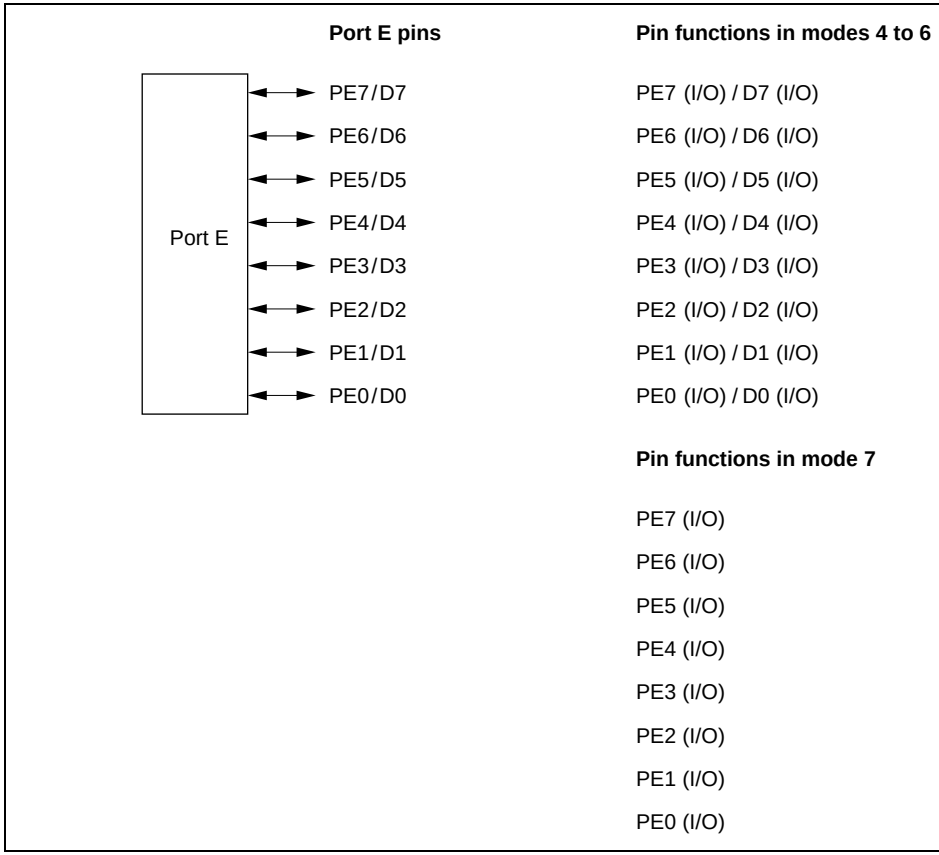


Figure 9.8 Port E Pin Functions

Port E register	PORTE	R	Undefined	H
Port E MOS pull-up control register	PEPCR	R/W	H'00	H

Note: * Lower 16 bits of the address.

Port E Data Direction Register (PEDDR)

Bit	:	7	6	5	4	3	2	1
		PE7DDR	PE6DDR	PE5DDR	PE4DDR	PE3DDR	PE2DDR	PE1DD
Initial value	:	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W

PEDDR is an 8-bit write-only register, the individual bits of which specify input or ou pins of port E. PEDDR cannot be read; if it is, an undefined value will be read.

PEDDR is initialized to H'00 by a reset, and in hardware standby mode. It retains its p software standby mode.

- Modes 4 to 6

  When 8-bit bus mode has been selected, port E pins function as I/O ports. Setting a to 1 makes the corresponding port E pin an output port, while clearing the bit to 0 pin an input port.

  When 16-bit bus mode has been selected, the input/output direction specification b ignored, and port E is designated for data I/O.

  For details of 8-bit and 16-bit bus modes, see section 7, Bus Controller.

- Mode 7

  Setting a PEDDR bit to 1 makes the corresponding port E pin an output port, while bit to 0 makes the pin an input port.

PE0).

PEDR is initialized to H'00 by a reset, and in hardware standby mode. It retains its prio
software standby mode.

Port E Register (PORTE)

Bit	:	7	6	5	4	3	2	1
		PE7	PE6	PE5	PE4	PE3	PE2	PE1
Initial value	:	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R

Note: * Determined by state of pins PE7 to PE0.

PORTE is an 8-bit read-only register that shows the pin states. It cannot be written to. W
output data for the port E pins (PE7 to PE0) must always be performed on PEDR.

If a port E read is performed while PEDDR bits are set to 1, the PEDR values are read.
read is performed while PEDDR bits are cleared to 0, the pin states are read.

After a reset and in hardware standby mode, PORTE contents are determined by the pi
PEDDR and PEDR are initialized. PORTE retains its prior state in software standby mo

incorporated into port E on an individual bit basis.

When a PEDDR bit is cleared to 0 (input port setting) with 8-bit bus mode selected in
6, or in mode 7, setting the corresponding PEPCR bit to 1 turns on the MOS input pul
corresponding pin.

PEPCR is initialized to H'00 by a reset, and in hardware standby mode. It retains its p
software standby mode.

9.9.3 Pin Functions

Port E pins also function as data bus input/output pins (D7 to D0). If at least one of ar
designated as 16-bit bus space in modes 4 to 6, port E pins automatically function as d
input/output pins. If all areas are designated as 8-bit bus space in modes 4 to 6, or in m
pin in port E functions as an input/output port, and input or output can be specified ind
for each pin. Port E pin functions are shown in table 9.19.

Table 9.19 Port E Pin Functions

<table>
<tr><th>Pin</th><th>Selection Method and Pin Functions</th></tr>
<tr><td>PE7/D7,
PE6/D6,
PE5/D5,
PE4/D4,
PE3/D3,
PE2/D2,
PE1/D1,
PE0/D0</td><td>The pin function is switched as shown below according to the operating
ABWCR in the bus controller, and PEDDR.
<table>
<tr><td>Operating mode</td><td colspan="2">Modes 4 to 6</td><td>Mod</td></tr>
<tr><td>ABWCR</td><td colspan="2">H'FF</td><td>Not H'FF</td><td>—</td></tr>
<tr><td>PEnDDR</td><td>0</td><td>1</td><td>—</td><td>0</td></tr>
<tr><td>Pin function</td><td>PEn input</td><td>PEn output</td><td>Data bus input/output (D7 to D0)</td><td>PEn input</td></tr>
</table></td></tr>
</table>

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

Table 9.20 summarizes the MOS input pull-up states.

Table 9.20 MOS Input Pull-Up States (Port E)

Modes		Power-On Reset	Hardware Standby Mode	Software Standby Mode	In Othe Operat
7		OFF	OFF	ON/OFF	ON/OF
4 to 6	8-bit bus				
	16-bit bus			OFF	OFF

Legend:
OFF : MOS input pull-up is always off.
ON/OFF : On when PEDDR = 0 and PEPCR = 1; otherwise off.

Note: * BUZZ output pin in the H8S/2626 Group only.

Figure 9.9 shows the port F pin configuration.

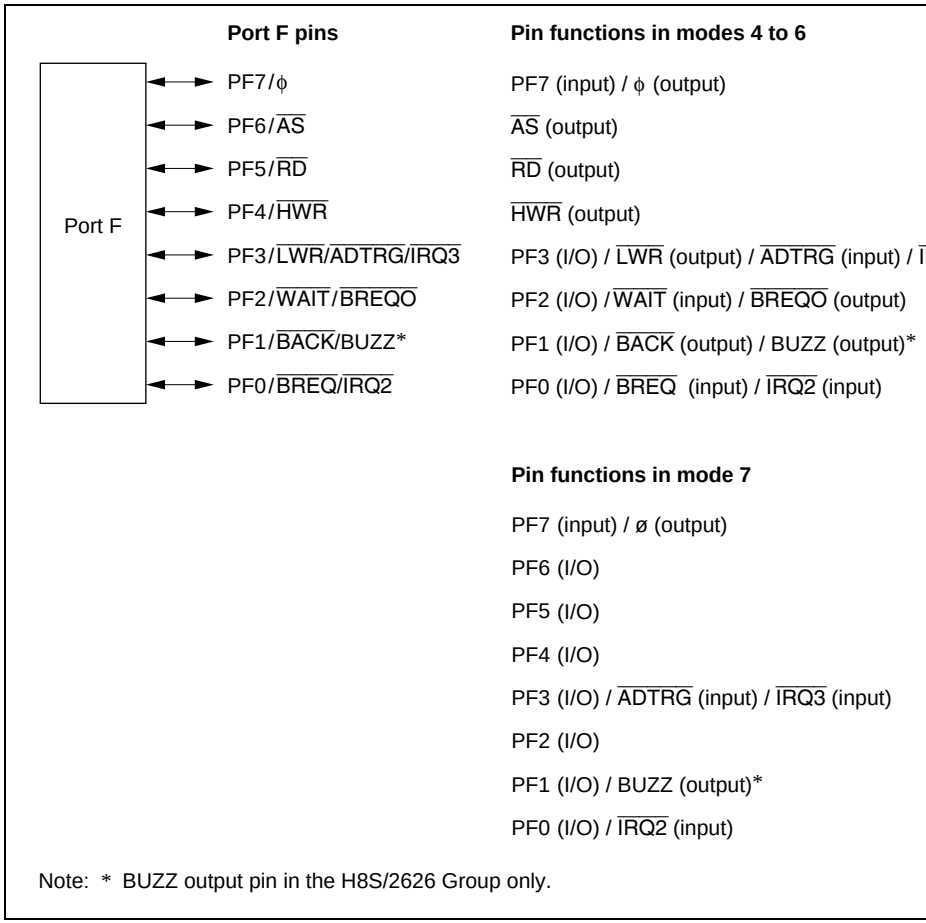


Figure 9.9 Port F Pin Functions

Port F register	PORTF	R	Undefined	H'

Notes: 1. Lower 16 bits of the address.
2. Initial value depends on the mode.

Port F Data Direction Register (PFDDR)

Bit	:	7	6	5	4	3	2	1
		PF7DDR	PF6DDR	PF5DDR	PF4DDR	PF3DDR	PF2DDR	PF1DDR
Modes 4 to 6								
Initial value	:	1	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W
Mode 7								
Initial value	:	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W

PFDDR is an 8-bit write-only register, the individual bits of which specify input or outp
pins of port F. PFDDR cannot be read; if it is, an undefined value will be read.

PFDDR is initialized by a reset, and in hardware standby mode, to H'80 in modes 4 to
H'00 in mode 7. It retains its prior state in software standby mode. The OPE bit in SBY
to select whether the bus control output pins retain their output state or become high-im
when a transition is made to software standby mode.

- Modes 4 to 6

  Pin PF7 functions as the ϕ output pin when the corresponding PFDDR bit is set to 1
  input port when the bit is cleared to 0.

  The input/output direction specified by PFDDR is ignored for pins PF6 to PF3, whi
  automatically designated as bus control outputs ($\overline{AS}$, $\overline{RD}$, $\overline{HWR}$, and $\overline{LWR}$).

  Pins PF2 to PF0 are designated as bus control input/output pins ($\overline{WAIT}$, $\overline{BREQO}$, $\overline{B}$

Bit	:	7	6	5	4	3	2	1
		—	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1D
Initial value	:	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PFDR is an 8-bit readable/writable register that stores output data for the port F pins (

PFDR is initialized to H'00 by a reset, and in hardware standby mode. It retains its pri
software standby mode.

Bit 7 in PFDR is reserved, and only 0 may be written to it.

Port F Register (PORTF)

Bit	:	7	6	5	4	3	2	1
		PF7	PF6	PF5	PF4	PF3	PF2	PF1
Initial value	:	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R

Note: * Determined by state of pins PF7 to PF0.

PORTF is an 8-bit read-only register that shows the pin states. It cannot be written to.
output data for the port F pins (PF7 to PF0) must always be performed on PFDR.

If a port F read is performed while PFDDR bits are set to 1, the PFDR values are read
read is performed while PFDDR bits are cleared to 0, the pin states are read.

After a reset and in hardware standby mode, PORTF contents are determined by the p
PFDDR and PFDR are initialized. PORTF retains its prior state in software standby m

Table 9.22 Port F Pin Functions

Pin	Selection Method and Pin Functions		
PF7/φ	The pin function is switched as shown below according to bit PF7DDR.		
	PF7DDR	0	1
	Pin function	PF7 input	φ output
PF6/$\overline{AS}$	The pin function is switched as shown below according to the operating mode and bit PF6DDR.		
	Operating Mode	Modes 4 to 6	Mode 7
	PF6DDR	—	0
	Pin function	$\overline{AS}$ output	PF6 input
PF5/$\overline{RD}$	The pin function is switched as shown below according to the operating mode and bit PF5DDR.		
	Operating Mode	Modes 4 to 6	Mode 7
	PF5DDR	—	0
	Pin function	$\overline{RD}$ output	PF5 input
PF4/$\overline{HWR}$	The pin function is switched as shown below according to the operating mode and bit PF4DDR.		
	Operating Mode	Modes 4 to 6	Mode 7
	PF4DDR	—	0
	Pin function	$\overline{HWR}$ output	PF4 input

Pin function	LWR output	PF3 input	PF3 output	PF3 input
		$\overline{\text{ADTRG}}$ input*[1]		
		$\overline{\text{IRQ3}}$ input*[2]		

Notes: 1. $\overline{\text{ADTRG}}$ input when TRGS0 = TRGS1 = 1.

2. When used as an external interrupt input pin, do not us pin for another function.

PF2/$\overline{\text{WAIT}}$/ $\overline{\text{BREQO}}$

The pin function is switched as shown below according to the com the operating mode, and bits BREQOE, WAITE, ABW5 to ABW2, PF2DDR.

Operating Mode	Modes 4 to 6				M
BREQOE	0			1	
WAITE	0		1	—	
PF2DDR	0	1	—	—	0
Pin function	PF2 input	PF2 output	$\overline{\text{WAIT}}$ input	$\overline{\text{BREQO}}$ output	PF2 input

PF1/$\overline{\text{BACK}}$/ BUZZ*

The pin function is switched as shown below according to the com the operating mode, and bits BRLE, BUZZE, and PF1DDR.

Operating Mode	Modes 4 to 6				Mod	
BRLE	0			1	—	
BUZZE	0		1	—	0	
PF1DDR	0	1	—	—	0	1
Pin function	PF1 input	PF1 output	BUZZ* output	$\overline{\text{BACK}}$ output	PF1 input	PF outp

Note: * BUZZ output pin in the H8S/2626 Group only.

		input	output	input	input
		$\overline{IRQ2}$ input			

10.1.1 Features

- Maximum 16-pulse input/output
  - A total of 16 timer general registers (TGRs) are provided (four each for channe and two each for channels 1, 2, 4, and 5), each of which can be set independent output compare/input capture register
  - TGRC and TGRD for channels 0 and 3 can also be used as buffer registers
- Selection of 8 counter input clocks for each channel
- The following operations can be set for each channel:
  - Waveform output at compare match: Selection of 0, 1, or toggle output
  - Input capture function: Selection of rising edge, falling edge, or both edge dete
  - Counter clear operation: Counter clearing possible by compare match or input
  - Synchronous operation:
    Multiple timer counters (TCNT) can be written to simultaneously
    Simultaneous clearing by compare match and input capture possible
    Register simultaneous input/output possible by counter synchronous operation
  - PWM mode: Any PWM output duty can be set
  - Maximum of 15-phase PWM output possible by combination with synchronou
- Buffer operation settable for channels 0 and 3
  - Input capture register double-buffering possible
  - Automatic rewriting of output compare register possible
- Phase counting mode settable independently for each of channels 1, 2, 4, and 5
  - Two-phase encoder pulse up/down-count possible
- Cascaded operation
  - Channel 2 (channel 5) input clock operates as 32-bit counter by setting channel 4) overflow/underflow
- Fast access via internal 16-bit bus
  - Fast access is possible via a 16-bit bus interface

- Programmable pulse generator (PPG) output trigger can be generated
  - Channel 0 to 3 compare match/input capture signals can be used as PPG output
- A/D converter conversion start trigger can be generated
  - Channel 0 to 5 compare match A/input capture A signals can be used as A/D co conversion start trigger
- Module stop mode can be set
  - As the initial setting, TPU operation is halted. Register access is enabled by exit stop mode.

Table 10.1 lists the functions of the TPU.

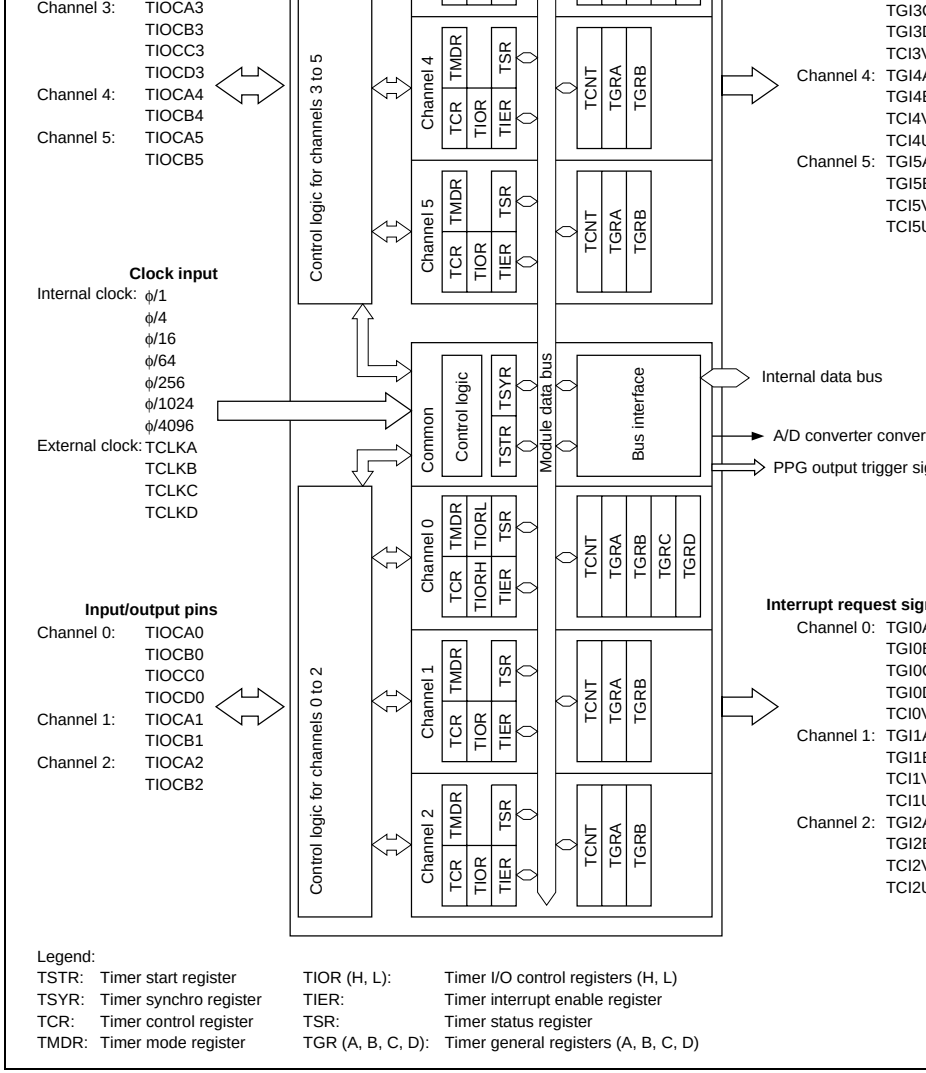
		TCLKC TCLKD	TCLKB	TCLKB TCLKC	ϕ/4096 TCLKA	TCLKC
General registers		TGR0A TGR0B	TGR1A TGR1B	TGR2A TGR2B	TGR3A TGR3B	TGR4A TGR4B
General registers/ buffer registers		TGR0C TGR0D	—	—	TGR3C TGR3D	—
I/O pins		TIOCA0 TIOCB0 TIOCC0 TIOCD0	TIOCA1 TIOCB1	TIOCA2 TIOCB2	TIOCA3 TIOCB3 TIOCC3 TIOCD3	TIOCA4 TIOCB4
Counter clear function		TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture
Compare match output	0 output	○	○	○	○	○
	1 output	○	○	○	○	○
	Toggle output	○	○	○	○	○
Input capture function		○	○	○	○	○
Synchronous operation		○	○	○	○	○
PWM mode		○	○	○	○	○
Phase counting mode		—	○	○	—	○
Buffer operation		○	—	—	○	—

PPG trigger	TGR0A/ TGR0B compare match or input capture	TGR1A/ TGR1B compare match or input capture	TGR2A/ TGR2B compare match or input capture	TGR3A/ TGR3B compare match or input capture	—	—
Interrupt sources	5 sources • Compare match or input capture 0A • Compare match or input capture 0B • Compare match or input capture 0C • Compare match or input capture 0D • Overflow	4 sources • Compare match or input capture 1A • Compare match or input capture 1B • Overflow • Underflow	4 sources • Compare match or input capture 2A • Compare match or input capture 2B • Overflow • Underflow	5 sources • Compare match or input capture 3A • Compare match or input capture 3B • Compare match or input capture 3C • Compare match or input capture 3D • Overflow	4 sources • Compare match or input capture 4A • Compare match or input capture 4B • Overflow • Underflow	4 s • C m i c • C m i c • C • U

Legend:

○ : Possible

— : Not possible



Legend:

TSTR:	Timer start register	TIOR (H, L):	Timer I/O control registers (H, L)
TSYR:	Timer synchro register	TIER:	Timer interrupt enable register
TCR:	Timer control register	TSR:	Timer status register
TMDR:	Timer mode register	TGR (A, B, C, D):	Timer general registers (A, B, C, D)

Figure 10.1 Block Diagram of TPU

				phase input)
	Clock input B	TCLKB	Input	External clock B input pin (Channels 1 and 5 phase counting phase input)
	Clock input C	TCLKC	Input	External clock C input pin (Channels 2 and 4 phase counting phase input)
	Clock input D	TCLKD	Input	External clock D input pin (Channels 2 and 4 phase counting phase input)
0	Input capture/out compare match A0	TIOCA0	I/O	TGR0A input capture input/output output/PWM output pin
	Input capture/out compare match B0	TIOCB0	I/O	TGR0B input capture input/output output/PWM output pin
	Input capture/out compare match C0	TIOCC0	I/O	TGR0C input capture input/output output/PWM output pin
	Input capture/out compare match D0	TIOCD0	I/O	TGR0D input capture input/output output/PWM output pin
1	Input capture/out compare match A1	TIOCA1	I/O	TGR1A input capture input/output output/PWM output pin
	Input capture/out compare match B1	TIOCB1	I/O	TGR1B input capture input/output output/PWM output pin
2	Input capture/out compare match A2	TIOCA2	I/O	TGR2A input capture input/output output/PWM output pin
	Input capture/out compare match B2	TIOCB2	I/O	TGR2B input capture input/output output/PWM output pin

	compare match D3			output/PWM output pin
4	Input capture/out compare match A4	TIOCA4	I/O	TGR4A input capture input/output output/PWM output pin
	Input capture/out compare match B4	TIOCB4	I/O	TGR4B input capture input/output output/PWM output pin
5	Input capture/out compare match A5	TIOCA5	I/O	TGR5A input capture input/output output/PWM output pin
	Input capture/out compare match B5	TIOCB5	I/O	TGR5B input capture input/output output/PWM output pin

	Timer mode register 0	TMDR0	R/W	H'C0	H
	Timer I/O control register 0H	TIOR0H	R/W	H'00	H
	Timer I/O control register 0L	TIOR0L	R/W	H'00	H
	Timer interrupt enable register 0	TIER0	R/W	H'40	H
	Timer status register 0	TSR0	R/(W)*2	H'C0	H
	Timer counter 0	TCNT0	R/W	H'0000	H
	Timer general register 0A	TGR0A	R/W	H'FFFF	H
	Timer general register 0B	TGR0B	R/W	H'FFFF	H
	Timer general register 0C	TGR0C	R/W	H'FFFF	H
	Timer general register 0D	TGR0D	R/W	H'FFFF	H
1	Timer control register 1	TCR1	R/W	H'00	H
	Timer mode register 1	TMDR1	R/W	H'C0	H
	Timer I/O control register 1	TIOR1	R/W	H'00	H
	Timer interrupt enable register 1	TIER1	R/W	H'40	H
	Timer status register 1	TSR1	R/(W)*2	H'C0	H
	Timer counter 1	TCNT1	R/W	H'0000	H
	Timer general register 1A	TGR1A	R/W	H'FFFF	H
	Timer general register 1B	TGR1B	R/W	H'FFFF	H
2	Timer control register 2	TCR2	R/W	H'00	H
	Timer mode register 2	TMDR2	R/W	H'C0	H
	Timer I/O control register 2	TIOR2	R/W	H'00	H
	Timer interrupt enable register 2	TIER2	R/W	H'40	H
	Timer status register 2	TSR2	R/(W)*2	H'C0	H
	Timer counter 2	TCNT2	R/W	H'0000	H
	Timer general register 2A	TGR2A	R/W	H'FFFF	H
	Timer general register 2B	TGR2B	R/W	H'FFFF	H

	Timer counter 3	TCNT3	R/W	H'0000	
	Timer general register 3A	TGR3A	R/W	H'FFFF	
	Timer general register 3B	TGR3B	R/W	H'FFFF	
	Timer general register 3C	TGR3C	R/W	H'FFFF	
	Timer general register 3D	TGR3D	R/W	H'FFFF	
4	Timer control register 4	TCR4	R/W	H'00	
	Timer mode register 4	TMDR4	R/W	H'C0	
	Timer I/O control register 4	TIOR4	R/W	H'00	
	Timer interrupt enable register 4	TIER4	R/W	H'40	
	Timer status register 4	TSR4	R/(W)*2	H'C0	
	Timer counter 4	TCNT4	R/W	H'0000	
	Timer general register 4A	TGR4A	R/W	H'FFFF	
	Timer general register 4B	TGR4B	R/W	H'FFFF	
5	Timer control register 5	TCR5	R/W	H'00	
	Timer mode register 5	TMDR5	R/W	H'C0	
	Timer I/O control register 5	TIOR5	R/W	H'00	
	Timer interrupt enable register 5	TIER5	R/W	H'40	
	Timer status register 5	TSR5	R/(W)*2	H'C0	
	Timer counter 5	TCNT5	R/W	H'0000	
	Timer general register 5A	TGR5A	R/W	H'FFFF	
	Timer general register 5B	TGR5B	R/W	H'FFFF	
All	Timer start register	TSTR	R/W	H'00	
	Timer synchro register	TSYR	R/W	H'00	
	Module stop control register A	MSTPCRA	R/W	H'3F	

Notes: 1. Lower 16 bits of the address.
2. Only 0 can be written, for flag clearing.

	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1
Initial value :	0	0	0	0	0	0	0
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Channel 1: TCR1

Channel 2: TCR2

Channel 4: TCR4

Channel 5: TCR5

Bit :	7	6	5	4	3	2	1
	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1
Initial value :	0	0	0	0	0	0	0
R/W :	—	R/W	R/W	R/W	R/W	R/W	R/W

The TCR registers are 8-bit registers that control the TCNT channels. The TPU has six registers, one for each of channels 0 to 5. The TCR registers are initialized to H'00 by a in hardware standby mode.

TCR register settings should be made only when TCNT operation is stopped.

		1	0	TCNT cleared by TGRB compare m capture
			1	TCNT cleared by counter clearing fo channel performing synchronous cle synchronous operation*1
	1	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRC compare m capture*2
		1	0	TCNT cleared by TGRD compare m capture*2
			1	TCNT cleared by counter clearing fo channel performing synchronous cle synchronous operation*1

Channel	Bit 7 Reserved*3	Bit 6 CCLR1	Bit 5 CCLR0	Description
1, 2, 4, 5	0	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRA compare m capture
		1	0	TCNT cleared by TGRB compare m capture
			1	TCNT cleared by counter clearing fo channel performing synchronous cle synchronous operation*1

Notes: 1. Synchronous operation setting is performed by setting the SYNC bit in TSY
2. When TGRC or TGRD is used as a buffer register, TCNT is not cleared bec buffer register setting has priority, and compare match/input capture does n
3. Bit 7 is reserved in channels 1, 2, 4, and 5. It is always read as 0 and canno modified.

	1	Count at falling edge
1	—	Count at both edges

Note: Internal clock edge selection is valid when the input clock is φ/4 or slower. This s ignored if the input clock is φ/1, or when overflow/underflow of another channel is

Bits 2 to 0—Time Prescaler 2 to 0 (TPSC2 to TPSC0): These bits select the TCNT c clock. The clock source can be selected independently for each channel. Table 10.4 sho clock sources that can be set for each channel.

Table 10.4 TPU Clock Sources

Channel	Internal Clock							External Clock			
	φ/1	φ/4	φ/16	φ/64	φ/256	φ/1024	φ/4096	TCLKA	TCLKB	TCLKC	TCLKD
0	○	○	○	○				○	○	○	○
1	○	○	○	○	○			○	○		
2	○	○	○	○		○		○	○	○	
3	○	○	○	○	○	○	○	○			
4	○	○	○	○		○		○		○	
5	○	○	○	○	○			○		○	○

Legend:

○ : Setting

Blank : No setting

		1		External clock: counts on TCLKB pi
		1	0	External clock: counts on TCLKC pi
			1	External clock: counts on TCLKD pi

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
1	0	0	0	Internal clock: counts on ϕ/1
			1	Internal clock: counts on ϕ/4
		1	0	Internal clock: counts on ϕ/16
			1	Internal clock: counts on ϕ/64
	1	0	0	External clock: counts on TCLKA pi
			1	External clock: counts on TCLKB pi
		1	0	Internal clock: counts on ϕ/256
			1	Counts on TCNT2 overflow/underflo

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

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

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

			1	Internal clock: counts on φ/1024
		1	0	Internal clock: counts on φ/256
			1	Internal clock: counts on φ/4096

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
4	0	0	0	Internal clock: counts on φ/1 (
			1	Internal clock: counts on φ/4
		1	0	Internal clock: counts on φ/16
			1	Internal clock: counts on φ/64
	1	0	0	External clock: counts on TCLKA pin
			1	External clock: counts on TCLKC pin
		1	0	Internal clock: counts on φ/1024
			1	Counts on TCNT5 overflow/underflow

Note: This setting is ignored when channel 4 is in phase counting mode.

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
5	0	0	0	Internal clock: counts on φ/1 (
			1	Internal clock: counts on φ/4
		1	0	Internal clock: counts on φ/16
			1	Internal clock: counts on φ/64
	1	0	0	External clock: counts on TCLKA pin
			1	External clock: counts on TCLKC pin
		1	0	Internal clock: counts on φ/256
			1	External clock: counts on TCLKD pin

Note: This setting is ignored when channel 5 is in phase counting mode.

R/W	:	—	—	R/W	R/W	R/W	R/W	R/W

Channel 1: TMDR1

Channel 2: TMDR2

Channel 4: TMDR4

Channel 5: TMDR5

Bit	:	7	6	5	4	3	2	1
		—	—	—	—	MD3	MD2	MD1
Initial value	:	1	1	0	0	0	0	0
R/W	:	—	—	—	—	R/W	R/W	R/W

The TMDR registers are 8-bit readable/writable registers that are used to set the opera for each channel. The TPU has six TMDR registers, one for each channel. The TMDR are initialized to H'C0 by a reset, and in hardware standby mode.

TMDR register settings should be made only when TCNT operation is stopped.

Bits 7 and 6—Reserved: These bits are always read as 1 and cannot be modified.

Bit 5—Buffer Operation B (BFB): Specifies whether TGRB is to operate in the norm TGRB and TGRD are to be used together for buffer operation. When TGRD is used a register, TGRD input capture/output compare is not generated.

In channels 1, 2, 4, and 5, which have no TGRD, bit 5 is reserved. It is always read as cannot be modified.

Bit 5 BFB	Description
0	TGRB operates normally
1	TGRB and TGRD used together for buffer operation

0	TGRA operates normally	(
1	TGRA and TGRC used together for buffer operation	

Bits 3 to 0—Modes 3 to 0 (MD3 to MD0): These bits are used to set the timer operati

Bit 3 MD3*[1]	Bit 2 MD2*[2]	Bit 1 MD1	Bit 0 MD0	Description	
0	0	0	0	Normal operation	(
			1	Reserved	
		1	0	PWM mode 1	
			1	PWM mode 2	
	1	0	0	Phase counting mode 1	
			1	Phase counting mode 2	
		1	0	Phase counting mode 3	
			1	Phase counting mode 4	
1	*	*	*	—	

*

Notes: 1. MD3 is a reserved bit. In a write, it should always be written with 0.
2. Phase counting mode cannot be set for channels 0 and 3. In this case, 0 sho be written to MD2.

Channel 5: TIOR5

Bit	:	7	6	5	4	3	2	1
		IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1
Initial value	:	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Channel 0: TIOR0L

Channel 3: TIOR3L

Bit	:	7	6	5	4	3	2	1
		IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1
Initial value	:	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: When TGRC or TGRD is designated for buffer operation, this setting is invalid a register operates as a buffer register.

The TIOR registers are 8-bit registers that control the TGR registers. The TPU has eig registers, two each for channels 0 and 3, and one each for channels 1, 2, 4, and 5. The registers are initialized to H'00 by a reset, and in hardware standby mode.

Care is required since TIOR is affected by the TMDR setting. The initial output speci TIOR is valid when the counter is stopped (the CST bit in TSTR is cleared to 0). Note PWM mode 2, the output at the point at which the counter is cleared to 0 is specified.

			1	compare register	Initial output is 0 output	0 output at compa
		1	0			1 output at compa
			1			Toggle output at c match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compa
		1	0			1 output at compa
			1			Toggle output at c match
1	0	0	0	TGR0B is input capture register	Capture input source is TIOCB0 pin	Input capture at ri
			1			Input capture at fa
		1	*			Input capture at b
	1	*	*		Capture input source is channel 1/count clock	Input capture at T count-up/count-do

*

			1		Initial output is 1 output	0 output at comp
		1	0			1 output at comp
			1			Toggle output at match
1	0	0	0	TGR0D is input capture register*2	Capture input source is TIOCD0 pin	Input capture at
			1			Input capture at
		1	*			Input capture at
	1	*	*		Capture input source is channel 1/count clock	Input capture at count-up/count-d

Notes: 1. When bits TPSC2 to TPSC0 in TCR1 are set to B'000 and ϕ/1 is used as th count clock, this setting is invalid and input capture is not generated.

2. When the BFB bit in TMDR0 is set to 1 and TGR0D is used as a buffer regi setting is invalid and input capture/output compare is not generated.

				1		Initial output is 1 output	0 output at compa
			1	0			1 output at compa
				1			Toggle output at c match
	1	0	0	0	TGR1B is input capture register	Capture input source is TIOCB1 pin	Input capture at ri
				1			Input capture at fa
			1	*			Input capture at b
		1	*	*		Capture input source is TGR0C compare match/ input capture	Input capture at g of TGR0C compa input capture

*

Channel	Bit 7 IOB3	Bit 6 IOB2	Bit 5 IOB1	Bit 4 IOB0	Description		
2	0	0	0	0	TGR2B is output compare register	Output disabled	(I
				1		Initial output is 0 output	0 output at compa
			1	0			1 output at compa
				1			Toggle output at c match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compa
			1	0			1 output at compa
				1			Toggle output at c match
	1	*	0	0	TGR2B is input capture register	Capture input source is TIOCB2 pin	Input capture at ri
				1			Input capture at fa
			1	*			Input capture at b

*

	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at comp
		1	0			1 output at comp
			1			Toggle output at match
1	0	0	0	TGR3B is input capture register	Capture input source is TIOCB3 pin	Input capture at
			1			Input capture at
		1	*			Input capture at
	1	*	*		Capture input source is channel 4/count clock	Input capture at count-up/count-d

			1		Initial output is 1 output	0 output at compa
		1	0			1 output at compa
			1			Toggle output at c match
1	0	0	0	TGR3D is input capture register*2	Capture input source is TIOCD3 pin	Input capture at ri
			1			Input capture at fa
		1	*			Input capture at b
	1	*	*		Capture input source is channel 4/count clock	Input capture at T count-up/count-do

*

Notes: 1. When bits TPSC2 to TPSC0 in TCR4 are set to B'000 and φ/1 is used as the count clock, this setting is invalid and input capture is not generated.

2. When the BFB bit in TMDR3 is set to 1 and TGR3D is used as a buffer regis setting is invalid and input capture/output compare is not generated.

				1		Initial output is 1 output	0 output at comp
			1	0			1 output at comp
				1			Toggle output at match
	1	0	0	0	TGR4B is input capture register	Capture input source is TIOCB4 pin	Input capture at
				1			Input capture at
			1	*			Input capture at
		1	*	*		Capture input source is TGR3C compare match/ input capture	Input capture at of TGR3C comp input capture

Channel	Bit 7 IOB3	Bit 6 IOB2	Bit 5 IOB1	Bit 4 IOB0	Description		
5	0	0	0	0	TGR5B is output compare register	Output disabled	(
				1		Initial output is 0 output	0 output at comp
			1	0			1 output at comp
				1			Toggle output at match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at comp
			1	0			1 output at comp
				1			Toggle output at match
	1	*	0	0	TGR5B is input capture register	Capture input source is TIOCB5 pin	Input capture at
				1			Input capture at
			1	*			Input capture at

			1	compare register	Initial output is 0 output	0 output at compa
		1	0			1 output at compa
			1			Toggle output at c match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compa
		1	0			1 output at compa
			1			Toggle output at c match
1	0	0	0	TGR0A is input capture register	Capture input source is TIOCA0 pin	Input capture at ri
			1			Input capture at fa
		1	*			Input capture at b
	1	*	*		Capture input source is channel 1/count clock	Input capture at T count-up/count-do

*

	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at comp
		1	0			1 output at comp
			1			Toggle output at match
1	0	0	0	TGR0C is input capture register*1	Capture input source is TIOCC0 pin	Input capture at
			1			Input capture at
		1	*			Input capture at
	1	*	*		Capture input source is channel 1/count clock	Input capture at count-up/count-d

Note: 1. When the BFA bit in TMDR0 is set to 1 and TGR0C is used as a buffer regi setting is invalid and input capture/output compare is not generated.

		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compa
			1	0			1 output at compa
				1			Toggle output at c match
	1	0	0	0	TGR1A is input capture register	Capture input source is TIOCA1 pin	Input capture at ri
				1			Input capture at fa
			1	*			Input capture at b
		1	*	*		Capture input source is TGR0A compare match/ input capture	Input capture at g of channel 0/TGR compare match/in capture

*

Channel	Bit 3 IOA3	Bit 2 IOA2	Bit 1 IOA1	Bit 0 IOA0	Description		
2	0	0	0	0	TGR2A is output compare register	Output disabled	(I
				1		Initial output is 0 output	0 output at compa
			1	0			1 output at compa
				1			Toggle output at c match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compa
			1	0			1 output at compa
				1			Toggle output at c match
	1	*	0	0	TGR2A is input capture register	Capture input source is TIOCA2 pin	Input capture at ri
				1			Input capture at fa
			1	*			Input capture at b

*

	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at comp
		1	0			1 output at comp
			1			Toggle output at match
1	0	0	0	TGR3A is input capture register	Capture input source is TIOCA3 pin	Input capture at
			1			Input capture at
		1	*			Input capture at
	1	*	*		Capture input source is channel 4/count clock	Input capture at count-up/count-d

	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compa
		1	0			1 output at compa
			1			Toggle output at c match
1	0	0	0	TGR3C is input capture register*1	Capture input source is TIOCC3 pin	Input capture at ri
			1			Input capture at fa
		1	*			Input capture at b
	1	*	*		Capture input source is channel 4/count clock	Input capture at T count-up/count-do

*

Note: 1. When the BFA bit in TMDR3 is set to 1 and TGR3C is used as a buffer regis setting is invalid and input capture/output compare is not generated.

		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at comp
			1	0			1 output at comp
				1			Toggle output at match
	1	0	0	0	TGR4A is input capture register	Capture input source is TIOCA4 pin	Input capture at
				1			Input capture at
			1	*			Input capture at
		1	*	*		Capture input source is TGR3A compare match/ input capture	Input capture at of TGR3A comp input capture

Channel	Bit 3 IOA3	Bit 2 IOA2	Bit 1 IOA1	Bit 0 IOA0	Description		
5	0	0	0	0	TGR5A is output compare register	Output disabled	
				1		Initial output is 0 output	0 output at comp
			1	0			1 output at comp
				1			Toggle output at match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at comp
			1	0			1 output at comp
				1			Toggle output at match
	1	*	0	0	TGR5A is input capture register	Capture input source is TIOCA5 pin	Input capture at
				1			Input capture at
			1	*			Input capture at

Channel 1: TIER1
Channel 2: TIER2
Channel 4: TIER4
Channel 5: TIER5

Bit	:	7	6	5	4	3	2	1
		TTGE	—	TCIEU	TCIEV	—	—	TGIEB
Initial value	:	0	1	0	0	0	0	0
R/W	:	R/W	—	R/W	R/W	—	—	R/W

The TIER registers are 8-bit registers that control enabling or disabling of interrupt requ
each channel. The TPU has six TIER registers, one for each channel. The TIER register
initialized to H'40 by a reset, and in hardware standby mode.

Bit 7—A/D Conversion Start Request Enable (TTGE): Enables or disables generatio
conversion start requests by TGRA input capture/compare match.

Bit 7 TTGE	Description	
0	A/D conversion start request generation disabled	(I
1	A/D conversion start request generation enabled	

Bit 6—Reserved: This bit is always read as 1 and cannot be modified.

Bit 4—Overflow Interrupt Enable (TCIEV): Enables or disables interrupt requests
the TCFV flag when the TCFV flag in TSR is set to 1.

Bit 4 TCIEV	Description
0	Interrupt requests (TCIV) by TCFV disabled
1	Interrupt requests (TCIV) by TCFV enabled

Bit 3—TGR Interrupt Enable D (TGIED): Enables or disables interrupt requests (T
TGFD bit when the TGFD bit in TSR is set to 1 in channels 0 and 3.

In channels 1, 2, 4, and 5, bit 3 is reserved. It is always read as 0 and cannot be modifi

Bit 3 TGIED	Description
0	Interrupt requests (TGID) by TGFD bit disabled
1	Interrupt requests (TGID) by TGFD bit enabled

Bit 2—TGR Interrupt Enable C (TGIEC): Enables or disables interrupt requests (T
TGFC bit when the TGFC bit in TSR is set to 1 in channels 0 and 3.

In channels 1, 2, 4, and 5, bit 2 is reserved. It is always read as 0 and cannot be modifi

Bit 2 TGIEC	Description
0	Interrupt requests (TGIC) by TGFC bit disabled
1	Interrupt requests (TGIC) by TGFC bit enabled

Bit 0—TGR Interrupt Enable A (TGIEA): Enables or disables interrupt requests (TG
TGFA bit when the TGFA bit in TSR is set to 1.

Bit 0 TGIEA	Description	
0	Interrupt requests (TGIA) by TGFA bit disabled	(
1	Interrupt requests (TGIA) by TGFA bit enabled	

10.2.5 Timer Status Register (TSR)

Channel 0: TSR0
Channel 3: TSR3

Bit	:	7	6	5	4	3	2	1
		—	—	—	TCFV	TGFD	TGFC	TGFB
Initial value	:	1	1	0	0	0	0	0
R/W	:	—	—	—	R/(W)*	R/(W)*	R/(W)*	R/(W)*

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

Channel 1: TSR1
Channel 2: TSR2
Channel 4: TSR4
Channel 5: TSR5

Bit	:	7	6	5	4	3	2	1
		TCFD	—	TCFU	TCFV	—	—	TGFB
Initial value	:	1	1	0	0	0	0	0
R/W	:	R	—	R/(W)*	R/(W)*	—	—	R/(W)*

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

Bit 7 TCFD	Description
0	TCNT counts down
1	TCNT counts up

Bit 6—Reserved: This bit is always read as 1 and cannot be modified.

Bit 5—Underflow Flag (TCFU): Status flag that indicates that TCNT underflow has
when channels 1, 2, 4, and 5 are set to phase counting mode.

In channels 0 and 3, bit 5 is reserved. It is always read as 0 and cannot be modified.

Bit 5 TCFU	Description
0	[Clearing condition] When 0 is written to TCFU after reading TCFU = 1
1	[Setting condition] When the TCNT value underflows (changes from H'0000 to H'FFFF)

Bit 4—Overflow Flag (TCFV): Status flag that indicates that TCNT overflow has oc

Bit 4 TCFV	Description
0	[Clearing condition] When 0 is written to TCFV after reading TCFV = 1
1	[Setting condition] When the TCNT value overflows (changes from H'FFFF to H'0000)

	• When 0 is written to TGFD after reading TGFD = 1
1	[Setting conditions] • When TCNT = TGRD while TGRD is functioning as output compare reg • When TCNT value is transferred to TGRD by input capture signal while functioning as input capture register

Bit 2—Input Capture/Output Compare Flag C (TGFC): Status flag that indicates th occurrence of TGRC input capture or compare match in channels 0 and 3.

In channels 1, 2, 4, and 5, bit 2 is reserved. It is always read as 0 and cannot be modifie

Bit 2 TGFC	Description
0	[Clearing conditions] (• When DTC is activated by TGIC interrupt while DISEL bit of MRB in DT • When 0 is written to TGFC after reading TGFC = 1
1	[Setting conditions] • When TCNT = TGRC while TGRC is functioning as output compare reg • When TCNT value is transferred to TGRC by input capture signal while functioning as input capture register

1	[Setting conditions] • When TCNT = TGRB while TGRB is functioning as output compare re • When TCNT value is transferred to TGRB by input capture signal while functioning as input capture register

Bit 0—Input Capture/Output Compare Flag A (TGFA): Status flag that indicates t occurrence of TGRA input capture or compare match.

Bit 0 TGFA	Description
0	[Clearing conditions] • When DTC is activated by TGIA interrupt while DISEL bit of MRB in D • When 0 is written to TGFA after reading TGFA = 1
1	[Setting conditions] • When TCNT = TGRA while TGRA is functioning as output compare re • When TCNT value is transferred to TGRA by input capture signal while functioning as input capture register

Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/

Note: * These counters can be used as up/down-counters only in phase counting m
when counting overflow/underflow on another channel. In other cases they f
up-counters.

The TCNT registers are 16-bit counters. The TPU has six TCNT counters, one for each

The TCNT counters are initialized to H'0000 by a reset, and in hardware standby mode

The TCNT counters cannot be accessed in 8-bit units; they must always be accessed as
unit.

10.2.7 Timer General Register (TGR)

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2
Initial value :	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/

The TGR registers are 16-bit registers with a dual function as output compare and inpu
registers. The TPU has 16 TGR registers, four each for channels 0 and 3 and two each
1, 2, 4, and 5. TGRC and TGRD for channels 0 and 3 can also be designated for operat
buffer registers*. The TGR registers are initialized to H'FFFF by a reset, and in hardwa
mode.

The TGR registers cannot be accessed in 8-bit units; they must always be accessed as a

Note: * TGR buffer register combinations are TGRA—TGRC and TGRB—TGRD.

TSTR is initialized to H'00 by a reset, and in hardware standby mode. When setting th
mode in TMDR or setting the count clock in TCR, first stop the TCNT counter.

Bits 7 and 6—Reserved: Should always be written with 0.

Bits 5 to 0—Counter Start 5 to 0 (CST5 to CST0): These bits select operation or sto
TCNT.

Bit n CSTn	Description
0	TCNTn count operation is stopped
1	TCNTn performs count operation

Note: If 0 is written to the CST bit during operation with the TIOC pin designated for o
counter stops but the TIOC pin output compare output level is retained. If TIOR
when the CST bit is cleared to 0, the pin output level will be changed to the set
value.

10.2.9 Timer Synchro Register (TSYR)

Bit	:	7	6	5	4	3	2	1
		—	—	SYNC5	SYNC4	SYNC3	SYNC2	SYNC
Initial value	:	0	0	0	0	0	0	0
R/W	:	—	—	R/W	R/W	R/W	R/W	R/W

TSYR is an 8-bit readable/writable register that selects independent operation or synch
operation for the channel 0 to 4 TCNT counters. A channel performs synchronous ope
the corresponding bit in TSYR is set to 1.

TSYR is initialized to H'00 by a reset, and in hardware standby mode.

Bits 7 and 6—Reserved: Should always be written with 0.

Bit n SYNCn	Description
0	TCNTn operates independently (TCNT presetting/clearing is unrelated to other channels)
1	TCNTn performs synchronous operation TCNT synchronous presetting/synchronous clearing is possible

10.2.10 Module Stop Control Register A (MSTPCRA)

Bit	:	7	6	5	4	3	2	1
		MSTPA7	MSTPA6	MSTPA5	MSTPA4	MSTPA3	MSTPA2	MSTPA
Initial value	:	0	0	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MSTPCRA is an 8-bit readable/writable register that performs module stop mode contr

When the MSTPA5 bit in MSTPCRA is set to 1, TPU operation stops at the end of the and a transition is made to module stop mode. Registers cannot be read or written to in stop mode. For details, see sections 21A.5, 21B.5, Module Stop Mode.

MSTPCRA is initialized to H'3F by a reset and in hardware standby mode. It is not init software standby mode.

Bit 5—Module Stop (MSTPA5): Specifies the TPU module stop mode.

Bit 5 MSTPA5	Description
0	TPU module stop mode cleared
1	TPU module stop mode set

An example of 16-bit register access operation is shown in figure 10.2.

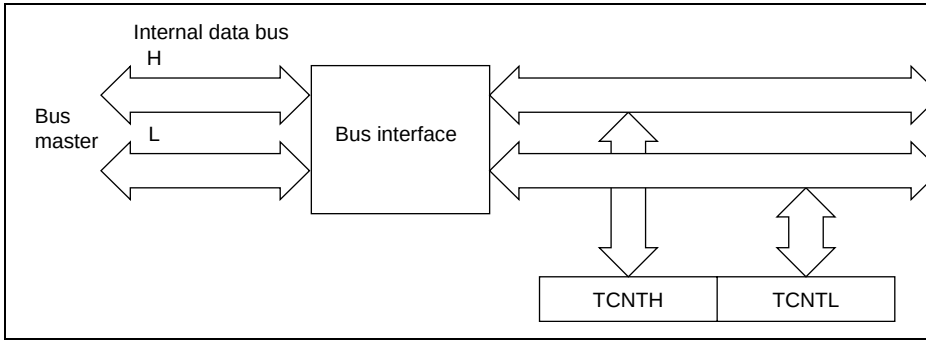


Figure 10.2 16-Bit Register Access Operation [Bus Master ↔ TCNT (16

10.3.2 8-Bit Registers

Registers other than TCNT and TGR are 8-bit. As the data bus to the CPU is 16 bits w
registers can be read and written to in 16-bit units. They can also be read and written t
units.

Examples of 8-bit register access operation are shown in figures 10.3 to 10.5.

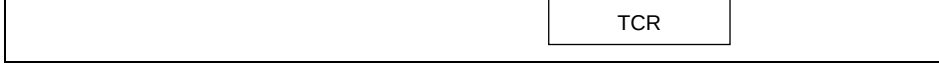


Figure 10.3 8-Bit Register Access Operation [Bus Master ↔ TCR (Upper 8

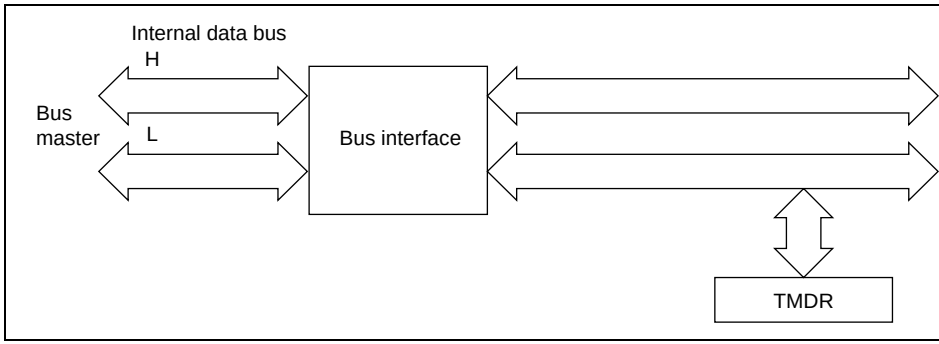


Figure 10.4 8-Bit Register Access Operation [Bus Master ↔ TMDR (Lower 8

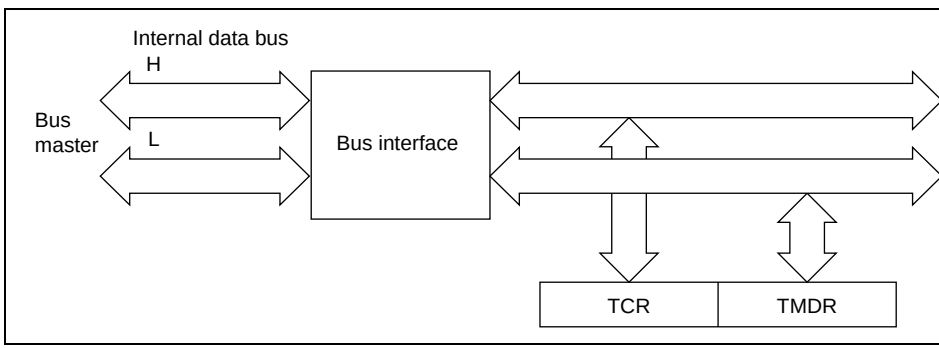


Figure 10.5 8-Bit Register Access Operation [Bus Master ↔ TCR and TMDR (

Each TGR can be used as an input capture register or output compare register.

Synchronous Operation: When synchronous operation is designated for a channel, T
that channel performs synchronous presetting. That is, when TCNT for a channel desi
synchronous operation is rewritten, the TCNT counters for the other channels are also
the same time. Synchronous clearing of the TCNT counters is also possible by setting
synchronization bits in TSYR for channels designated for synchronous operation.

Buffer Operation

- When TGR is an output compare register

  When a compare match occurs, the value in the buffer register for the relevant cha
  transferred to TGR.

- When TGR is an input capture register

  When input capture occurs, the value in TCNT is transfer to TGR and the value pr
  held in TGR is transferred to the buffer register.

Cascaded Operation: The channel 1 counter (TCNT1), channel 2 counter (TCNT2),
counter (TCNT4), and channel 5 counter (TCNT5) can be connected together to opera
bit counter.

PWM Mode: In this mode, a PWM waveform is output. The output level can be set b
TIOR. A PWM waveform with a duty of between 0% and 100% can be output, accor
setting of each TGR register.

Phase Counting Mode: In this mode, TCNT is incremented or decremented by detec
phases of two clocks input from the external clock input pins in channels 1, 2, 4, and 5
phase counting mode is set, the corresponding TCLK pin functions as the clock pin, a
performs up- or down-counting.

This can be used for two-phase encoder pulse input.

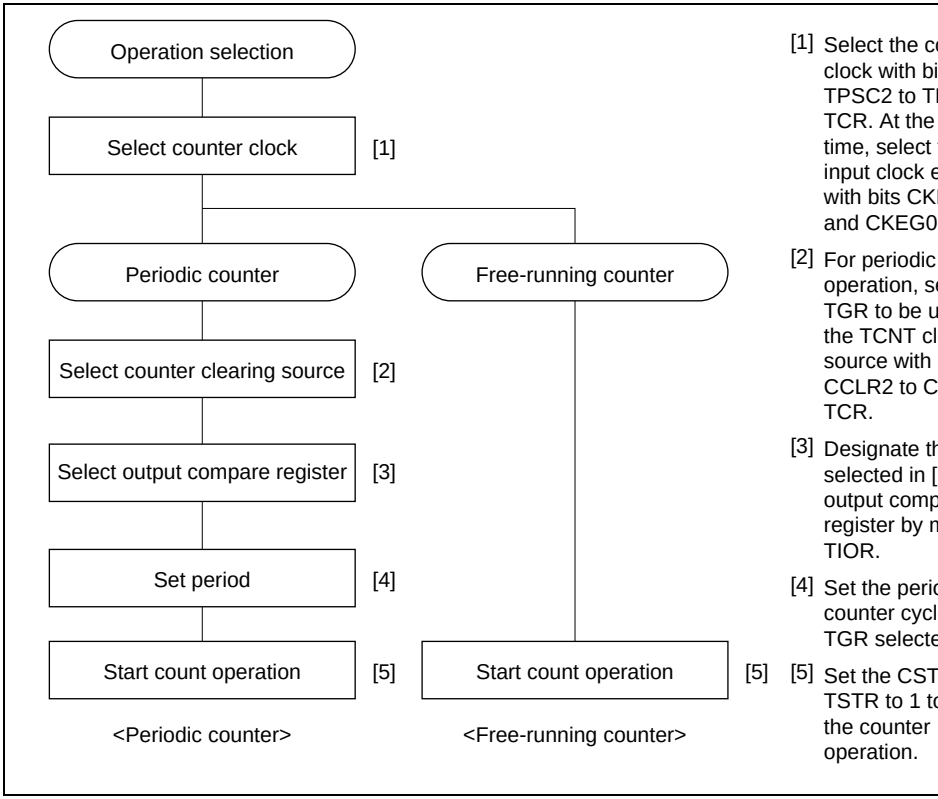


Figure 10.6 Example of Counter Operation Setting Procedure

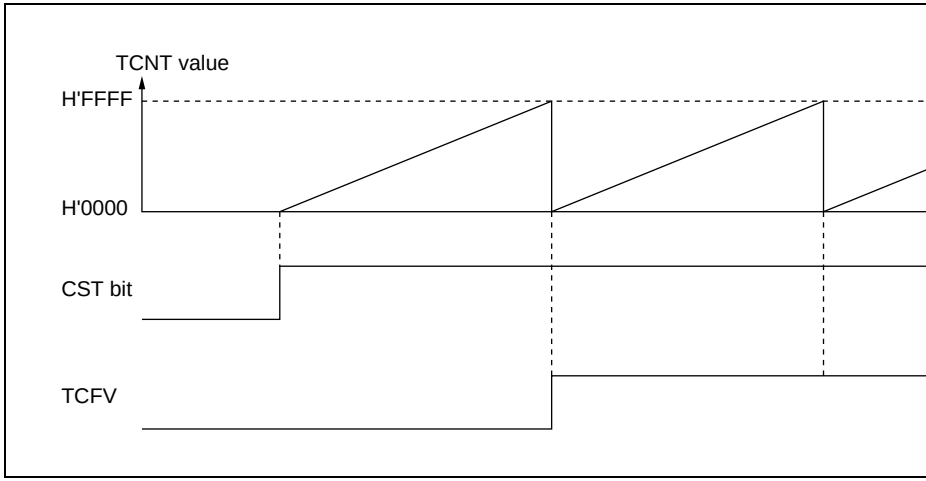


Figure 10.7 Free-Running Counter Operation

When compare match is selected as the TCNT clearing source, the TCNT counter
relevant channel performs periodic count operation. The TGR register for setting t
designated as an output compare register, and counter clearing by compare match
by means of bits CCLR2 to CCLR0 in TCR. After the settings have been made, T
up-count operation as periodic counter when the corresponding bit in TSTR is set
the count value matches the value in TGR, the TGF bit in TSR is set to 1 and TCN
to H'0000.

If the value of the corresponding TGIE bit in TIER is 1 at this point, the TPU requ
interrupt. After a compare match, TCNT starts counting up again from H'0000.

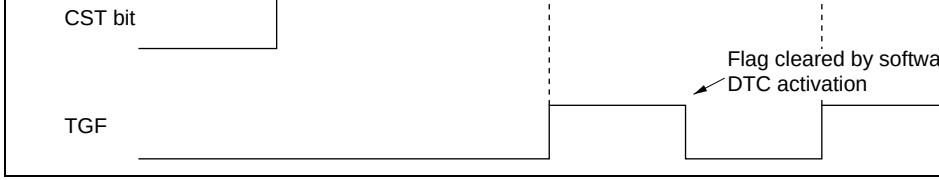


Figure 10.8 Periodic Counter Operation

Waveform Output by Compare Match: The TPU can perform 0, 1, or toggle output corresponding output pin using compare match.

- Example of setting procedure for waveform output by compare match
  Figure 10.9 shows an example of the setting procedure for waveform output by com

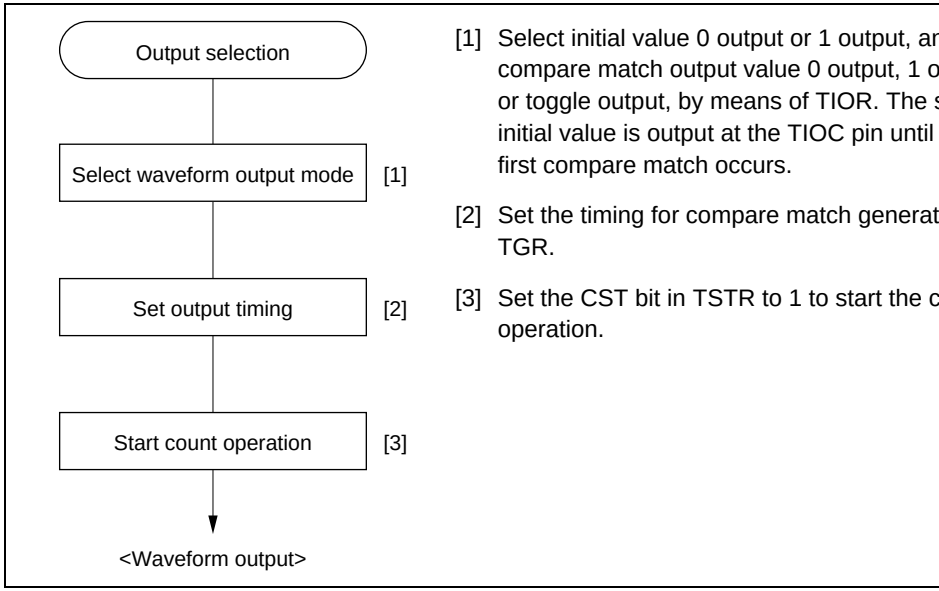


Figure 10.9 Example of Setting Procedure for Waveform Output by Compare

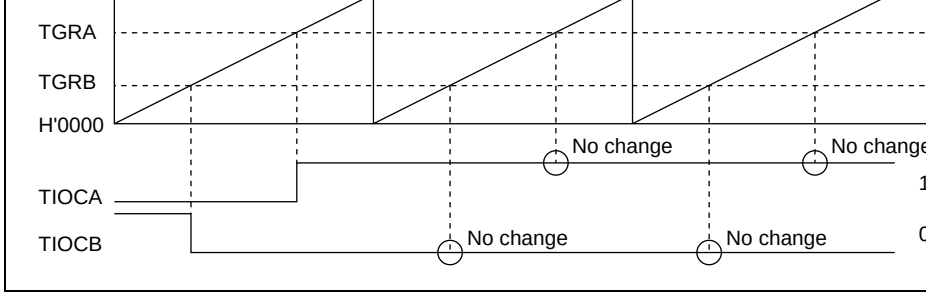


Figure 10.10 Example of 0 Output/1 Output Operation

Figure 10.11 shows an example of toggle output.

In this example TCNT has been designated as a periodic counter (with counter cle performed by compare match B), and settings have been made so that output is tog compare match A and compare match B.

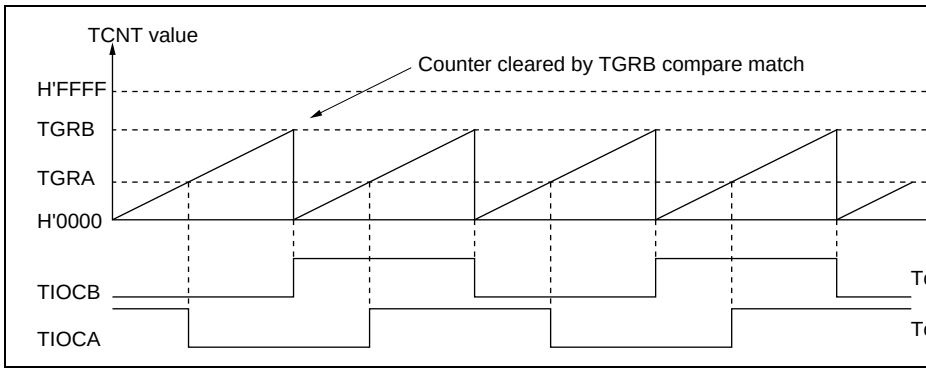


Figure 10.11 Example of Toggle Output Operation

Input capture will not be generated if φ/1 is selected.

- Example of input capture operation setting procedure
  Figure 10.12 shows an example of the input capture operation setting procedure.

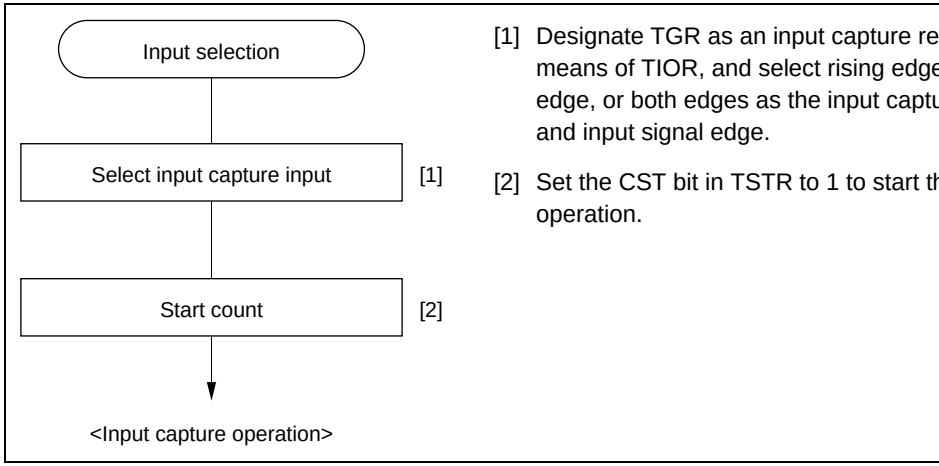


Figure 10.12 Example of Input Capture Operation Setting Procedure

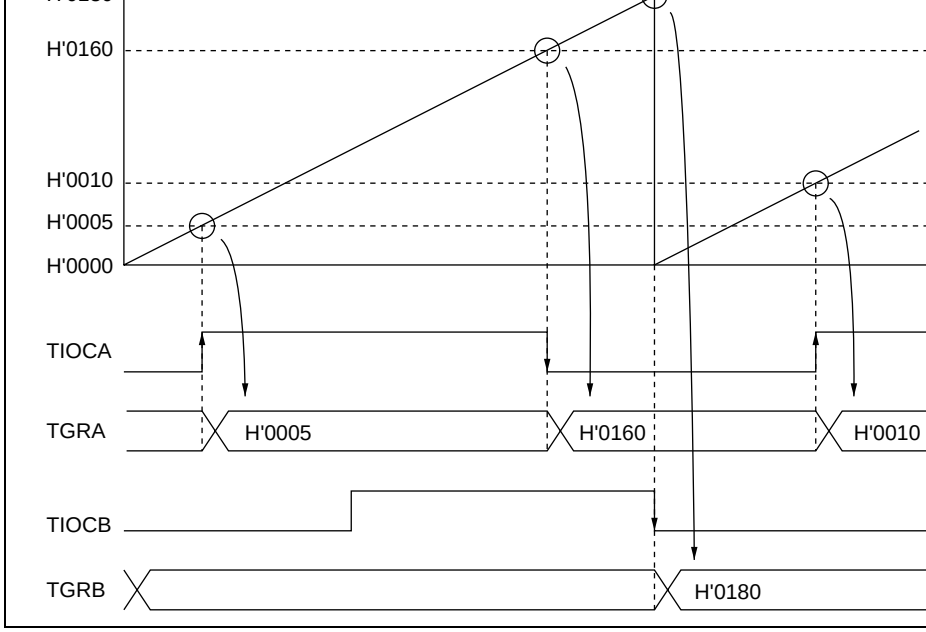


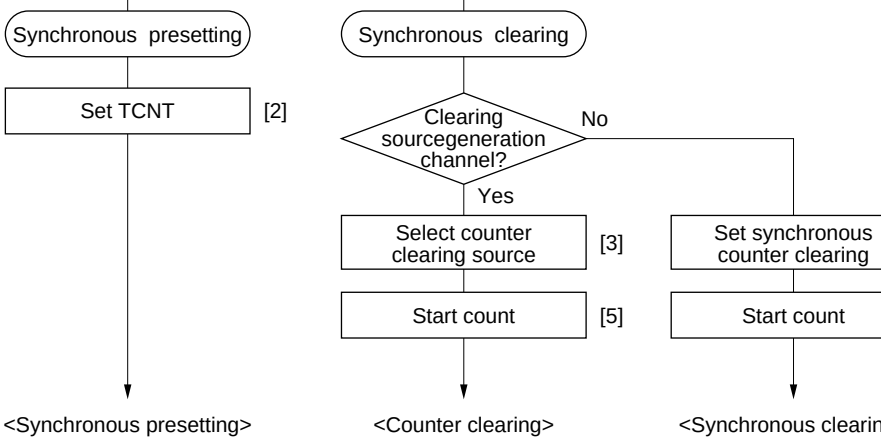
Figure 10.13 Example of Input Capture Operation

10.4.3 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 cl simultaneously by making the appropriate setting in TCR (synchronous clearing).

Synchronous operation enables TGR to be incremented with respect to a single time b

Channels 0 to 5 can all be designated for synchronous operation.



[1] Set to 1 the SYNC bits in TSYR corresponding to the channels to be designated for synchr operation.

[2] When the TCNT counter of any of the channels designated for synchronous operation is written to, the same value is simultaneously written to the other TCNT counters.

[3] Use bits CCLR2 to CCLR0 in TCR to specify TCNT clearing by input capture/output com etc.

[4] Use bits CCLR2 to CCLR0 in TCR to designate synchronous clearing for the counter cle source.

[5] Set to 1 the CST bits in TSTR for the relevant channels, to start the count operation.

Figure 10.14 Example of Synchronous Operation Setting Procedure

For details of PWM modes, see section 10.4.6, PWM Modes.

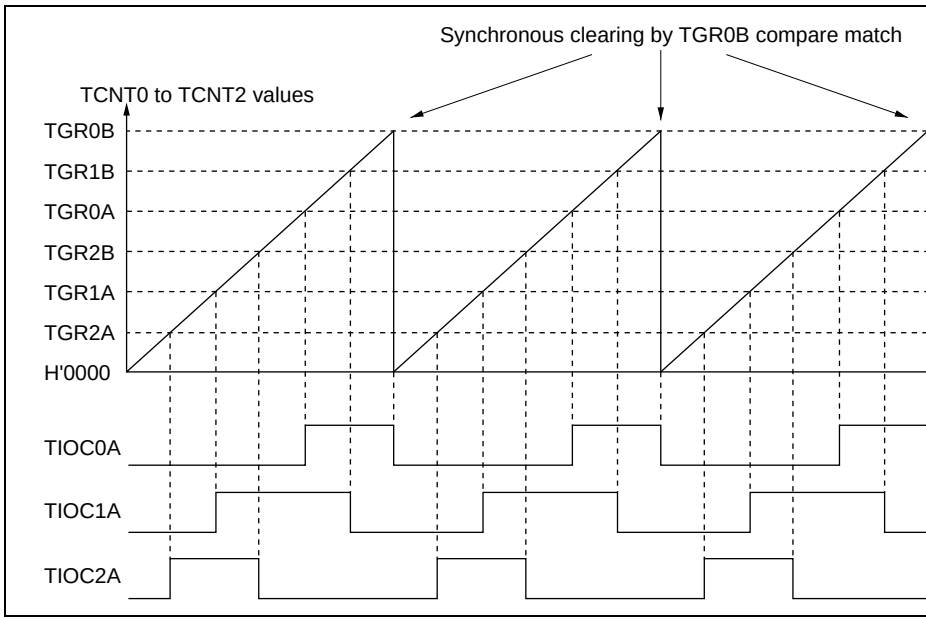


Figure 10.15 Example of Synchronous Operation

Table 10.5 Register Combinations in Buffer Operation

Channel	Timer General Register	Buffer Register
0	TGR0A	TGR0C
	TGR0B	TGR0D
3	TGR3A	TGR3C
	TGR3B	TGR3D

- When TGR is an output compare register

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

  This operation is illustrated in figure 10.16.

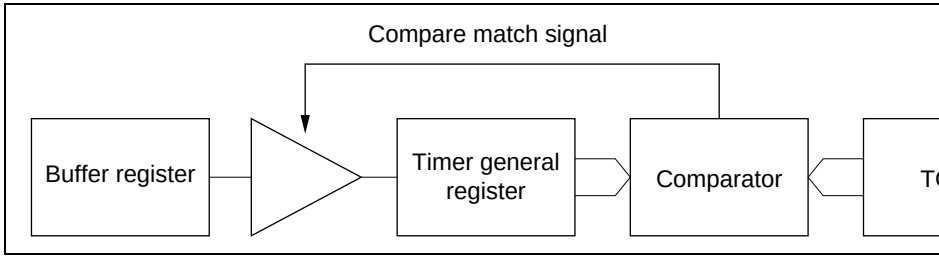


Figure 10.16 Compare Match Buffer Operation

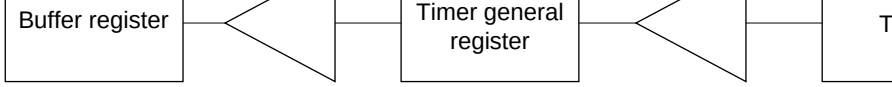
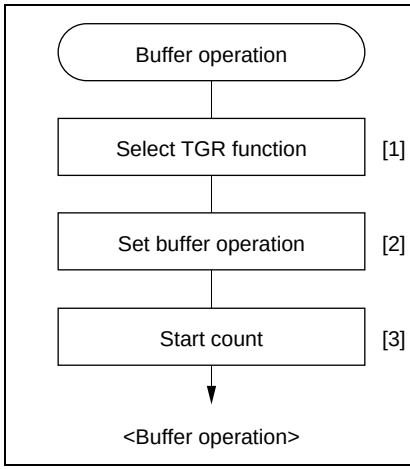


Figure 10.17 Input Capture Buffer Operation

Example of Buffer Operation Setting Procedure: Figure 10.18 shows an example o operation setting procedure.



[1] Designate TGR as an input capture regis output compare register by means of TIO

[2] Designate TGR for buffer operation with BFA and BFB in TMDR.

[3] Set the CST bit in TSTR to 1 to start the operation.

Figure 10.18 Example of Buffer Operation Setting Procedure

value in buffer register TGRC is simultaneously transferred to timer general registe
This operation is repeated each time compare match A occurs.

For details of PWM modes, see section 10.4.6, PWM Modes.

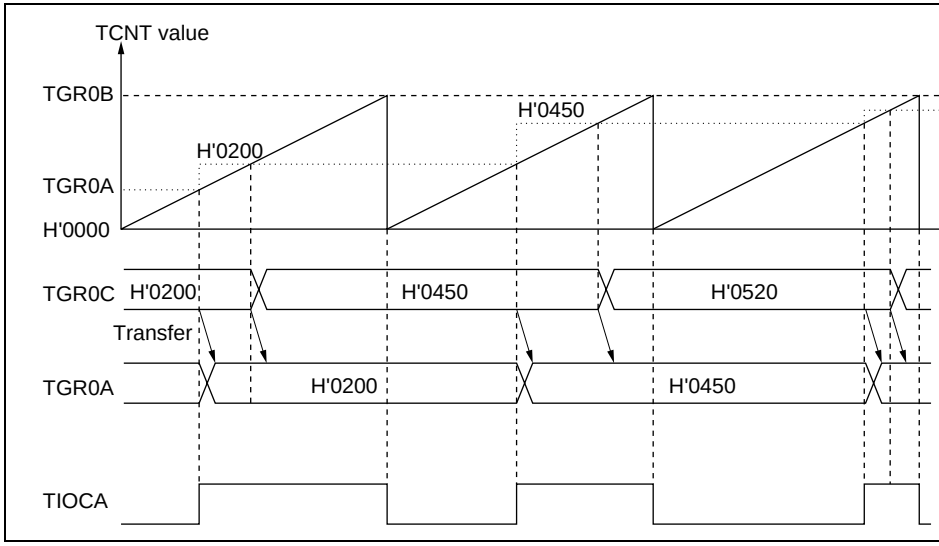


Figure 10.19 Example of Buffer Operation (1)

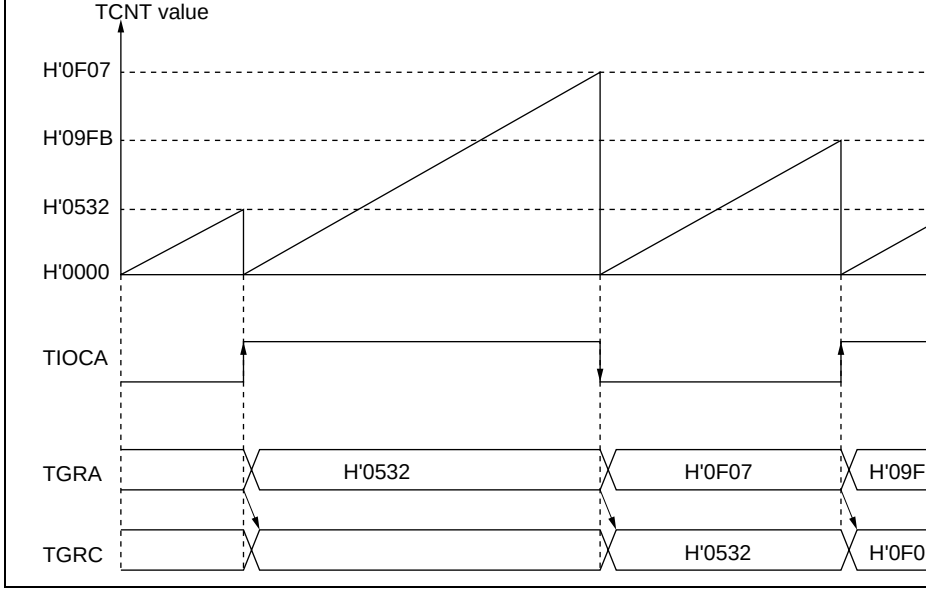


Figure 10.20 Example of Buffer Operation (2)

Underflow occurs only when the lower 16-bit TCNT is in phase-counting mode.

Table 10.6 shows the register combinations used in cascaded operation.

Note: When phase counting mode is set for channel 1 or 4, the counter clock setting i
and the counter operates independently in phase counting mode.

Table 10.6 Cascaded Combinations

Combination	Upper 16 Bits	Lower 16 Bits
Channels 1 and 2	TCNT1	TCNT2
Channels 4 and 5	TCNT4	TCNT5

Example of Cascaded Operation Setting Procedure: Figure 10.21 shows an example
setting procedure for cascaded operation.

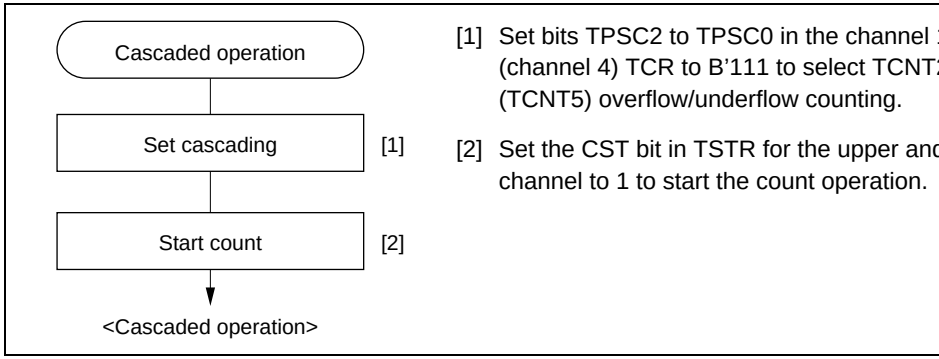


Figure 10.21 Cascaded Operation Setting Procedure

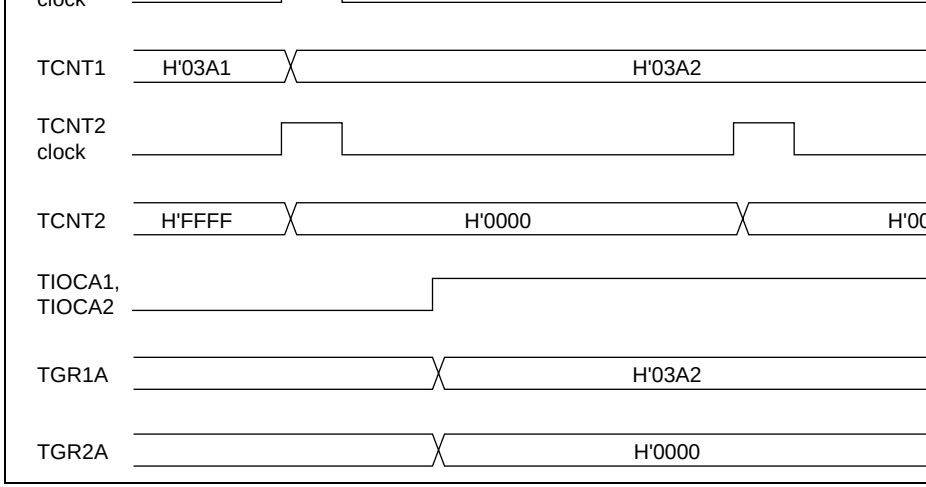


Figure 10.22 Example of Cascaded Operation (1)

Figure 10.23 illustrates the operation when counting upon TCNT2 overflow/underflow set for TCNT1, and phase counting mode has been designated for channel 2.

TCNT1 is incremented by TCNT2 overflow and decremented by TCNT2 underflow.

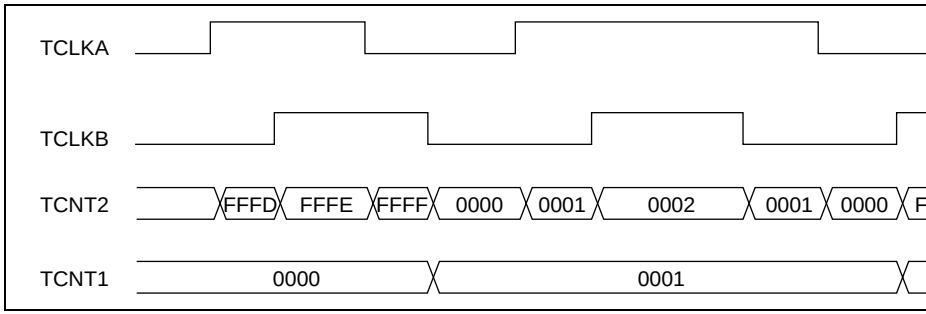


Figure 10.23 Example of Cascaded Operation (2)

There are two PWM modes, as described below.

- PWM mode 1

  PWM output is generated from the TIOCA and TIOCC pins by pairing TGRA with TGRC with TGRD. The output specified by bits IOA3 to IOA0 and IOC3 to IOC0 output from the TIOCA and TIOCC pins at compare matches A and C, and the outp specified by bits IOB3 to IOB0 and IOD3 to IOD0 in TIOR is output at compare m and D. The initial output value is the value set in TGRA or TGRC. If the set values TGRs are identical, the output value does not change when a compare match occurs

  In PWM mode 1, a maximum 8-phase PWM output is possible.

- PWM mode 2

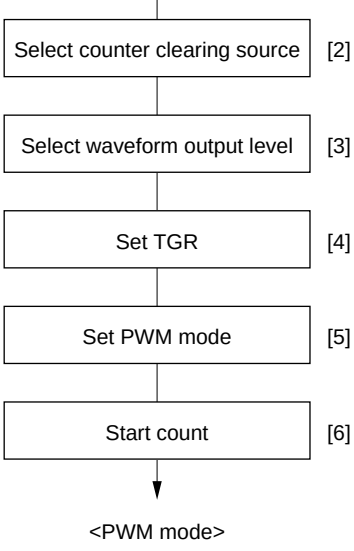
  PWM output is generated using one TGR as the cycle register and the others as duty The output specified in TIOR is performed by means of compare matches. Upon co clearing by a synchronization register compare match, the output value of each pin value set in TIOR. If the set values of the cycle and duty registers are identical, the value does not change when a compare match occurs.

  In PWM mode 2, a maximum 15-phase PWM output is possible by combined use w synchronous operation.

The correspondence between PWM output pins and registers is shown in table 10.7.

1	TGR1A	TIOCA1	TIOCA1
	TGR1B		TIOCB1
2	TGR2A	TIOCA2	TIOCA2
	TGR2B		TIOCB2
3	TGR3A	TIOCA3	TIOCA3
	TGR3B		TIOCB3
	TGR3C	TIOCC3	TIOCC3
	TGR3D		TIOCD3
4	TGR4A	TIOCA4	TIOCA4
	TGR4B		TIOCB4
5	TGR5A	TIOCA5	TIOCA5
	TGR5B		TIOCB5

Note: In PWM mode 2, PWM output is not possible for the TGR register in which the



[2] TGR to be used as the TCNT clearing sou

[3] Use TIOR to designate the TGR as an ou compare register, and select the initial val output value.

[4] Set the cycle in the TGR selected in [2], a the duty in the other the TGR.

[5] Select the PWM mode with bits MD3 to M TMDR.

[6] Set the CST bit in TSTR to 1 to start the c operation.

Figure 10.24 Example of PWM Mode Setting Procedure

Examples of PWM Mode Operation: Figure 10.25 shows an example of PWM mode operation.

In this example, TGRA compare match is set as the TCNT clearing source, 0 is set for t initial output value and output value, and 1 is set as the TGRB output value.

In this case, the value set in TGRA is used as the period, and the values set in TGRB re the duty.

Figure 10.25 Example of PWM Mode Operation (1)

Figure 10.26 shows an example of PWM mode 2 operation.

In this example, synchronous operation is designated for channels 0 and 1, TGR1B co
is set as the TCNT clearing source, and 0 is set for the initial output value and 1 for th
value of the other TGR registers (TGR0A to TGR0D, TGR1A), to output a 5-phase P
waveform.

In this case, the value set in TGR1B is used as the cycle, and the values set in the othe
the duty.

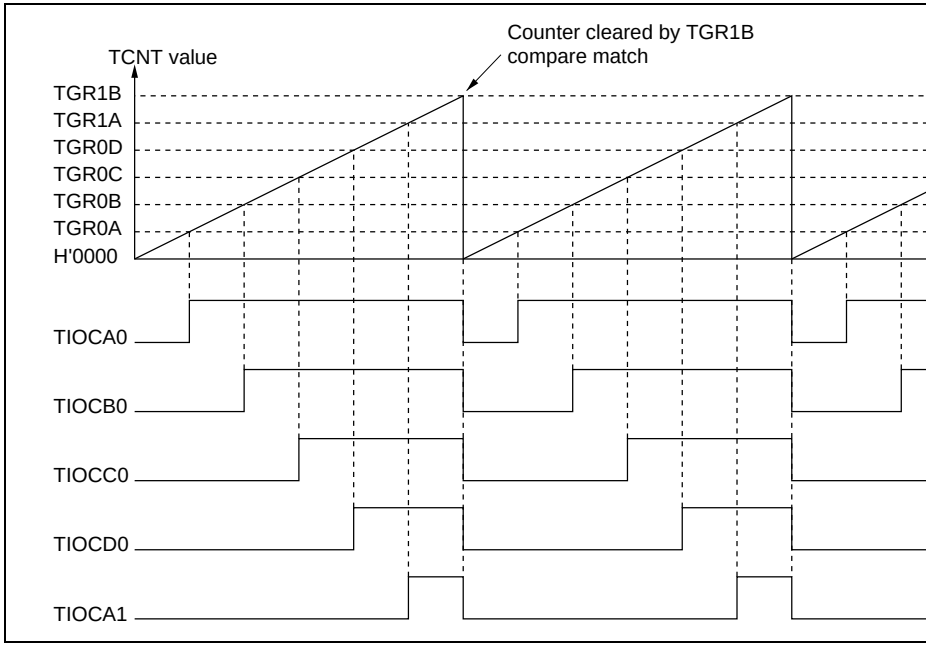


Figure 10.26 Example of PWM Mode Operation (2)

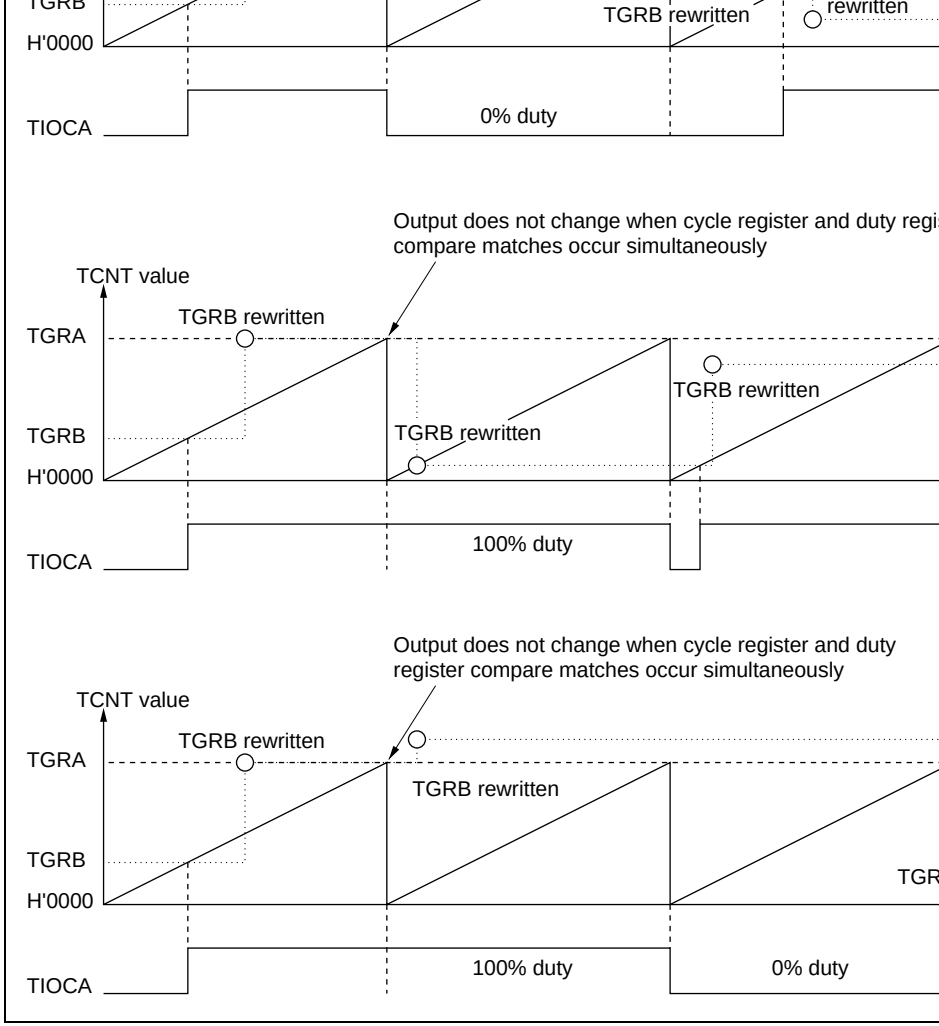


Figure 10.27 Example of PWM Mode Operation (3)

TIOR, TIER, and TGR are valid, and input capture/compare match and interrupt funct
used.

When overflow occurs while TCNT is counting up, the TCFV flag in TSR is set; when
occurs while TCNT is counting down, the TCFU flag is set.

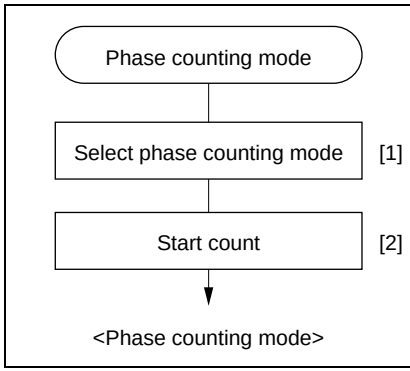
The TCFD bit in TSR is the count direction flag. Reading the TCFD flag provides an
whether TCNT is counting up or down.

Table 10.8 shows the correspondence between external clock pins and channels.

Table 10.8 Phase Counting Mode Clock Input Pins

	External Clock Pi	
Channels	**A-Phase**	**B-Phas**
When channel 1 or 5 is set to phase counting mode	TCLKA	TCLKB
When channel 2 or 4 is set to phase counting mode	TCLKC	TCLKD

Example of Phase Counting Mode Setting Procedure: Figure 10.28 shows an exam
phase counting mode setting procedure.



[1] Select phase counting mode with bits MD
MD0 in TMDR.

[2] Set the CST bit in TSTR to 1 to start the
operation.

Figure 10.28 Example of Phase Counting Mode Setting Procedure

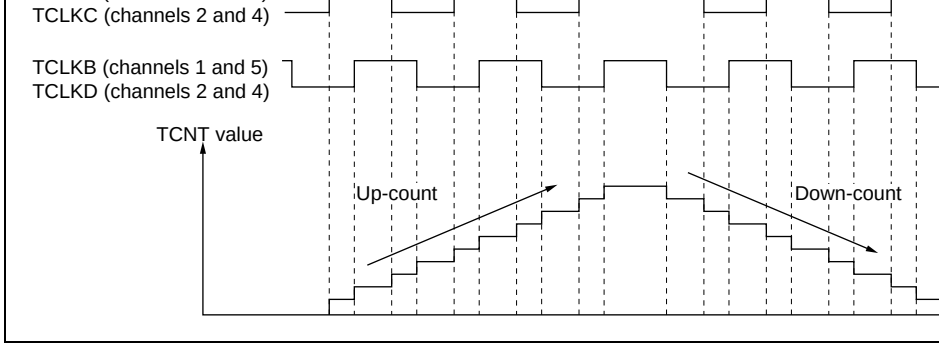


Figure 10.29 Example of Phase Counting Mode 1 Operation

Table 10.9 Up/Down-Count Conditions in Phase Counting Mode 1

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level	↑	Up-count
Low level	↓	Up-count
↑	Low level	Up-count
↓	High level	Up-count
High level	↓	Down-count
Low level	↑	Down-count
↑	High level	Down-count
↓	Low level	Down-count

Legend:

↑: Rising edge

↓: Falling edge

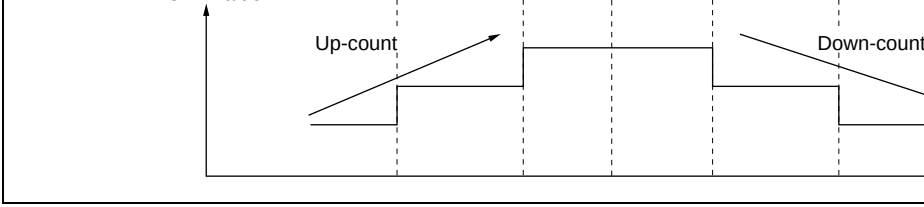


Figure 10.30 Example of Phase Counting Mode 2 Operation

Table 10.10 Up/Down-Count Conditions in Phase Counting Mode 2

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level	↑	Don't care
Low level	↓	Don't care
↑	Low level	Don't care
↓	High level	Up-count
High level	↓	Don't care
Low level	↑	Don't care
↑	High level	Don't care
↓	Low level	Down-coun

Legend:

↑: Rising edge

↓: Falling edge

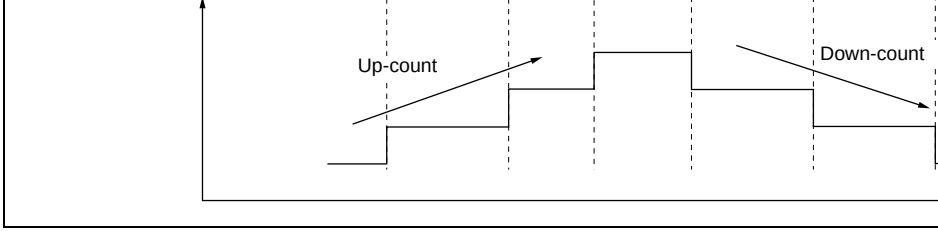


Figure 10.31 Example of Phase Counting Mode 3 Operation

Table 10.11 Up/Down-Count Conditions in Phase Counting Mode 3

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level	↑	Don't care
Low level	↓	Don't care
↑	Low level	Don't care
↓	High level	Up-count
High level	↓	Down-count
Low level	↑	Don't care
↑	High level	Don't care
↓	Low level	Don't care

Legend:

↑: Rising edge

↓: Falling edge

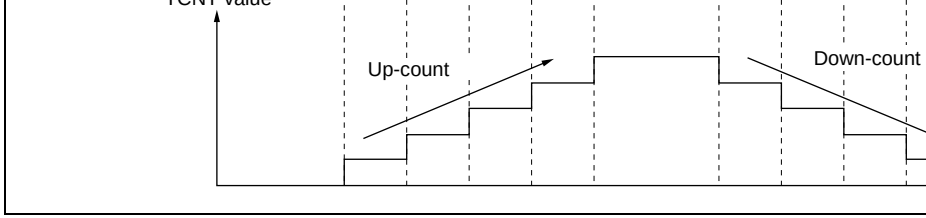


Figure 10.32 Example of Phase Counting Mode 4 Operation

Table 10.12 Up/Down-Count Conditions in Phase Counting Mode 4

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level	↑	Up-count
Low level	↓	Up-count
↑	Low level	Don't care
↓	High level	Don't care
High level	↓	Down-coun
Low level	↑	Down-coun
↑	High level	Don't care
↓	Low level	Don't care

Legend:
↑: Rising edge
↓: Falling edge

control period. TGR0B is used for input capture, with TGR0B and TGR0D operating in
mode. The channel 1 counter input clock is designated as the TGR0B input capture sou
detection of the pulse width of 2-phase encoder 4-multiplication pulses is performed.

TGR1A and TGR1B for channel 1 are designated for input capture, channel 0 TGR0A
TGR0C compare matches are selected as the input capture source, and store the up/dow
values for the control periods.

This procedure enables accurate position/speed detection to be achieved.

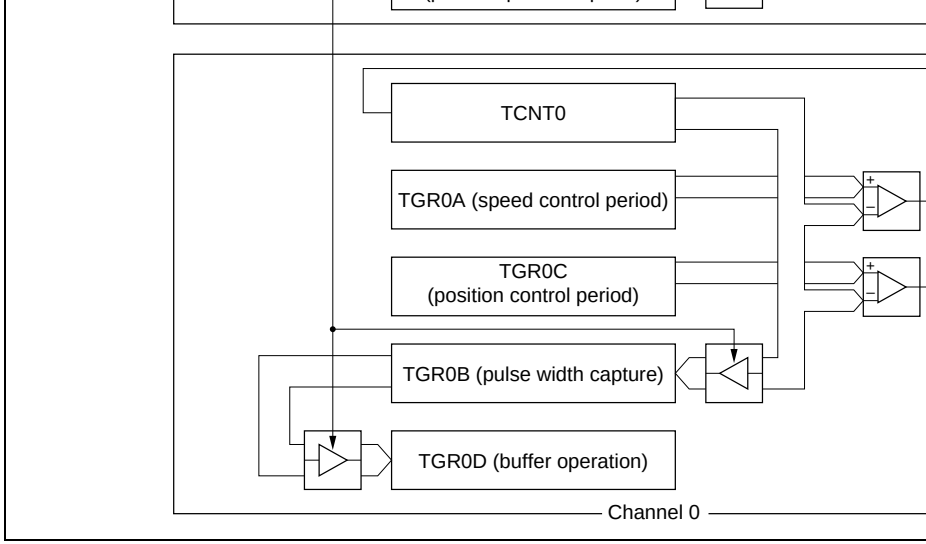


Figure 10.33 Phase Counting Mode Application Example

10.5 Interrupts

10.5.1 Interrupt Sources and Priorities

There are three kinds of TPU interrupt source: TGR input capture/compare match, TC overflow, and TCNT underflow. Each interrupt source has its own status flag and enal bit, allowing generation of interrupt request signals to be enabled or disabled individu

When an interrupt request is generated, the corresponding status flag in TSR is set to 1 corresponding enable/disable bit in TIER is set to 1 at this time, an interrupt is request interrupt request is cleared by clearing the status flag to 0.

Relative channel priorities can be changed by the interrupt controller, but the priority a channel is fixed. For details, see section 5, Interrupt Controller.

	TGI0D	TGR0D input capture/compare match	Possible
	TCI0V	TCNT0 overflow	Not possible
1	TGI1A	TGR1A input capture/compare match	Possible
	TGI1B	TGR1B input capture/compare match	Possible
	TCI1V	TCNT1 overflow	Not possible
	TCI1U	TCNT1 underflow	Not possible
2	TGI2A	TGR2A input capture/compare match	Possible
	TGI2B	TGR2B input capture/compare match	Possible
	TCI2V	TCNT2 overflow	Not possible
	TCI2U	TCNT2 underflow	Not possible
3	TGI3A	TGR3A input capture/compare match	Possible
	TGI3B	TGR3B input capture/compare match	Possible
	TGI3C	TGR3C input capture/compare match	Possible
	TGI3D	TGR3D input capture/compare match	Possible
	TCI3V	TCNT3 overflow	Not possible
4	TGI4A	TGR4A input capture/compare match	Possible
	TGI4B	TGR4B input capture/compare match	Possible
	TCI4V	TCNT4 overflow	Not possible
	TCI4U	TCNT4 underflow	Not possible
5	TGI5A	TGR5A input capture/compare match	Possible
	TGI5B	TGR5B input capture/compare match	Possible
	TCI5V	TCNT5 overflow	Not possible
	TCI5U	TCNT5 underflow	Not possible

Note: This table shows the initial state immediately after a reset. The relative channel p
can be changed by the interrupt controller.

each channel.

Underflow Interrupt: An interrupt is requested if the TCIEU bit in TIER is set to 1 v
TCFU flag in TSR is set to 1 by the occurrence of TCNT underflow on a channel. The
request is cleared by clearing the TCFU flag to 0. The TPU has four underflow interr
each for channels 1, 2, 4, and 5.

10.5.2 DTC Activation

The DTC can be activated by the TGR input capture/compare match interrupt for a ch
details, see section 8, Data Transfer Controller (DTC).

A total of 16 TPU input capture/compare match interrupts can be used as DTC activat
four each for channels 0 and 3, and two each for channels 1, 2, 4, and 5.

10.5.3 A/D Converter Activation

The A/D converter can be activated by the TGRA input capture/compare match for a c

If the TTGE bit in TIER is set to 1 when the TGFA flag in TSR is set to 1 by the occu
TGRA input capture/compare match on a particular channel, a request to start A/D co
sent to the A/D converter. If the TPU conversion start trigger has been selected on the
converter side at this time, A/D conversion is started.

In the TPU, a total of six TGRA input capture/compare match interrupts can be used a
converter conversion start sources, one for each channel.

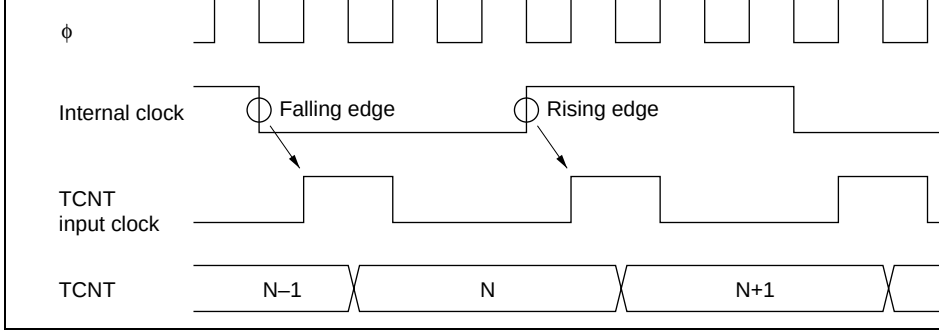


Figure 10.34 Count Timing in Internal Clock Operation

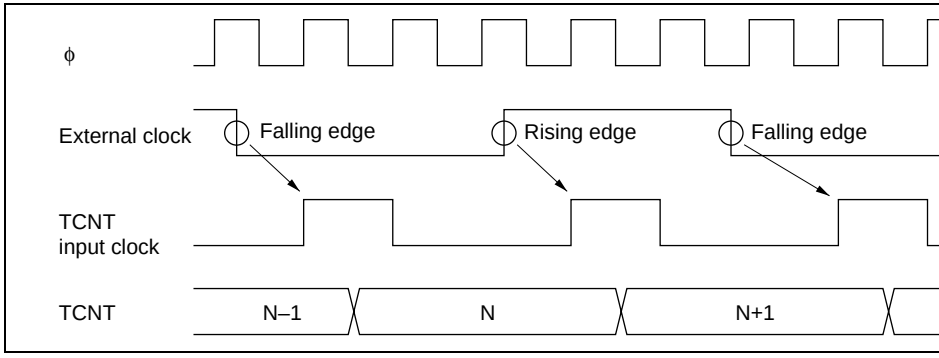


Figure 10.35 Count Timing in External Clock Operation

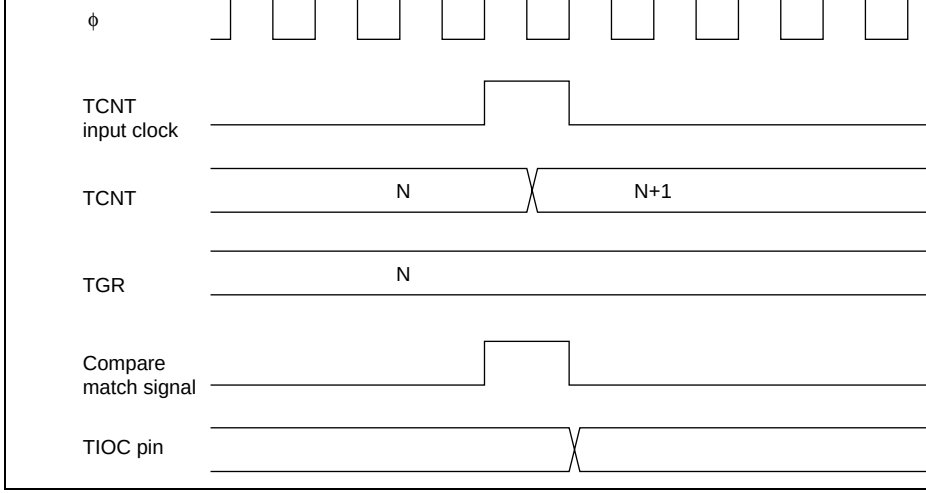


Figure 10.36 Output Compare Output Timing

Input Capture Signal Timing: Figure 10.37 shows input capture signal timing.

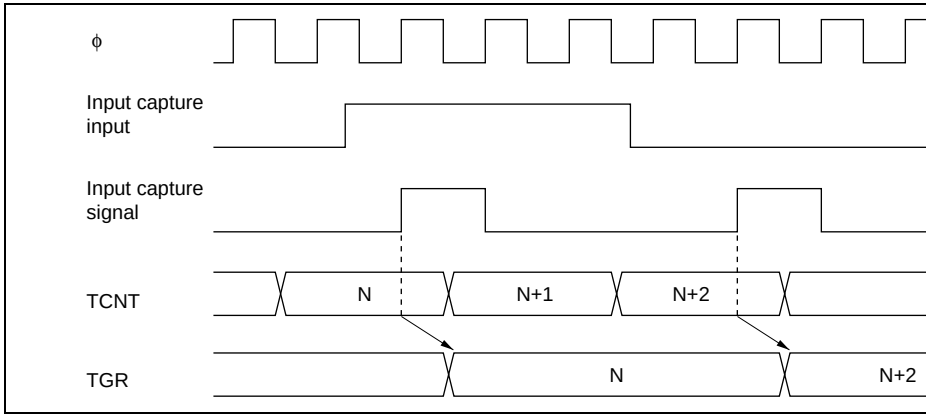


Figure 10.37 Input Capture Input Signal Timing

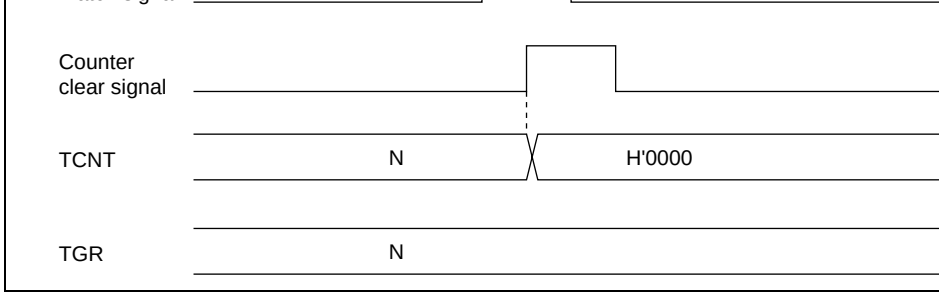


Figure 10.38 Counter Clear Timing (Compare Match)

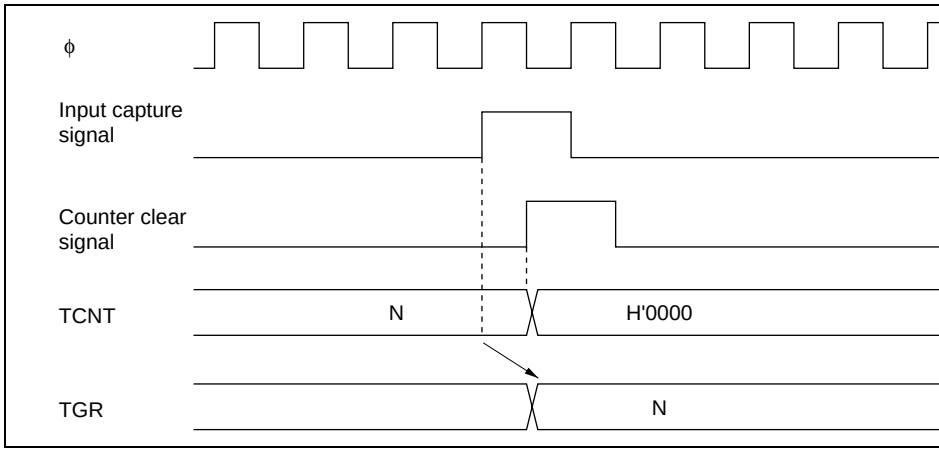


Figure 10.39 Counter Clear Timing (Input Capture)

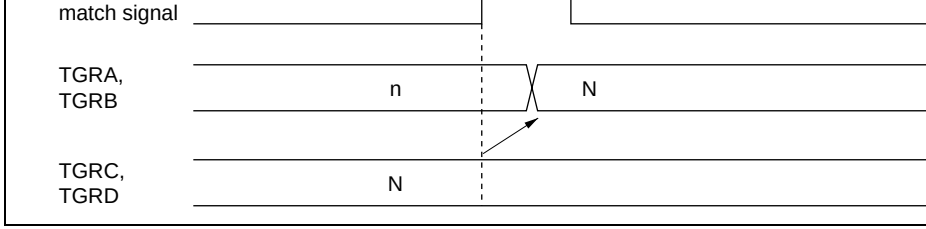


Figure 10.40 Buffer Operation Timing (Compare Match)

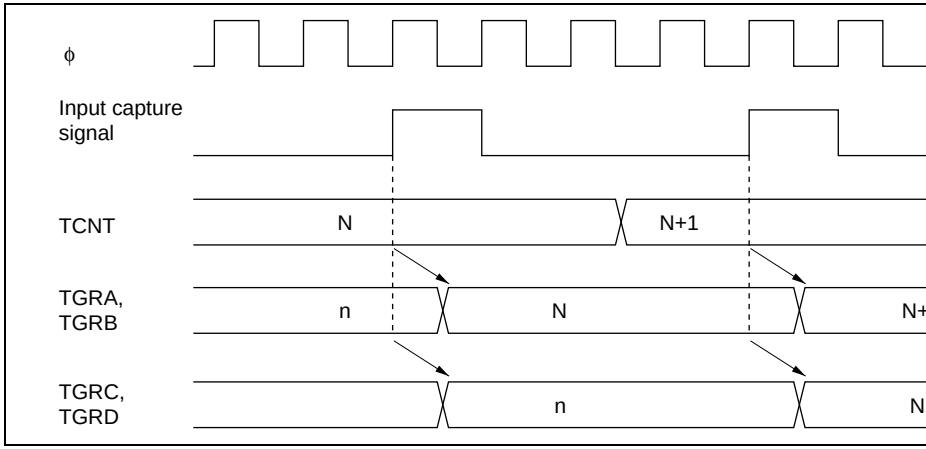


Figure 10.41 Buffer Operation Timing (Input Capture)

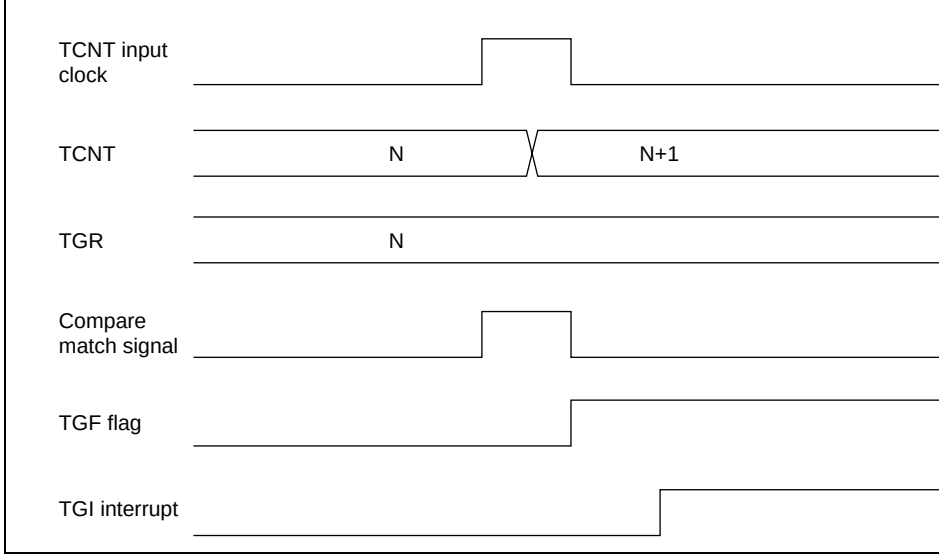


Figure 10.42 TGI Interrupt Timing (Compare Match)

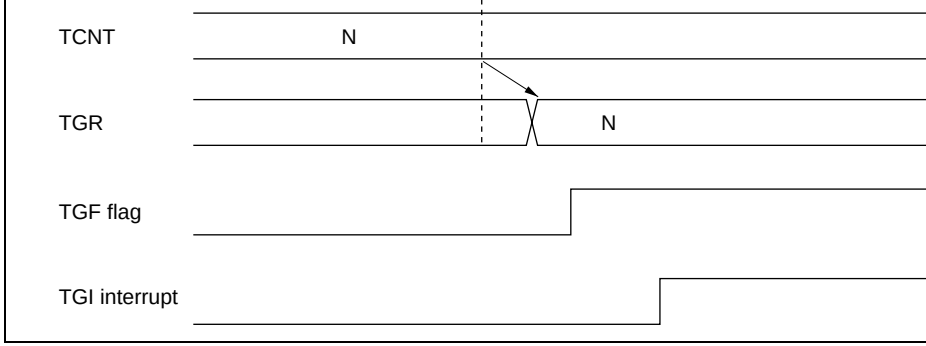


Figure 10.43 TGI Interrupt Timing (Input Capture)

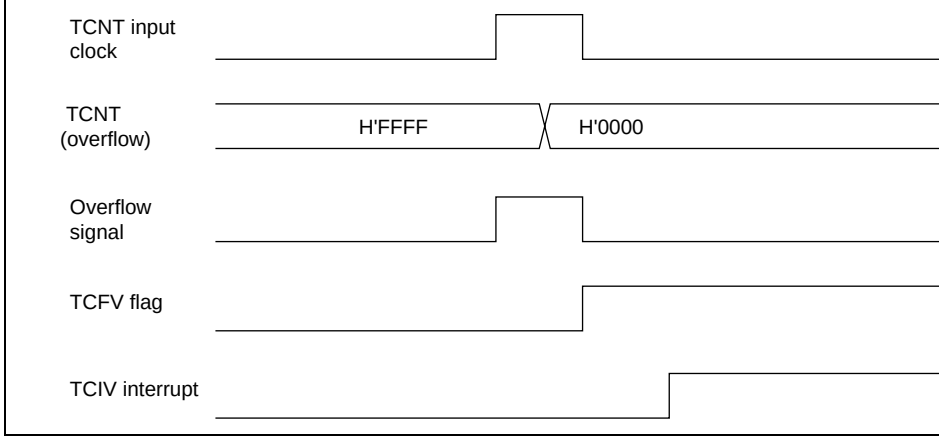


Figure 10.44 TCIV Interrupt Setting Timing

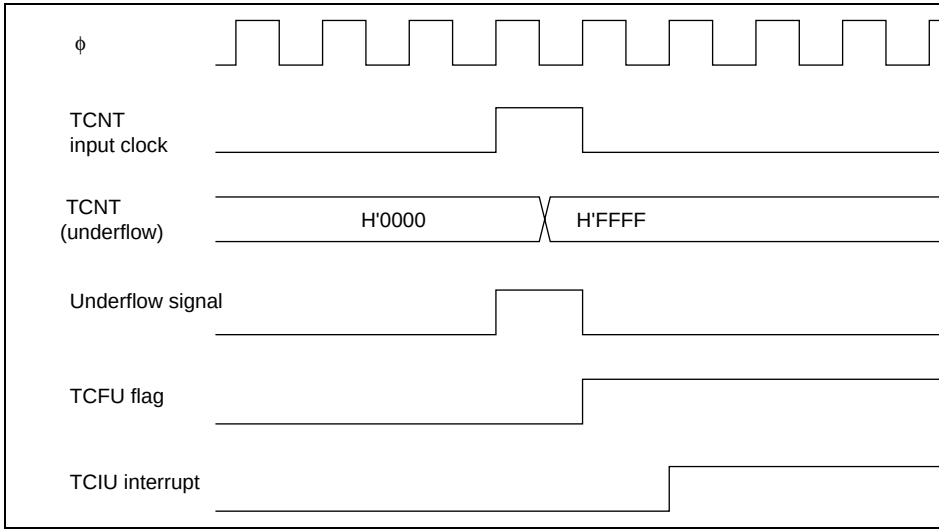


Figure 10.45 TCIU Interrupt Setting Timing

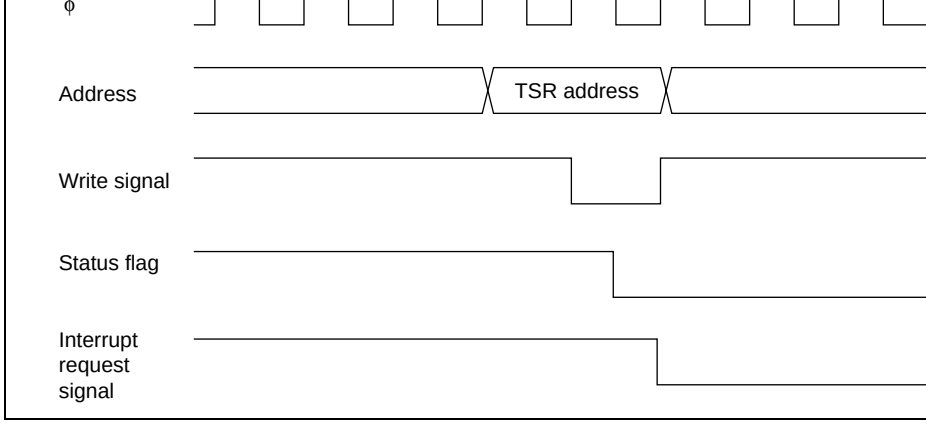


Figure 10.46 Timing for Status Flag Clearing by CPU

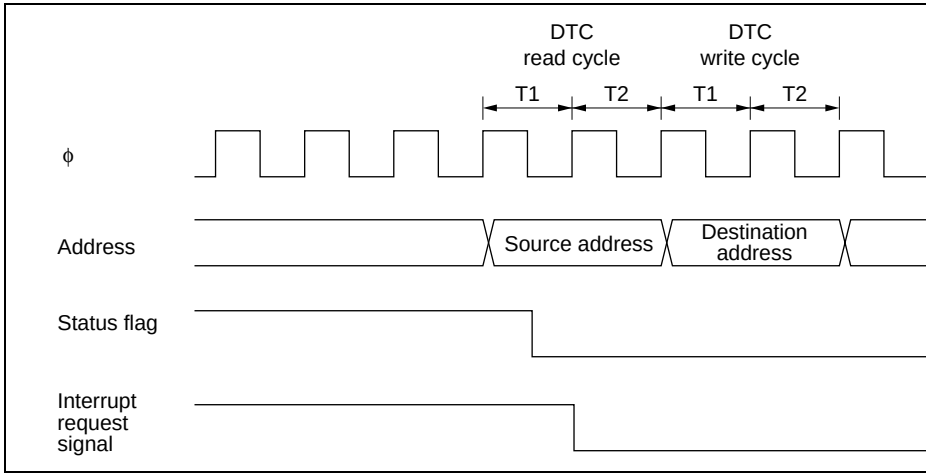


Figure 10.47 Timing for Status Flag Clearing by DTC Activation

In phase counting mode, the phase difference and overlap between the two input clocks least 1.5 states, and the pulse width must be at least 2.5 states. Figure 10.48 shows the i conditions in phase counting mode.

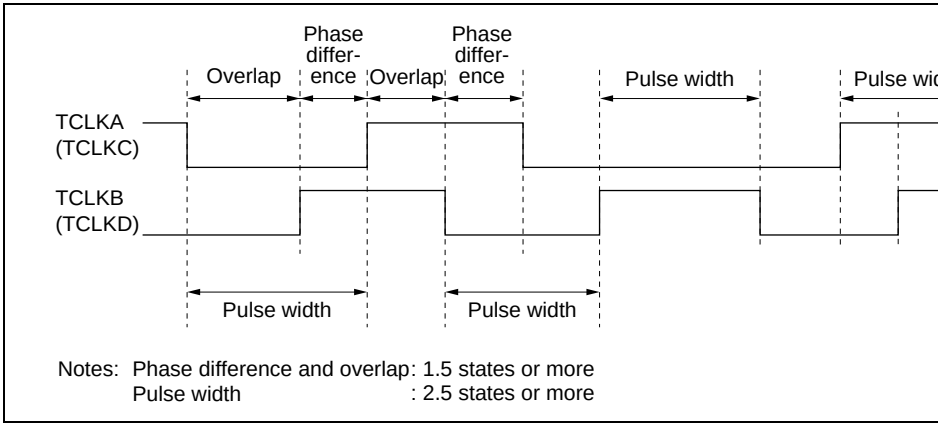


Figure 10.48 Phase Difference, Overlap, and Pulse Width in Phase Counting

Caution on Period Setting: When counter clearing by compare match is set, TCNT is the final state in which it matches the TGR value (the point at which the count value ma TCNT is updated). Consequently, the actual counter frequency is given by the followin

$$f = \frac{\phi}{(N + 1)}$$

Where f: Counter frequency
ϕ: Operating frequency
N: TGR set value

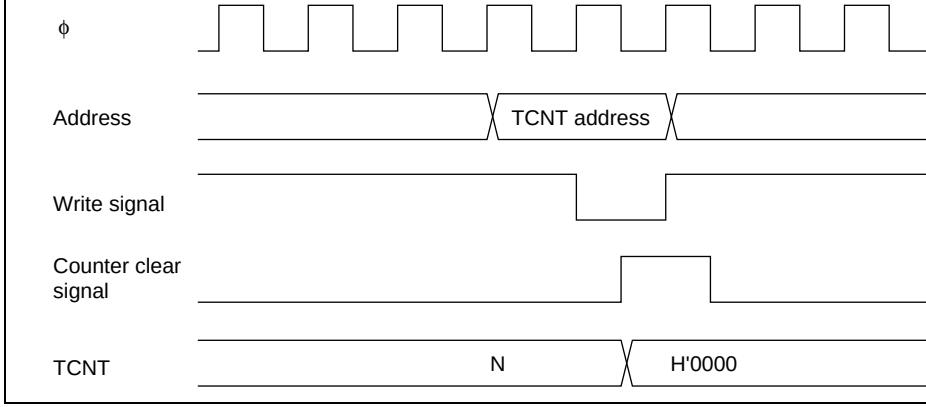


Figure 10.49 Contention between TCNT Write and Clear Operation

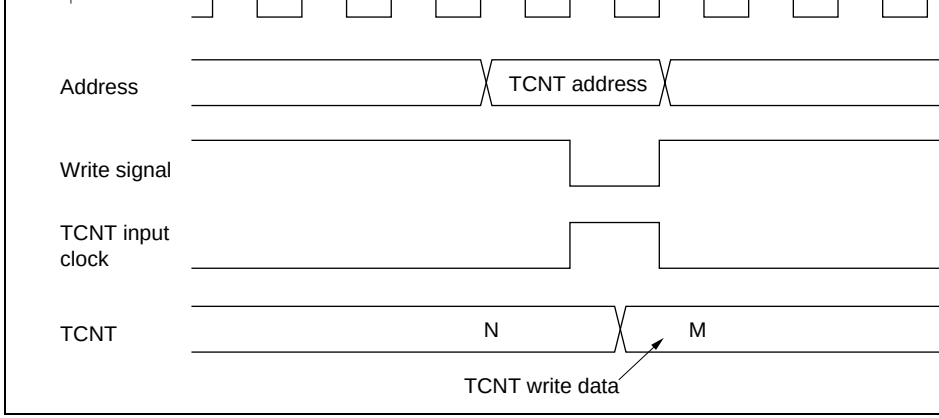


Figure 10.50 Contention between TCNT Write and Increment Operatio

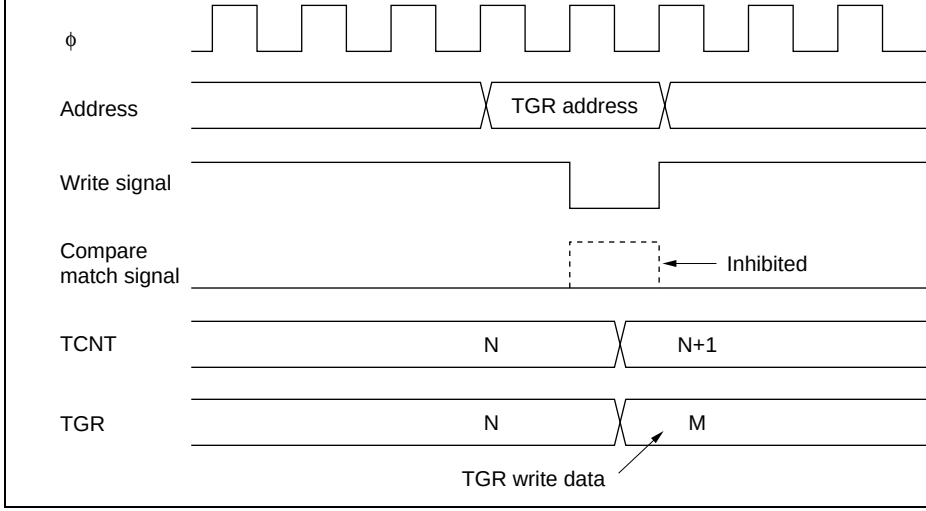


Figure 10.51 Contention between TGR Write and Compare Match

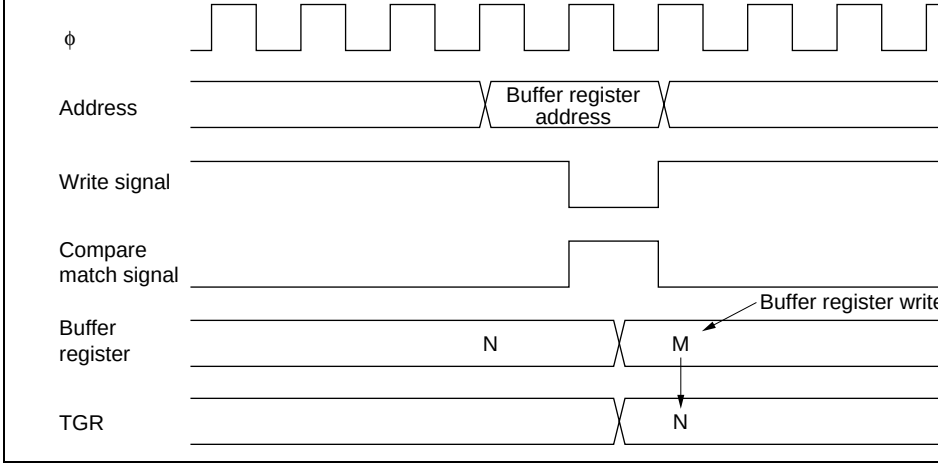


Figure 10.52 Contention between Buffer Register Write and Compare Ma

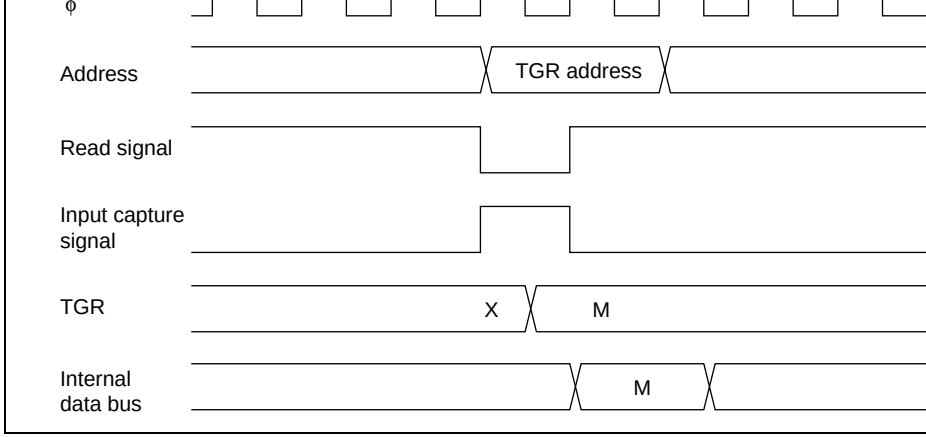


Figure 10.53 Contention between TGR Read and Input Capture

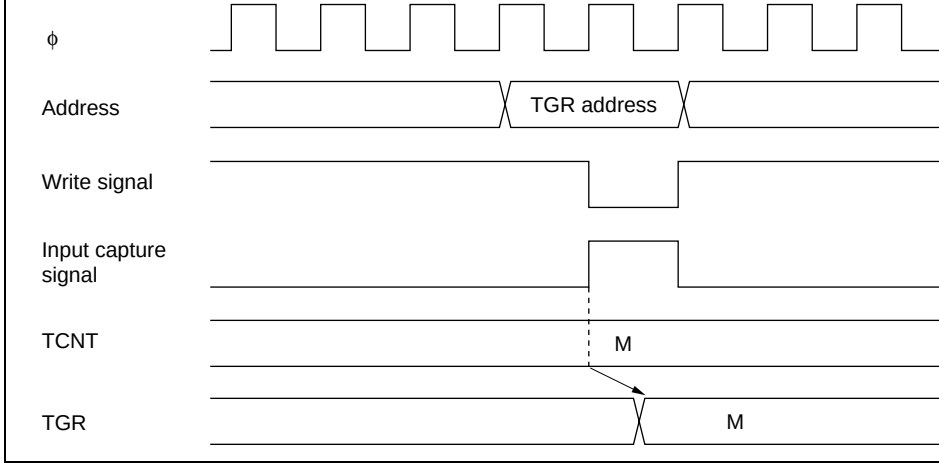


Figure 10.54 Contention between TGR Write and Input Capture

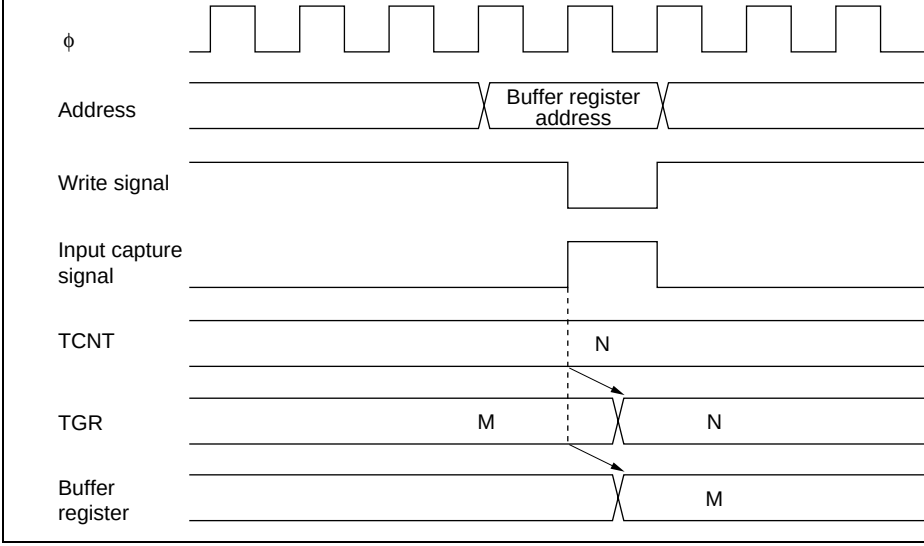


Figure 10.55 Contention between Buffer Register Write and Input Capt

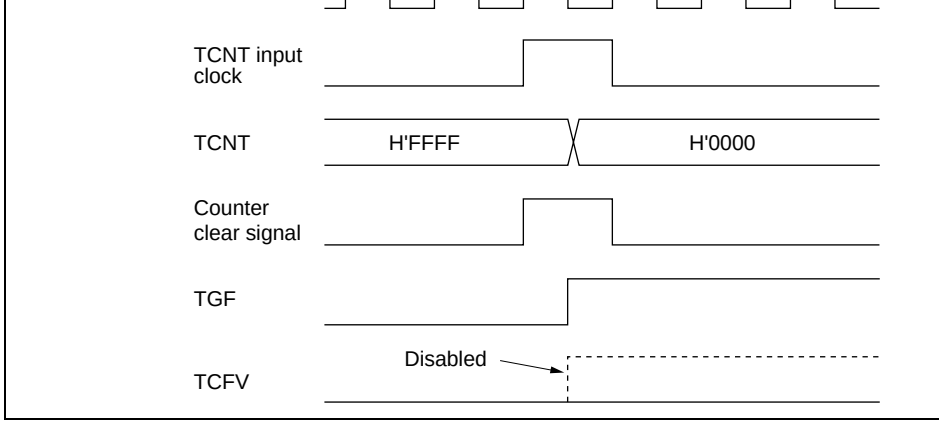


Figure 10.56 Contention between Overflow and Counter Clearing

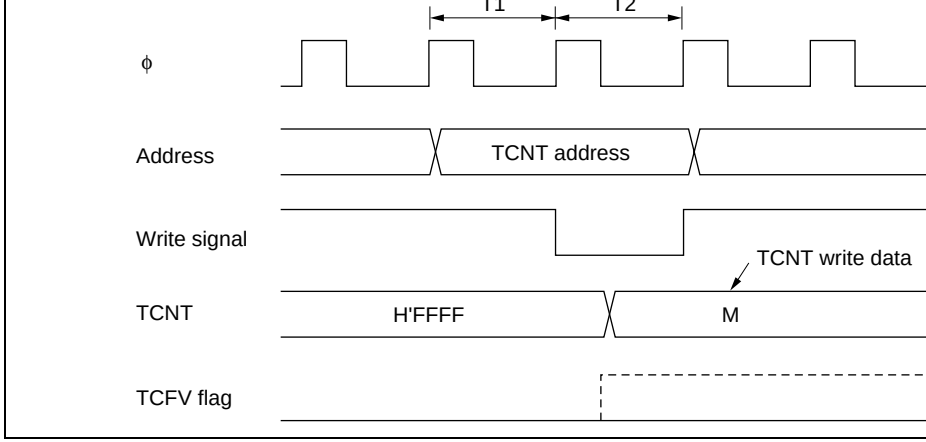


Figure 10.57 Contention between TCNT Write and Overflow

Multiplexing of I/O Pins: In the H8S/2626 Group and H8S/2623 Group, the TCLKA multiplexed with the TIOCC0 I/O pin, the TCLKB input pin with the TIOCD0 I/O pi TCLKC input pin with the TIOCB1 I/O pin, and the TCLKD input pin with the TIOC When an external clock is input, compare match output should not be performed from multiplexed pin.

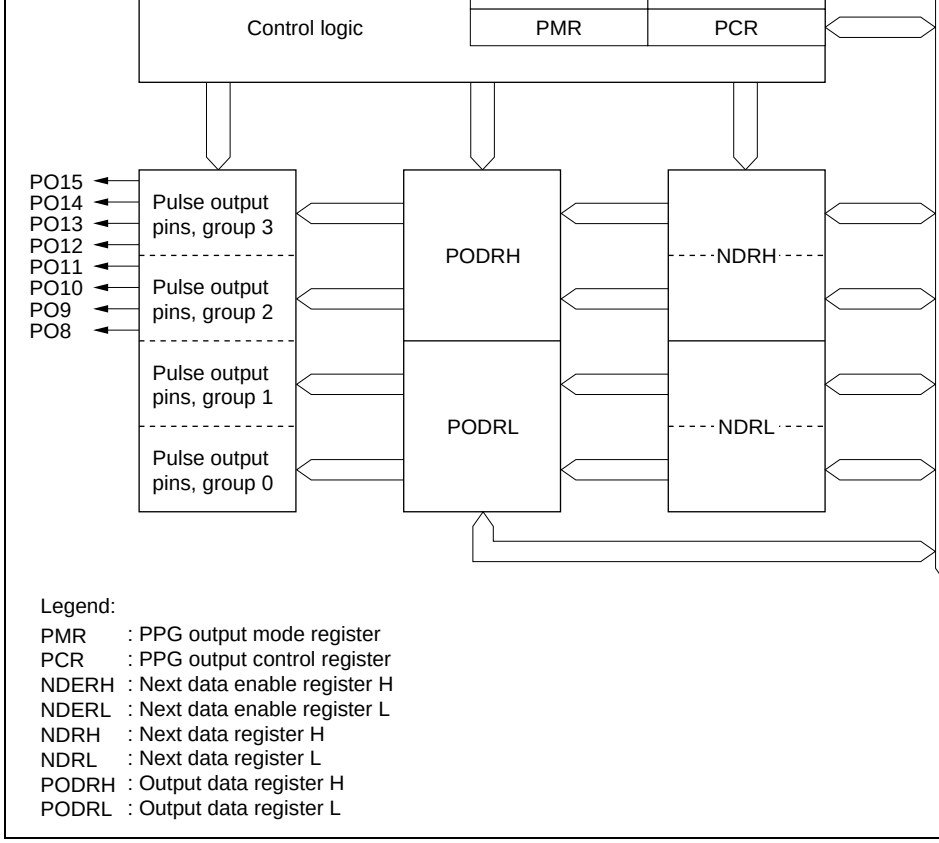
Interrupts and Module Stop Mode: If module stop mode is entered when an interru requested, it will not be possible to clear the CPU interrupt source or the DTC activati Interrupts should therefore be disabled before entering module stop mode.

simultaneously and independently.

11.1.1 Features

PPG features are listed below.

- 8-bit output data
  - Maximum 8-bit data can be output, and output can be enabled on a bit-by-bit b
- Two output groups
  - Output trigger signals can be selected in 4-bit groups to provide up to two diffe outputs
- Selectable output trigger signals
  - Output trigger signals can be selected for each group from the compare match four TPU channels
- Non-overlap mode
  - A non-overlap margin can be provided between pulse outputs
- Can operate together with the data transfer controller (DTC)
  - The compare match signals selected as output trigger signals can activate the D sequential output of data without CPU intervention
- Settable inverted output
  - Inverted data can be output for each group
- Module stop mode can be set
  - As the initial setting, PPG operation is halted. Register access is enabled by ex stop mode



Legend:

PMR	: PPG output mode register
PCR	: PPG output control register
NDERH	: Next data enable register H
NDERL	: Next data enable register L
NDRH	: Next data register H
NDRL	: Next data register L
PODRH	: Output data register H
PODRL	: Output data register L

Figure 11.1 Block Diagram of PPG

Pulse output 9	PO9	Output	
Pulse output 10	PO10	Output	
Pulse output 11	PO11	Output	
Pulse output 12	PO12	Output	Group 3 pulse output
Pulse output 13	PO13	Output	
Pulse output 14	PO14	Output	
Pulse output 15	PO15	Output	

PPG output mode register	PMR	R/W	H'F0	H'F
Next data enable register H	NDERH	R/W	H'00	H'F
Next data enable register L*4	NDERL	R/W	H'00	H'F
Output data register H	PODRH	R/(W)*2	H'00	H'F
Output data register L*4	PODRL	R/(W)*2	H'00	H'F
Next data register H	NDRH	R/W	H'00	H'F H'F
Next data register L*4	NDRL	R/W	H'00	H'F H'F
Port 1 data direction register	P1DDR	W	H'00	H'F
Module stop control register A	MSTPCRA	R/W	H'3F	H'F

Notes: 1. Lower 16 bits of the address.

2. Bits used for pulse output cannot be written to.

3. When the same output trigger is selected for pulse output groups 2 and 3 by setting, the NDRH address is H'FE2C. When the output triggers are differen NDRH address is H'FE2E for group 2 and H'FE2C for group 3
Similarly, when the same output trigger is selected for pulse output groups 0 the PCR setting, the NDRL address is H'FE2D. When the output triggers ar the NDRL address is H'FE2F for group 0 and H'FE2D for group 1.

4. The H8S/2626 Group and H8S/2623 Group have no pins corresponding to P (pulse output groups 0 and 1).

Initial value :	0	0	0	0	0	0	0
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W

NDERL

Bit :	7	6	5	4	3	2	1
	NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER
Initial value :	0	0	0	0	0	0	0
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W

NDERH and NDERL are 8-bit readable/writable registers that enable or disable pulse bit-by-bit basis.

If a bit is enabled for pulse output by NDERH or NDERL, the NDR value is automati transferred to the corresponding PODR bit when the TPU compare match event specif occurs, updating the output value. If pulse output is disabled, the bit value is not trans NDR to PODR and the output value does not change.

NDERH and NDERL are each initialized to H'00 by a reset and in hardware standby m are not initialized in software standby mode.

NDERH Bits 7 to 0—Next Data Enable 15 to 8 (NDER15 to NDER8): These bits e disable pulse output on a bit-by-bit basis.

Bits 7 to 0 NDER15 to NDER8	Description
0	Pulse outputs PO15 to PO8 are disabled (NDR15 to NDR8 ar transferred to POD15 to POD8)
1	Pulse outputs PO15 to PO8 are enabled (NDR15 to NDR8 ar to POD15 to POD8)

11.2.2 Output Data Registers H and L (PODRH, PODRL)

PODRH

Bit	:	7	6	5	4	3	2	1
		POD15	POD14	POD13	POD12	POD11	POD10	POD9
Initial value	:	0	0	0	0	0	0	0
R/W	:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

PODRL

Bit	:	7	6	5	4	3	2	1
		POD7	POD6	POD5	POD4	POD3	POD2	POD1
Initial value	:	0	0	0	0	0	0	0
R/W	:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * A bit that has been set for pulse output by NDER is read-only.

PODRH and PODRL are 8-bit readable/writable registers that store output data for use output. However, the H8S/2626 Group and H8S/2623 Group have no pins correspondi PODRL.

NDRH and NDRL are each initialized to H'00 by a reset and in hardware standby mod
not initialized in software standby mode.

11.2.4 Notes on NDR Access

The NDRH and NDRL addresses differ depending on whether pulse output groups ha
output trigger or different output triggers.

Same Trigger for Pulse Output Groups: If pulse output groups 2 and 3 are triggered
same compare match event, the NDRH address is H'FE2C. The upper 4 bits belong to
and the lower 4 bits to group 2. Address H'FE2E consists entirely of reserved bits tha
modified and are always read as 1.

- Address H'FE2C

Bit	:	7	6	5	4	3	2	1
		NDR15	NDR14	NDR13	NDR12	NDR11	NDR10	NDR9
Initial value	:	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Address H'FE2E

Bit	:	7	6	5	4	3	2	1
		—	—	—	—	—	—	—
Initial value	:	1	1	1	1	1	1	1
R/W	:	—	—	—	—	—	—	—

If pulse output groups 0 and 1 are triggered by the same compare match event, the ND
is H'FE2D. The upper 4 bits belong to group 1 and the lower 4 bits to group 0. Addre
consists entirely of reserved bits that cannot be modified and are always read as 1. Ho
H8S/2626 Group and H8S/2623 Group have no output pins corresponding to pulse ou
0 and 1.

Bit	:	7	6	5	4	3	2	1
		—	—	—	—	—	—	—
Initial value	:	1	1	1	1	1	1	1
R/W	:	—	—	—	—	—	—	—

Different Triggers for Pulse Output Groups: If pulse output groups 2 and 3 are trigg
different compare match events, the address of the upper 4 bits in NDRH (group 3) is H
the address of the lower 4 bits (group 2) is H'FE2E. Bits 3 to 0 of address H'FE2C and
of address H'FE2E are reserved bits that cannot be modified and are always read as 1.

- Address H'FE2C

Bit	:	7	6	5	4	3	2	1
		NDR15	NDR14	NDR13	NDR12	—	—	—
Initial value	:	0	0	0	0	1	1	1
R/W	:	R/W	R/W	R/W	R/W	—	—	—

- Address H'FE2E

Bit	:	7	6	5	4	3	2	1
		—	—	—	—	NDR11	NDR10	NDR9
Initial value	:	1	1	1	1	0	0	0
R/W	:	—	—	—	—	R/W	R/W	R/W

If pulse output groups 0 and 1 are triggered by different compare match event, the addr
upper 4 bits in NDRL (group 1) is H'FE2D and the address of the lower 4 bits (group 0)
H'FE2F. Bits 3 to 0 of address H'FE2D and bits 7 to 4 of address H'FE2F are reserved b
cannot be modified and are always read as 1. However, the H8S/2626 Group and H8S/
have no output pins corresponding to pulse output groups 0 and 1.

Bit	:	7	6	5	4	3	2	1
		—	—	—	—	NDR3	NDR2	NDR1
Initial value	:	1	1	1	1	0	0	0
R/W	:	—	—	—	—	R/W	R/W	R/W

11.2.5 PPG Output Control Register (PCR)

Bit	:	7	6	5	4	3	2	1
		G3CMS1	G3CMS0	G2CMS1	G2CMS0	G1CMS1	G1CMS0	G0CMS
Initial value	:	1	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PCR is an 8-bit readable/writable register that selects output trigger signals for PPG ou group-by-group basis.

PCR is initialized to H'FF by a reset and in hardware standby mode. It is not initialize software standby mode.

Bits 7 and 6—Group 3 Compare Match Select 1 and 0 (G3CMS1, G3CMS0): The select the compare match that triggers pulse output group 3 (pins PO15 to PO12).

Bit 7 G3CMS1	Bit 6 G3CMS0	Description Output Trigger for Pulse Output Group 3
0	0	Compare match in TPU channel 0
	1	Compare match in TPU channel 1
1	0	Compare match in TPU channel 2
	1	Compare match in TPU channel 3

	1	Compare match in TPU channel 3

Bits 3 and 2—Group 1 Compare Match Select 1 and 0 (G1CMS1, G1CMS0): Thes
select the compare match that triggers pulse output group 1 (pins PO7 to PO4). Howev
H8S/2626 Group and H8S/2623 Group have no output pins corresponding to pulse outp

Bit 3 G1CMS1	Bit 2 G1CMS0	Description Output Trigger for Pulse Output Group 1
0	0	Compare match in TPU channel 0
	1	Compare match in TPU channel 1
1	0	Compare match in TPU channel 2
	1	Compare match in TPU channel 3

Bits 1 and 0—Group 0 Compare Match Select 1 and 0 (G0CMS1, G0CMS0): Thes
select the compare match that triggers pulse output group 0 (pins PO3 to PO0). Howev
H8S/2626 Group and H8S/2623 Group have no output pins corresponding to pulse outp

Bit 1 G0CMS1	Bit 0 G0CMS0	Description Output Trigger for Pulse Output Group 0
0	0	Compare match in TPU channel 0
	1	Compare match in TPU channel 1
1	0	Compare match in TPU channel 2
	1	Compare match in TPU channel 3

PMR is an 8-bit readable/writable register that selects pulse output inversion and non-
operation for each group.

The output trigger period of a non-overlapping operation PPG output waveform is set
and the non-overlap margin is set in TGRA. The output values change at compare ma

For details, see section 11.3.4, Non-Overlapping Pulse Output.

PMR is initialized to H'F0 by a reset and in hardware standby mode. It is not initializ
software standby mode.

Bit 7—Group 3 Inversion (G3INV): Selects direct output or inverted output for puls
group 3 (pins PO15 to PO12).

Bit 7 G3INV	Description
0	Inverted output for pulse output group 3 (low-level output at pin for a 1 in P
1	Direct output for pulse output group 3 (high-level output at pin for a 1 in PO

Bit 6—Group 2 Inversion (G2INV): Selects direct output or inverted output for puls
group 2 (pins PO11 to PO8).

Bit 6 G2INV	Description
0	Inverted output for pulse output group 2 (low-level output at pin for a 1 in P
1	Direct output for pulse output group 2 (high-level output at pin for a 1 in PO

Bit 4—Group 0 Inversion (G0INV): Selects direct output or inverted output for pulse
group 0 (pins PO3 to PO0). However, the H8S/2626 Group and H8S/2623 Group have
corresponding to pulse output group 0.

Bit 4 G0INV	Description
0	Inverted output for pulse output group 0 (low-level output at pin for a 1 in P
1	Direct output for pulse output group 0 (high-level output at pin for a 1 in PO

Bit 3—Group 3 Non-Overlap (G3NOV): Selects normal or non-overlapping operatio
output group 3 (pins PO15 to PO12).

Bit 3 G3NOV	Description
0	Normal operation in pulse output group 3 (output values updated at compar in the selected TPU channel)
1	Non-overlapping operation in pulse output group 3 (independent 1 and 0 ou compare match A or B in the selected TPU channel)

Bit 2—Group 2 Non-Overlap (G2NOV): Selects normal or non-overlapping operatio
output group 2 (pins PO11 to PO8).

Bit 2 G2NOV	Description
0	Normal operation in pulse output group 2 (output values updated at compar in the selected TPU channel)
1	Non-overlapping operation in pulse output group 2 (independent 1 and 0 ou compare match A or B in the selected TPU channel)

1	Non-overlapping operation in pulse output group 1 (independent 1 and 0 c compare match A or B in the selected TPU channel)

Bit 0—Group 0 Non-Overlap (G0NOV): Selects normal or non-overlapping operati output group 0 (pins PO3 to PO0). However, the H8S/2626 Group and H8S/2623 Gro pins corresponding to pulse output group 0.

Bit 0 G0NOV	Description
0	Normal operation in pulse output group 0 (output values updated at compa in the selected TPU channel)
1	Non-overlapping operation in pulse output group 0 (independent 1 and 0 c compare match A or B in the selected TPU channel)

11.2.7 Port 1 Data Direction Register (P1DDR)

Bit	:	7	6	5	4	3	2	1
		P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DD
Initial value	:	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W

P1DDR is an 8-bit write-only register, the individual bits of which specify input or ou pins of port 1.

Port 1 is multiplexed with pins PO15 to PO8. Bits corresponding to pins used for PPG be set to 1. For further information about P1DDR, see section 9.2, Port 1.

When the MSTPA3 bit in MSTPCRA is set to 1, PPG operation stops at the end of the
and a transition is made to module stop mode. Registers cannot be read or written to in
stop mode. For details, see sections 21A.5, 21B.5, Module Stop Mode.

MSTPCRA is initialized to H'3F by a reset and in hardware standby mode. It is not init
software standby mode.

Bit 3—Module Stop (MSTPA3): Specifies the PPG module stop mode.

Bit 3 MSTPA3	Description	
0	PPG module stop mode cleared	
1	PPG module stop mode set	(

transferred to PODR to update the output values.

Figure 11.2 illustrates the PPG output operation and table 11.3 summarizes the PPG o
conditions.

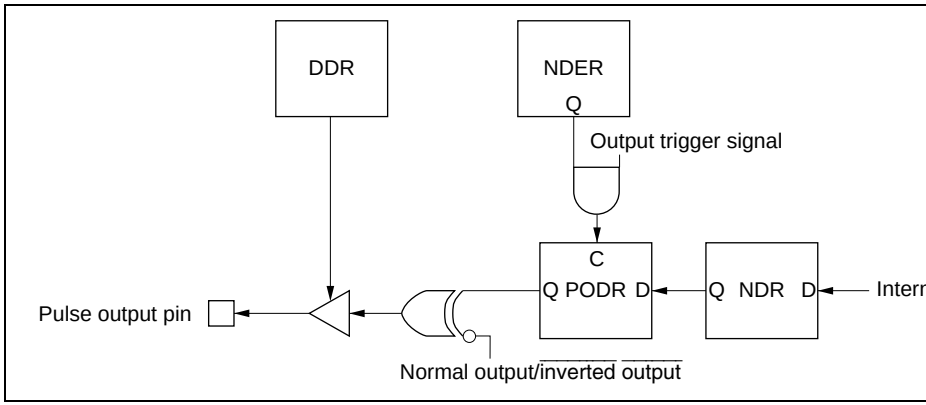


Figure 11.2 PPG Output Operation

Table 11.3 PPG Operating Conditions

NDER	DDR	Pin Function
0	0	Generic input port
	1	Generic output port
1	0	Generic input port (but the PODR bit is a read-only bit, and w compare match occurs, the NDR bit value is transferred to the
	1	PPG pulse output

Sequential output of data of up to 16 bits is possible by writing new output data to NI
the next compare match. For details of non-overlapping operation, see section 11.3.4,
Overlapping Pulse Output.

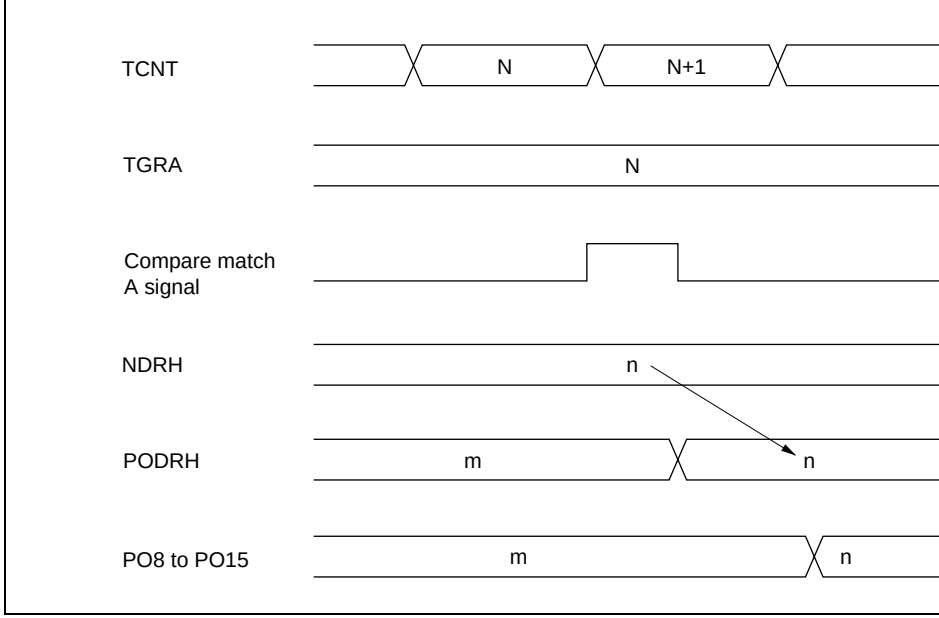
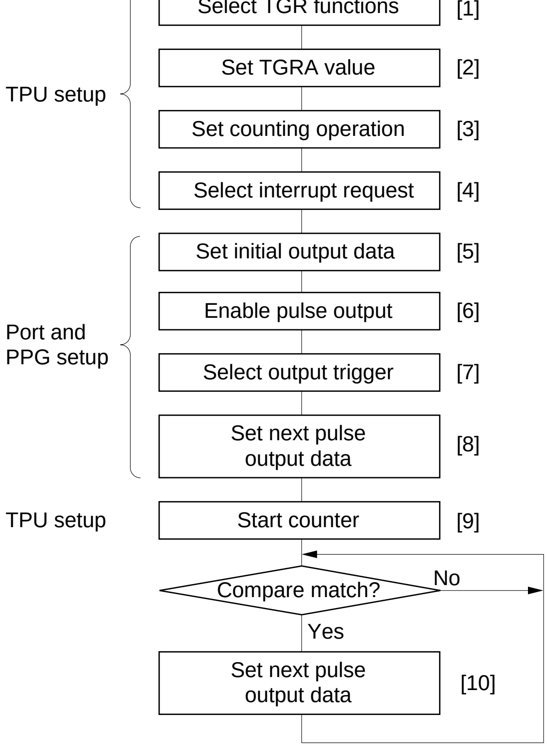


Figure 11.3 Timing of Transfer and Output of NDR Contents (Example



disabled)

[2] Set the PPG output trigger

[3] Select the counter clock so
with bits TPSC2 to TPSC0
Select the counter clear so
with bits CCLR1 and CCLR

[4] Enable the TGIA interrupt
The DTC can also be set u
transfer data to NDR.

[5] Set the initial output values
PODR.

[6] Set the DDR and NDER bi
pins to be used for pulse ou
1.

[7] Select the TPU compare m
event to be used as the ou
trigger in PCR.

[8] Set the next pulse output v
NDR.

[9] Set the CST bit in TSTR to
start the TCNT counter.

[10] At each TGIA interrupt, se

Figure 11.4 Setup Procedure for Normal Pulse Output (Example)

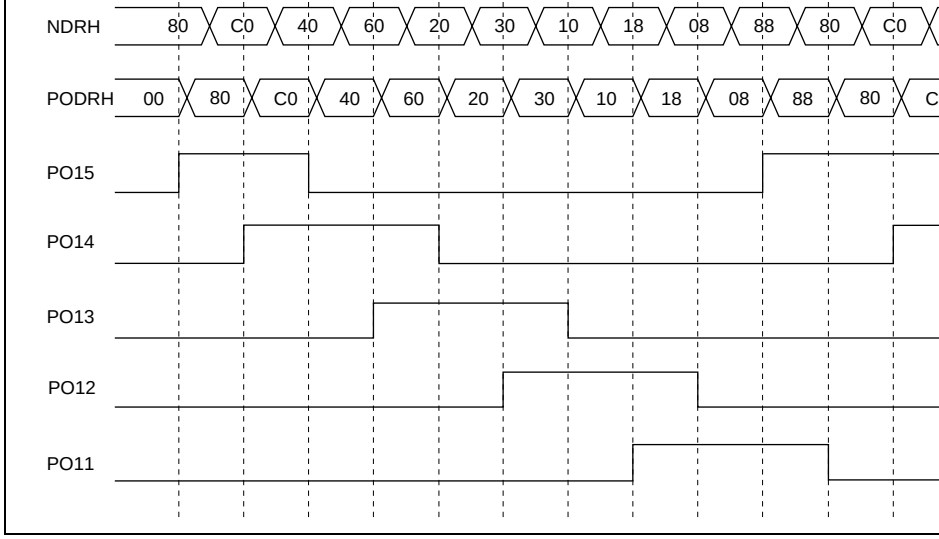


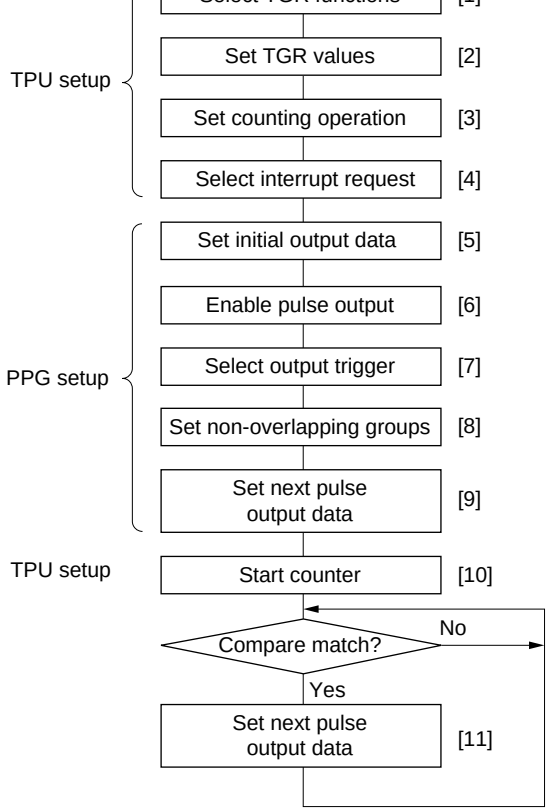
Figure 11.5 Normal Pulse Output Example (Five-Phase Pulse Output)

[1] Set up the TPU channel to be used as the output trigger channel so that TGRA is an compare register and the counter will be cleared by compare match A. Set the trigg TGRA and set the TGIEA bit in TIER to 1 to enable the compare match A (TGIA)

[2] Write H'F8 in P1DDR and NDERH, and set the G3CMS1, G3CMS0, G2CMS1, an bits in PCR to select compare match in the TPU channel set up in the previous step output trigger. Write output data H'80 in NDRH.

[3] The timer counter in the TPU channel starts. When compare match A occurs, the N contents are transferred to PODRH and output. The TGIA interrupt handling routin next output data (H'C0) in NDRH.

[4] Five-phase overlapping pulse output (one or two phases active at a time) can be obt subsequently by writing H'40, H'60, H'20, H'30. H'10, H'18, H'08, H'88, ... at succes interrupts. If the DTC is set for activation by this interrupt, pulse output can be obt without imposing a load on the CPU.



[2] Set the pulse output trigge
in TGRB and the non-over
margin in TGRA.

[3] Select the counter clock s
with bits TPSC2 to TPSC0
Select the counter clear so
with bits CCLR1 and CCL

[4] Enable the TGIA interrupt
The DTC can also be set u
transfer data to NDR.

[5] Set the initial output value
PODR.

[6] Set the DDR and NDER b
pins to be used for pulse o
1.

[7] Select the TPU compare m
event to be used as the pu
output trigger in PCR.

[8] In PMR, select the groups
operate in non-overlap mo

[9] Set the next pulse output v
NDR.

[10] Set the CST bit in TSTR to
start the TCNT counter.

[11] At each TGIA interrupt, se
output values in NDR.

Figure 11.6 Setup Procedure for Non-Overlapping Pulse Output (Exam

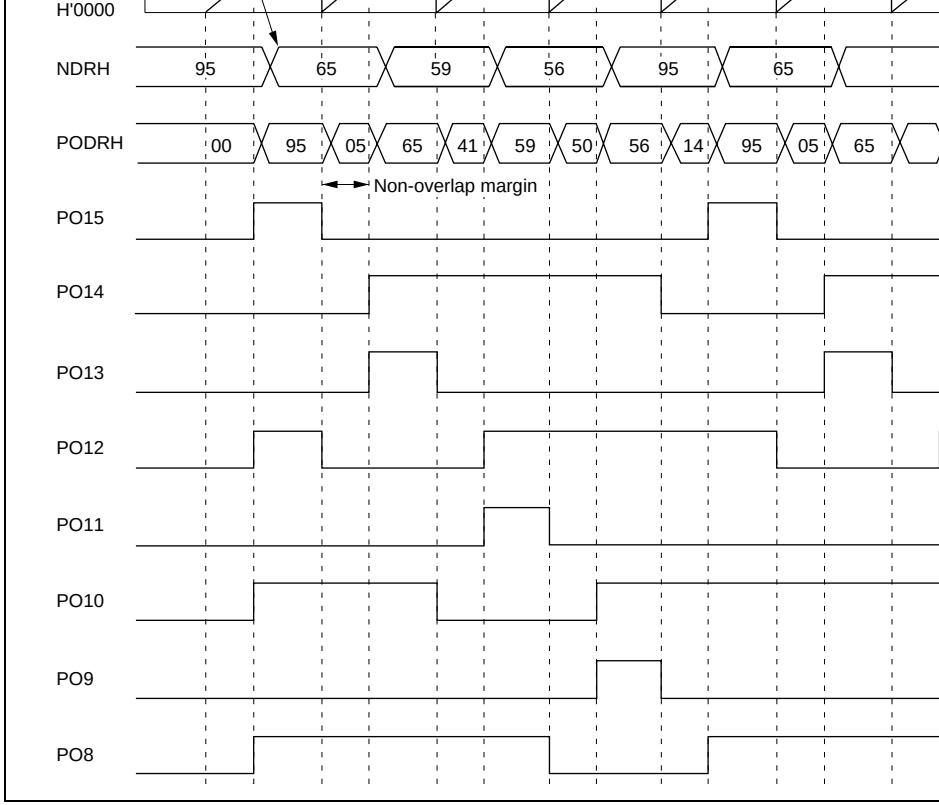


Figure 11.7 Non-Overlapping Pulse Output Example (Four-Phase Compleme

Write output data H'55 in NDRH.

[3] The timer counter in the TPU channel starts. When a compare match with TGRB outputs change from 1 to 0. When a compare match with TGRA occurs, outputs ch to 1 (the change from 0 to 1 is delayed by the value set in TGRA). The TGIA inter handling routine writes the next output data (H'65) in NDRH.

[4] Four-phase complementary non-overlapping pulse output can be obtained subsequ writing H'59, H'56, H'95, ... at successive TGIA interrupts. If the DTC is set for ac this interrupt, pulse output can be obtained without imposing a load on the CPU.

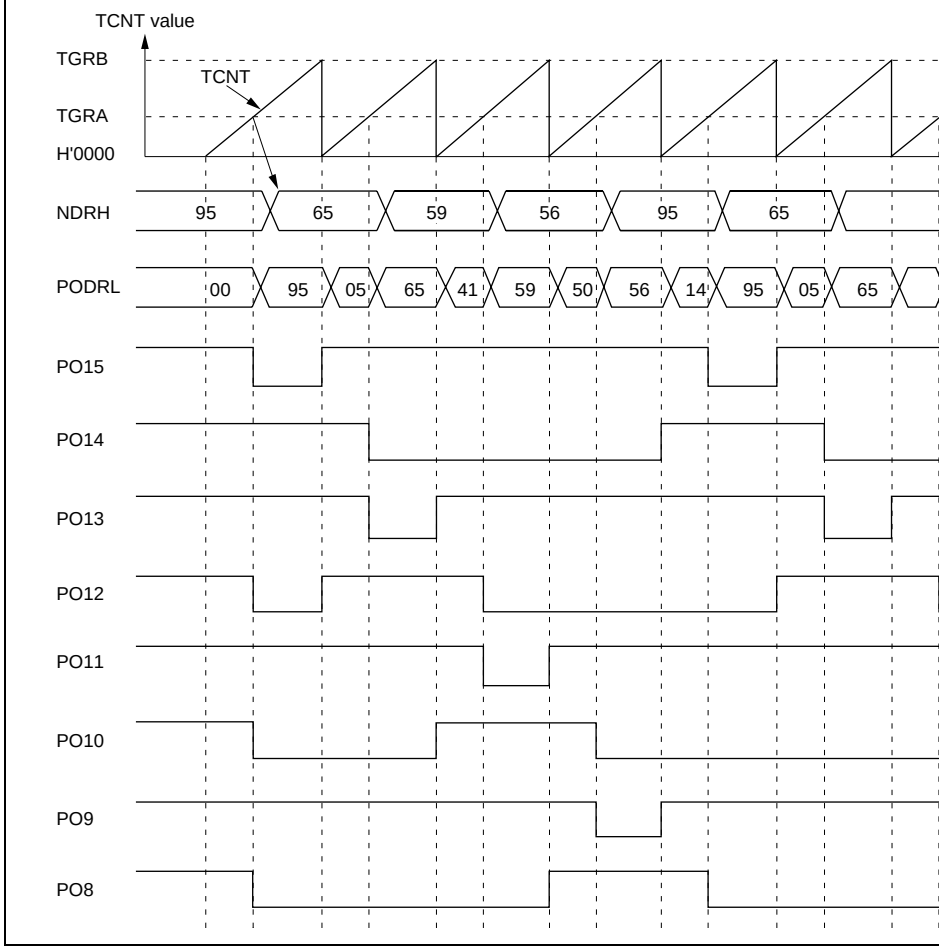


Figure 11.8 Inverted Pulse Output (Example)

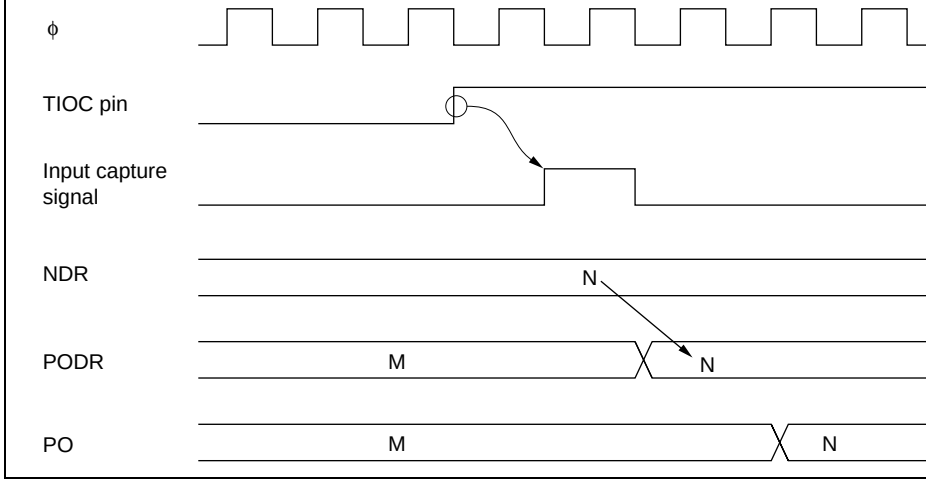


Figure 11.9 Pulse Output Triggered by Input Capture (Example)

occur.

Note on Non-Overlapping Output: During non-overlapping operation, the transfer of values to PODR bits takes place as follows.

- NDR bits are always transferred to PODR bits at compare match A.
- At compare match B, NDR bits are transferred only if their value is 0. Bits are not t if their value is 1.

Figure 11.10 illustrates the non-overlapping pulse output operation.

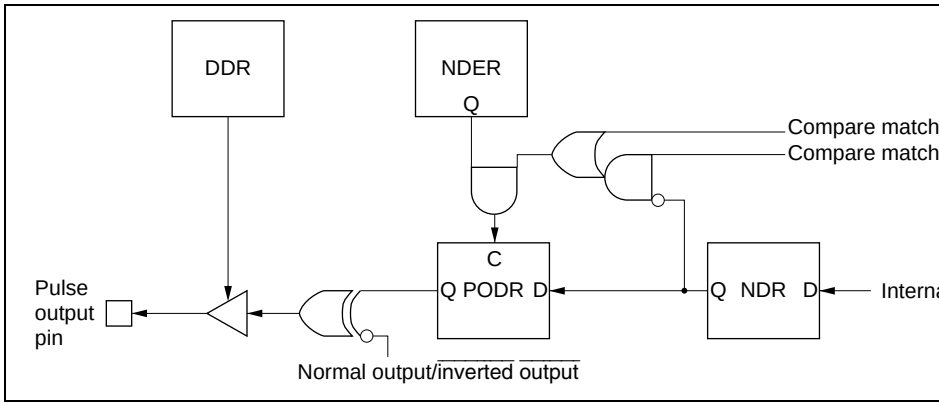


Figure 11.10 Non-Overlapping Pulse Output

Therefore, 0 data can be transferred ahead of 1 data by making compare match B occur compare match A. The NDR contents should not be altered during the interval from co match B to compare match A (the non-overlap margin).

This can be accomplished by having the TGIA interrupt handling routine write the next NDR, or by having the TGIA interrupt activate the DTC. Note, however, that the next d be written before the next compare match B occurs.

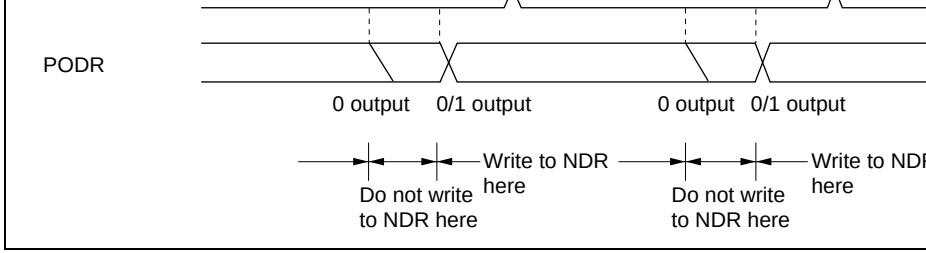


Figure 11.11 Non-Overlapping Operation and NDR Write Timing

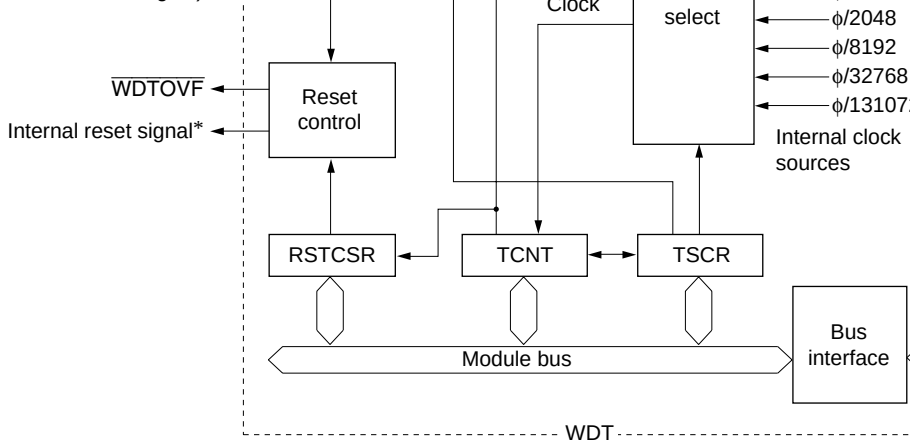
allowing it to overflow. At the same time, the WDT can also generate an internal reset
the H8S/2626 Group or H8S/2623 Group.

When this watchdog function is not needed, the WDT can be used as an interval timer
timer operation, an interval timer interrupt is generated each time the counter overflow

12.1.1 Features

WDT features are listed below.

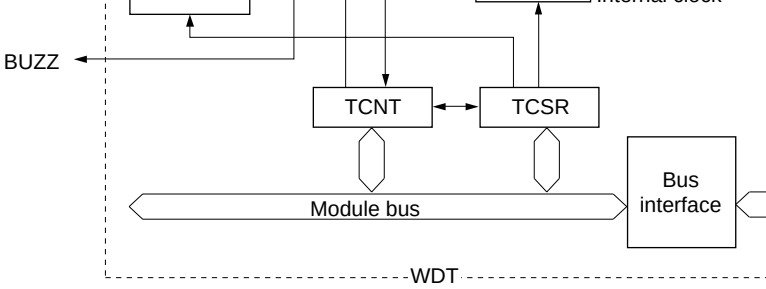
- Switchable between watchdog timer mode and interval timer mode
- $\overline{\text{WDTOVF}}$ output when in watchdog timer mode
  If the counter overflows, the WDT outputs $\overline{\text{WDTOVF}}$. It is possible to select whet
  is internally reset or an NMI interrupt is generated at the same time.
- Interrupt generation when in interval timer mode
  If the counter overflows, the WDT generates an interval timer interrupt.
- WDT0 and WDT1 respectively allow eight and sixteen types of counter input cloc
  selected
  The maximum interval of the WDT is given as a system clock cycle × 131072 × 2
  A subclock may be selected for the input counter of WDT1.
  Where a subclock is selected, the maximum interval is given as a subclock cycle ×
- Selected clock can be output from the BUZZ output pin (WDT1)



Legend:
TCSR : Timer control/status register
TCNT : Timer counter
RSTCSR : Reset control/status register

Note: * The type of internal reset signal depends on a register setting.

Figure 12.1 (a) Block Diagram of WDT0



Legend:
TCSR : Timer control/status register
TCNT : Timer counter

Note: * An internal reset signal can be generated by setting the register.

Figure 12.1 (b) Block Diagram of WDT1

Buzzer output*	BUZZ	Output	Outputs clock selected by watchdo (WDT1)

Note: * Cannot be used in the H8S/2623 Group.

12.1.4 Register Configuration

Table 12.2 summarizes the WDT register configuration. These registers control clock s WDT mode switching, and the reset signal.

Table 12.2 WDT Registers

Channel	Name	Abbreviation	R/W	Initial Value	Ad Write*
0	Timer control/status register 0	TCSR0	R/(W)*3	H'18	H'FF74
	Timer counter 0	TCNT0	R/W	H'00	H'FF74
	Reset control/status register	RSTCSR	R/(W)*3	H'1F	H'FF76
1*4	Timer control/status register 1	TCSR1	R/(W)*3	H'00	H'FFA2
	Timer counter 1	TCNT1	R/W	H'00	H'FFA3
All	Pin function control register	PFCR	R/W	H'0D/H'00	H'FDE

Notes: 1. Lower 16 bits of the address.
2. For details of write operations, see section 12.2.5, Notes on Register Access
3. Only a write of 0 is permitted to bit 7, to clear the flag.
4. Cannot be used in the H8S/2623 Group.

TCNT is an 8-bit readable/writable* up-counter.

When the TME bit is set to 1 in TCSR, TCNT starts counting pulses generated from th
clock source selected by bits CKS2 to CKS0 in TCSR. When the count overflows (cha
H'FF to H'00), either the watchdog timer overflow signal ($\overline{\text{WDTOVF}}$) or an interval ti
(WOVI) is generated, depending on the mode selected by the WT/$\overline{\text{IT}}$ bit in TCSR.

TCNT is initialized to H'00 by a reset, in hardware standby mode, or when the TME b
to 0. It is not initialized in software standby mode.

Note: * TCNT is write-protected by a password to prevent accidental overwriting.
see section 12.2.5, Notes on Register Access.

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

- TCSR1*[1]

Bit	:	7	6	5	4	3	2	1
		OVF	WT/$\overline{IT}$	TME	PSS	RST/$\overline{NMI}$	CKS2	CKS1
Initial value	:	0	0	0	0	0	0	0
R/W	:	R/(W)*[2]	R/W	R/W	R/W	R/W	R/W	R/W

Notes: 1. Cannot be used in the H8S/2623 Group.
2. Only a 0 can be written, for flag clearing.

TCSR is an 8-bit readable/writable* register. Its functions include selecting the clock so input to TCNT, and the timer mode.

TCSR0 (TCSR1) is initialized to H'18 (H'00) by a reset and in hardware standby mode. initialized in software standby mode.

Note: * TCSR is write-protected by a password to prevent accidental overwriting. F see section 12.2.5, Notes on Register Access.

Bit 7—Overflow Flag (OVF): Indicates that TCNT has overflowed from H'FF to H'00

Bit 7

OVF	Description
0	[Clearing conditions] (I • Cleared when 0 is written to the TME bit (Only applies to WDT1) • Cleared by reading TCSR when OVF = 1, then writing 0 to OVF
1	[Setting condition] When TCNT overflows (changes from H'FF to H'00) When internal reset request generation is selected in watchdog timer mode cleared automatically by the internal reset.

interval timer. When TCNT overflows, WDT0 generates the $\overline{\text{WDTOVF}}$ signal when i
timer mode, or a WOVI interrupt request to the CPU when in interval timer mode. W
generates a reset or NMI interrupt request when in watchdog timer mode, or a WOVI
request to the CPU when in interval timer mode.

- WDT0 Mode Select

WDT0

WT/$\overline{\text{IT}}$	Description
0	Interval timer mode: WDT0 requests an interval timer interrupt (WOVI) from the CPU when the TCNT overflows.
1	Watchdog timer mode: WDT0 outputs a $\overline{\text{WDTOVF}}$ signal when the TCNT

Note: * For details on a TCNT overflow in watchdog timer mode, see section 12.2.3 Control/Status Register (RSTCSR).

- WDT1 Mode Select*

WDT1

WT/$\overline{\text{IT}}$	Description
0	Interval timer mode: WDT1 requests an interval timer interrupt (WOVI) from the CPU when the TCNT overflows.
1	Watchdog timer mode: WDT1 requests a reset or an NMI interrupt from the CPU when the TCNT overflows.

Note: * Cannot be used in the H8S/2623 Group.

Bit 5—Timer Enable (TME): Selects whether TCNT runs or is halted.

Bit 5

TME	Description
0	TCNT is initialized to H'00 and halted
1	TCNT counts

WDT1 TCSR Bit 4

PSS	Description
0	The TCNT counts frequency-division clock pulses of the φ based prescaler (PSM).
1	The TCNT counts frequency-division clock pulses of the φ SUB-based pr (PSS).

WDT0 TCSR Bit 3—Reserved Bit: This bit is always read as 1 and cannot be modifi

WDT1 TCSR Bit 3—Reset or NMI (RST/$\overline{\text{NMI}}$): This bit is used to choose between a reset request and an NMI request when the TCNT overflows during the watchdog time

This bit cannot be used in the H8S/2623 Group.

Bit 3

RTS/$\overline{\text{NMI}}$	Description
0	NMI request.
1	Internal reset request.

	1	0	φ/128	1.6 ms
		1	φ/512	6.6 ms
1	0	0	φ/2048	26.2 ms
		1	φ/8192	104.9 ms
	1	0	φ/32768	419.4 ms
		1	φ/131072	1.68 s

Note: * An overflow period is the time interval between the start of counting up from TCNT and the occurrence of a TCNT overflow.

			1	φ/512	6.6 ms
	1	0	0	φ/2048	26.2 ms
			1	φ/8192	104.9 ms
		1	0	φ/32768	419.4 ms
			1	φ/131072	1.68 s
1	0	0	0	φSUB/2	15.6 ms
			1	φSUB/4	31.3 ms
		1	0	φSUB/8	62.5 ms
			1	φSUB/16	125 ms
	1	0	0	φSUB/32	250 ms
			1	φSUB/64	500 ms
		1	0	φSUB/128	1 s
			1	φSUB/256	2 s

Notes: 1. An overflow period is the time interval between the start of counting up from TCNT and the occurrence of a TCNT overflow.
2. Cannot be used in the H8S/2623 Group.

12.2.3 Reset Control/Status Register (RSTCSR)

Bit	:	7	6	5	4	3	2	1
		WOVF	RSTE	RSTS	—	—	—	—
Initial value	:	0	0	0	1	1	1	1
R/W	:	R/(W)*	R/W	R/W	—	—	—	—

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

RSTCSR is an 8-bit readable/writable* register that controls the generation of the inter signal when TCNT overflows, and selects the type of internal reset signal.

Bit 7

WOVF	Description
0	[Clearing condition] Cleared by reading TCSR when WOVF = 1, then writing 0 to WOVF
1	[Setting condition] Set when TCNT overflows (changed from H'FF to H'00) during watchdog t operation

Bit 6—Reset Enable (RSTE): Specifies whether or not a reset signal is generated in TCNT overflows during watchdog timer operation.

Bit 6

RSTE	Description
0	Reset signal is not generated if TCNT overflows*
1	Reset signal is generated if TCNT overflows

Note: * The modules within the chip are not reset, but TCNT and TCSR within the W reset.

Bit 5—Reset Select (RSTS): Selects the type of internal reset generated if TCNT ove during watchdog timer operation.

For details of the types of reset, see section 4, Exception Handling.

Bit 5

RSTS	Description
0	Power-on reset
1	Setting prohibited

Bits 4 to 0—Reserved: These bits are always read as 1 and cannot be modified.

expanded mode.

Only bit 5 is described here. For details of the other bits, see section 7.2.6, Pin Functio
Register (PFCR).

Bit 5—BUZZ Output Enable (BUZZE)*: Enables or disables BUZZ output from the
The WDT1 input clock selected with bits PSS and CKS2 to CKS0 is output as the BUZ

Note: * In the H8S/2623 Group this bit is reserved, and must be written with 0.

Bit 5

BUZZE	Description	
0	Functions as PF1 I/O pin	(In
1	Functions as BUZZ output pin	

12.2.5 Notes on Register Access

The watchdog timer's TCNT, TCSR, and RSTCSR registers differ from other registers
more difficult to write to. The procedures for writing to and reading these registers are
below.

Writing to TCNT and TCSR: These registers must be written to by a word transfer in
They cannot be written to with byte instructions.

Figure 12.2 shows the format of data written to TCNT and TCSR. TCNT and TCSR bo
same write address. For a write to TCNT, the upper byte of the written word must cont
and the lower byte must contain the write data. For a write to TCSR, the upper byte of t
word must contain H'A5 and the lower byte must contain the write data. This transfers
data from the lower byte to TCNT or TCSR.

Figure 12.2 Format of Data Written to TCNT and TCSR

Writing to RSTCSR: RSTCSR must be written to by word transfer instruction to add H'FF76. It cannot be written to with byte instructions.

Figure 12.3 shows the format of data written to RSTCSR. The method of writing 0 to bit differs from that for writing to the RSTE and RSTS bits.

To write 0 to the WOVF bit, the write data must have H'A5 in the upper byte and H'00 lower byte. This clears the WOVF bit to 0, but has no effect on the RSTE and RSTS b to the RSTE and RSTS bits, the upper byte must contain H'5A and the lower byte mus write data. This writes the values in bits 6 and 5 of the lower byte into the RSTE and but has no effect on the WOVF bit.

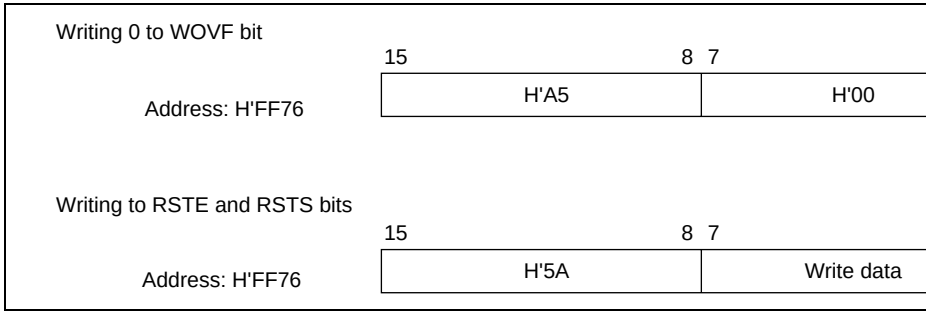


Figure 12.3 Format of Data Written to RSTCSR

Reading TCNT, TCSR, and RSTCSR (WDT0): These registers are read in the same other registers. The read addresses are H'FF74 for TCSR, H'FF75 for TCNT, and H'FF RSTCSR.

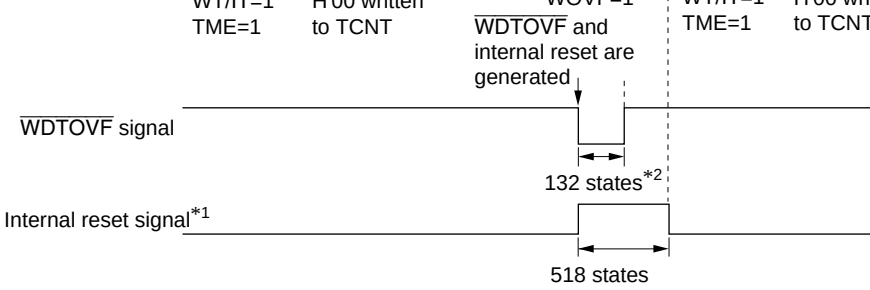
the WDT0 the WDTOVF signal is output. This is shown in figure 12.4 (a). This WDTC
can be used to reset the system. The $\overline{\text{WDTOVF}}$ signal is output for 132 states when RS
for 130 states when RSTE = 0.

If TCNT overflows when 1 is set in the RSTE bit in RSTCSR, a signal that resets the ch
internally is generated at the same time as the $\overline{\text{WDTOVF}}$ signal. This reset can be selec
power-on reset or a manual reset, depending on the setting of the RSTS bit in RSTCSR
internal reset signal is output for 518 states.

If a reset caused by a signal input to the $\overline{\text{RES}}$ pin occurs at the same time as a reset caus
WDT overflow, the $\overline{\text{RES}}$ pin reset has priority and the WOVF bit in RSTCSR is cleared

In the case of WDT1, the chip is reset, or an NMI interrupt request is generated, for 51
clock periods (516ϕ) (515 or 516 states when the clock source is ϕSUB (PSS = 1)). Th
illustrated in figure 12.4 (b).

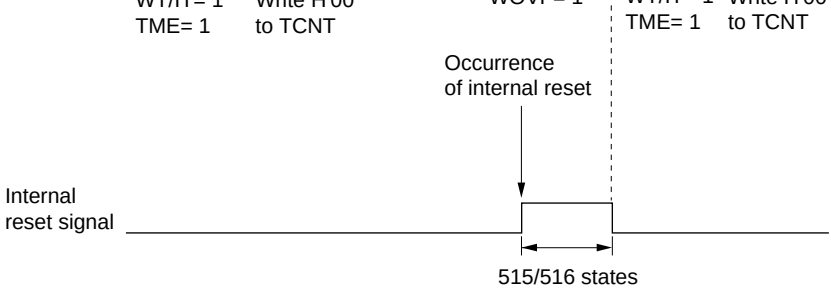
An NMI request from the watchdog timer and an interrupt request from the NMI pin ar
treated as having the same vector. So, avoid handling an NMI request from the watchd
and an interrupt request from the NMI pin at the same time.



Legend:
WT/$\overline{IT}$: Timer mode select bit
TME : Timer enable bit

Notes: 1. The internal reset signal is generated only if the RSTE bit is set to 1.
2. 130 states when the RSTE bit is cleared to 0.

Figure 12.4 (a) WDT0 Watchdog Timer Operation



WT/$\overline{IT}$: Timer Mode Select bit
TME : Timer Enable bit

Note: * The WOVF bit is set to 1 and then cleared to 0 by an internal reset.

Figure 12.4 (b) WDT1 Operation in Watchdog Timer Mode

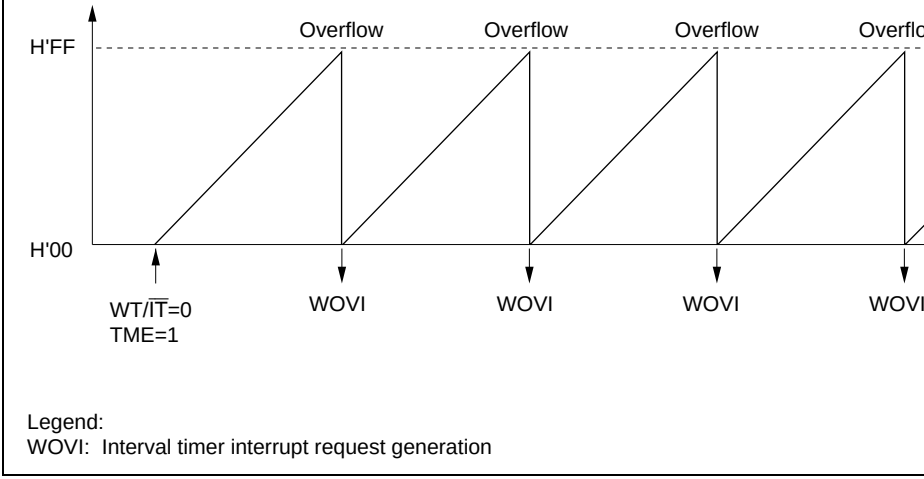


Figure 12.5 Interval Timer Operation

12.3.3 Timing of Setting Overflow Flag (OVF)

The OVF flag is set to 1 if TCNT overflows during interval timer operation. At the sa
interval timer interrupt (WOVI) is requested. This timing is shown in figure 12.6.

With WDT1, the OVF bit of the TCSR is set to 1 and a simultaneous NMI interrupt is
when the TCNT overflows if the NMI request has been chosen in the watchdog timer

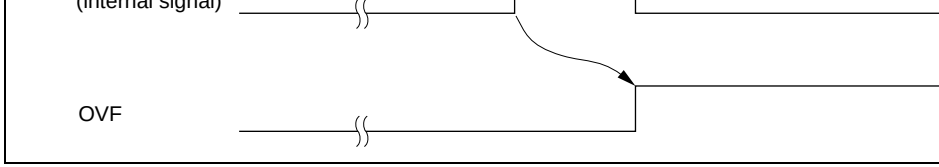


Figure 12.6 Timing of Setting of OVF

12.3.4 Timing of Setting of Watchdog Timer Overflow Flag (WOVF)

In the WDT0, the WOVF flag is set to 1 if TCNT overflows during watchdog timer ope
the same time, the $\overline{\text{WDTOVF}}$ signal goes low. If TCNT overflows while the RSTE bit i
is set to 1, an internal reset signal is generated for the entire chip. Figure 12.7 shows the
this case.

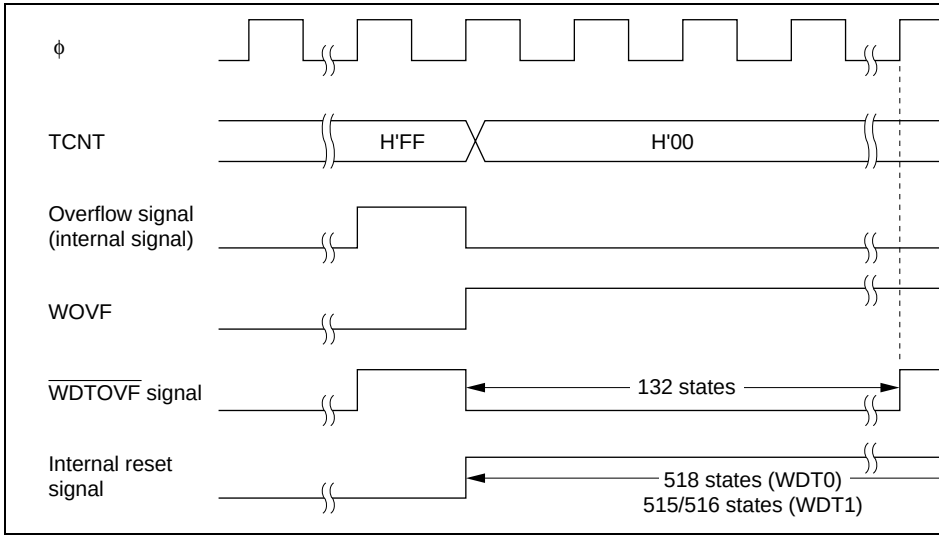


Figure 12.7 Timing of Setting of WOVF

12.5 Usage Notes

12.5.1 Contention between Timer Counter (TCNT) Write and Increment

If a timer counter clock pulse is generated during the T_2 state of a TCNT write cycle, t
takes priority and the timer counter is not incremented. Figure 12.8 shows this operati

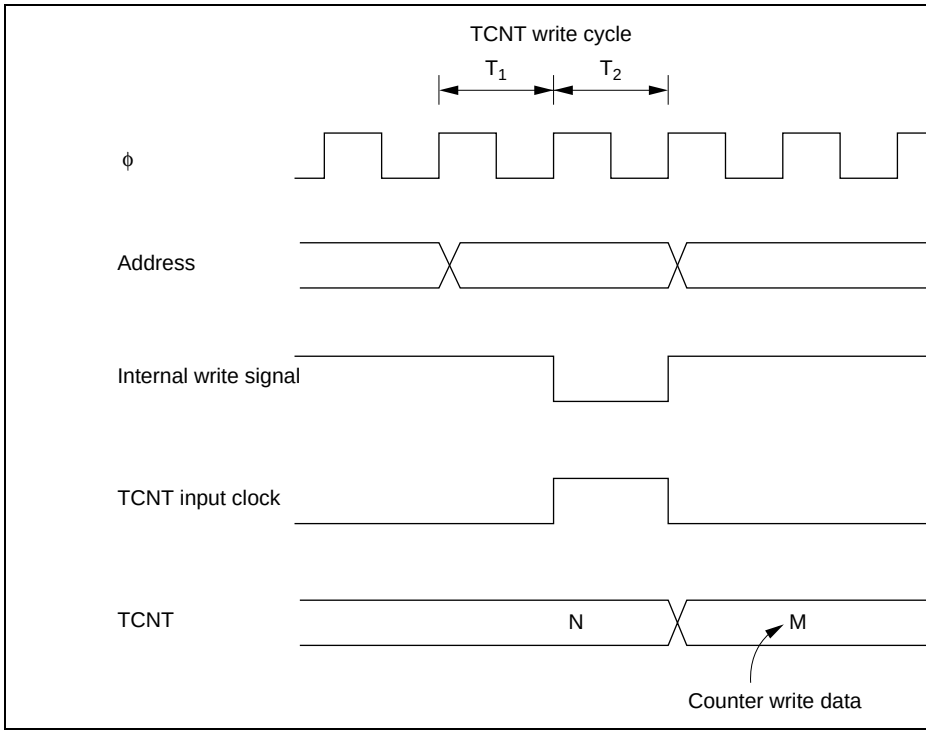


Figure 12.8 Contention between TCNT Write and Increment

If the mode is switched from watchdog timer to interval timer, or vice versa, while the
operating, errors could occur in the incrementation. Software must stop the watchdog ti
clearing the TME bit to 0) before switching the mode.

12.5.4 System Reset by $\overline{\text{WDTOVF}}$ Signal

If the $\overline{\text{WDTOVF}}$ output signal is input to the $\overline{\text{RES}}$ pin of the H8S/2626 Group or H8S/2
the chip will not be initialized correctly. Make sure that the $\overline{\text{WDTOVF}}$ signal is not inp
to the $\overline{\text{RES}}$ pin. To reset the entire system by means of the $\overline{\text{WDTOVF}}$ signal, use the ci
in figure 12.9.

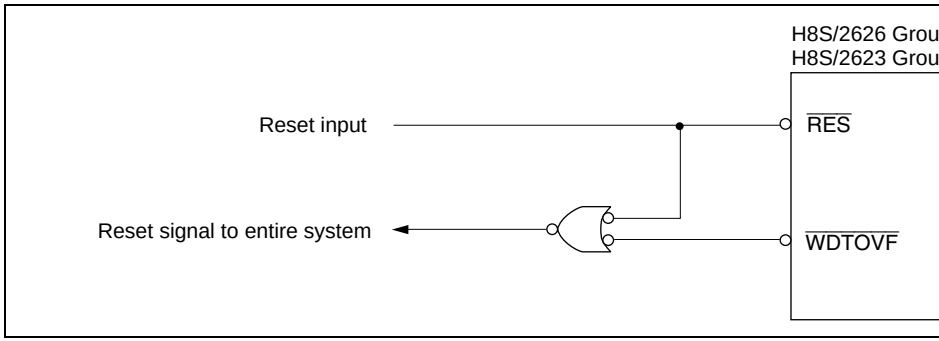


Figure 12.9 Circuit for System Reset by $\overline{\text{WDTOVF}}$ Signal (Example)

12.5.5 Internal Reset in Watchdog Timer Mode

The H8S/2626 Group or H8S/2623 Group is not reset internally if TCNT overflows wh
RSTE bit is cleared to 0 during watchdog timer operation, but TCNT and TCSR of the
reset.

TCNT, TCSR, and RSTCSR cannot be written to while the $\overline{\text{WDTOVF}}$ signal is low. Al
that a read of the WOVF flag is not recognized during this period. To clear the WOVF
therefore, read TCSR after the $\overline{\text{WDTOVF}}$ signal goes high, then write 0 to the WOVF f

(multiprocessor communication function).

13.1.1 Features

SCI features are listed below.

- Choice of asynchronous or clocked synchronous serial communication mode

  Asynchronous mode

  — Serial data communication executed using asynchronous system in which sync is achieved character by character

  Serial data communication can be carried out with standard asynchronous com chips such as a Universal Asynchronous Receiver/Transmitter (UART) or Asy Communication Interface Adapter (ACIA)

  — A multiprocessor communication function is provided that enables serial data communication with a number of processors

  — Choice of 12 serial data transfer formats

Data length	:	7 or 8 bits
Stop bit length	:	1 or 2 bits
Parity	:	Even, odd, or none
Multiprocessor bit	:	1 or 0

  — Receive error detection : Parity, overrun, and framing errors

  — Break detection : Break can be detected by reading the RxD pin level case of a framing error

  Clocked Synchronous mode

  — Serial data communication synchronized with a clock

  Serial data communication can be carried out with other chips that have a sync communication function

  — One serial data transfer format

Data length	:	8 bits

  — Receive error detection : Overrun errors detected

Note: * Descriptions in this section refer to LSB-first transfer.

- On-chip baud rate generator allows any bit rate to be selected
- Choice of serial clock source: internal clock from baud rate generator or external cl SCK pin
- Four interrupt sources
  - Four interrupt sources — transmit-data-empty, transmit-end, receive-data-full, a error — that can issue requests independently
  - The transmit-data-empty interrupt and receive data full interrupts can activate th transfer controller (DTC) to execute data transfer
- Module stop mode can be set
  - As the initial setting, SCI operation is halted. Register access is enabled by exiti stop mode.

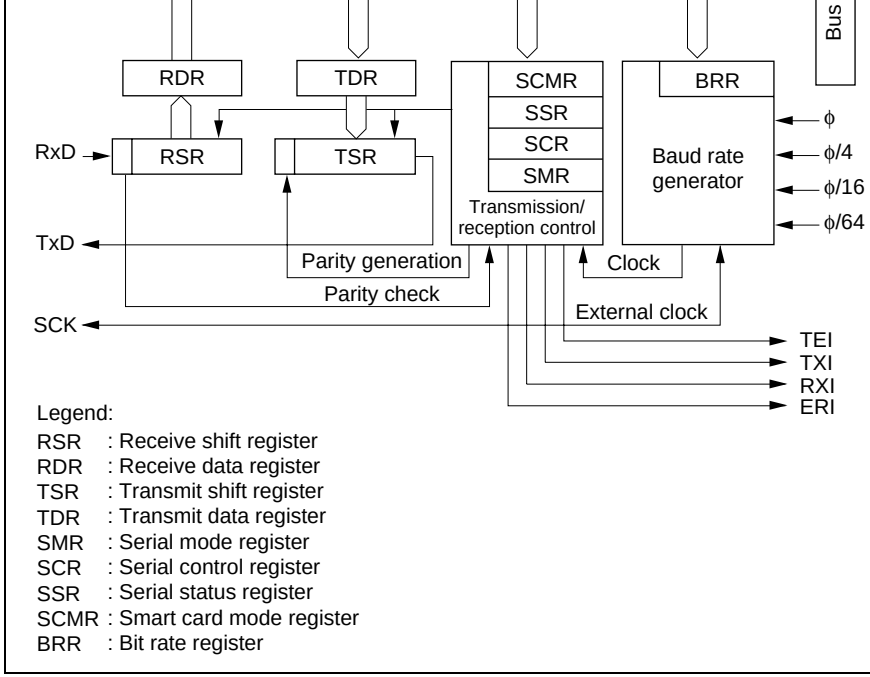


Figure 13.1 Block Diagram of SCI

	Transmit data pin 0	TxD0	Output	SCI0 transmit data outpu
1	Serial clock pin 1	SCK1	I/O	SCI1 clock input/output
	Receive data pin 1	RxD1	Input	SCI1 receive data input
	Transmit data pin 1	TxD1	Output	SCI1 transmit data outpu
2	Serial clock pin 2	SCK2	I/O	SCI2 clock input/output
	Receive data pin 2	RxD2	Input	SCI2 receive data input
	Transmit data pin 2	TxD2	Output	SCI2 transmit data outpu

Note: * Pin names SCK, RxD, and TxD are used in the text for all channels, omitting channel designation.

0	Serial mode register 0	SMR0	R/W	H'00
	Bit rate register 0	BRR0	R/W	H'FF
	Serial control register 0	SCR0	R/W	H'00
	Transmit data register 0	TDR0	R/W	H'FF
	Serial status register 0	SSR0	R/(W)*2	H'84
	Receive data register 0	RDR0	R	H'00
	Smart card mode register 0	SCMR0	R/W	H'F2
1	Serial mode register 1	SMR1	R/W	H'00
	Bit rate register 1	BRR1	R/W	H'FF
	Serial control register 1	SCR1	R/W	H'00
	Transmit data register 1	TDR1	R/W	H'FF
	Serial status register 1	SSR1	R/(W)*2	H'84
	Receive data register 1	RDR1	R	H'00
	Smart card mode register 1	SCMR1	R/W	H'F2
2	Serial mode register 2	SMR2	R/W	H'00
	Bit rate register 2	BRR2	R/W	H'FF
	Serial control register 2	SCR2	R/W	H'00
	Transmit data register 2	TDR2	R/W	H'FF
	Serial status register 2	SSR2	R/(W)*2	H'84
	Receive data register 2	RDR2	R	H'00
	Smart card mode register 2	SCMR2	R/W	H'F2
All	Module stop control register B	MSTPCRB	R/W	H'FF

Notes: 1. Lower 16 bits of the address.
2. Only 0 can be written, for flag clearing.

RSR is a register used to receive serial data.

The SCI sets serial data input from the RxD pin in RSR in the order received, starting v
LSB (bit 0), and converts it to parallel data. When one byte of data has been received, i
transferred to RDR automatically.

RSR cannot be directly read or written to by the CPU.

13.2.2 Receive Data Register (RDR)

Bit	:	7	6	5	4	3	2	1
Initial value	:	0	0	0	0	0	0	0
R/W	:	R	R	R	R	R	R	R

RDR is a register that stores received serial data.

When the SCI has received one byte of serial data, it transfers the received serial data f
RDR where it is stored, and completes the receive operation. After this, RSR is receive

Since RSR and RDR function as a double buffer in this way, enables continuous receiv
operations to be performed.

RDR is a read-only register, and cannot be written to by the CPU.

RDR is initialized to H'00 by a reset, in standby mode, watch mode, subactive mode, su
mode, or module stop mode.

To perform serial data transmission, the SCI first transfers transmit data from TDR to
sends the data to the TxD pin starting with the LSB (bit 0).

When transmission of one byte is completed, the next transmit data is transferred from
TSR, and transmission started, automatically. However, data transfer from TDR to TS
performed if the TDRE bit in SSR is set to 1.

TSR cannot be directly read or written to by the CPU.

13.2.4 Transmit Data Register (TDR)

Bit	:	7	6	5	4	3	2	1
Initial value	:	1	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TDR is an 8-bit register that stores data for serial transmission.

When the SCI detects that TSR is empty, it transfers the transmit data written in TDR
starts serial transmission. Continuous serial transmission can be carried out by writing
transmit data to TDR during serial transmission of the data in TSR.

TDR can be read or written to by the CPU at all times.

TDR is initialized to H'FF by a reset, in standby mode, watch mode, subactive mode, s
mode, or module stop mode.

generator clock source.

SMR can be read or written to by the CPU at all times.

SMR is initialized to H'00 by a reset and in hardware standby mode. It retains its previo
module stop mode, software standby mode, watch mode, subactive mode, and subsleep

Bit 7—Communication Mode (C/$\overline{A}$): Selects asynchronous mode or clocked synchro
as the SCI operating mode.

Bit 7

C/$\overline{A}$	Description	
0	Asynchronous mode	(In
1	Clocked synchronous mode	

Bit 6—Character Length (CHR): Selects 7 or 8 bits as the data length in asynchronou
clocked synchronous mode, a fixed data length of 8 bits is used regardless of the CHR s

Bit 6

CHR	Description	
0	8-bit data	(In
1	7-bit data*	

Note: * When 7-bit data is selected, the MSB (bit 7) of TDR is not transmitted, and it
possible to choose between LSB-first or MSB-first transfer.

Bit 5—Parity Enable (PE): In asynchronous mode, selects whether or not parity bit ad
performed in transmission, and parity bit checking in reception. In clocked synchronou
with a multiprocessor format, parity bit addition and checking is not performed, regardl
PE bit setting.

Bit 4—Parity Mode (O/$\overline{E}$): Selects either even or odd parity for use in parity additio checking.

The O/$\overline{E}$ bit setting is only valid when the PE bit is set to 1, enabling parity bit additio checking, in asynchronous mode. The O/$\overline{E}$ bit setting is invalid in clocked synchronou when parity addition and checking is disabled in asynchronous mode, and when a mul format is used.

Bit 4

O/$\overline{E}$	Description
0	Even parity*1 (I
1	Odd parity*2

Notes: 1. When even parity is set, parity bit addition is performed in transmission so t number of 1 bits in the transmit character plus the parity bit is even.
In reception, a check is performed to see if the total number of 1 bits in the character plus the parity bit is even.
2. When odd parity is set, parity bit addition is performed in transmission so th number of 1 bits in the transmit character plus the parity bit is odd.
In reception, a check is performed to see if the total number of 1 bits in the character plus the parity bit is odd.

Bit 3—Stop Bit Length (STOP): Selects 1 or 2 bits as the stop bit length in asynchro The STOP bits setting is only valid in asynchronous mode. If clocked synchronous mo STOP bit setting is invalid since stop bits are not added.

Bit 3

STOP	Description
0	1 stop bit: In transmission, a single 1 bit (stop bit) is added to the end of a transmit character before it is sent. (I
1	2 stop bits: In transmission, two 1 bits (stop bits) are added to the end of a character before it is sent.

Communication Function.

Bit 2	
MP	**Description**
0	Multiprocessor function disabled (In
1	Multiprocessor format selected

Bits 1 and 0—Clock Select 1 and 0 (CKS1, CKS0): These bits select the clock source baud rate generator. The clock source can be selected from φ, φ/4, φ/16, and φ/64, accor setting of bits CKS1 and CKS0.

For the relation between the clock source, the bit rate register setting, and the baud rate section 13.2.8, Bit Rate Register (BRR).

Bit 1	Bit 0	
CKS1	**CKS0**	**Description**
0	0	φ clock (In
	1	φ/4 clock
1	0	φ/16 clock
	1	φ/64 clock

SCR is a register that performs enabling or disabling of SCI transfer operations, serial
in asynchronous mode, and interrupt requests, and selection of the serial clock source.

SCR can be read or written to by the CPU at all times.

SCR is initialized to H'00 by a reset and in hardware standby mode. It retains its previ
module stop mode, software standby mode, watch mode, subactive mode, and subslee

Bit 7—Transmit Interrupt Enable (TIE): Enables or disables transmit data empty i
(TXI) request generation when serial transmit data is transferred from TDR to TSR an
flag in SSR is set to 1.

Bit 7

TIE	Description	
0	Transmit data empty interrupt (TXI) requests disabled	(I
1	Transmit data empty interrupt (TXI) requests enabled	

Note: TXI interrupt request cancellation can be performed by reading 1 from the TDR
clearing it to 0, or clearing the TIE bit to 0.

1	Receive data full interrupt (RXI) request and receive error interrupt (ERI) re enabled

Note: * RXI and ERI interrupt request cancellation can be performed by reading 1 fr RDRF flag, or the FER, PER, or ORER flag, then clearing the flag to 0, or cl RIE bit to 0.

Bit 5—Transmit Enable (TE): Enables or disables the start of serial transmission by t

Bit 5

TE	Description
0	Transmission disabled*1 (In
1	Transmission enabled*2

Notes: 1. The TDRE flag in SSR is fixed at 1.
2. In this state, serial transmission is started when transmit data is written to TD TDRE flag in SSR is cleared to 0.
SMR setting must be performed to decide the transfer format before setting to 1.

Bit 4—Receive Enable (RE): Enables or disables the start of serial reception by the SC

Bit 4

RE	Description
0	Reception disabled*1 (In
1	Reception enabled*2

Notes: 1. Clearing the RE bit to 0 does not affect the RDRF, FER, PER, and ORER fla retain their states.
2. Serial reception is started in this state when a start bit is detected in asynchr mode or serial clock input is detected in clocked synchronous mode.
SMR setting must be performed to decide the transfer format before setting to 1.

		[Clearing conditions] • When the MPIE bit is cleared to 0 • When MPB= 1 data is received
1		Multiprocessor interrupts enabled* Receive interrupt (RXI) requests, receive error interrupt (ERI) requests, ar of the RDRF, FER, and ORER flags in SSR are disabled until data with th multiprocessor bit set to 1 is received.

Note: * When receive data including MPB = 0 is received, receive data transfer fro RDR, receive error detection, and setting of the RDRF, FER, and ORER fla is not performed. When receive data including MPB = 1 is received, the MP is set to 1, the MPIE bit is cleared to 0 automatically, and generation of RXI interrupts (when the TIE and RIE bits in SCR are set to 1) and FER and OR setting is enabled.

Bit 2—Transmit End Interrupt Enable (TEIE): Enables or disables transmit end in (TEI) request generation when there is no valid transmit data in TDR in MSB data tra

Bit 2

TEIE	Description
0	Transmit end interrupt (TEI) request disabled* (I
1	Transmit end interrupt (TEI) request enabled*

Note: * TEI cancellation can be performed by reading 1 from the TDRE flag in SSR clearing it to 0 and clearing the TEND flag to 0, or clearing the TEIE bit to 0

SMR before setting the CKE1 and CKE0 bits.

For details of clock source selection, see table 13.9.

Bit 1	Bit 0		
CKE1	**CKE0**	**Description**	
0	0	Asynchronous mode	Internal clock/SCK pin functions as I/O p
		Clocked synchronous mode	Internal clock/SCK pin functions as seria output*[1]
	1	Asynchronous mode	Internal clock/SCK pin functions as cloc
		Clocked synchronous mode	Internal clock/SCK pin functions as seria output
1	0	Asynchronous mode	External clock/SCK pin functions as cloc
		Clocked synchronous mode	External clock/SCK pin functions as seri input
	1	Asynchronous mode	External clock/SCK pin functions as cloc
		Clocked synchronous mode	External clock/SCK pin functions as seri input

Notes: 1. Initial value
2. Outputs a clock of the same frequency as the bit rate.
3. Inputs a clock with a frequency 16 times the bit rate.

SSR is an 8-bit register containing status flags that indicate the operating status of the multiprocessor bits.

SSR can be read or written to by the CPU at all times. However, 1 cannot be written t TDRE, RDRF, ORER, PER, and FER. Also note that in order to clear these flags they read as 1 beforehand. The TEND flag and MPB flag are read-only flags and cannot be

SSR is initialized to H'84 by a reset, in standby mode, watch mode, subactive mode, s mode, or module stop mode.

Bit 7—Transmit Data Register Empty (TDRE): Indicates that data has been transfe TDR to TSR and the next serial data can be written to TDR.

Bit 7

TDRE	Description
0	[Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DTC is activated by a TXI interrupt and writes data to TDR
1	[Setting conditions] • When the TE bit in SCR is 0 • When data is transferred from TDR to TSR and data can be written to

When serial reception ends normally and receive data is transferred from R

Note: RDR and the RDRF flag are not affected and retain their previous values when a detected during reception or when the RE bit in SCR is cleared to 0.

If reception of the next data is completed while the RDRF flag is still set to 1, an error will occur and the receive data will be lost.

Bit 5—Overrun Error (ORER): Indicates that an overrun error occurred during recep causing abnormal termination.

Bit 5

ORER	Description
0	[Clearing condition] (Initi
	When 0 is written to ORER after reading ORER = 1
1	[Setting condition]
	When the next serial reception is completed while RDRF = 1*2

Notes: 1. The ORER flag is not affected and retains its previous state when the RE bit cleared to 0.

2. The receive data prior to the overrun error is retained in RDR, and the data r subsequently is lost. Also, subsequent serial reception cannot be continued ORER flag is set to 1. In clocked synchronous mode, serial transmission can continued, either.

	When the SCI checks whether the stop bit at the end of the receive data w reception ends, and the stop bit is 0 *2

Notes: 1. The FER flag is not affected and retains its previous state when the RE bit cleared to 0.

2. In 2-stop-bit mode, only the first stop bit is checked for a value of 0; the sec is not checked. If a framing error occurs, the receive data is transferred to R RDRF flag is not set. Also, subsequent serial reception cannot be continue FER flag is set to 1. In clocked synchronous mode, serial transmission can continued, either.

Bit 3—Parity Error (PER): Indicates that a parity error occurred during reception us addition in asynchronous mode, causing abnormal termination.

Bit 3

PER	Description
0	[Clearing condition] (In
	When 0 is written to PER after reading PER = 1
1	[Setting condition]
	When, in reception, the number of 1 bits in the receive data plus the parity match the parity setting (even or odd) specified by the O/$\overline{E}$ bit in SMR*2

Notes: 1. The PER flag is not affected and retains its previous state when the RE bit cleared to 0.

2. If a parity error occurs, the receive data is transferred to RDR but the RDRF set. Also, subsequent serial reception cannot be continued while the PER f 1. In clocked synchronous mode, serial transmission cannot be continued,

	• When 0 is written to TDRE after reading TDRE = 1
	• When the DTC is activated by a TXI interrupt and writes data to TDR
1	[Setting conditions] (In
	• When the TE bit in SCR is 0
	• When TDRE = 1 at transmission of the last bit of a 1-byte serial transmi

Bit 1—Multiprocessor Bit (MPB): When reception is performed using multiprocesso asynchronous mode, MPB stores the multiprocessor bit in the receive data.

MPB is a read-only bit, and cannot be modified.

Bit 1

MPB	Description
0	[Clearing condition] (Ini
	When data with a 0 multiprocessor bit is received
1	[Setting condition]
	When data with a 1 multiprocessor bit is received

Note: * Retains its previous state when the RE bit in SCR is cleared to 0 with multipr format.

Bit 0—Multiprocessor Bit Transfer (MPBT): When transmission is performed using multiprocessor format in asynchronous mode, MPBT stores the multiprocessor bit to be the transmit data.

The MPBT bit setting is invalid when multiprocessor format is not used, when not trans and in clocked synchronous mode.

Bit	:	7	6	5	4	3	2	1
Initial value	:	1	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BRR is an 8-bit register that sets the serial transfer bit rate in accordance with the bau generator operating clock selected by bits CKS1 and CKS0 in SMR.

BRR can be read or written to by the CPU at all times.

BRR is initialized to H'FF by a reset and in hardware standby mode. It retains its previ module stop mode, software standby mode, watch mode, subactive mode, and subslee

As baud rate generator control is performed independently for each channel, different be set for each channel.

Table 13.3 shows sample BRR settings in asynchronous mode, and table 13.4 shows s settings in clocked synchronous mode.

600	0	207	0.16	0	255	0.00	1	64	0.16
1200	0	103	0.16	0	127	0.00	0	129	0.16
2400	0	51	0.16	0	63	0.00	0	64	0.16
4800	0	25	0.16	0	31	0.00	0	32	–1.36
9600	0	12	0.16	0	15	0.00	0	15	1.73
19200	—	—	—	0	7	0.00	0	7	1.73
31250	0	3	0.00	0	4	–1.70	0	4	0.00
38400	—	—	—	0	3	0.00	0	3	1.73

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

1200	0	255	0.00	1	64	0.16	1	77	0.16	1	79
2400	0	127	0.00	0	129	0.16	0	155	0.16	0	159
4800	0	63	0.00	0	64	0.16	0	77	0.16	0	79
9600	0	31	0.00	0	32	–1.36	0	38	0.16	0	39
19200	0	15	0.00	0	15	1.73	0	19	–2.34	0	19
31250	0	9	–1.70	0	9	0.00	0	11	0.00	0	11
38400	0	7	0.00	0	7	1.73	0	9	–2.34	0	9

	ϕ = 14 MHz			**ϕ = 14.7456 MHz**			**ϕ = 16 MHz**			**ϕ = 17**	
Bit Rate (bit/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**
110	2	248	–0.17	3	64	0.70	3	70	0.03	3	75
150	2	181	0.13	2	191	0.00	2	207	0.13	2	223
300	2	90	0.13	2	95	0.00	2	103	0.13	2	111
600	1	181	0.13	1	191	0.00	1	207	0.13	1	223
1200	1	90	0.13	1	95	0.00	1	103	0.13	1	111
2400	0	181	0.13	0	191	0.00	0	207	0.13	0	223
4800	0	90	0.13	0	95	0.00	0	103	0.13	0	111
9600	0	45	–0.93	0	47	0.00	0	51	0.13	0	55
19200	0	22	–0.93	0	23	0.00	0	25	0.13	0	27
31250	0	13	0.00	0	14	–1.70	0	15	0.00	0	13
38400	—	—	—	0	11	0.00	0	12	0.13	0	13

1200	1	116	0.16	1	127	0.00	1	129	0.16
2400	0	233	0.16	0	255	0.00	1	64	0.16
4800	0	116	0.16	0	127	0.00	0	129	0.16
9600	0	58	–0.69	0	63	0.00	0	64	0.16
19200	0	28	1.02	0	31	0.00	0	32	–1.36
31250	0	17	0.00	0	19	–1.70	0	19	0.00
38400	0	14	–2.34	0	15	0.00	0	15	1.73

2.5 k	1	99	1	199	1	249	2	99	2
5 k	0	199	1	99	1	124	1	199	1
10 k	0	99	0	199	0	249	1	99	1
25 k	0	39	0	79	0	99	0	159	0
50 k	0	19	0	39	0	49	0	79	0
100 k	0	9	0	19	0	24	0	39	0
250 k	0	3	0	7	0	9	0	15	0
500 k	0	1	0	3	0	4	0	7	0
1 M	0	0*	0	1			0	3	0
2.5 M					0	0*			0
5 M									0

Legend:

Blank: Cannot be set.

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

*: Continuous transfer is not possible.

Note: As far as possible, the setting should be made so that the error is no more than

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

Where B: Bit rate (bit/s)
N: BRR setting for baud rate generator ($0 \leq N \leq 255$)
ϕ: Operating frequency (MHz)
n: Baud rate generator input clock (n = 0 to 3)
(See the table below for the relation between n and the clock.)

		SMR Setting	
n	**Clock**	**CKS1**	**CKS0**
0	ϕ	0	0
1	ϕ/4	0	1
2	ϕ/16	1	0
3	ϕ/64	1	1

The bit rate error in asynchronous mode is found from the following formula:

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

3	150250	0	0
6	187500	0	0
6.144	192000	0	0
7.3728	230400	0	0
8	250000	0	0
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
19.6608	614400	0	0
20	625000	0	0

7.3728	1.8432	115200
8	2.0000	125000
9.8304	2.4576	153600
10	2.5000	156250
12	3.0000	187500
12.288	3.0720	192000
14	3.5000	218750
14.7456	3.6864	230400
16	4.0000	250000
17.2032	4.3008	268800
18	4.5000	281250
19.6608	4.9152	307200
20	5.0000	312500

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

ϕ (MHz)	**External Input Clock (MHz)**	**Maximum Bit Rate (**
4	0.6667	666666.7
6	1.0000	1000000.0
8	1.3333	1333333.3
10	1.6667	1666666.7
12	2.0000	2000000.0
14	2.3333	2333333.3
16	2.6667	2666666.7
18	3.0000	3000000.0
20	3.3333	3333333.3

mode 7-bit data, LSB-first or MSB-first can be selected regardless of the serial comm
mode. The descriptions in this chapter refer to LSB-first transfer.

For details of the other bits in SCMR, see 14.2.1, Smart Card Mode Register (SCMR)

SCMR is initialized to H'F2 by a reset and in hardware standby mode. It retains its pre
in module stop mode, software standby mode, watch mode, subactive mode, and subs

Bits 7 to 4—Reserved: These bits are always read as 1 and cannot be modified.

Bit 3—Smart Card Data Transfer Direction (SDIR): Selects the serial/parallel con
format.

This bit is valid when 8-bit data is used as the transmit/receive format.

Bit 3

SDIR	Description	
0	TDR contents are transmitted LSB-first	(
	Receive data is stored in RDR LSB-first	
1	TDR contents are transmitted MSB-first	
	Receive data is stored in RDR MSB-first	

1	TDR contents are inverted before being transmitted Receive data is stored in RDR in inverted form

Bit 1—Reserved: This bit is always read as 1 and cannot be modified.

Bit 0—Smart Card Interface Mode Select (SMIF): When the smart card interface op
normal SCI, 0 should be written in this bit.

Bit 0

SMIF	Description
0	Operates as normal SCI (smart card interface function disabled) (In
1	Smart card interface function enabled

13.2.10 Module Stop Control Register B (MSTPCRB)

MSTPCRB

Bit	:	7	6	5	4	3	2	1
		MSTPB7	MSTPB6	MSTPB5	MSTPB4	MSTPB3	MSTPB2	MSTPB1
Initial value	:	1	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MSTPCRB is 8-bit readable/writable registers that perform module stop mode control.

When one of bits MSTPB7 to MSTPB5 is set to 1, SCI0, SCI1, or SCI2, respectively, s
operation at the end of the bus cycle, and enters module stop mode. For details, see sec
21A.5, 21B.5, Module Stop Mode.

MSTPCRB is initialized to H'FF by a reset and in hardware standby mode. They are n
initialized in software standby mode.

Bit 6

MSTPB6	Description
0	SCI1 module stop mode is cleared
1	SCI1 module stop mode is set

Bit 5—Module Stop (MSTPB5): Specifies the SCI2 module stop mode.

Bit 5

MSTPB5	Description
0	SCI2 module stop mode is cleared
1	SCI2 module stop mode is set

Selection of asynchronous or clocked synchronous mode and the transmission format is using SMR as shown in table 13.8. The SCI clock is determined by a combination of th in SMR and the CKE1 and CKE0 bits in SCR, as shown in table 13.9.

Asynchronous Mode

- Data length: Choice of 7 or 8 bits
- Choice of parity addition, multiprocessor bit addition, and addition of 1 or 2 stop bi combination of these parameters determines the transfer format and character lengt
- Detection of framing, parity, and overrun errors, and breaks, during reception
- Choice of internal or external clock as SCI clock source
  - When internal clock is selected:

    The SCI operates on the baud rate generator clock and a clock with the same fre the bit rate can be output
  - When external clock is selected:

    A clock with a frequency of 16 times the bit rate must be input (the on-chip bau generator is not used)

Clocked Synchronous Mode

- Transfer format: Fixed 8-bit data
- Detection of overrun errors during reception
- Choice of internal or external clock as SCI clock source
  - When internal clock is selected:

    The SCI operates on the baud rate generator clock and a serial clock is output of
  - When external clock is selected:

    The on-chip baud rate generator is not used, and the SCI operates on the input s

			1	0				Yes
				1				
	1		0	0		7-bit data		No
				1				
			1	0				Yes
				1				
	0	1	—	0	Asynchronous mode (multi-processor format)	8-bit data	Yes	No
			—	1				
	1		—	0		7-bit data		
			—	1				
1	—	—	—	—	Clocked synchronous mode	8-bit data	No	

Table 13.9 SMR and SCR Settings and SCI Clock Source Selection

SMR	SCR Setting			SCI Transmit/Receive Clock	
Bit 7	Bit 1	Bit 0		Clock Source	SCK Pin Function
C/$\overline{A}$	CKE1	CKE0	Mode		
0	0	0	Asynchronous mode	Internal	SCI does not use SCK pin
		1			Outputs clock with same freq rate
	1	0		External	Inputs clock with frequency o the bit rate
		1			
1	0	0	Clocked synchronous mode	Internal	Outputs serial clock
		1			
	1	0		External	Inputs serial clock
		1			

transfer.

Figure 13.2 shows the general format for asynchronous serial communication.

In asynchronous serial communication, the transmission line is usually held in the mark level). The SCI monitors the transmission line, and when it goes to the space state (low recognizes a start bit and starts serial communication.

One serial communication character consists of a start bit (low level), followed by data first order), a parity bit (high or low level), and finally stop bits (high level).

In asynchronous mode, the SCI performs synchronization at the falling edge of the star reception. The SCI samples the data on the 8th pulse of a clock with a frequency of 16 length of one bit, so that the transfer data is latched at the center of each bit.

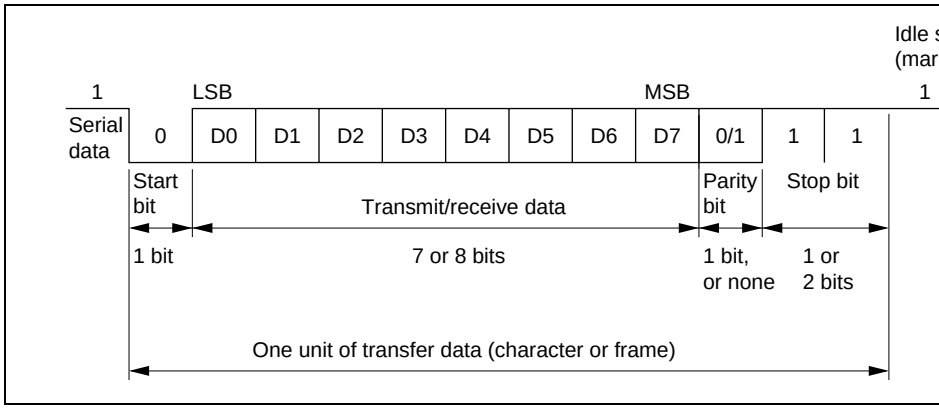
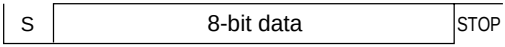
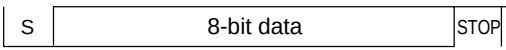
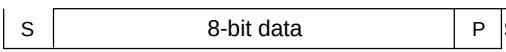
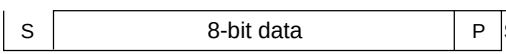
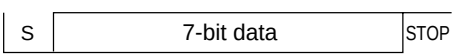
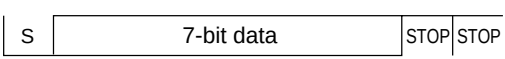
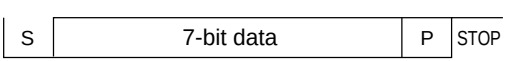
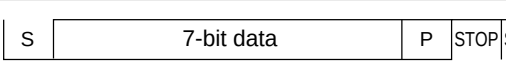
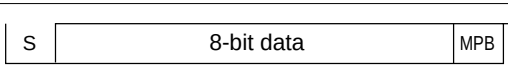
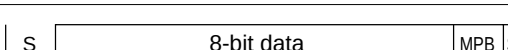
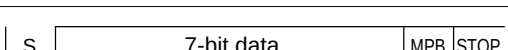
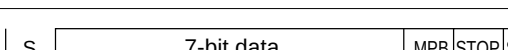


Figure 13.2 Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, Two Stop Bits)

0	0	0	0	
0	0	0	1	
0	1	0	0	
0	1	0	1	
1	0	0	0	
1	0	0	1	
1	1	0	0	
1	1	0	1	
0	—	1	0	
0	—	1	1	
1	—	1	0	
1	—	1	1	

Legend:

S : Start bit

STOP : Stop bit

P : Parity bit

MPB : Multiprocessor bit

When the SCI is operated on an internal clock, the clock can be output from the SCK p frequency of the clock output in this case is equal to the bit rate, and the phase is such t rising edge of the clock is in the middle of the transmit data, as shown in figure 13.3.

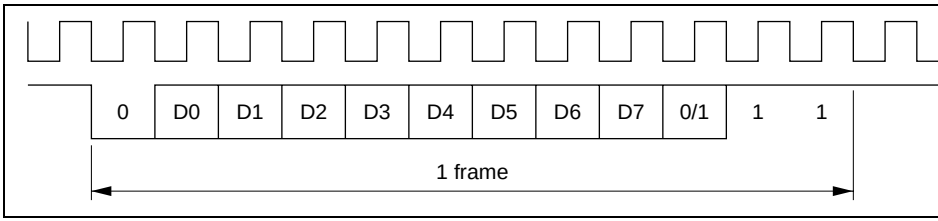


Figure 13.3 Relation between Output Clock and Transfer Data Phase (Asynchronous Mode)

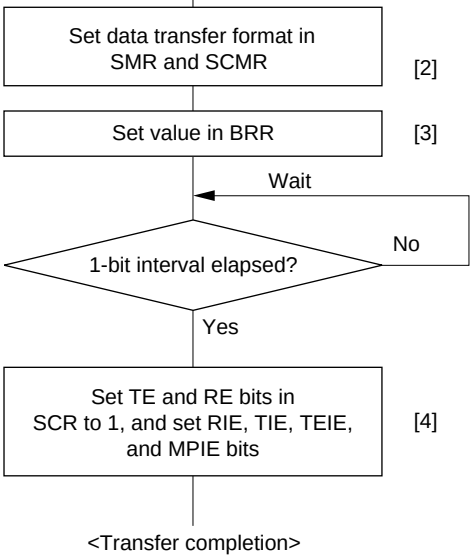
Data Transfer Operations

SCI initialization (asynchronous mode): Before transmitting and receiving data, you clear the TE and RE bits in SCR to 0, then initialize the SCI as described below.

When the operating mode, transfer format, etc., is changed, the TE and RE bits must be 0 before making the change using the following procedure. When the TE bit is cleared TDRE flag is set to 1 and TSR is initialized. Note that clearing the RE bit to 0 does not contents of the RDRF, PER, FER, and ORER flags, or the contents of RDR.

When an external clock is used the clock should not be stopped during operation, inclu initialization, since operation is uncertain.

Figure 13.4 shows a sample SCI initialization flowchart.



[2] Set the data transfer format i and SCMR.

[3] Write a value corresponding bit rate to BRR. Not necessa external clock is used.

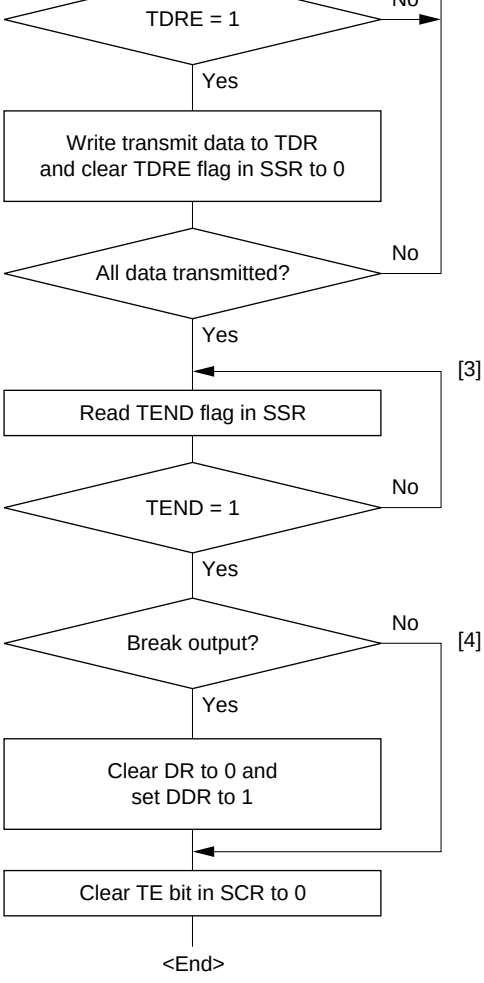
[4] Wait at least one bit interval, set the TE bit or RE bit in SC Also set the RIE, TIE, TEIE, a MPIE bits.

Setting the TE and RE bits e the TxD and RxD pins to be u

Figure 13.4 Sample SCI Initialization Flowchart

Serial data transmission (asynchronous mode): Figure 13.5 shows a sample flowch transmission.

The following procedure should be used for serial data transmission.



write:
Read SSR and check that the
TDRE flag is set to 1, then wri
transmit data to TDR and clea
TDRE flag to 0.

[3] Serial transmission continuatio
procedure:
To continue serial transmissio
read 1 from the TDRE flag to
confirm that writing is possible
then write data to TDR, and th
clear the TDRE flag to 0. Chec
and clearing of the TDRE flag
automatic when the DTC is
activated by a transmit data em
interrupt (TXI) request, and da
written to TDR.

[4] Break output at the end of seri
transmission:
To output a break in serial
transmission, set DDR for the
corresponding to the TxD pin t
clear DR to 0, then clear the T
in SCR to 0.

Figure 13.5 Sample Serial Transmission Flowchart

The serial transmit data is sent from the TXD pin in the following order.

[a] Start bit:

One 0-bit is output.

[b] Transmit data:

8-bit or 7-bit data is output in LSB-first order.

[c] Parity bit or multiprocessor bit:

One parity bit (even or odd parity), or one multiprocessor bit is output.

A format in which neither a parity bit nor a multiprocessor bit is output can als
selected.

[d] Stop bit(s):

One or two 1-bits (stop bits) are output.

[e] Mark state:

1 is output continuously until the start bit that starts the next transmission is se

[3] The SCI checks the TDRE flag at the timing for sending the stop bit.

If the TDRE flag is cleared to 0, the data is transferred from TDR to TSR, the stop
and then serial transmission of the next frame is started.

If the TDRE flag is set to 1, the TEND flag in SSR is set to 1, the stop bit is sent, a
“mark state” is entered in which 1 is output continuously. If the TEIE bit in SCR is
this time, a TEI interrupt request is generated.

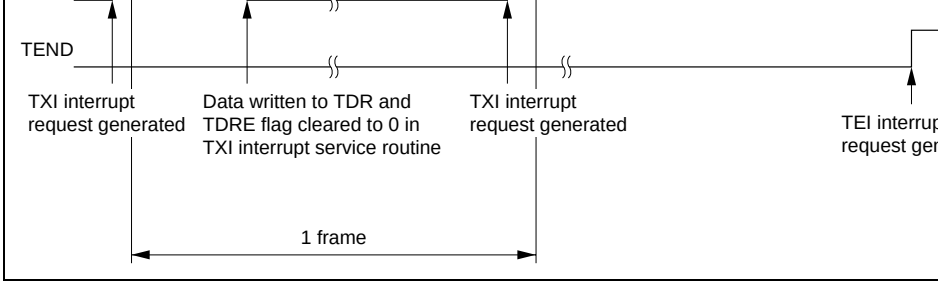
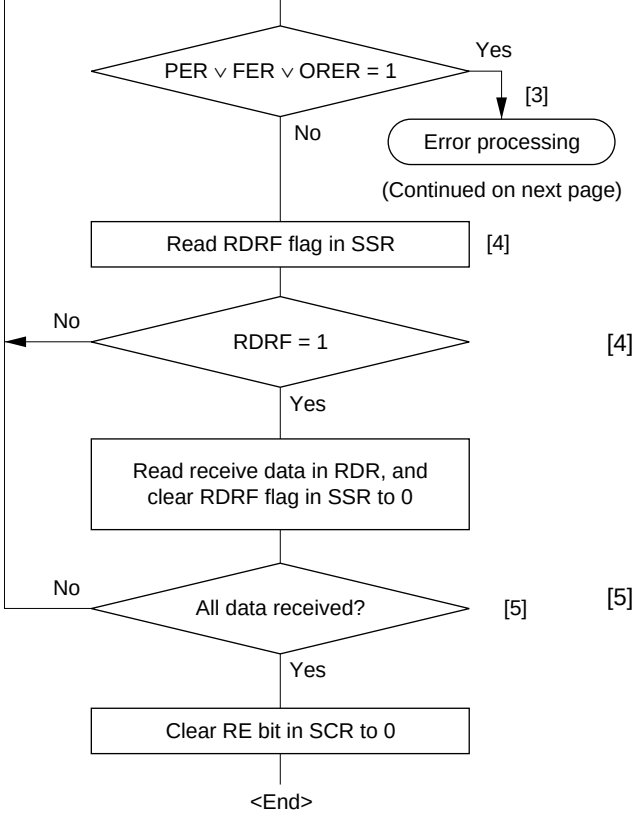


**Figure 13.6 Example of Operation in Transmission in Asynchronous Mo
(Example with 8-Bit Data, Parity, One Stop Bit)**

Serial data reception (asynchronous mode): Figure 13.7 shows a sample flowchart fo
reception.

The following procedure should be used for serial data reception.



SSR to identify the erro
performing the appropri
processing, ensure that
ORER, PER, and FER
all cleared to 0. Recepti
be resumed if any of the
are set to 1. In the case
framing error, a break c
detected by reading the
the input port correspon
the RxD pin.

[4] SCI status check and re
data read :
Read SSR and check th
= 1, then read the recei
RDR and clear the RDR
0. Transition of the RDR
from 0 to 1 can also be
by an RXI interrupt.

[5] Serial reception continu
procedure:
To continue serial recep
before the stop bit for th
frame is received, read
RDRF flag, read RDR,
the RDRF flag to 0. The
flag is cleared automati
when DTC is activated
interrupt and the RDR v
read.

Figure 13.7 Sample Serial Reception Data Flowchart

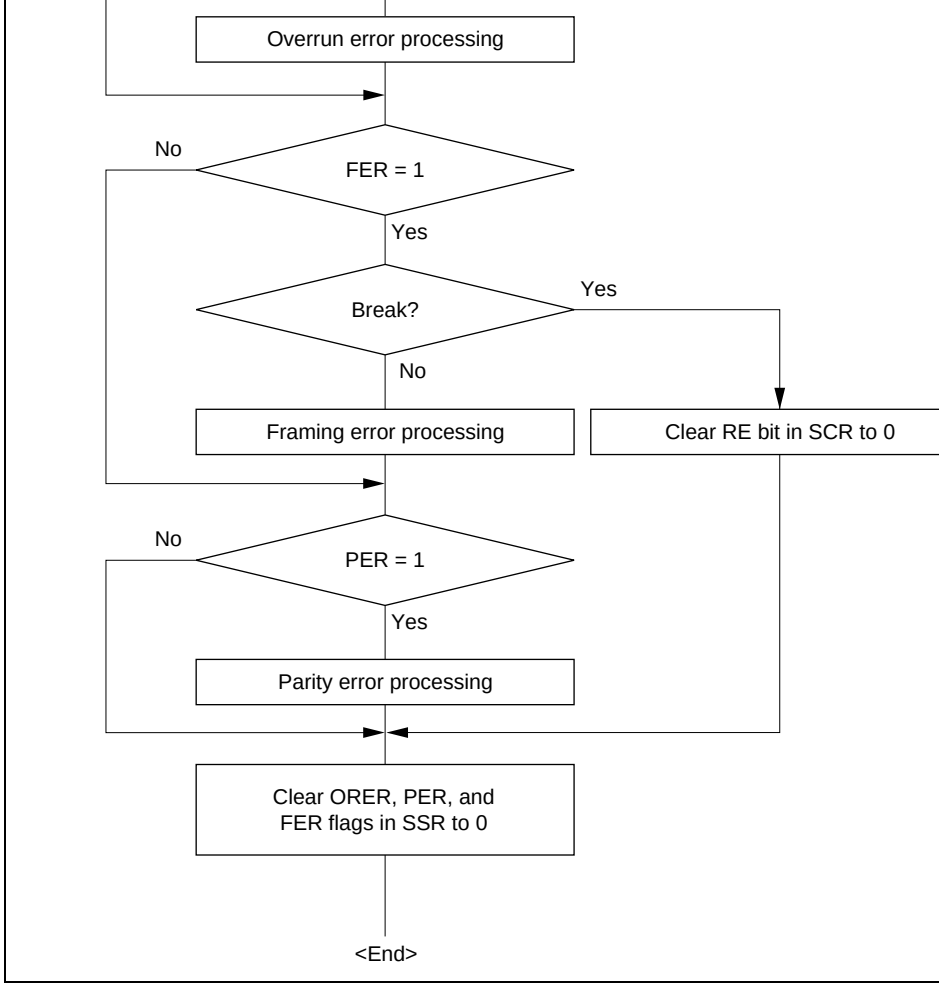


Figure 13.7 Sample Serial Reception Data Flowchart (cont)

[a] Parity check:

The SCI checks whether the number of 1 bits in the receive data agrees with th
(even or odd) set in the O/$\overline{E}$ bit in SMR.

[b] Stop bit check:

The SCI checks whether the stop bit is 1.

If there are two stop bits, only the first is checked.

[c] Status check:

The SCI checks whether the RDRF flag is 0, indicating that the receive data ca
transferred from RSR to RDR.

If all the above checks are passed, the RDRF flag is set to 1, and the receive data i
RDR.

If a receive error* is detected in the error check, the operation is as shown in table

Note: * Subsequent receive operations cannot be performed when a receive error h
Also note that the RDRF flag is not set to 1 in reception, and so the error fl
cleared to 0.

[4] If the RIE bit in SCR is set to 1 when the RDRF flag changes to 1, a receive data f
(RXI) request is generated.

Also, if the RIE bit in SCR is set to 1 when the ORER, PER, or FER flag changes
receive error interrupt (ERI) request is generated.

Figure 13.8 shows an example of the operation for reception in asynchronous mode.

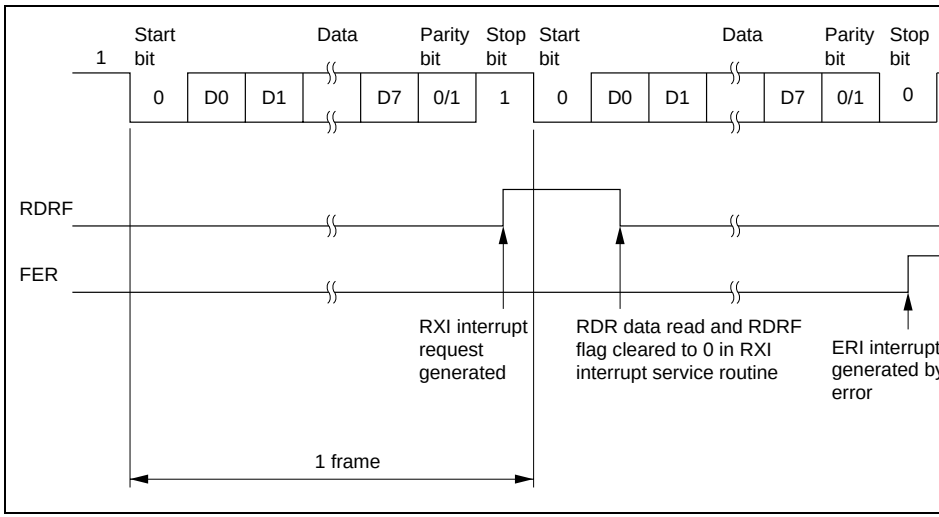


Figure 13.8 Example of SCI Operation in Reception (Example with 8-Bit Data, Parity, One Stop Bit)

The serial communication cycle consists of two component cycles: an ID transmission which specifies the receiving station , and a data transmission cycle. The multiprocess to differentiate between the ID transmission cycle and the data transmission cycle.

The transmitting station first sends the ID of the receiving station with which it wants serial communication as data with a 1 multiprocessor bit added. It then sends transmit with a 0 multiprocessor bit added.

The receiving station skips the data until data with a 1 multiprocessor bit is sent.

When data with a 1 multiprocessor bit is received, the receiving station compares that own ID. The station whose ID matches then receives the data sent next. Stations whos not match continue to skip the data until data with a 1 multiprocessor bit is again recei way, data communication is carried out among a number of processors.

Figure 13.9 shows an example of inter-processor communication using the multiproce

Data Transfer Format

There are four data transfer formats.

When the multiprocessor format is specified, the parity bit specification is invalid.

For details, see table 13.10.

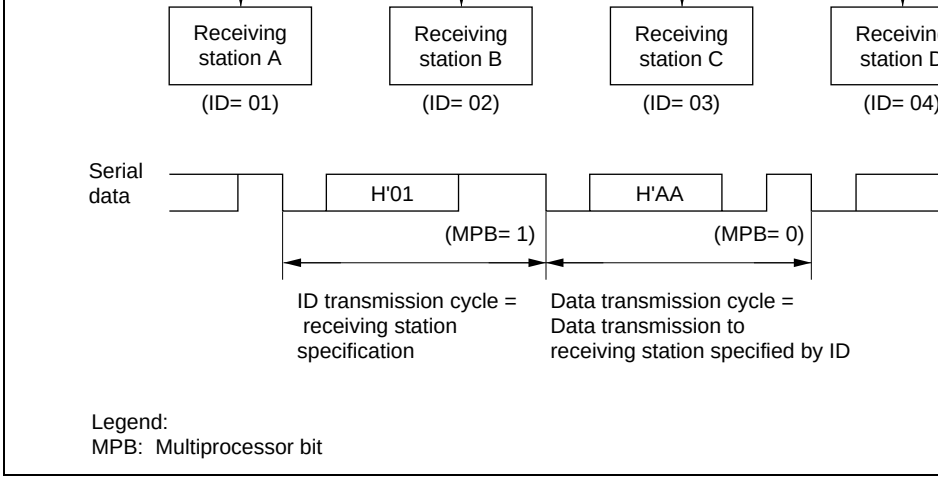
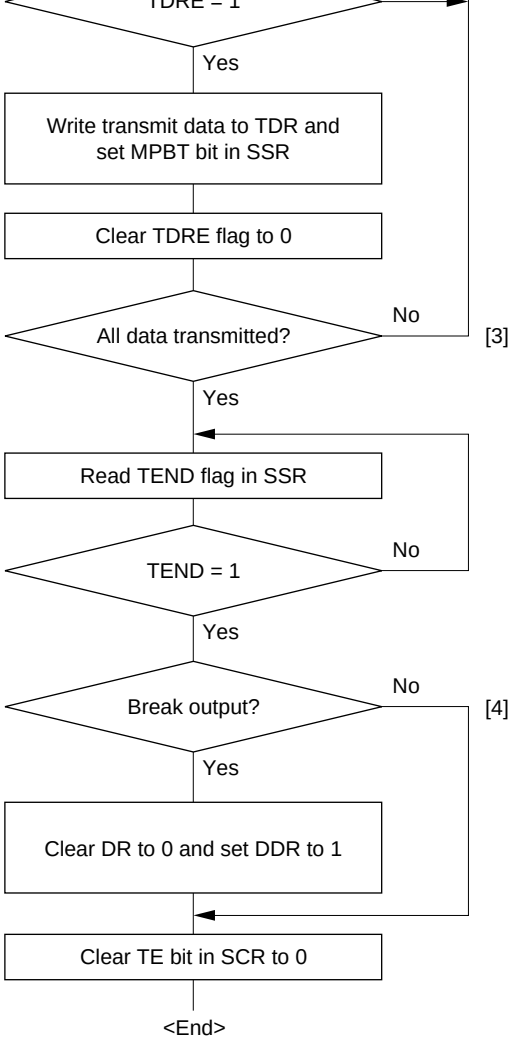


Figure 13.9 Example of Inter-Processor Communication Using Multiprocessor (Transmission of Data H'AA to Receiving Station A)

Data Transfer Operations

Multiprocessor serial data transmission: Figure 13.10 shows a sample flowchart for multiprocessor serial data transmission.

The following procedure should be used for multiprocessor serial data transmission.



data write.
Read SSR and check that t
TDRE flag is set to 1, then
transmit data to TDR. Set t
MPBT bit in SSR to 0 or 1.
Finally, clear the TDRE flag

[3] Serial transmission continu
procedure:
To continue serial transmis
be sure to read 1 from the
flag to confirm that writing i
possible, then write data to
and then clear the TDRE fl
0. Checking and clearing o
TDRE flag is automatic wh
DTC is activated by a trans
data empty interrupt (TXI)
request, and data is written
TDR.

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

Figure 13.10 Sample Multiprocessor Serial Transmission Flowchart

The serial transmit data is sent from the TXD pin in the following order.

[a] Start bit:

One 0-bit is output.

[b] Transmit data:

8-bit or 7-bit data is output in LSB-first order.

[c] Multiprocessor bit

One multiprocessor bit (MPBT value) is output.

[d] Stop bit(s):

One or two 1-bits (stop bits) are output.

[e] Mark state:

1 is output continuously until the start bit that starts the next transmission is sent

[3] The SCI checks the TDRE flag at the timing for sending the stop bit.

If the TDRE flag is cleared to 0, data is transferred from TDR to TSR, the stop bit i then serial transmission of the next frame is started.

If the TDRE flag is set to 1, the TEND flag in SSR is set to 1, the stop bit is sent, ar mark state is entered in which 1 is output continuously. If the TEIE bit in SCR is se time, a transmission end interrupt (TEI) request is generated.

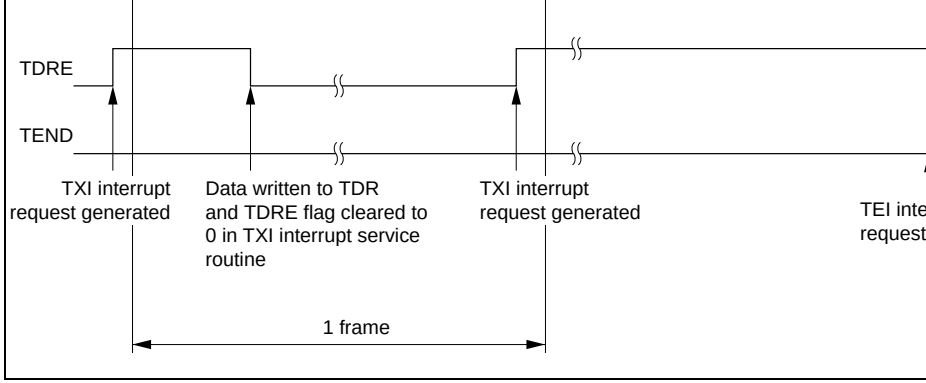
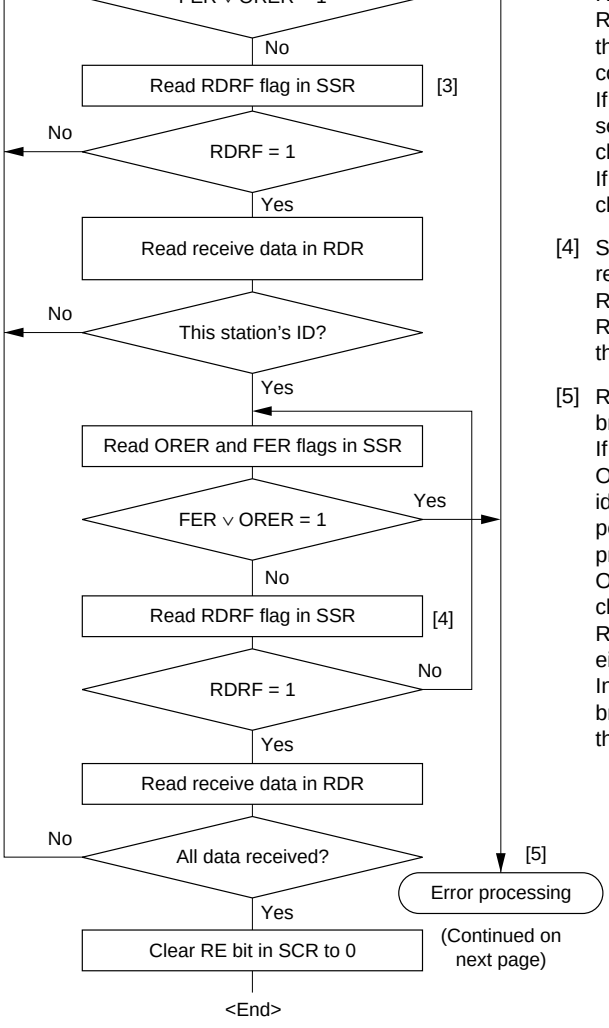


Figure 13.11 Example of SCI Operation in Transmission (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)

Multiprocessor serial data reception: Figure 13.12 shows a sample flowchart for multiprocessor serial reception.

The following procedure should be used for multiprocessor serial data reception.



RDRF flag is set to 1, then re
the receive data in RDR and
compare it with this station's
If the data is not this station's
set the MPIE bit to 1 again, a
clear the RDRF flag to 0.
If the data is this station's ID
clear the RDRF flag to 0.

[4] SCI status check and data reception:
Read SSR and check that th
RDRF flag is set to 1, then re
the data in RDR.

[5] Receive error processing an
break detection:
If a receive error occurs, rea
ORER and FER flags in SSR
identify the error. After
performing the appropriate e
processing, ensure that the
ORER and FER flags are all
cleared to 0.
Reception cannot be resume
either of these flags is set to
In the case of a framing erro
break can be detected by re
the RxD pin value.

Figure 13.12 Sample Multiprocessor Serial Reception Flowchart

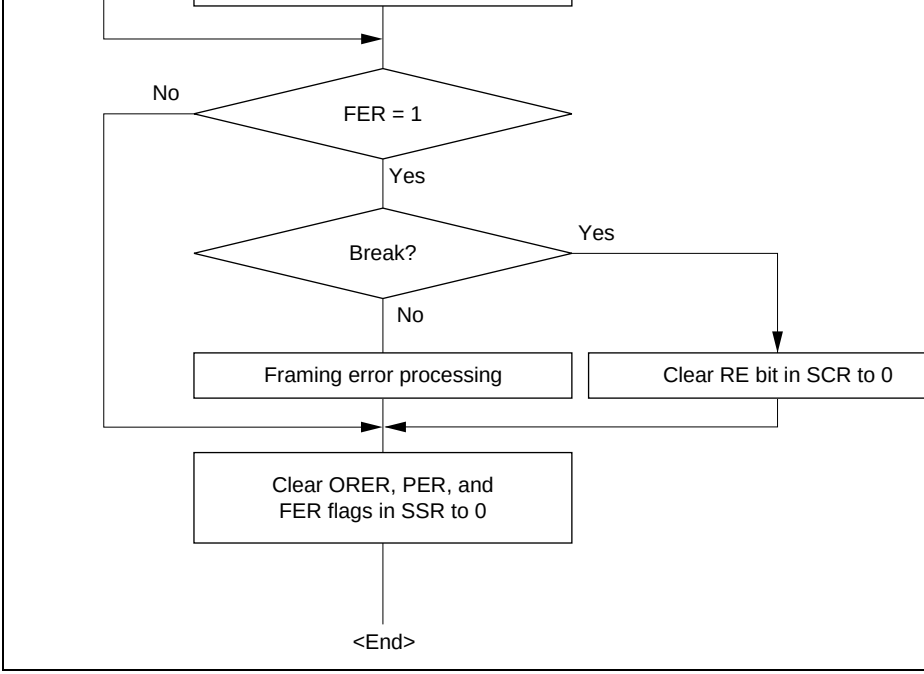


Figure 13.12 Sample Multiprocessor Serial Reception Flowchart (con

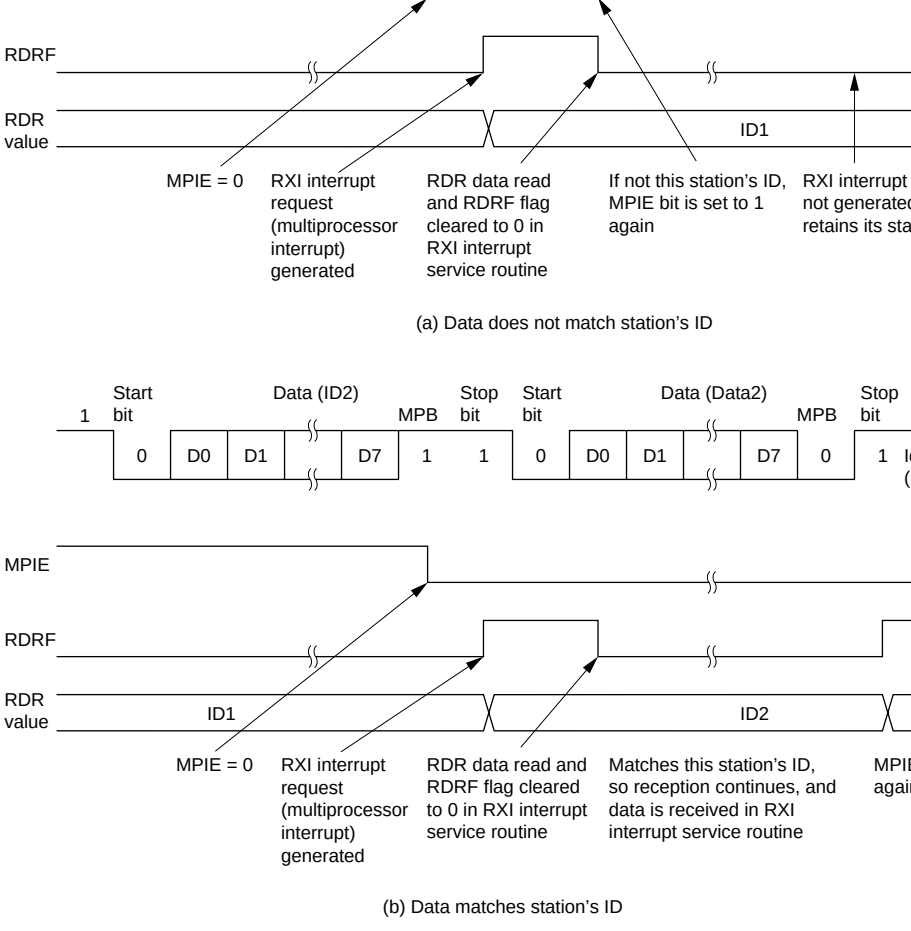


Figure 13.13 Example of SCI Operation in Reception (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)

Figure 13.14 shows the general format for clocked synchronous serial communication

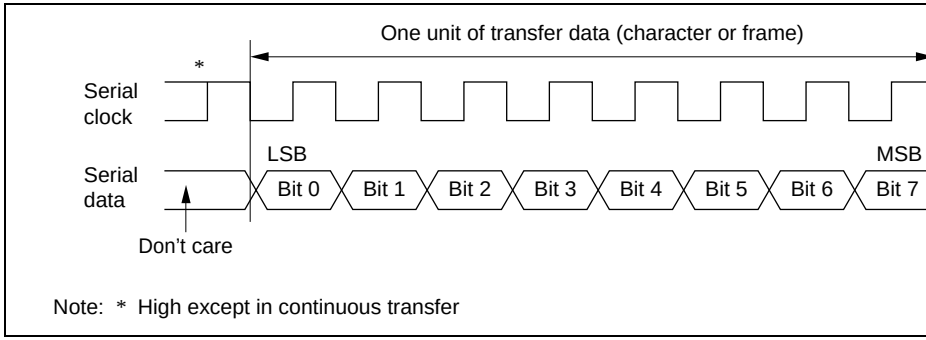


Figure 13.14 Data Format in Synchronous Communication

In clocked synchronous serial communication, data on the transmission line is output
falling edge of the serial clock to the next. Data confirmation is guaranteed at the risin
the serial clock.

In clocked serial communication, one character consists of data output starting with th
ending with the MSB. After the MSB is output, the transmission line holds the MSB s

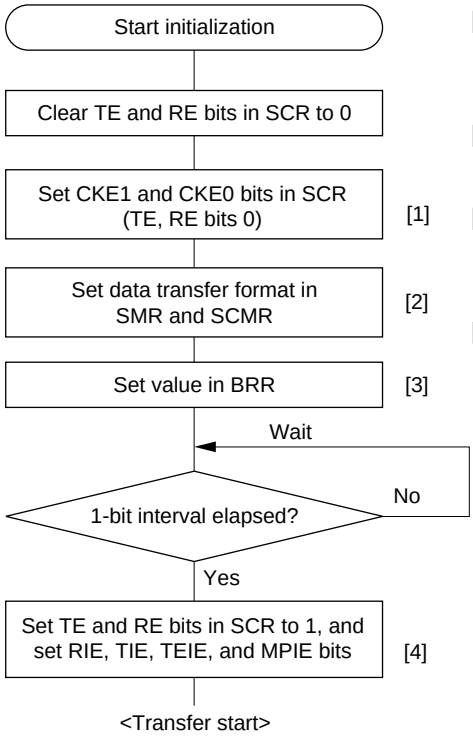
In clocked synchronous mode, the SCI receives data in synchronization with the rising
serial clock.

Either an internal clock generated by the on-chip baud rate generator or an external seri
input at the SCK pin can be selected, according to the setting of the C/$\overline{A}$ bit in SMR an
and CKE0 bits in SCR. For details of SCI clock source selection, see table 13.9.

When the SCI is operated on an internal clock, the serial clock is output from the SCK

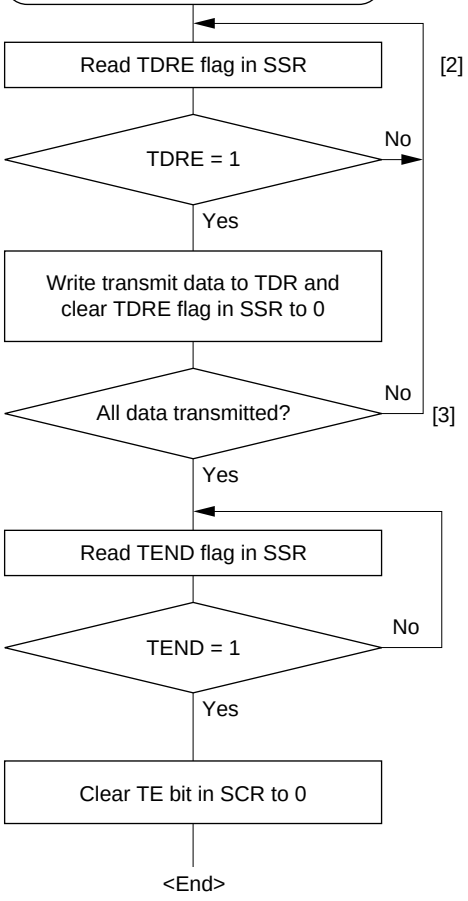
Eight serial clock pulses are output in the transfer of one character, and when no transfe
performed the clock is fixed high. When only receive operations are performed, howev
serial clock is output until an overrun error occurs or the RE bit is cleared to 0. If you w
perform receive operations in units of one character, you should select an external cloc
clock source.

Figure 13.15 shows a sample SCI initialization flowchart.



Note: In simultaneous transmit and receive operations, the TE and RE bits should both be to 0 or set to 1 simultaneously.

Figure 13.15 Sample SCI Initialization Flowchart



pin.

[2] SCI status check and transmit d
write:
Read SSR and check that the T
flag is set to 1, then write transm
to TDR and clear the TDRE flag

[3] Serial transmission continuation
procedure:
To continue serial transmission,
sure to read 1 from the TDRE fl
confirm that writing is possible,
write data to TDR, and then clea
TDRE flag to 0.
Checking and clearing of the TD
flag is automatic when the DTC
activated by a transmit data em
interrupt (TXI) request and data
written to TDR.

Figure 13.16 Sample Serial Transmission Flowchart

external clock has been specified, data is output synchronized with the input clock

The serial transmit data is sent from the TxD pin starting with the LSB (bit 0) and the MSB (bit 7).

[3] The SCI checks the TDRE flag at the timing for sending the MSB (bit 7).

If the TDRE flag is cleared to 0, data is transferred from TDR to TSR, and serial tr of the next frame is started.

If the TDRE flag is set to 1, the TEND flag in SSR is set to 1, the MSB (bit 7) is se TxD pin maintains its state.

If the TEIE bit in SCR is set to 1 at this time, a TEI interrupt request is generated.

[4] After completion of serial transmission, the SCK pin is fixed high.

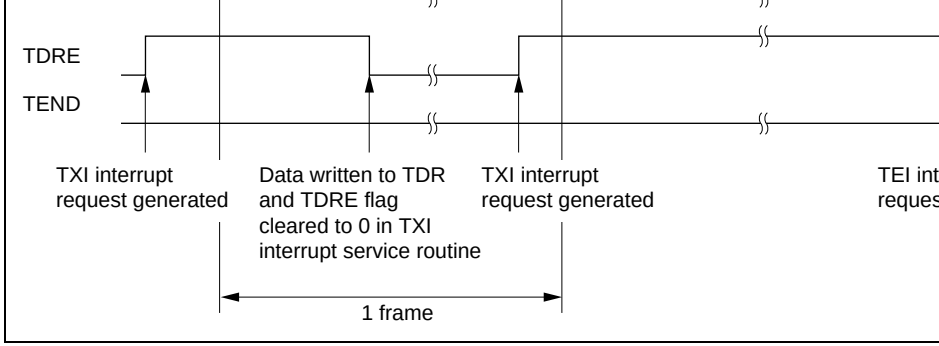


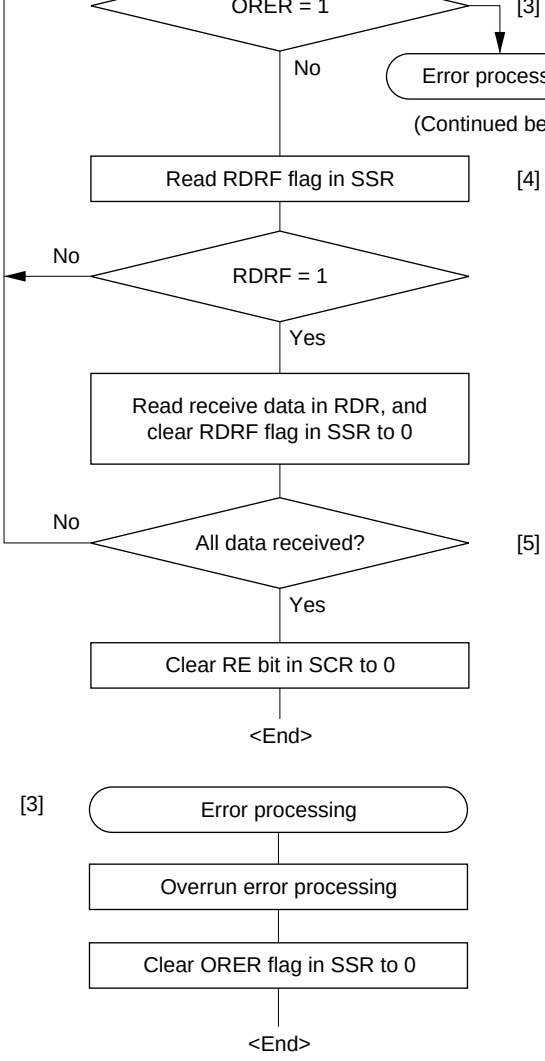
Figure 13.17 Example of SCI Operation in Transmission

Serial data reception (clocked synchronous mode): Figure 13.18 shows a sample flo serial reception.

The following procedure should be used for serial data reception.

When changing the operating mode from asynchronous to clocked synchronous, be sur that the ORER, PER, and FER flags are all cleared to 0.

The RDRF flag will not be set if the FER or PER flag is set to 1, and neither transmit n operations will be possible.



[3] processing, clear the C
to 0. Transfer cannot b
if the ORER flag is set

[4] SCI status check and r
data read:
Read SSR and check t
RDRF flag is set to 1, t
the receive data in RDI
clear the RDRF flag to
Transition of the RDRF
0 to 1 can also be iden
an RXI interrupt.

[5] Serial reception contin
procedure:
To continue serial rece
before the MSB (bit 7)
current frame is receiv
reading the RDRF flag
RDR, and clearing the
to 0. The RDRF flag is
automatically when the
activated by a receive
interrupt (RXI) request
RDR value is read.

Figure 13.18 Sample Serial Reception Flowchart

receive error is detected in the error check, the operation is as shown in table 13.11.

Neither transmit nor receive operations can be performed subsequently when a rece has been found in the error check.

[3] If the RIE bit in SCR is set to 1 when the RDRF flag changes to 1, a receive data fu (RXI) request is generated.

Also, if the RIE bit in SCR is set to 1 when the ORER flag changes to 1, a receive e interrupt (ERI) request is generated.

Figure 13.19 shows an example of SCI operation in reception.

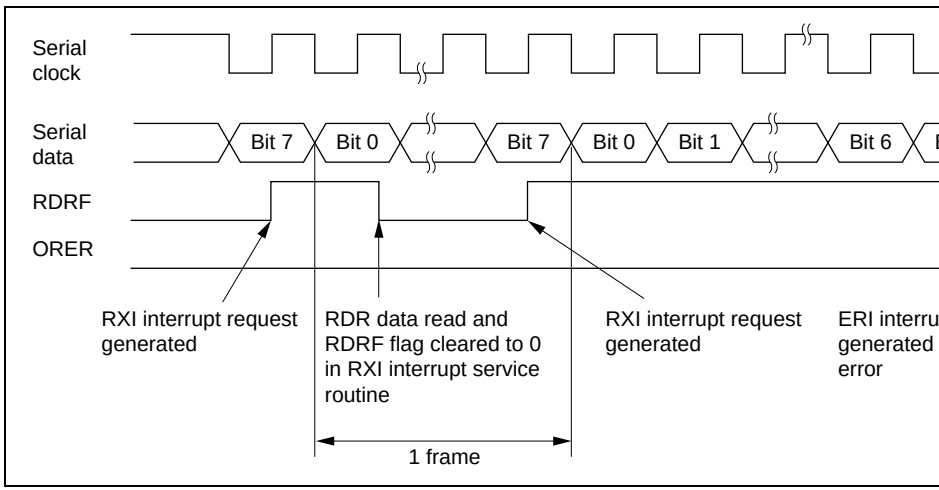
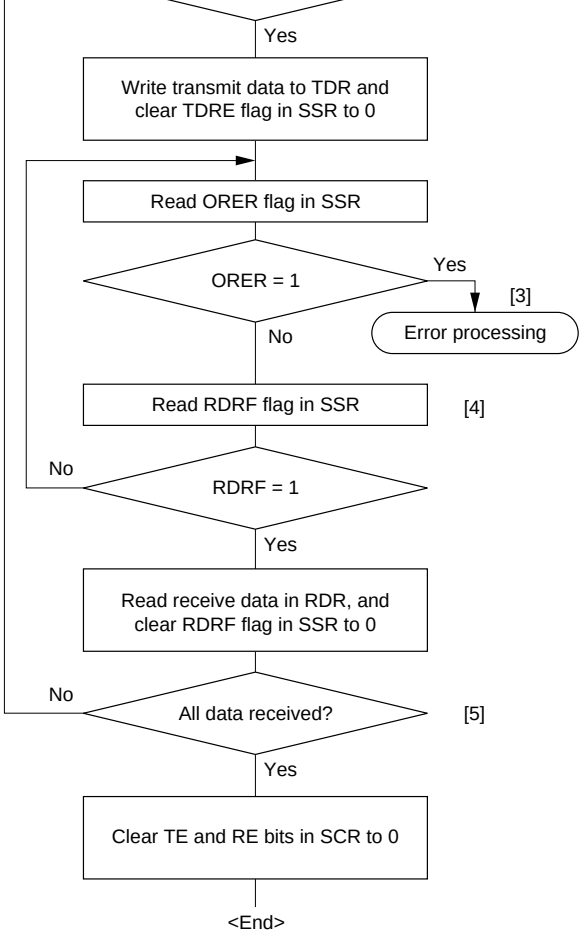


Figure 13.19 Example of SCI Operation in Reception

Simultaneous serial data transmission and reception (clocked synchronous mode): 13.20 shows a sample flowchart for simultaneous serial transmit and receive operations

The following procedure should be used for simultaneous serial data transmit and recei operations.



Note: When switching from transmit or receive operation to simultaneous transmit and receive operations, first clear the TE bit and RE bit to 0, then set both these bits to 1 simultaneously.

transmit data to TDR and
TDRE flag to 0.
Transition of the TDRE fla
to 1 can also be identified
interrupt.

[3] Receive error processing:
If a receive error occurs, r
ORER flag in SSR , and a
performing the appropriate
processing, clear the ORE
0. Transmission/reception
resumed if the ORER flag
1.

[4] SCI status check and rece
read:
Read SSR and check that
RDRF flag is set to 1, then
receive data in RDR and c
RDRF flag to 0. Transition
RDRF flag from 0 to 1 can
identified by an RXI interru

[5] Serial transmission/recept
continuation procedure:
To continue serial transmi
reception, before the MSB
the current frame is receiv
reading the RDRF flag, re
RDR, and clearing the RD
0. Also, before the MSB (b
the current frame is transr
read 1 from the TDRE flag
confirm that writing is poss
Then write data to TDR an
the TDRE flag to 0.
Checking and clearing of t
flag is automatic when the
activated by a transmit da
interrupt (TXI) request and
written to TDR. Also, the F
is cleared automatically w
DTC is activated by a rece
full interrupt (RXI) request
RDR value is read.

Figure 13.20 Sample Flowchart of Simultaneous Serial Transmit and Receive

in SSR is set to 1, a TEI interrupt request is generated. A TXI interrupt can activate the
perform data transfer. The TDRE flag is cleared to 0 automatically when data transfer i
performed by the DTC. The DTC cannot be activated by a TEI interrupt request.

When the RDRF flag in SSR is set to 1, an RXI interrupt request is generated. When th
PER, or FER flag in SSR is set to 1, an ERI interrupt request is generated. An RXI inte
activate the DTC to perform data transfer. The RDRF flag is cleared to 0 automatically
transfer is performed by the DTC. The DTC cannot be activated by an ERI interrupt re

	TEI	Interrupt due to transmission end (TEND)	Not possible
1	ERI	Interrupt due to receive error (ORER, FER, or PER)	Not possible
	RXI	Interrupt due to receive data full state (RDRF)	Possible
	TXI	Interrupt due to transmit data empty state (TDRE)	Possible
	TEI	Interrupt due to transmission end (TEND)	Not possible
2	ERI	Interrupt due to receive error (ORER, FER, or PER)	Not possible
	RXI	Interrupt due to receive data full state (RDRF)	Possible
	TXI	Interrupt due to transmit data empty state (TDRE)	Possible
	TEI	Interrupt due to transmission end (TEND)	Not possible

Note: * This table shows the initial state immediately after a reset. Relative prioritie channels can be changed by means of the interrupt controller.

A TEI interrupt is requested when the TEND flag is set to 1 while the TEIE bit is set t TEND flag is cleared at the same time as the TDRE flag. Consequently, if a TEI inter TXI interrupt are requested simultaneously, the TXI interrupt may have priority for ac with the result that the TDRE and TEND flags are cleared. Note that the TEI interrup accepted in this case.

written to TDR when the TDRE flag is cleared to 0, the data stored in TDR will be lost not yet been transferred to TSR. It is therefore essential to check that the TDRE flag is before writing transmit data to TDR.

Operation when Multiple Receive Errors Occur Simultaneously

If a number of receive errors occur at the same time, the state of the status flags in SSR shown in table 13.13. If there is an overrun error, data is not transferred from RSR to R the receive data is lost.

Table 13.13 State of SSR Status Flags and Transfer of Receive Data

SSR Status Flags				Receive Data Transfer	
RDRF	ORER	FER	PER	RSR to RDR	Receive Error Statu
1	1	0	0	X	Overrun error
0	0	1	0	○	Framing error
0	0	0	1	○	Parity error
1	1	1	0	X	Overrun error + fram
1	1	0	1	X	Overrun error + pari
0	0	1	1	○	Framing error + pari
1	1	1	1	X	Overrun error + fram parity error

Notes: ○: Receive data is transferred from RSR to RDR.
X: Receive data is not transferred from RSR to RDR.

whose direction (input or output) is determined by DR and DDR. This can be used to

Between serial transmission initialization and setting of the TE bit to 1, the mark state
by the value of DR (the pin does not function as the TxD pin until the TE bit is set to
Consequently, DDR and DR for the port corresponding to the TxD pin are first set to

To send a break during serial transmission, first clear DR to 0, then clear the TE bit to

When the TE bit is cleared to 0, the transmitter is initialized regardless of the current t
state, the TxD pin becomes an I/O port, and 0 is output from the TxD pin.

Receive Error Flags and Transmit Operations (Clocked Synchronous Mode Only
Transmission cannot be started when a receive error flag (ORER, PER, or FER) is set
the TDRE flag is cleared to 0. Be sure to clear the receive error flags to 0 before starti
transmission.

Note also that receive error flags cannot be cleared to 0 even if the RE bit is cleared to

Receive Data Sampling Timing and Reception Margin in Asynchronous Mode:
In asynchronous mode, the SCI operates on a basic clock with a frequency of 16 times
rate.

In reception, the SCI samples the falling edge of the start bit using the basic clock, and
internal synchronization. Receive data is latched internally at the rising edge of the 8th
basic clock. This is illustrated in figure 13.21.

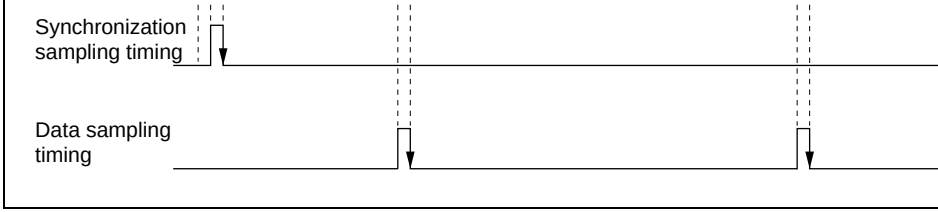


Figure 13.21 Receive Data Sampling Timing in Asynchronous Mode

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

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

Where M: Reception margin (%)
N: Ratio of bit rate to clock (N = 16)
D: Clock duty (D = 0 to 1.0)
L: Frame length (L = 9 to 12)
F: Absolute value of clock rate deviation

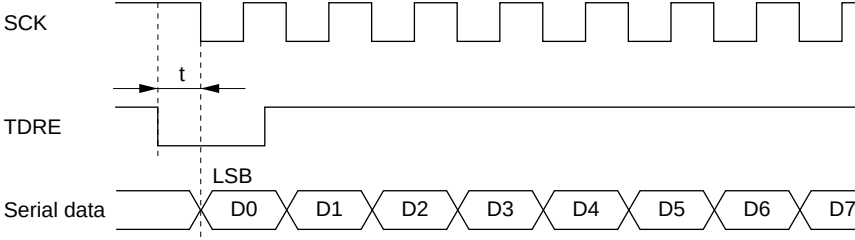
Assuming values of F = 0 and D = 0.5 in formula (1), a reception margin of 46.875% is formula (2) below.

When D = 0.5 and F = 0,

$$M = \left(0.5 - \frac{1}{2 \times 16}\right) \times 100\%$$

$$= 46.875\% \quad \ldots$$

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



Note: When operating on an external clock, set t >4 clocks.

Figure 13.22 Example of Clocked Synchronous Transmission by DT

Operation in Case of Mode Transition

- Transmission

  Operation should be stopped (by clearing TE, TIE, and TEIE to 0) before making
  stop mode, software standby mode, watch mode, subactive mode, or subsleep mod
  TSR, TDR, and SSR are reset. The output pin states in module stop mode, softwa
  mode, watch mode, subactive mode, or subsleep mode depend on the port settings
  becomes high-level output after the relevant mode is cleared. If a transition is ma
  transmission, the data being transmitted will be undefined. When transmitting wit
  changing the transmit mode after the relevant mode is cleared, transmission can be
  setting TE to 1 again, and performing the following sequence: SSR read -> TDR w
  TDRE clearance. To transmit with a different transmit mode after clearing the rel
  the procedure must be started again from initialization. Figure 13.23 shows a samp
  for mode transition during transmission. Port pin states are shown in figures 13.24

  Operation should also be stopped (by clearing TE, TIE, and TEIE to 0) before mak
  transition from transmission by DTC transfer to module stop mode, software stand
  watch mode, subactive mode, or subsleep mode transition. To perform transmissi
  DTC after the relevant mode is cleared, setting TE and TIE to 1 will set the TXI fl
  DTC transmission.

Figure 13.26 shows a sample flowchart for mode transition during reception.

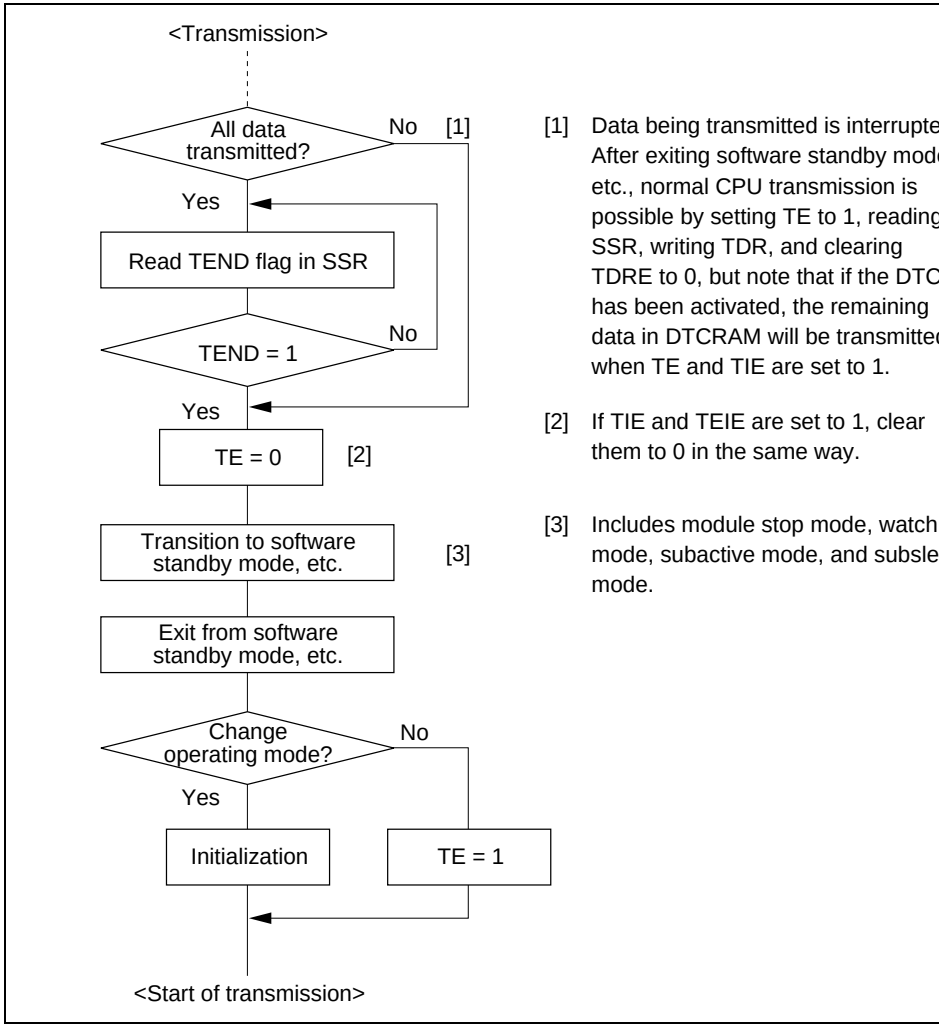


Figure 13.23 Sample Flowchart for Mode Transition during Transmissi

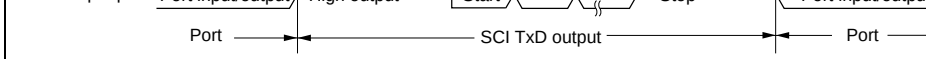
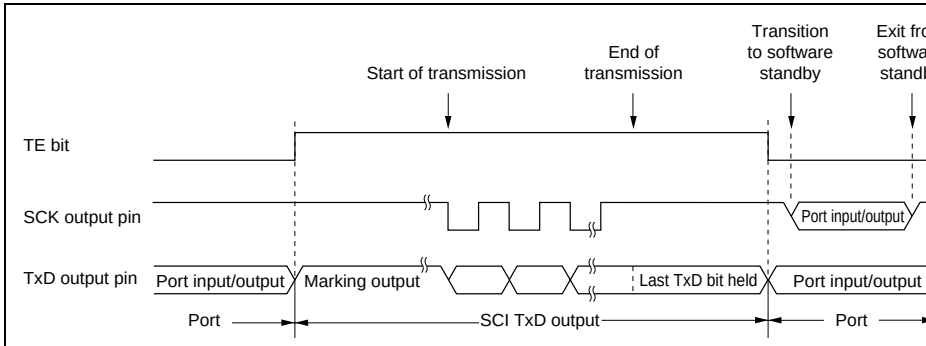
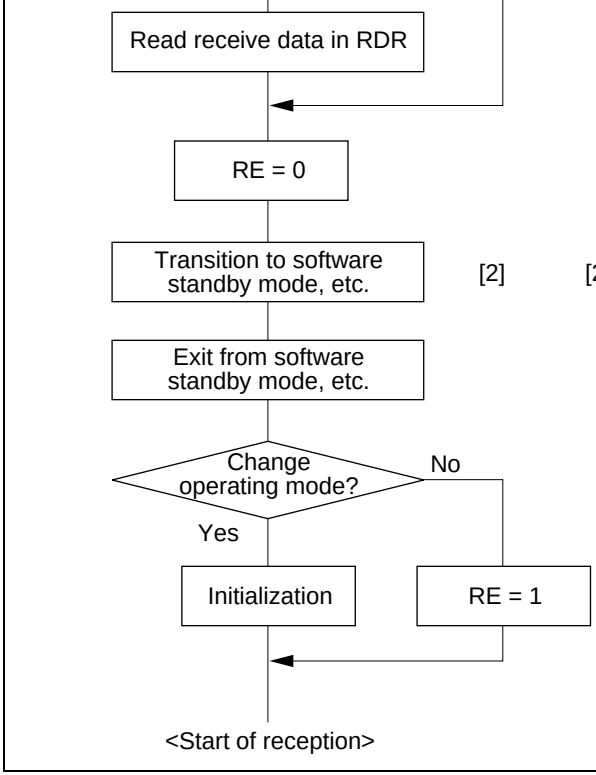


Figure 13.24 Asynchronous Transmission Using Internal Clock



Note: * Initialized by software standby.

Figure 13.25 Synchronous Transmission Using Internal Clock



[2] Includes module stop mode, watch mode, subactive mode and subsleep mode.

Figure 13.26 Sample Flowchart for Mode Transition during Reception

4. Occurrence of low-level output (see figure 13.27)

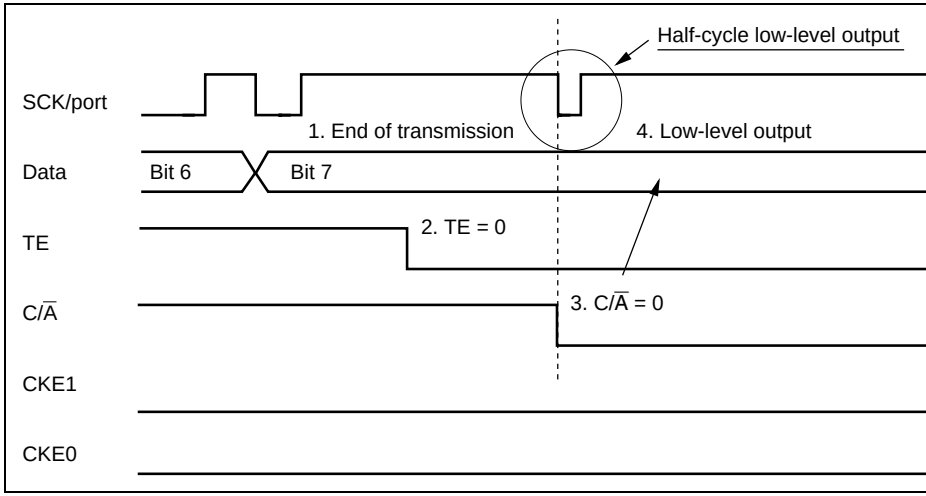


Figure 13.27 Operation when Switching from SCK Pin Function to Port Pin

4. C/$\overline{A}$ bit = 0 ... switchover to port output
5. **CKE1 bit = 0**

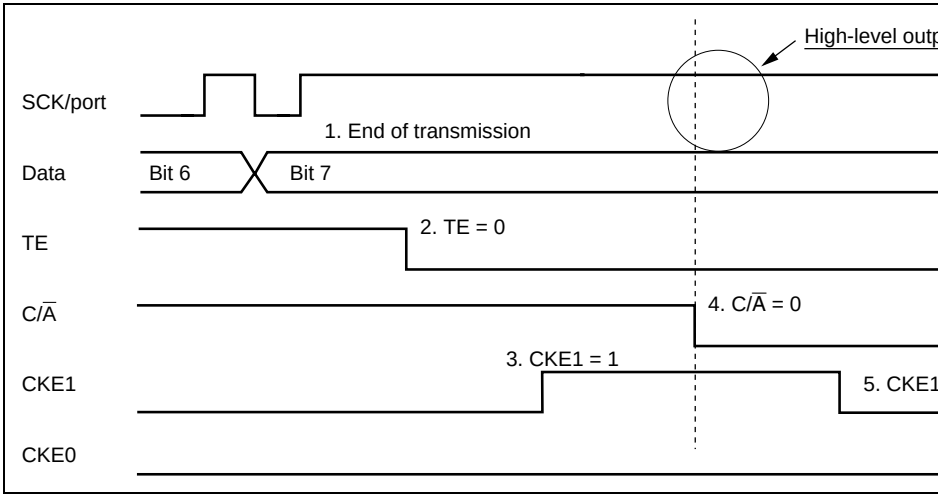


Figure 13.28 Operation when Switching from SCK Pin Function to Port Pin F (Example of Preventing Low-Level Output)

Switching between the normal serial communication interface and the Smart Card inte
carried out by means of a register setting.

14.1.1 Features

Features of the Smart Card interface supported by the H8S/2626 Group and H8S/2623
as follows.

- Asynchronous mode
  - Data length: 8 bits
  - Parity bit generation and checking
  - Transmission of error signal (parity error) in receive mode
  - Error signal detection and automatic data retransmission in transmit mode
  - Direct convention and inverse convention both supported
- On-chip baud rate generator allows any bit rate to be selected
- Three interrupt sources
  - Three interrupt sources (transmit data empty, receive data full, and transmit/rec
    that can issue requests independently
  - The transmit data empty interrupt and receive data full interrupt can activate th
    transfer controller (DTC) to execute data transfer

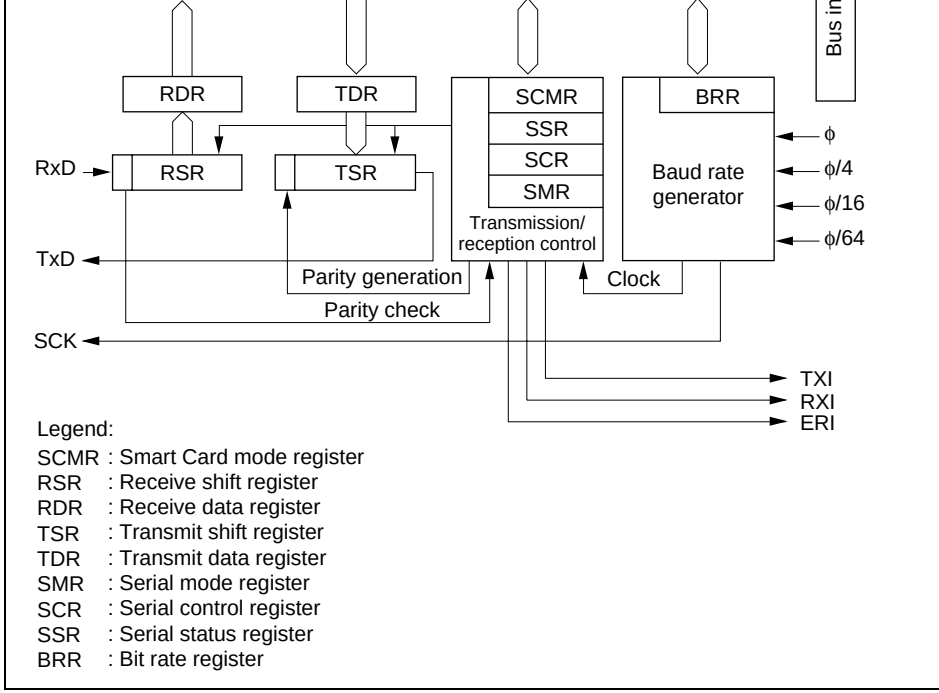


Figure 14.1 Block Diagram of Smart Card Interface

	Receive data pin 0	RxD0	Input	SCI0 receive data in
	Transmit data pin 0	TxD0	Output	SCI0 transmit data o
1	Serial clock pin 1	SCK1	I/O	SCI1 clock input/outp
	Receive data pin 1	RxD1	Input	SCI1 receive data in
	Transmit data pin 1	TxD1	Output	SCI1 transmit data o
2	Serial clock pin 2	SCK2	I/O	SCI2 clock input/outp
	Receive data pin 2	RxD2	Input	SCI2 receive data in
	Transmit data pin 2	TxD2	Output	SCI2 transmit data o

0	Serial mode register 0	SMR0	R/W	H'00	H
	Bit rate register 0	BRR0	R/W	H'FF	H
	Serial control register 0	SCR0	R/W	H'00	H
	Transmit data register 0	TDR0	R/W	H'FF	H
	Serial status register 0	SSR0	R/(W)*2	H'84	H
	Receive data register 0	RDR0	R	H'00	H
	Smart card mode register 0	SCMR0	R/W	H'F2	H
1	Serial mode register 1	SMR1	R/W	H'00	H
	Bit rate register 1	BRR1	R/W	H'FF	H
	Serial control register 1	SCR1	R/W	H'00	H
	Transmit data register 1	TDR1	R/W	H'FF	H
	Serial status register 1	SSR1	R/(W)*2	H'84	H
	Receive data register 1	RDR1	R	H'00	H
	Smart card mode register 1	SCMR1	R/W	H'F2	H
2	Serial mode register 2	SMR2	R/W	H'00	H
	Bit rate register 2	BRR2	R/W	H'FF	H
	Serial control register 2	SCR2	R/W	H'00	H
	Transmit data register 2	TDR2	R/W	H'FF	H
	Serial status register 2	SSR2	R/(W)*2	H'84	H
	Receive data register 2	RDR2	R	H'00	H
	Smart card mode register 2	SCMR2	R/W	H'F2	H
All	Module stop control register B	MSTPCRB	R/W	H'FF	H

Notes: 1. Lower 16 bits of the address.
2. Only 0 can be written, for flag clearing.

	—	—	—	—	SDIR	SINV	—
Initial value :	1	1	1	1	0	0	1
R/W :	—	—	—	—	R/W	R/W	—

SCMR is an 8-bit readable/writable register that selects the Smart Card interface funct

SCMR is initialized to H'F2 by a reset and in standby mode.

Bits 7 to 4—Reserved: These bits are always read as 1 and cannot be modified.

Bit 3—Smart Card Data Transfer Direction (SDIR): Selects the serial/parallel con format.

Bit 3 SDIR	Description
0	TDR contents are transmitted LSB-first
	Receive data is stored in RDR LSB-first
1	TDR contents are transmitted MSB-first
	Receive data is stored in RDR MSB-first

Bit 2—Smart Card Data Invert (SINV): Specifies inversion of the data logic level. function is used together with the SDIR bit for communication with an inverse conver The SINV bit does not affect the logic level of the parity bit. For parity-related setting see section 14.3.4, Register Settings.

Bit 1—Reserved: This bit is always read as 1 and cannot be modified.

Bit 0—Smart Card Interface Mode Select (SMIF): Enables or disables the Smart Ca
function.

Bit 0 SMIF	Description
0	Smart Card interface function is disabled
1	Smart Card interface function is enabled

14.2.2 Serial Status Register (SSR)

Bit	:	7	6	5	4	3	2	1
		TDRE	RDRF	ORER	ERS	PER	TEND	MPB
Initial value	:	1	0	0	0	0	1	0
R/W	:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R

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

Bit 4 of SSR has a different function in Smart Card interface mode. Coupled with this,
conditions for bit 2, TEND, are also different.

Bits 7 to 5—Operate in the same way as for the normal SCI. For details, see section 13
Status Register (SSR).

Bit 4—Error Signal Status (ERS): In Smart Card interface mode, bit 4 indicates the s
error signal sent back from the receiving end in transmission. Framing errors are not de
Smart Card interface mode.

[Setting condition]

When the low level of the error signal is sampled

Note: Clearing the TE bit in SCR to 0 does not affect the ERS flag, which retains its p state.

Bits 3 to 0—Operate in the same way as for the normal SCI. For details, see section 1 Status Register (SSR).

However, the setting conditions for the TEND bit, are as shown below.

Bit 2 TEND	Description
0	Transmission is in progress [Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DTC is activated by a TXI interrupt and write data to TDR
1	Transmission has ended [Setting conditions] • Upon reset, and in standby mode or module stop mode • When the TE bit in SCR is 0 and the ERS bit is also 0 • When TDRE = 1 and ERS = 0 (normal transmission) 2.5 etu after trans 1-byte serial character when GM = 0 and BLK = 0 • When TDRE = 1 and ERS = 0 (normal transmission) 1.5 etu after trans 1-byte serial character when GM = 0 and BLK = 1 • When TDRE = 1 and ERS = 0 (normal transmission) 1.0 etu after trans 1-byte serial character when GM = 1 and BLK = 0 • When TDRE = 1 and ERS = 0 (normal transmission) 1.0 etu after trans 1-byte serial character when GM = 1 and BLK = 1

Note: etu: Elementary Time Unit (time for transfer of 1 bit)

The function of bits 7, 6, 3, and 2 of SMR changes in Smart Card interface mode.

Bit 7—GSM Mode (GM): Sets the smart card interface function to GSM mode.

This bit is cleared to 0 when the normal smart card interface is used. In GSM mode, th
to 1, the timing of setting of the TEND flag that indicates transmission completion is ac
and clock output control mode addition is performed. The contents of the clock output
mode addition are specified by bits 1 and 0 of the serial control register (SCR).

Bit 7 GM	Description
0	Normal smart card interface mode operation (In • TEND flag generation 12.5 etu (11.5 etu in block transfer mode) after be start bit • Clock output ON/OFF control only
1	GSM mode smart card interface mode operation • TEND flag generation 11.0 etu after beginning of start bit • High/low fixing control possible in addition to clock output ON/OFF cont SCR)

Note: etu: Elementary time unit (time for transfer of 1 bit)

1	Block transfer mode operation • Error signal transmission/detection and automatic data retransmission performed • TXI interrupt generated by TDRE flag • TEND flag set 11.5 etu after start of transmission (11.0 etu in GSM mo

Note: etu: Elementary Time Unit (time for transfer of 1 bit)

Bits 3 and 2—Basic Clock Pulse 1 and 2 (BCP1, BCP0): These bits specify the num clock periods in a 1-bit transfer interval on the Smart Card interface.

Bit 3	Bit 2		
BCP1	**BCP0**	**Description**	
0	1	32 clock periods	(
	0	64 clock periods	
1	1	372 clock periods	
	0	256 clock periods	

Bits 5, 4, 1, and 0: Operate in the same way as for the normal SCI. For details, see se Serial Mode Register (SMR).

In smart card interface mode, the function of bits 1 and 0 of SCR changes when bit 7 of mode register (SMR) is set to 1.

Bits 7 to 2—Operate in the same way as for the normal SCI.

For details, see section 13.2.6, Serial Control Register (SCR).

Bits 1 and 0—Clock Enable 1 and 0 (CKE1, CKE0): These bits are used to select the source and enable or disable clock output from the SCK pin.

In smart card interface mode, in addition to the normal switching between clock output and disabling, the clock output can be specified as to be fixed high or low.

SCMR	SMR	SCR Setting		SCK Pin Function
SMIF	C/$\overline{A}$, GM	CKE1	CKE0	
0	See the SCI			
1	0	0	0	Operates as port I/O pin
1	0	0	1	Outputs clock as SCK output pin
1	1	0	0	Operates as SCK output pin, with outp low
1	1	0	1	Outputs clock as SCK output pin
1	1	1	0	Operates as SCK output pin, with outp high
1	1	1	1	Outputs clock as SCK output pin

bit) is left between the end of the parity bit and the start of the next frame.

- If a parity error is detected during reception, a low error signal level is output for c period, 10.5 etu after the start bit.
- If the error signal is sampled during transmission, the same data is transmitted auto after the elapse of 2 etu or longer. (except in block transfer mode)
- Only asynchronous communication is supported; there is no clocked synchronous communication function.

14.3.2 Pin Connections

Figure 14.2 shows a schematic diagram of Smart Card interface related pin connectio

In communication with an IC card, since both transmission and reception are carried o single data transmission line, the TxD pin and RxD pin should be connected with the data transmission line should be pulled up to the V_{CC} power supply with a resistor.

When the clock generated on the Smart Card interface is used by an IC card, the SCK input to the CLK pin of the IC card. No connection is needed if the IC card uses an int

LSI port output is used as the reset signal.

Other pins must normally be connected to the power supply or ground.

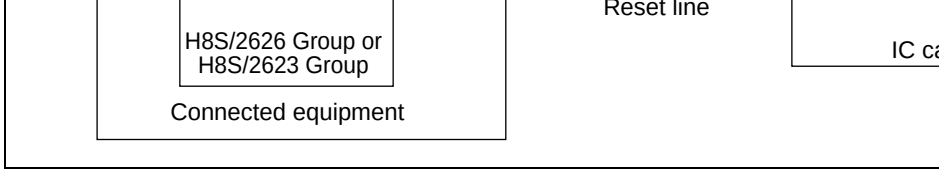


Figure 14.2 Schematic Diagram of Smart Card Interface Pin Connectio

Note: If an IC card is not connected, and the TE and RE bits are both set to 1, closed transmission/reception is possible, enabling self-diagnosis to be carried out.

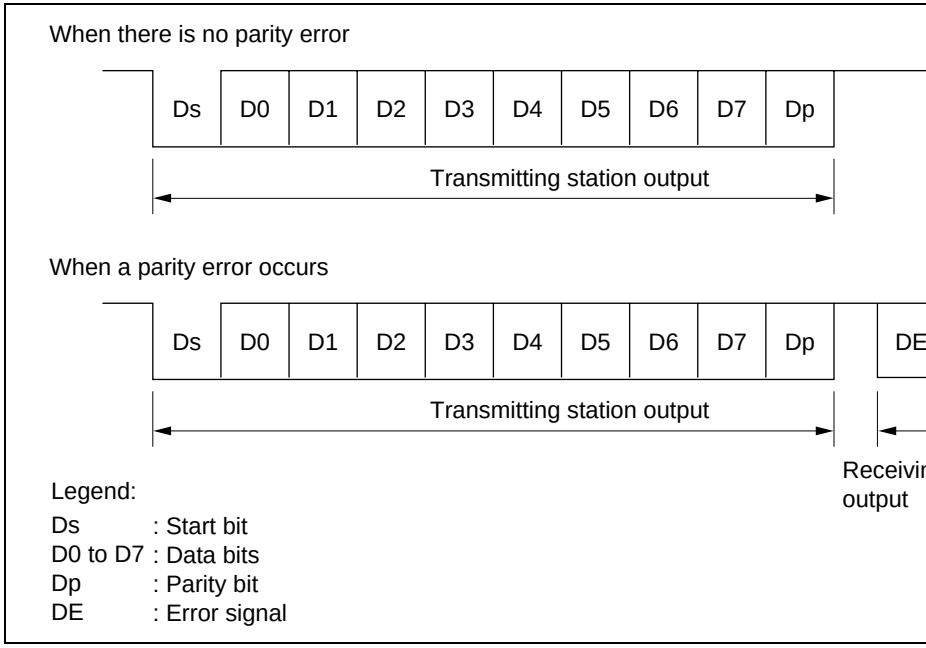


Figure 14.3 Normal Smart Card Interface Data Format

line is pulled high with a pull-up resistor.

[4] The receiving station carries out a parity check.

If there is no parity error and the data is received normally, the receiving station wa reception of the next data.

If a parity error occurs, however, the receiving station outputs an error signal (DE, l to request retransmission of the data. After outputting the error signal for the prescr of time, the receiving station places the signal line in the high-impedance state agai signal line is pulled high again by a pull-up resistor.

[5] If the transmitting station does not receive an error signal, it proceeds to transmit th frame.

If it does receive an error signal, however, it returns to step [2] and retransmits the e data.

(2) Block Transfer Mode

The operation sequence in block transfer mode is as follows.

[1] When the data line in not in use it is in the high-impedance state, and is fixed high v up resistor.

[2] The transmitting station starts transfer of one frame of data. The data frame starts v bit (Ds, low-level), followed by 8 data bits (D0 to D7) and a parity bit (Dp).

[3] With the Smart Card interface, the data line then returns to the high-impedance stat line is pulled high with a pull-up resistor.

[4] After reception, a parity error check is carried out, but an error signal is not output e error has occurred. When an error occurs reception cannot be continued, so the erro should be cleared to 0 before the parity bit of the next frame is received.

[5] The transmitting station proceeds to transmit the next data frame.

	Bit						
Register	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**
SMR	GM	BLK	1	O/$\overline{E}$	BCP1	BCP0	CKS1
BRR	BRR7	BRR6	BRR5	BRR4	BRR3	BRR2	BRR1
SCR	TIE	RIE	TE	RE	0	0	CKE1*
TDR	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1
SSR	TDRE	RDRF	ORER	ERS	PER	TEND	0
RDR	RDR7	RDR6	RDR5	RDR4	RDR3	RDR2	RDR1
SCMR	—	—	—	—	SDIR	SINV	—

Notes: — : Unused bit.

*: The CKE1 bit must be cleared to 0 when the GM bit in SMR is cleared to 0

SMR Setting: The GM bit is cleared to 0 in normal smart card interface mode, and se GSM mode. The O/$\overline{E}$ bit is cleared to 0 if the IC card is of the direct convention type, if of the inverse convention type.

Bits CKS1 and CKS0 select the clock source of the on-chip baud rate generator. Bits I BCP0 select the number of basic clock periods in a 1-bit transfer interval. For details, 14.3.5, Clock.

The BLK bit is cleared to 0 in normal smart card interface mode, and set to 1 in block mode.

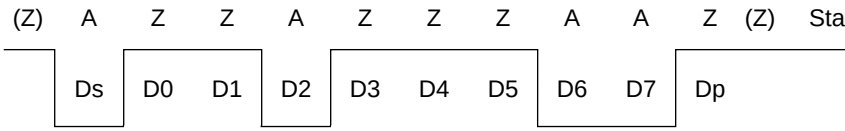
BRR Setting: BRR is used to set the bit rate. See section 14.3.5, Clock, for the metho calculating the value to be set.

SCR Setting: The function of the TIE, RIE, TE, and RE bits is the same as for the no For details, see section 13, Serial Communication Interface (SCI).

Bits CKE1 and CKE0 specify the clock output. When the GM bit in SMR is cleared t bits to B'00 if a clock is not to be output, or to B'01 if a clock is to be output. When th SMR is set to 1, clock output is performed. The clock output can also be fixed high o

types of IC card (direct convention and inverse convention).

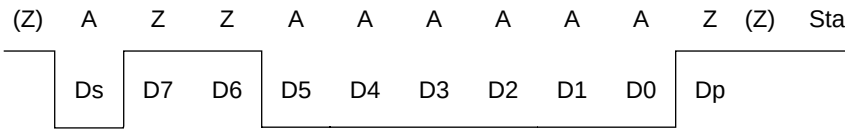
- Direct convention (SDIR = SINV = O/$\overline{E}$ = 0)



With the direct convention type, the logic 1 level corresponds to state Z and the log
state A, and transfer is performed in LSB-first order. The start character data above

The parity bit is 1 since even parity is stipulated for the Smart Card.

- Inverse convention (SDIR = SINV = O/$\overline{E}$ = 1)



With the inverse convention type, the logic 1 level corresponds to state A and the lo
to state Z, and transfer is performed in MSB-first order. The start character data abo

The parity bit is 0, corresponding to state Z, since even parity is stipulated for the S

With the H8S/2626 Group and H8S/2623 Group, inversion specified by the SINV b
only to the data bits, D7 to D0. For parity bit inversion, the O/$\overline{E}$ bit in SMR is set to
mode (the same applies to both transmission and reception).

frequency is determined by the bit rate and the setting of bits BCP1 and BCP0.

$$B = \frac{\phi}{S \times 2^{2n+1} \times (N + 1)} \times 10^6$$

Where: N = Value set in BRR ($0 \leq N \leq 255$)
B = Bit rate (bit/s)
ϕ = Operating frequency (MHz)
n = See table 14.4
S = Number of internal clocks in 1-bit period, set by BCP1 and BCP0

Table 14.4 Correspondence between n and CKS1, CKS0

n	CKS1	CKS0
0	0	0
1		1
2	1	0
3		1

Table 14.5 Examples of Bit Rate B (bit/s) for Various BRR Settings (When n = 0 and S = 372)

	ϕ (MHz)					
N	**10.00**	**10.714**	**13.00**	**14.285**	**16.00**	**18.00**
0	13441	14400	17473	19200	21505	24194
1	6720	7200	8737	9600	10753	12097
2	4480	4800	5824	6400	7168	8065

Note: Bit rates are rounded to the nearest whole number.

bit/s	ϕ (MHz)													
	7.1424		10.00		10.7136		13.00		14.2848		16.00		18.00	
	N	Error	N	Error	N	Error	N	Error	N	Error	N	Error	N	Error
9600	0	0.00	1	30	1	25	1	8.99	1	0.00	1	12.01	2	15.99

Table 14.7 Maximum Bit Rate at Various Frequencies (Smart Card Interface M (when S = 372)

ϕ (MHz)	Maximum Bit Rate (bit/s)	N	n
7.1424	9600	0	0
10.00	13441	0	0
10.7136	14400	0	0
13.00	17473	0	0
14.2848	19200	0	0
16.00	21505	0	0
18.00	24194	0	0
20.00	26882	0	0

The bit rate error is given by the following formula:

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

[3] Set the GM, BLK, O/$\overline{E}$, BCP1, BCP0, CKS1, CKS0 bits in SMR. Set the PE bit to

[4] Set the SMIF, SDIR, and SINV bits in SCMR.
When the SMIF bit is set to 1, the TxD and RxD pins are both switched from ports and are placed in the high-impedance state.

[5] Set the value corresponding to the bit rate in BRR.

[6] Set the CKE0 and CKE1 bits in SCR. Clear the TIE, RIE, TE, RE, MPIE, and TE
If the CKE0 bit is set to 1, the clock is output from the SCK pin.

[7] Wait at least one bit interval, then set the TIE, RIE, TE, and RE bits in SCR. Do n bit and RE bit at the same time, except for self-diagnosis.

[3] Repeat steps [2] and [3] until it can be confirmed that the TEND flag in SSR is set t

[4] Write the transmit data to TDR, clear the TDRE flag to 0, and perform the transmit
The TEND flag is cleared to 0.

[5] When transmitting data continuously, go back to step [2].

[6] To end transmission, clear the TE bit to 0.

With the above processing, interrupt servicing or data transfer by the DTC is possible.

If transmission ends and the TEND flag is set to 1 while the TIE bit is set to 1 and inter
requests are enabled, a transmit data empty interrupt (TXI) request will be generated. If
occurs in transmission and the ERS flag is set to 1 while the RIE bit is set to 1 and inter
requests are enabled, a transfer error interrupt (ERI) request will be generated.

The timing for setting the TEND flag depends on the value of the GM bit in SMR. The
flag set timing is shown in figure 14.6.

If the DTC is activated by a TXI request, the number of bytes set in the DTC can be tra
automatically, including automatic retransmission.

For details, see Interrupt Operation (Except Block Transfer Mode) and Data Transfer C
DTC below.

Note: For block transfer mode, see section 13.3.2, Operation in Asynchronous Mode.

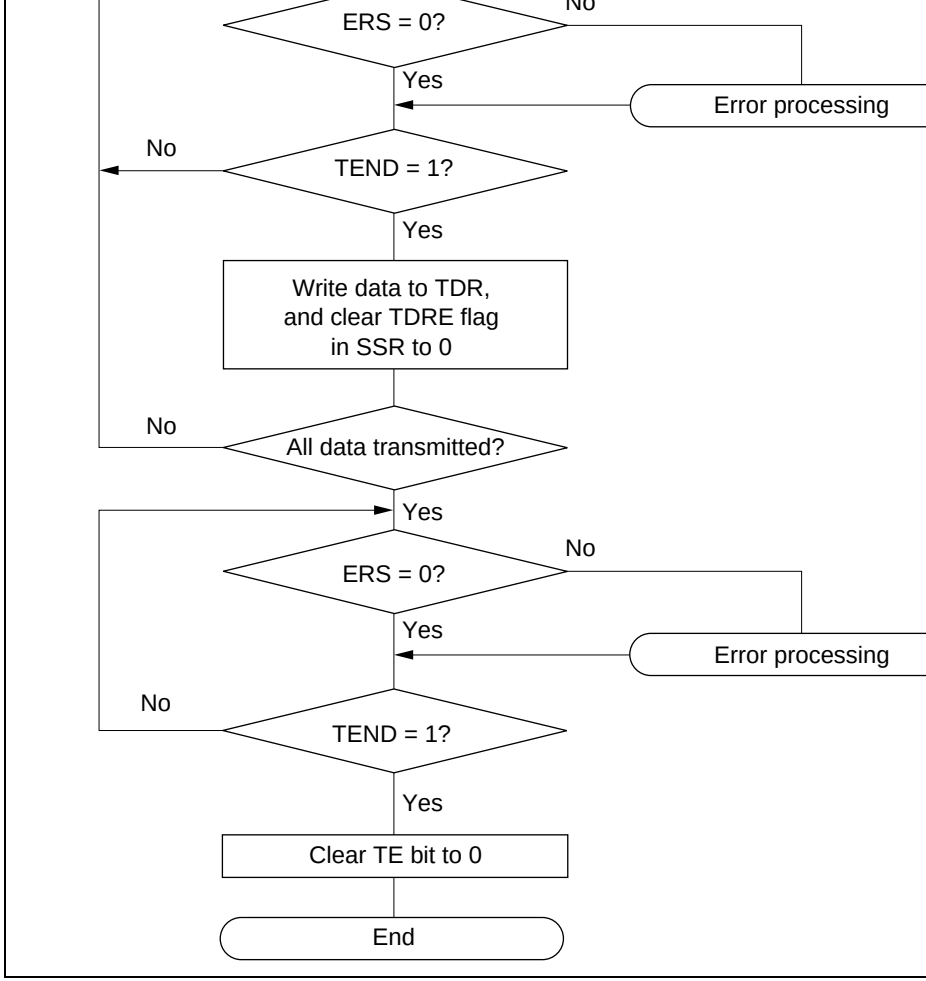
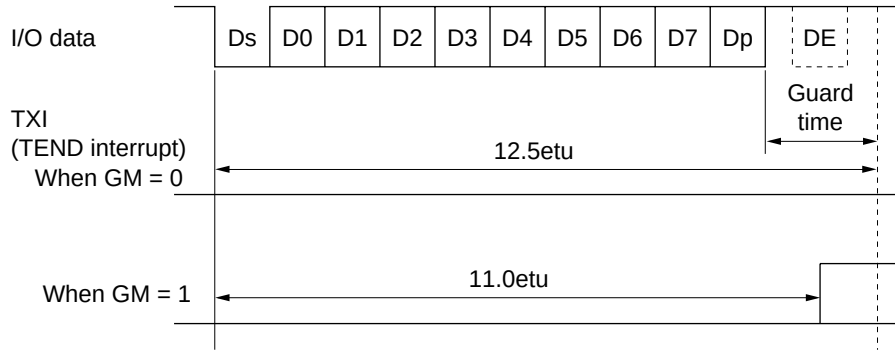


Figure 14.4 Example of Transmission Processing Flow

In case of normal transmission: TEND flag is set
In case of transmit error: ERS flag is set
Steps (2) and (3) above are repeated until the TEND

Note: When the ERS flag is set, it should be cleared until transfer of the last bit (D7 in L transmission, D0 in MSB-first transmission) of the next transfer data to be transmi been completed.

Figure 14.5 Relation between Transmit Operation and Internal Registe



Legend:
Ds : Start bit
D0 to D7 : Data bits
Dp : Parity bit
DE : Error signal
etu: Elementary Time Unit (time for transfer of 1 bit)

Figure 14.6 TEND Flag Generation Timing in Transmission Operation

[3] Repeat steps [2] and [3] until it can be confirmed that the RDRF flag is set to 1.

[4] Read the receive data from RDR.

[5] When receiving data continuously, clear the RDRF flag to 0 and go back to step [2

[6] To end reception, clear the RE bit to 0.

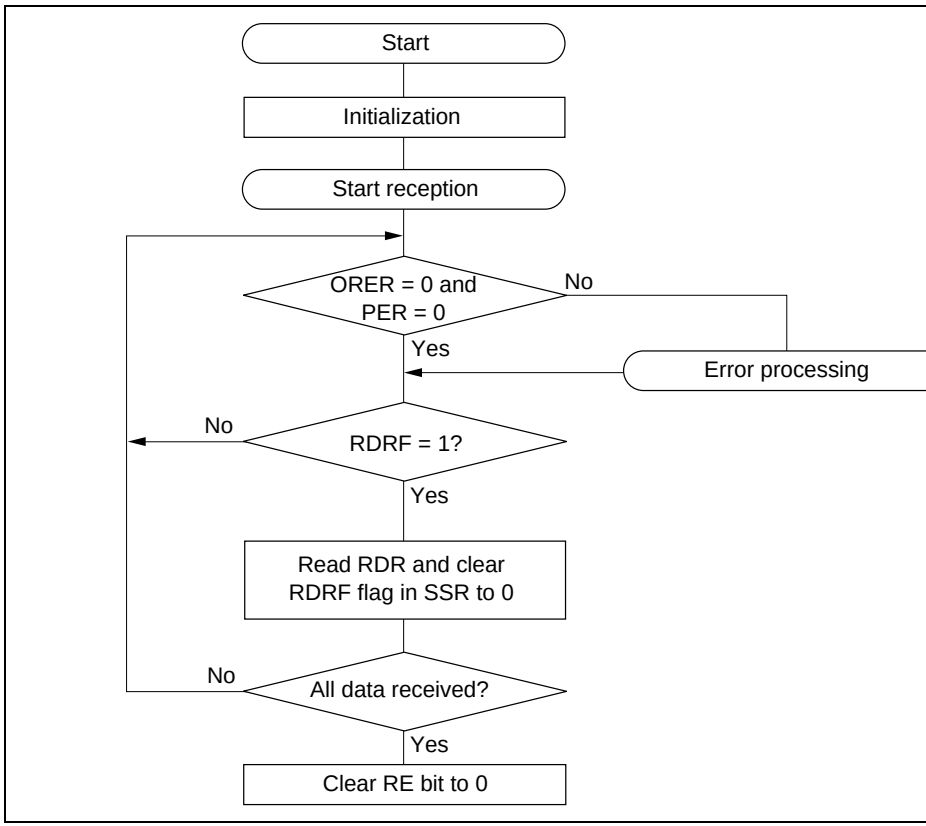


Figure 14.7 Example of Reception Processing Flow

For details, see Interrupt Operation (Except Block Transfer Mode) and Data Transfer C
DTC below.

If a parity error occurs during reception and the PER is set to 1, the received data is stil
transferred to RDR, and therefore this data can be read.

Note: For block transfer mode, see section 13.3.2, Operation in Asynchronous Mode.

Mode Switching Operation: When switching from receive mode to transmit mode, fir
that the receive operation has been completed, then start from initialization, clearing RI
and setting TE bit to 1. The RDRF flag or the PER and ORER flags can be used to chec
receive operation has been completed.

When switching from transmit mode to receive mode, first confirm that the transmit op
been completed, then start from initialization, clearing TE bit to 0 and setting RE bit to
TEND flag can be used to check that the transmit operation has been completed.

Fixing Clock Output Level: When the GM bit in SMR is set to 1, the clock output lev
fixed with bits CKE1 and CKE0 in SCR. At this time, the minimum clock pulse width
made the specified width.

Figure 14.8 shows the timing for fixing the clock output level. In this example, GSM is
CKE1 is cleared to 0, and the CKE0 bit is controlled.

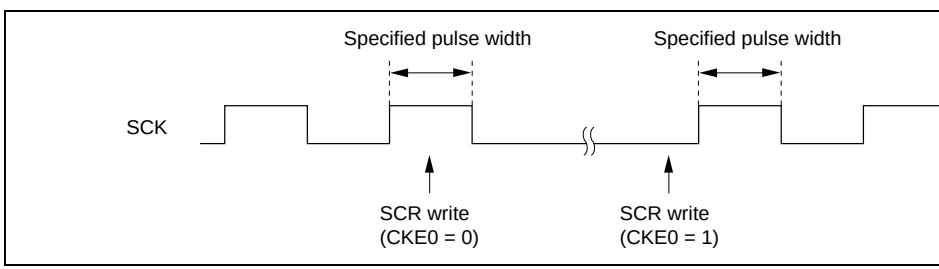


Figure 14.8 Timing for Fixing Clock Output Level

When any of flags ORER, PER, and ERS in SSR is set to 1, an ERI interrupt request i
The relationship between the operating states and interrupt sources is shown in table 1

Note: For block transfer mode, see section 13.4, SCI Interrupts.

Table 14.8 Smart Card Mode Operating States and Interrupt Sources

Operating State		Flag	Enable Bit	Interrupt Source	DTC
Transmit Mode	Normal operation	TEND	TIE	TXI	Possi
	Error	ERS	RIE	ERI	Not p
Receive Mode	Normal operation	RDRF	RIE	RXI	Possi
	Error	PER, ORER	RIE	ERI	Not p

Data Transfer Operation by DTC: In smart card mode, as with the normal SCI, tran
carried out using the DTC. In a transmit operation, the TDRE flag is also set to 1 at th
as the TEND flag in SSR, and a TXI interrupt is generated. If the TXI request is desig
beforehand as a DTC activation source, the DTC will be activated by the TXI request,
of the transmit data will be carried out. The TDRE and TEND flags are automatically
when data transfer is performed by the DTC. In the event of an error, the SCI retransm
data automatically. During this period, TEND remains cleared to 0 and the DTC is no
Therefore, the SCI and DTC will automatically transmit the specified number of bytes
retransmission in the event of an error. However, the ERS flag is not cleared automati
an error occurs, and so the RIE bit should be set to 1 beforehand so that an ERI reques
generated in the event of an error, and the ERS flag will be cleared.

When performing transfer using the DTC, it is essential to set and enable the DTC bef
out SCI setting. For details of the DTC setting procedures, see section 8, Data Transfe
(DTC).

In a receive operation, an RXI interrupt request is generated when the RDRF flag in S
1. If the RXI request is designated beforehand as a DTC activation source, the DTC w

Switching the Mode: When switching between smart card interface mode and softwar
mode, the following switching procedure should be followed in order to maintain the cl

- When changing from smart card interface mode to software standby mode

[1] Set the data register (DR) and data direction register (DDR) corresponding to the SC
the value for the fixed output state in software standby mode.

[2] Write 0 to the TE bit and RE bit in the serial control register (SCR) to halt transmit/
operation. At the same time, set the CKE1 bit to the value for the fixed output state
software standby mode.

[3] Write 0 to the CKE0 bit in SCR to halt the clock.

[4] Wait for one serial clock period.
During this interval, clock output is fixed at the specified level, with the duty preser

[5] Make the transition to the software standby state.

- When returning to smart card interface mode from software standby mode

[6] Exit the software standby state.

[7] Write 1 to the CKE0 bit in SCR and output the clock. Signal generation is started w
normal duty.

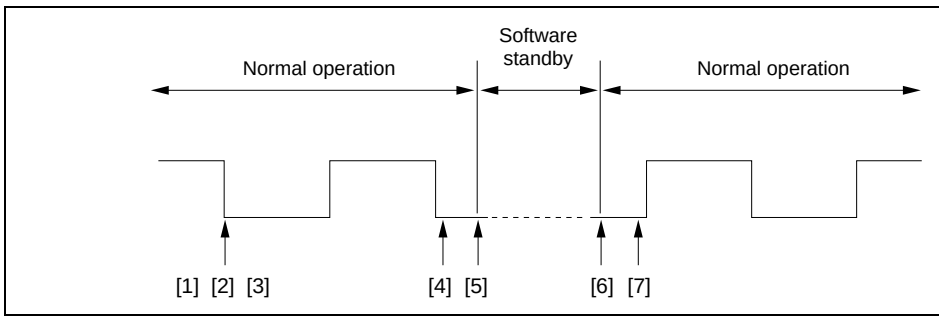


Figure 14.9 Clock Halt and Restart Procedure

[4] Set the CKE0 bit in SCR to 1 to start clock output.

14.3.8 Operation in Block Transfer Mode

Operation in block transfer mode is the same as in SCI asynchronous mode, except fo
following points. For details, see section 13.3.2, Operation in Asynchronous Mode.

(1) Data Format

The data format is 8 bits with parity. There is no stop bit, but there is a 2-bit (1-bit or
reception) error guard time.

Also, except during transmission (with start bit, data bits, and parity bit), the transmiss
to the high-impedance state, so the signal lines must be fixed high with a pull-up resis

(2) Transmit/Receive Clock

Only an internal clock generated by the on-chip baud rate generator can be used as the
transmit/receive clock. The number of basic clock periods in a 1-bit transfer interval c
32, 64, 372, or 256 with bits BCP1 and BCP0. For details, see section 14.3.5, Clock.

(3) ERS (FER) Flag

As with the normal Smart Card interface, the ERS flag indicates the error signal status
error signal transmission and reception is not performed, this flag is always cleared to

In reception, the SCI samples the falling edge of the start bit using the basic clock, and internal synchronization. Receive data is latched internally at the rising edge of the 16t 186th, or 128th pulse of the basic clock. Figure 14.10 shows the receive data sampling when using a clock of 372 times the transfer rate.

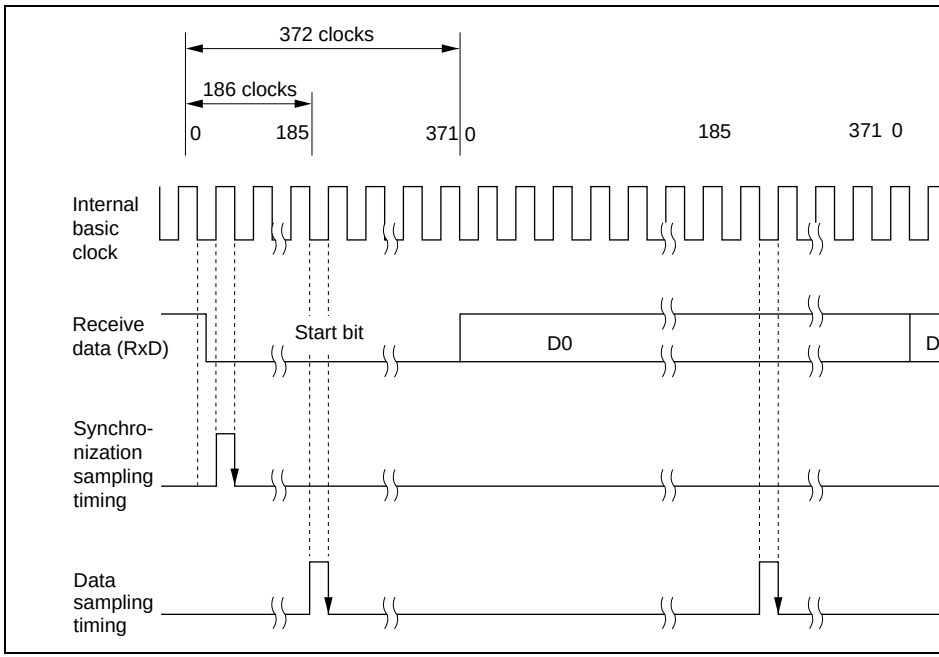


Figure 14.10 Receive Data Sampling Timing in Smart Card Mode (Using Clock of 372 Times the Transfer Rate)

D: Clock duty (D = 0 to 1.0)
L: Frame length (L = 10)
F: Absolute value of clock frequency deviation

Assuming values of F = 0, D = 0.5 and N = 372 in the above formula, the reception m
formula is as follows.

When D = 0.5 and F = 0,

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

[2] The RDRF bit in SSR is not set for a frame in which an error has occurred.

[3] If no error is found when the received parity bit is checked, the PER bit in SSR is n

[4] If no error is found when the received parity bit is checked, the receive operation is have been completed normally, and the RDRF flag in SSR is automatically set to 1. bit in SCR is enabled at this time, an RXI interrupt request is generated.

If DTC data transfer by an RXI source is enabled, the contents of RDR can be read automatically. When the RDR data is read by the DTC, the RDRF flag is automatic to 0.

[5] When a normal frame is received, the pin retains the high-impedance state at the tin error signal transmission.

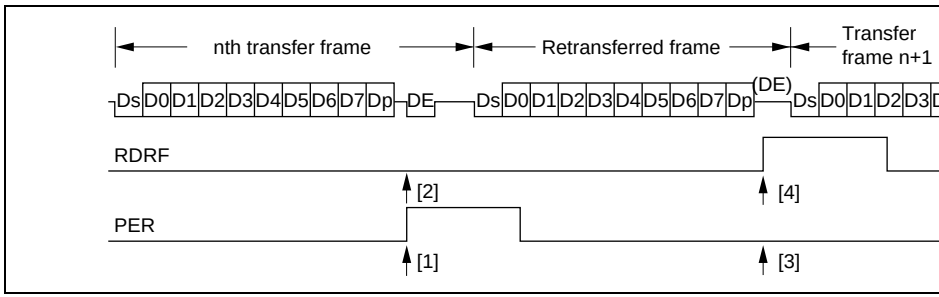


Figure 14.11 Retransfer Operation in SCI Receive Mode

is received.

[8] If an error signal is not sent back from the receiving end, the ERS bit in SSR is not

[9] If an error signal is not sent back from the receiving end, transmission of one fram a retransfer, is judged to have been completed, and the TEND bit in SSR is set to 1 bit in SCR is enabled at this time, a TXI interrupt request is generated.

If data transfer by the DTC by means of the TXI source is enabled, the next data c to TDR automatically. When data is written to TDR by the DTC, the TDRE bit is automatically cleared to 0.

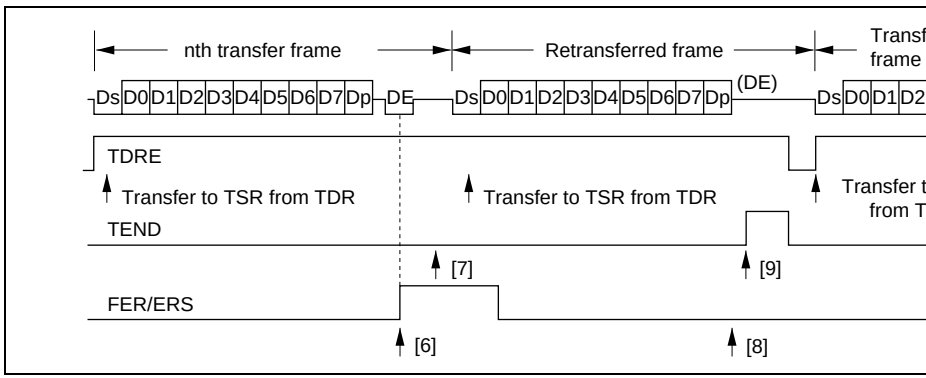


Figure 14.12 Retransfer Operation in SCI Transmit Mode

Reference: BOSCH CAN Specification Version 2.0 1991, Robert Bosch GmbH

15.1.1 Features

- CAN version: Bosch 2.0B active compatible
  - Communication systems:
    NRZ (Non-Return to Zero) system (with bit-stuffing function)
    Broadcast communication system
  - Transmission path: Bidirectional 2-wire serial communication
  - Communication speed: Max. 1 Mbps
  - Data length: 0 to 8 bytes
- Number of channels: 1
- Data buffers: 16 (one receive-only buffer and 15 buffers settable for transmission/
- Data transmission: Choice of two methods:
  - Mailbox (buffer) number order (low-to-high)
  - Message priority (identifier) high-to-low order
- Data reception: Two methods:
  - Message identifier match (transmit/receive-setting buffers)
  - Reception with message identifier masked (receive-only)
- CPU interrupts: Five interrupt vectors:
  - Error interrupt
  - Reset processing interrupt
  - Message reception interrupt (mailboxes 1 to 15)
  - Message reception interrupt (mailbox 0)
  - Message transmission interrupt
- HCAN operating modes: Support for various modes:
  - Hardware reset
  - Software reset

15.1.2 Block Diagram

Figure 15.1 shows a block diagram of the HCAN.

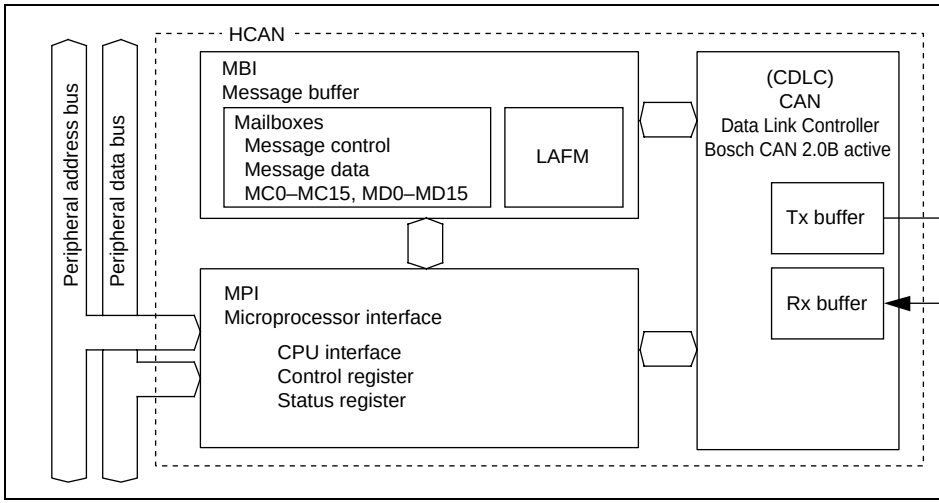


Figure 15.1 HCAN Block Diagram

Message Buffer Interface (MBI): The MBI, consisting of mailboxes and a local accep mask (LAFM), stores CAN transmit/receive messages (identifiers, data, etc.) Transmit are written by the CPU. For receive messages, the data received by the CDLC is stored automatically.

Microprocessor Interface (MPI): The MPI, consisting of a bus interface, control regis register, etc., controls HCAN internal data, statuses, and so forth.

CAN Data Link Controller (CDLC): The CDLC performs transmission and reception messages conforming to the Bosch CAN Ver. 2.0B active standard (data frames, remot error frames, overload frames, inter-frame spacing), as well as CRC checking, bus arbit other functions.

Name	Abbreviation	Input/Output	Function
HCAN transmit data pin	HTxD	Output	CAN bus trans
HCAN receive data pin	HRxD	Input	CAN bus rece

A bus driver is necessary between the pins and the CAN bus. A Philips PCA82C250 c
model is recommended.

Bit configuration register	BCR	R/W	H'0000	H'F802	8/
Mailbox configuration register	MBCR	R/W	H'0100	H'F804	8/
Transmit wait register	TXPR	R/W	H'0000	H'F806	8/
Transmit wait cancel register	TXCR	R/W	H'0000	H'F808	8/
Transmit acknowledge register	TXACK	R/W	H'0000	H'F80A	8/
Abort acknowledge register	ABACK	R/W	H'0000	H'F80C	8/
Receive complete register	RXPR	R/W	H'0000	H'F80E	8/
Remote request register	RFPR	R/W	H'0000	H'F810	8/
Interrupt register	IRR	R/W	H'0100	H'F812	8/
Mailbox interrupt mask register	MBIMR	R/W	H'FFFF	H'F814	8/
Interrupt mask register	IMR	R/W	H'FEFF	H'F816	8/
Receive error counter	REC	R	H'00	H'F818	8
Transmit error counter	TEC	R	H'00	H'F819	8
Unread message status register	UMSR	R/W	H'0000	H'F81A	8/
Local acceptance filter mask L	LAFML	R/W	H'0000	H'F81C	8/
Local acceptance filter mask H	LAFMH	R/W	H'0000	H'F81E	8/

Message control 7 [1:8]	MC7 [1:8]	R/W	Undefined	H'F858	8
Message control 8 [1:8]	MC8 [1:8]	R/W	Undefined	H'F860	8
Message control 9 [1:8]	MC9 [1:8]	R/W	Undefined	H'F868	8
Message control 10 [1:8]	MC10 [1:8]	R/W	Undefined	H'F870	8
Message control 11 [1:8]	MC11 [1:8]	R/W	Undefined	H'F878	8
Message control 12 [1:8]	MC12 [1:8]	R/W	Undefined	H'F880	8
Message control 13 [1:8]	MC13 [1:8]	R/W	Undefined	H'F888	8
Message control 14 [1:8]	MC14 [1:8]	R/W	Undefined	H'F890	8
Message control 15 [1:8]	MC15 [1:8]	R/W	Undefined	H'F898	8
Message data 0 [1:8]	MD0 [1:8]	R/W	Undefined	H'F8B0	8
Message data 1 [1:8]	MD1 [1:8]	R/W	Undefined	H'F8B8	8
Message data 2 [1:8]	MD2 [1:8]	R/W	Undefined	H'F8C0	8
Message data 3 [1:8]	MD3 [1:8]	R/W	Undefined	H'F8C8	8
Message data 4 [1:8]	MD4 [1:8]	R/W	Undefined	H'F8D0	8
Message data 5 [1:8]	MD5 [1:8]	R/W	Undefined	H'F8D8	8
Message data 6 [1:8]	MD6 [1:8]	R/W	Undefined	H'F8E0	8
Message data 7 [1:8]	MD7 [1:8]	R/W	Undefined	H'F8E8	8
Message data 8 [1:8]	MD8 [1:8]	R/W	Undefined	H'F8F0	8
Message data 9 [1:8]	MD9 [1:8]	R/W	Undefined	H'F8F8	8
Message data 10 [1:8]	MD10 [1:8]	R/W	Undefined	H'F900	8
Message data 11 [1:8]	MD11 [1:8]	R/W	Undefined	H'F908	8
Message data 12 [1:8]	MD12 [1:8]	R/W	Undefined	H'F910	8
Message data 13 [1:8]	MD13 [1:8]	R/W	Undefined	H'F918	8
Message data 14 [1:8]	MD14 [1:8]	R/W	Undefined	H'F920	8
Message data 15 [1:8]	MD15 [1:8]	R/W	Undefined	H'F928	8
Module stop control register C	MSTPCRC	R/W	H'FF	H'FDEA	8

Note: * Lower 16 bits of the address.

Bit:	7	6	5	4	3	2	1
	MCR7	—	MCR5	—	—	MCR2	MCR1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R	R/W	R	R	R/W	R/W

Bit 7—HCAN Sleep Mode Release (MCR7): Enables or disables HCAN sleep mode bus operation.

Bit 7

MCR7	Description
0	HCAN sleep mode release by CAN bus operation disabled (In
1	HCAN sleep mode release by CAN bus operation enabled

Bit 6—Reserved: This bit always reads 0. The write value should always be 0.

Bit 5—HCAN Sleep Mode (MCR5): Enables or disables HCAN sleep mode transition

Bit 5

MCR5	Description
0	HCAN sleep mode released (In
1	Transition to HCAN sleep mode enabled

Bits 4 and 3—Reserved: These bits always read 0. The write value should always be 0

Bit 1—Halt Request (MCR1): Controls halting of the HCAN module.

Bit 1	
MCR1	**Description**
0	HCAN normal operating mode
1	HCAN halt mode transition request

Bit 0—Reset Request (MCR0): Controls resetting of the HCAN module.

Bit 0	
MCR0	**Description**
0	Normal operating mode (MCR0 = 0 and GSR3 = 0) [Setting condition] When 0 is written after an HCAN reset
1	HCAN reset mode transition request

In order for GSR3 to change from 1 to 0 after 0 is written to MCR0, time is required b
HCAN is internally reset. There is consequently a delay before GSR3 is cleared to 0 a
is cleared to 0.

Initial value:	0	0	0	0	1	1	0
R/W:	R	R	R	R	R	R	R

Bits 7 to 4—Reserved: These bits always read 0.

Bit 3—Reset Status Bit (GSR3): Indicates whether the HCAN module is in the norma state or the reset state. Writes are invalid.

Bit 3

GSR3	Description
0	Normal operating state [Setting condition] After an HCAN internal reset
1	Configuration mode [Reset condition] MCR0 reset mode and sleep mode (Ir

Bit 2—Message Transmission Status Flag (GSR2): Flag that indicates whether the m currently in the message transmission period. The “message transmission period” is the from the start of message transmission (SOF) until the end of a 3-bit intermission interv EOF (End of Frame). Writes are invalid.

Bit 2

GSR2	Description
0	Message transmission period
1	[Reset Condition] Idle period (Ir

Bit 0—Bus Off Flag (GSR0): Flag that indicates the bus off state. Writes are invalid.

Bit 0

GSR0	Description
0	[Reset condition]
	Recovery from bus off state (
1	When TEC ≥ 256 (bus off state)

15.2.3 Bit Configuration Register (BCR)

The bit configuration register (BCR) is a 16-bit readable/writable register that is used bit timing parameters and the baud rate prescaler.

BCR

Bit:	15	14	13	12	11	10	9
	BCR7	BCR6	BCR5	BCR4	BCR3	BCR2	BCR
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	BCR15	BCR14	BCR13	BCR12	BCR11	BCR10	BCR
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

1	Bit synchronization width = 4 time quanta

Bits 13 to 8—Baud Rate Prescaler (BRP): These bits are used to set the CAN bus ba

Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8		
BCR5	**BCR4**	**BCR3**	**BCR2**	**BCR1**	**BCR0**	**Description**	
0	0	0	0	0	0	2 × system clock	(I
0	0	0	0	0	1	4 × system clock	
0	0	0	0	1	0	6 × system clock	
.	.	.	.	.	.	.	
.	.	.	.	.	.	.	
.	.	.	.	.	.	.	
1	1	1	1	1	1	128 × system clock	

Bit 7—Bit Sample Point (BSP): Sets the point at which data is sampled.

Bit 7		
BCR15	**Description**	
0	Bit sampling at one point (end of time segment 1 (TSEG1))	(I
1	Bit sampling at three points (end of TSEG1 and preceding and following tim quantum)	

		1	TSEG2 = 4 time quanta
1	0	0	TSEG2 = 5 time quanta
		1	TSEG2 = 6 time quanta
	1	0	TSEG2 = 7 time quanta
		1	TSEG2 = 8 time quanta

Bits 3 to 0—Time Segment 1 (TSEG1): These bits are used to set the segment for ab
output buffer, CAN bus, and input buffer delay. A value from 1 to 16 can be set.

Bit 3	Bit 2	Bit 1	Bit 0	
BCR11	**BCR10**	**BCR9**	**BCR8**	**Description**
0	0	0	0	Setting prohibited
0	0	0	1	Setting prohibited
0	0	1	0	Setting prohibited
0	0	1	1	TSEG1 = 4 time quanta
0	1	0	0	TSEG1 = 5 time quanta
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
1	1	1	1	TSEG1 = 16 time quanta

Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	MBCR15	MBCR14	MBCR13	MBCR12	MBCR11	MBCR10	MBCR9
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 to 9 and 7 to 0—Mailbox Setting Register (MBCR7 to MBCR1, MBCR15 t MBCR8): These bits set the polarity of the corresponding mailboxes.

Bit x	
MBCRx	**Description**
0	Corresponding mailbox is set for transmission (In
1	Corresponding mailbox is set for reception

Bit 8—Reserved: This bit always reads 1. The write value should always be 1.

Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	TXPR15	TXPR14	TXPR13	TXPR12	TXPR11	TXPR10	TXPR
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 to 9 and 7 to 0—Transmit Wait Register (TXPR7 to TXPR1, TXPR15 to
These bits set a transmit wait for the corresponding mailboxes.

Bit x	
TXPRx	**Description**
0	Transmit message idle state in corresponding mailbox (
	[Clearing condition]
	Message transmission completion and cancellation completion
1	Transmit message transmit wait in corresponding mailbox (CAN bus arbitr

Bit 8—Reserved: This bit always reads 0. The write value should always be 0.

Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	TXCR15	TXCR14	TXCR13	TXCR12	TXCR11	TXCR10	TXCR9
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 to 9 and 7 to 0—Transmit Wait Cancel Register (TXCR7 to TXCR1, TXC TXCR8): These bits control cancellation of transmit wait messages in the correspondi mailboxes.

Bit x	
TXCRx	**Description**
0	Transmit message cancellation idle state in corresponding mailbox (I
	[Clearing condition]
	Completion of TXPR clearing (when transmit message is canceled normall
1	TXPR cleared for corresponding mailbox (transmit message cancellation)

Bit 8—Reserved: This bit always reads 0. The write value should always be 0.

Initial value:	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W

Bit:	7	6	5	4	3	2	1
	TXACK15	TXACK14	TXACK13	TXACK12	TXACK11	TXACK10	TXAC
Initial value:	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W

Note: * Can only be written with 1 for flag clearing.

Bits 15 to 9 and 7 to 0—Transmit Acknowledge Register (TXACK7 to TXACK1, to TXACK8): These bits indicate that a transmit message in the corresponding HCAN has been transmitted normally.

Bit x	
TXACKx	**Description**
0	[Clearing condition]
	Writing 1 (
1	Completion of message transmission for corresponding mailbox

Bit 8—Reserved: This bit always reads 0. The write value should always be 0.

Initial value:	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Bit:	7	6	5	4	3	2	1
	ABACK15	ABACK14	ABACK13	ABACK12	ABACK11	ABACK10	ABACK
Initial value:	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Can only be written with 1 for flag clearing.

Bits 15 to 9 and 7 to 0—Abort Acknowledge Register (ABACK7 to ABACK1, ABA ABACK8): These bits indicate that a transmit message in the corresponding mailbox h canceled (aborted) normally.

Bit x

ABACKx	Description
0	[Clearing condition]
	Writing 1 (Ir
1	Completion of transmit message cancellation for corresponding mailbox

Bit 8—Reserved: This bit always reads 0. The write value should always be 0.

Bit:	15	14	13	12	11	10	9
	RXPR7	RXPR6	RXPR5	RXPR4	RXPR3	RXPR2	RXPF
Initial value:	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W

Bit:	7	6	5	4	3	2	1
	RXPR15	RXPR14	RXPR13	RXPR12	RXPR11	RXPR10	RXPF
Initial value:	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W

Note: * Can only be written with 1 for flag clearing.

Bits 15 to 0—Receive Complete Register (RXPR7 to RXPR0, RXPR15 to RXPR8 indicate that a receive message has been received normally in the corresponding mailb

Bit x

RXPRx	Description
0	[Clearing condition]
	Writing 1 (
1	Completion of message (data frame or remote frame) reception in corresp mailbox

	RFPR7	RFPR6	RFPR5	RFPR4	RFPR3	RFPR2	RFPR
Initial value:	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Bit:	7	6	5	4	3	2	1
	RFPR15	RFPR14	RFPR13	RFPR12	RFPR11	RFPR10	RFPR9
Initial value:	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Can only be written with 1 for flag clearing.

Bits 15 to 0—Remote Request Register (RFPR7 to PFPR0, RFPR15 to PFDR8): T
indicate that a remote frame has been received normally in the corresponding mailbox.

Bit x	
RFPRx	**Description**
0	[Clearing condition]
	Writing 1 (I
1	Completion of remote frame reception in corresponding mailbox

Initial value:	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R

Bit:	7	6	5	4	3	2	1
	—	—	—	IRR12	—	—	IRR
Initial value:	0	0	0	0	0	0	0
R/W:	—	—	—	R/(W)*	—	—	R

Note: * Can only be written with 1 for flag clearing.

Bit 15—Overload Frame/Bus Off Recovery Interrupt Flag (IRR7): Status flag ind
the HCAN has transmitted an overload frame or recovered from the bus off state.

Bit 15

IRR7	Description
0	[Clearing condition] Writing 1 (
1	Overload frame transmission or recovery from bus off state [Setting conditions] Error active/passive state • When overload frame is transmitted Bus off state • When 11 recessive bits is received 128 times (REC ≥ 128)

	[Setting condition]
	When TEC ≥ 256

Bit 13—Error Passive Interrupt Flag (IRR5): Status flag indicating the error passive caused by the transmit/receive error counter.

Bit 13

IRR5	Description
0	[Clearing condition]
	Writing 1 (In
1	Error passive state caused by transmit/receive error
	[Setting condition]
	When TEC ≥ 128 or REC ≥ 128

Bit 12—Receive Overload Warning Interrupt Flag (IRR4): Status flag indicating th warning state caused by the receive error counter.

Bit 12

IRR4	Description
0	[Clearing condition]
	Writing 1 (In
1	Error warning state caused by receive error
	[Setting condition]
	When REC ≥ 96

	[Setting condition]
	When TEC ≥ 96

Bit 10—Remote Frame Request Interrupt Flag (IRR2): Status flag indicating that frame has been received in a mailbox (buffer).

Bit 10

IRR2	Description
0	[Clearing condition]
	Clearing of all bits in RFPR (remote request register) of mailbox for which interrupt requests are enabled by MBIMR (
1	Remote frame received and stored in mailbox
	[Setting conditions]
	When remote frame reception is completed, when corresponding MBIMR = 0

Bit 9—Receive Message Interrupt Flag (IRR1): Status flag indicating that a mailbo receive message has been received normally.

Bit 9

IRR1	Description
0	[Clearing condition]
	Clearing of all bits in RXPR (receive complete register) of mailbox for whic interrupt requests are enabled by MBIMR (
1	Data frame or remote frame received and stored in mailbox
	[Setting conditions]
	When data frame or remote frame reception is completed, when respon MBIMR = 0

	Writing 1	
1	Hardware reset (HCAN module stop*, software standby)	(Ir
	[Setting condition]	
	When reset processing is completed after a hardware reset (HCAN module software standby)	

Note: * After reset or hardware standby release, the module stop bit is initialized to 1 the HCAN enters the module stop state.

Bits 7 to 5, 3, and 2—Reserved: These bits always read 0. The write value should alw

Bit 4—Bus Operation Interrupt Flag (IRR12): Status flag indicating detection of a d due to bus operation when the HCAN module is in HCAN sleep mode.

Bit 4

IRR12	Description	
0	CAN bus idle state	(Ir
	[Clearing condition]	
	Writing 1	
1	CAN bus operation in HCAN sleep mode	
	[Setting condition]	
	Bus operation (dominant bit detection) in HCAN sleep mode	

Bit 1—Unread Interrupt Flag (IRR9): Status flag indicating that a receive message h overwritten while still unread.

Bit 0—Mailbox Empty Interrupt Flag (IRR8): Status flag indicating that the next tr
message can be stored in the mailbox.

Bit 0

IRR8	Description
0	[Clearing condition]
	Writing 1 (
1	Transmit message has been transmitted or aborted, and new message ca
	[Setting condition]
	When TXPR (transmit wait register) is cleared by completion of transmissi completion of transmission abort

15.2.12 Mailbox Interrupt Mask Register (MBIMR)

The mailbox interrupt mask register (MBIMR) is a 16-bit readable/writable register co
flags that enable or disable individual mailbox (buffer) interrupt requests.

MBIMR

Bit:	15	14	13	12	11	10	9
	MBIMR7	MBIMR6	MBIMR5	MBIMR4	MBIMR3	MBIMR2	MBIM
Initial value:	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	MBIMR15	MBIMR14	MBIMR13	MBIMR12	MBIMR11	MBIMR10	MBIM
Initial value:	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	Interrupt request to CPU due to RXPR setting
1	Interrupt requests to CPU disabled (In

15.2.13 Interrupt Mask Register (IMR)

The interrupt mask register (IMR) is a 16-bit readable/writable register containing flags enable or disable requests by individual interrupt sources.

IMR

Bit:	15	14	13	12	11	10	9
	IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1
Initial value:	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	—	—	—	IMR12	—	—	IMR9
Initial value:	1	1	1	1	1	1	1
R/W:	R	R	R	R/W	R	R	R/W

Bit 15—Overload Frame/Bus Off Recovery Interrupt Mask (IMR7): Enables or dis overload frame/bus off recovery interrupt requests.

Bit 15

IMR7	Description
0	Overload frame/bus off recovery interrupt request (OVR0) to CPU by IRR7
1	Overload frame/bus off recovery interrupt request (OVR0) to CPU by IRR7 (In

Bit 13—Error Passive Interrupt Mask (IMR5): Enables or disables error passive in requests caused by the transmit/receive error counter.

Bit 13

IMR5	Description
0	Error passive interrupt request (ERS0) to CPU by IRR5 enabled
1	Error passive interrupt request (ERS0) to CPU by IRR5 disabled (

Bit 12—Receive Overload Warning Interrupt Mask (IMR4): Enables or disables e interrupt requests caused by the receive error counter.

Bit 12

IMR4	Description
0	REC error warning interrupt request (OVR0) to CPU by IRR4 enabled
1	REC error warning interrupt request (OVR0) to CPU by IRR4 disabled (

Bit 11—Transmit Overload Warning Interrupt Mask (IMR3): Enables or disables warning interrupt requests caused by the transmit error counter.

Bit 11

IMR3	Description
0	TEC error warning interrupt request (OVR0) to by IRR3 CPU enabled
1	TEC error warning interrupt request (OVR0) to by IRR3 CPU disabled (

Bit 9—Receive Message Interrupt Mask (IMR1): Enables or disables message recep
interrupt requests.

Bit 9

IMR1	Description
0	Message reception interrupt request (RM1) to CPU by IRR1 enabled
1	Message reception interrupt request (RM1) to CPU by IRR1 disabled (In

Bit 8—Reserved: This bit always reads 0. The write value should always be 0.

Bits 7 to 5, 3, and 2—Reserved: These bits always read 1. The write value should alw

Bit 4—Bus Operation Interrupt Mask (IMR12): Enables or disables interrupt reques
bus operation in sleep mode.

Bit 4

IMR12	Description
0	Bus operation interrupt request (OVR0) to CPU by IRR12 enabled
1	Bus operation interrupt request (OVR0) to CPU by IRR12 disabled (In

Bit 1—Unread Interrupt Mask (IMR9): Enables or disables unread receive message
interrupt requests.

Bit 1

IMR9	Description
0	Unread message overwrite interrupt request (OVR0) to CPU by IRR9 enab
1	Unread message overwrite interrupt request (OVR0) to CPU by IRR9 disab (In

15.2.14 Receive Error Counter (REC)

The receive error counter (REC) is an 8-bit read-only register that functions as a count the number of receive message errors on the CAN bus. The count value is stipulated in protocol.

REC

Bit:	7	6	5	4	3	2	1
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

15.2.15 Transmit Error Counter (TEC)

The transmit error counter (TEC) is an 8-bit read-only register that functions as a coun indicating the number of transmit message errors on the CAN bus. The count value is the CAN protocol.

TEC

Bit:	7	6	5	4	3	2	1
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9
	UMSR7	UMSR6	UMSR5	UMSR4	UMSR3	UMSR2	UMSR
Initial value:	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Bit:	7	6	5	4	3	2	1
	UMSR15	UMSR14	UMSR13	UMSR12	UMSR11	UMSR10	UMSR
Initial value:	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Can only be written with 1 for flag clearing.

Bits 15 to 0—Unread Message Status Flags (UMSR7 to UMSR0, UMSR15 to UMS
Status flags indicating that an unread receive message has been overwritten.

Bit x	
UMSRx	**Description**
0	[Clearing condition] Writing 1 (In
1	Unread receive message is overwritten by a new message [Setting condition] When a new message is received before RXPR is cleared

Bit:	15	14	13	12	11	10	9
	LAFML7	LAFML6	LAFML5	LAFML4	LAFML3	LAFML2	LAFM
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	LAFML15	LAFML14	LAFML13	LAFML12	LAFML11	LAFML10	LAFM
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

LAFMH

Bit:	15	14	13	12	11	10	9
	LAFMH7	LAFMH6	LAFMH5	—	—	—	LAFM
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R	R	R	R/W

Bit:	7	6	5	4	3	2	1
	LAFMH15	LAFMH14	LAFMH13	LAFMH12	LAFMH11	LAFMH10	LAFM
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

1	Stored in MC0, MD0 (receive-only mailbox) regardless of bit match betwee message identifier and receive message identifier

LAFMH Bits 12 to 10—Reserved: These bits always read 0. The write value should a

**LAFMH Bits 9 and 8, LAFML Bits 15 to 0—18-Bit Identifier Filter (LAFMH1, LA
LAFML7 to LAFML0, LAFML15 to LAFML8):** Filter mask bits for the 18 bits of th
message identifier (extended).

Bit x LAFMHx LAFMLx	Description
0	Stored in MC0 (receive-only mailbox) depending on bit match between MC identifier and receive message identifier (In
1	Stored in MC0 (receive-only mailbox) regardless of bit match between MC0 identifier and receive message identifier

	—	—	—	—	DLC3	DLC2	DLC
Initial value:	*	*	*	*	*	*	*
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MCx [2]

Bit:	7	6	5	4	3	2	1
	—	—	—	—	—	—	—
Initial value:	*	*	*	*	*	*	*
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MCx [3]

Bit:	7	6	5	4	3	2	1
	—	—	—	—	—	—	—
Initial value:	*	*	*	*	*	*	*
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MCx [4]

Bit:	7	6	5	4	3	2	1
	—	—	—	—	—	—	—
Initial value:	*	*	*	*	*	*	*
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID
Initial value:	*	*	*	*	*	*	*
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MCx [7]

Bit:	7	6	5	4	3	2	1
	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID
Initial value:	*	*	*	*	*	*	*
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MCx [8]

Bit:	7	6	5	4	3	2	1
	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID
Initial value:	*	*	*	*	*	*	*
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

*

MCx[1] Bits 7 to 4—Reserved: The initial value of these bits is undefined; they must initialized (by writing 0 or 1).

			1	Data length = 3 bytes
	1	0	0	Data length = 4 bytes
			1	Data length = 5 bytes
		1	0	Data length = 6 bytes
			1	Data length = 7 bytes
1	0/1	0/1	0/1	Data length = 8 bytes

MCx[2] Bits 7 to 0—Reserved: The initial value of these bits is undefined; they mus initialized (by writing 0 or 1).

MCx[3] Bits 7 to 0—Reserved: The initial value of these bits is undefined; they mus initialized (by writing 0 or 1).

MCx[4] Bits 7 to 0—Reserved: The initial value of these bits is undefined; they mus initialized (by writing 0 or 1).

MCx[6] Bits 7 to 0—Standard Identifier (STD_ID10 to STD_ID3):
MCx[5] Bits 7 to 5—Standard Identifier (STD_ID2 to STD_ID0):
These bits set the identifier (standard identifier) of data frames and remote frames.

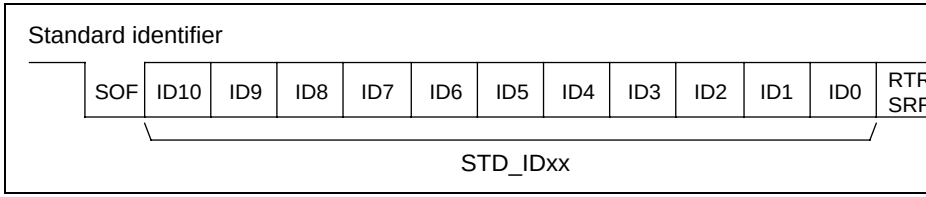


Figure 15.2 Standard Identifier

MCx[5] Bit 3—Identifier Extension (IDE): Used to distinguish between the standard extended format of data frames and remote frames.

Bit 3

IDE	Description
0	Standard format
1	Extended format

MCx[5] Bit 2—Reserved: The initial value of this bit is undefined; it must be initialize writing 0 or 1).

MCx[5] Bits 1 and 0—Extended Identifier (EXD_ID17, EXD_ID16):
MCx[8] Bits 7 to 0—Extended Identifier (EXD_ID15 to EXD_ID8):
MCx[7] Bits 7 to 0—Extended Identifier (EXD_ID7 to EXD_ID0):
These bits set the identifier (extended identifier) of data frames and remote frames.

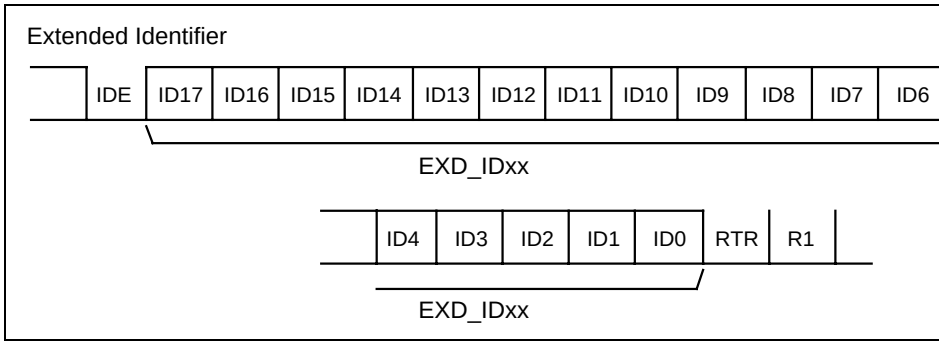


Figure 15.3 Extended Identifier

Initial value:	*	*	*	*	*	*	*
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx [2]

Bit:	7	6	5	4	3	2	1
Initial value:	*	*	*	*	*	*	*
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx [3]

Bit:	7	6	5	4	3	2	1
Initial value:	*	*	*	*	*	*	*
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx [4]

Bit:	7	6	5	4	3	2	1
Initial value:	*	*	*	*	*	*	*
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
Initial value:	*	*	*	*	*	*	*
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx [7]

Bit:	7	6	5	4	3	2	1
Initial value:	*	*	*	*	*	*	*
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx [8]

Bit:	7	6	5	4	3	2	1
Initial value:	*	*	*	*	*	*	*
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

*

MSTPCRC is an 8-bit readable/writable register that performs module stop mode cont

When the MSTPC3 bit is set to 1, HCAN operation is stopped at the end of the bus cy
module stop mode is entered. Register read/write accesses are not possible in module
For details, see sections 21A.5, 21B.5, Module Stop Mode.

MSTPCRC is initialized to H'FF by a reset, and in hardware standby mode. It is not i
software standby mode.

Bit 3—Module Stop (MSTPC3): Specifies the HCAN module stop mode.

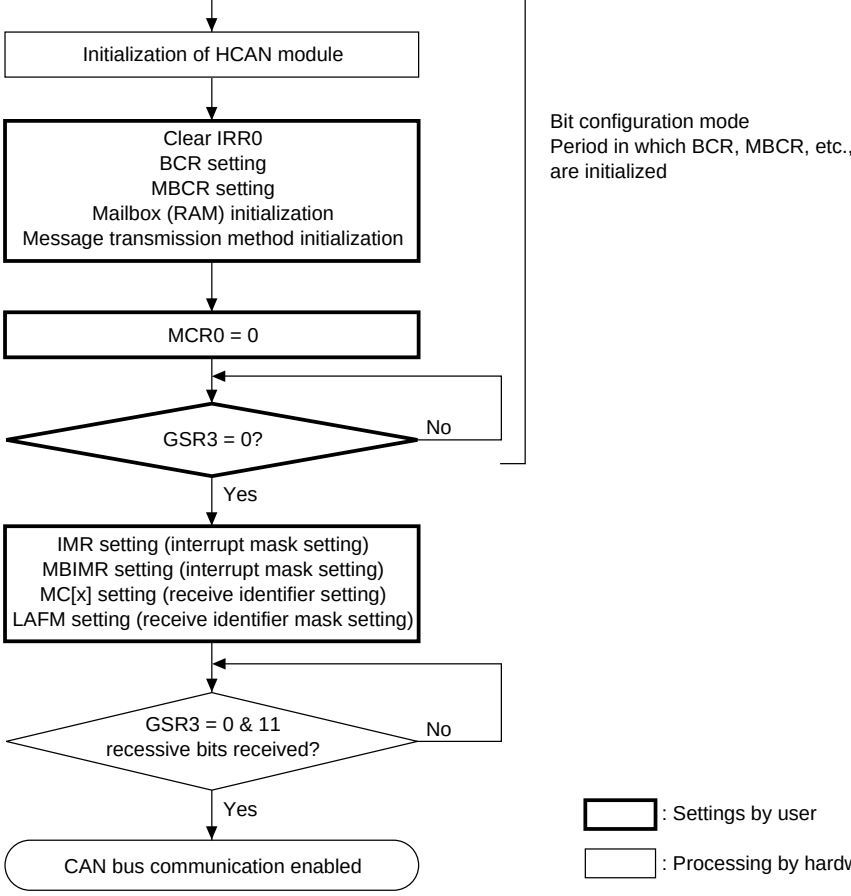
Bit 3

MSTPC3	Description
0	HCAN module stop mode is cleared
1	HCAN module stop mode is set (

bit (GSR3) in GSR within the HCAN (hardware reset). At the same time, all internal r
initialized. However mailbox contents are retained. A flowchart of this reset is shown
15.4.

Note: * In a reset and in hardware standby mode, the module stop bit is initialized to
HCAN enters the module stop state.

Software Reset (Write to MCR0): In normal operation initialization is performed by s
MCR reset request bit (MCR0) in MCR (Software reset). With this kind of reset, if the
controller is performing a communication operation (transmission or reception), the ini
state is not entered until the message has been completed. During initialization, the rese
(GSR3) in GSR is set. In this kind of initialization, the error counters (TEC and REC) a
initialized but other registers and RAM (mailboxes) are not. A flowchart of this reset is
figure 15.5.



Notes: 1. When IRR0 is set to 1 (automatically) due to a hardware reset*2, a "hardwa
initiated reset processing" interrupt is generated.
2. In a reset and in hardware standby mode, the module stop bit is initialized t
the HCAN enters the module stop state.

Figure 15.4 Hardware Reset Flowchart

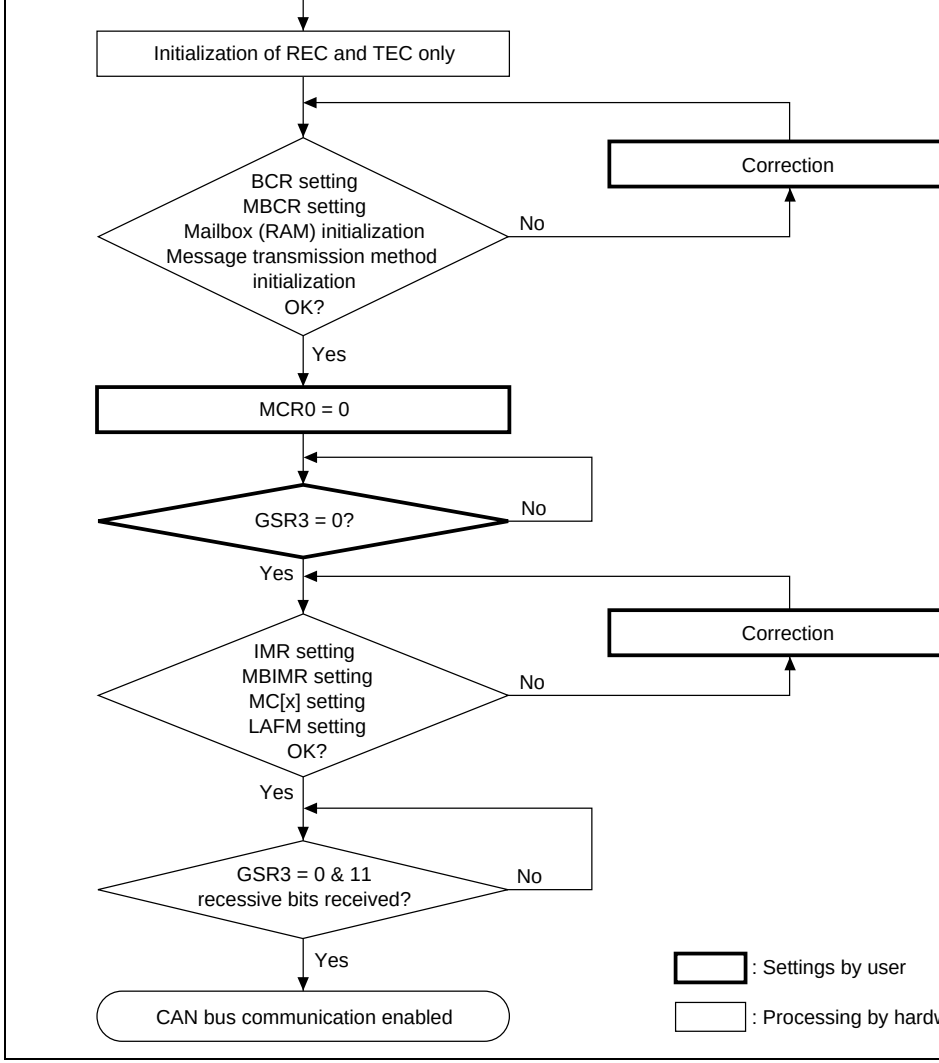


Figure 15.5 Software Reset Flowchart

- Message transmission method setting

These initial settings must be made while the HCAN is in bit configuration mode. Co
mode is a state in which the reset request bit (MCR0) in the master control register (M
the reset status bit in the general status register (GSR) is also 1 (GSR3 = 1). Configura
exited by clearing the reset request bit in MCR to 0; when MCR0 is cleared to 0, the H
automatically clears the reset state bit (GSR3) in the general status register (GSR). Th
sequence then begins, and communication with the CAN bus is possible as soon as the
ends. The power-up sequence consists of the detection of 11 consecutive recessive bit

IRR0 Clearing: The reset interrupt flag (IRR0) is always set after a reset or recovery
software standby mode. As an HCAN interrupt is initiated immediately when interrup
enabled, IRR0 should be cleared.

Bit Rate and Bit Timing Settings: As bit rate settings, a baud rate setting and bit tim
must be made each time a CAN node begins communication. The baud rate and bit tir
are made in the bit configuration register (BCR).

a. Note

   BCR can be written to at all times, but should only be modified in configuration m

   Settings should be made so that all CAN controllers connected to the CAN bus hav
   baud rate and bit width.

   Refer to table 15.3 for the range of values that can be used as settings (TSEG1, TS
   sample point, and SJW) for BCR.

b. Value Setting Ranges

- The bit width consists of the total of the settable Time Quanta (TQ). TQ (number of clocks) is determined by the baud rate prescaler (BRP).

$$TQ = \frac{2 \times (BRP + 1)}{f_{CLK}}$$

- The value of SJW is stipulated in the CAN specifications.

$$3 \geq SJW \geq 0$$

- The minimum value of TSEG1 is stipulated in the CAN specifications.

$$TSEG1 > TSEG2$$

- The minimum value of TSEG2 is stipulated in the CAN specifications.

$$TSEG2 \geq SJW$$

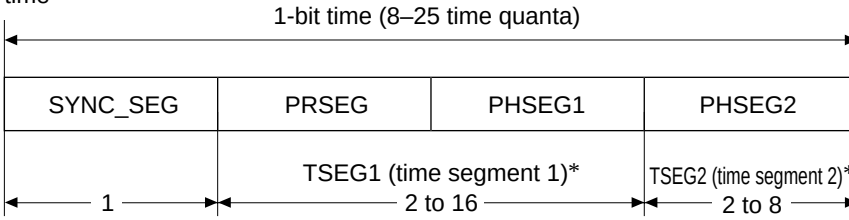
The following formula is used to calculate the baud rate.

$$\text{Bit rate [b/s]} = \frac{f_{CLK}}{2 \times (BRP + 1) \times (3 + TSEG1 + TSEG2)}$$

Note: $f_{CLK} = \phi$ (system clock)
The BCR values are used for BRP, TSEG1, and TSEG2.

Example: With a 1 Mb/s baud rate and a 20 MHz input clock:

$$1 \text{ Mb/s} = \frac{20 \text{ MHz}}{2 \times (0 + 1) \times (3 + 4 + 3)}$$



Legend:

SYNC_SEG: Segment for establishing synchronization of nodes on the CAN bus. (N bit edge transitions occur in this segment.)
PRSEG: Segment for compensating for physical delay between networks.
PHSEG1: Buffer segment for correcting phase drift (positive). (This segment is ex when synchronization (resynchronization) is established.)
PHSEG2: Buffer segment for correcting phase drift (negative). (This segment is shortened when synchronization (resynchronization) is established.)

Note: * The Time Quanta value for TSEG1 and TSEG2 is the TSEG value + 1.

Figure 15.6 Detailed Description of One Bit

HCAN bit rate calculation:

$$\text{Bit rate} = \frac{f_{CLK}}{2 \times (BRP + 1) \times (3 + TSEG1 + TSEG2)}$$

Note: $f_{CLK} = \phi$ (system clock)
The BCR values are used for BRP, TSEG1, and TSEG2.

BCR Setting Constraints

TSEG1 > TSEG2 ≥ SJW (SJW = 0 to 3)

TSEG2 > B'001 (BRP = B'000000)

TSEG2 > B'000 (BRP > B'000000)

	0101	Yes	Yes	Yes	Yes	No	No
	0110	Yes*	Yes	Yes	Yes	Yes	No
	0111	Yes*	Yes	Yes	Yes	Yes	Yes
	1000	Yes*	Yes	Yes	Yes	Yes	Yes
	1001	Yes*	Yes	Yes	Yes	Yes	Yes
	1010	Yes*	Yes	Yes	Yes	Yes	Yes
	1011	Yes*	Yes	Yes	Yes	Yes	Yes
	1100	Yes*	Yes	Yes	Yes	Yes	Yes
	1101	Yes*	Yes	Yes	Yes	Yes	Yes
	1110	Yes*	Yes	Yes	Yes	Yes	Yes
	1111	Yes*	Yes	Yes	Yes	Yes	Yes

Notes: The time quanta value for TSEG1 and TSEG2 is the TSEG value + 1.

* Setting is enabled except when BRP [13:8] = B'000000.

Mailbox Transmit/Receive Settings: HCAN0, 1 each have 16 mailboxes. Mailbox 0 i
only, while mailboxes 1 to 15 can be set for transmission or reception. Mailboxes that c
for transmission or reception must be designated either for transmission use or for recep
before communication begins. The Initial status of mailboxes 1 to 15 is for transmissio
mailbox 0 is for reception only). Mailbox transmit/receive settings are not initialized by
reset.

- Setting for transmission

  Transmit mailbox setting (mailboxes 1 to 15)

  Clearing a bit to 0 in the mailbox configuration register (MBCR) designates the cor
  mailbox for transmission use. After a reset, mailboxes are initialized for transmissio
  this setting is not necessary.

- Setting for reception

  Transmit/receive mailbox setting (mailboxes 1 to 15)

all registers and RAM (message control/data, control registers, status registers, etc.) ar
Message control/data (MCx[x], MDx[x]) only are in RAM, and so their values are unc
Initial values must therefore be set in all the mailboxes (by writing 0s or 1s).

Setting the Message Transmission Method: Either of the following message transm
methods can be selected with the message transmission method bit (MCR2) in the ma
register (MCR):

a. Transmission order determined by message identifier priority
b. Transmission order determined by mailbox number priority

When a is selected, if a number of messages are designated as waiting for transmissio
1), the message with the highest priority set in the message identifier (MCx[5]–MCx[8
in the transmit buffer. CAN bus arbitration is then carried out for the message in the tr
buffer, and message transmission is performed when the transmission right is acquired
TXPR bit is set, internal arbitration is performed again, and the highest-priority messa
and stored in the transmit buffer.

When b is selected, if a number of messages are designated as waiting for transmissio
1), messages are stored in the transmit buffer in low-to-high mailbox order (priority or
mailbox 1 > mailbox 15). CAN bus arbitration is then carried out for the messages in t
buffer, and message transmission is performed when the bus is acquired.

b. Bit rate settings
c. Mailbox transmit/receive settings
d. Mailbox initialization
e. Message transmission method setting

Interrupt and transmit data settings

a. CPU interrupt source setting
b. Arbitration field setting
c. Control field setting
d. Data field setting

Message transmission and interrupts

a. Message transmission wait
b. Message transmission completion and interrupt
c. Message transmission abort
d. Message retransmission

Set values relating to the CAN bus communication speed and resynchronization. R
Rate and Bit Timing Settings in 15.3.2, Initialization after Hardware Reset, for det

- Mailbox transmit/receive settings

  Mailbox transmit/receive settings should be made in advance. A total of 15 mailbo
  set for transmission or reception (mailboxes 1 to 15). To set a mailbox for transmis
  the corresponding bit to 0 in the mailbox configuration register (MBCR). Refer to
  Transmit/Receive Settings in 15.3.2, Initialization after Hardware Reset, for detail

- Mailbox initialization

  As message control/data registers (MCx[x], MDx[x]) are configured in RAM, thei
  values after powering on are undefined, and so bit initialization is necessary. Write
  the mailboxes. See Mailbox (Message Control/Data (MCx[x], MDx[x]) Initial Set
  15.3.2, Initialization after Hardware Reset, for details.

- Message transmission method setting

  Set the transmission method for mailboxes designated for transmission. The follow
  transmission methods can be used. Refer to Setting the Message Transmission Me
  15.3.2, Initialization after Hardware Reset, for details.

  a. Transmission order determined by message identifier priority
  b. Transmission order determined by mailbox number priority

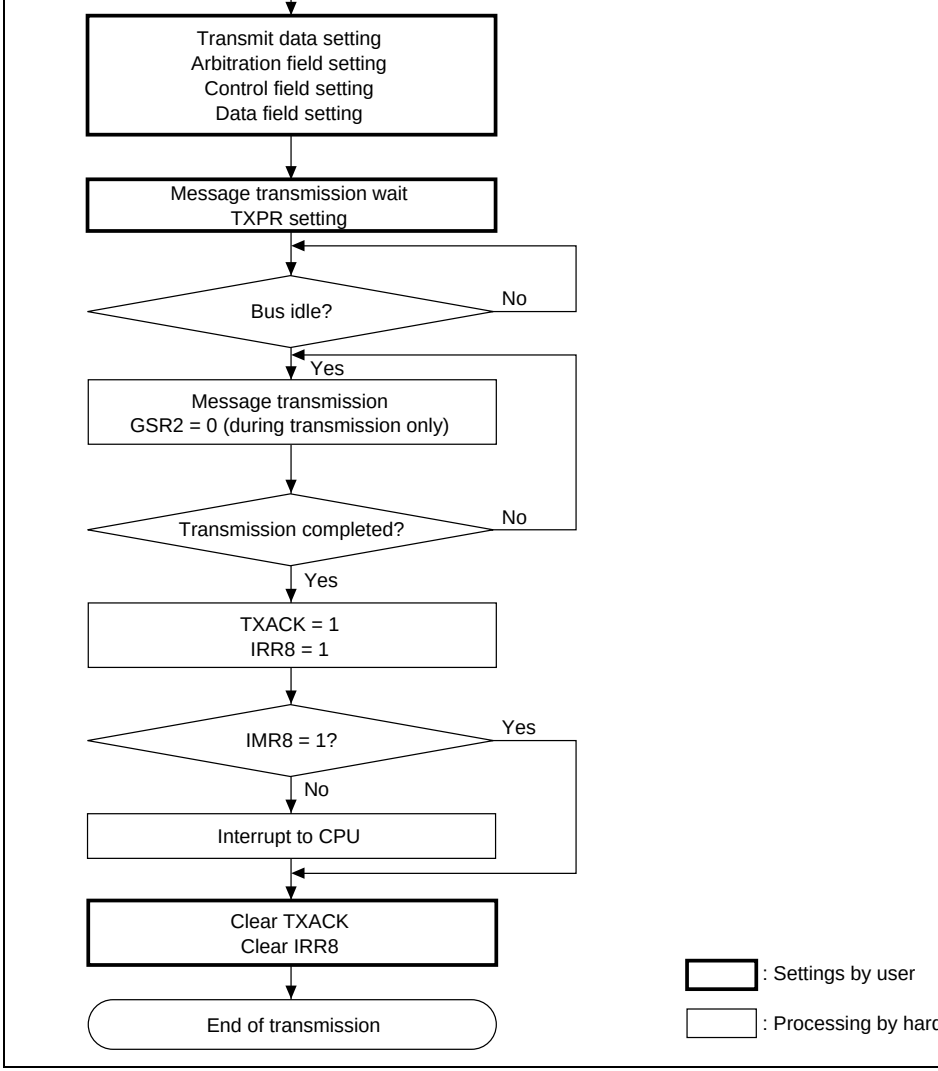


Figure 15.7 Transmission Flowchart

Transmission acknowledge and transmission abort acknowledge interrupts can be
individual mailboxes in the mailbox interrupt mask register (MBIMR). Interrupt re
interrupts can be masked in the interrupt mask register (IMR).

- Arbitration field setting

  In the arbitration field, the 11-bit identifier (STD_ID0–STD_ID10) and RTR bit (s
  format) or 29-bit identifier (STD_ID0–STD_ID10, EXT_ID0–EXT_ID17) and ID
  (extended format) are set. The registers to be set are MCx[5]–MCx[8].

- Control field setting

  In the control field, the byte length of the data to be transmitted is set in DLC0–DI
  register to be set is MCx[1].

- Data field setting

  In the data field, the data to be transmitted is set in byte units in the range of 0 to 8
  registers to be set are MDx[1]–MDx[8].

The number of bytes in the data actually transmitted depends on the data length code (
control field. If a value exceeding the value set in DLC is set in the data field, only the
bytes set in DLC will actually be transmitted.

Message Transmission and Interrupts:

- Message transmission wait

  If message transmission is to be performed after completion of the message contro
  MCx[8]) and message data (MDx[1]–MDx[8]) settings, transmission is started by
  corresponding mailbox transmit wait bit (TXPR1–TXPR15) to 1 in the transmit w
  (TXPR). The following two transmission methods can be used:

  a. Transmission order determined by message identifier priority
  b. Transmission order determined by mailbox number priority

  When a is selected, if a number of messages are designated as waiting for transmis
  = 1), messages are stored in the transmit buffer in low-to-high mailbox order (prio

same way, and message transmission is performed when the transmission right is ac

- Message transmission completion and interrupt

  When a message is transmitted error-free using the above procedure, the correspond acknowledge bit (TXACK1–TXACK15) in the transmit acknowledge register (TXA transmit wait bit (TXPR1–TXPR15) in the transmit wait register (TXPR) are autom initialized. Also, if the corresponding bit (MBIMR1–MBIMR15) in the mailbox int register (MBIMR) and the mailbox empty interrupt bit (IRR8) in the interrupt mask (IMR) are set to the interrupt enable state at the same time, an interrupt can be sent

- Message transmission cancellation

  Transmission cancellation can be specified for a message stored in a mailbox as a tr message. A transmit wait message is canceled by setting the bit for the correspondi (TXCR1–TXCR15) to 1 in the transmit cancel register (TXCR). When cancellation executed, the transmit wait register (TXPR) is automatically reset, and the correspo set to 1 in the abort acknowledge register (ABACK). An interrupt to the CPU can b Also, if the mailbox empty interrupt (IRR8) is enabled for the bits (MBIMR1–MBI corresponding to the mailbox interrupt mask register (MBIMR) and interrupt mask (IMR), interrupts may be sent to the CPU.

  However, a transmit wait message cannot be canceled at the following times:

  a. During internal arbitration or CAN bus arbitration
  b. During data frame or remote frame transmission

  Also, transmission cannot be canceled by clearing the transmit wait register (TXPR 15.8 shows a flowchart of transmit message cancellation.

- Message retransmission

  If transmission of a transmit message is aborted in the following cases, the message retransmitted automatically:

  a. CAN bus arbitration failure (failure to acquire the bus)
  b. Error during transmission (bit error, stuff error, CRC error, frame error, ACK

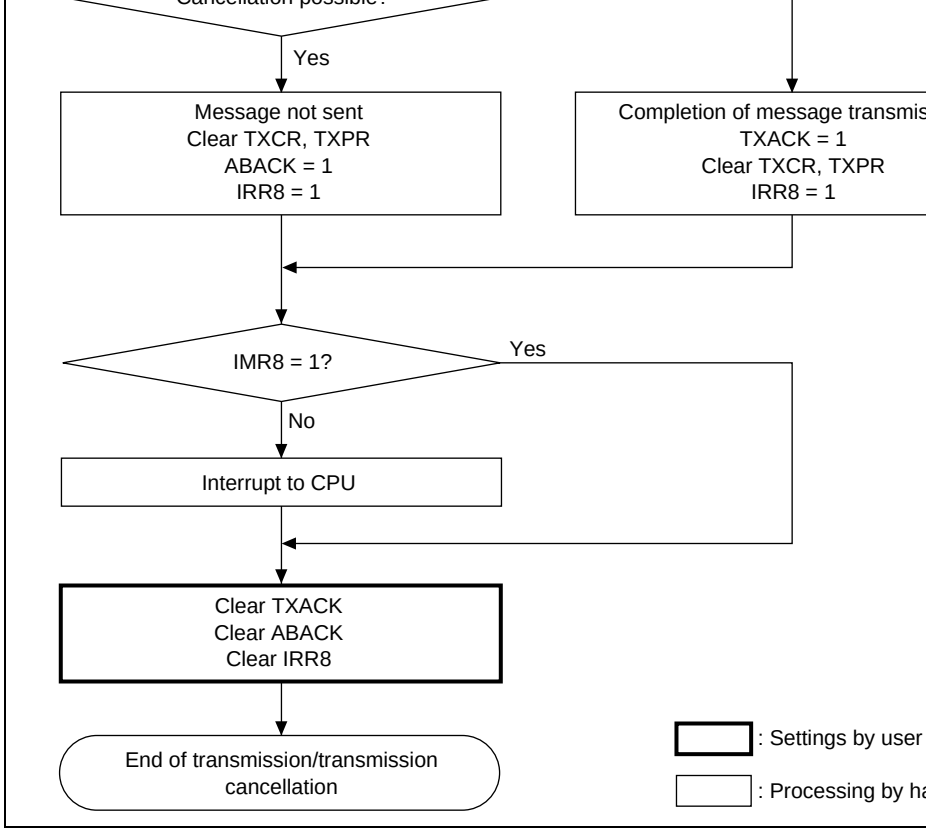


Figure 15.8 Transmit Message Cancellation Flowchart

b. Bit rate settings
c. Mailbox transmit/receive settings
d. Mailbox (RAM) initialization

Interrupt and receive message settings

a. CPU interrupt source setting
b. Arbitration field setting
c. Local acceptance filter mask (LAFM) settings

Message reception and interrupts

a. Message reception CRC check
b. Data frame reception
c. Remote frame reception
d. Unread message reception

Initialization (After Hardware Reset Only): These settings should be made while the
in bit configuration mode.

- IRR0 clearing

  The reset interrupt flag (IRR0) is always set after a reset or recovery from software
  mode. As an HCAN interrupt is initiated immediately when interrupts are enabled,
  should be cleared.

- Bit rate settings

  Set values relating to the CAN bus communication speed and resynchronization. Re
  Rate and Bit Timing Setting in 15.3.2, Initialization after Hardware Reset, for detail

- Mailbox transmit/receive settings

  Each channel has one receive-only mailbox (mailbox 0) plus 15 mailboxes that can
  reception. Thus a total of 16 mailboxes can be used for reception. To set a mailbox
  reception, set the corresponding bit to 1 in the mailbox configuration register (MBC

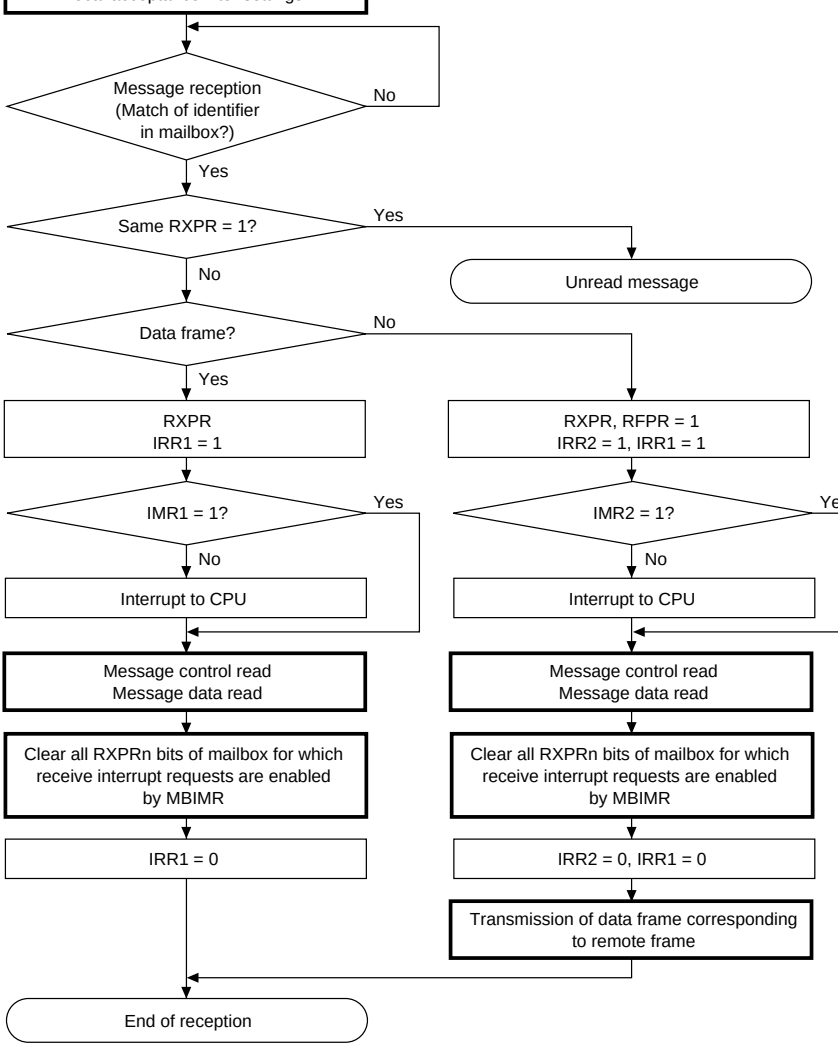


Figure 15.9 Reception Flowchart

- CPU interrupt source settings

  When transmitting, transmission acknowledge and transmission abort acknowledg
  can be masked for individual mailboxes in the mailbox interrupt mask register (MI
  When receiving, data frame and remote frame receive wait interrupts can be mask
  register (IRR) interrupts can be masked in the interrupt mask register (IMR).

- Arbitration field setting

  In the arbitration field, the identifier (STD_ID0–STD_ID10, EXT_ID0–EXT_ID1
  message to be received is set. If all the bits in the set identifier do not match, the m
  stored in a mailbox.

  Example: Mailbox 1 010_1010_1010 (standard identifier)

  Only one kind of message identifier can be received by MB1
  Identifier 1: 010_1010_1010

- Local acceptance filter mask (LAFM) setting

  The local acceptance filter mask is provided for mailbox 0 (MC0[x], MD0[x]) only
  Don’t care specification to be made for all bits in the received identifier. This allow
  kinds of messages to be received.

  Example: Mailbox 0 010_1010_1010 (standard identifier)
  LAFM 000_0000_0011 (0: Care, 1: Don’t care)

  A total of four kinds of message identifiers can be received by MB0
  Identifier 1: 010_1010_1000
  Identifier 2: 010_1010_1001
  Identifier 3: 010_1010_1010
  Identifier 4: 010_1010_1011

the mailbox (and also LAFM in the case of mailbox 0 only) and the identifier of the
message are compared, and if a complete match is found, the message is stored in th
The message identifier comparison is carried out on each mailbox in turn, starting w
mailbox 0 and ending with mailbox 15. If a complete match is found, the comparisc
that point, the message is stored in the matching mailbox, and the corresponding rec
complete bit (RXPR0–RXPR15) is set in the receive complete register (RXPR). Ho
when a mailbox 0 LAFM comparison is carried out, even if the identifier matches, t
comparison sequence does not end at that point, but continues with mailbox 1 and th
remaining mailboxes. It is therefore possible for a message matching mailbox 0 to b
by another mailbox (however, the same message cannot be stored in more than one
mailboxes 1 to 15). If the corresponding bit (MBIMR0–MBIMR15) in the mailbox
mask register (MBIMR) and the receive message interrupt mask (IMR1) in the inter
register (IMR) are set to the interrupt enable value at this time, an interrupt can be s
CPU.

- Remote frame reception

  Two kinds of messages—data frames and remote frames—can be stored in mailbox
  remote frame differs from a data frame in that the remote reception request bit (RTI
  message control register (MC[x]5) and the data field are 0 bytes. The data length to
  in a data frame must be stored in the data length code (DLC) in the control field.

  When a remote frame (RTR = recessive) is received, the corresponding bit is set in
  request wait register (RFPR). If the corresponding bit (MBIMR0–MBIMR15) in the
  interrupt mask register (MBIMR) and the remote frame request interrupt mask (IRR
  interrupt mask register (IMR) are set to the interrupt enable value at this time, an in
  be sent to the CPU.

- Unread message reception

  When the identifier in a mailbox matches a receive message, the message is stored i
  mailbox. If a message overwrite occurs before the CPU reads the message, the corre
  bit (UMSR0–UMSR15) is set in the unread message register (UMSR). In overwritin
  unread message, when a new message is received before the corresponding bit in th
  complete register (RXPR) has been cleared, the unread message register (UMSR) is

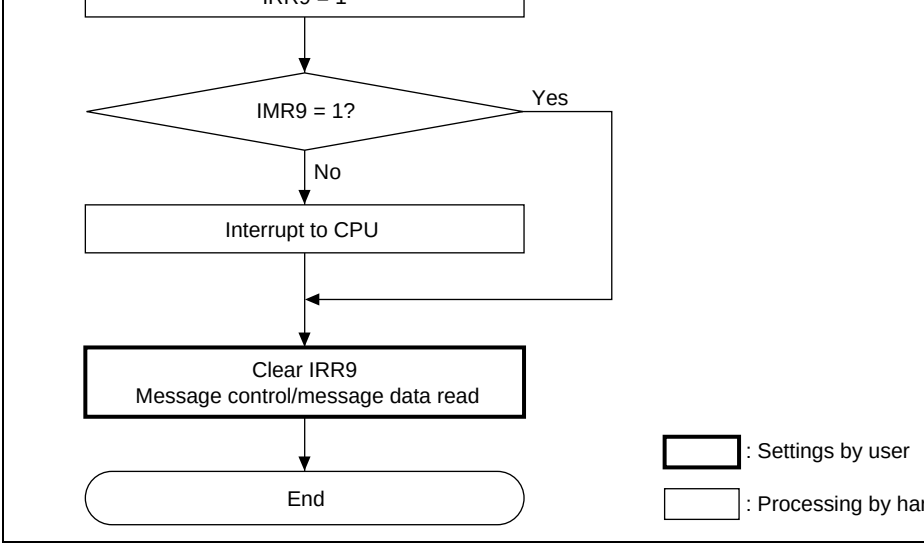


Figure 15.10 Unread Message Overwrite Flowchart

15.3.5 HCAN Sleep Mode

The HCAN is provided with an HCAN sleep mode that places the HCAN module in t
state to reduce current dissipation. Figure 15.11 shows a flowchart of the HCAN sleep

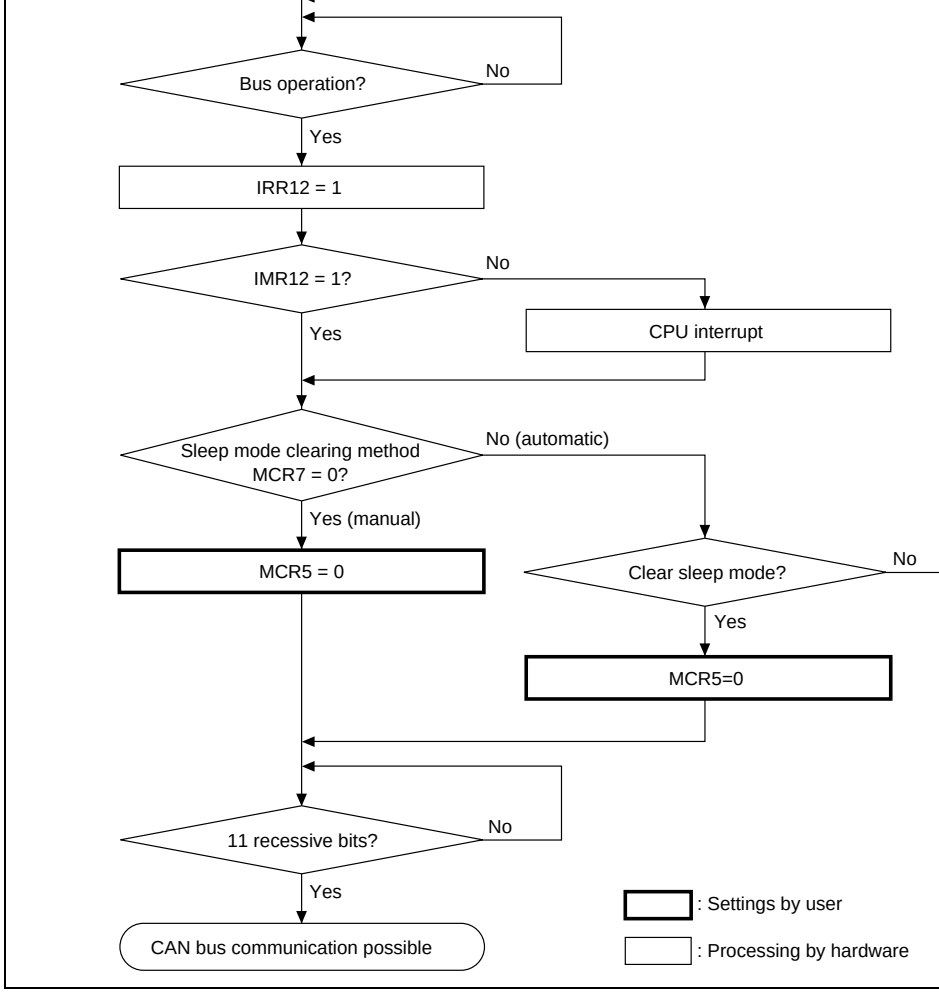


Figure 15.11 HCAN Sleep Mode Flowchart

Eleven recessive bits must be received after HCAN sleep mode is cleared before CAN communication is enabled again.

Clearing by software: HCAN sleep mode is cleared by writing a 0 to MCR5 from the

Clearing by CAN bus operation: Clearing by CAN bus operation occurs automatica CAN bus performs an operation and this change is detected. In this case, the first mess received in the mailbox, and normal reception starts from the next message. When a c detected on the CAN bus in HCAN sleep mode, the bus operation interrupt flag (IRR1 the interrupt register (IRR). If the bus interrupt mask (IMR12) in the interrupt mask re is set to the interrupt enable value at this time, an interrupt can be sent to the CPU.

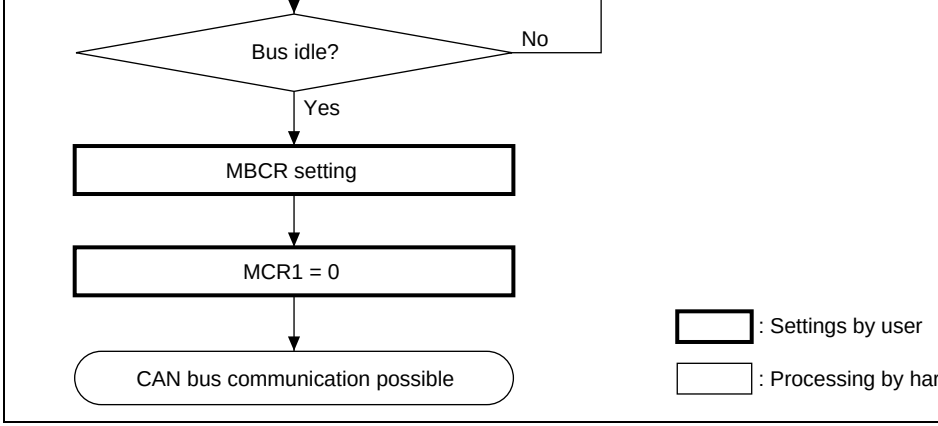


Figure 15.12 HCAN Halt Mode Flowchart

HCAN halt mode is entered by setting the halt request bit (MCR1) to 1 in the master co register (MCR). If the CAN bus is operating, the transition to HCAN halt mode is delay the bus becomes idle.

HCAN halt mode is cleared by clearing MCR1 to 0.

Table 15.5 HCAN Interrupt Sources

IPR Bits	Vector	Vector Number	IRR Bit	Description
IPRM (6–4)	ERS0	104	IRR5	Error passive interrupt (TEC ≥ 128 o 128)
			IRR6	Bus off interrupt (TEC ≥ 256)
	OVR0	105	IRR0	Hardware reset processing interrupt
			IRR2	Remote frame reception interrupt
			IRR3	Error warning interrupt (TEC ≥ 96)
			IRR4	Error warning interrupt (REC ≥ 96)
			IRR7	Overload frame transmission interru recovery interrupt (11 recessive bits
			IRR9	Unread message overwrite interrupt
			IRR12	HCAN sleep mode CAN bus operati
	RM0	106	IRR1	Mailbox 0 message reception interru
	RM1	107	IRR1	Mailbox 1–15 message reception int
IPRM (2–0)	SLE0	108	IRR8	Message transmission/cancellation i

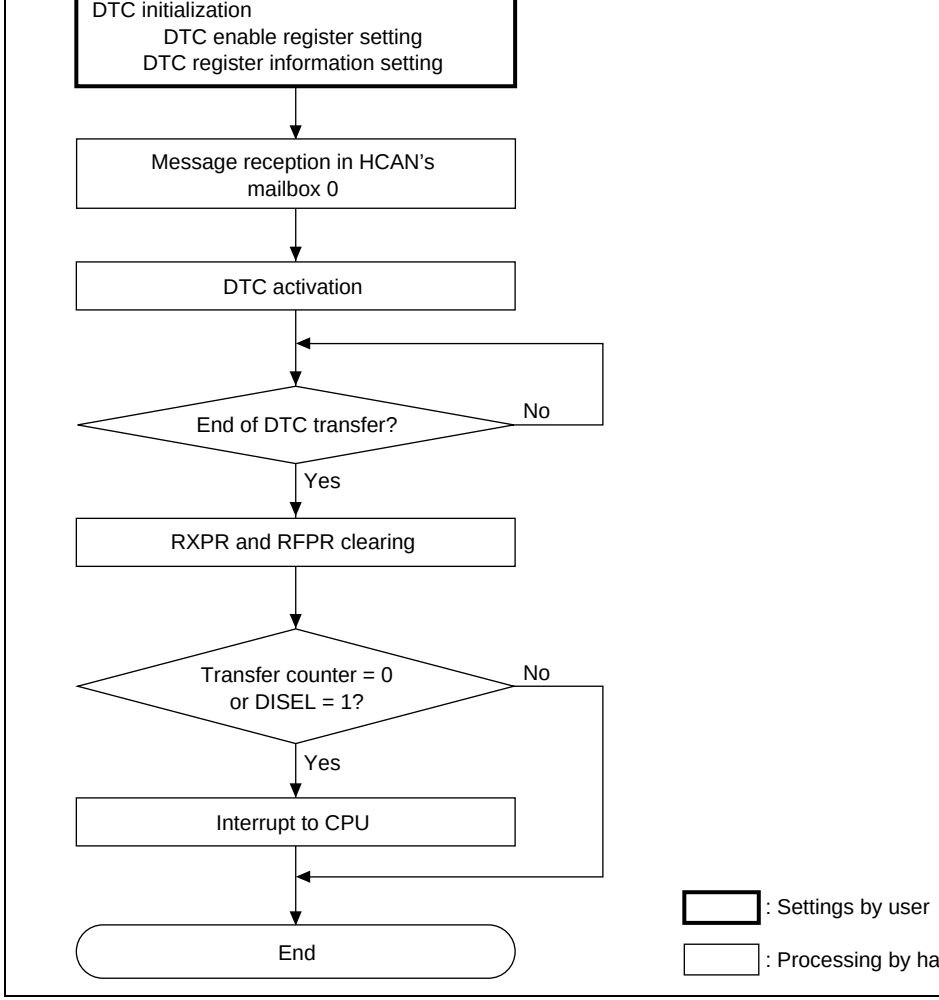


Figure 15.13 DTC Transfer Flowchart

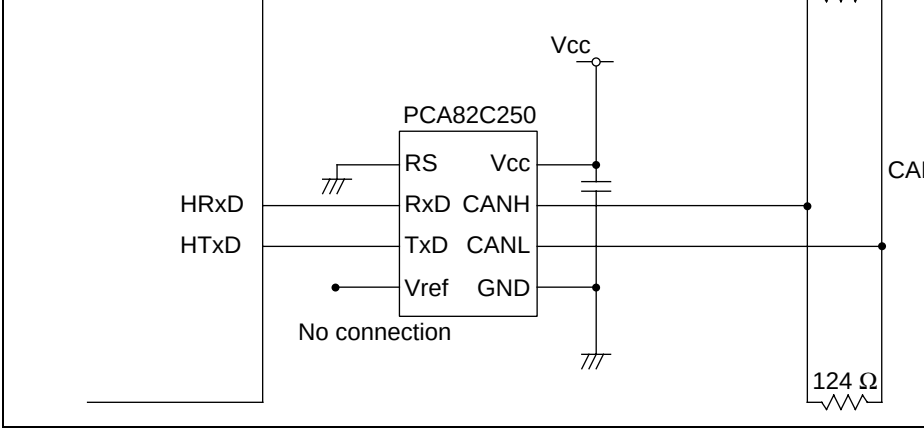


Figure 15.14 High-Speed Interface Using PCA82C250

15.5 Usage Notes

1. Reset

   The HCAN is reset by a reset, and in hardware standby mode and software standby the registers are initialized in a reset, but mailboxes (message control (MCx[x])/m (MDx[x]) are not. However, after powering on, mailboxes (message control (MCx[x])/message data (MDx[x]) are initialized, and their values are undefined. T mailbox initialization must always be carried out after a reset or a transition to har standby mode or software standby mode. Also, the reset interrupt flag (IRR0) is al after reset input or recovery from software standby mode. As this bit cannot be ma interrupt mask register (IMR), if HCAN interrupts are set as enabled by the interru without this flag having been cleared, an HCAN interrupt will be initiated immedi must therefore be cleared during initialization.

2. HCAN sleep mode

   The bus operation interrupt flag (IRR12) in the interrupt register (IRR) is set by bu in HCAN sleep mode. Therefore, this flag is not used by the HCAN to indicate sle

In the case of error active and error passive, REC and TEC normally count up and d
bus off state, 11-bit recessive sequences are counted (REC + 1) using REC. If REC
during the count, IRR4 and GSR1 are set, and if REC reaches 128, IRR7 is set.

5. Register access

   Byte or word access can be used on all HCAN registers. Longword access cannot b

6. HCAN medium-speed mode

   HCAN registers cannot be read or written to in medium-speed mode.

7. Register retention during standby

   All HCAN registers are initialized in hardware standby mode and software standby

8. Using bit operation instructions

   Start flags in HCAN are cleared by writing 1 to them; there is no need to use bit ope
   instructions to clear them. To clear a flag, use the MOV instruction to write a 1 to th
   cleared.

9. HTxD pin output in error passive state

   If the HRxD pin becomes fixed at 1 during message transmission or reception when
   is in the error active state, the HTxD pin will output 0 continuously while in the erro
   state. To stop continuous 0 output to the CAN bus, disable the HCAN by means of
   warning interrupt or by setting the HCAN module stop mode through detection of a
   state by the HxRD pin monitor.

10. Transition to HCAN sleep mode

    The HCAN stops (transmission/reception stops) when MCR0 is cleared to 0 immed
    an HCAN sleep mode transition effected by setting TXPR of the HCAN to 1 and se
    MCR5 to 1. When a transition is made to the HCAN sleep mode by means of the a
    a 10-cycle wait should be inserted after the TxPR setting. After an HCAN sleep mo
    transition, release the HCAN sleep mode by clearing MCR5 to 0.

If TxPR is set before the HCAN goes to the bus off state, and a transition is made t
state with transmission incomplete, cancellation will be performed even if TxCR is
the bus off period, and the message will be transmitted after a transition to the erro
state.

16.1.1 Features

A/D converter features are listed below.

- 10-bit resolution
- Sixteen input channels
- Settable analog conversion voltage range
  - — Conversion of analog voltages with the reference voltage pin (Vref) as the anal voltage
- High-speed conversion
  - — Minimum conversion time: 13.3 µs per channel (at 20-MHz operation)
- Choice of single mode or scan mode
  - — Single mode: Single-channel A/D conversion
  - — Scan mode: Continuous A/D conversion on 1 to 4 channels
- Four data registers
  - — Conversion results are held in a 16-bit data register for each channel
- Sample and hold function
- Three kinds of conversion start
  - — Choice of software or timer conversion start trigger (TPU), or $\overline{\text{ADTRG}}$ pin
- A/D conversion end interrupt generation
  - — A/D conversion end interrupt (ADI) request can be generated at the end of A/D
- Module stop mode can be set
  - — As the initial setting, A/D converter operation is halted. Register access is enab exiting module stop mode.

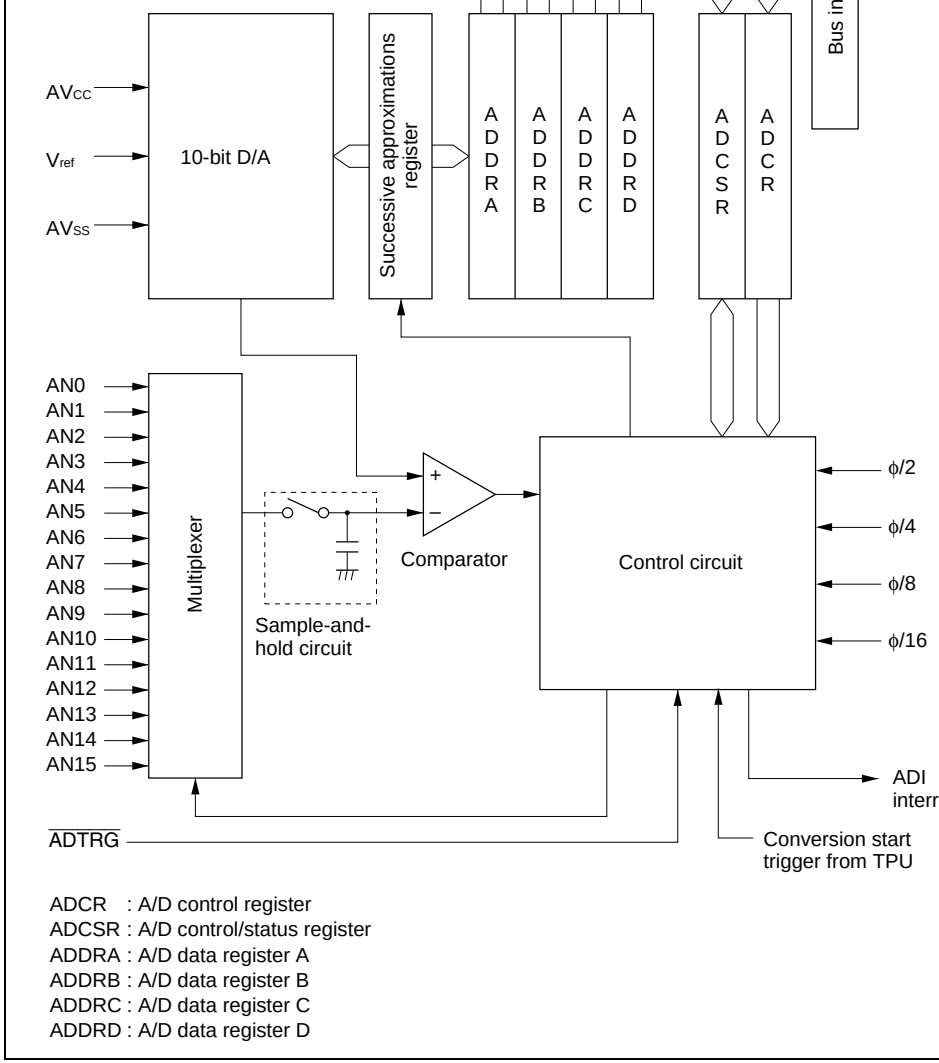


Figure 16.1 Block Diagram of A/D Converter

comprising channel set 1, analog input pins 0 to 3 and 8 to 11 (AN0 to AN3, AN8 to A
comprising group 0, and analog input pins 4 to 7 and 12 to 15 (AN4 to AN7, AN12 to
comprising group 1.

Table 16.1 A/D Converter Pins

Pin Name	Symbol	I/O	Function
Analog power supply pin	AVCC	Input	Analog block power supply
Analog ground pin	AVSS	Input	Analog block ground and reference
Reference voltage pin	Vref	Input	A/D conversion reference voltage
Analog input pin 0	AN0	Input	Channel set 0 (CH3 = 0) group 0 an
Analog input pin 1	AN1	Input	
Analog input pin 2	AN2	Input	
Analog input pin 3	AN3	Input	
Analog input pin 4	AN4	Input	Channel set 0 (CH3 = 0) group 1 an
Analog input pin 5	AN5	Input	
Analog input pin 6	AN6	Input	
Analog input pin 7	AN7	Input	
Analog input pin 8	AN8	Input	Channel set 1 (CH3 = 1) group 0 an
Analog input pin 9	AN9	Input	
Analog input pin 10	AN10	Input	
Analog input pin 11	AN11	Input	
Analog input pin 12	AN12	Input	Channel set 1 (CH3 = 1) group 1 an
Analog input pin 13	AN13	Input	
Analog input pin 14	AN14	Input	
Analog input pin 15	AN15	Input	
A/D external trigger input pin	$\overline{\text{ADTRG}}$	Input	External trigger input for starting A/D

A/D data register BH	ADDRBH	R	H'00	H'
A/D data register BL	ADDRBL	R	H'00	H'
A/D data register CH	ADDRCH	R	H'00	H'
A/D data register CL	ADDRCL	R	H'00	H'
A/D data register DH	ADDRDH	R	H'00	H'
A/D data register DL	ADDRDL	R	H'00	H'
A/D control/status register	ADCSR	R/(W)*2	H'00	H'
A/D control register	ADCR	R/W	H'33	H'
Module stop control register A	MSTPCRA	R/W	H'3F	H'

Notes: 1. Lower 16 bits of the address.
2. Bit 7 can only be written with 0 for flag clearing.

R/W : R R R R R R R R R R R R R R R R

There are four 16-bit read-only ADDR registers, ADDRA to ADDRD, used to store th
A/D conversion.

The 10-bit data resulting from A/D conversion is transferred to the ADDR register for
channel and stored there. The upper 8 bits of the converted data are transferred to the
(bits 15 to 8) of ADDR, and the lower 2 bits are transferred to the lower byte (bits 7 a
stored. Bits 5 to 0 are always read as 0.

The correspondence between the analog input channels and ADDR registers is shown
table 16.3.

ADDR can always be read by the CPU. The upper byte can be read directly, but for th
byte, data transfer is performed via a temporary register (TEMP). For details, see secti
Interface to Bus Master.

The ADDR registers are initialized to H'0000 by a reset, and in standby mode or modu
mode.

Table 16.3 Analog Input Channels and Corresponding ADDR Registers

Analog Input Channel				
Channel Set 0 (CH3 = 0)		Channel Set 1 (CH3 = 1)		
Group 0	Group 1	Group 0	Group 1	A/D Data Registe
AN0	AN4	AN8	AN12	ADDRA
AN1	AN5	AN9	AN13	ADDRB
AN2	AN6	AN10	AN14	ADDRC
AN3	AN7	AN11	AN15	ADDRD

Note: * Only 0 can be written to bit 7, to clear this flag.

ADCSR is an 8-bit readable/writable register that controls A/D conversion operations.

ADCSR is initialized to H'00 by a reset, and in hardware standby mode or module stop

Bit 7—A/D End Flag (ADF): Status flag that indicates the end of A/D conversion.

Bit 7

ADF	Description
0	[Clearing conditions] (In • When 0 is written to the ADF flag after reading ADF = 1 • When the DTC is activated by an ADI interrupt and ADDR is read
1	[Setting conditions] • Single mode: When A/D conversion ends • Scan mode: When A/D conversion ends on all specified channels

Bit 6—A/D Interrupt Enable (ADIE): Selects enabling or disabling of interrupt (ADI at the end of A/D conversion.

Bit 6

ADIE	Description
0	A/D conversion end interrupt (ADI) request disabled (In
1	A/D conversion end interrupt (ADI) request enabled

0	• A/D conversion stopped
1	• Single mode: A/D conversion is started. Cleared to 0 automatically wh conversion on the specified channel ends • Scan mode: A/D conversion is started. Conversion continues sequer selected channels until ADST is cleared to 0 by software a transition to standby mode or module stop mode.

Bit 4—Scan Mode (SCAN): Selects single mode or scan mode as the A/D conversio mode. See section 16.4, Operation, for single mode and scan mode operation. Only se bit while conversion is stopped (ADST = 0).

Bit 4

SCAN	Description
0	Single mode (I
1	Scan mode

Bit 3—Channel Select 3 (CH3): Switches the analog input pins assigned to group 0 c Setting CH3 to 1 enables AN8 to AN15 to be used instead of AN0 to AN7.

Bit 3

CH3	Description
0	AN8 to AN11 are group 0 analog input pins, AN12 to AN15 are group 1 ar pins
1	AN0 to AN3 are group 0 analog input pins, AN4 to AN7 are group 1 analog (

			1	AN1	AN0, AN1
		1	0	AN2	AN0 to AN2
			1	AN3	AN0 to AN3
	1	0	0	AN4	AN4
			1	AN5	AN4, AN5
		1	0	AN6	AN4 to AN6
			1	AN7	AN4 to AN7
1	0	0	0	AN8	AN8
			1	AN9	AN8, AN9
		1	0	AN10	AN8 to AN10
			1	AN11	AN8 to AN11
	1	0	0	AN12	AN12
			1	AN13	AN12, AN13
		1	0	AN14	AN12 to AN14
			1	AN15	AN12 to AN15

conversion operations and sets the A/D conversion time.

ADCR is initialized to H'33 by a reset, and in standby mode or module stop mode.

Bits 7 and 6—Timer Trigger Select 1 and 0 (TRGS1, TRGS0): Select enabling or
the start of A/D conversion by a trigger signal. Only set bits TRGS1 and TRGS0 whil
is stopped (ADST = 0).

Bit 7	Bit 6	
TRGS1	**TRGS0**	**Description**
0	0	A/D conversion start by software is enabled
	1	A/D conversion start by TPU conversion start trigger is enable
1	0	Setting prohibited
	1	A/D conversion start by external trigger pin ($\overline{\text{ADTRG}}$) is enabl

Bits 5, 4, 1, and 0—Reserved: These bits are always read as 1 and cannot be modifie

Bits 3 and 2—Clock Select 1 and 0 (CKS1, CKS0): These bits select the A/D conve
The conversion time should be changed only when ADST = 0. Make a setting that giv
not lower than that shown in table 22-8.

Bit 3	Bit 2		
CKS1	**CKS0**	**Description**	
0	0	Conversion time = 530 states (max.)	(
	1	Conversion time = 266 states (max.)	
1	0	Conversion time = 134 states (max.)	
	1	Conversion time = 68 states (max.)	

When the MSTPA1 bit in MSTPCR is set to 1, A/D converter operation stops at the en
cycle and a transition is made to module stop mode. Registers cannot be read or written
module stop mode. For details, see sections 21A.5, 21B.5, Module Stop Mode.

MSTPCRA is initialized to H'3F by a reset and in hardware standby mode. It is not init
software standby mode.

Bit 1—Module Stop (MSTPA1): Specifies the A/D converter module stop mode.

Bit 1

MSTPA1	Description
0	A/D converter module stop mode cleared
1	A/D converter module stop mode set (In

When reading ADDR. always read the upper byte before the lower byte. It is possible
the upper byte, but if only the lower byte is read, incorrect data may be obtained.

Figure 16.2 shows the data flow for ADDR access.

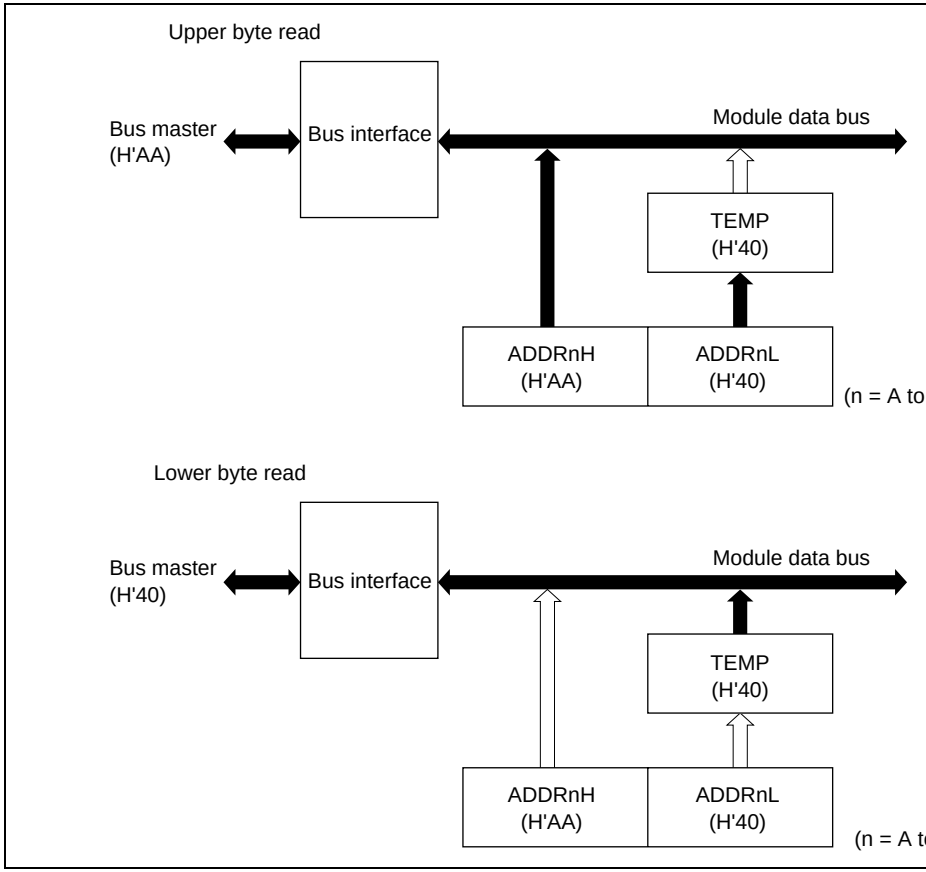


Figure 16.2 ADDR Access Operation (Reading H'AA40)

conversion is started when the ADST bit is set to 1, according to the software or extern
input. The ADST bit remains set to 1 during A/D conversion, and is automatically clear
when conversion ends.

On completion of conversion, the ADF flag is set to 1. If the ADIE bit is set to 1 at this
ADI interrupt request is generated. The ADF flag is cleared by writing 0 after reading A

When the operating mode or analog input channel must be changed during analog conv
prevent incorrect operation, first clear the ADST bit to 0 in ADCSR to halt A/D conver
making the necessary changes, set the ADST bit to 1 to start A/D conversion again. Th
bit can be set at the same time as the operating mode or input channel is changed.

Typical operations when channel 1 (AN1) is selected in single mode are described next
16.3 shows a timing diagram for this example.

[1] Single mode is selected (SCAN = 0), input channel AN1 is selected (CH3 = 0, CH2
CH1 = 0, CH0 = 1), the A/D interrupt is enabled (ADIE = 1), and A/D conversion i
(ADST = 1).

[2] When A/D conversion is completed, the result is transferred to ADDRB. At the sar
ADF flag is set to 1, the ADST bit is cleared to 0, and the A/D converter becomes id

[3] Since ADF = 1 and ADIE = 1, an ADI interrupt is requested.

[4] The A/D interrupt handling routine starts.

[5] The routine reads ADCSR, then writes 0 to the ADF flag.

[6] The routine reads and processes the connection result (ADDRB).

[7] Execution of the A/D interrupt handling routine ends. After that, if the ADST bit is
A/D conversion starts again and steps [2] to [7] are repeated.

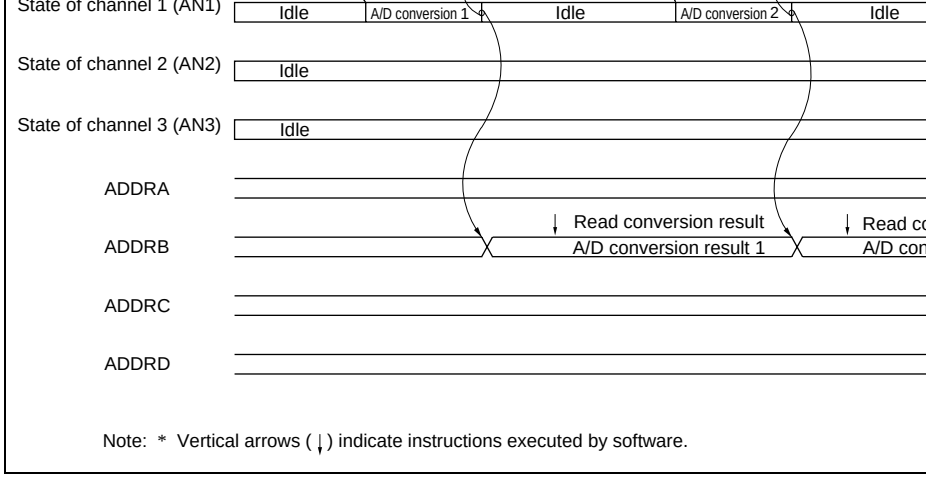


Figure 16.3 Example of A/D Converter Operation (Single Mode, Channel 1 S

When the operating mode or analog input channel must be changed during analog conv
prevent incorrect operation, first clear the ADST bit to 0 in ADCSR to halt A/D conver
making the necessary changes, set the ADST bit to 1 to start A/D conversion again fron
channel (AN0). The ADST bit can be set at the same time as the operating mode or inp
is changed.

Typical operations when three channels (AN0 to AN2) are selected in scan mode are de
next. Figure 16.4 shows a timing diagram for this example.

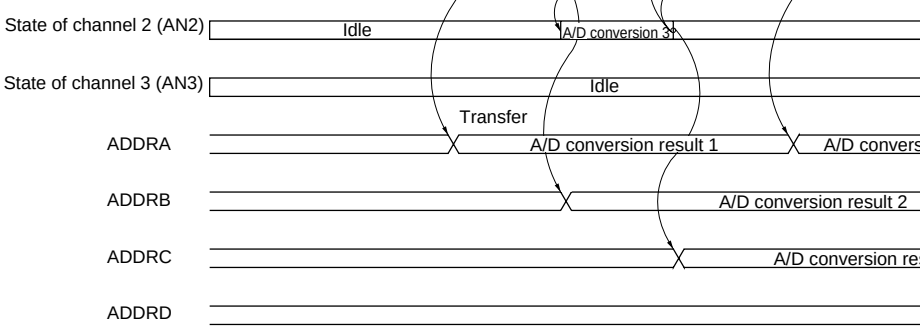
[1] Scan mode is selected (SCAN = 1), channel set 0 is selected (CH3 = 0), scan group
selected (CH2 = 0), analog input channels AN0 to AN2 are selected (CH1 = 1, CH0
A/D conversion is started (ADST = 1)

[2] When A/D conversion of the first channel (AN0) is completed, the result is transfer
ADDRA. Next, conversion of the second channel (AN1) starts automatically.

[3] Conversion proceeds in the same way through the third channel (AN2).

[4] When conversion of all the selected channels (AN0 to AN2) is completed, the ADF
to 1 and conversion of the first channel (AN0) starts again. If the ADIE bit is set to
time, an ADI interrupt is requested after A/D conversion ends.

[5] Steps [2] to [4] are repeated as long as the ADST bit remains set to 1. When the AI
cleared to 0, A/D conversion stops. After that, if the ADST bit is set to 1, A/D conv
starts again from the first channel (AN0).

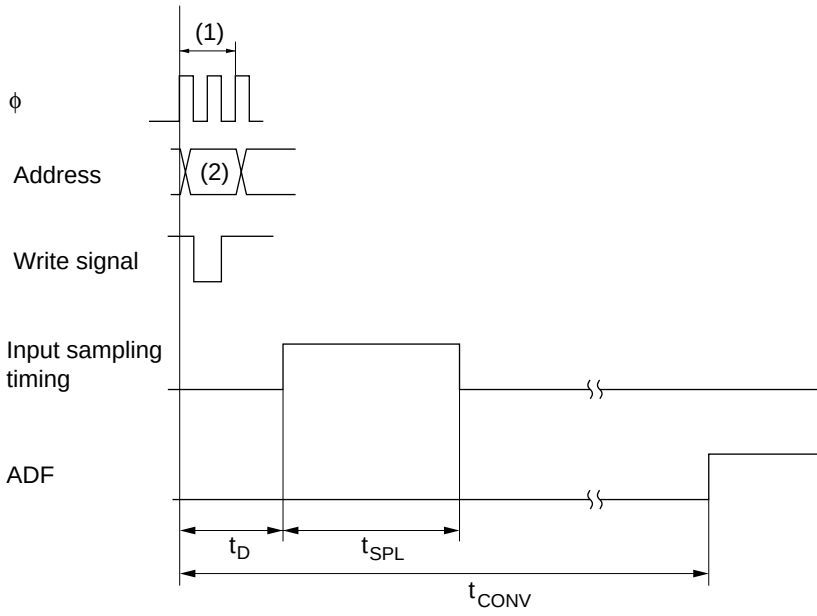


Notes: 1. Vertical arrows (↓) indicate instructions executed by software.
2. Data currently being converted is ignored.

Figure 16.4 Example of A/D Converter Operation (Scan Mode, 3 Channels AN0 to AN2 Selected)

time therefore varies within the ranges indicated in table 16.4.

In scan mode, the values given in table 16.4 apply to the first conversion time. The val
in table 16.5 apply to the second and subsequent conversions. In both cases, set bits CK
CKS0 in ADCR to give a value not lower than that shown in table 22-8.



Legend:
(1) : ADCSR write cycle
(2) : ADCSR address
t_D : A/D conversion start delay
t_{SPL} : Input sampling time
t_{CONV} : A/D conversion time

Figure 16.5 A/D Conversion Timing

A/D conversion time	t_{CONV}	515	—	530	259	—	266	131	—	134	67

Note: Values in the table are the number of states.

Table 16.5 A/D Conversion Time (Scan Mode)

CKS1	CKS0	Conversion Time (State)
0	0	512 (Fixed)
	1	256 (Fixed)
1	0	128 (Fixed)
	1	64 (Fixed)

16.4.4 External Trigger Input Timing

A/D conversion can be externally triggered. When the TRGS1 and TRGS0 bits are se
ADCR, external trigger input is enabled at the $\overline{\text{ADTRG}}$ pin. A falling edge at the $\overline{\text{AD}}$
the ADST bit to 1 in ADCSR, starting A/D conversion. Other operations, in both sing
modes, are the same as if the ADST bit has been set to 1 by software. Figure 16.6 sho
timing.

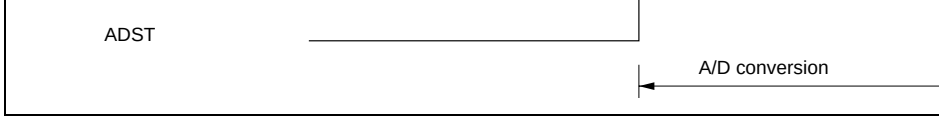


Figure 16.6 External Trigger Input Timing

16.5 Interrupts

The A/D converter generates an A/D conversion end interrupt (ADI) at the end of A/D
ADI interrupt requests can be enabled or disabled by means of the ADIE bit in ADCSR

The DTC can be activated by an ADI interrupt. Having the converted data read by the I
response to an ADI interrupt enables continuous conversion to be achieved without imp
load on software.

The A/D converter interrupt source is shown in table 16.6.

Table 16.6 A/D Converter Interrupt Source

Interrupt Source	Description	DTC Activation
ADI	Interrupt due to end of conversion	Possible

AVSS ≤ ANn ≤ Vref.

(2) Relation between AVCC, AVSS and VCC, VSS

As the relationship between AVCC, AVSS and VCC, VSS, set AVSS = VSS. If t converter is not used, the AVCC and AVSS pins must on no account be left open.

(3) Vref input range

The analog reference voltage input at the Vref pin set in the range Vref ≤ AVCC.

If conditions (1), (2), and (3) above are not met, the reliability of the device may be ad affected.

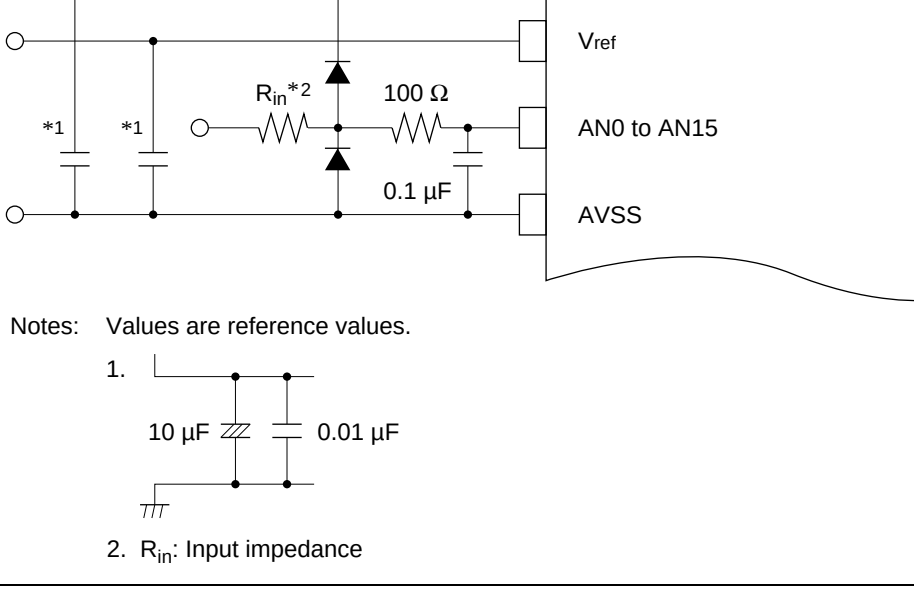
Notes on Board Design: In board design, digital circuitry and analog circuitry should mutually isolated as possible, and layout in which digital circuit signal lines and analo signal lines cross or are in close proximity should be avoided as far as possible. Failu may result in incorrect operation of the analog circuitry due to inductance, adversely a conversion values.

Also, digital circuitry must be isolated from the analog input signals (AN0 to AN15), reference power supply (Vref), and analog power supply (AVCC) by the analog groun Also, the analog ground (AVSS) should be connected at one point to a stable digital g on the board.

Notes on Noise Countermeasures: A protection circuit connected to prevent damage abnormal voltage such as an excessive surge at the analog input pins (AN0 to AN15) a reference power supply (Vref) should be connected between AVCC and AVSS as sho 16.7.

Also, the bypass capacitors connected to AVCC and Vref and the filter capacitor conn AN0 to AN15 must be connected to AVSS.

If a filter capacitor is connected as shown in figure 16.7, the input currents at the analo (AN0 to AN15) are averaged, and so an error may arise. Also, when A/D conversion frequently, as in scan mode, if the current charged and discharged by the capacitance



Notes: Values are reference values.

1.

2. R_{in}: Input impedance

Figure 16.7 Example of Analog Input Protection Circuit

Table 16.7 Analog Pin Specifications

Item	Min	Max	Unit
Analog input capacitance	—	20	pF
Permissible signal source impedance	—	5	kΩ

Figure 16.8 Analog Input Pin Equivalent Circuit

A/D Conversion Precision Definitions: H8S/2626 Group and H8S/2623 Group A/D
precision definitions are given below.

- Resolution

  The number of A/D converter digital output codes

- Offset error

  The deviation of the analog input voltage value from the ideal A/D conversion cha
  when the digital output changes from the minimum voltage value B'0000000000 (
  B'0000000001 (H'01) (see figure 16.10).

- Full-scale error

  The deviation of the analog input voltage value from the ideal A/D conversion cha
  when the digital output changes from B'1111111110 (H'3E) to B'1111111111 (H'3
  figure 16.10).

- Quantization error

  The deviation inherent in the A/D converter, given by 1/2 LSB (see figure 16.9).

- Nonlinearity error

  The error with respect to the ideal A/D conversion characteristic between the zero
  the full-scale voltage. Does not include the offset error, full-scale error, or quantiz

- Absolute precision

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

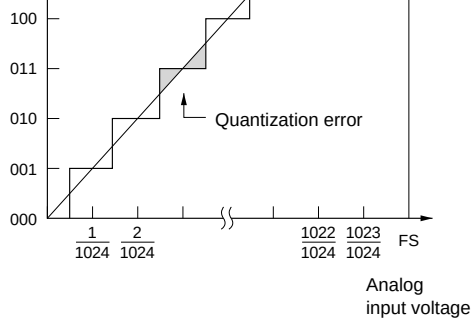


Figure 16.9 A/D Conversion Precision Definitions (1)

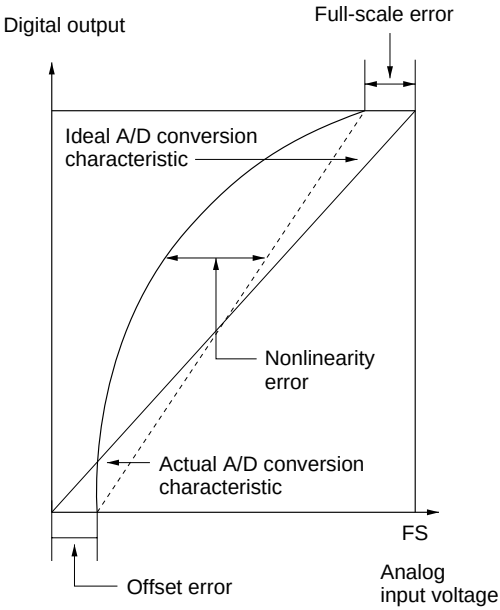


Figure 16.10 A/D Conversion Precision Definitions (2)

only the internal input resistance of 10 kΩ, and the signal source impedance is ignored

However, since a low-pass filter effect is obtained in this case, it may not be possible t
analog signal with a large differential coefficient (e.g., 5 mV/µs or greater).

When converting a high-speed analog signal, a low-impedance buffer should be insert

Influences on Absolute Precision: Adding capacitance results in coupling with GND
therefore noise in GND may adversely affect absolute precision. Be sure to make the
to an electrically stable GND such as AVSS.

Care is also required to insure that filter circuits do not communicate with digital signa
mounting board, so acting as antennas.

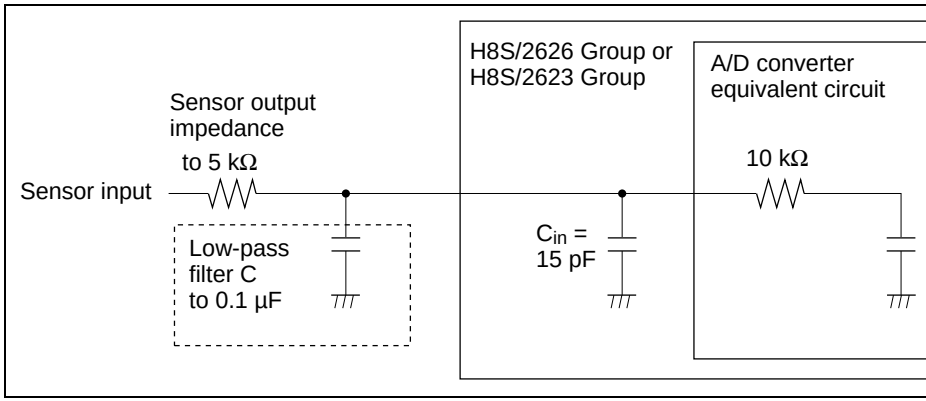


Figure 16.11 Example of Analog Input Circuit

17.1.1 Features

The D/A converter has the following features.

- 8-bit resolution
- Two output channels
- Conversion time: maximum 10 µs (with 20 pF capacitive load)
- Output voltage: 0 V to Vref
- D/A output retention in software standby mode
- Module stop mode setting possible
  - The initial setting is for D/A converter operation to be halted. Register access i clearing module stop mode.

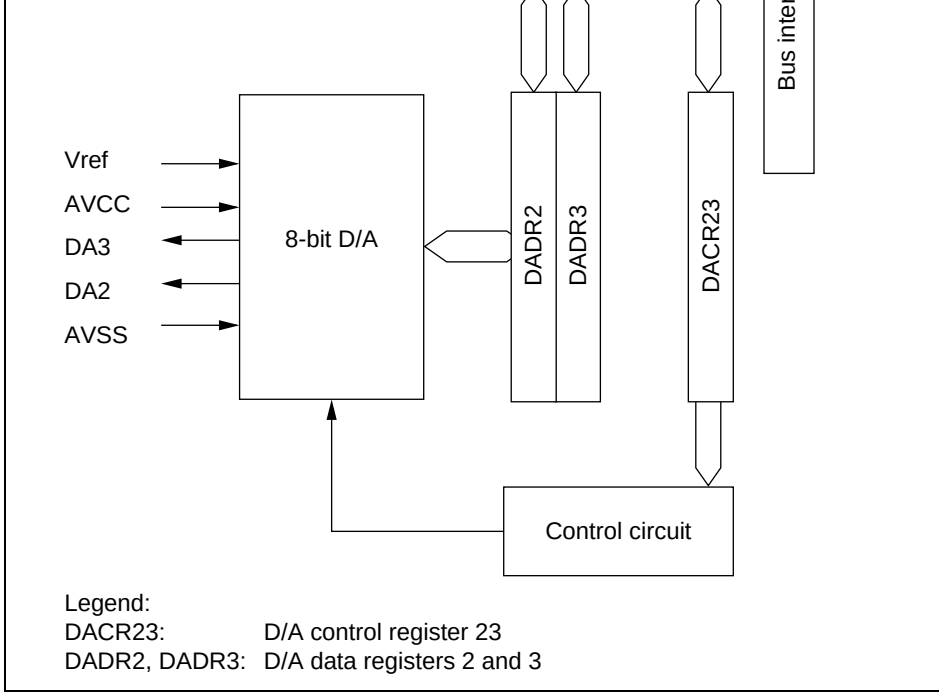


Figure 17.1 Block Diagram of D/A Converter

Analog output pin 2	DA2	Output	Channel 2 analog output
Analog output pin 3	DA3	Output	Channel 3 analog output
Reference voltage pin	Vref	Input	Analog reference voltage

17.1.4 Register Configuration

Table 17.2 summarizes the registers of the D/A converter.

Table 17.2 D/A Converter Registers

Channel	Name	Abbreviation	R/W	Initial Value
2, 3	D/A data register 2	DADR2	R/W	H'00
	D/A data register 3	DADR3	R/W	H'00
	D/A control register 23	DACR23	R/W	H'1F
All	Module stop control register C	MSTPCRC	R/W	H'FF

Note: * Lower 16 bits of the address

DADR2 and DADR3 are 8-bit readable/writable registers that store the data to be conv
When analog output is enabled, the values in DADR2 and DADR3 are constantly conv
output at the analog output pins.

The D/A data registers are initialized to H'00 by a reset and in hardware standby mode.

17.2.2 D/A Control Register 23 (DACR23)

Bit	:	7	6	5	4	3	2	1
		DAOE1	DAOE0	DAE	—	—	—	—
Initial value	:	0	0	0	1	1	1	1
R/W	:	R/W	R/W	R/W	—	—	—	—

DACR23 is an 8-bit readable/writable register that controls the operation of the D/A co

DACR23 is initialized to H'1F by a reset and in hardware standby mode.

Bit 7—D/A Output Enable 1 (DAOE1): Controls D/A conversion and analog output.

Bit 7

DAOE1	Description
0	DA3 analog output is disabled (Ir
1	Channel 3 D/A conversion and DA3 analog output are enabled

Bit 5—D/A Enable (DAE): Controls D/A conversion, together with bits DAOE0 and
When the DAE bit is cleared to 0, D/A conversion is controlled independently in chan
When the DAE bit is set to 1, D/A conversion is controlled together in channels 2 and

Output of the conversion result is always controlled independently by bits DAOE0 an

Bit 7	Bit 6	Bit 5	
DAOE1	**DAOE0**	**DAE**	**Description**
0	0	*	D/A conversion is disabled in channels 2 and 3 (
	1	0	D/A conversion is enabled in channel 2
			D/A conversion is disabled in channel 3
		1	D/A conversion is enabled in channels 2 and 3
0	0	0	D/A conversion is disabled in channel 2
			D/A conversion is enabled in channel 3
		1	D/A conversion is enabled in channels 2 and 3
	1	*	D/A conversion is enabled in channels 2 and 3

If the chip enters software standby mode while D/A conversion is enabled, the D/A ou
retained and the analog power supply current is the same as the analog power supply c
during D/A conversion. If it is necessary to reduce the analog power supply current in
standby mode, D/A output should be disabled by clearing both the DAOE0 bit and the
to 0.

Bits 4 to 0—Reserved: These bits are always read as 1, and cannot be modified.

When the MSTPC5 bit is set to 1, D/A converter operation is stopped at the end of the l and module stop mode is entered. Register read/write accesses are not possible in modu mode. For details, see section 21B.5, Module Stop Mode.

MSTPCRC is initialized to H'FF by a reset and in hardware standby mode. It is not init software standby mode.

Bit 5—Module Stop (MSTPC5): Specifies module stop mode for the D/A converter (and 3).

Bit 5

MSTPC5	Description
0	D/A converter (channels 2 and 3) module stop mode is cleared
1	D/A converter (channels 2 and 3) module stop mode is set (In

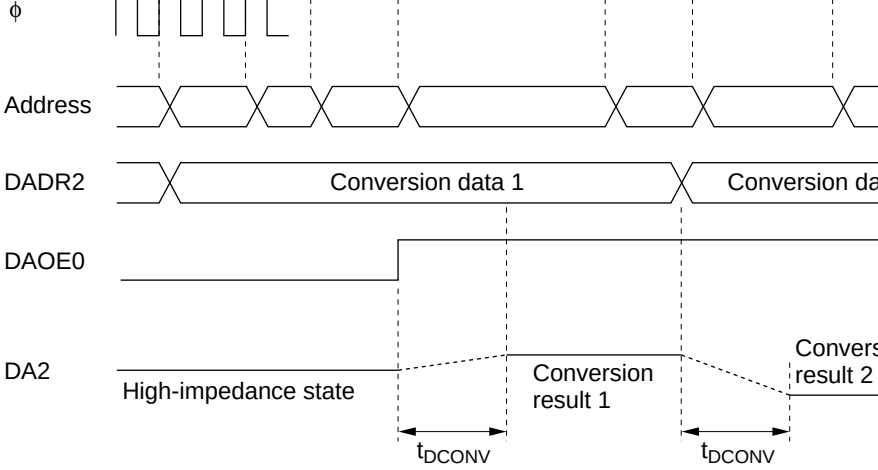
17.3 Operation

The D/A converter has two built-in D/A conversion circuits that can perform conversio independently.

D/A conversion is performed constantly while enabled in DACR23. If the DADR2 or D value is modified, conversion of the new data begins immediately. The conversion resu output when bits DAOE0 and DAOE1 are set to 1.

An example of D/A conversion on channel 2 is given below. The timing is shown in fig

1. Data to be converted is written in DADR2.
2. Bit DAOE0 is set to 1 in DACR23. D/A conversion starts and DA2 becomes an out The conversion result is output after the conversion time. The output value is (DAD contents/256) × Vref. Output of this conversion result continues until the value in D modified or the DAOE0 bit is cleared to 0.



Legend:
t_{DCONV}: D/A conversion time

Figure 17.2 Example of D/A Converter Operation

and word data. This makes it possible to perform fast word data transfer.

The on-chip RAM can be enabled or disabled by means of the RAM enable bit (RAM
system control register (SYSCR).

18.1.1 Block Diagram

Figure 18.1 shows a block diagram of the on-chip RAM.

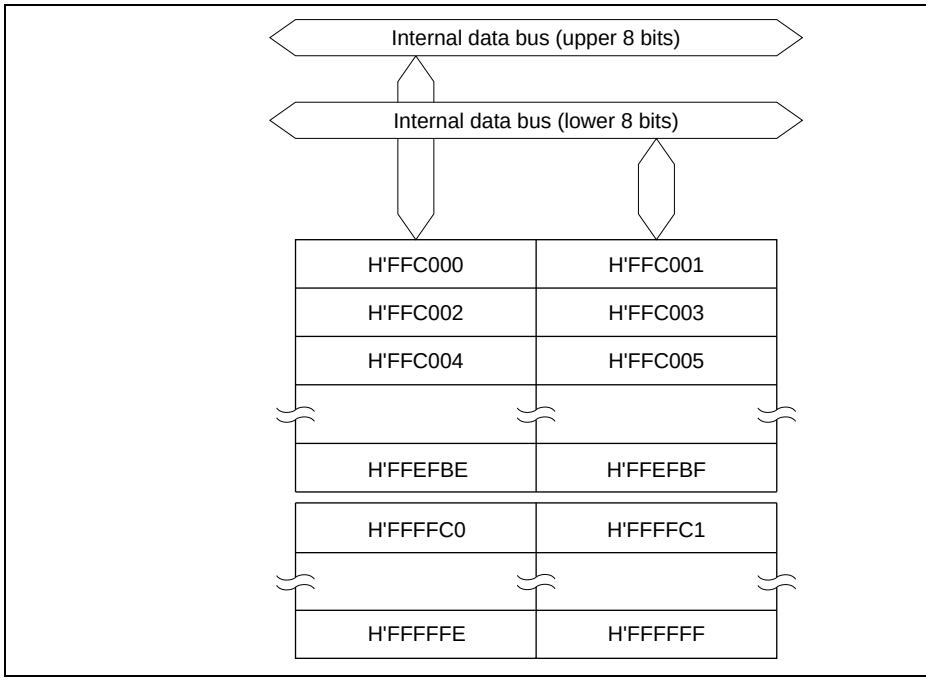


Figure 18.1 Block Diagram of RAM (H8S/2623)

Note: * Lower 16 bits of the address.

18.2 Register Descriptions

18.2.1 System Control Register (SYSCR)

Bit	:	7	6	5	4	3	2	1
		MACS	—	INTM1	INTM0	NMIEG	—	—
Initial value	:	0	0	0	0	0	0	0
R/W	:	R/W	—	R/W	R/W	R/W	R/W	—

The on-chip RAM is enabled or disabled by the RAME bit in SYSCR. For details of ot
SYSCR, see section 3.2.2, System Control Register (SYSCR).

Bit 0—RAM Enable (RAME): Enables or disables the on-chip RAM. The RAME bit
initialized when the reset state is released. It is not initialized in software standby mode

Note: When the DTC is used, the RAME bit must be set to 1.

Bit 0

RAME	Description
0	On-chip RAM is disabled
1	On-chip RAM is enabled (In

and read in byte or word units. Each type of access can be performed in one state.

Even addresses use the upper 8 bits, and odd addresses use the lower 8 bits. Word data at an even address.

18.4 Usage Notes

When Using the DTC: DTC register information can be located in addresses H'FFEE H'FFEFBF. When the DTC is used, the RAME bit must not be cleared to 0.

Reserved Areas: Addresses H'FFB000 to H'FFBFFF in the H8S/2626 and H8S/2623, to H'FFCFFF in the H8S/2625 and H8S/2622, and H'FFB000 to H'FFDFFF in the H8S H8S/2621, are reserved areas that cannot be read or written to. When the RAME bit is 0, external address space is accessed.

Four flash memory operating modes
  - Program mode
  - Erase mode
  - Program-verify mode
  - Erase-verify mode
- Programming/erase methods

  The flash memory is programmed 128 bytes at a time. Block erase (in single-block be performed. To erase the entire flash memory, each block must be erased in turn erasing can be performed as required on 4-kbyte, 32-kbyte, and 64-kbyte blocks.
- Programming/erase times

  The flash memory programming time is 10 ms (typ.) for simultaneous 128-byte pr equivalent to 78 µs (typ.) per byte, and the erase time is 100 ms (typ.).
- Reprogramming capability

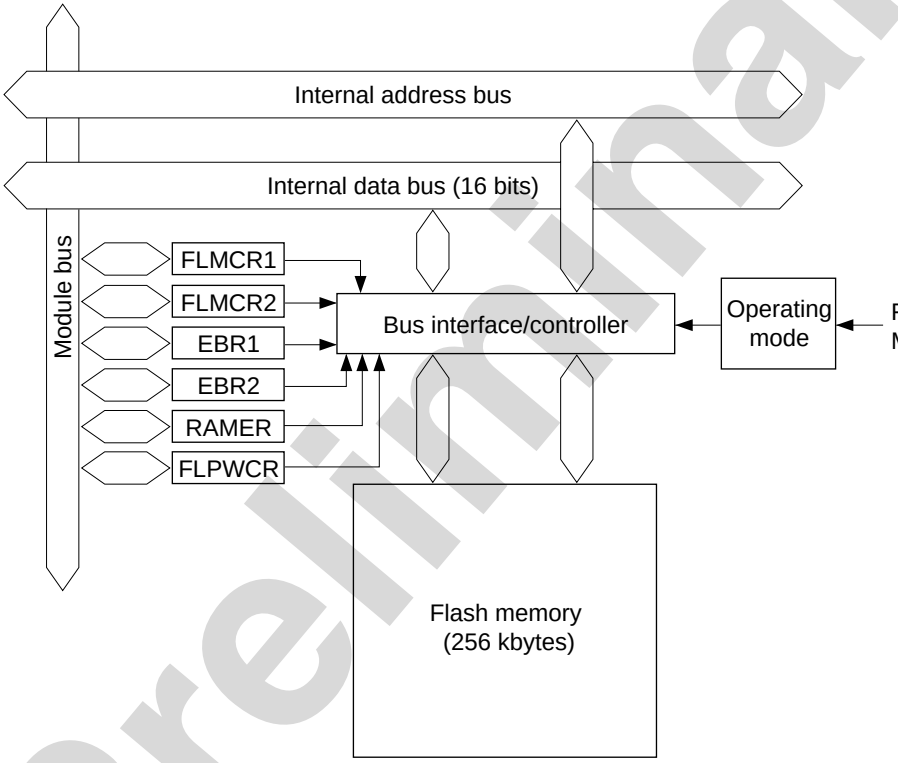
  The flash memory can be reprogrammed up to 100 times.
- On-board programming modes

  There are two modes in which flash memory can be programmed/erased/verified c
  - Boot mode
  - User program mode
- Automatic bit rate adjustment

  With data transfer in boot mode, the LSI's bit rate can be automatically adjusted to transfer bit rate of the host.
- Flash memory emulation in RAM

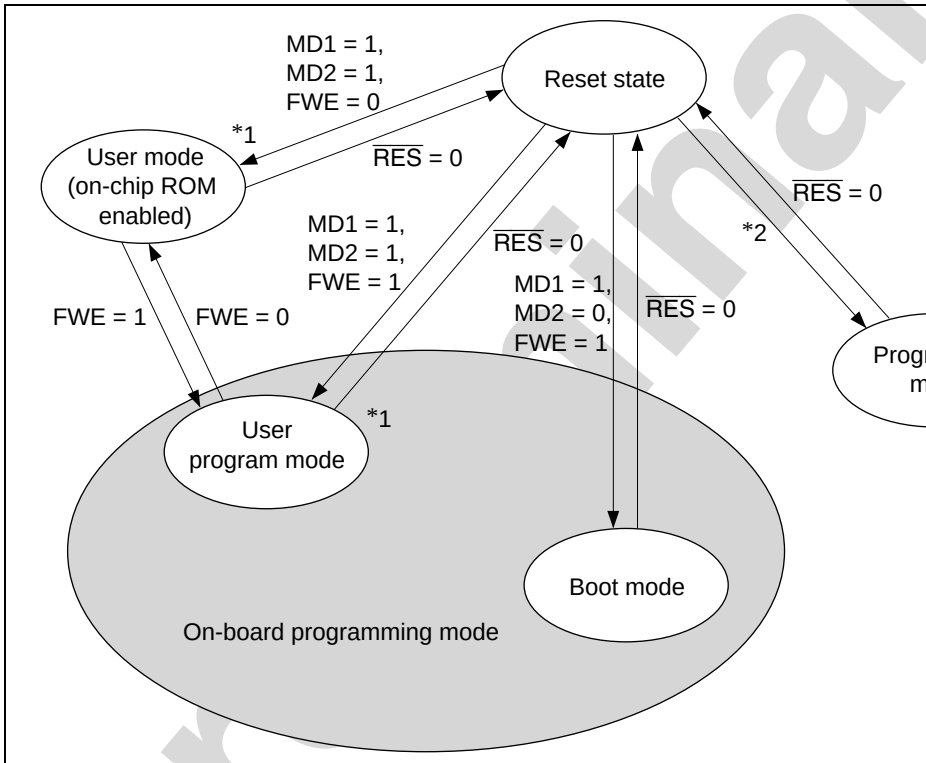
  Flash memory programming can be emulated in real time by overlapping a part of flash memory.
- Protect modes

  There are three protect modes, hardware, software, and error protection which allo status to be designated for flash memory program/erase/verify operations.



Legend:
FLMCR1: Flash memory control register 1
FLMCR2: Flash memory control register 2
EBR1: Erase block register 1
EBR2: Erase block register 2
RAMER: RAM emulation register
FLPWCR: Flash memory power control register

Figure 19.1 Block Diagram of Flash Memory

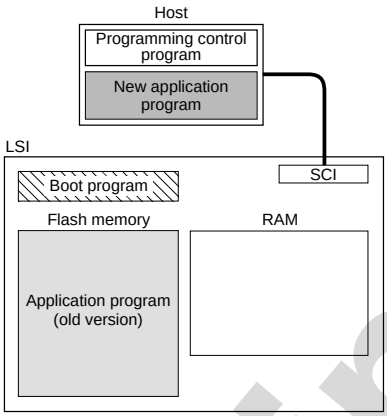


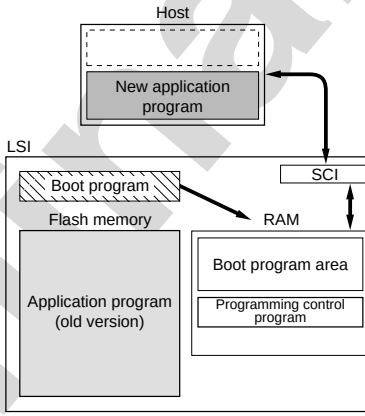
Notes: Only make a transition between user mode and user program mode when th
not accessing the flash memory.

1. RAM emulation possible
2. MD0 = 0, MD1 = 0, MD2 = 0, P14 = 0, P16 = 0, PF0 = 1

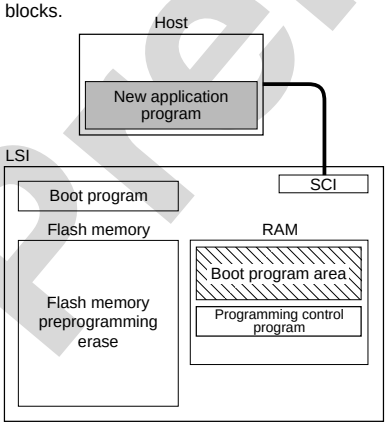
Figure 19.2 Flash Memory State Transitions

boot program required for flash memory era
is automatically transferred to the RAM boo
program area.

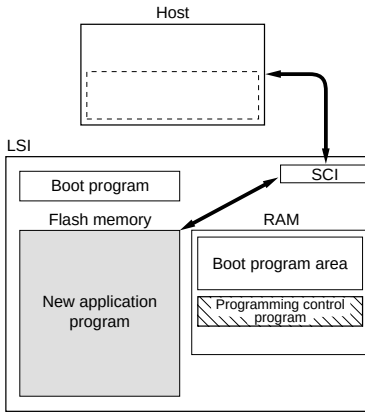




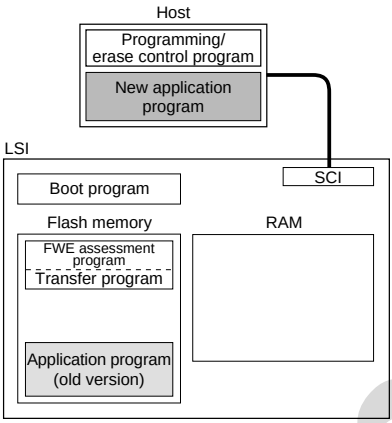
3. Flash memory initialization
   The erase program in the boot program area (in RAM) is executed, and the flash memory is initialized (to H'FF). In boot mode, total flash memory erasure is performed, without regard to blocks.



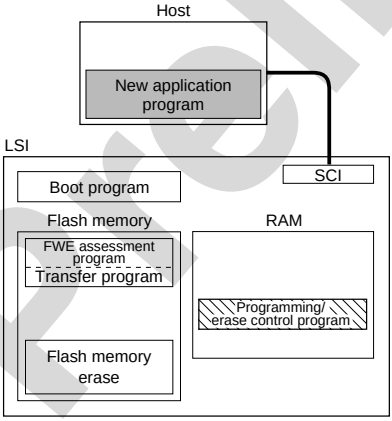
4. Writing new application program
   The programming control program transfer
   from the host to RAM is executed, and the
   application program in the host is written in
   flash memory.

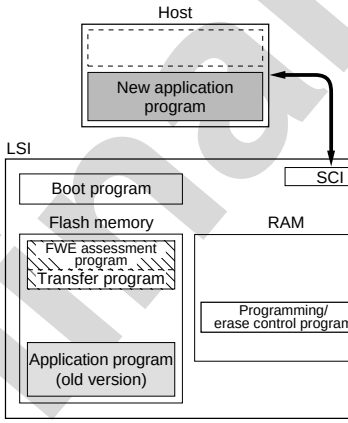


Program executio

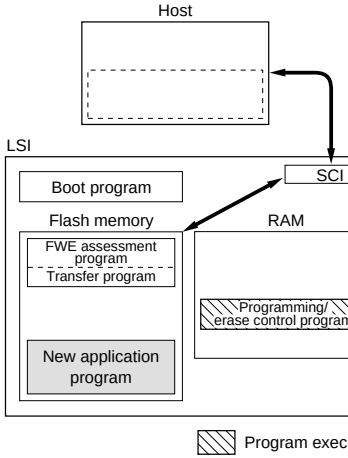


3. Flash memory initialization
The programming/erase program in RAM is executed, and the flash memory is initialized (to H'FF). Erasing can be performed in block units, but not in byte units.





4. Writing new application program
Next, the new application program in the
written into the erased flash memory blo
not write to unerased blocks.



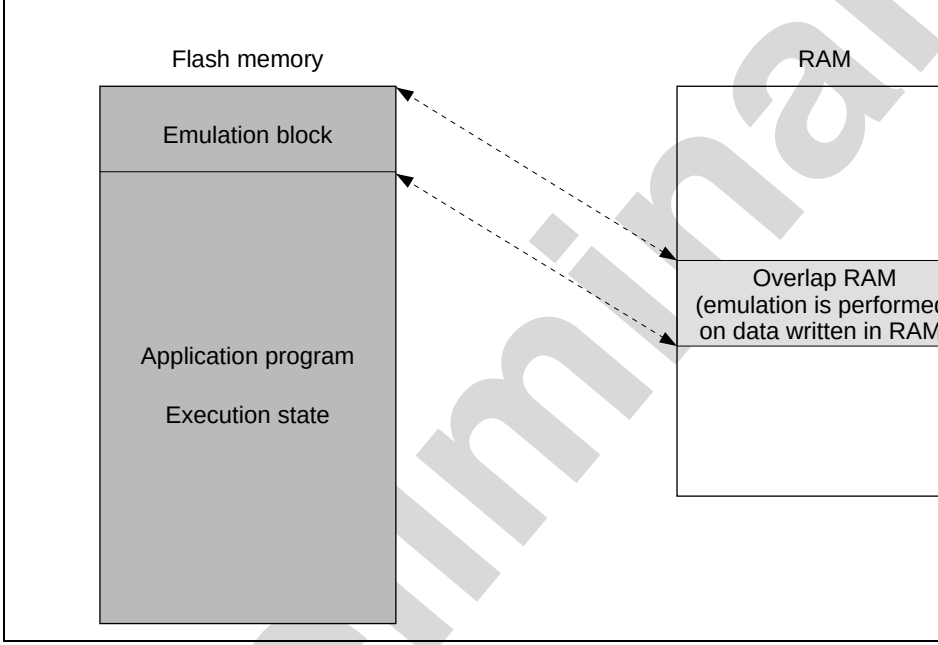


Figure 19.3 Reading Overlap RAM Data in User Mode or User Program M

When overlap RAM data is confirmed, the RAMS bit is cleared, RAM overlap is releas
writes should actually be performed to the flash memory.

When the programming control program is transferred to RAM, ensure that the transfer
and the overlap RAM do not overlap, as this will cause data in the overlap RAM to be

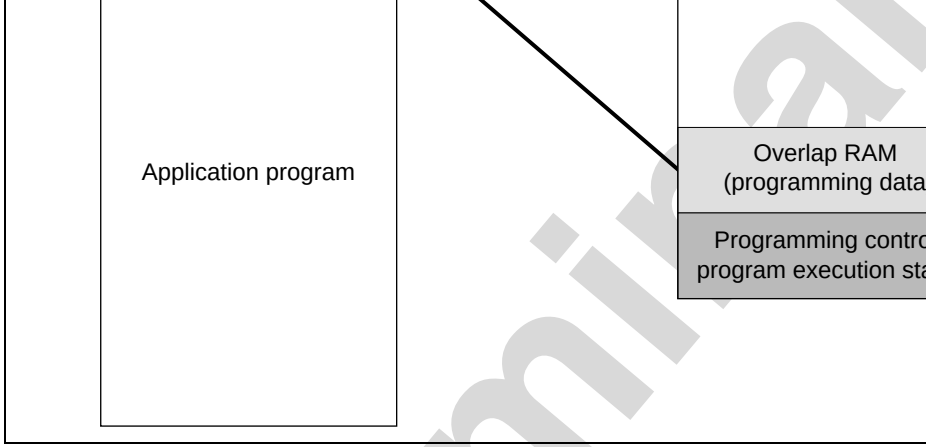


Figure 19.4 Writing Overlap RAM Data in User Program Mode

19.2.5 Differences between Boot Mode and User Program Mode

Table 19.1 Differences between Boot Mode and User Program Mode

	Boot Mode	User Program Mo
Total erase	Yes	Yes
Block erase	No	Yes
Programming control program*	Program/program-verify	Erase/erase-verify Program/program- Emulation

Note: * To be provided by the user, in accordance with the recommended algorithm

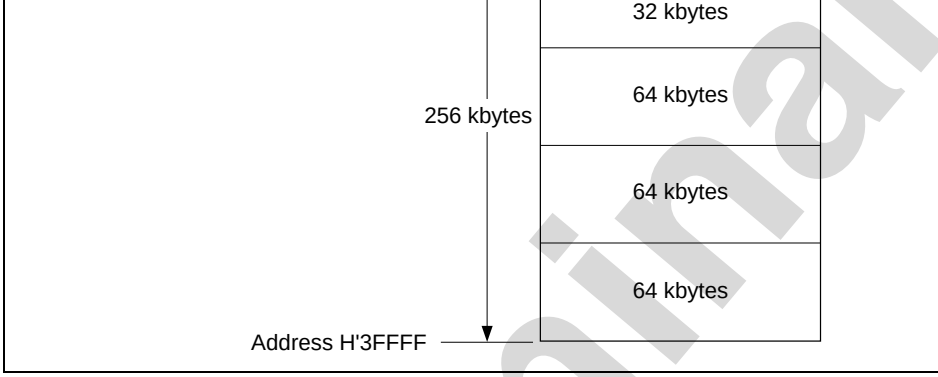


Figure 19.5 Flash Memory Block Configuration

19.3 Pin Configuration

The flash memory is controlled by means of the pins shown in table 19.2.

Table 19.2 Pin Configuration

Pin Name	Abbreviation	I/O	Function
Reset	$\overline{\text{RES}}$	Input	Reset
Flash write enable	FWE	Input	Flash memory program/erase protection b
Mode 2	MD2	Input	Sets MCU operating mode
Mode 1	MD1	Input	Sets MCU operating mode
Mode 0	MD0	Input	Sets MCU operating mode
Port F0	PF0	Input	Sets MCU operating mode in programmer
Port 16	P16	Input	Sets MCU operating mode in programmer
Port 14	P14	Input	Sets MCU operating mode in programmer
Transmit data	TxD2	Output	Serial transmit data output
Receive data	RxD2	Input	Serial receive data input

Flash memory control register 1	FLMCR1*5	R/W*2	H'00*3
Flash memory control register 2	FLMCR2*5	R*2	H'00
Erase block register 1	EBR1*5	R/W*2	H'00*4
Erase block register 2	EBR2*5	R/W*2	H'00*4
RAM emulation register	RAMER*5	R/W	H'00
Flash memory power control register*6	FLPWCR*5	R/W*2	H'00*4
Serial control register X	SCRX	R/W	H'00

Notes: 1. Lower 16 bits of the address.
2. To access these registers, set the FLSHE bit to 1 in serial control register X
FLSHE is set to 1, if the chip is in a mode in which the on-chip flash memor
disabled, a read will return H'00 and writes are invalid. Writes are also inva
FWE bit in FLMCR1 is not set to 1.
3. When a high level is input to the FWE pin, the initial value is H'80.
4. When a low level is input to the FWE pin, or if a high level is input and the S
FLMCR1 is not set, these registers are initialized to H'00.
5. FLMCR1, FLMCR2, EBR1, EBR2, RAMER, and FLPWCR are 8-bit registe
Use byte access on these registers.
6. An invalid register in the H8S/2623.

19.5 Register Descriptions

19.5.1 Flash Memory Control Register 1 (FLMCR1)

FLMCR1 is an 8-bit register used for flash memory operating mode control. Program-
or erase-verify mode for addresses H'00000 to H'3FFFF is entered by setting SWE1 bi
FWE = 1, then setting the PV1 or EV1 bit. Program mode for addresses H'00000 to H
entered by setting SWE1 bit to 1 when FWE = 1, then setting the PSU1 bit, and finally
P1 bit. Erase mode for addresses H'00000 to H'3FFFF is entered by setting SWE1 bit
FWE = 1, then setting the ESU1 bit, and finally setting the E1 bit. FLMCR1 is initializ
reset, and in hardware standby mode and software standby mode. Its initial value is H'

Initial value:	—*	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Determined by the state of the FWE pin.

Bit 7—Flash Write Enable Bit (FWE): Sets hardware protection against flash memor programming/erasing.

Bit 7

FWE	Description
0	When a low level is input to the FWE pin (hardware-protected state)
1	When a high level is input to the FWE pin

Bit 6—Software Write Enable Bit 1 (SWE1): This bit selects write and erase valid/in flash memory. Set it when setting bits 5 to 0, bits 7 to 0 of EBR1, and bits 3 to 0 of EB

Bit 6

SWE1	Description
0	Writes disabled (In
1	Writes enabled [Setting condition] When FWE = 1

	[Setting condition] When FWE = 1 and SWE1 = 1

Bit 4—Program Setup Bit 1 (PSU1): Prepares for a transition to program mode. Set
before setting the P1 bit in FLMCR1 to 1. Do not set the SWE1, ESU1, EV1, PV1, E
at the same time.

Bit 4

PSU1	Description
0	Program setup cleared
1	Program setup [Setting condition] When FWE = 1 and SWE1 = 1

Bit 3—Erase-Verify 1 (EV1): Selects erase-verify mode transition or clearing. Do no
SWE1, ESU1, PSU1, PV1, E1, or P1 bit at the same time.

Bit 3

EV1	Description
0	Erase-verify mode cleared
1	Transition to erase-verify mode [Setting condition] When FWE = 1 and SWE1 = 1

	When FWE = 1 and SWE1 = 1

Bit 1—Erase 1 (E1): Selects erase mode transition or clearing. Do not set the SWE1, E PSU1, EV1, PV1, or P1 bit at the same time.

Bit 1

E1	Description
0	Erase mode cleared
1	Transition to erase mode [Setting condition] When FWE = 1, SWE1 = 1, and ESU1 = 1

Bit 0—Program 1 (P1): Selects program mode transition or clearing. Do not set the SW PSU1, ESU1, EV1, PV1, or E1 bit at the same time.

Bit 0

P1	Description
0	Program mode cleared
1	Transition to program mode [Setting condition] When FWE = 1, SWE1 = 1, and PSU1 = 1

Initial value:	0	0	0	0	0	0	0
R/W:	R	—	—	—	—	—	—

Note: FLMCR2 is a read-only register, and should not be written to.

Bit 7—Flash Memory Error (FLER): Indicates that an error has occurred during an flash memory (programming or erasing). When FLER is set to 1, flash memory goes t protection state.

Bit 7

FLER	Description
0	Flash memory is operating normally (
	Flash memory program/erase protection (error protection) is disabled
	[Clearing condition]
	Reset or hardware standby mode
1	An error has occurred during flash memory programming/erasing
	Flash memory program/erase protection (error protection) is enabled
	[Setting condition]
	See section 19.8.3, Error Protection

Bits 6 to 0—Reserved: These bits always read 0.

cleared to 0. When on-chip flash memory is disabled, a read will return H'00, and write
invalid.

The flash memory block configuration is shown in table 19.4.

Bit:	7	6	5	4	3	2	1
	EB7	EB6	EB5	EB4	EB3	EB2	EB1
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

19.5.4 Erase Block Register 2 (EBR2)

EBR2 is an 8-bit register that specifies the flash memory erase area block by block. EB
initialized to H'00 by a reset, in hardware standby mode and software standby mode, w
level is input to the FWE pin. Bit 0 will be initialized to 0 if bit SWE1 of FLMCR1 is n
though a high level is input to pin FWE. When a bit in EBR2 is set to 1, the correspond
can be erased. Other blocks are erase-protected. Only one of the bits of EBR1 and EBR
combined can be set. Do not set more than one bit, as this will cause all the bits in both
EBR2 to be automatically cleared to 0. Bits 7 to 4 are reserved and must only be writte
When on-chip flash memory is disabled, a read will return H'00, and writes are invalid.

The flash memory block configuration is shown in table 19.4.

Bit:	7	6	5	4	3	2	1
	—	—	—	—	EB11	EB10	EB9
Initial value:	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

EB5 (4 kbytes)	H'005000–H'005FFF
EB6 (4 kbytes)	H'006000–H'006FFF
EB7 (4 kbytes)	H'007000–H'007FFF
EB8 (32 kbytes)	H'008000–H'00FFFF
EB9 (64 kbytes)	H'010000–H'01FFFF
EB10 (64 kbytes)	H'020000–H'02FFFF
EB11 (64 kbytes)	H'030000–H'03FFFF

19.5.5 RAM Emulation Register (RAMER)

RAMER specifies the area of flash memory to be overlapped with part of RAM when real-time flash memory programming. RAMER initialized to H'00 by a reset and in ha standby mode. It is not initialized in software standby mode. RAMER settings should user mode or user program mode.

Flash memory area divisions are shown in table 19.5. To ensure correct operation of t function, the ROM for which RAM emulation is performed should not be accessed im after this register has been modified. Normal execution of an access immediately after modification is not guaranteed.

Bit:	7	6	5	4	3	2	1
	—	—	—	—	RAMS	RAM2	RAM
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W

Bits 7 and 6—Reserved: These bits always read 0.

Bits 5 and 4—Reserved: Only 0 may be written to these bits.

Bits 2 to 0—Flash Memory Area Selection: These bits are used together with bit 3 to flash memory area to be overlapped with RAM. (See table 19.5.)

Table 19.5 Flash Memory Area Divisions

Addresses	Block Name	RAMS	RAM2	RAM1
H'FFD000–H'FFDFFF	RAM area 4 kbytes	0	*	*
H'000000–H'000FFF	EB0 (4 kbytes)	1	0	0
H'001000–H'001FFF	EB1 (4 kbytes)	1	0	0
H'002000–H'002FFF	EB2 (4 kbytes)	1	0	1
H'003000–H'003FFF	EB3 (4 kbytes)	1	0	1
H'004000–H'004FFF	EB4 (4 kbytes)	1	1	0
H'005000–H'005FFF	EB5 (4 kbytes)	1	1	0
H'006000–H'006FFF	EB6 (4 kbytes)	1	1	1
H'007000–H'007FFF	EB7 (4 kbytes)	1	1	1

*

FLPWCR enables or disables a transition to the flash memory power-down mode whe
switches to subactive mode.

Note: * An invalid register in the H8S/2623.

Bit 7—Power-Down Disable (PDWND): Enables or disables a transition to the flash
power-down mode when the LSI switches to subactive mode.

Bit 7	
PDWND	**Description**
0	Transition to flash memory power-down mode enabled (
1	Transition to flash memory power-down mode disabled

Bits 6 to 0—Reserved: These bits always read 0.

19.5.7 Serial Control Register X (SCRX)

Bit	7	6	5	4	3	2	1
	—	—	—	—	FLSHE	—	—
Initial value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SCRX is an 8-bit readable/writable register that controls on-chip flash memory.

SCRX is initialized to H'00 by a reset and in hardware standby mode.

Bits 7 to 4—Reserved: Only 0 may be written to these bits.

0	Flash control registers deselected in area H'FFFFA8 to H'FFFFAC	(In
1	Flash control registers selected in area H'FFFFA8 to H'FFFFAC	

Bits 2 to 0—Reserved: Only 0 may be written to these bits.

19.6 On-Board Programming Modes

When pins are set to on-board programming mode and a reset-start is executed, a transi made to the on-board programming state in which program/erase/verify operations can performed on the on-chip flash memory. There are two on-board programming modes: and user program mode. The pin settings for transition to each of these modes are show 19.6. For a diagram of the transitions to the various flash memory modes, see figure 19

Table 19.6 Setting On-Board Programming Modes

Mode		FWE	MD2	MD1
Boot mode	Expanded mode	1	0	1
	Single-chip mode		0	1
User program mode	Expanded mode	1	1	1
	Single-chip mode		1	1

programming control program received via the SCI is written into the programming co
program area in on-chip RAM. After the transfer is completed, control branches to the
address of the programming control program area and the programming control progra
state is entered (flash memory programming is performed).

The transferred programming control program must therefore include coding that follo
programming algorithm given later.

The system configuration in boot mode is shown in figure 19.6, and the boot mode exe
procedure in figure 19.7.

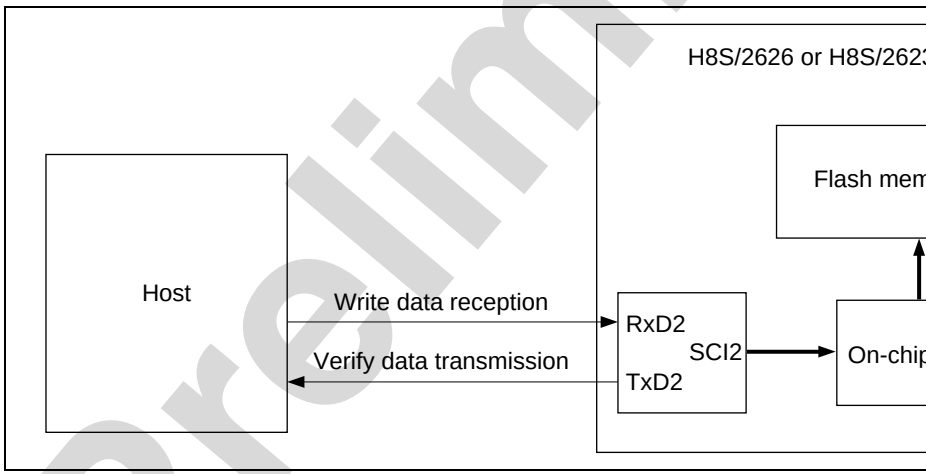
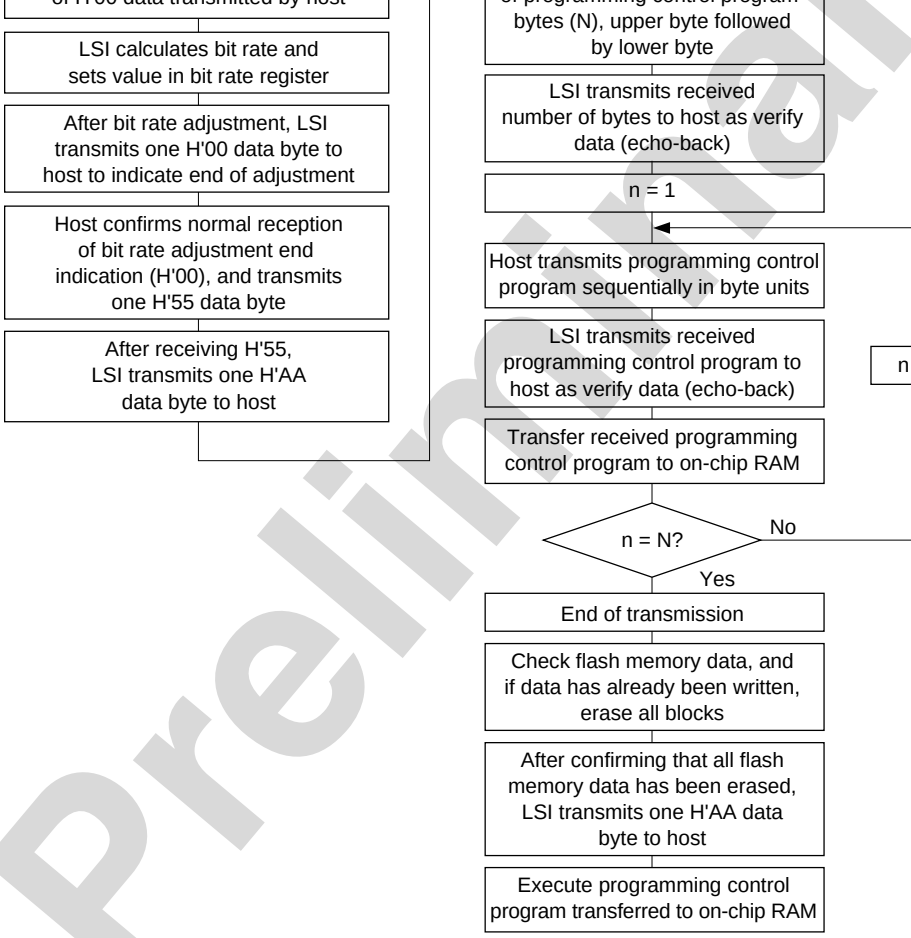


Figure 19.6 System Configuration in Boot Mode



Note: If a memory cell does not operate normally and cannot be erased, one H'FF byte is transmitted as an erase error, and the erase operation and subsequent operation are halted.

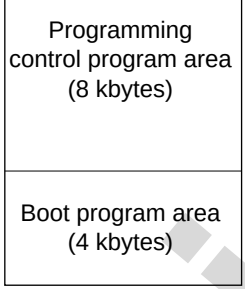
Figure 19.7 Boot Mode Execution Procedure

When boot mode is initiated, the LSI measures the low period of the asynchronous SC
communication data (H'00) transmitted continuously from the host. The SCI transmit/
format should be set as follows: 8-bit data, 1 stop bit, no parity. The LSI calculates the
the transmission from the host from the measured low period, and transmits one H'00
host to indicate the end of bit rate adjustment. The host should confirm that this adjust
indication (H'00) has been received normally, and transmit one H'55 byte to the LSI. I
cannot be performed normally, initiate boot mode again (reset), and repeat the above o
Depending on the host's transmission bit rate and the LSI's system clock frequency, th
a discrepancy between the bit rates of the host and the LSI. Set the host transfer bit rat
4,800, 9,600 or 19,200 bps to operate the SCI properly.

Table 19.7 shows host transfer bit rates and system clock frequencies for which autom
adjustment of the LSI bit rate is possible. The boot program should be executed within
clock range.

**Table 19.7 System Clock Frequencies for which Automatic Adjustment of LSI I
Possible**

Host Bit Rate	System Clock Frequency for Which Automatic Adjustment of LSI Bit Rate is Possible
2,400 bps	2 to 8 MHz
4,800 bps	4 to 16 MHz
9,600 bps	8 to 20 MHz
19,200 bps	16 to 20 MHz



Note: The boot program area cannot be used until a transition is made to the execution the programming control program transferred to RAM. Note also that the boot pr remains in this area of the on-chip RAM even after control branches to the progra control program.

Figure 19.8 RAM Areas in Boot Mode

Notes on Use of Boot Mode:

- When the chip comes out of reset in boot mode, it measures the low-level period of the SCI's RxD2 pin. The reset should end with RxD2 high. After the reset ends, it t approximately 100 states before the chip is ready to measure the low-level period o pin.
- In boot mode, if any data has been programmed into the flash memory (if all data i flash memory blocks are erased. Boot mode is for use when user program mode is u such as the first time on-board programming is performed, or if the program activat program mode is accidentally erased.
- Interrupts cannot be used while the flash memory is being programmed or erased.
- The RxD2 and TxD2 pins should be pulled up on the board.

The initial values of other on-chip registers are not changed.

- Boot mode can be entered by making the pin settings shown in table 19.6 and exec reset-start.

  Boot mode can be cleared by driving the reset pin low, waiting at least 20 states, th the FWE pin and mode pins, and executing reset release[*1]. Boot mode can also be WDT overflow reset.

  Do not change the mode pin input levels in boot mode, and do not drive the FWE p while the boot program is being executed or while flash memory is being program erased[*2].

- If the mode pin input levels are changed (for example, from low to high) during a state of ports with multiplexed address functions and bus control output pins ($\overline{AS}$, will change according to the change in the microcomputer's operating mode[*3].

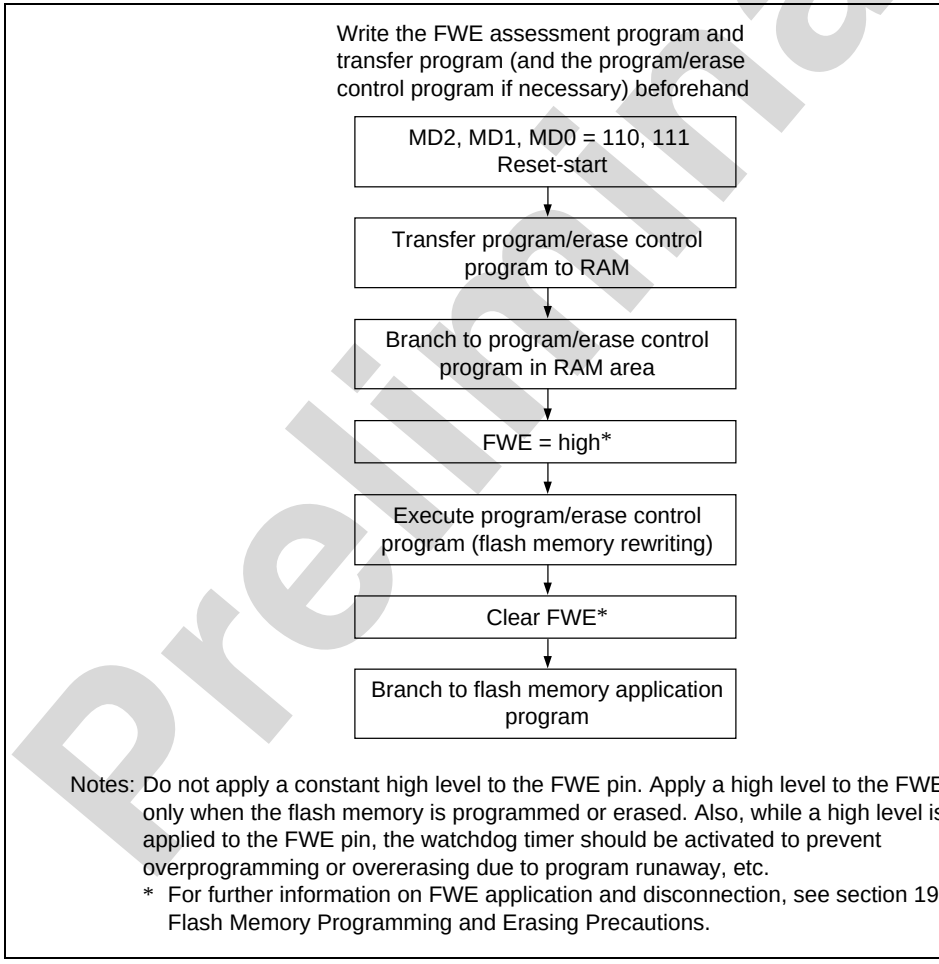
  Therefore, care must be taken to make pin settings to prevent these pins from beco signal pins during a reset, or to prevent collision with signals outside the microcon

Notes: 1. Mode pin and FWE pin input must satisfy the mode programming setup tim states) with respect to the reset release timing.
2. For further information on FWE application and disconnection, see section Flash Memory Programming and Erasing Precautions.
3. See appendix D, Pin States.

19.6.2 User Program Mode

When set to user program mode, the chip can program and erase its flash memory by e user program/erase control program. Therefore, on-board reprogramming of the on-ch memory can be carried out by providing on-board means of FWE control and supply o programming data, and storing a program/erase control program in part of the program necessary.

Figure 19.9 shows the procedure for executing the program/erase control program when transferred to on-chip RAM.



Notes: Do not apply a constant high level to the FWE pin. Apply a high level to the FWE only when the flash memory is programmed or erased. Also, while a high level is applied to the FWE pin, the watchdog timer should be activated to prevent overprogramming or overerasing due to program runaway, etc.

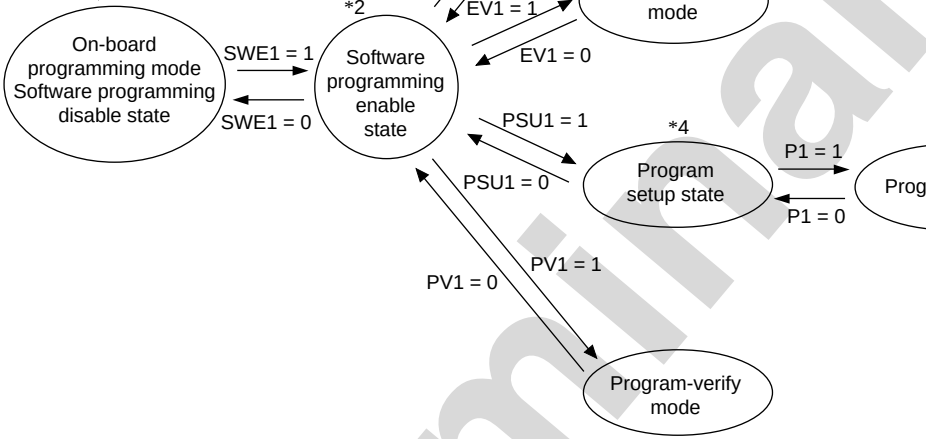
* For further information on FWE application and disconnection, see section 19 Flash Memory Programming and Erasing Precautions.

Figure 19.9 User Program Mode Execution Procedure

(user program) that controls flash memory programming/erasing should be located and on-chip RAM or external memory. If the program is to be located in external memory instruction for writing to flash memory, and the following instruction, should be place RAM. Also ensure that the DTC is not activated before or after execution of the flash write instruction.

In the following operation descriptions, wait times after setting or clearing individual FLMCR1 are given as parameters; for details of the wait times, see section 22.6, Flash Characteristics.

Notes: 1. Operation is not guaranteed if setting/resetting of the SWE1, ESU1, PSU1, E1, and P1 bits in FLMCR1 is executed by a program in flash memory.
2. When programming or erasing, set FWE to 1 (programming/erasing will n executed if FWE = 0).
3. Programming must be executed in the erased state. Do not perform additio programming on addresses that have already been programmed.



Notes: In order to perform a normal read of flash memory, SWE1 must be cleared to 0. Also note that v
can be performed during the programming/erasing process.

1. : Normal mode : On-board programming mode
2. Do not make a state transition by setting or clearing multiple bits simultaneously.
3. After a transition from erase mode to the erase setup state, do not enter erase mode without
through the software programming enable state.
4. After a transition from program mode to the program setup state, do not enter program mode
passing through the software programming enable state.

Figure 19.10 FLMCR1 Bit Settings and State Transitions

the maximum number of programming operations (N) are shown in table 22.10.

Following the elapse of (×0) µs or more after the SWE1 bit is set to 1 in FLMCR1, 12
program data is stored in the program data area and reprogram data area, and the 128-
the program data area in RAM is written consecutively to the program address (the lo
the first address written to must be H'00 or H'80). 128 consecutive byte data transfers
performed. The program address and program data are latched in the flash memory. A
data transfer must be performed even if writing fewer than 128 bytes; in this case, H'F
be written to the extra addresses.

Next, the watchdog timer is set to prevent overprogramming in the event of program r
Set 6.6 ms as the WDT overflow period. After this, preparation for program mode (pr
is carried out by setting the PSU1 bit in FLMCR1, and after the elapse of (y) µs or mo
operating mode is switched to program mode by setting the P1 bit in FLMCR1. The ti
which the P1 bit is set is the flash memory programming time. Refer to the table in fig
for the programming time.

should be made to the addresses to be read. The dummy write should be executed after
of (γ) µs or more. When the flash memory is read in this state (verify data is read in 16-
the data at the latched address is read. Wait at least (ε) µs after the dummy write before
this read operation. Next, the originally written data is compared with the verify data, a
reprogram data is computed (see figure 19.11) and transferred to RAM. After verificati
bytes of data has been completed, exit program-verify mode, wait for at least (η) µs, th
SWE1 bit in FLMCR1. If reprogramming is necessary, set program mode again, and re
program/program-verify sequence as before. The maximum number of repetitions of th
program/program-verify sequence is indicated by the maximum programming count (N
However, ensure that the program/program-verify sequence is not repeated more than (
the same bits.

Notes on Program/Program-Verify Procedure

1. In order to perform 128-byte-unit programming, the lower 8 bits of the write start a
   be H'00 or H'80.

2. When performing continuous writing of 128-byte data to flash memory, byte-unit tr
   should be used.

   128-byte data transfer is necessary even when writing fewer than 128 bytes of data.
   H'FF data to the extra addresses.

3. Verify data is read in word units.

4. The write pulse is applied and a flash memory write executed while the P1 bit in FL
   set. In the H8S/2626 and H8S/2623, write pulses should be applied as follows in the
   program/program-verify procedure to prevent voltage stress on the device and loss
   data reliability.
   a. After write pulse application, perform a verify-read in program-verify mode and
      write pulse again for any bits read as 1 (reprogramming processing). When all th
      bits in the 128-byte write data are read as 0 in the verify-read operation, the
      program/program-verify procedure is completed. In the H8S/2626 and H8S/262

5. The period for which the PT bit in FLMCR1 is set (the write pulse width) should b according to the degree of progress through the program/program-verify procedure detailed wait time specifications, see section 22.6, Flash Memory Characteristics.

6. The program/program-verify flowchart for the H8S/2626 and H8S/2623 is shown 19.11.

   To cover the points noted above, bits on which reprogramming processing is to be and bits on which additional programming is to be executed, must be determined a below.

   Since reprogram data and additional-programming data vary according to the prog programming procedure, it is recommended that the following data storage areas (each) be provided in RAM.

   Reprogram Data Computation Table

(D)	Result of Verify-Read after Write Pulse Application (V)	(X) Result of Operation	Comments
0	0	1	Programming completed: repr processing not to be executed
	1	0	Programming incomplete: rep processing to be executed
1	0	1	—
	1		Still in erased state: no action

Legend:

(D): Source data of bits on which programming is executed

(X): Source data of bits on which reprogramming is executed

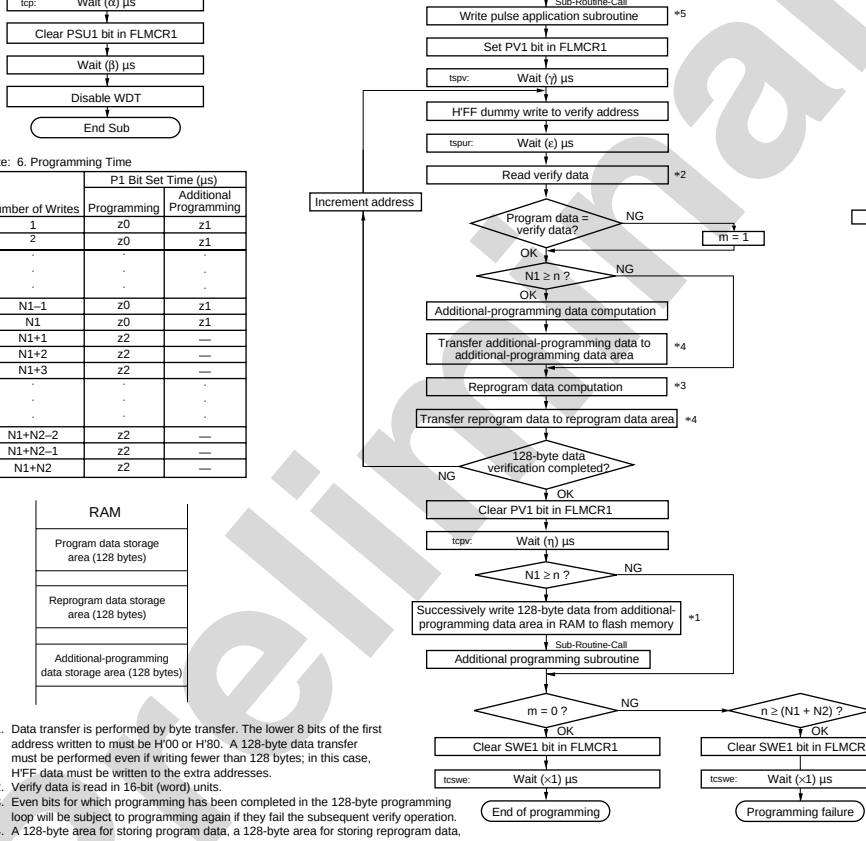
		incomplete: additional programmi processing not to be executed
1	0	Programming already completed: programming processing not to b
	1	Still in erased state: no action

Legend:

(Y): Data of bits on which additional programming is executed

(X'): Data of bits on which reprogramming is executed in a certain reprogramming loop

7. It is necessary to execute additional programming processing during the course of t H8S/2626 or H8S/2623 program/program-verify procedure. However, once 128-by programming is finished, additional programming should not be carried out on the s address area. When executing reprogramming, an erase must be executed first. Note normal operation of reads, etc., is not guaranteed if additional programming is perfo addresses for which a program/program-verify operation has finished.



Note: 6. Programming Time

Number of Writes	P1 Bit Set Time (µs)	
	Programming	Additional Programming
1	z0	z1
2	z0	z1
. . .	. . .	. . .
N1–1	z0	z1
N1	z0	z1
N1+1	z2	—
N1+2	z2	—
N1+3	z2	—
. . .	. . .	. . .
N1+N2–2	z2	—
N1+N2–1	z2	—
N1+N2	z2	—

Notes: 1. Data transfer is performed by byte transfer. The lower 8 bits of the first address written to must be H'00 or H'80. A 128-byte data transfer must be performed even if writing fewer than 128 bytes; in this case, H'FF data must be written to the extra addresses.
2. Verify data is read in 16-bit (word) units.
3. Even bits for which programming has been completed in the 128-byte programming loop will be subject to programming again if they fail the subsequent verify operation.
4. A 128-byte area for storing program data, a 128-byte area for storing reprogram data, and a 128-byte area for storing additional-programming data must be provided in RAM. The reprogram and additional-programming data contents are modified as programming proceeds.
5. A write pulse of 30 µs or 200 µs is applied according to the progress of the programming operation. See note *6 for details of the pulse widths. When writing of additional-programming data is executed, a 10 µs write pulse should be applied. Reprogram data X' means reprogram data when the write pulse i

Reprogram Data Computation Table

Original Data (D)	Verify Data (V)	Reprogram Data (X)	Comments
0	0	1	Programming complete
0	1	0	Programming is incomplete: reprogramming should be performed
1	0	1	
1	1	1	Left in the erased state

Additional-Programming Data Computation Table

Reprogram Data (X')	Verify Data (V)	Additional-Programming Data (X)	Comment
0	0	0	Additional programming sh
0	1	1	Additional programming sh
1	0	1	Additional programming sh
1	1	1	Additional programming sh

Figure 19.11 Program/Program-Verify Flowchart

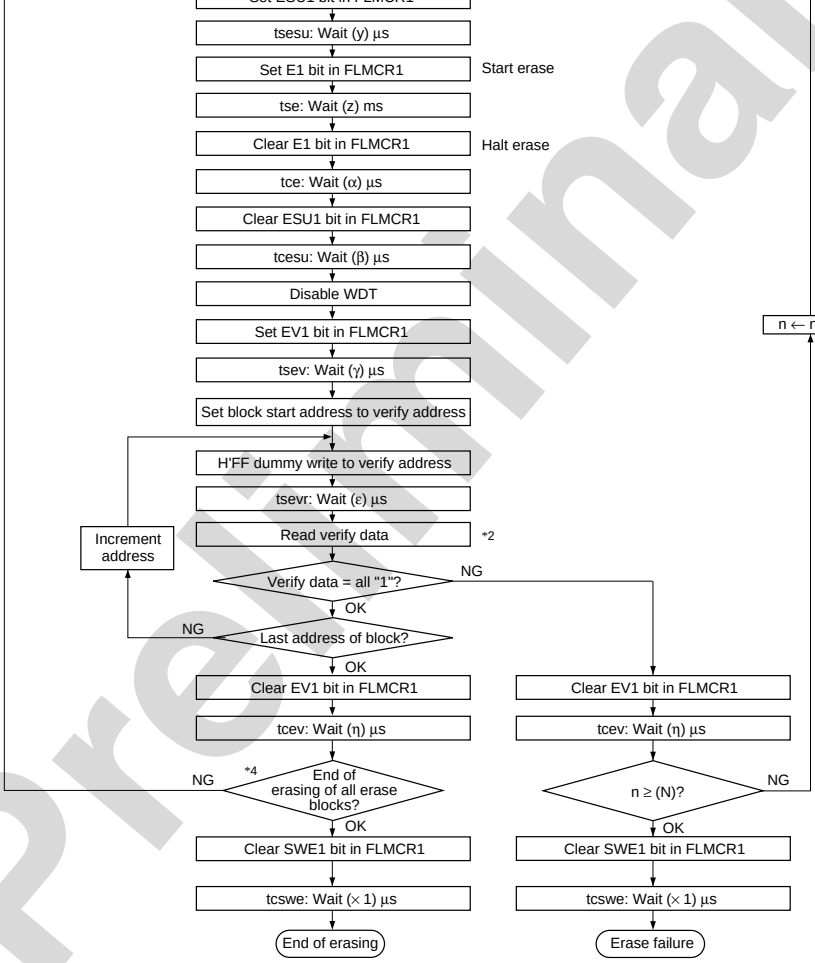
setup) is performed next by setting the ESU1 bit in FLMCR1. The operating mode is t
switched to erase mode by setting the E1 bit in FLMCR1 after the elapse of at least (y)
time during which the E1 bit is set is the flash memory erase time. Ensure that the erase
not exceed (z) ms.

Note: With flash memory erasing, preprogramming (setting all memory data in the m
be erased to all 0) is not necessary before starting the erase procedure.

19.7.4 Erase-Verify Mode

In erase-verify mode, data is read after memory has been erased to check whether it has
correctly erased.

After the elapse of the fixed erase time, clear the E1 bit in FLMCR1, then wait for at le
before clearing the ESU1 bit to exit erase mode. After exiting erase mode, the watchdo
cleared after the elapse of (β) µs or more. The operating mode is then switched to erase
mode by setting the EV1 bit in FLMCR1. Before reading in erase-verify mode, a dumm
H'FF data should be made to the addresses to be read. The dummy write should be exec
the elapse of (γ) µs or more. When the flash memory is read in this state (verify data is
bit units), the data at the latched address is read. Wait at least (ε) µs after the dummy w
performing this read operation. If the read data has been erased (all 1), a dummy write i
performed to the next address, and erase-verify is performed. If the read data is unerase
mode again and repeat the erase/erase-verify sequence in the same way. The maximum
reoperations of the erase/erase-verify sequence is indicated by the maximum erase coun
However, ensure that the erase/erase-verify sequence is not repeated more than (N) tim
verification is completed, exit erase-verify mode, and wait for at least (η) µs. If erasure
completed on all the erase blocks, clear the SWE1 bit in FLMCR1. If there are any une
blocks, make a 1 bit setting for the flash memory area to be erased, and repeat the erase
verify sequence as before.



Notes: 1. Preprogramming (setting erase block data to all "0") is not necessary.
2. Verify data is read in 16-bit (W) units.
3. Set only one bit in EBR1 and 2. More than 2 bits cannot be set.
4. Erasing is performed in block units. To erase a number of blocks, each block must be erased in turn.

Figure 19.12 Erase/Erase-Verify Flowchart

disabled or aborted. Hardware protection is reset by settings in flash memory control re
(FLMCR1), flash memory control register 2 (FLMCR2), erase block register 1 (EBR1)
block register 2 (EBR2). The FLMCR1, FLMCR2, EBR1, and EBR2 settings are retain
error-protected state. (See table 19.8.)

Table 19.8 Hardware Protection

Item	Description	Functio	
		Program	Er
FWE pin protection	• When a low level is input to the FWE pin, FLMCR1, FLMCR2, (except bit FLER) EBR1, and EBR2 are initialized, and the program/erase-protected state is entered.	Yes	Ye
Reset/standby protection	• In a reset (including a WDT reset) and in standby mode, FLMCR1, FLMCR2, EBR1, and EBR2 are initialized, and the program/erase-protected state is entered. • In a reset via the $\overline{\text{RES}}$ pin, the reset state is not entered unless the $\overline{\text{RES}}$ pin is held low until oscillation stabilizes after powering on. In the case of a reset during operation, hold the $\overline{\text{RES}}$ pin low for the $\overline{\text{RES}}$ pulse width specified in the AC Characteristics section.	Yes	Ye

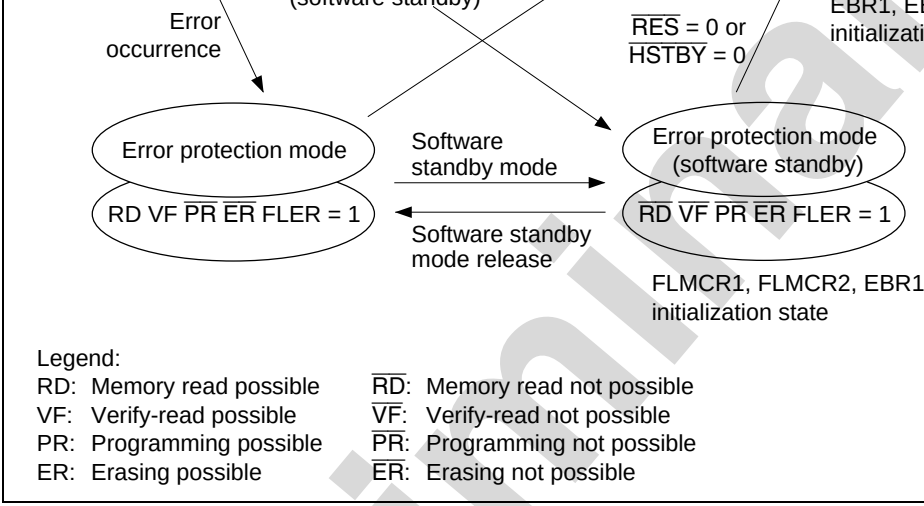
Item	Description	Program	E
SWE bit protection	• Setting bit SWE1 in FLMCR1 to 0 will place area H'000000 to H'03FFFF in the program/erase-protected state. (Execute the program in the on-chip RAM, external memory)	Yes	Y
Block specification protection	• Erase protection can be set for individual blocks by settings in erase block register 1 (EBR1) and erase block register 2 (EBR2). • Setting EBR1 and EBR2 to H'00 places all blocks in the erase-protected state.	—	Y
Emulation protection	• Setting the RAMS bit to 1 in the RAM emulation register (RAMER) places all blocks in the program/erase-protected state.	Yes	Y

EBR1, and EBR2 settings are retained, but program mode or erase mode is aborted at t
which the error occurred. Program mode or erase mode cannot be re-entered by re-setti
or E1 bit. However, PV1 and EV1 bit setting is enabled, and a transition can be made t
mode.

FLER bit setting conditions are as follows:

1. When the flash memory of the relevant address area is read during programming/er
   (including vector read and instruction fetch)
2. When a SLEEP instruction (including software standby) is executed during
   programming/erasing

Error protection is released only by a reset and in hardware standby mode.



Legend:

RD:	Memory read possible	$\overline{RD}$:	Memory read not possible
VF:	Verify-read possible	$\overline{VF}$:	Verify-read not possible
PR:	Programming possible	$\overline{PR}$:	Programming not possible
ER:	Erasing possible	$\overline{ER}$:	Erasing not possible

Figure 19.13 Flash Memory State Transitions

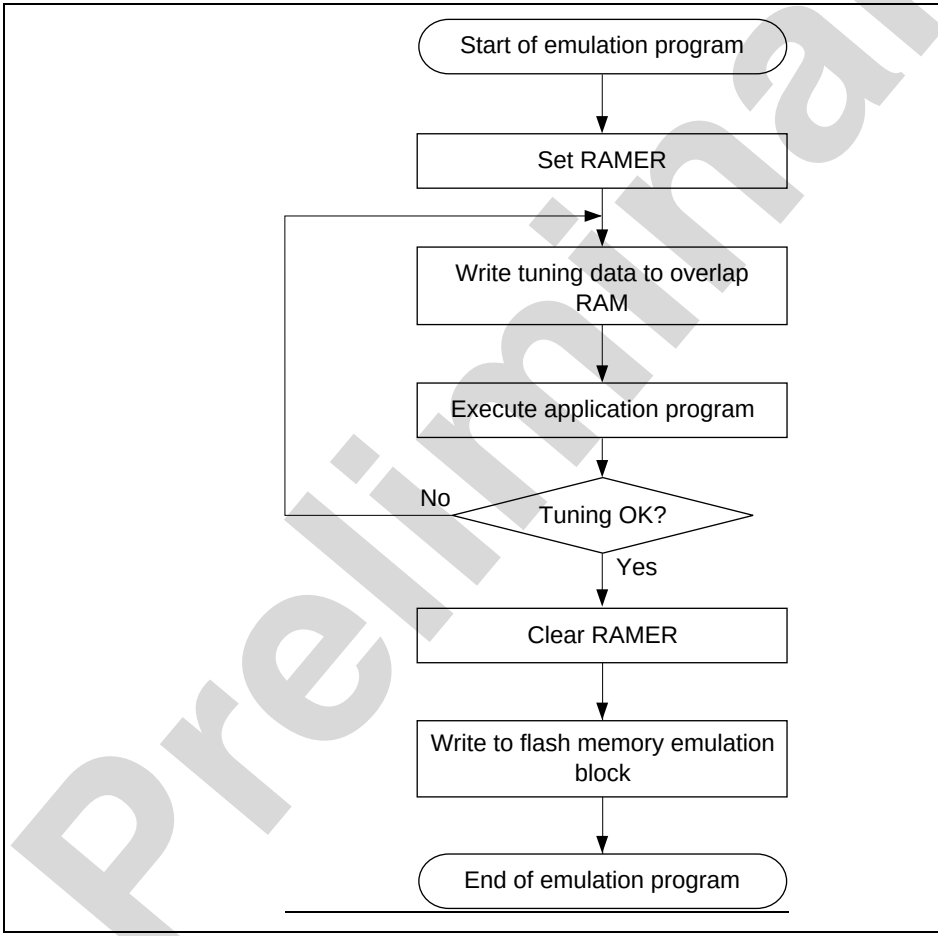


Figure 19.14 Flowchart for Flash Memory Emulation in RAM

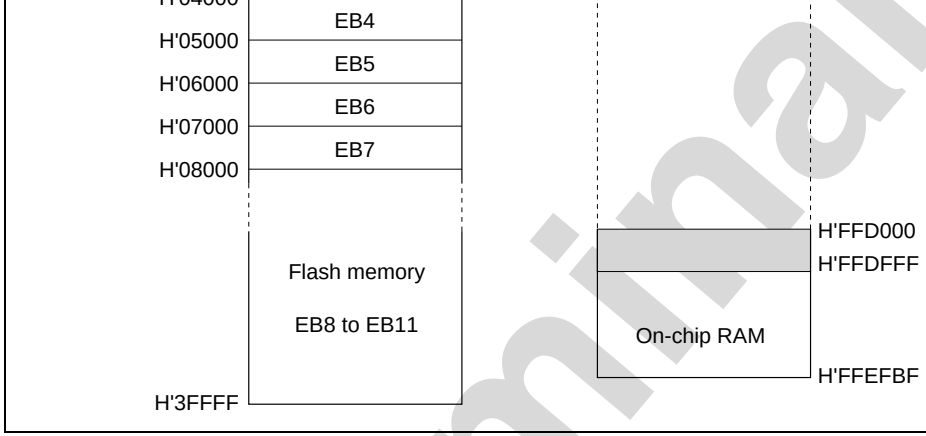


Figure 19.15 Example of RAM Overlap Operation

Example in which Flash Memory Block Area EB0 is Overlapped

1. Set bits RAMS, RAM2 to RAM0 in RAMER to 1, 0, 0, 0, to overlap part of RAM area (EB0) for which real-time programming is required.
2. Real-time programming is performed using the overlapping RAM.
3. After the program data has been confirmed, the RAMS bit is cleared, releasing RA
4. The data written in the overlapping RAM is written into the flash memory space (I

Notes: 1. When the RAMS bit is set to 1, program/erase protection is enabled for all regardless of the value of RAM2 to RAM0 (emulation protection). In this s the P1 or E1 bit in flash memory control register 1 (FLMCR1), will not cau transition to program mode or erase mode. When actually programming or flash memory area, the RAMS bit should be cleared to 0.

2. A RAM area cannot be erased by execution of software in accordance with algorithm while flash memory emulation in RAM is being used.
3. Block area EB0 contains the vector table. When performing RAM emulati vector table is needed in the overlap RAM.

2. In the interrupt exception handling sequence during programming or erasing, the ve not be read correctly*[2], possibly resulting in MCU runaway.
3. If interrupt occurred during boot program execution, it would not be possible to exe normal boot mode sequence.

For these reasons, in on-board programming mode alone there are conditions for disabl interrupt, as an exception to the general rule. However, this provision does not guarante erasing and programming or MCU operation. All requests, including NMI interrupt, mu therefore be restricted inside and outside the MCU when programming or erasing flash NMI interrupt is also disabled in the error-protection state while the P1 or E1 bit remai FLMCR1.

Notes: 1. Interrupt requests must be disabled inside and outside the MCU until the pro control program has completed programming.
2. The vector may not be read correctly in this case for the following two reaso
   - If flash memory is read while being programmed or erased (while the P1 is set in FLMCR1), correct read data will not be obtained (undetermined be returned).
   - If the interrupt entry in the vector table has not been programmed yet, in exception handling will not be executed correctly.

19.11 Flash Memory Programmer Mode

Programs and data can be written and erased in programmer mode as well as in the on- programming modes. In programmer mode, flash memory read mode, auto-program mo erase mode, and status read mode are supported. In auto-program mode, auto-erase mo status read mode, a status polling procedure is used, and in status read mode, detailed in signals are output after execution of an auto-program or auto-erase operation.

In programmer mode, set the mode pins to programmer mode (see table 19.10) and inp MHz input clock.

Table 19.10 shows the pin settings for programmer mode.

19.11.1 Socket Adapter Pin Correspondence Diagram

Connect the socket adapter to the chip as shown in figure 19.17. This will enable conv
40-pin arrangement. The on-chip ROM memory map is shown in figure 19.16, and the
adapter pin correspondence diagram in figure 19.17.

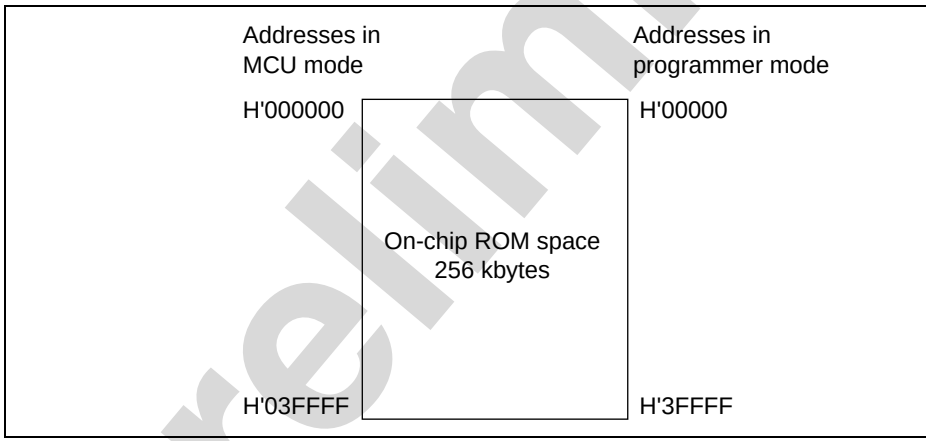
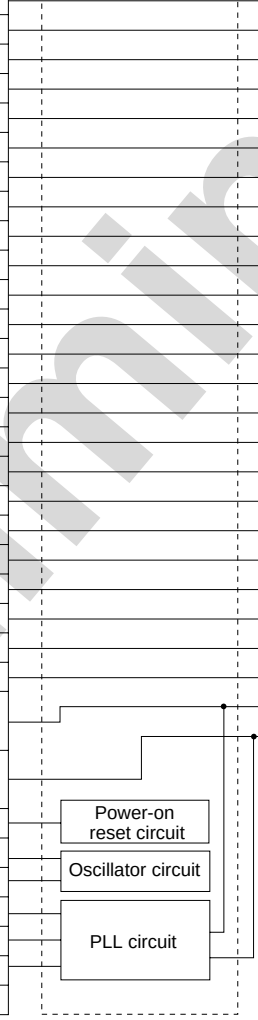


Figure 19.16 On-Chip ROM Memory Map

34	A6	27	A6
35	A7	28	A7
36	A8	29	A8
38	A9	31	A9
40	A10	32	A10
41	A11	33	A11
42	A12	34	A12
43	A13	35	A13
44	A14	36	A14
45	A15	37	A15
46	A16	38	A16
47	A17	39	A17
20	D8	19	I/O0
21	D9	18	I/O1
22	D10	17	I/O2
23	D11	16	I/O3
24	D12	15	I/O4
25	D13	14	I/O5
26	D14	13	I/O6
27	D15	12	I/O7
19	$\overline{CE}$	2	$\overline{CE}$
16	$\overline{OE}$	20	$\overline{OE}$
18	$\overline{WE}$	3	$\overline{WE}$
67	FWE	4	FWE
6, 9, 13, 17, 39, 52, 61, 62, 63, 75, 76, 77, 97	VCC	1, 40	VCC
2, 4, 8, 14, 15, 37, 48, 49, 53, 54, 55, 56, 65, 94, 95, 98	VSS	11, 30	VSS
		5, 6, 7	NC
60	$\overline{RES}$	8	A20
64	XTAL	9	A19
66	EXTAL	10	A18
59	PLL VCC		
58	PLLCAP		
57	PLL VSS		
Other than the above	NC (OPEN)		



Legend:
FWE: Flash write ena
I/O7–I/O0: Data input/outp
A18–A0: Address input
$\overline{CE}$: Chip enable
$\overline{OE}$: Output enable
$\overline{WE}$: Write enable

Figure 19.17 Socket Adapter Pin Correspondence Diagram

Auto-program mode supports programming of 128 bytes at a time. Status polling i
confirm the end of auto-programming.

- Auto-Erase Mode

  Auto-erase mode supports automatic erasing of the entire flash memory. Status po
  to confirm the end of auto-programming.

- Status Read Mode

  Status polling is used for auto-programming and auto-erasing, and normal termina
  confirmed by reading the I/O6 signal. In status read mode, error information is out
  error occurs.

Table 19.11 Settings for Various Operating Modes in Programmer Mode

	Pin Names					
Mode	**FWE**	**$\overline{CE}$**	**$\overline{OE}$**	**$\overline{WE}$**	**I/O7– I/O0**	**A**
Read	H or L	L	L	H	Data output	A
Output disable	H or L	L	H	H	Hi-Z	X
Command write	H or L*3	L	H	L	Data input	A
Chip disable*1	H or L	H	X	X	Hi-Z	X

Notes: 1. Chip disable is not a standby state; internally, it is an operation state.
2. Ain indicates that there is also address input in auto-program mode.
3. For command writes in auto-program and auto-erase modes, input a high l
FWE pin.

Notes: 1. In auto-program mode, 129 cycles are required for command writing by a sin
128-byte write.
2. In memory read mode, the number of cycles depends on the number of addr
cycles (n).

19.11.3 Memory Read Mode

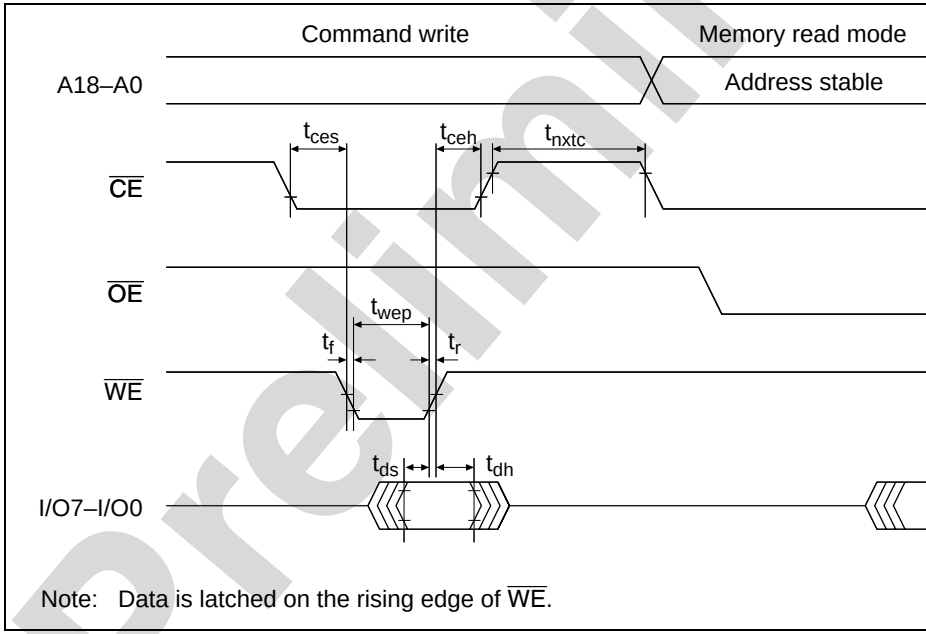
1. After completion of auto-program/auto-erase/status read operations, a transition is m
command wait state. When reading memory contents, a transition to memory read m
first be made with a command write, after which the memory contents are read.

2. In memory read mode, command writes can be performed in the same way as in the
wait state.

3. Once memory read mode has been entered, consecutive reads can be performed.

4. After powering on, memory read mode is entered.

Data hold time	t_{dh}	50	—	ns
Data setup time	t_{ds}	50	—	ns
Write pulse width	t_{wep}	70	—	ns
$\overline{WE}$ rise time	t_r	—	30	ns
$\overline{WE}$ fall time	t_f	—	30	ns



Note: Data is latched on the rising edge of $\overline{WE}$.

Figure 19.18 Timing Waveforms for Memory Read after Memory Wr

Data hold time	t_{dh}	50	—	ns
Data setup time	t_{ds}	50	—	ns
Write pulse width	t_{wep}	70	—	ns
$\overline{WE}$ rise time	t_r	—	30	ns
$\overline{WE}$ fall time	t_f	—	30	ns

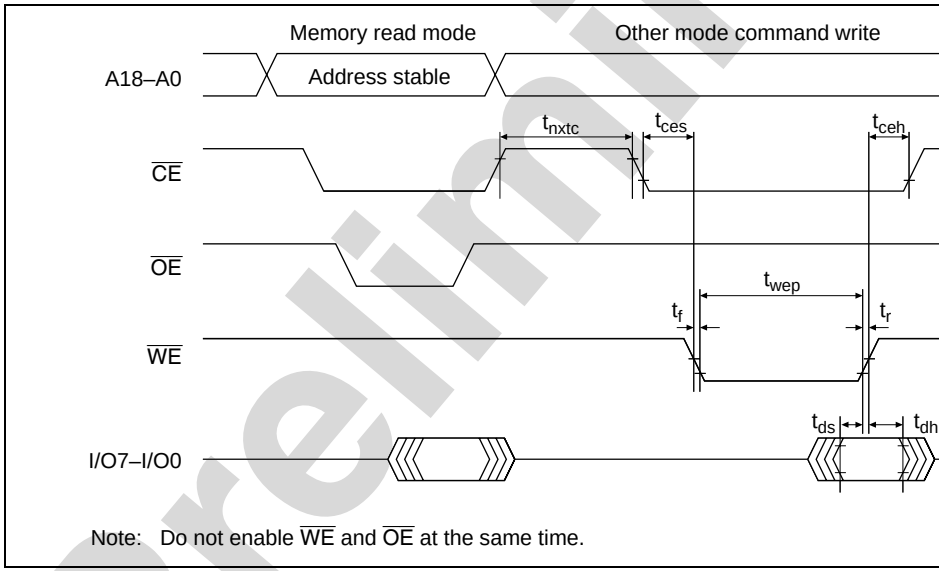


Figure 19.19 Timing Waveforms in Transition from Memory Read Mode to Ano

Output disable delay time	t_{df}	—	100	ns
Data output hold time	t_{oh}	5	—	ns

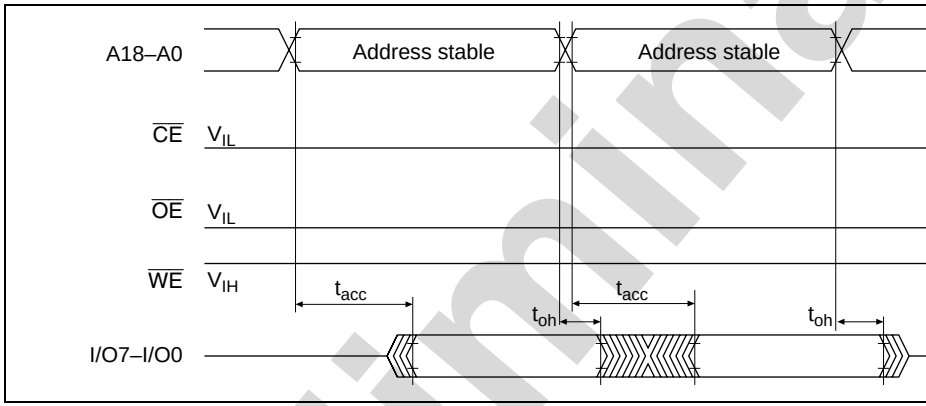


Figure 19.20 $\overline{CE}$ and $\overline{OE}$ Enable State Read Timing Waveforms

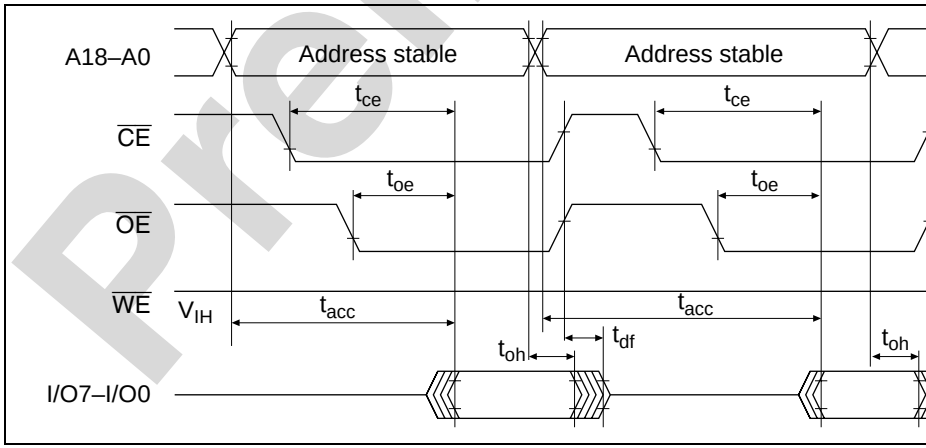


Figure 19.21 $\overline{CE}$ and $\overline{OE}$ Clock System Read Timing Waveforms

is input, processing will switch to a memory write operation but a write error will b

4. Memory address transfer is performed in the second cycle (figure 19.22). Do not pe transfer after the third cycle.

5. Do not perform a command write during a programming operation.

6. Perform one auto-program operation for a 128-byte block for each address. Two or additional programming operations cannot be performed on a previously programm block.

7. Confirm normal end of auto-programming by checking I/O6. Alternatively, status r can also be used for this purpose (I/O7 status polling uses the auto-program operati decision pin).

8. Status polling I/O6 and I/O7 pin information is retained until the next command wr as the next command write has not been performed, reading is possible by enabling $\overline{OE}$.

Data hold time	t_{dh}	50	—	ns
Data setup time	t_{ds}	50	—	ns
Write pulse width	t_{wep}	70	—	ns
Status polling start time	t_{wsts}	1	—	ms
Status polling access time	t_{spa}	—	150	ns
Address setup time	t_{as}	0	—	ns
Address hold time	t_{ah}	60	—	ns
Memory write time	t_{write}	1	3000	ms
Write setup time	t_{pns}	100	—	ns
Write end setup time	t_{pnh}	100	—	ns
$\overline{WE}$ rise time	t_r	—	30	ns
$\overline{WE}$ fall time	t_f	—	30	ns

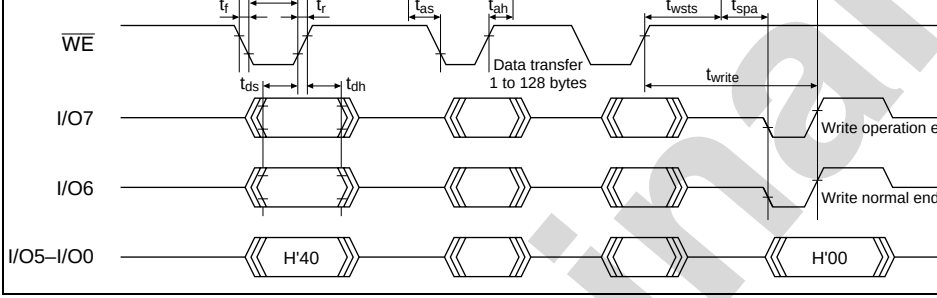


Figure 19.22 Auto-Program Mode Timing Waveforms

19.11.5 Auto-Erase Mode

1. Auto-erase mode supports only entire memory erasing.
2. Do not perform a command write during auto-erasing.
3. Confirm normal end of auto-erasing by checking I/O6. Alternatively, status read mo be used for this purpose (I/O7 status polling uses the auto-erase operation end decis
4. Status polling I/O6 and I/O7 pin information is retained until the next command wr as the next command write has not been performed, reading is possible by enabling $\overline{OE}$.

Data hold time	t_{dh}	50	—	ns
Data setup time	t_{ds}	50	—	ns
Write pulse width	t_{wep}	70	—	ns
Status polling start time	t_{ests}	1	—	ms
Status polling access time	t_{spa}	—	150	ns
Memory erase time	t_{erase}	100	40000	ms
Erase setup time	t_{ens}	100	—	ns
Erase end setup time	t_{enh}	100	—	ns
$\overline{WE}$ rise time	t_r	—	30	ns
$\overline{WE}$ fall time	t_f	—	30	ns

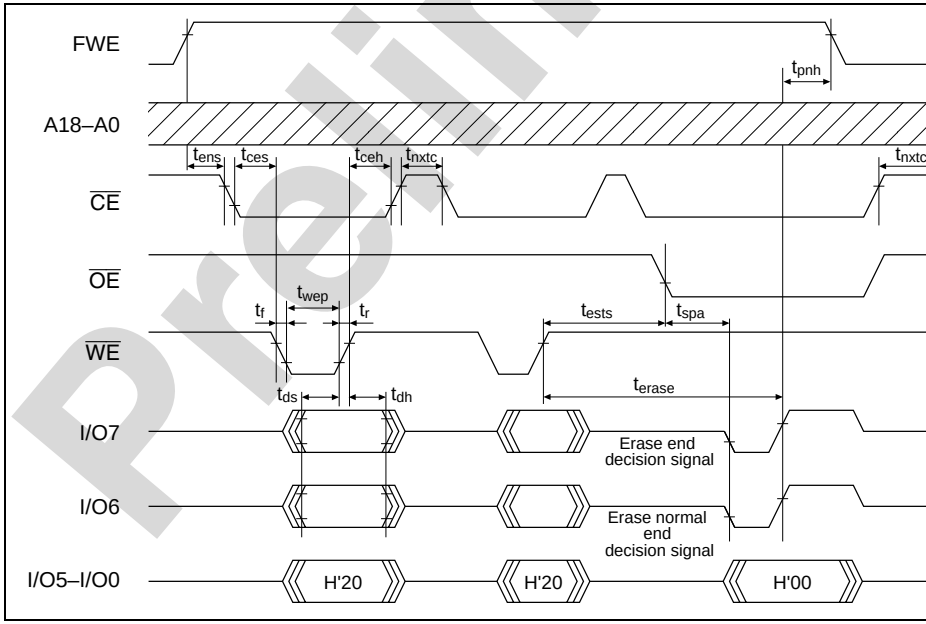
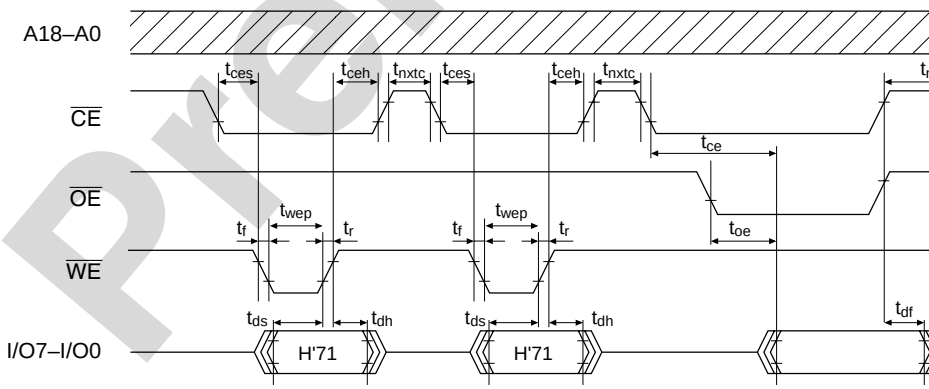


Figure 19.23 Auto-Erase Mode Timing Waveforms

Conditions: V_{CC} = 3.3 V ±0.3 V, V_{SS} = 0 V, T_a = 25°C ±5°C

Item	Symbol	Min	Max	Uni
Read time after command write	t_{nxtc}	20	—	µs
$\overline{CE}$ hold time	t_{ceh}	0	—	ns
$\overline{CE}$ setup time	t_{ces}	0	—	ns
Data hold time	t_{dh}	50	—	ns
Data setup time	t_{ds}	50	—	ns
Write pulse width	t_{wep}	70	—	ns
$\overline{OE}$ output delay time	t_{oe}	—	150	ns
Disable delay time	t_{df}	—	100	ns
$\overline{CE}$ output delay time	t_{ce}	—	150	ns
$\overline{WE}$ rise time	t_r	—	30	ns
$\overline{WE}$ fall time	t_f	—	30	ns



Note: I/O2 and I/O3 are undefined.

Figure 19.24 Status Read Mode Timing Waveforms

	Abnormal end: 1	Otherwise: 0	error: 1 Otherwise: 0	Otherwise: 0	Otherwise: 0

Note: I/O2 and I/O3 are undefined.

19.11.7 Status Polling

1. The I/O7 status polling flag indicates the operating status in auto-program/auto-era
2. The I/O6 status polling flag indicates a normal or abnormal end in auto-program/a mode.

Table 19.20 Status Polling Output Truth Table

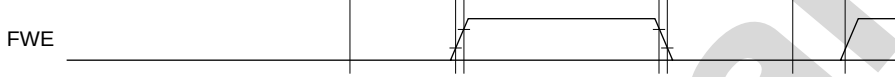
Pin Name	During Internal Operation	Abnormal End	—	Norm
I/O7	0	1	0	1
I/O6	0	0	1	1
I/O0–I/O5	0	0	0	0

19.11.8 Programmer Mode Transition Time

Commands cannot be accepted during the oscillation stabilization period or the progra setup period. After the programmer mode setup time, a transition is made to memory r

Table 19.21 Stipulated Transition Times to Command Wait State

Item	Symbol	Min	Max	Un
Standby release (oscillation stabilization time)	t_{osc1}	30	—	ms
Programmer mode setup time	t_{bmv}	10	—	ms
V_{CC} hold time	t_{dwn}	0	—	ms



Note: When using other than the automatic write mode and automatic erase mode, drive the input pin low.

Figure 19.25 Oscillation Stabilization Time, Boot Program Transfer Time, Power-Down Sequence

19.11.9 Notes on Memory Programming

1. When programming addresses which have previously been programmed, carry out erasing before auto-programming.
2. When performing programming using programmer mode on a chip that has been programmed/erased in an on-board programming mode, auto-erasing is recommend carrying out auto-programming.

Notes: 1. The flash memory is initially in the erased state when the device is shipped For other chips for which the erasure history is unknown, it is recommended erasing be executed to check and supplement the initialization (erase) level.

2. Auto-programming should be performed once only on the same address blo Additional programming cannot be performed on previously programmed a blocks.

read when the LSI is operating on the subclock.

(3) Standby mode: All flash memory circuits are halted, and the flash memory cannot written to.

States (2) and (3) are flash memory power-down states. Table 19.22 shows the corresp between the operating states of the LSI and the flash memory.

Table 19.22 Flash Memory Operating States

LSI Operating State	Flash Memory Operating State
High-speed mode	Normal mode (read/write)
Medium-speed mode	
Sleep mode	
Subactive mode	When PDWND = 0: Power-down mode (read-onl
Subsleep mode	When PDWND = 1: Normal mode (read-only)
Watch mode	Standby mode
Software standby mode	
Hardware standby mode	

19.12.1 Note on Power-Down States

When the flash memory is in a power-down state, part or all of the internal power sup is halted. Therefore, a power supply circuit stabilization period must be provided whe to normal operation. When the flash memory returns to its normal operating state from down state, bits STS2 to STS0 in SBYCR must be set to provide a wait time of at leas (power supply stabilization time), even if an oscillation stabilization period is not nece

flash memory (F-ZTAT256V3A).

Do not select the HN27C4096 setting for the PROM programmer, and only use the
socket adapter. Failure to observe these points may result in damage to the device.

2. Powering on and off (see figures 19.26 to 19.28)

Do not apply a high level to the FWE pin until V_{CC} has stabilized. Also, drive the FW
before turning off V_{CC}.

When applying or disconnecting V_{CC} power, fix the FWE pin low and place the flas
in the hardware protection state.

The power-on and power-off timing requirements should also be satisfied in the eve
power failure and subsequent recovery.

3. FWE application/disconnection (see figures 19.26 to 19.28)

FWE application should be carried out when MCU operation is in a stable condition
operation is not stable, fix the FWE pin low and set the protection state.

The following points must be observed concerning FWE application and disconnect
prevent unintentional programming or erasing of flash memory:

- Apply FWE when the V_{CC} voltage has stabilized within its rated voltage range.

  Apply FWE when oscillation has stabilized (after the elapse of the oscillation se
  time).
- In boot mode, apply and disconnect FWE during a reset.
- In user program mode, FWE can be switched between high and low level regard
  $\overline{\text{RES}}$ input.

  FWE input can also be switched during execution of a program in flash memory
- Do not apply FWE if program runaway has occurred.
- Disconnect FWE only when the SWE1, ESU1, PSU1, EV1, PV1, P1, and E1 bit
  FLMCR1 are cleared.

  Make sure that the SWE1, ESU1, PSU1, EV1, PV1, P1, and E1 bits are not set b
  when applying or disconnecting FWE.

subjecting the device to voltage stress or sacrificing program data reliability. When
P1 or E1 bit in FLMCR1, the watchdog timer should be set beforehand as a precau
program runaway, etc.

6. **Do not set or clear the SWE1 bit during execution of a program in flash mem**

   Do not set or clear the SWE1 bit during execution of a program in flash memory. W
   least 100 µs after clearing the SWE1 bit before executing a program or reading dat
   memory. When the SWE1 bit is set, data in flash memory can be rewritten, but wh
   1, flash memory can only be read in program-verify or erase-verify mode. Access
   memory only for verify operations (verification during programming/erasing). Do
   SWE1 bit during programming, erasing, or verifying.

   Similarly, when using the RAM emulation function while a high level is being inp
   FWE pin, the SWE1 bit must be cleared before executing a program or reading da
   memory. However, the RAM area overlapping flash memory space can be read an
   regardless of whether the SWE1 bit is set or cleared.

7. **Do not use interrupts while flash memory is being programmed or erased.**

   All interrupt requests, including NMI, should be disabled during FWE application
   priority to program/erase operations.

8. **Do not perform additional programming. Erase the memory before reprogra**

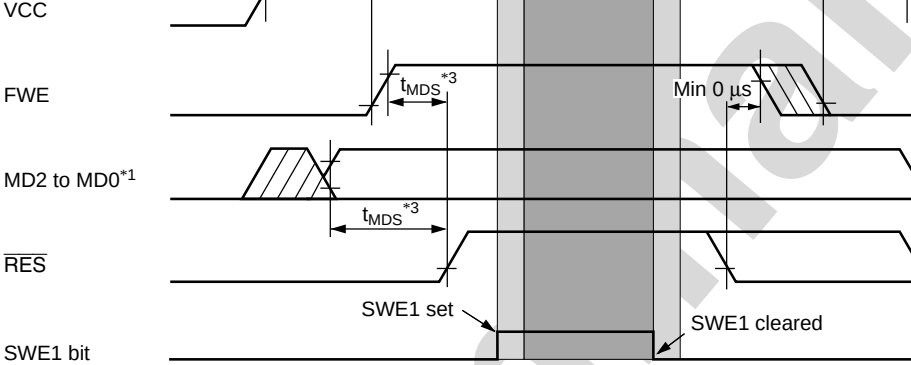
   In on-board programming, perform only one programming operation on a 128-byt
   programming unit block. In programmer mode, also, perform only one programmi
   on a 128-byte programming unit block.

9. **Before programming, check that the chip is correctly mounted in the PROM programmer.**

   Overcurrent damage to the device can result if the index marks on the PROM prog
   socket, socket adapter, and chip are not correctly aligned.

10. **Do not touch the socket adapter or chip during programming.**

    Touching either of these can cause contact faults and write errors.

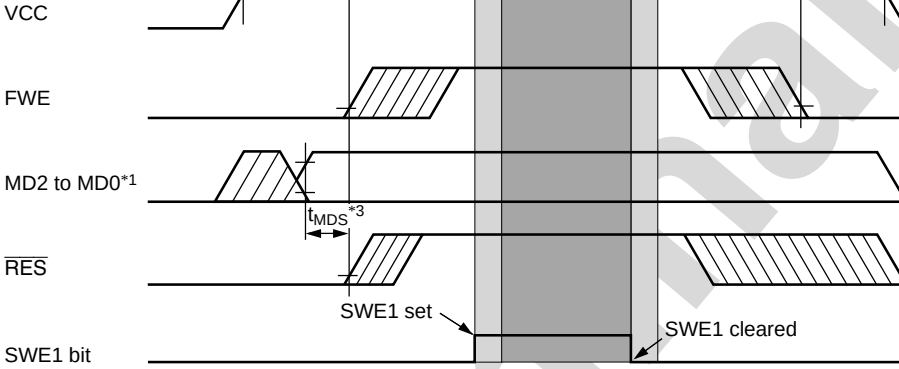


Period during which flash memory access is prohibited
(x: Wait time after setting SWE1 bit)*2

Period during which flash memory can be programmed
(Execution of program in flash memory prohibited, and data reads other than verify operations prohibited)

Notes: 1. Except when switching modes, the level of the mode pins (MD2–MD0) must be fixed until off by pulling the pins up or down.
2. See section 22.6, Flash Memory Characteristics.
3. Mode programming setup time t_{MDS} (min) = 200 ns

Figure 19.26 Power-On/Off Timing (Boot Mode)

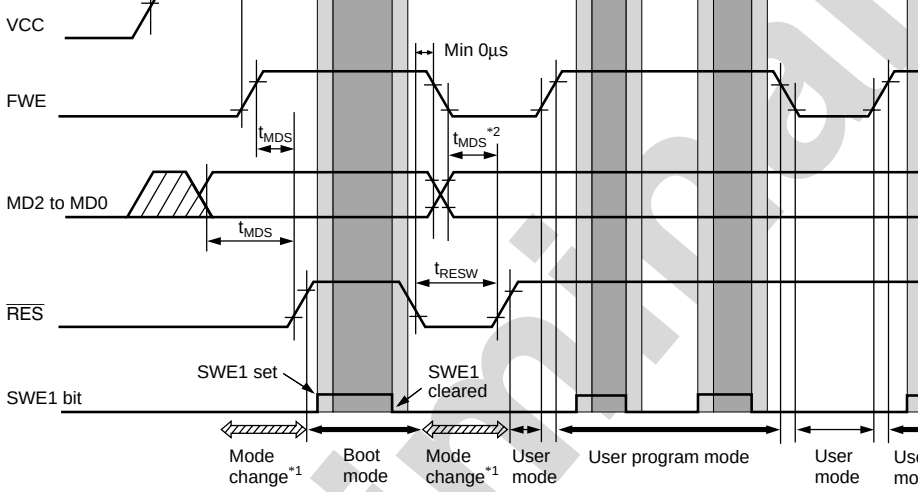


Period during which flash memory access is prohibited
(x: Wait time after setting SWE1 bit)*2

Period during which flash memory can be programmed
(Execution of program in flash memory prohibited, and data reads other than verify operation prohibited)

Notes: 1. Except when switching modes, the level of the mode pins (MD2–MD0) must be fixed u off by pulling the pins up or down.
2. See section 22.6, Flash Memory Characteristics.
3. Mode programming setup time t_{MDS} (min) = 200 ns

Figure 19.27 Power-On/Off Timing (User Program Mode)



Period during which flash memory access is prohibited
(x: Wait time after setting SWE1 bit)*3

Period during which flash memory can be programmed
(Execution of program in flash memory prohibited, and data reads other than verify operations prohibited)

Notes: 1. When entering boot mode or making a transition from boot mode to another mode, mode switching mu
out by means of RES input. The state of ports with multiplexed address functions and bus control outpu
(AS, RD, WR) will change during this switchover interval (the interval during which the RES pin input is
and therefore these pins should not be used as output signals during this time.
2. When making a transition from boot mode to another mode, a mode programming setup time, t_{MDS} (mi
is necessary with respect to the RES clearance timing.
3. See section 22.6, Flash Memory Characteristics.

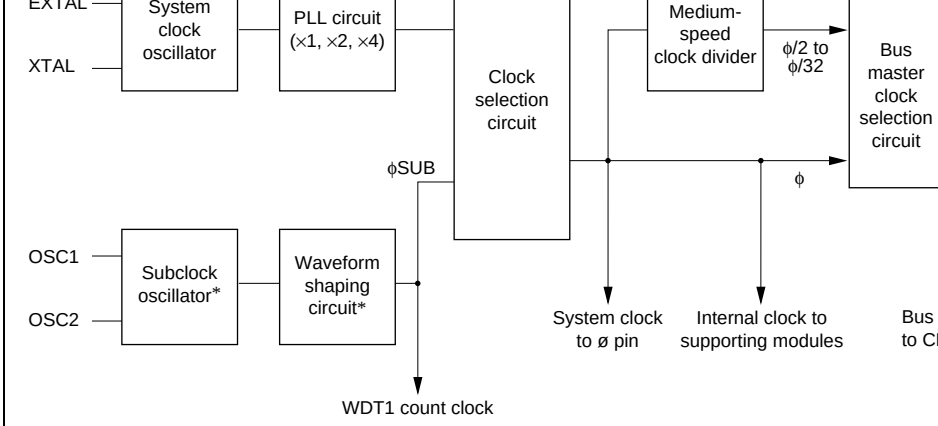
Figure 19.28 Mode Transition Timing
(Example: Boot Mode → User Mode ↔ User Program Mode)

Table 19.23 Registers Present in F-ZTAT Version but Absent in Mask ROM Ver

Register	Abbreviation	Address
Flash memory control register 1	FLMCR1	H'FFA8
Flash memory control register 2	FLMCR2	H'FFA9
Erase block register 1	EBR1	H'FFAA
Erase block register 2	EBR2	H'FFAB
RAM emulation register	RAMER	H'FEDB
Flash memory power control register	FLPWCR	H'FFAC

The clock pulse generator consists of an oscillator, PLL (phase-locked loop) circuit, c
selection circuit, medium-speed clock divider, bus master clock selection circuit, subc
oscillator*, and waveform shaping circuit*. The frequency can be changed by means c
circuit in the CPG. Frequency changes are performed by software by means of setting
system clock control register (SCKCR) and low-power control register (LPWRCR).

Note: * Supported only in the H8S/2626 Group; not available in the H8S/2623 Gro



Legend:
LPWRCR: Low-power control register
SCKCR: System clock control register

Note: * Supported only in the H8S/2626 Group, not available in the H8S/2623 Group.

Figure 20.1 Block Diagram of Clock Pulse Generator

20.1.2 Register Configuration

The clock pulse generator is controlled by SCKCR and LPWRCR. Table 20.1 shows th
configuration.

Table 20.1 Clock Pulse Generator Register

Name	Abbreviation	R/W	Initial Value	Ad
System clock control register	SCKCR	R/W	H'00	H'
Low-power control register	LPWRCR	R/W	H'00	H'

Note: * Lower 16 bits of the address.

R/W . R/W — — — R/W R/W R/W

SCKCR is an 8-bit readable/writable register that performs φ clock output control, sel operation when the PLL circuit frequency multiplication factor is changed, and mediu mode control.

SCKCR is initialized to H'00 by a reset and in hardware standby mode. It is not initial software standby mode.

Bit 7—φ Clock Output Disable (PSTOP): Controls φ output.

	Description			
Bit 7 PSTOP	**High-speed Mode, Medium-Speed Mode**	**Sleep Mode**	**Software Standby Mode**	**Hardw Standb**
0	φ output (initial value)	φ output	Fixed high	High im
1	Fixed high	Fixed high	Fixed high	High im

Bits 6 to 4—Reserved: These bits are always read as 0 and cannot be modified.

Bit 3—Frequency Multiplication Factor Switching Mode Select (STCS): Selects t when the PLL circuit frequency multiplication factor is changed.

Bit 3 STCS	**Description**
0	Specified multiplication factor is valid after transition to software standby m (
1	Specified multiplication factor is valid immediately after STC bits are rewrit

Bits 2 to 0—System Clock Select 2 to 0 (SCK2 to SCK0): These bits select the bus clock.

	1	Medium-speed clock is φ/32
1	—	—

20.2.2 Low-Power Control Register (LPWRCR)

Bit	7	6	5	4	3	2	1
	DTON	LSON	NESEL	SUBSTP	RFCUT	—	STC1
Initial value	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W

LPWRCR is an 8-bit readable/writable register that performs power-down mode contro
LPWRCR is initialized to H'00 by a reset and in hardware standby mode. It is not initi
manual reset or in software standby mode.

Bits 7 to 2—Reserved: The function of these bits differs between the H8S/2623 Group
H8S/2626 Group.

For details see sections 21A.2.3, 21B.2.3, Low-Power Control Register (LPWRCR).

Bits 1 and 0—Frequency Multiplication Factor (STC1, STC0): The STC bits specif
frequency multiplication factor of the PLL circuit.

Bit 1	Bit 0	
STC1	**STC0**	**Description**
0	0	×1 (In
	1	×2
1	0	×4
	1	Setting prohibited

In either case, the input clock should not exceed 20 MHz.

20.3.1 Connecting a Crystal Resonator

Circuit Configuration: A crystal resonator can be connected as shown in the exampl
20.2. Select the damping resistance R_d according to table 20.2. An AT-cut parallel-res
crystal should be used.

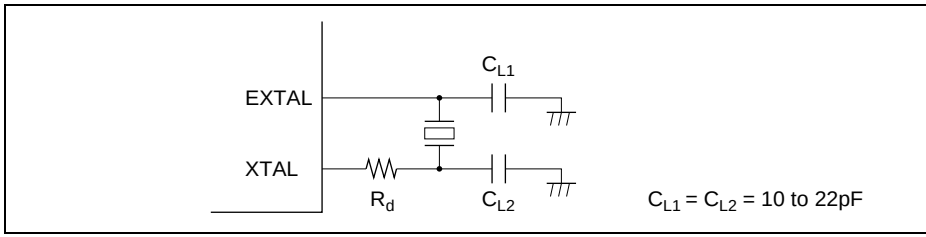


Figure 20.2 Connection of Crystal Resonator (Example)

Table 20.2 Damping Resistance Value

Frequency (MHz)	4	8	12	16
R_d (Ω)	500	200	0	0

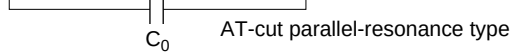


Figure 20.3 Crystal Resonator Equivalent Circuit

Table 20.3 Crystal Resonator Parameters

Frequency (MHz)	4	8	12	16	2
R_S max (Ω)	120	80	60	50	4
C_0 max (pF)	7	7	7	7	7

Note on Board Design: When a crystal resonator is connected, the following points sh noted:

Other signal lines should be routed away from the oscillator circuit to prevent induction interfering with correct oscillation. See figure 20.4.

When designing the board, place the crystal resonator and its load capacitors as close a to the XTAL and EXTAL pins.

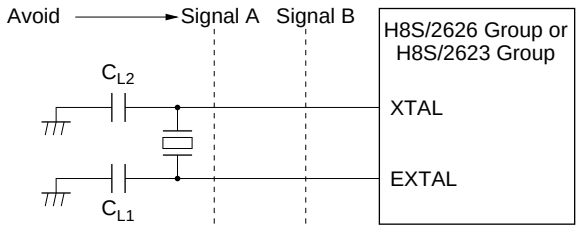
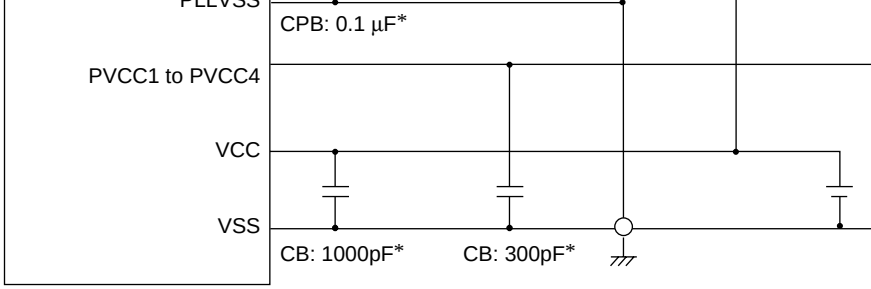


Figure 20.4 Example of Incorrect Board Design



Note: * CB and CPB are laminated ceramic capacitors.

Figure 20.5 Points for Attention when Using PLL Oscillation Circui

Place oscillation stabilization capacitor C1 and resistor R1 close to the PLLCAP pin, a that no other signal lines cross this line. Supply the C1 ground from PLLVSS.

Separate $PLLV_{CC}$ and $PLLV_{SS}$ from the other V_{CC}/V_{SS} and PV_{CC}/PV_{SS} lines at the board supply source, and be sure to insert bypass capacitors CPB and CB close to the pins.

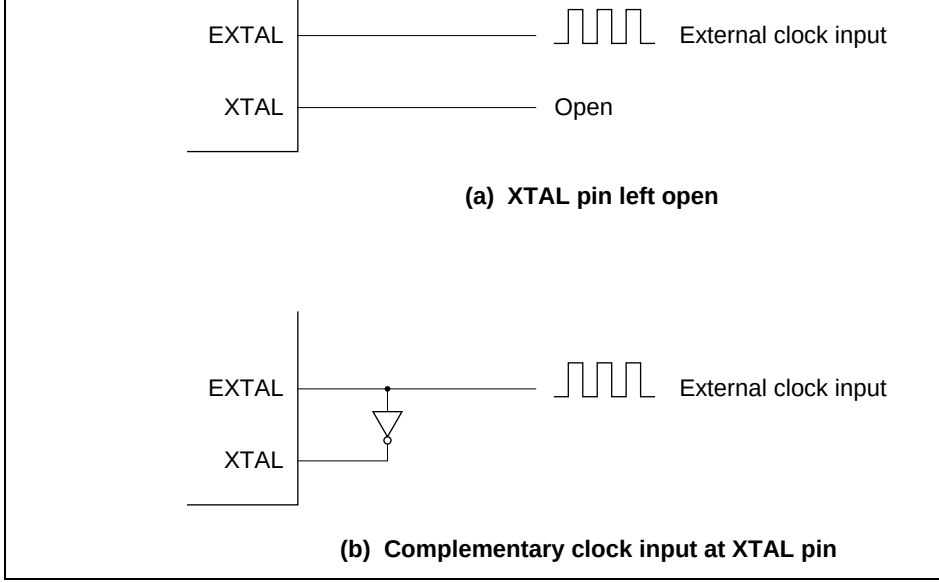


Figure 20.6 External Clock Input (Examples)

External clock input low pulse width	t_{EXL}	15	—	ns	Figure 20.7
External clock input high pulse width	t_{EXH}	15	—	ns	
External clock rise time	t_{EXr}	—	5	ns	
External clock fall time	t_{EXf}	—	5	ns	
Clock low pulse width level	t_{CL}	0.4	0.6	t_{cyc}	$\phi \geq 5$ MHz
		80	—	ns	$\phi < 5$ MHz
Clock high pulse width level	t_{CH}	0.4	0.6	t_{cyc}	$\phi \geq 5$ MHz
		80	—	ns	$\phi < 5$ MHz

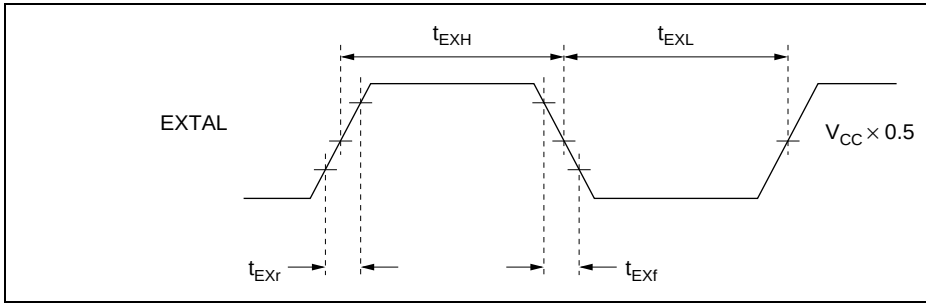


Figure 20.7 External Clock Input Timing

When the multiplication factor of the PLL circuit is changed, the operation varies accor
setting of the STCS bit in SCKCR.

When STCS = 0 (initial value), the setting becomes valid after a transition to software s
mode. The transition time count is performed in accordance with the setting of bits STS
in SBYCR.

[1] The initial PLL circuit multiplication factor is 1.

[2] A value is set in bits STS2 to STS0 to give the specified transition time.

[3] The target value is set in STC1 and STC0, and a transition is made to software stand

[4] The clock pulse generator stops and the value set in STC1 and STC0 becomes valid

[5] Software standby mode is cleared, and a transition time is secured in accordance wi
setting in STS2 to STS0.

[6] After the set transition time has elapsed, the LSI resumes operation using the target
multiplication factor.

If a PC break is set for the SLEEP instruction that causes a transition to software standb
[3], software standby mode is entered and break exception handling is executed after th
oscillation stabilization time. In this case, the instruction following the SLEEP instructi
executed after execution of the RTE instruction.

When STCS = 1, the LSI operates on the changed multiplication factor immediately aft
STC1 and STC0 are rewritten.

20.5 Medium-Speed Clock Divider

The medium-speed clock divider divides the system clock to generate $\phi/2$, $\phi/4$, $\phi/8$, $\phi/1$

(1) Connecting 32.768kHz Crystal Oscillator

To supply a clock to the subclock oscillator, connect a 32.768kHz crystal oscillator, as
figure 20.8. See section 20.3.1, Connecting a Crystal Resonator.

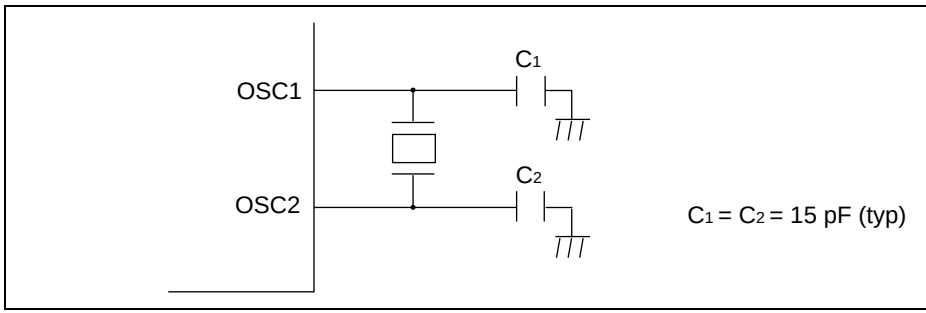


Figure 20.8 Example Connection of 32.768 kHz Crystal Oscillator

Figure 20.9 shows the equivalence circuit for a 32.768kHz oscillator.

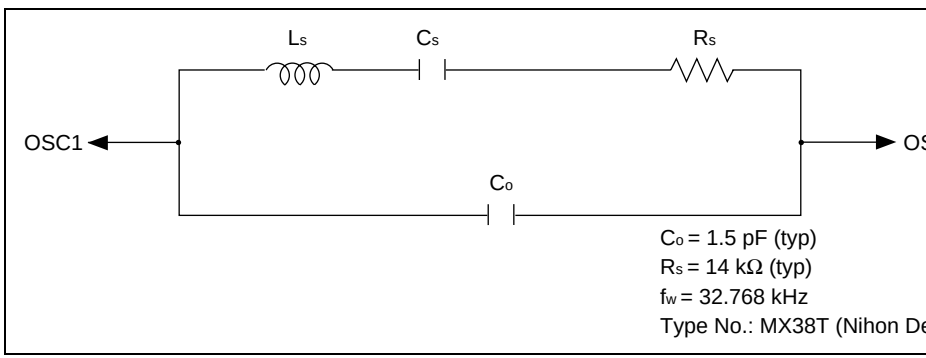


Figure 20.9 Equivalence Circuit for 32.768 kHz Oscillator

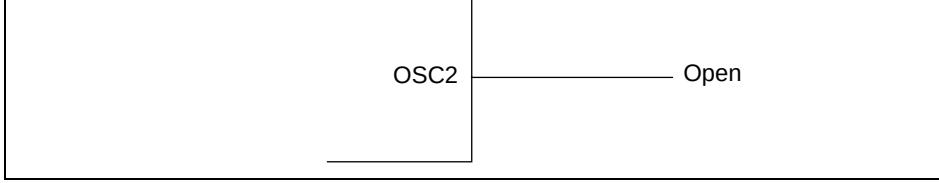


Figure 20.10 Pin Handling When Subclock Not Required

20.8 Subclock Waveform Shaping Circuit (H8S/2626 Group Only

To eliminate noise from the subclock input to OSC1, the subclock is sampled using the clock ϕ. The sampling frequency is set using the NESEL bit of LPWRCR. For details, s 21B.2.3, Low Power Control Register (LPWRCR).

No sampling is performed in sub-active mode, sub-sleep mode, or watch mode.

20.9 Note on Crystal Resonator

Since various characteristics related to the crystal resonator are closely linked to the use design, thorough evaluation is necessary on the user's part, for both the mask versions a F-ZTAT versions, using the resonator connection examples shown in this section as a g the resonator circuit ratings will depend on the floating capacitance of the resonator and mounting circuit, the ratings should be determined in consultation with the resonator manufacturer. The design must ensure that a voltage exceeding the maximum rating is n to the oscillator pin.

In addition to the normal program execution state, the H8S/2623 Group has five powe
modes in which operation of the CPU and oscillator is halted and power dissipation is
Low-power operation can be achieved by individually controlling the CPU, on-chip su
modules, and so on.

The H8S/2623 operating modes are as follows:

(1) High-speed mode
(2) Medium-speed mode
(3) Sleep mode
(4) Module stop mode
(5) Software standby mode
(6) Hardware standby mode

(2) to (6) are power-down modes. Sleep mode is CPU states, medium-speed mode is a
bus master state, and module stop mode is an internal peripheral function (including b
other than the CPU) state. Some of these states can be combined.

After a reset, the LSI is in high-speed mode with modules other than the DTC in modu
mode.

Note: Subclock functions (subactive mode, subsleep mode, and watch mode) are not
the H8S/2623 Group.

Table 21A.1 shows the internal state of the LSI in the respective modes.

Figure 21A.1 is a mode transition diagram.

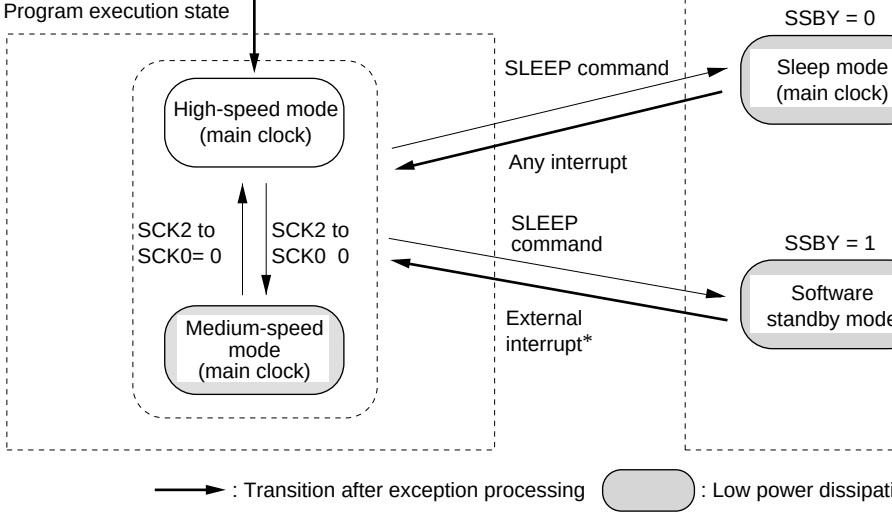
Interrupts	IRQ0–IRQ5						
Peripheral functions	WDT0	Functioning	Functioning	Functioning	—	Halted (retained)	H
	DTC	Functioning	Medium-speed operation	Functioning	Halted (retained)	Halted (retained)	H
	TPU	Functioning	Functioning (PBC medium-speed operation)	Functioning	Halted (retained)	Halted (retained)	H
	PBC						
	PPG						
	SCI0	Functioning	Functioning	Functioning	Halted (reset)	Halted (reset)	H
	SCI1						
	SCI2						
	PWM						
	A/D						
	RAM	Functioning	Functioning	Functioning (DTC)	Functioning	Retained	R
	I/O	Functioning	Functioning	Functioning	Functioning	Retained	H
	HCAN	Functioning	Functioning*	Functioning	Halted (reset)	Halted (reset)	H

Notes: “Halted (retained)” means that internal register values are retained. The internal
“operation suspended.”

“Halted (reset)” means that internal register values and internal states are initializ

In module stop mode, only modules for which a stop setting has been made are
(reset or retained).

* Note, however, that registers cannot be read or written to.



Notes: When a transition is made between modes by means of an interrupt, the transition cann
on interrupt source generation alone. Ensure that interrupt handling is performed after a
interrupt request.
From any state except hardware standby mode, a transition to the reset state occurs wh
driven low.
From any state, a transition to hardware standby mode occurs when $\overline{\text{STBY}}$ is driven low

* NMI and IRQ0 to IRQ5

Figure 21A.1 Mode Transition Diagram

System clock control register	SCKCR	R/W	H'00	H'F
Low power control register	LPWRCR	R/W	H'00	H'F
Module stop control register A, B, C	MSTPCRA	R/W	H'3F	H'F
	MSTPCRB	R/W	H'FF	H'F
	MSTPCRC	R/W	H'FF	H'F

Note: * Lower 16 bits of the address.

21A.2 Register Descriptions

21A.2.1 Standby Control Register (SBYCR)

Bit	:	7	6	5	4	3	2	1
		SSBY	STS2	STS1	STS0	OPE	—	—
Initial value	:	0	0	0	0	1	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	—	—

SBYCR is an 8-bit readable/writable register that performs power-down mode control.

SBYCR is initialized to H'08 by a reset and in hardware standby mode. It is not initializ software standby mode.

Bit 7—Software Standby (SSBY): When making a low power dissipation mode transi executing the SLEEP instruction, the operating mode is determined in combination wit control bits.

Note that the value of the SSBY bit does not change even when shifting between mode interrupts.

for clock stabilization when shifting to high-speed mode or medium-speed mode by us
specific interrupt or command to cancel software standby mode. With a quartz oscillat
21A.4), select a wait time of 8ms (oscillation stabilization time) or more, depending o
operating frequency. With an external clock, there are no specific wait requirements.

Bit 6	Bit 5	Bit 4	
STS2	**STS1**	**STS0**	**Description**
0	0	0	Standby time = 8192 states (I
		1	Standby time = 16384 states
	1	0	Standby time = 32768 states
		1	Standby time = 65536 states
1	0	0	Standby time = 131072 states
		1	Standby time = 262144 states
	1	0	Reserved
		1	Standby time = 16 states

Bit 3—Output Port Enable (OPE): This bit specifies whether the output of the addr
bus control signals ($\overline{AS}$, $\overline{RD}$, $\overline{HWR}$, $\overline{LWR}$) is retained or set to high-impedance state i
software standby mode.

Bit 3	
OPE	**Description**
0	In software standby mode, address bus and bus control signals are high-i
1	In software standby mode, the output state of the address bus and bus co is retained. (I

Bits 2 to 0—Reserved: These bits always return 0 when read, and cannot be written t

SCKCR is an 8-bit readable/writable register that performs φ clock output control and
speed mode control.

SCKCR is initialized to H'00 by a reset and in hardware standby mode. It is not initializ
software standby mode.

Bit 7—φ Clock Output Disable (PSTOP): In combination with the DDR of the applic
this bit controls φ output. See section 21A.8, φ Clock Output Disable Function, for deta

	Description			
Bit 7 PSTOP	**High-Speed Mode, Medium-Speed Mode**	**Sleep Mode**	**Software Standby Mode**	**Hardwar Mode**
0	φ output (initial value)	φ output	Fixed high	High imp
1	Fixed high	Fixed high	Fixed high	High imp

Bits 6 to 4—Reserved: These bits are always read as 0 and cannot be modified.

Bit 3—Frequency Multiplication Factor Switching Mode Select (STCS): Selects th
when the PLL circuit frequency multiplication factor is changed.

Bit 3 STCS	**Description**
0	Specified multiplication factor is valid after transition to software standby m (I
1	Specified multiplication factor is valid immediately after STC bits are rewritt

Bits 2 to 0—System clock select (SCK2 to SCK0): These bits select the bus master cl
high-speed mode, and medium-speed mode.

	1	Medium-speed clock is φ/32	
1	—	—	

21A.2.3 Low-Power Control Register (LPWRCR)

Bit	:	7	6	5	4	3	2	1
		DTON	LSON	NESEL	SUBSTP	RFCUT	—	STC1
Initial value	:	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The LPWRCR is an 8-bit read/write register that controls the low power dissipation m

The LPWRCR is initialized to H'00 at a reset and when in hardware standby mode. It initialized in software standby mode. The following describes bits 7 to 2. For details o see section 20.2.2, Low-Power Control Register (LPWRCR).

Bits 7 to 4—Reserved: Bits DTON, LSON, NESEL, and SUBSTP must always be w in the H8S/2623 Group, as this version does not support subclock operation.

Bit 3—Oscillation Circuit Feedback Resistance Control Bit (RFCUT): This bit tu internal feedback resistance of the main clock oscillation circuit ON/OFF.

Bit 3

RFCUT	Description
0	When the main clock is oscillating, sets the feedback resistance ON. When t clock is stopped, sets the feedback resistance OFF. (I
1	Sets the feedback resistance OFF.

Bit 2—Reserved: Only write 0 to this bit.

MSTPCRB

Bit	:	7	6	5	4	3	2	1
		MSTPB7	MSTPB6	MSTPB5	MSTPB4	MSTPB3	MSTPB2	MSTPB
Initial value	:	1	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MSTPCRC

Bit	:	7	6	5	4	3	2	1
		MSTPC7	MSTPC6	MSTPC5	MSTPC4	MSTPC3	MSTPC2	MSTPC
Initial value	:	1	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MSTPCR, comprising three 8-bit readable/writable registers, performs module stop mo

MSTPCRA to MSTPCRC are initialized to H'3FFFFF by a reset and in hardware stand
They are not initialized in software standby mode.

**MSTPCRA/MSTPCRB/MSTPCRC Bits 7 to 0—Module Stop (MSTPA7 to MSTP
MSTPB7 to MSTPB0, MSTPC7 to MSTPC0, MSTPD7 and MSTPD6):** These bits
module stop mode. See table 21A.3 for the method of selecting the on-chip peripheral f

MSTPCRA/MSTPCRB/ MSTPCRC Bits 7 to 0	
MSTPA7 to MSTPA0, MSTPB7 to MSTPB0, MSTPC7 to MSTPC0	**Description**
0	Module stop mode is cleared (initial value of MSTPA7 and MS
1	Module stop mode is set (initial value of MSTPA5–0, MSTPB7- MSTPC7–0)

the bus master operating clock. For example, if ϕ/4 is selected as the operating clock,
memory is accessed in 4 states, and internal I/O registers in 8 states.

Medium-speed mode is cleared by clearing all of bits SCK2 to SCK0 to 0. A transitio
high-speed mode and medium-speed mode is cleared at the end of the current bus cyc

If a SLEEP instruction is executed when the SSBY bit in SBYCR is cleared to 0, a tra
made to sleep mode. When sleep mode is cleared by an interrupt, medium-speed mod

When the SLEEP instruction is executed with the SSBY bit = 1, operation shifts to the
standby mode. When software standby mode is cleared by an external interrupt, mediu
mode is restored.

When the $\overline{\text{RES}}$ pin is set low and medium-speed mode is cancelled, operation shifts to
state. The same applies in the case of a reset caused by overflow of the watchdog time

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

Figure 21A.2 shows the timing for transition to and clearance of medium-speed mode

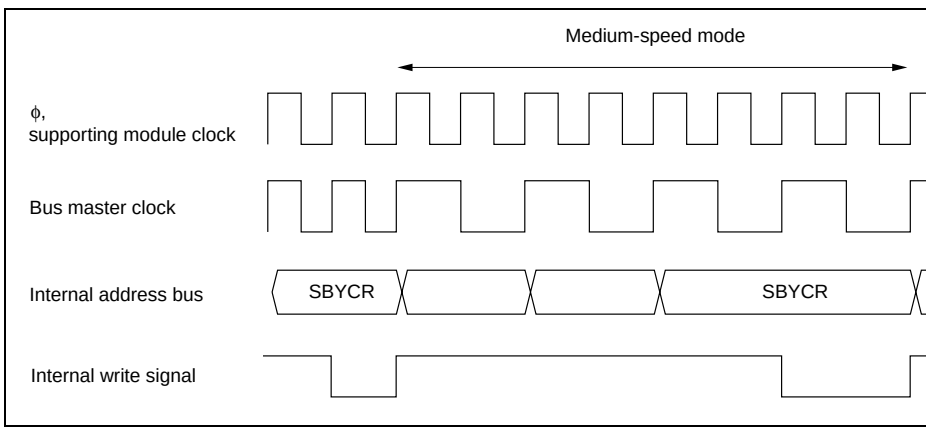


Figure 21A.2 Medium-Speed Mode Transition and Clearance Timin

21A.4.2 Exiting Sleep Mode

Sleep mode is exited by any interrupt, or signals at the $\overline{\text{RES}}$, or $\overline{\text{STBY}}$ pins.

Exiting Sleep Mode by Interrupts: When an interrupt occurs, sleep mode is exited an exception processing starts. Sleep mode is not exited if the interrupt is disabled, or inte than NMI are masked by the CPU.

Exiting Sleep Mode by $\overline{\text{RES}}$ pin: Setting the $\overline{\text{RES}}$ pin level Low selects the reset state stipulated reset input duration, driving the $\overline{\text{RES}}$ pin High starts the CPU performing res exception processing.

Exiting Sleep Mode by $\overline{\text{STBY}}$ Pin: When the $\overline{\text{STBY}}$ pin level is driven Low, a transiti to hardware standby mode.

21A.5 Module Stop Mode

21A.5.1 Module Stop Mode

Module stop mode can be set for individual on-chip supporting modules.

When the corresponding MSTP bit in MSTPCR is set to 1, module operation stops at th the bus cycle and a transition is made to module stop mode. The CPU continues operati independently.

Table 21A.3 shows MSTP bits and the corresponding on-chip supporting modules.

When the corresponding MSTP bit is cleared to 0, module stop mode is cleared and th starts operating at the end of the bus cycle. In module stop mode, the internal states of m other than the SCI, A/D converter and HCAN are retained.

After reset clearance, all modules other than DTC are in module stop mode.

	MSTPA5	16-bit timer pulse unit (TPU)
	MSTPA4*	
	MSTPA3	Programmable pulse generator (PPG)
	MSTPA2*	
	MSTPA1	A/D converter
	MSTPA0*	
MSTPCRB	MSTPB7	Serial communication interface 0 (SCI0)
	MSTPB6	Serial communication interface 1 (SCI1)
	MSTPB5	Serial communication interface 2 (SCI2)
	MSTPB4*	
	MSTPB3*	
	MSTPB2*	
	MSTPB1*	
	MSTPB0*	
MSTPCRC	MSTPC7*	
	MSTPC6*	
	MSTPC5*	
	MSTPC4	PC break controller (PBC)
	MSTPC3	HCAN
	MSTPC2*	
	MSTPC1*	
	MSTPC0*	

Note: * MSTPA7 is a readable/writable bit with an initial value of 0.
MSTPA4, MSTPA2, MSTPA0, MSTPB4 to MSTPB0, MSTPC7 to MSTPC5
MSTPC2 to MSTPC0 are readable/writable bits with an initial value of 1 an
always be written with 1.

module stop mode. Consequently, if module stop mode is entered when an interrupt ha
requested, it will not be possible to clear the CPU interrupt source or the DTC activatio
Interrupts should therefore be disabled before entering module stop mode.

Writing to MSTPCR: MSTPCR should only be written to by the CPU.

21A.6 Software Standby Mode

21A.6.1 Software Standby Mode

A transition is made to software standby mode when the SLEEP instruction is executed
SBYCR SSBY bit = 1. In this mode, the CPU, on-chip supporting modules, and oscilla
However, the contents of the CPU's internal registers, RAM data, and the states of on-c
supporting modules other than the SCI, A/D converter, HCAN and I/O ports, are retain
Whether the address bus and bus control signals are placed in the high-impedance state
the output state can be specified by the OPE bit in SBYCR.

In this mode the oscillator stops, and therefore power dissipation is significantly reduce

21A.6.2 Clearing Software Standby Mode

Software standby mode is cleared by an external interrupt (NMI pin, or pins $\overline{\text{IRQ0}}$ to $\overline{\text{IR}}$
means of the $\overline{\text{RES}}$ pin or $\overline{\text{STBY}}$ pin.

- Clearing with an interrupt

  When an NMI or IRQ0 to IRQ5 interrupt request signal is input, clock oscillation st
  after the elapse of the time set in bits STS2 to STS0 in SYSCR, stable clocks are su
  the entire chip, software standby mode is cleared, and interrupt exception handling

  When clearing software standby mode with an IRQ0 to IRQ5 interrupt, set the corre
  enable bit to 1 and ensure that no interrupt with a higher priority than interrupts IRQ
  is generated. Software standby mode cannot be cleared if the interrupt has been mas
  CPU side or has been designated as a DTC activation source.

21A.6.3 Setting Oscillation Stabilization Time after Clearing Software Standby

Bits STS2 to STS0 in SBYCR should be set as described below.

Using a Crystal Oscillator: Set bits STS2 to STS0 so that the standby time is at leas
oscillation stabilization time).

Table 21A.4 shows the standby times for different operating frequencies and settings
to STS0.

Table 21A.4 Oscillation Stabilization Time Settings

STS2	STS1	STS0	Standby Time	20 MHz	16 MHz	12 MHz	10 MHz	8 MHz	6 MHz
0	0	0	8192 states	0.41	0.51	0.68	0.8	1.0	1.3
		1	16384 states	0.82	1.0	1.3	1.6	2.0	2.7
	1	0	32768 states	1.6	2.0	2.7	3.3	4.1	5.5
		1	65536 states	3.3	4.1	5.5	6.6	8.2	10.9
1	0	0	131072 states	6.6	8.2	10.9	13.1	16.4	21.8
		1	262144 states	13.1	16.4	21.8	26.2	32.8	43.6
	1	0	Reserved	—	—	—	—	—	—
		1	16 states*	0.8	1.0	1.3	1.6	2.0	1.7

: Recommended time setting

Note: * Do not use this setting.

Using an External Clock: It is necessary to allow time for the PLL circuit to stabiliz
the standby time should be set to a value of 2 ms or greater.

Software standby mode is then cleared at the rising edge on the NMI pin.

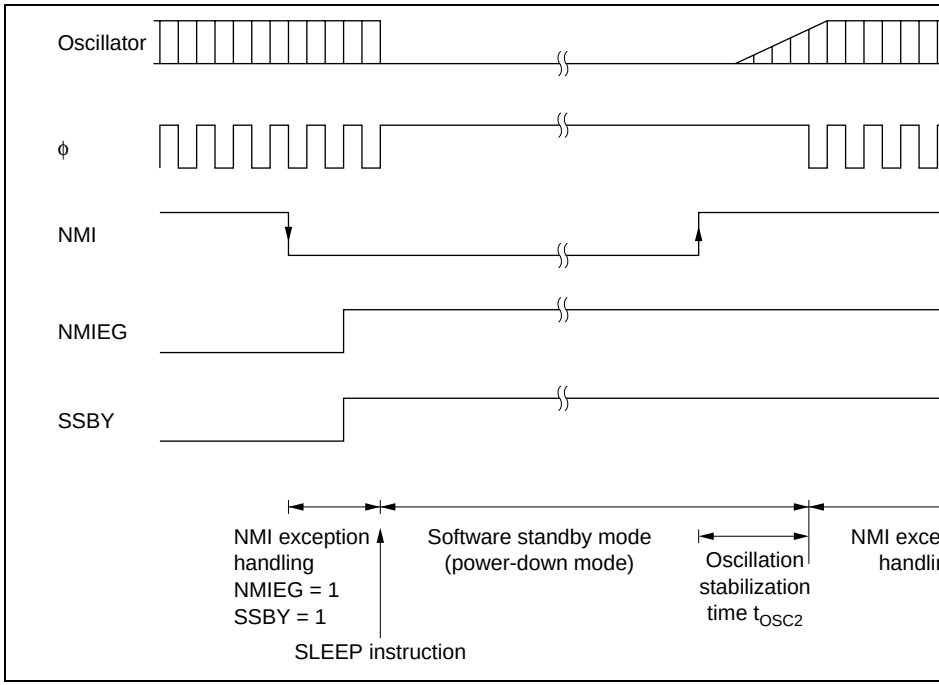


Figure 21A.3 Software Standby Mode Application Example

Write Data Buffer Function: The write data buffer function and software standby m
be used at the same time. When the write data buffer function is used, the WDBE bit i
should be cleared to 0 to cancel the write data buffer function before entering software
mode. Also check that external writes have finished, by reading external addresses, et
executing a SLEEP instruction to enter software standby mode. See section 7.7, Write
Function, for details of the write data buffer function.

21A.7 Hardware Standby Mode

21A.7.1 Hardware Standby Mode

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode fro

In hardware standby mode, all functions enter the reset state and stop operation, result
significant reduction in power dissipation. As long as the prescribed voltage is supplie
RAM data is retained. I/O ports are set to the high-impedance state.

In order to retain on-chip RAM data, the RAME bit in SYSCR should be cleared to 0
driving the $\overline{\text{STBY}}$ pin low.

Do not change the state of the mode pins (MD2 to MD0) while the H8S/2623 Group i
hardware standby mode.

Hardware standby mode is cleared by means of the $\overline{\text{STBY}}$ pin and the $\overline{\text{RES}}$ pin. When
pin is driven high while the $\overline{\text{RES}}$ pin is low, the reset state is set and clock oscillation
Ensure that the $\overline{\text{RES}}$ pin is held low until the clock oscillator stabilizes (at least 8 ms—
oscillation stabilization time—when using a crystal oscillator). When the $\overline{\text{RES}}$ pin is s
driven high, a transition is made to the program execution state via the reset exception
state.

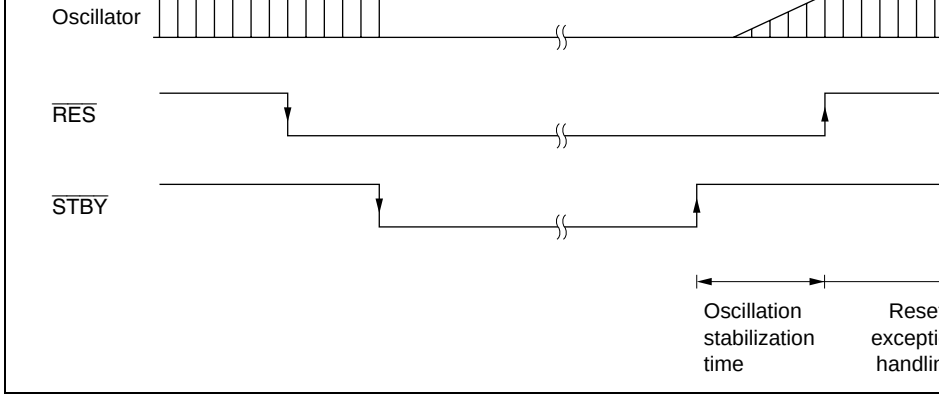


Figure 21A.4 Hardware Standby Mode Timing

21A.8 ϕ Clock Output Disabling Function

Output of the ϕ clock can be controlled by means of the PSTOP bit in SCKCR, and DD corresponding port. When the PSTOP bit is set to 1, the ϕ clock stops at the end of the b and ϕ output goes high. ϕ clock output is enabled when the PSTOP bit is cleared to 0. W for the corresponding port is cleared to 0, ϕ clock output is disabled and input port mod Table 21A.5 shows the state of the ϕ pin in each processing state.

Table 21A.5 ϕ Pin State in Each Processing State

DDR	0	1	1
PSTOP	—	0	1
Hardware standby mode	High impedance	High impedance	High impe
Software standby	High impedance	Fixed high	Fixed hig
Sleep mode	High impedance	ϕ output	Fixed hig
High-speed mode, medium-speed mode	High impedance	ϕ output	Fixed hig

Low-power operation can be achieved by individually controlling the CPU, on-chip su
modules, and so on.

The H8S/2626 operating modes are as follows:

(1) High-speed mode
(2) Medium-speed mode
(3) Subactive mode*
(4) Sleep mode
(5) Subsleep mode*
(6) Watch mode*
(7) Module stop mode
(8) Software standby mode
(9) Hardware standby mode

(2) to (9) are power-down modes. Sleep mode and sub-sleep mode are CPU states, me
mode is a CPU and bus master state, sub-active mode is a CPU and bus master and int
peripheral function state, and module stop mode is an internal peripheral function (inc
masters other than the CPU) state. Some of these states can be combined.

After a reset, the LSI is in high-speed mode with modules other than the DTC in modu
mode.

Note: * Subclock functions are available in the H8S/2626 Group.
See section 20.7, Subclock Oscillator (H8S/2626 Group Only), for the met
pins OSC1 and OSC2 when not used.

Table 21B.1 shows the internal state of the LSI in the respective modes. Table 21B.2 s
conditions for shifting between the power-down modes.

Figure 21B.1 is a mode transition diagram.

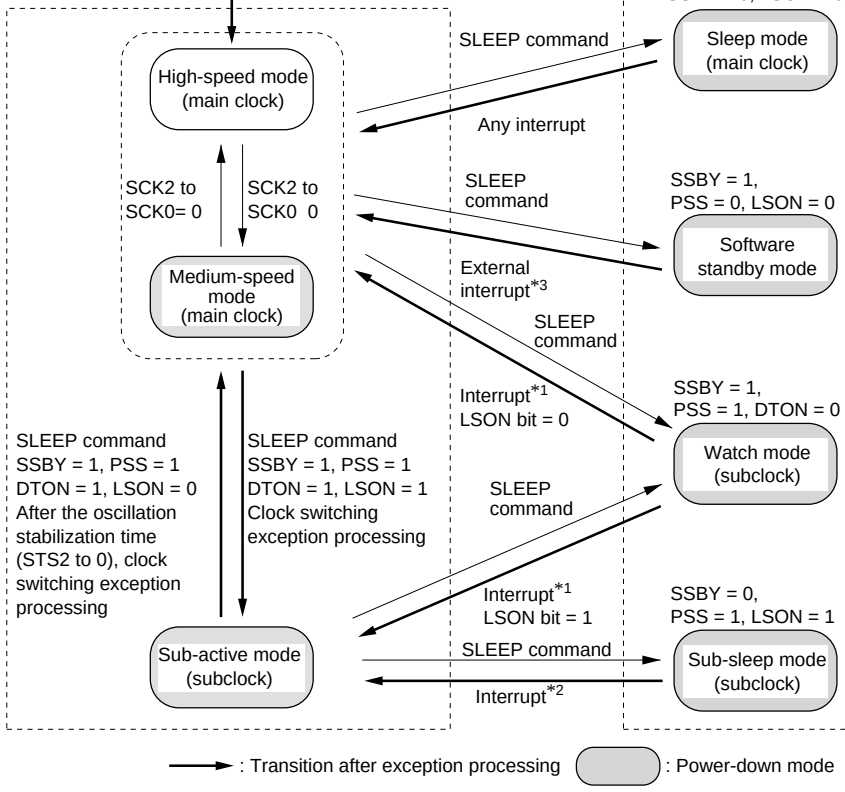
			operation		speed operation				
External interrupts	NMI	Function-ing	Function-ing	Function-ing	Function-ing	Function-ing	Function-ing	Function-ing	Function-ing
	IRQ0–IRQ5								
Peripheral functions	WDT1	Function-ing	Function-ing	Function-ing	—	Subclock operation	Subclock operation	Subclock operation	Halted (retained)
	WDT0	Function-ing	Function-ing	Function-ing	—	Halted (retained)	Subclock operation	Subclock operation	Halted (retained)
	DTC	Function-ing	Medium-speed operation	Function-ing	Halted (retained)	Halted (retained)	Halted (retained)	Halted (retained)	Halted (retained)
	TPU	Function-ing	Function-ing (PBC medium-speed operation)	Function-ing	Halted (retained)	Halted (retained)	Halted (retained) (PBC subclock operation)	Halted (retained)	Halted (retained)
	PBC								
	PPG								
	D/A2, 3								
	SCI0	Function-ing	Function-ing	Function-ing	Halted (reset)	Halted (reset)	Halted (reset)	Halted (reset)	Halted (reset)
	SCI1								
	SCI2								
	PWM								
	A/D								
	RAM	Function-ing	Function-ing	Function-ing (DTC)	Function-ing	Retained	Function-ing	Retained	Retained
	I/O	Function-ing	Function-ing	Function-ing	Function-ing	Retained	Function-ing	Retained	Retained
	HCAN	Function-ing	Function-ing*	Function-ing	Halted (reset)	Halted (reset)	Halted (reset)	Halted (reset)	Halted (reset)

Notes: “Halted (retained)” means that internal register values are retained. The internal
“operation suspended.”

“Halted (reset)” means that internal register values and internal states are initializ

In module stop mode, only modules for which a stop setting has been made are
(reset or retained).

* Note, however, that registers cannot be read or written to.



Notes: When a transition is made between modes by means of an interrupt, the transition cannot be on interrupt source generation alone. Ensure that interrupt handling is performed after accep interrupt request.
From any state except hardware standby mode, a transition to the reset state occurs when $\overline{R}$ driven Low.
From any state, a transition to hardware standby mode occurs when $\overline{STBY}$ is driven low.
Always select high-speed mode before making a transition to watch mode or sub-active mod

1. NMI, IRQ0 to IRQ5, and WDT1 interrupts
2. NMI, IRQ0 to IRQ5, WDT0 interrupts, and WDT1 interrupt.
3. NMI and IRQ0 to IRQ5

Figure 21B.1 Mode Transition Diagram

	1	0	1	*	—	—
	1	1	0	0	Watch	High-speed
	1	1	1	0	Watch	Sub-active
	1	1	0	1	—	—
	1	1	1	1	Sub-active	—
Sub-active	0	0	*	*	—	—
	0	1	0	*	—	—
	0	1	1	*	Sub-sleep	Sub-active
	1	0	*	*	—	—
	1	1	0	0	Watch	High-speed
	1	1	1	0	Watch	Sub-active
	1	1	0	1	High-speed	—
	1	1	1	1	—	—

Legend:

—: Do not set.

*: Don’t care

System clock control register	SCKCR	R/W	H'00	H
Low-power control register	LPWRCR	R/W	H'00	H
Timer control/status register	TCSR	R/W	H'00	H
Module stop control register A, B, C	MSTPCRA	R/W	H'3F	H
	MSTPCRB	R/W	H'FF	H
	MSTPCRC	R/W	H'FF	H

Note: * Lower 16 bits of the address.

21B.2 Register Descriptions

21B.2.1 Standby Control Register (SBYCR)

Bit	:	7	6	5	4	3	2	1
		SSBY	STS2	STS1	STS0	OPE	—	—
Initial value	:	0	0	0	0	1	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	—	—

SBYCR is an 8-bit readable/writable register that performs power-down mode control

SBYCR is initialized to H'08 by a reset and in hardware standby mode. It is not initial
software standby mode.

0	Shifts to sleep mode when the SLEEP instruction is executed in high-speed mode or medium-speed mode. Shifts to sub-sleep mode when the SLEEP instruction is executed in sub-active mode. (In
1	Shifts to software standby mode, sub-active mode, and watch mode when t instruction is executed in high-speed mode or medium-speed mode. Shifts to watch mode or high-speed mode when the SLEEP instruction is ex sub-active mode.

Bits 6 to 4—Standby Timer Select 2 to 0 (STS2 to STS0): These bits select the MCU for clock stabilization when shifting to high-speed mode or medium-speed mode by usi specific interrupt or command to cancel software standby mode, watch mode, or sub-ac With a quartz oscillator (table 21B.5), select a wait time of 8ms (oscillation stabilizatio more, depending on the operating frequency. With an external clock, there are no speci requirements.

Bit 6	Bit 5	Bit 4	
STS2	**STS1**	**STS0**	**Description**
0	0	0	Standby time = 8192 states (In
		1	Standby time = 16384 states
	1	0	Standby time = 32768 states
		1	Standby time = 65536 states
1	0	0	Standby time = 131072 states
		1	Standby time = 262144 states
	1	0	Reserved
		1	Standby time = 16 states

1	In software standby mode, watch mode, and when making a direct transiti output state of the address bus and bus control signals is retained. (I

Bits 2 to 0—Reserved: These bits always return 0 when read, and cannot be written t

21B.2.2 System Clock Control Register (SCKCR)

Bit	:	7	6	5	4	3	2	1
		PSTOP	—	—	—	STCS	SCK2	SCK1
Initial value	:	0	0	0	0	0	0	0
R/W	:	R/W	—	—	—	R/W	R/W	R/W

SCKCR is an 8-bit readable/writable register that performs ϕ clock output control and speed mode control.

SCKCR is initialized to H'00 by a reset and in hardware standby mode. It is not initial software standby mode.

Bit 7—ϕ Clock Output Disable (PSTOP): In combination with the DDR of the appl this bit controls ϕ output. See section 21B.12, ϕ Clock Output Disabling Function, for

	Description			
Bit 7 PSTOP	**High-Speed Mode, Medium-Speed Mode, Sub-Active Mode**	**Sleep Mode, Sub-Sleep Mode**	**Software Standby Mode, Watch Mode, Direct Transition**	**Hardw Stand**
0	ϕ output (initial value)	ϕ output	Fixed high	High i
1	Fixed high	Fixed high	Fixed high	High i

Bits 6 to 4—Reserved: These bits are always read as 0 and cannot be modified.

Bits 2 to 0—System clock select (SCK2 to SCK0): These bits select the bus master cl
high-speed mode, medium-speed mode, and sub-active mode.

Set SCK2 to SCK0 all to 0 when shifting to operation in watch mode or sub-active mod

Bit 2	Bit 1	Bit 0	
SCK2	**SCK1**	**SCK0**	**Description**
0	0	0	Bus master in high-speed mode (In
		1	Medium-speed clock is ϕ/2
	1	0	Medium-speed clock is ϕ/4
		1	Medium-speed clock is ϕ/8
1	0	0	Medium-speed clock is ϕ/16
		1	Medium-speed clock is ϕ/32
	1	—	—

21B.2.3 Low-Power Control Register (LPWRCR)

Bit :	7	6	5	4	3	2	1
	DTON	LSON	NESEL	SUBSTP	RFCUT	—	STC1
Initial value :	0	0	0	0	0	0	0
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The LPWRCR is an 8-bit read/write register that controls the low power dissipation mo

The LPWRCR is initialized to H'00 at a reset and when in hardware standby mode. It is
initialized in software standby mode. The following describes bits 7 to 2. For details of
see section 20.2.2, Low-Power Control Register (LPWRCR).

0	• When the SLEEP instruction is executed in high-speed mode or medium- mode, operation shifts to sleep mode, software standby mode, or watch m • When the SLEEP instruction is executed in sub-active mode, operation s to sub-sleep mode or watch mode. (I
1	• When the SLEEP instruction is executed in high-speed mode or medium- mode, operation shifts directly to sub-active mode*, or shifts to sleep mo software standby mode. • When the SLEEP instruction is executed in sub-active mode, operation s to high-speed mode, or shifts to sub-sleep mode.

Note: * Always set high-speed mode when shifting to watch mode or sub-active mo

Bit 6—Low-Speed ON Flag (LSON): When shifting to low power dissipation mode the SLEEP instruction, this bit specifies the operating mode, in combination with othe bits. This bit also controls whether to shift to high-speed mode or sub-active mode wh mode is cancelled.

Bit 6

LSON	Description
0	• When the SLEEP instruction is executed in high-speed mode or medium- mode, operation shifts to sleep mode, software standby mode, or watch m • When the SLEEP instruction is executed in sub-active mode, operation s watch mode or shifts directly to high-speed mode. • Operation shifts to high-speed mode when watch mode is cancelled. (
1	• When the SLEEP instruction is executed in high-speed mode, operation watch mode or sub-active mode. • When the SLEEP instruction is executed in sub-active mode, operation s sleep mode or watch mode. • Operation shifts to sub-active mode when watch mode is cancelled.

Note: * Always set high-speed mode when shifting to watch mode or sub-active mo

Bit 4—Subclock enable (SUBSTP): This bit enables/disables subclock generation.

Bit 4

SUBSTP	Description
0	Enables subclock generation (In
1	Disables subclock generation

Bit 3—Oscillation Circuit Feedback Resistance Control Bit (RFCUT): This bit turn internal feedback resistance of the main clock oscillation circuit ON/OFF.

Bit 3

RFCUT	Description
0	When the main clock is oscillating, sets the feedback resistance ON. When th clock is stopped, sets the feedback resistance OFF. (In
1	Sets the feedback resistance OFF.

Bit 2—Reserved: Only write 0 to this bit.

TCSR is an 8-bit read/write register that selects the clock input to WDT1 TCNT and t

Here, we describe bit 4. For details of the other bits in this register, see section 12.2.2, Control/Status Register (TCSR).

The TCSR is initialized to H'00 at a reset and when in hardware standby mode. It is no in software standby mode.

Bit 4—Prescaler select (PSS): This bit selects the clock source input to WDT1 TCNT

It also controls operation when shifting low power dissipation modes. The operating m selected after the SLEEP instruction is executed is determined in combination with oth bits.

For details, see the description for clock selection in section 12.2.2, Timer Control/Sta (TCSR), and this section.

Bit 4

PSS	Description
0	• TCNT counts the divided clock from the ϕ -based prescaler (PSM). • When the SLEEP instruction is executed in high-speed mode or medium- mode, operation shifts to sleep mode or software standby mode.
1	• TCNT counts the divided clock from the ϕsubclock-based prescaler (PSS • When the SLEEP instruction is executed in high-speed mode or medium- mode, operation shifts to sleep mode, watch mode*, or sub-active mode* • When the SLEEP instruction is executed in sub-active mode*, operation sub-sleep mode*, watch mode*, or high-speed mode.

Note: * Always set high-speed mode when shifting to watch mode or sub-active mo

MSTPCRB

Bit	:	7	6	5	4	3	2	1
		MSTPB7	MSTPB6	MSTPB5	MSTPB4	MSTPB3	MSTPB2	MSTPB
Initial value	:	1	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MSTPCRC

Bit	:	7	6	5	4	3	2	1
		MSTPC7	MSTPC6	MSTPC5	MSTPC4	MSTPC3	MSTPC2	MSTPC
Initial value	:	1	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MSTPCR, comprising three 8-bit readable/writable registers, performs module stop mo

MSTPCRA to MSTPCRC are initialized to H'3FFFFF by a reset and in hardware stand
They are not initialized in software standby mode.

**MSTPCRA/MSTPCRB/MSTPCRC Bits 7 to 0—Module Stop (MSTPA7 to MSTP
MSTPB7 to MSTPB0, MSTPC7 to MSTPC0):** These bits specify module stop mode
21B.4 for the method of selecting the on-chip peripheral functions.

MSTPCRA/MSTPCRB/ MSTPCRC Bits 7 to 0	
MSTPA7 to MSTPA0, MSTPB7 to MSTPB0, MSTPC7 to MSTPC0	**Description**
0	Module stop mode is cleared (initial value of MSTPA7 and MS
1	Module stop mode is set (initial value of MSTPA5–0, MSTPB7- MSTPC7–0)

the bus master operating clock. For example, if ϕ/4 is selected as the operating clock, memory is accessed in 4 states, and internal I/O registers in 8 states.

Medium-speed mode is cleared by clearing all of bits SCK2 to SCK0 to 0. A transition high-speed mode and medium-speed mode is cleared at the end of the current bus cyc

If a SLEEP instruction is executed when the SSBY bit in SBYCR is cleared to 0, and LPWRCR is cleared to 0, a transition is made to sleep mode. When sleep mode is clea interrupt, medium-speed mode is restored.

When the SLEEP instruction is executed with the SSBY bit = 1, LPWRCR LSON bit TCSR (WDT1) PSS bit = 0, operation shifts to the software standby mode. When soft standby mode is cleared by an external interrupt, medium-speed mode is restored.

When the $\overline{\text{RES}}$ pin is set low and medium-speed mode is cancelled, operation shifts to state. The same applies in the case of a reset caused by overflow of the watchdog time

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

Figure 21B.2 shows the timing for transition to and clearance of medium-speed mode.

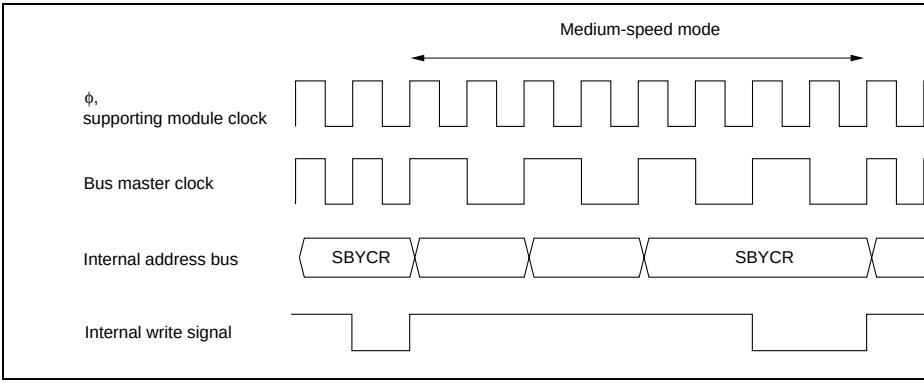


Figure 21B.2 Medium-Speed Mode Transition and Clearance Timin

21B.4.2 Exiting Sleep Mode

Sleep mode is exited by any interrupt, or signals at the $\overline{\text{RES}}$, or $\overline{\text{STBY}}$ pins.

Exiting Sleep Mode by Interrupts: When an interrupt occurs, sleep mode is exited an exception processing starts. Sleep mode is not exited if the interrupt is disabled, or inte than NMI are masked by the CPU.

Exiting Sleep Mode by $\overline{\text{RES}}$ pin: Setting the $\overline{\text{RES}}$ pin level Low selects the reset state stipulated reset input duration, driving the $\overline{\text{RES}}$ pin High starts the CPU performing res exception processing.

Exiting Sleep Mode by $\overline{\text{STBY}}$ Pin: When the $\overline{\text{STBY}}$ pin level is driven Low, a transiti to hardware standby mode.

21B.5 Module Stop Mode

21B.5.1 Module Stop Mode

Module stop mode can be set for individual on-chip supporting modules.

When the corresponding MSTP bit in MSTPCR is set to 1, module operation stops at th the bus cycle and a transition is made to module stop mode. The CPU continues operati independently.

Table 21B.4 shows MSTP bits and the corresponding on-chip supporting modules.

When the corresponding MSTP bit is cleared to 0, module stop mode is cleared and th starts operating at the end of the bus cycle. In module stop mode, the internal states of m other than the SCI, A/D converter and HCAN are retained.

After reset clearance, all modules other than DTC are in module stop mode.

	MSTPA5	16-bit timer pulse unit (TPU)
	MSTPA4*	
	MSTPA3	Programmable pulse generator (PPG)
	MSTPA2*	
	MSTPA1	A/D converter
	MSTPA0*	
MSTPCRB	MSTPB7	Serial communication interface 0 (SCI0)
	MSTPB6	Serial communication interface 1 (SCI1)
	MSTPB5	Serial communication interface 2 (SCI2)
	MSTPB4*	
	MSTPB3*	
	MSTPB2*	
	MSTPB1*	
	MSTPB0*	
MSTPCRC	MSTPC7*	
	MSTPC6*	
	MSTPC5	D/A converter (channels 2, 3)
	MSTPC4	PC break controller (PBC)
	MSTPC3	HCAN
	MSTPC2*	
	MSTPC1*	
	MSTPC0*	

Note: * MSTPA7 is a readable/writable bit with an initial value of 0.

MSTPA4, MSTPA2, MSTPA0, MSTPB4 to MSTPB0, MSTPC7 to MSTPC4
MSTPC2 to MSTPC0 are readable/writable bits with an initial value of 1 an
always be written with 1.

module stop mode. Consequently, if module stop mode is entered when an interrupt ha
requested, it will not be possible to clear the CPU interrupt source or the DTC activatio
Interrupts should therefore be disabled before entering module stop mode.

Writing to MSTPCR: MSTPCR should only be written to by the CPU.

21B.6 Software Standby Mode

21B.6.1 Software Standby Mode

A transition is made to software standby mode when the SLEEP instruction is executed
SBYCR SSBY bit = 1 and the LPWRCR LSON bit = 0, and the TCSR (WDT1) PSS bi
this mode, the CPU, on-chip supporting modules, and oscillator all stop. However, the
the CPU's internal registers, RAM data, and the states of on-chip supporting modules o
the SCI, A/D converter, HCAN and I/O ports, are retained. Whether the address bus an
control signals are placed in the high-impedance state or retain the output state can be s
the OPE bit in SBYCR.

In this mode the oscillator stops, and therefore power dissipation is significantly reduce

21B.6.2 Clearing Software Standby Mode

Software standby mode is cleared by an external interrupt (NMI pin, or pins $\overline{\text{IRQ0}}$ to $\overline{\text{IR}}$
means of the $\overline{\text{RES}}$ pin or $\overline{\text{STBY}}$ pin.

- Clearing with an interrupt

  When an NMI or IRQ0 to IRQ5 interrupt request signal is input, clock oscillation st
  after the elapse of the time set in bits STS2 to STS0 in SYSCR, stable clocks are su
  the entire chip, software standby mode is cleared, and interrupt exception handling

  When clearing software standby mode with an IRQ0 to IRQ5 interrupt, set the corre
  enable bit to 1 and ensure that no interrupt with a higher priority than interrupts IRQ

- Clearing with the $\overline{\text{STBY}}$ pin
  When the $\overline{\text{STBY}}$ pin is driven Low, a transition is made to hardware standby mode

21B.6.3 Setting Oscillation Stabilization Time after Clearing Software Standby

Bits STS2 to STS0 in SBYCR should be set as described below.

Using a Crystal Oscillator: Set bits STS2 to STS0 so that the standby time is at leas
oscillation stabilization time).

Table 21B.5 shows the standby times for different operating frequencies and settings
to STS0.

Table 21B.5 Oscillation Stabilization Time Settings

STS2	STS1	STS0	Standby Time	20 MHz	16 MHz	12 MHz	10 MHz	8 MHz	6 MHz
0	0	0	8192 states	0.41	0.51	0.68	0.8	1.0	1.3
		1	16384 states	0.82	1.0	1.3	1.6	2.0	2.7
	1	0	32768 states	1.6	2.0	2.7	3.3	4.1	5.5
		1	65536 states	3.3	4.1	5.5	6.6	8.2	10.9
1	0	0	131072 states	6.6	8.2	10.9	13.1	16.4	21.8
		1	262144 states	13.1	16.4	21.8	26.2	32.8	43.6
	1	0	Reserved	—	—	—	—	—	—
		1	16 states*	0.8	1.0	1.3	1.6	2.0	1.7

: Recommended time setting

Note: * Do not use this setting.

Using an External Clock: It is necessary to allow time for the PLL circuit to stabiliz
the standby time should be set to a value of 2 ms or greater.

Software standby mode is then cleared at the rising edge on the NMI pin.

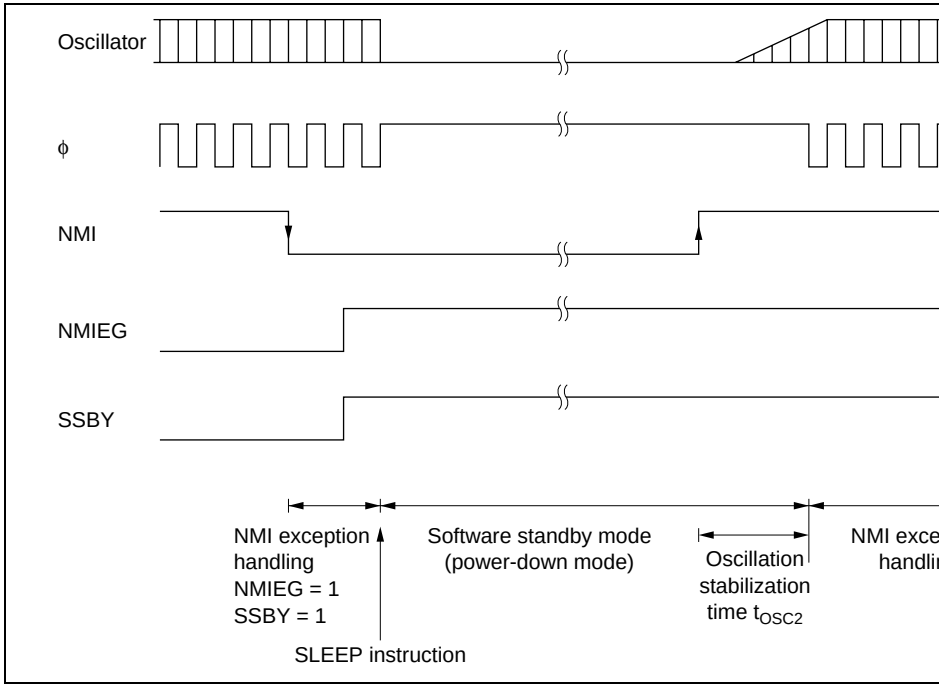


Figure 21B.3 Software Standby Mode Application Example

Write Data Buffer Function: The write data buffer function and software standby m
be used at the same time. When the write data buffer function is used, the WDBE bit i
should be cleared to 0 to cancel the write data buffer function before entering software
mode. Also check that external writes have finished, by reading external addresses, et
executing a SLEEP instruction to enter software standby mode. See section 7.7, Write
Function, for details of the write data buffer function.

21B.7 Hardware Standby Mode

21B.7.1 Hardware Standby Mode

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode fro

In hardware standby mode, all functions enter the reset state and stop operation, result
significant reduction in power dissipation. As long as the prescribed voltage is supplie
RAM data is retained. I/O ports are set to the high-impedance state.

In order to retain on-chip RAM data, the RAME bit in SYSCR should be cleared to 0
driving the $\overline{\text{STBY}}$ pin low.

Do not change the state of the mode pins (MD2 to MD0) while the H8S/2626 Group i
hardware standby mode.

Hardware standby mode is cleared by means of the $\overline{\text{STBY}}$ pin and the $\overline{\text{RES}}$ pin. When
pin is driven high while the $\overline{\text{RES}}$ pin is low, the reset state is set and clock oscillation
Ensure that the $\overline{\text{RES}}$ pin is held low until the clock oscillator stabilizes (at least 8 ms—
oscillation stabilization time—when using a crystal oscillator). When the $\overline{\text{RES}}$ pin is s
driven high, a transition is made to the program execution state via the reset exception
state.

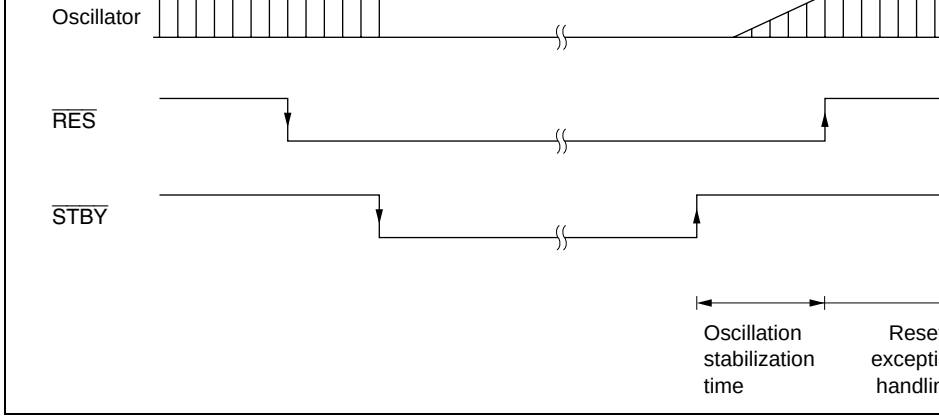


Figure 21B.4 Hardware Standby Mode Timing

21B.8 Watch Mode

21B.8.1 Watch Mode

CPU operation makes a transition to watch mode when the SLEEP instruction is execu
speed mode or sub-active mode with SBYCR SSBY=1, LPWRCR DTON = 0, and TCS
(WDT1) PSS = 1.

In watch mode, the CPU is stopped and supporting modules other than WDT1 are also
The contents of the CPU is internal registers, the data in internal RAM, and the statuses
internal supporting modules (excluding the SCI, ADC, HCAN) and I/O ports are retain

= 1. When a transition is made to high-speed mode, a stable clock is supplied to all LS
and interrupt exception processing starts after the time set in SBYCR STS2 to STS0 h
In the case of IRQ0 to IRQ5 interrupts, no transition is made from watch mode if the
corresponding enable bit has been cleared to 0, and, in the case of interrupts from the
supporting modules, the interrupt enable register has been set to disable the reception
interrupt, or is masked by the CPU.

See section 21B.6.3, Setting Oscillation Stabilization Time after Clearing Software St
for how to set the oscillation stabilization time when making a transition from watch m
high-speed mode.

(2) Exiting Watch Mode by $\overline{\text{RES}}$ pins

For exiting watch mode by the $\overline{\text{RES}}$ pins, see, Clearing with the $\overline{\text{RES}}$ pins in section 2
Clearing Software Standby Mode.

(3) Exiting Watch Mode by $\overline{\text{STBY}}$ pin

When the $\overline{\text{STBY}}$ pin level is driven Low, a transition is made to hardware standby mo

21B.8.3 Notes

(1) I/O Port Status

The status of the I/O ports is retained in watch mode. Also, when the OPE bit is set to
address bus and bus control signals continue to be output. Therefore, when a High lev
the current consumption is not diminished by the amount of current to support the Hig
output.

(2) Current Consumption when Waiting for Oscillation Stabilization

The current consumption increases during stabilization of oscillation.

also stopped. The contents of the CPUís internal registers, the data in internal RAM, an
statuses of the internal supporting modules (excluding the SCI, ADC, HCAN) and I/O
retained.

21B.9.2 Exiting Sub-Sleep Mode

Sub-sleep mode is exited by an interrupt (interrupts from internal supporting modules,
$\overline{\text{IRQ0}}$ to $\overline{\text{IRQ5}}$), or signals at the $\overline{\text{RES}}$ or $\overline{\text{STBY}}$ pins.

(1) Exiting Sub-Sleep Mode by Interrupts

When an interrupt occurs, sub-sleep mode is exited and interrupt exception processing

In the case of IRQ0 to IRQ5 interrupts, sub-sleep mode is not cancelled if the corresp
enable bit has been cleared to 0, and, in the case of interrupts from the internal supporti
modules, the interrupt enable register has been set to disable the reception of that interr
masked by the CPU.

(2) Exiting Sub-Sleep Mode by $\overline{\text{RES}}$

For exiting sub-sleep mode by the $\overline{\text{RES}}$ pins, see, Clearing with the $\overline{\text{RES}}$ pins in section
Clearing Software Standby Mode.

(3) Exiting Sub-Sleep Mode by $\overline{\text{STBY}}$ Pin

When the $\overline{\text{STBY}}$ pin level is driven Low, a transition is made to hardware standby mod

is made to sub-active mode.

In sub-active mode, the CPU operates at low speed on the subclock, and the program i
step by step. Supporting modules other than WDT0, and WDT1 are also stopped.

When operating the CPU in sub-active mode, the SCKCR SCK2 to SCK0 bits must b

21B.10.2 Exiting Sub-Active Mode

Sub-active mode is exited by the SLEEP instruction or the $\overline{\text{RES}}$ or $\overline{\text{STBY}}$ pins.

(1) Exiting Sub-Active Mode by SLEEP Instruction

When the SLEEP instruction is executed with the SBYCR SSBY bit = 1, LPWRCR D
0, and TCSR (WDT1) PSS bit = 1, the CPU exits sub-active mode and a transition is m
watch mode. When the SLEEP instruction is executed with the SBYCR SSBY bit = 0
LSON bit = 1, and TCSR (WDT1) PSS bit = 1, a transition is made to sub-sleep mode
when the SLEEP instruction is executed with the SBYCR SSBY bit = 1, LPWRCR D
LSON bit = 0, and TCSR (WDT1) PSS bit = 1, a direct transition is made to high-spe
(SCK0 to SCK2 all 0).

See section 21B.11, Direct Transitions, for details of direct transitions.

(2) Exiting Sub-Active Mode by $\overline{\text{RES}}$ Pins

For exiting sub-active mode by the $\overline{\text{RES}}$ pins, see, Clearing with the $\overline{\text{RES}}$ pins in secti
Clearing Software Standby Mode.

(3) Exiting Sub-Active Mode by $\overline{\text{STBY}}$ Pin

When the $\overline{\text{STBY}}$ pin level is driven Low, a transition is made to hardware standby mo

transition interrupt exception processing starts.

(1) Direct Transitions from High-Speed Mode to Sub-Active Mode

Execute the SLEEP instruction in high-speed mode when the SBYCR SSBY bit = 1, LI
LSON bit = 1, and DTON bit = 1, and TSCR (WDT1) PSS bit = 1 to make a transition
active mode.

(2) Direct Transitions from Sub-Active Mode to High-Speed Mode

Execute the SLEEP instruction in sub-active mode when the SBYCR SSBY bit = 1, LP
LSON bit = 0, and DTON bit = 1, and TSCR (WDT1) PSS bit = 1 to make a direct tran
high-speed mode after the time set in SBYCR STS2 to STS0 has elapsed.

21B.12 φ Clock Output Disabling Function

Output of the φ clock can be controlled by means of the PSTOP bit in SCKCR, and DD
corresponding port. When the PSTOP bit is set to 1, the φ clock stops at the end of the l
and φ output goes high. φ clock output is enabled when the PSTOP bit is cleared to 0. W
for the corresponding port is cleared to 0, φ clock output is disabled and input port mod
Table 21B.6 shows the state of the φ pin in each processing state.

Table 21B.6 φ Pin State in Each Processing State

DDR	0	1	1
PSTOP	—	0	1
Hardware standby mode	High impedance	High impedance	High impe
Software standby mode, watch mode, and direct transition	High impedance	Fixed high	Fixed hig
Sleep mode and subsleep mode	High impedance	φ output	Fixed hig
High-speed mode, medium-speed mode, and subactive mode	High impedance	φ output	Fixed hig

module stop mode has been cleared and high-speed mode or medium-speed mode
entered.

2. The on-chip peripheral modules (DTC and TPU) which halt operation in sub-activ
cannot clear an interrupt in sub-active mode. Therefore, if a transition is made to s
mode while an interrupt is requested, the CPU interrupt source cannot be cleared.
interrupts of each on-chip peripheral module before executing a SLEEP instructio
sub-active mode or watch mode.

Item	Symbol	Value
Power supply voltage	V_{CC} $PLLV_{CC}$	–0.3 to +4.3
	$PV_{CC1–4}$	–0.3 to +7.0
Input voltage (XTAL, EXTAL, OSC1, OSC2)	V_{in}	–0.3 to V_{CC} +0.3
Input voltage (ports 4 and 9)	V_{in}	–0.3 to AV_{CC} +0.3
Input voltage (except ports 4 and 9)	V_{in}	–0.3 to PV_{CC} +0.3
Reference voltage	V_{ref}	–0.3 to AV_{CC} +0.3
Analog power supply voltage	AV_{CC}	–0.3 to +7.0
Analog input voltage	V_{AN}	–0.3 to AV_{CC} +0.3
Operating temperature	T_{opr}	Regular specifications: –20 to +75
		Wide-range specifications: –40 to +8
Storage temperature	T_{stg}	–55 to +125

Caution: Permanent damage to the chip may result if absolute maximum rating are ex

Item		Symbol	Min	Typ	Max	Unit	Te Co
Schmitt trigger input voltage	$\overline{IRQ0}$ to $\overline{IRQ5}$	V_T^-	$PV_{CC} \times 0.2$	—	—	V	
		V_T^+	—	—	$PV_{CC} \times 0.7$	V	
		$V_T^+ - V_T^-$	$PV_{CC} \times 0.05$	—	—	V	
Input high voltage	$\overline{RES}$, $\overline{STBY}$, NMI, MD2 to MD0, FWE	V_{IH}	$PV_{CC} \times 0.9$	—	PV_{CC} + 0.3	V	
	EXTAL, OSC1		$V_{CC} \times 0.7$	—	V_{CC} + 0.3	V	
	Ports 1, A to F, HRxD		$PV_{CC} \times 0.7$	—	PV_{CC} + 0.3	V	
	Port 4 and 9		$AV_{CC} \times 0.7$	—	AV_{CC} + 0.3	V	
Input low voltage	$\overline{RES}$, $\overline{STBY}$, NMI, MD2 to MD0, FWE	V_{IL}	–0.3	—	$PV_{CC} \times 0.1$	V	
	EXTAL, OSC1		–0.3	—	$V_{CC} \times 0.2$	V	
	Ports 1, A to F, HRxD		–0.3	—	$PV_{CC} \times 0.2$	V	
	Ports 4 and 9		–0.3	—	$AV_{CC} \times 0.2$	V	
Output high voltage	All output pins	V_{OH}	PV_{CC} – 0.5	—	—	V	I_{OH}
			PV_{CC} – 1.0	—	—	V	I_{OH}
Output low voltage	All output pins	V_{OL}	—	—	0.4	V	I_{OL}
Input leakage current	$\overline{RES}$	$\lvert I_{in} \rvert$	—	—	1.0	µA	V_{in} P
	$\overline{STBY}$, NMI, HRxD, MD2 to MD0, FWE		—	—	1.0	µA	
	Ports 4 and 9		—	—	1.0	µA	V_{in} A

capacitance		C_{in}					
	NMI		—	—	30	pF	f
	All input pins except $\overline{RES}$ and NMI		—	—	15	pF	
Current dissipation*2	Normal operation	I_{CC}*4	—	55 V_{CC} = 3.3 V	65 V_{CC} = 3.6 V	mA	f
	Sleep mode		—	40 V_{CC} = 3.3 V	50 V_{CC} = 3.6 V	mA	f
	All modules stopped		—	40	—	mA	f V (v
	Medium-speed mode (ϕ/32)		—	30	—	mA	f V (v
	Subactive mode		—	90 V_{CC} = 3.3 V	200	µA	U k n
	Subsleep mode		—	60 V_{CC} = 3.3 V	120	µA	U k n
	Watch mode		—	12 V_{CC} = 3.3 V	30	µA	U k n
	Standby mode*3		—	2.0	5.0	µA	T
			—	—	20	µA	5

Analog power supply current	During A/D and D/A conversion	AI_{CC}	—	1.0	2.0	mA	AV
	Idle		—	—	5.0	µA	
Reference power supply current	During A/D and D/A conversion	AI_{CC}	—	2.5	4.0	mA	V_{re}
	Idle		—	—	5.0	µA	
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Notes: 1. If the A/D and D/A converter is not used, do not leave the AV_{cc}, V_{ref}, and AV open. Apply a voltage between 4.5 V and 5.5 V to the AV_{CC} and V_{ref} pins by them to PV_{CC}, for instance. Set $V_{ref} \leq AV_{CC}$.

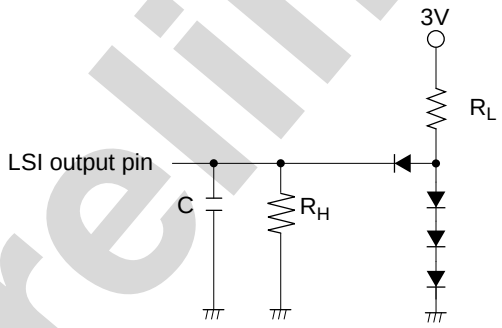
2. Current dissipation values are for $V_{IH} = V_{CC}$ (EXTAL, OSC1), AV_{CC} (ports 4 an PV_{CC} (other), and $V_{IL} = 0$ V, with all output pins unloaded and the on-chip MO transistors in the off state.
3. The values are for $V_{RAM} \leq PV_{CC} < 3.0$ V, V_{IH} min = $V_{CC} – 0.1$ V, and V_{IL} max = 0
4. I_{CC} depends on V_{CC} and f as follows:
   I_{CC} max = 8.0 (mA) + 0.8 (mA/(MHz × V)) × V_{CC} × f (normal operation)
   I_{CC} max = 8.0 (mA) + 0.58 (mA/(MHz × V)) × V_{CC} × f (sleep mode)
5. Applies to the mask ROM version only.

Permissible output low current (total)	Total of all output pins	PV_{CC} = 4.5 V to 5.5 V	$\sum I_{OL}$	—	—	1
Permissible output high current (per pin)	All output pins	PV_{CC} = 4.5 V to 5.5 V	$-I_{OH}$	—	—	2
Permissible output high current (total)	Total of all output pins	PV_{CC} = 4.5 V to 5.5 V	$\sum -I_{OH}$	—	—	3

Note: * To protect chip reliability, do not exceed the output current values in table 2

22.3 AC Characteristics

Figure 22.1 show, the test conditions for the AC characteristics.



C = 50 pF: Ports 10 to 13, A to F
(In case of expansion bus control signal output pin setting)
C = 30 pF: All ports
R_L = 2.4 kΩ
R_H = 12 kΩ
Input/output timing measurement levels
- Low level : 0.8 V
- High level : 2.0 V

Figure 22.1 Output Load Circuit

Item	Symbol	Min	Max	Unit	Test Co
Clock cycle time	t_{cyc}	50	250	ns	Figure 2
Clock high pulse width	t_{CH}	15	—	ns	
Clock low pulse width	t_{CL}	15	—	ns	
Clock rise time	t_{Cr}	—	5	ns	
Clock fall time	t_{Cf}	—	5	ns	
Oscillation stabilization time at reset (crystal)	t_{OSC1}	20	—	ms	Figure 2
Oscillation stabilization time in software standby (crystal)	t_{OSC2}	8	—	ms	Figure 2 Figure 2
External clock output stabilization delay time	t_{DEXT}	2	—	ms	Figure 2
32-kHz clock oscillation settling time	t_{OSC3}	—	2	s	
Sub clock oscillator frequency	f_{SUB}	32.768	—	kHz	
Sub clock (ϕ_{SUB}) cycle time	t_{SUB}	30.5	—	µs	

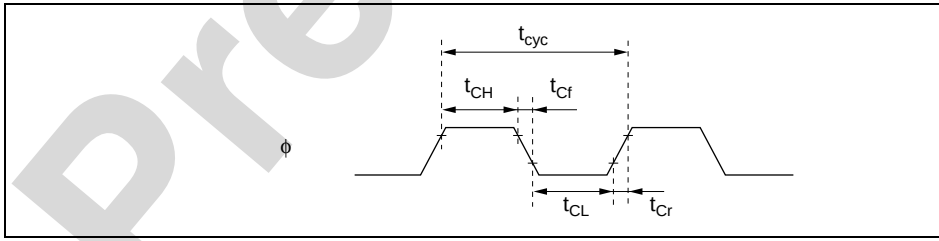


Figure 22.2 System Clock Timing

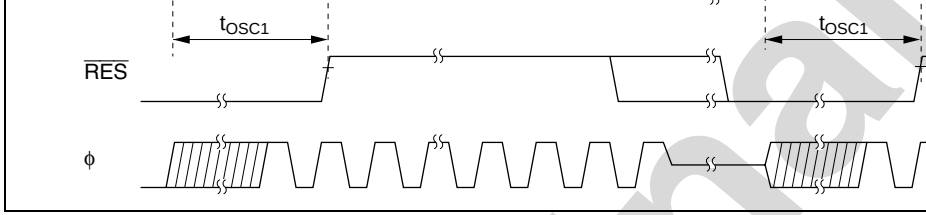


Figure 22.3 Oscillation Stabilization Timing

22.3.2 Control Signal Timing

Table 22.5 lists the control signal timing.

Table 22.5 Control Signal Timing

Conditions: V_{CC} = $PLLV_{CC}$ = 3.0 V to 3.6 V, PV_{CC} = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5
V_{ref} = 4.5 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 32.768 kHz, 4 to 20 MHz, T_a
+75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifi

Item	Symbol	Min	Max	Unit	Test
$\overline{RES}$ setup time	t_{RESS}	200	—	ns	Figur
$\overline{RES}$ pulse width	t_{RESW}	20	—	t_{cyc}	
NMI setup time	t_{NMIS}	150	—	ns	Figur
NMI hold time	t_{NMIH}	10	—		
NMI pulse width (exiting software standby mode)	t_{NMIW}	200	—	ns	
$\overline{IRQ}$ setup time	t_{IRQS}	150	—	ns	
$\overline{IRQ}$ hold time	t_{IRQH}	10	—	ns	
$\overline{IRQ}$ pulse width (exiting software standby mode)	t_{IRQW}	200	—	ns	

Figure 22.4 Reset Input Timing

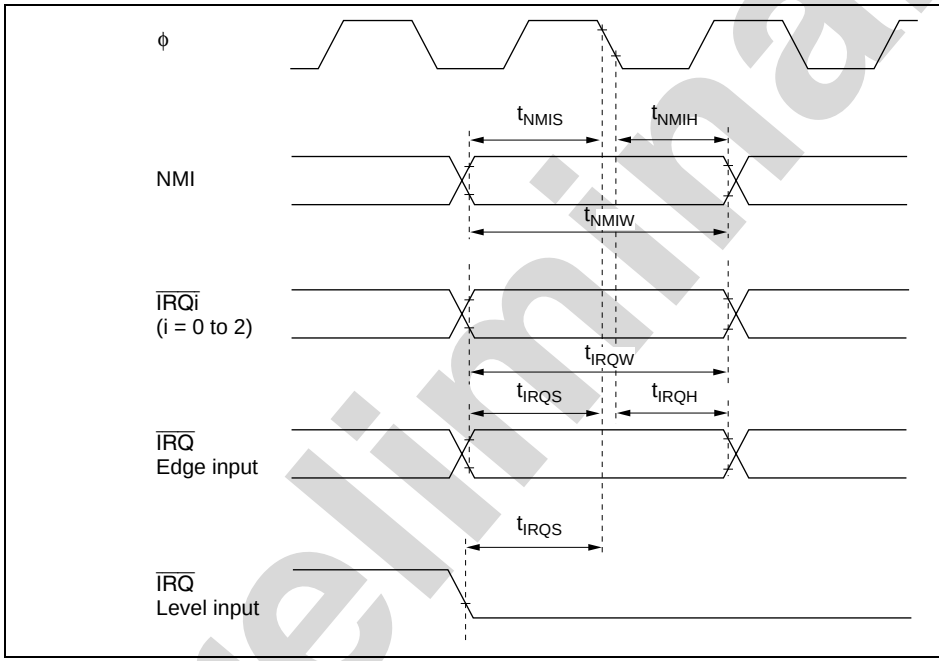


Figure 22.5 Interrupt Input Timing

Item	Symbol	Min	Max	Unit	Test
Address delay time	t_{AD}	—	25	ns	Figu
Address setup time	t_{AS}	$0.5 \times t_{cyc} - 20$	—	ns	Figu
Address hold time	t_{AH}	$0.5 \times t_{cyc} - 15$	—	ns	
$\overline{AS}$ delay time	t_{ASD}	—	20	ns	
$\overline{RD}$ delay time 1	t_{RSD1}	—	20	ns	
$\overline{RD}$ delay time 2	t_{RSD2}	—	20	ns	
Read data setup time	t_{RDS}	15	—	ns	
Read data hold time	t_{RDH}	0	—	ns	
Read data access time 1	t_{ACC1}	—	$1.0 \times t_{cyc} - 35$	ns	
Read data access time 2	t_{ACC2}	—	$1.5 \times t_{cyc} - 25$	ns	
Read data access time 3	t_{ACC3}	—	$2.0 \times t_{cyc} - 35$	ns	
Read data access time 4	t_{ACC4}	—	$2.5 \times t_{cyc} - 25$	ns	
Read data access time 5	t_{ACC5}	—	$3.0 \times t_{cyc} - 35$	ns	
$\overline{WR}$ delay time 1	t_{WRD1}	—	20	ns	
$\overline{WR}$ delay time 2	t_{WRD2}	—	20	ns	
$\overline{WR}$ pulse width 1	t_{WSW1}	$1.0 \times t_{cyc} - 20$	—	ns	
$\overline{WR}$ pulse width 2	t_{WSW2}	$1.5 \times t_{cyc} - 20$	—	ns	
Write data delay time	t_{WDD}	—	30	ns	
Write data setup time	t_{WDS}	$0.5 \times t_{cyc} - 20$	—	ns	
Write data hold time	t_{WDH}	$0.5 \times t_{cyc} - 10$	—	ns	
$\overline{WAIT}$ setup time	t_{WTS}	30	—	ns	Figu
$\overline{WAIT}$ hold time	t_{WTH}	5	—	ns	

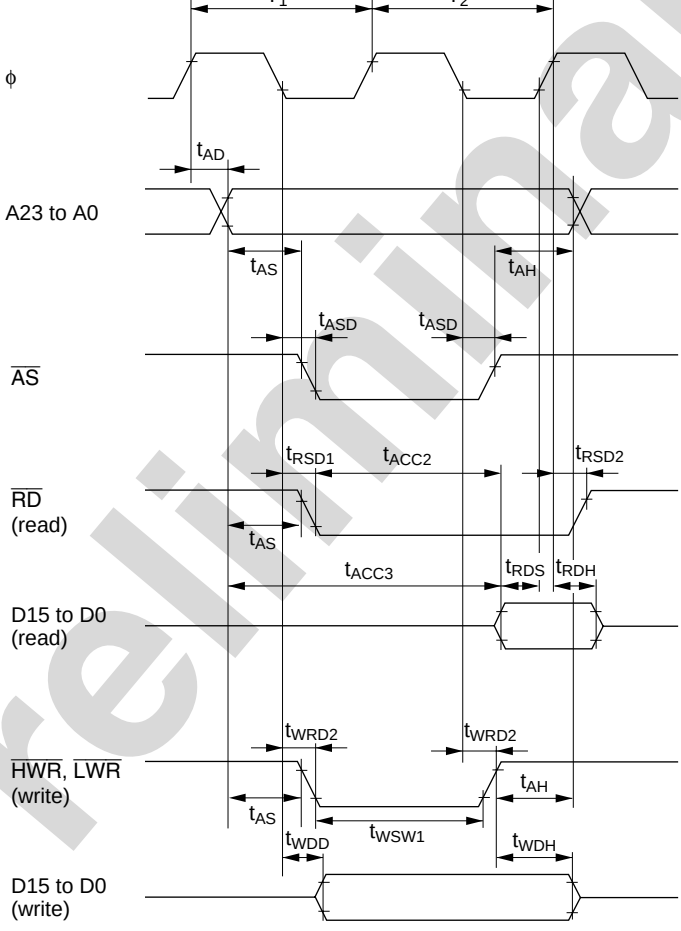


Figure 22.6 Basic Bus Timing (Two-State Access)

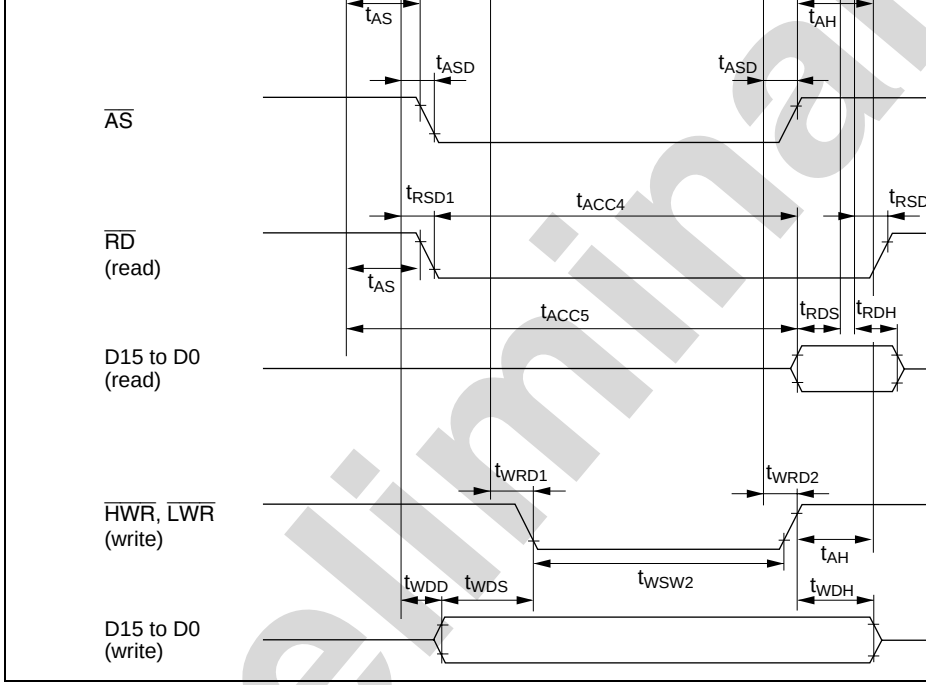


Figure 22.7 Basic Bus Timing (Three-State Access)

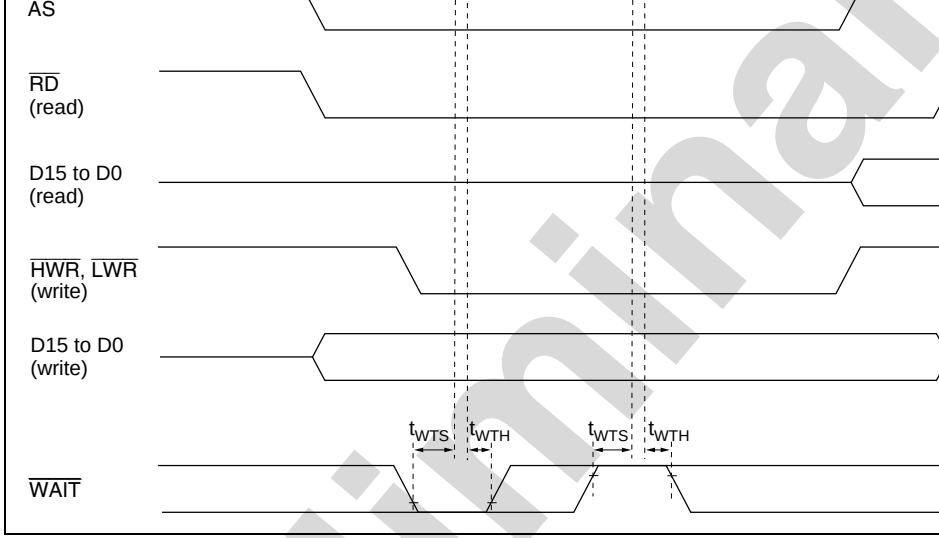


Figure 22.8 Basic Bus Timing (Three-State Access with One Wait State

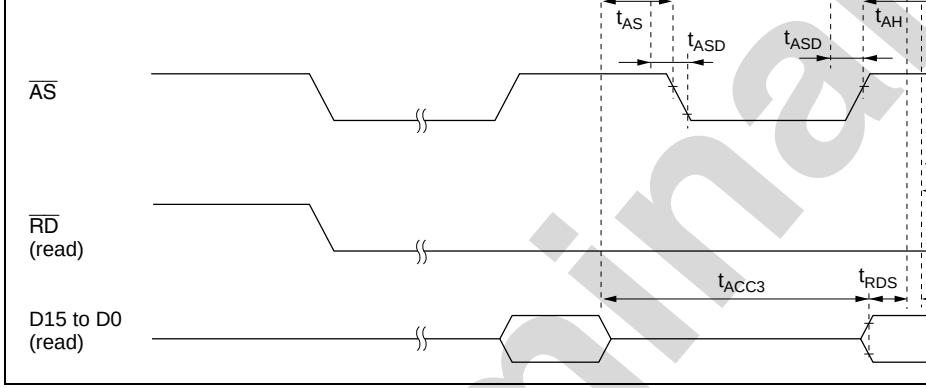


Figure 22.9 Burst ROM Access Timing (Two-State Access)

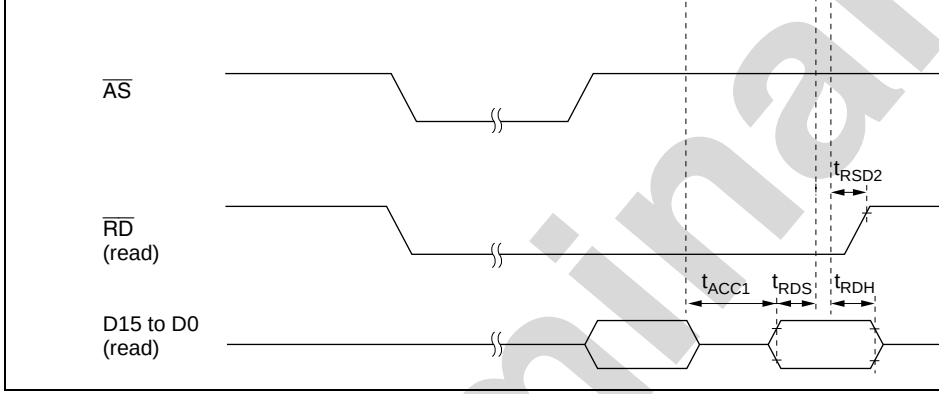


Figure 22.10 Burst ROM Access Timing (One-State Access)

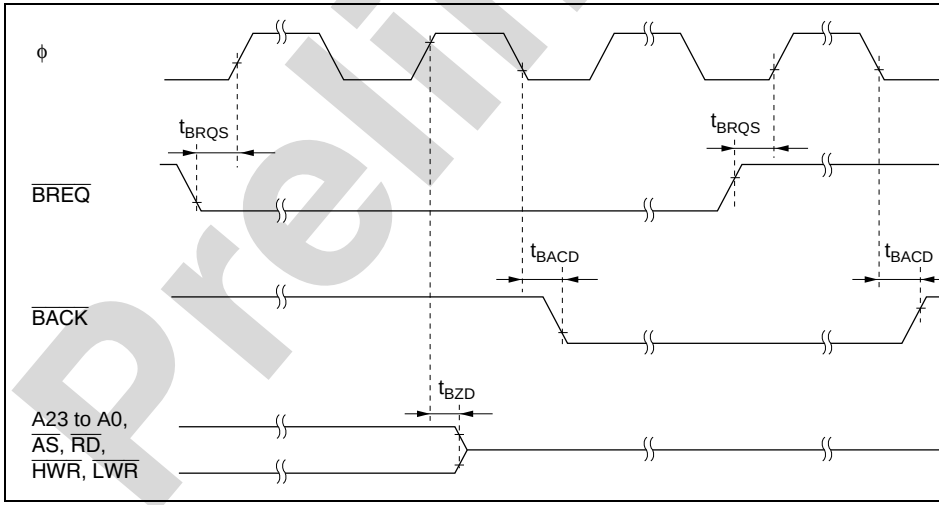


Figure 22.11 External Bus Release Timing

Item			Symbol	Min	Max	Unit	Test C
I/O port	Output data delay time		t_{PWD}	—	50	ns	Figure
	Input data setup time		t_{PRS}	30	—		
	Input data hold time		t_{PRH}	30	—		
TPU	Timer output delay time		t_{TOCD}	—	50	ns	Figure
	Timer input setup time		t_{TICS}	30	—		
	Timer clock input setup time		t_{TCKS}	30	—	ns	Figure
	Timer clock pulse width	Single edge	t_{TCKWH}	1.5	—	t_{cyc}	
		Both edges	t_{TCKWL}	2.5	—		
SCI	Input clock cycle	Asynchro-nous	t_{Scyc}	4	—	t_{cyc}	Figure
		Synchronous		6	—		
	Input clock pulse width		t_{SCKW}	0.4	0.6	t_{Scyc}	
	Input clock rise time		t_{SCKr}	—	1.5	t_{cyc}	
	Input clock fall time		t_{SCKf}	—	1.5		
	Transmit data delay time		t_{TXD}	—	50	ns	Figure
	Receive data setup time (synchronous)		t_{RXS}	50	—		
	Receive data hold time (synchronous)		t_{RXH}	50	—		
A/D converter	Trigger input setup time		t_{TRGS}	30	—	ns	Figure
HCAN*	Transmit data delay time		t_{HTXD}	—	100	ns	Figure
	Transmit data setup time		t_{HRXS}	100	—		
	Transmit data hold time		t_{HRXH}	100	—		

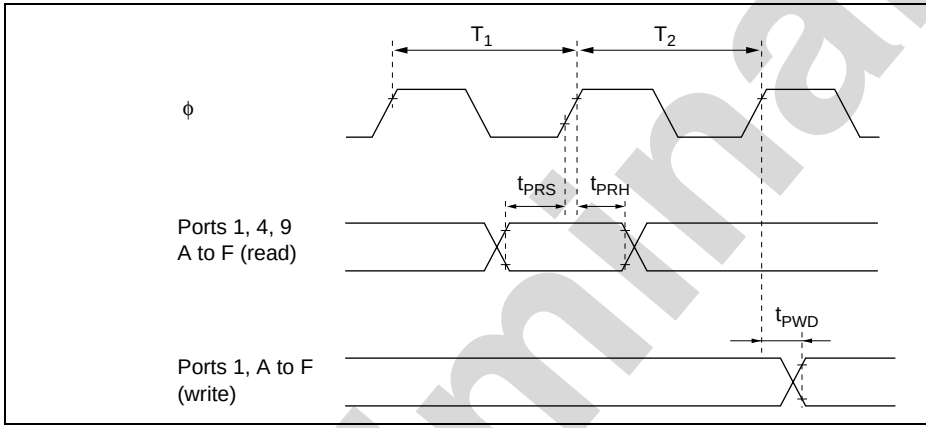


Figure 22.13 I/O Port Input/Output Timing

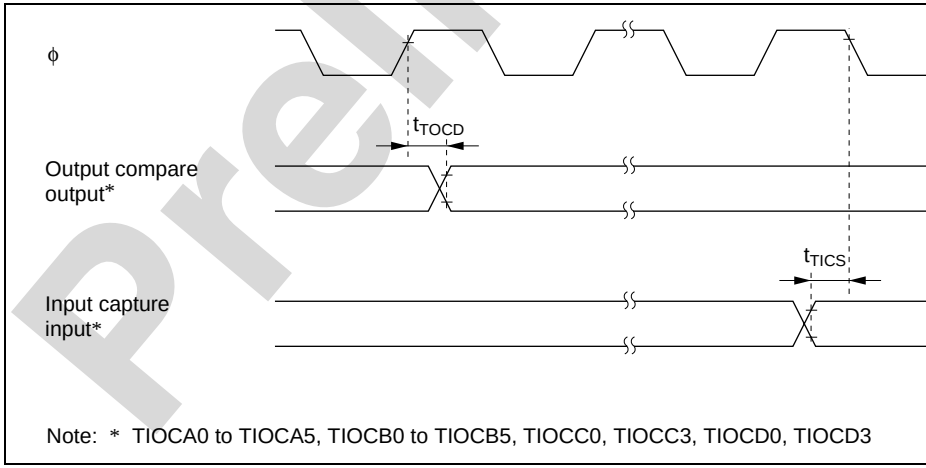


Figure 22.14 TPU Input/Output Timing

Figure 22.15 TPU Clock Input Timing

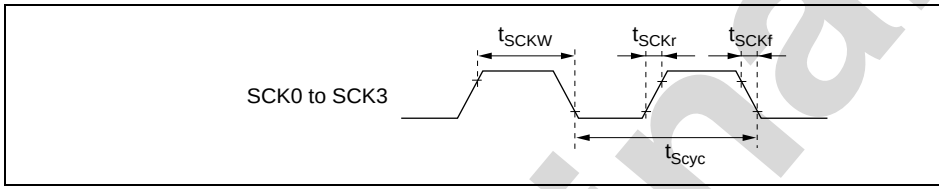


Figure 22.16 SCK Clock Input Timing

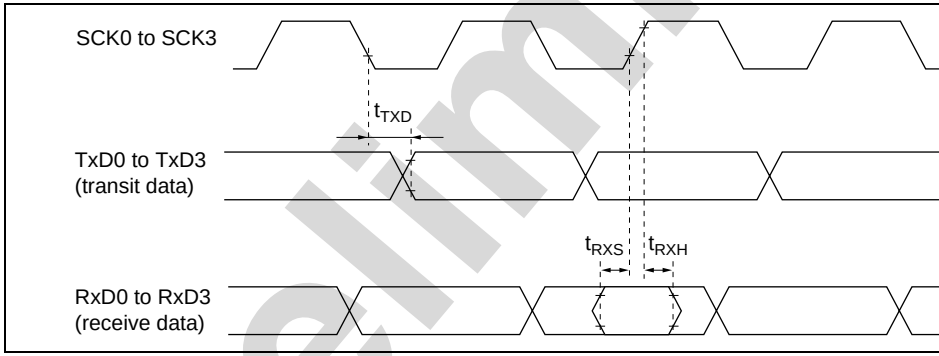


Figure 22.17 SCI Input/Output Timing (Clock Synchronous Mode)

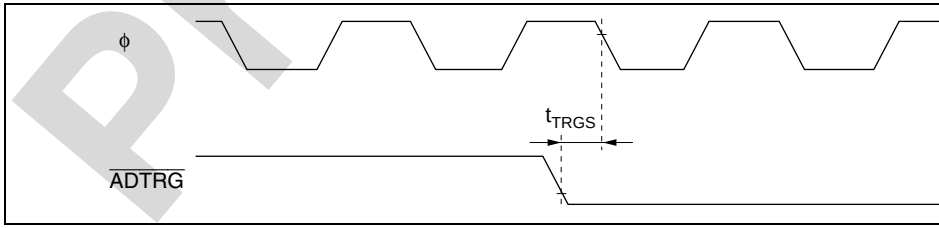


Figure 22.18 A/D Converter External Trigger Input Timing

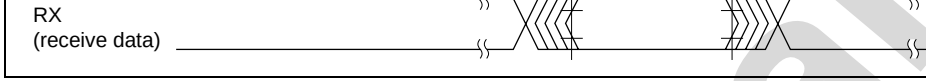


Figure 22.19 HCAN Input/Output Timing

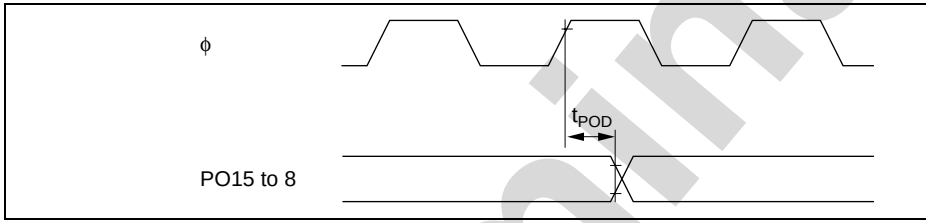


Figure 22.20 PPG Output Timing

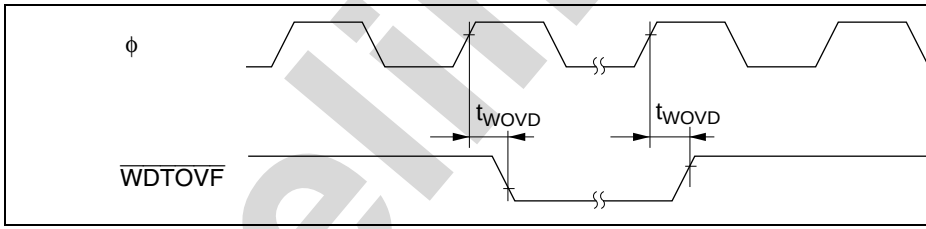


Figure 22.21 WDT0 Output Timing

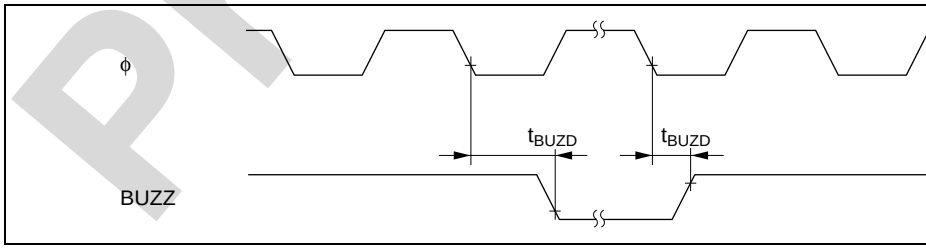


Figure 22.22 WDT1 Output Timing

Item	Min	Typ	Max	Unit	Test C
Resolution	10	10	10	bits	
Conversion time	—	—	—	µs	AV_{CC} <
	10	—	—		AV_{CC} ≥
Analog input capacitance	—	—	20	pF	
Permissible signal-source impedance	—	—	5	kΩ	
Nonlinearity error	—	—	±3.5	LSB	
Offset error	—	—	±3.5	LSB	
Full-scale error	—	—	±3.5	LSB	
Quantization	—	±0.5	—	LSB	
Absolute accuracy	—	—	±4.0	LSB	

22.5 D/A Conversion Characteristics

Table 22.9 shows the D/A conversion characteristics.

Table 22.9 D/A Conversion Characteristics

Conditions: V_{CC} = $PLLV_{CC}$ = 3.0 V to 3.6 V, PV_{CC} = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.
V_{ref} = 4.5 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 4 to 20 MHz, T_a = –20°C to +
(regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Item	Min	Typ	Max	Unit	Test Conditi
Resolution	8	8	8	bits	
Conversion time	—	—	10	µs	20-pF capaci
Absolute accuracy	—	±1.5	±2.0	LSB	2-MΩ resistiv
	—	—	±1.5	LSB	4-MΩ resistiv

Erase time*1*3*5		t_E	—	100	1000
Number of rewrites		N_{WEC}	—	—	100
Programming	Wait time after SWE1 bit setting*1	x0	1	—	—
	Wait time after PSU1 bit setting*1	y	50	—	—
	Wait time after P1 bit setting*1*4	z0	—	—	30
		z1	—	—	10
		z2	—	—	200
	Wait time after P1 bit clearing*1	α	5	—	—
	Wait time after PSU1 bit clearing*1	β	5	—	—
	Wait time after PV1 bit setting*1	γ	4	—	—
	Wait time after H'FF dummy write*1	ε	2	—	—
	Wait time after PV1 bit clearing*1	η	2	—	—
	Maximum number of writes*1*4	N1	—	—	6
		N2	—	—	994
Common	Wait time after SWE1 bit clearing*1	x1	100	—	—
Erasing	Wait time after SWE1 bit setting*1	x	1	—	—
	Wait time after ESU1 bit setting*1	y	100	—	—
	Wait time after E1 bit setting*1*5	z	—	—	10
	Wait time after E1 bit clearing*1	α	10	—	—
	Wait time after ESU1 bit clearing*1	β	10	—	—
	Wait time after EV1 bit setting*1	γ	6	—	—
	Wait time after H'FF dummy write*1	ε	2	—	—
	Wait time after EV1 bit clearing*1	η	4	—	—
	Maximum number of erases*1*5	N	—	—	100

Notes:
1. Follow the program/erase algorithms when making the time settings.
2. Programming time per 128 bytes. (Indicates the total time during which the P1 bit is memory control register 1 (FLMCR1). Does not include the program-verify time.)
3. Time to erase one block. (Indicates the time during which the E1 bit is set in FLMCR include the erase-verify time.)
4. Maximum programming time
   (t_P(max) = Wait time after P1 bit setting (z) × maximum number of writes (N))
   (z0 + z1) × 6 + z2 × 994
5. Maximum erase time
   (t_E(max) = Wait time after E1 bit setting (z) × maximum number of erases (N))

Rs	General register (source)
Rn	General register*
ERn	General register (32-bit register)
MAC	Multiply-and-accumulate register (32-bit register)
(EAd)	Destination operand
(EAs)	Source operand
EXR	Extended control register
CCR	Condition-code register
N	N (negative) flag in CCR
Z	Z (zero) flag in CCR
V	V (overflow) flag in CCR
C	C (carry) flag in CCR
PC	Program counter
SP	Stack pointer
#IMM	Immediate data
disp	Displacement
+	Add
–	Subtract
×	Multiply
÷	Divide
∧	Logical AND
∨	Logical OR
⊕	Logical exclusive OR
→	Transfer from the operand on the left to the operand on the ri transition from the state on the left to the state on the right
¬	Logical NOT (logical complement)
( ) < >	Contents of operand
:8/:16/:24/:32	8-, 16-, 24-, or 32-bit length

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

	Mnemonic	Operand Size	Addressing Mode/ Instruction Length (Bytes)								Operation	Condition Code				
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	—	I	H	N	Z	V	C
MOV	MOV.B #xx:8,Rd	B	2								#xx:8→Rd8	—	—	↕	↕	0	—
	MOV.B Rs,Rd	B		2							Rs8→Rd8	—	—	↕	↕	0	—
	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:32,ERs),Rd	B				8					@(d:32,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:32,Rd	B						6			@aa:32→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:32,ERd)	B				8					Rs8→@(d:32,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:32	B						6			Rs8→@aa:32	—	—	↕	↕	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	—

RENESAS

Mnemonic	Operand Size	Addressing Mode/ Instruction Length (Bytes)								Operation	Condition Code		
		#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa		I	H	N
MOV	MOV.W @(d:16,ERs),Rd	W			4					@(d:16,ERs)→Rd16	—	—	↕
	MOV.W @(d:32,ERs),Rd	W			8					@(d:32,ERs)→Rd16	—	—	↕
	MOV.W @ERs+,Rd	W				2				@ERs→Rd16,ERs32+2→ERs32	—	—	↕
	MOV.W @aa:16,Rd	W					4			@aa:16→Rd16	—	—	↕
	MOV.W @aa:32,Rd	W					6			@aa:32→Rd16	—	—	↕
	MOV.W Rs,@ERd	W		2						Rs16→@ERd	—	—	↕
	MOV.W Rs,@(d:16,ERd)	W			4					Rs16→@(d:16,ERd)	—	—	↕
	MOV.W Rs,@(d:32,ERd)	W			8					Rs16→@(d:32,ERd)	—	—	↕
	MOV.W Rs,@-ERd	W				2				ERd32-2→ERd32,Rs16→@ERd	—	—	↕
	MOV.W Rs,@aa:16	W					4			Rs16→@aa:16	—	—	↕
	MOV.W Rs,@aa:32	W					6			Rs16→@aa:32	—	—	↕
	MOV.L #xx:32,ERd	L	6							#xx:32→ERd32	—	—	↕
	MOV.L ERs,ERd	L		2						ERs32→ERd32	—	—	↕
	MOV.L @ERs,ERd	L			4					@ERs→ERd32	—	—	↕
	MOV.L @(d:16,ERs),ERd	L				6				@(d:16,ERs)→ERd32	—	—	↕
	MOV.L @(d:32,ERs),ERd	L				10				@(d:32,ERs)→ERd32	—	—	↕
	MOV.L @ERs+,ERd	L					4			@ERs→ERd32,ERs32+4→ERs32	—	—	↕
	MOV.L @aa:16,ERd	L						6		@aa:16→ERd32	—	—	↕
	MOV.L @aa:32,ERd	L						8		@aa:32→ERd32	—	—	↕

Mnemonic	Operand Size	Addressing Mode/ Instruction Length (Bytes)								Operation	Condition Code					
		#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa		I	H	U	N	Z	V
MOV	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:32,ERd)	L			10					ERs32→@(d:32,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:32	L					8			ERs32→@aa:32	—	—	—	↕	↕	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
LDM	LDM @SP+,(ERm-ERn)	L							4	(@SP→ERn32,SP+4→SP) Repeated for each register restored	—	—	—	—	—	—
STM	STM (ERm-ERn),@-SP	L							4	(SP-4→SP, ERn32→@SP) Repeated for each register saved	—	—	—	—	—	—
MOVFPE	MOVFPE @aa:16,Rd	Cannot be used in the H8S/2626 Group or H8S/2623 Group														
MOVTPE	MOVTPE Rs,@aa:16	Cannot be used in the H8S/2626 Group or H8S/2623 Group														

Mnemonic	Operand Size	Addressing Mode/ Instruction Length (Bytes)								Operation	Condition Code			
		#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	—	I	H	N	Z
ADD	ADD.B #xx:8,Rd	B	2							Rd8+#xx:8→Rd8	—	↕	↕	↕
	ADD.B Rs,Rd	B	2							Rd8+Rs8→Rd8	—	↕	↕	↕
	ADD.W #xx:16,Rd	W	4							Rd16+#xx:16→Rd16	—	[3]	↕	↕
	ADD.W Rs,Rd	W	2							Rd16+Rs16→Rd16	—	[3]	↕	↕
	ADD.L #xx:32,ERd	L	6							ERd32+#xx:32→ERd32	—	[4]	↕	↕
	ADD.L ERs,ERd	L	2							ERd32+ERs32→ERd32	—	[4]	↕	↕
ADDX	ADDX #xx:8,Rd	B	2							Rd8+#xx:8+C→Rd8	—	↕	↕	[5]
	ADDX Rs,Rd	B	2							Rd8+Rs8+C→Rd8	—	↕	↕	[5]
ADDS	ADDS #1,ERd	L	2							ERd32+1→ERd32	—	—	—	—
	ADDS #2,ERd	L	2							ERd32+2→ERd32	—	—	—	—
	ADDS #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	—	[3]	↕	↕

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code		
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N
SUB	SUB.W Rs,Rd	W	2									Rd16-Rs16→Rd16	—	[3]	↕
	SUB.L #xx:32,ERd	L	6									ERd32-#xx:32→ERd32	—	[4]	↕
	SUB.L ERs,ERd	L		2								ERd32-ERs32→ERd32	—	[4]	↕
SUBX	SUBX #xx:8,Rd	B	2									Rd8-#xx:8-C→Rd8	—	↕	↕
	SUBX Rs,Rd	B		2								Rd8-Rs8-C→Rd8	—	↕	↕
SUBS	SUBS #1,ERd	L		2								ERd32-1→ERd32	—	—	—
	SUBS #2,ERd	L		2								ERd32-2→ERd32	—	—	—
	SUBS #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	—	—	↕
	DEC.L #1,ERd	L		2								ERd32-1→ERd32	—	—	↕
	DEC.L #2,ERd	L		2								ERd32-2→ERd32	—	—	↕
DAS	DAS Rd	B		2								Rd8 decimal adjust→Rd8	—	*	↕
MULXU	MULXU.B Rs,Rd	B		2								Rd8×Rs8→Rd16 (unsigned multiplication)	—	—	—
	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)	—	—	↕

RENESAS

Mnemonic	Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code		
		#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N
DIVXU	DIVXU.B Rs,Rd	B	2								Rd16÷Rs8→Rd16 (RdH: remainder, RdL: quotient) (unsigned division)	—	—	[1]
	DIVXU.W Rs,ERd	W	2								ERd32÷Rs16→ERd32 (Ed: remainder, Rd: quotient) (unsigned division)	—	—	[2]
DIVXS	divxs.B Rs,Rd	B	4								Rd16÷Rs8→Rd16 (RdH: remainder, RdL: quotient) (signed division)	—	—	[1]
	DIVXS.W Rs,ERd	W	4								ERd32÷Rs16→ERd32 (Ed: remainder, Rd: quotient) (signed division)	—	—	[2]
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	—	[3]	↔
	CMP.W Rs,Rd	W		2							Rd16-Rs16	—	[3]	↔
	CMP.L #xx:32,ERd	L	6								ERd32-#xx:32	—	[4]	↔
	CMP.L ERs,ERd	L		2							ERd32-ERs32	—	[4]	↔
NEG	NEG.B Rd	B		2							0-Rd8→Rd8	—	↔	↔
	NEG.W Rd	W		2							0-Rd16→Rd16	—	↔	↔
	NEG.L ERd	L		2							0-ERd32→ERd32	—	↔	↔
EXTU	EXTU.W Rd	W		2							0→(<bit 15 to 8> of Rd16)	—	—	0
	EXTU.L ERd	L		2							0→(<bit 31 to 16> of ERd32)	—	—	0

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)										Operation	Condi	
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	—			I	H
EXTS	EXTS.W Rd	W		2								(<bit 7> of Rd16)→ (<bit 15 to 8> of Rd16)	—	—
	EXTS.L ERd	L		2								(<bit 15> of ERd32)→ (<bit 31 to 16> of ERd32)	—	—
TAS	TAS @ERd *3	B			4							@ERd-0→CCR set, (1)→ (<bit 7> of @ERd)	—	—
MAC	MAC @ERn+,@ERm+	—					4					@ERn×@ERm+MAC→MAC (signed multiplication) ERn+2→ERn,ERm+2→ERm	—	—
CLRMAC	CLRMAC	—									2	0→MACH,MACL	—	—
LDMAC	LDMAC ERs,MACH	L		2								ERs→MACH	—	—
	LDMAC ERs,MACL	L		2								ERs→MACL	—	—
STMAC	STMAC MACH,ERd	L		2								MACH→ERd	—	—
	STMAC MACL,ERd	L		2								MACL→ERd	—	—

RENESAS

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code			
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	—		I	H	N	Z
AND	AND.B #xx:8,Rd	B	2									Rd8∧#xx:8→Rd8	—	—	↕	↕
	AND.B Rs,Rd	B		2								Rd8∧Rs8→Rd8	—	—	↕	↕
	AND.W #xx:16,Rd	W	4									Rd16∧#xx:16→Rd16	—	—	↕	↕
	AND.W Rs,Rd	W		2								Rd16∧Rs16→Rd16	—	—	↕	↕
	AND.L #xx:32,ERd	L	6									ERd32∧#xx:32→ERd32	—	—	↕	↕
	AND.L ERs,ERd	L		4								ERd32∧ERs32→ERd32	—	—	↕	↕
OR	OR.B #xx:8,Rd	B	2									Rd8∨#xx:8→Rd8	—	—	↕	↕
	OR.B Rs,Rd	B		2								Rd8∨Rs8→Rd8	—	—	↕	↕
	OR.W #xx:16,Rd	W	4									Rd16∨#xx:16→Rd16	—	—	↕	↕
	OR.W Rs,Rd	W		2								Rd16∨Rs16→Rd16	—	—	↕	↕
	OR.L #xx:32,ERd	L	6									ERd32∨#xx:32→ERd32	—	—	↕	↕
	OR.L ERs,ERd	L		4								ERd32∨ERs32→ERd32	—	—	↕	↕
XOR	XOR.B #xx:8,Rd	B	2									Rd8⊕#xx:8→Rd8	—	—	↕	↕
	XOR.B Rs,Rd	B		2								Rd8⊕Rs8→Rd8	—	—	↕	↕
	XOR.W #xx:16,Rd	W	4									Rd16⊕#xx:16→Rd16	—	—	↕	↕
	XOR.W Rs,Rd	W		2								Rd16⊕Rs16→Rd16	—	—	↕	↕
	XOR.L #xx:32,ERd	L	6									ERd32⊕#xx:32→ERd32	—	—	↕	↕
	XOR.L ERs,ERd	L		4								ERd32⊕ERs32→ERd32	—	—	↕	↕
NOT	NOT.B Rd	B		2								¬ Rd8→Rd8	—	—	↕	↕
	NOT.W Rd	W		2								¬ Rd16→Rd16	—	—	↕	↕

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)										Operation	Condition Code			
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	—			I	H	N	Z
SHAL	SHAL.B Rd	B		2										—	—	↔	↔
	SHAL.B #2,Rd	B		2										—	—	↔	↔
	SHAL.W Rd	W		2										—	—	↔	↔
	SHAL.W #2,Rd	W		2										—	—	↔	↔
	SHAL.L ERd	L		2										—	—	↔	↔
	SHAL.L #2,ERd	L		2										—	—	↔	↔
SHAR	SHAR.B Rd	B		2										—	—	↔	↔
	SHAR.B #2,Rd	B		2										—	—	↔	↔
	SHAR.W Rd	W		2										—	—	↔	↔
	SHAR.W #2,Rd	W		2										—	—	↔	↔
	SHAR.L ERd	L		2										—	—	↔	↔
	SHAR.L #2,ERd	L		2										—	—	↔	↔
SHLL	SHLL.B Rd	B		2										—	—	↔	↔
	SHLL.B #2,Rd	B		2										—	—	↔	↔
	SHLL.W Rd	W		2										—	—	↔	↔
	SHLL.W #2,Rd	W		2										—	—	↔	↔
	SHLL.L ERd	L		2										—	—	↔	↔
	SHLL.L #2,ERd	L		2										—	—	↔	↔

RENESAS

Mnemonic	Operand Size	Addressing Mode/ Instruction Length (Bytes)										Operation	Condition Code					
		#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|			I	H	N	Z	V	C
SHLR	SHLR.B Rd	B	2									0 → MSB → LSB → C	—	—	↕	↕	0	↕
	SHLR.B #2,Rd	B	2										—	—	↕	↕	0	↕
	SHLR.W Rd	W	2										—	—	↕	↕	0	↕
	SHLR.W #2,Rd	W	2										—	—	↕	↕	0	↕
	SHLR.L ERd	L	2										—	—	↕	↕	0	↕
	SHLR.L #2,ERd	L	2										—	—	↕	↕	0	↕
ROTXL	ROTXL.B Rd	B	2									C MSB ← LSB	—	—	↕	↕	0	↕
	ROTXL.B #2,Rd	B	2										—	—	↕	↕	0	↕
	ROTXL.W Rd	W	2										—	—	↕	↕	0	↕
	ROTXL.W #2,Rd	W	2										—	—	↕	↕	0	↕
	ROTXL.L ERd	L	2										—	—	↕	↕	0	↕
	ROTXL.L #2,ERd	L	2										—	—	↕	↕	0	↕
ROTXR	ROTXR.B Rd	B	2									MSB → LSB → C	—	—	↕	↕	0	↕
	ROTXR.B #2,Rd	B	2										—	—	↕	↕	0	↕
	ROTXR.W Rd	W	2										—	—	↕	↕	0	↕
	ROTXR.W #2,Rd	W	2										—	—	↕	↕	0	↕
	ROTXR.L ERd	L	2										—	—	↕	↕	0	↕
	ROTXR.L #2,ERd	L	2										—	—	↕	↕	0	↕

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)										Operation	Condition Code				
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	—			I	H	U	N	Z
ROTL	ROTL.B Rd	B		2										—	—	(?)	(?)	(?)
	ROTL.B #2,Rd	B		2										—	—	(?)	(?)	(?)
	ROTL.W Rd	W		2										—	—	(?)	(?)	(?)
	ROTL.W #2,Rd	W		2										—	—	(?)	(?)	(?)
	ROTL.L ERd	L		2										—	—	(?)	(?)	(?)
	ROTL.L #2,ERd	L		2										—	—	(?)	(?)	(?)
ROTR	ROTR.B Rd	B		2										—	—	(?)	(?)	(?)
	ROTR.B #2,Rd	B		2										—	—	(?)	(?)	(?)
	ROTR.W Rd	W		2										—	—	(?)	(?)	(?)
	ROTR.W #2,Rd	W		2										—	—	(?)	(?)	(?)
	ROTR.L ERd	L		2										—	—	(?)	(?)	(?)
	ROTR.L #2,ERd	L		2										—	—	(?)	(?)	(?)

RENESAS

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)								Operation	Condition Code			
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa		I	H	N	Z
BSET	BSET #xx:3,Rd	B	2								(#xx:3 of Rd8)←1	—	—	—	—
	BSET #xx:3,@ERd	B		4							(#xx:3 of @ERd)←1	—	—	—	—
	BSET #xx:3,@aa:8	B						4			(#xx:3 of @aa:8)←1	—	—	—	—
	BSET #xx:3,@aa:16	B						6			(#xx:3 of @aa:16)←1	—	—	—	—
	BSET #xx:3,@aa:32	B						8			(#xx:3 of @aa:32)←1	—	—	—	—
	BSET Rn,Rd	B	2								(Rn8 of Rd8)←1	—	—	—	—
	BSET Rn,@ERd	B		4							(Rn8 of @ERd)←1	—	—	—	—
	BSET Rn,@aa:8	B						4			(Rn8 of @aa:8)←1	—	—	—	—
	BSET Rn,@aa:16	B						6			(Rn8 of @aa:16)←1	—	—	—	—
	BSET Rn,@aa:32	B						8			(Rn8 of @aa:32)←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 #xx:3,@aa:16	B						6			(#xx:3 of @aa:16)←0	—	—	—	—
	BCLR #xx:3,@aa:32	B						8			(#xx:3 of @aa:32)←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	—	—	—	—
	BCLR Rn,@aa:16	B						6			(Rn8 of @aa:16)←0	—	—	—	—

Mnemonic	Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code				
		#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	—		I	H	N	Z	V
BCLR	BCLR Rn,@aa:32	B					8				(Rn8 of @aa:32)←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 #xx:3,@aa:16	B					6				(#xx:3 of @aa:16)←[¬ (#xx:3 of @aa:16)]	—	—	—	—	—
	BNOT #xx:3,@aa:32	B					8				(#xx:3 of @aa:32)←[¬ (#xx:3 of @aa:32)]	—	—	—	—	—
	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)]	—	—	—	—	—
	BNOT Rn,@aa:16	B					6				(Rn8 of @aa:16)←[¬ (Rn8 of @aa:16)]	—	—	—	—	—
	BNOT Rn,@aa:32	B					8				(Rn8 of @aa:32)←[¬ (Rn8 of @aa:32)]	—	—	—	—	—
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	—	—	—	↕	—

RENESAS

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)								Operation	Condition Code				
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa		I	H	N	Z	V
BTST	BTST #xx:3,@aa:32	B						8			¬ (#xx:3 of @aa:32)→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	—	—	—	↕	—
	BTST Rn,@aa:16	B						6			¬ (Rn8 of @aa:16)→Z	—	—	—	↕	—
	BTST Rn,@aa:32	B						8			¬ (Rn8 of @aa:32)→Z	—	—	—	↕	—
BLD	BLD #xx:3,Rd	B	2								(#xx:3 of Rd8)→C	—	—	—	—	—
	BLD #xx:3,@ERd	B		4							(#xx:3 of @ERd)→C	—	—	—	—	—
	BLD #xx:3,@aa:8	B						4			(#xx:3 of @aa:8)→C	—	—	—	—	—
	BLD #xx:3,@aa:16	B						6			(#xx:3 of @aa:16)→C	—	—	—	—	—
	BLD #xx:3,@aa:32	B						8			(#xx:3 of @aa:32)→C	—	—	—	—	—
BILD	BILD #xx:3,Rd	B	2								¬ (#xx:3 of Rd8)→C	—	—	—	↕	—
	BILD #xx:3,@ERd	B		4							¬ (#xx:3 of @ERd)→C	—	—	—	↕	—
	BILD #xx:3,@aa:8	B						4			¬ (#xx:3 of @aa:8)→C	—	—	—	↕	—
	BILD #xx:3,@aa:16	B						6			¬ (#xx:3 of @aa:16)→C	—	—	—	↕	—
	BILD #xx:3,@aa:32	B						8			¬ (#xx:3 of @aa:32)→C	—	—	—	↕	—
BST	BST #xx:3,Rd	B	2								C→(#xx:3 of Rd8)	—	—	—	—	—
	BST #xx:3,@ERd	B		4							C→(#xx:3 of @ERd24)	—	—	—	—	—
	BST #xx:3,@aa:8	B						4			C→(#xx:3 of @aa:8)	—	—	—	—	—

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)								Operation	Condition Code	
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa		I	H
BST	BST #xx:3,@aa:16	B						6			C→(#xx:3 of @aa:16)	—	—
	BST #xx:3,@aa:32	B						8			C→(#xx:3 of @aa:32)	—	—
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)	—	—
	BIST #xx:3,@aa:16	B						6			¬ C→(#xx:3 of @aa:16)	—	—
	BIST #xx:3,@aa:32	B						8			¬ C→(#xx:3 of @aa:32)	—	—
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	—	—
	BAND #xx:3,@aa:16	B						6			C∧(#xx:3 of @aa:16)→C	—	—
	BAND #xx:3,@aa:32	B						8			C∧(#xx:3 of @aa:32)→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	—	—
	BIAND #xx:3,@aa:16	B						6			C∧[¬ (#xx:3 of @aa:16)]→C	—	—
	BIAND #xx:3,@aa:32	B						8			C∧[¬ (#xx:3 of @aa:32)]→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	—	—

RENESAS

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)								Operation	Condition Codes				
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa		I	H	U	N	Z
BOR	BOR #xx:3,@aa:8	B						4			C∨(#xx:3 of @aa:8)→C	—	—	—	—	—
	BOR #xx:3,@aa:16	B						6			C∨(#xx:3 of @aa:16)→C	—	—	—	—	—
	BOR #xx:3,@aa:32	B						8			C∨(#xx:3 of @aa:32)→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	—	—	—	—	—
	BIOR #xx:3,@aa:16	B						6			C∨[¬ (#xx:3 of @aa:16)]→C	—	—	—	—	—
	BIOR #xx:3,@aa:32	B						8			C∨[¬ (#xx:3 of @aa:32)]→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	—	—	—	—	—
	BXOR #xx:3,@aa:16	B						6			C⊕(#xx:3 of @aa:16)→C	—	—	—	—	—
	BXOR #xx:3,@aa:32	B						8			C⊕(#xx:3 of @aa:32)→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	—	—	—	—	—
	BIXOR #xx:3,@aa:16	B						6			C⊕[¬ (#xx:3 of @aa:16)]→C	—	—	—	—	—
	BIXOR #xx:3,@aa:32	B						8			C⊕[¬ (#xx:3 of @aa:32)]→C	—	—	—	—	—

	Mnemonic	Operand Size	Addressing Mode/ Instruction Length (Bytes)										Operation	Branching Condition	Condition Code			
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	—				I	H	N	Z
Bcc	BRA d:8(BT d:8)	—							2			if condition is true then PC←PC+d else next;	Always		—	—	—	—
	BRA d:16(BT d:16)	—							4						—	—	—	—
	BRN d:8(BF d:8)	—							2				Never		—	—	—	—
	BRN d:16(BF d:16)	—							4						—	—	—	—
	BHI d:8	—							2				C∨Z=0		—	—	—	—
	BHI d:16	—							4						—	—	—	—
	BLS d:8	—							2				C∨Z=1		—	—	—	—
	BLS d:16	—							4						—	—	—	—
	BCC d:B(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						—	—	—	—

RENESAS

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition				
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	—		Branching Condition	I	H	N	Z
Bcc	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					—	—	—	—

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code							
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	—		I	H	U	N	Z	V	C
JMP	JMP @ERn	—			2							PC← ERn	—	—	—	—	—	—	—	—
	JMP @aa:24	—						4				PC← aa:24	—	—	—	—	—	—	—	—
	JMP @@aa:8	—								2		PC← @aa:8	—	—	—	—	—	—	—	—
BSR	BSR d:8	—							2			PC→@-SP,PC←PC+d:8	—	—	—	—	—	—	—	—
	BSR d:16	—							4			PC→@-SP,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+	—	—	—	—	—	—	—	—

RENESAS

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition			
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	—		I	H	N	Z
TRAPA	TRAPA #xx:2	—										PC→@-SP,CCR→@-SP, EXR→@-SP,<vector>→PC	1	—	—	—
RTE	RTE	—										EXR←@SP+,CCR←@SP+, PC←@SP+	↕	↕	↕	↕
SLEEP	SLEEP	—										Transition to power-down state	—	—	—	—
LDC	LDC #xx:8,CCR	B	2									#xx:8→CCR	↕	↕	↕	↕
	LDC #xx:8,EXR	B	4									#xx:8→EXR	—	—	—	—
	LDC Rs,CCR	B		2								Rs8→CCR	↕	↕	↕	↕
	LDC Rs,EXR	B		2								Rs8→EXR	—	—	—	—
	LDC @ERs,CCR	W			4							@ERs→CCR	↕	↕	↕	↕
	LDC @ERs,EXR	W			4							@ERs→EXR	—	—	—	—
	LDC @(d:16,ERs),CCR	W				6						@(d:16,ERs)→CCR	↕	↕	↕	↕
	LDC @(d:16,ERs),EXR	W				6						@(d:16,ERs)→EXR	—	—	—	—
	LDC @(d:32,ERs),CCR	W				10						@(d:32,ERs)→CCR	↕	↕	↕	↕
	LDC @(d:32,ERs),EXR	W				10						@(d:32,ERs)→EXR	—	—	—	—
	LDC @ERs+,CCR	W					4					@ERs→CCR,ERs32+2→ERs32	↕	↕	↕	↕
	LDC @ERs+,EXR	W					4					@ERs→EXR,ERs32+2→ERs32	—	—	—	—
	LDC @aa:16,CCR	W						6				@aa:16→CCR	↕	↕	↕	↕
	LDC @aa:16,EXR	W						6				@aa:16→EXR	—	—	—	—
	LDC @aa:32,CCR	W						8				@aa:32→CCR	↕	↕	↕	↕
	LDC @aa:32,EXR	W						8				@aa:32→EXR	—	—	—	—

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)								Operation	Condition Code		
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa		I	H	N
STC	STC CCR,Rd	B	2								CCR→Rd8	—	—	—
	STC EXR,Rd	B	2								EXR→Rd8	—	—	—
	STC CCR,@ERd	W		4							CCR→@ERd	—	—	—
	STC EXR,@ERd	W		4							EXR→@ERd	—	—	—
	STC CCR,@(d:16,ERd)	W			6						CCR→@(d:16,ERd)	—	—	—
	STC EXR,@(d:16,ERd)	W			6						EXR→@(d:16,ERd)	—	—	—
	STC CCR,@(d:32,ERd)	W			10						CCR→@(d:32,ERd)	—	—	—
	STC EXR,@(d:32,ERd)	W			10						EXR→@(d:32,ERd)	—	—	—
	STC CCR,@-ERd	W				4					ERd32-2→ERd32,CCR→@ERd	—	—	—
	STC EXR,@-ERd	W				4					ERd32-2→ERd32,EXR→@ERd	—	—	—
	STC CCR,@aa:16	W					6				CCR→@aa:16	—	—	—
	STC EXR,@aa:16	W					6				EXR→@aa:16	—	—	—
	STC CCR,@aa:32	W					8				CCR→@aa:32	—	—	—
	STC EXR,@aa:32	W					8				EXR→@aa:32	—	—	—
ANDC	ANDC #xx:8,CCR	B	2								CCR∧#xx:8→CCR	↔	↔	↔
	ANDC #xx:8,EXR	B	4								EXR∧#xx:8→EXR	—	—	—
ORC	ORC #xx:8,CCR	B	2								CCR∨#xx:8→CCR	↔	↔	↔
	ORC #xx:8,EXR	B	4								EXR∨#xx:8→EXR	—	—	—
XORC	XORC #xx:8,CCR	B	2								CCR⊕#xx:8→CCR	↔	↔	↔
	XORC #xx:8,EXR	B	4								EXR⊕#xx:8→EXR	—	—	—

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)										Operation	Condition Code			
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa				I	H	N	Z
EEPMOV	EEPMOV.B	—									4	if R4L≠0 Repeat @ER5→@ER6 ER5+1→ER5 ER6+1→ER6 R4L-1→R4L Until R4L=0 else next;	—	—	—	—
	EEPMOV.W	—									4	if R4≠0 Repeat @ER5→@ER6 ER5+1→ER5 ER6+1→ER6 R4-1→R4 Until R4=0 else next;	—	—	—	—

Notes: 1. The number of states is the number of states required for execution when the instruction and its operands are located in on-chip memory.
2. n is the initial value of R4L or R4.
3. Only register ER0, ER1, ER4, or ER5 should be used when using the TAS instruction.
[1] Seven states for saving or restoring two registers, nine states for three registers, or eleven states for four registers.
[2] Cannot be used in the H8S/2626 Group or H8S/2623 Group.
[3] Set to 1 when a carry or borrow occurs at bit 11; otherwise cleared to 0.
[4] Set to 1 when a carry or borrow occurs at bit 27; otherwise cleared to 0.
[5] Retains its previous value when the result is zero; otherwise cleared to 0.
[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.
[9] One additional state is required for execution when EXR is valid.

Instruction	Mnemonic	Size	Instruction Format							
			1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
ADD	ADD.B #xx:8,Rd	B	8 rd	IMM						
	ADD.B Rs,Rd	B	0 8	rs rd						
	ADD.W #xx:16,Rd	W	7 9	1 rd	IMM					
	ADD.W Rs,Rd	W	0 9	rs rd						
	ADD.L #xx:32,ERd	L	7 A	1 0 erd	IMM					
	ADD.L ERs,ERd	L	0 A	1 ers 0 erd						
ADDS	ADDS #1,ERd	L	0 B	0 0 erd						
	ADDS #2,ERd	L	0 B	8 0 erd						
	ADDS #4,ERd	L	0 B	9 0 erd						
ADDX	ADDX #xx:8,Rd	B	9 rd	IMM						
	ADDX Rs,Rd	B	0 E	rs rd						
AND	AND.B #xx:8,Rd	B	E rd	IMM						
	AND.B Rs,Rd	B	1 6	rs rd						
	AND.W #xx:16,Rd	W	7 9	6 rd	IMM					
	AND.W Rs,Rd	W	6 6	rs rd						
	AND.L #xx:32,ERd	L	7 A	6 0 erd	IMM					
	AND.L ERs,ERd	L	0 1	F 0	6 6	0 ers 0 erd				
ANDC	ANDC #xx:8,CCR	B	0 6	IMM						
	ANDC #xx:8,EXR	B	0 1	4 1	0 6	IMM				
BAND	BAND #xx:3,Rd	B	7 6	0 IMM rd						
	BAND #xx:3,@ERd	B	7 C	0 erd 0	7 6	0 IMM 0				
	BAND #xx:3,@aa:8	B	7 E	abs	7 6	0 IMM 0				
	BAND #xx:3,@aa:16	B	6 A	1 0	abs		7 6	0 IMM 0		
	BAND #xx:3,@aa:32	B	6 A	3 0	abs				7 6	0 IMM
Bcc	BRA d:8 (BT d:8)	—	4 0	disp						
	BRA d:16 (BT d:16)	—	5 8	0 0	disp					
	BRN d:8 (BF d:8)	—	4 1	disp						
	BRN d:16 (BF d:16)	—	5 8	1 0	disp					

Instruction	Mnemonic	Size	Instruction Format							
			1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8
Bcc	BHI d:8	—	4 2	disp						
	BHI d:16	—	5 8	2 0	disp					
	BLS d:8	—	4 3	disp						
	BLS d:16	—	5 8	3 0	disp					
	BCC d:8 (BHS d:8)	—	4 4	disp						
	BCC d:16 (BHS d:16)	—	5 8	4 0	disp					
	BCS d:8 (BLO d:8)	—	4 5	disp						
	BCS d:16 (BLO d:16)	—	5 8	5 0	disp					
	BNE d:8	—	4 6	disp						
	BNE d:16	—	5 8	6 0	disp					
	BEQ d:8	—	4 7	disp						
	BEQ d:16	—	5 8	7 0	disp					
	BVC d:8	—	4 8	disp						
	BVC d:16	—	5 8	8 0	disp					
	BVS d:8	—	4 9	disp						
	BVS d:16	—	5 8	9 0	disp					
	BPL d:8	—	4 A	disp						
	BPL d:16	—	5 8	A 0	disp					
	BMI d:8	—	4 B	disp						
	BMI d:16	—	5 8	B 0	disp					
	BGE d:8	—	4 C	disp						
	BGE d:16	—	5 8	C 0	disp					
	BLT d:8	—	4 D	disp						
	BLT d:16	—	5 8	D 0	disp					
	BGT d:8	—	4 E	disp						
	BGT d:16	—	5 8	E 0	disp					
	BLE d:8	—	4 F	disp						
	BLE d:16	—	5 8	F 0	disp					

Instruction	Mnemonic	Size	Instruction Format																									
			1st byte			2nd byte			3rd byte			4th byte			5th byte			6th byte			7th byte			8				
BCLR	BCLR #xx:3,Rd	B	7		2	0	IMM	rd																				
	BCLR #xx:3,@ERd	B	7		D	0	erd	0	7		2	0	IMM	0														
	BCLR #xx:3,@aa:8	B	7		F	abs			7		2	0	IMM	0														
	BCLR #xx:3,@aa:16	B	6		A	1		8	abs						7		2	0	IMM	0								
	BCLR #xx:3,@aa:32	B	6		A	3		8	abs												7		2	0	IMM	0		
	BCLR Rn,Rd	B	6		2	rn		rd																				
	BCLR Rn,@ERd	B	7		D	0	erd	0	6		2	rn		0														
	BCLR Rn,@aa:8	B	7		F	abs			6		2	rn		0														
	BCLR Rn,@aa:16	B	6		A	1		8	abs						6		2	rn		0								
	BCLR Rn,@aa:32	B	6		A	3		8	abs												6		2	rn		0		
BIAND	BIAND #xx:3,Rd	B	7		6	1	IMM	rd																				
	BIAND #xx:3,@ERd	B	7		C	0	erd	0	7		6	1	IMM	0														
	BIAND #xx:3,@aa:8	B	7		E	abs			7		6	1	IMM	0														
	BIAND #xx:3,@aa:16	B	6		A	1		0	abs						7		6	1	IMM	0								
	BIAND #xx:3,@aa:32	B	6		A	3		0	abs												7		6	1	IMM	0		
BILD	BILD #xx:3,Rd	B	7		7	1	IMM	rd																				
	BILD #xx:3,@ERd	B	7		C	0	erd	0	7		7	1	IMM	0														
	BILD #xx:3,@aa:8	B	7		E	abs			7		7	1	IMM	0														
	BILD #xx:3,@aa:16	B	6		A	1		0	abs						7		7	1	IMM	0								
	BILD #xx:3,@aa:32	B	6		A	3		0	abs												7		7	1	IMM	0		
BIOR	BIOR #xx:3,Rd	B	7		4	1	IMM	rd																				
	BIOR #xx:3,@ERd	B	7		C	0	erd	0	7		4	1	IMM	0														
	BIOR #xx:3,@aa:8	B	7		E	abs			7		4	1	IMM	0														
	BIOR #xx:3,@aa:16	B	6		A	1		0	abs						7		4	1	IMM	0								
	BIOR #xx:3,@aa:32	B	6		A	3		0	abs												7		4	1	IMM	0		

Instruction	Mnemonic	Size	Instruction Format: 1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
BIST	BIST #xx:3,Rd	B	6 7	1 IMM rd						
	BIST #xx:3,@ERd	B	7 D	0 erd 0	6 7	1 IMM 0				
	BIST #xx:3,@aa:8	B	7 F	abs	6 7	1 IMM 0				
	BIST #xx:3,@aa:16	B	6 A	1 8	abs		6 7	1 IMM 0		
	BIST #xx:3,@aa:32	B	6 A	3 8	abs				6 7	1 IM
BIXOR	BIXOR #xx:3,Rd	B	7 5	1 IMM rd						
	BIXOR #xx:3,@ERd	B	7 C	0 erd 0	7 5	1 IMM 0				
	BIXOR #xx:3,@aa:8	B	7 E	abs	7 5	1 IMM 0				
	BIXOR #xx:3,@aa:16	B	6 A	1 0	abs		7 5	1 IMM 0		
	BIXOR #xx:3,@aa:32	B	6 A	3 0	abs				7 5	1 IM
BLD	BLD #xx:3,Rd	B	7 7	0 IMM rd						
	BLD #xx:3,@ERd	B	7 C	0 erd 0	7 7	0 IMM 0				
	BLD #xx:3,@aa:8	B	7 E	abs	7 7	0 IMM 0				
	BLD #xx:3,@aa:16	B	6 A	1 0	abs		7 7	0 IMM 0		
	BLD #xx:3,@aa:32	B	6 A	3 0	abs				7 7	0 IM
BNOT	BNOT #xx:3,Rd	B	7 1	0 IMM rd						
	BNOT #xx:3,@ERd	B	7 D	0 erd 0	7 1	0 IMM 0				
	BNOT #xx:3,@aa:8	B	7 F	abs	7 1	0 IMM 0				
	BNOT #xx:3,@aa:16	B	6 A	1 8	abs		7 1	0 IMM 0		
	BNOT #xx:3,@aa:32	B	6 A	3 8	abs				7 1	0 IM
	BNOT Rn,Rd	B	6 1	rn rd						
	BNOT Rn,@ERd	B	7 D	0 erd 0	6 1	rn 0				
	BNOT Rn,@aa:8	B	7 F	abs	6 1	rn 0				
	BNOT Rn,@aa:16	B	6 A	1 8	abs		6 1	rn 0		
	BNOT Rn,@aa:32	B	6 A	3 8	abs				6 1	rn

RENESAS

Instruction	Mnemonic	Size	Instruction Format																													
			1st byte				2nd byte				3rd byte				4th byte				5th byte				6th byte				7th byte				8	
BOR	BOR #xx:3,Rd	B	7	4	0	IMM	rd																									
	BOR #xx:3,@ERd	B	7	C	0	erd	0				7	4	0	IMM	0																	
	BOR #xx:3,@aa:8	B	7	E	abs						7	4	0	IMM	0																	
	BOR #xx:3,@aa:16	B	6	A	1	0					abs								7	4	0	IMM	0									
	BOR #xx:3,@aa:32	B	6	A	3	0					abs																7	4	0	IM		
BSET	BSET #xx:3,Rd	B	7	0	0	IMM	rd																									
	BSET #xx:3,@ERd	B	7	D	0	erd	0				7	0	0	IMM	0																	
	BSET #xx:3,@aa:8	B	7	F	abs						7	0	0	IMM	0																	
	BSET #xx:3,@aa:16	B	6	A	1	8					abs								7	0	0	IMM	0									
	BSET #xx:3,@aa:32	B	6	A	3	8					abs																7	0	0	IM		
	BSET Rn,Rd	B	6	0	rn		rd																									
	BSET Rn,@ERd	B	7	D	0	erd	0				6	0	rn		0																	
	BSET Rn,@aa:8	B	7	F	abs						6	0	rn		0																	
	BSET Rn,@aa:16	B	6	A	1	8					abs								6	0	rn		0									
	BSET Rn,@aa:32	B	6	A	3	8					abs																6	0	rn			
BSR	BSR d:8	—	5	5	disp																											
	BSR d:16	—	5	C	0	0					disp																					
BST	BST #xx:3,Rd	B	6	7	0	IMM	rd																									
	BST #xx:3,@ERd	B	7	D	0	erd	0				6	7	0	IMM	0																	
	BST #xx:3,@aa:8	B	7	F	abs						6	7	0	IMM	0																	
	BST #xx:3,@aa:16	B	6	A	1	8					abs								6	7	0	IMM	0									
	BST #xx:3,@aa:32	B	6	A	3	8					abs																6	7	0	IM		
BTST	BTST #xx:3,Rd	B	7	3	0	IMM	rd																									
	BTST #xx:3,@ERd	B	7	C	0	erd	0				7	3	0	IMM	0																	
	BTST #xx:3,@aa:8	B	7	E	abs						7	3	0	IMM	0																	
	BTST #xx:3,@aa:16	B	6	A	1	0					abs								7	3	0	IMM	0									
	BTST #xx:3,@aa:32	B	6	A	3	0					abs																7	3	0	IM		
	BTST Rn,Rd	B	6	3	rn		rd																									

Instruction	Mnemonic	Size	Instruction Format 1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8
BTST	BTST Rn,@aa:8	B	7 E	abs	6 3	rn 0				
	BTST Rn,@aa:16	B	6 A	1 0	abs		6 3	rn 0		
	BTST Rn,@aa:32	B	6 A	3 0	abs				6 3	r
BXOR	BXOR #xx:3,Rd	B	7 5	0 IMM rd						
	BXOR #xx:3,@ERd	B	7 C	0 erd 0	7 5	0 IMM 0				
	BXOR #xx:3,@aa:8	B	7 E	abs	7 5	0 IMM 0				
	BXOR #xx:3,@aa:16	B	6 A	1 0	abs		7 5	0 IMM 0		
	BXOR #xx:3,@aa:32	B	6 A	3 0	abs				7 5	0 I
CLRMAC	CLRMAC	—	0 1	A 0						
CMP	CMP.B #xx:8,Rd	B	A rd	IMM						
	CMP.B Rs,Rd	B	1 C	rs rd						
	CMP.W #xx:16,Rd	W	7 9	2 rd	IMM					
	CMP.W Rs,Rd	W	1 D	rs rd						
	CMP.L #xx:32,ERd	L	7 A	2 0 erd	IMM					
	CMP.L ERs,ERd	L	1 F	1 ers 0 erd						
DAA	DAA Rd	B	0 F	0 rd						
DAS	DAS Rd	B	1 F	0 rd						
DEC	DEC.B Rd	B	1 A	0 rd						
	DEC.W #1,Rd	W	1 B	5 rd						
	DEC.W #2,Rd	W	1 B	D rd						
	DEC.L #1,ERd	L	1 B	7 0 erd						
	DEC.L #2,ERd	L	1 B	F 0 erd						
DIVXS	DIVXS.B Rs,Rd	B	0 1	D 0	5 1	rs rd				
	DIVXS.W Rs,ERd	W	0 1	D 0	5 3	rs 0 erd				
DIVXU	DIVXU.B Rs,Rd	B	5 1	rs rd						
	DIVXU.W Rs,ERd	W	5 3	rs 0 erd						
EEPMOV	EEPMOV.B	—	7 B	5 C	5 9	8 F				
	EEPMOV.W	—	7 B	D 4	5 9	8 F				

Instruction	Mnemonic	Size	Instruction Format: 1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8
EXTS	EXTS.W Rd	W	1 7	D rd						
	EXTS.L ERd	L	1 7	F 0 erd						
EXTU	EXTU.W Rd	W	1 7	5 rd						
	EXTU.L ERd	L	1 7	7 0 erd						
INC	INC.B Rd	B	0 A	0 rd						
	INC.W #1,Rd	W	0 B	5 rd						
	INC.W #2,Rd	W	0 B	D rd						
	INC.L #1,ERd	L	0 B	7 0 erd						
	INC.L #2,ERd	L	0 B	F 0 erd						
JMP	JMP @ERn	—	5 9	0 ern 0						
	JMP @aa:24	—	5 A	abs						
	JMP @@aa:8	—	5 B	abs						
JSR	JSR @ERn	—	5 D	0 ern 0						
	JSR @aa:24	—	5 E	abs						
	JSR @@aa:8	—	5 F	abs						
LDC	LDC #xx:8,CCR	B	0 7	IMM						
	LDC #xx:8,EXR	B	0 1	4 1	0 7	IMM				
	LDC Rs,CCR	B	0 3	0 rs						
	LDC Rs,EXR	B	0 3	1 rs						
	LDC @ERs,CCR	W	0 1	4 0	6 9	0 ers 0				
	LDC @ERs,EXR	W	0 1	4 1	6 9	0 ers 0				
	LDC @(d:16,ERs),CCR	W	0 1	4 0	6 F	0 ers 0	disp			
	LDC @(d:16,ERs),EXR	W	0 1	4 1	6 F	0 ers 0	disp			
	LDC @(d:32,ERs),CCR	W	0 1	4 0	7 8	0 ers 0	6 B	2 0		
	LDC @(d:32,ERs),EXR	W	0 1	4 1	7 8	0 ers 0	6 B	2 0		
	LDC @ERs+,CCR	W	0 1	4 0	6 D	0 ers 0				
	LDC @ERs+,EXR	W	0 1	4 1	6 D	0 ers 0				
	LDC @aa:16,CCR	W	0 1	4 0	6 B	0 0	disp			

Instruction	Mnemonic	Size	Instruction Format																									
			1st byte				2nd byte				3rd byte				4th byte				5th byte				6th byte				7th byte			
LDC	LDC @aa:32,CCR	W	0	1			4	0			6	B			2	0			abs											
	LDC @aa:32,EXR	W	0	1			4	1			6	B			2	0			abs											
LDM	LDM.L @SP+, (ERn-ERn+1)	L	0	1			1	0			6	D			7	0	ern+1													
	LDM.L @SP+, (ERn-ERn+2)	L	0	1			2	0			6	D			7	0	ern+2													
	LDM.L @SP+, (ERn-ERn+3)	L	0	1			3	0			6	D			7	0	ern+3													
LDMAC	LDMAC ERs,MACH	L	0	3			2	0	ers																					
	LDMAC ERs,MACL	L	0	3			3	0	ers																					
MAC	MAC @ERn+,@ERm+	—	0	1			6	0			6	D			0	ern	0	erm												
MOV	MOV.B #xx:8,Rd	B	F	rd			IMM																							
	MOV.B Rs,Rd	B	0	C			rs	rd																						
	MOV.B @ERs,Rd	B	6	8			0	ers	rd																					
	MOV.B @(d:16,ERs),Rd	B	6	E			0	ers	rd		disp																			
	MOV.B @(d:32,ERs),Rd	B	7	8			0	ers	0		6	A			2	rd			disp											
	MOV.B @ERs+,Rd	B	6	C			0	ers	rd																					
	MOV.B @aa:8,Rd	B	2	rd			abs																							
	MOV.B @aa:16,Rd	B	6	A			0	rd			abs																			
	MOV.B @aa:32,Rd	B	6	A			2	rd			abs																			
	MOV.B Rs,@ERd	B	6	8			1	erd	rs																					
	MOV.B Rs,@(d:16,ERd)	B	6	E			1	erd	rs		disp																			
	MOV.B Rs,@(d:32,ERd)	B	7	8			0	erd	0		6	A			A	rs			disp											
	MOV.B Rs,@-ERd	B	6	C			1	erd	rs																					
	MOV.B Rs,@aa:8	B	3	rs			abs																							
	MOV.B Rs,@aa :16	B	6	A			8	rs			abs																			
	MOV.B Rs,@aa:32	B	6	A			A	rs			abs																			
	MOV.W #xx:16,Rd	W	7	9			0	rd			IMM																			
	MOV.W Rs,Rd	W	0	D			rs	rd																						
	MOV.W @ERs,Rd	W	6	9			0	ers	rd																					
	MOV.W @(d:16,ERs),Rd	W	6	F			0	ers	rd		disp																			

Instruction	Mnemonic	Size	Instruction Format							
			1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8
MOV	MOV.W @ERs+,Rd	W	6 D	0 ers rd						
	MOV.W @aa:16,Rd	W	6 B	0 rd	abs					
	MOV.W @aa:32,Rd	W	6 B	2 rd	abs					
	MOV.W Rs,@ERd	W	6 9	1 erd rs						
	MOV.W Rs,@(d:16,ERd)	W	6 F	1 erd rs	disp					
	MOV.W Rs,@(d:32,ERd)	W	7 8	0 erd 0	6 B	A rs	disp			
	MOV.W Rs,@-ERd	W	6 D	1 erd rs						
	MOV.W Rs,@aa:16	W	6 B	8 rs	abs					
	MOV.W Rs,@aa:32	W	6 B	A rs	abs					
	MOV.L #xx:32,Rd	L	7 A	0 0 erd	IMM					
	MOV.L ERs,ERd	L	0 F	1 ers 0 erd						
	MOV.L @ERs,ERd	L	0 1	0 0	6 9	0 ers 0 erd				
	MOV.L @(d:16,ERs),ERd	L	0 1	0 0	6 F	0 ers 0 erd	disp			
	MOV.L @(d:32,ERs),ERd	L	0 1	0 0	7 8	0 ers 0	6 B	2 0 erd		
	MOV.L @ERs+,ERd	L	0 1	0 0	6 D	0 ers 0 erd				
	MOV.L @aa:16 ,ERd	L	0 1	0 0	6 B	0 0 erd	abs			
	MOV.L @aa:32 ,ERd	L	0 1	0 0	6 B	2 0 erd	abs			
	MOV.L ERs,@ERd	L	0 1	0 0	6 9	1 erd 0 ers				
	MOV.L ERs,@(d:16,ERd)	L	0 1	0 0	6 F	1 erd 0 ers	disp			
	MOV.L ERs,@(d:32,ERd)*1	L	0 1	0 0	7 8	0 erd 0	6 B	A 0 ers		
	MOV.L ERs,@-ERd	L	0 1	0 0	6 D	1 erd 0 ers				
	MOV.L ERs,@aa:16	L	0 1	0 0	6 B	8 0 ers	abs			
	MOV.L ERs,@aa:32	L	0 1	0 0	6 B	A 0 ers	abs			
MOVFPE	MOVFPE @aa:16,Rd	B	Cannot be used in the H8S/2626 Group or H8S/2623 Group.							
MOVTPE	MOVTPE Rs,@aa:16	B								
MULXS	MULXS.B Rs,Rd	B	0 1	C 0	5 0	rs rd				
	MULXS.W Rs,ERd	W	0 1	C 0	5 2	rs 0 erd				
MULXU	MULXU.B Rs,Rd	B	5 0	rs rd						

Instruction	Mnemonic	Size	Instruction Format								
			1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8
NEG	NEG.B Rd	B	1 7	8 rd						
	NEG.W Rd	W	1 7	9 rd						
	NEG.L ERd	L	1 7	B 0 erd						
NOP	NOP	—	0 0	0 0						
NOT	NOT.B Rd	B	1 7	0 rd						
	NOT.W Rd	W	1 7	1 rd						
	NOT.L ERd	L	1 7	3 0 erd						
OR	OR.B #xx:8,Rd	B	C rd	IMM						
	OR.B Rs,Rd	B	1 4	rs rd						
	OR.W #xx:16,Rd	W	7 9	4 rd	IMM					
	OR.W Rs,Rd	W	6 4	rs rd						
	OR.L #xx:32,ERd	L	7 A	4 0 erd	IMM					
	OR.L ERs,ERd	L	0 1	F 0	6 4	0 ers 0 erd				
ORC	ORC #xx:8,CCR	B	0 4	IMM						
	ORC #xx:8,EXR	B	0 1	4 1	0 4	IMM				
POP	POP.W Rn	W	6 D	7 m						
	POP.L ERn	L	0 1	0 0	6 D	7 0 ern				
PUSH	PUSH.W Rn	W	6 D	F m						
	PUSH.L ERn	L	0 1	0 0	6 D	F 0 ern				
ROTL	ROTL.B Rd	B	1 2	8 rd						
	ROTL.B #2, Rd	B	1 2	C rd						
	ROTL.W Rd	W	1 2	9 rd						
	ROTL.W #2, Rd	W	1 2	D rd						
	ROTL.L ERd	L	1 2	B 0 erd						
	ROTL.L #2, ERd	L	1 2	F 0 erd						

RENESAS

Instruction	Mnemonic	Size	Instruction Format							
			1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8
ROTR	ROTR.B Rd	B	1 3	8 rd						
	ROTR.B #2, Rd	B	1 3	C rd						
	ROTR.W Rd	W	1 3	9 rd						
	ROTR.W #2, Rd	W	1 3	D rd						
	ROTR.L ERd	L	1 3	B 0 erd						
	ROTR.L #2, ERd	L	1 3	F 0 erd						
ROTXL	ROTXL.B Rd	B	1 2	0 rd						
	ROTXL.B #2, Rd	B	1 2	4 rd						
	ROTXL.W Rd	W	1 2	1 rd						
	ROTXL.W #2, Rd	W	1 2	5 rd						
	ROTXL.L ERd	L	1 2	3 0 erd						
	ROTXL.L #2, ERd	L	1 2	7 0 erd						
ROTXR	ROTXR.B Rd	B	1 3	0 rd						
	ROTXR.B #2, Rd	B	1 3	4 rd						
	ROTXR.W Rd	W	1 3	1 rd						
	ROTXR.W #2, Rd	W	1 3	5 rd						
	ROTXR.L ERd	L	1 3	3 0 erd						
	ROTXR.L #2, ERd	L	1 3	7 0 erd						
RTE	RTE	—	5 6	7 0						
RTS	RTS	—	5 4	7 0						
SHAL	SHAL.B Rd	B	1 0	8 rd						
	SHAL.B #2, Rd	B	1 0	C rd						
	SHAL.W Rd	W	1 0	9 rd						
	SHAL.W #2, Rd	W	1 0	D rd						
	SHAL.L ERd	L	1 0	B 0 erd						
	SHAL.L #2, ERd	L	1 0	F 0 erd						

Instruction	Mnemonic	Size	Instruction Format							
			1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8
SHAR	SHAR.B Rd	B	1 1	8 rd						
	SHAR.B #2, Rd	B	1 1	C rd						
	SHAR.W Rd	W	1 1	9 rd						
	SHAR.W #2, Rd	W	1 1	D rd						
	SHAR.L ERd	L	1 1	B 0 erd						
	SHAR.L #2, ERd	L	1 1	F 0 erd						
SHLL	SHLL.B Rd	B	1 0	0 rd						
	SHLL.B #2, Rd	B	1 0	4 rd						
	SHLL.W Rd	W	1 0	1 rd						
	SHLL.W #2, Rd	W	1 0	5 rd						
	SHLL.L ERd	L	1 0	3 0 erd						
	SHLL.L #2, ERd	L	1 0	7 0 erd						
SHLR	SHLR.B Rd	B	1 1	0 rd						
	SHLR.B #2, Rd	B	1 1	4 rd						
	SHLR.W Rd	W	1 1	1 rd						
	SHLR.W #2, Rd	W	1 1	5 rd						
	SHLR.L ERd	L	1 1	3 0 erd						
	SHLR.L #2, ERd	L	1 1	7 0 erd						
SLEEP	SLEEP	—	0 1	8 0						
STC	STC.B CCR,Rd	B	0 2	0 rd						
	STC.B EXR,Rd	B	0 2	1 rd						
	STC.W CCR,@ERd	W	0 1	4 0	6 9	1 erd 0				
	STC.W EXR,@ERd	W	0 1	4 1	6 9	1 erd 0				
	STC.W CCR,@(d:16,ERd)	W	0 1	4 0	6 F	1 erd 0	disp			
	STC.W EXR,@(d:16,ERd)	W	0 1	4 1	6 F	1 erd 0	disp			
	STC.W CCR,@(d:32,ERd)	W	0 1	4 0	7 8	0 erd 0	6 B	A 0		
	STC.W EXR,@(d:32,ERd)	W	0 1	4 1	7 8	0 erd 0	6 B	A 0		
	STC.W CCR,@-ERd	W	0 1	4 0	6 D	1 erd 0				

RENESAS

Instruction	Mnemonic	Size	Instruction Format																											
			1st byte				2nd byte				3rd byte				4th byte				5th byte				6th byte				7th byte			
STC	STC.W CCR,@aa:16	W	0	1	4	0	6	B	8	0	abs																			
	STC.W EXR,@aa:16	W	0	1	4	1	6	B	8	0	abs																			
	STC.W CCR,@aa:32	W	0	1	4	0	6	B	A	0	abs																			
	STC.W EXR,@aa:32	W	0	1	4	1	6	B	A	0	abs																			
STM	STM.L(ERn-ERn+1), @-SP	L	0	1	1	0	6	D	F	0	ern																			
	STM.L (ERn-ERn+2), @-SP	L	0	1	2	0	6	D	F	0	ern																			
	STM.L (ERn-ERn+3), @-SP	L	0	1	3	0	6	D	F	0	ern																			
STMAC	STMAC MACH,ERd	L	0	2	2	0	ers																							
	STMAC MACL,ERd	L	0	2	3	0	ers																							
SUB	SUB.B Rs,Rd	B	1	8	rs		rd																							
	SUB.W #xx:16,Rd	W	7	9	3		rd		IMM																					
	SUB.W Rs,Rd	W	1	9	rs		rd																							
	SUB.L #xx:32,ERd	L	7	A	3	0	erd		IMM																					
	SUB.L ERs,ERd	L	1	A	1	ers	0	erd																						
SUBS	SUBS #1,ERd	L	1	B	0		0	erd																						
	SUBS #2,ERd	L	1	B	8		0	erd																						
	SUBS #4,ERd	L	1	B	9		0	erd																						
SUBX	SUBX #xx:8,Rd	B	B	rd	IMM																									
	SUBX Rs,Rd	B	1	E	rs		rd																							
TAS	TAS @ERd*2	B	0	1	E		0		7	B	0	erd	C																	
TRAPA	TRAPA #x:2	—	5	7	00	IMM	0																							
XOR	XOR.B #xx:8,Rd	B	D	rd	IMM																									
	XOR.B Rs,Rd	B	1	5	rs		rd																							
	XOR.W #xx:16,Rd	W	7	9	5		rd		IMM																					
	XOR.W Rs,Rd	W	6	5	rs		rd																							
	XOR.L #xx:32,ERd	L	7	A	5	0	erd		IMM																					
	XOR.L ERs,ERd	L	0	1	F		0		6	5	0	ers	0	erd																

Instruc-tion	Mnemonic	Size	Instruction Format															
			1st byte		2nd byte		3rd byte		4th byte	5th byte	6th byte	7th byte	8					
XORC	XORC #xx:8,CCR	B	0	5	IMM													
	XORC #xx:8,EXR	B	0	1	4	1	0	5	IMM									

Notes: 1. Bit 7 of the 4th byte of the MOV.L ERs, @(d:32,ERd) instruction can be either 1 or 0.
2. Only register ER0, ER1, ER4, or ER5 should be used when using the TAS instruction.

Legend:

IMM:	Immediate data (2, 3, 8, 16, or 32 bits)
abs:	Absolute address (8, 16, 24, or 32 bits)
disp:	Displacement (8, 16, or 32 bits)
rs, rd, rn:	Register field (4 bits specifying an 8-bit or 16-bit register. The symbols rs, rd, and rn correspond to ope
ers, erd, ern, erm:	Register field (3 bits specifying an address register or 32-bit register. The symbols ers, erd, ern, and er symbols ERs, ERd, ERn, and ERm.)

The register fields specify general registers as follows.

Address Register 32-Bit Register		16-Bit Register		8-Bit Register	
Register Field	General Register	Register Field	General Register	Register Field	General Register
000	ER0	0000	R0	0000	R0H
001	ER1	0001	R1	0001	R1H
•	•	•	•	•	•
•	•	•	•	•	•
•	•	•	•	•	•
111	ER7	0111	R7	0111	R7H
		1000	E0	1000	R0L
		1001	E1	1001	R1L
		•	•	•	•
		•	•	•	•
		•	•	•	•
		1111	E7	1111	R7L

Instruction code

1st byte		2nd byte	
AH	AL	BH	BL

Instruction when most significant bit of BH is 0.

Instruction when most significant bit of BH is 1.

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

Note: * Cannot be used in the H8S/2626 Group or H8S/2623 Group.

Instruction code

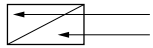
1st byte		2nd byte	
AH	AL	BH	BL

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

RENESAS

Instruction code

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



AH AL BH BL CH \ CL	0	1	2	3	4	5	6	7	8	9	A	B	C
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 specification field.
2. aa is the absolute address specification.

Instruction code	1st byte		2nd byte		3rd byte		4th byte		5th byte		6th byte	
	AH	AL	BH	BL	CH	CL	DH	DL	EH	EL	FH	FL

Instruction whe
Instruction whe

EL / AHALBHBLCHCLDHDLEH	0	1	2	3	4	5	6	7	8	9	A	B	C
6A10aaaa6*				BTST									
6A10aaaa7*					BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD					
6A18aaaa6*	BSET	BNOT	BCLR					BST / BIST					
6A18aaaa7*													

Instruction code	1st byte		2nd byte		3rd byte		4th byte		5th byte		6th byte		7th byte		8th b
	AH	AL	BH	BL	CH	CL	DH	DL	EH	EL	FH	FL	GH	GL	HH

Instruction whe
Instruction whe

GL / AHALBHBL ... FHFLGH	0	1	2	3	4	5	6	7	8	9	A	B	C
6A30aaaaaaaa6*				BTST									
6A30aaaaaaaa7*					BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD					
6A38aaaaaaaa6*	BSET	BNOT	BCLR					BST / BIST					
6A38aaaaaaaa7*													

Execution states = $I \times S_I + J \times S_J + K \times S_K + L \times S_L + M \times S_M + N \times S_N$

Examples: Advanced mode, program code and stack located in external memory, on-c
supporting modules accessed in two states with 8-bit bus width, external devices access
states with one wait state and 16-bit bus width.

1. BSET #0, @FFFFC7:8

   From table A.5:

   $I = L = 2, \quad J = K = M = N = 0$

   From table A.4:

   $S_I = 4, \quad S_L = 2$

   Number of states required for execution = $2 \times 4 + 2 \times 2 = 12$

2. JSR @@30

   From table A.5:

   $I = J = K = 2, \quad L = M = N = 0$

   From table A.4:

   $S_I = S_J = S_K = 4$

   Number of states required for execution = $2 \times 4 + 2 \times 4 + 2 \times 4 = 24$

Branch address read	S_J						
Stack operation	S_K						
Byte data access	S_L		2		2	3 + m	
Word data access	S_M		4		4	6 + 2m	
Internal operation	S_N	1	1	1	1	1	1

Legend:

m: Number of wait states inserted into external device access

	ADD.L ERs,ERd	1		
ADDS	ADDS #1/2/4,ERd	1		
ADDX	ADDX #xx:8,Rd	1		
	ADDX Rs,Rd	1		
AND	AND.B #xx:8,Rd	1		
	AND.B Rs,Rd	1		
	AND.W #xx:16,Rd	2		
	AND.W Rs,Rd	1		
	AND.L #xx:32,ERd	3		
	AND.L ERs,ERd	2		
ANDC	ANDC #xx:8,CCR	1		
	ANDC #xx:8,EXR	2		
BAND	BAND #xx:3,Rd	1		
	BAND #xx:3,@ERd	2		1
	BAND #xx:3,@aa:8	2		1
	BAND #xx:3,@aa:16	3		1
	BAND #xx:3,@aa:32	4		1
Bcc	BRA d:8 (BT d:8)	2		
	BRN d:8 (BF d:8)	2		
	BHI d:8	2		
	BLS d:8	2		
	BCC d:8 (BHS d:8)	2		
	BCS d:8 (BLO d:8)	2		
	BNE d:8	2		
	BEQ d:8	2		
	BVC d:8	2		
	BVS d:8	2		
	BPL d:8	2		
	BMI d:8	2		
	BGE d:8	2		
	BLT d:8	2		
	BGT d:8	2		
	BLE d:8	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	
BCLR	BCLR #xx:3,Rd	1	
	BCLR #xx:3,@ERd	2	2
	BCLR #xx:3,@aa:8	2	2
	BCLR #xx:3,@aa:16	3	2
	BCLR #xx:3,@aa:32	4	2
	BCLR Rn,Rd	1	
	BCLR Rn,@ERd	2	2
	BCLR Rn,@aa:8	2	2
	BCLR Rn,@aa:16	3	2
	BCLR Rn,@aa:32	4	2
BIAND	BIAND #xx:3,Rd	1	
	BIAND #xx:3,@ERd	2	1
	BIAND #xx:3,@aa:8	2	1
	BIAND #xx:3,@aa:16	3	1
	BIAND #xx:3,@aa:32	4	1
BILD	BILD #xx:3,Rd	1	
	BILD #xx:3,@ERd	2	1
	BILD #xx:3,@aa:8	2	1
	BILD #xx:3,@aa:16	3	1
	BILD #xx:3,@aa:32	4	1

	BIST #xx:3,@aa:8	2	2
	BIST #xx:3,@aa:16	3	2
	BIST #xx:3,@aa:32	4	2
BIXOR	BIXOR #xx:3,Rd	1	
	BIXOR #xx:3,@ERd	2	1
	BIXOR #xx:3,@aa:8	2	1
	BIXOR #xx:3,@aa:16	3	1
	BIXOR #xx:3,@aa:32	4	1
BLD	BLD #xx:3,Rd	1	
	BLD #xx:3,@ERd	2	1
	BLD #xx:3,@aa:8	2	1
	BLD #xx:3,@aa:16	3	1
	BLD #xx:3,@aa:32	4	1
BNOT	BNOT #xx:3,Rd	1	
	BNOT #xx:3,@ERd	2	2
	BNOT #xx:3,@aa:8	2	2
	BNOT #xx:3,@aa:16	3	2
	BNOT #xx:3,@aa:32	4	2
	BNOT Rn,Rd	1	
	BNOT Rn,@ERd	2	2
	BNOT Rn,@aa:8	2	2
	BNOT Rn,@aa:16	3	2
	BNOT Rn,@aa:32	4	2
BOR	BOR #xx:3,Rd	1	
	BOR #xx:3,@ERd	2	1
	BOR #xx:3,@aa:8	2	1
	BOR #xx:3,@aa:16	3	1
	BOR #xx:3,@aa:32	4	1

	BSET Rn,@aa:8	2		2
	BSET Rn,@aa:16	3		2
	BSET Rn,@aa:32	4		2
BSR	BSR d:8	2	2	
	BSR d:16	2	2	
BST	BST #xx:3,Rd	1		
	BST #xx:3,@ERd	2		2
	BST #xx:3,@aa:8	2		2
	BST #xx:3,@aa:16	3		2
	BST #xx:3,@aa:32	4		2
BTST	BTST #xx:3,Rd	1		
	BTST #xx:3,@ERd	2		1
	BTST #xx:3,@aa:8	2		1
	BTST #xx:3,@aa:16	3		1
	BTST #xx:3,@aa:32	4		1
	BTST Rn,Rd	1		
	BTST Rn,@ERd	2		1
	BTST Rn,@aa:8	2		1
	BTST Rn,@aa:16	3		1
	BTST Rn,@aa:32	4		1
BXOR	BXOR #xx:3,Rd	1		
	BXOR #xx:3,@ERd	2		1
	BXOR #xx:3,@aa:8	2		1
	BXOR #xx:3,@aa:16	3		1
	BXOR #xx:3,@aa:32	4		1
CLRMAC	CLRMAC	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		

	DIVXU.W Rs,ERd	1					
EEPMOV	EEPMOV.B	2				2n + 2*2	
	EEPMOV.W	2				2n + 2*2	
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	2				
JSR	JSR @ERn	2			2		
	JSR @aa:24	2			2		
	JSR @@aa:8	2	2		2		
LDC	LDC #xx:8,CCR	1					
	LDC #xx:8,EXR	2					
	LDC Rs,CCR	1					
	LDC Rs,EXR	1					
	LDC @ERs,CCR	2					1
	LDC @ERs,EXR	2					1
	LDC @(d:16,ERs),CCR	3					1
	LDC @(d:16,ERs),EXR	3					1
	LDC @(d:32,ERs),CCR	5					1
	LDC @(d:32,ERs),EXR	5					1
	LDC @ERs+,CCR	2					1
	LDC @ERs+,EXR	2					1
	LDC @aa:16,CCR	3					1
	LDC @aa:16,EXR	3					1
	LDC @aa:32,CCR	4					1
	LDC @aa:32,EXR	4					1

MOV.B Rs,Rd	1			
MOV.B @ERs,Rd	1		1	
MOV.B @(d:16,ERs),Rd	2		1	
MOV.B @(d:32,ERs),Rd	4		1	
MOV.B @ERs+,Rd	1		1	
MOV.B @aa:8,Rd	1		1	
MOV.B @aa:16,Rd	2		1	
MOV.B @aa:32,Rd	3		1	
MOV.B Rs,@ERd	1		1	
MOV.B Rs,@(d:16,ERd)	2		1	
MOV.B Rs,@(d:32,ERd)	4		1	
MOV.B Rs,@-ERd	1		1	
MOV.B Rs,@aa:8	1		1	
MOV.B Rs,@aa:16	2		1	
MOV.B Rs,@aa:32	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:32,ERs),Rd	4			1
MOV.W @ERs+,Rd	1			1
MOV.W @aa:16,Rd	2			1
MOV.W @aa:32,Rd	3			1
MOV.W Rs,@ERd	1			1
MOV.W Rs,@(d:16,ERd)	2			1
MOV.W Rs,@(d:32,ERd)	4			1
MOV.W Rs,@-ERd	1			1
MOV.W Rs,@aa:16	2			1
MOV.W Rs,@aa:32	3			1
MOV.L #xx:32,ERd	3			
MOV.L ERs,ERd	1			
MOV.L @ERs,ERd	2			2

	MOV.L ERs,@(d:32,ERd)	5	2
	MOV.L ERs,@-ERd	2	2
	MOV.L ERs,@aa:16	3	2
	MOV.L ERs,@aa:32	4	2
MOVFPE	MOVFPE @:aa:16,Rd	Can not be used in the H8S/2626 Group or H8S/2623 Group.	
MOVTPE	MOVTPE Rs,@:aa:16		
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	
OR	OR.B #xx:8,Rd	1	
	OR.B Rs,Rd	1	
	OR.W #xx:16,Rd	2	
	OR.W Rs,Rd	1	
	OR.L #xx:32,ERd	3	
	OR.L ERs,ERd	2	
ORC	ORC #xx:8,CCR	1	
	ORC #xx:8,EXR	2	
POP	POP.W Rn	1	1
	POP.L ERn	2	2
PUSH	PUSH.W Rn	1	1
	PUSH.L ERn	2	2

	ROTR.B #2,Rd	1	
	ROTR.W Rd	1	
	ROTR.W #2,Rd	1	
	ROTR.L ERd	1	
	ROTR.L #2,ERd	1	
ROTXL	ROTXL.B Rd	1	
	ROTXL.B #2,Rd	1	
	ROTXL.W Rd	1	
	ROTXL.W #2,Rd	1	
	ROTXL.L ERd	1	
	ROTXL.L #2,ERd	1	
ROTXR	ROTXR.B Rd	1	
	ROTXR.B #2,Rd	1	
	ROTXR.W Rd	1	
	ROTXR.W #2,Rd	1	
	ROTXR.L ERd	1	
	ROTXR.L #2,ERd	1	
RTE	RTE	2	2/3*1
RTS	RTS	2	2
SHAL	SHAL.B Rd	1	
	SHAL.B #2,Rd	1	
	SHAL.W Rd	1	
	SHAL.W #2,Rd	1	
	SHAL.L ERd	1	
	SHAL.L #2,ERd	1	
SHAR	SHAR.B Rd	1	
	SHAR.B #2,Rd	1	
	SHAR.W Rd	1	
	SHAR.W #2,Rd	1	
	SHAR.L ERd	1	
	SHAR.L #2,ERd	1	

	SHLR.B #2,Rd	1			
	SHLR.W Rd	1			
	SHLR.W #2,Rd	1			
	SHLR.L ERd	1			
	SHLR.L #2,ERd	1			
SLEEP	SLEEP	1			
STC	STC.B CCR,Rd	1			
	STC.B EXR,Rd	1			
	STC.W CCR,@ERd	2			1
	STC.W EXR,@ERd	2			1
	STC.W CCR,@(d:16,ERd)	3			1
	STC.W EXR,@(d:16,ERd)	3			1
	STC.W CCR,@(d:32,ERd)	5			1
	STC.W EXR,@(d:32,ERd)	5			1
	STC.W CCR,@-ERd	2			1
	STC.W EXR,@-ERd	2			1
	STC.W CCR,@aa:16	3			1
	STC.W EXR,@aa:16	3			1
	STC.W CCR,@aa:32	4			1
	STC.W EXR,@aa:32	4			1
STM	STM.L (ERn-ERn+1),@-SP	2	4		
	STM.L (ERn-ERn+2),@-SP	2	6		
	STM.L (ERn-ERn+3),@-SP	2	8		
STMAC*3	STMAC MACH,ERd	1			
	STMAC MACL,ERd	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			

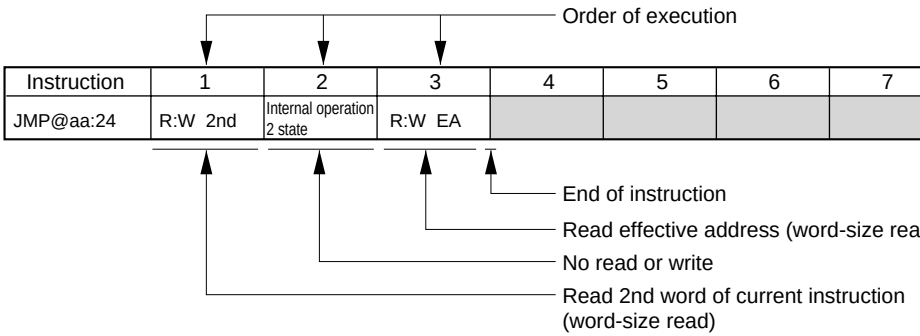
	XOR.L ERs,ERd	2
XORC	XORC #xx:8,CCR	1
	XORC #xx:8,EXR	2

Notes: 1. 2 when EXR is invalid, 3 when EXR is valid.
2. 5 for concatenated execution, 4 otherwise.
3. An internal operation may require between 0 and 3 additional states, deper preceding instruction.
4. Only register ER0, ER1, ER4, or ER5 should be used when using the TAS

A.5 Bus States during Instruction Execution

Table A.6 indicates the types of cycles that occur during instruction execution by the C table A.4 for the number of states per cycle.

How to Read the Table:



3rd	Address of 3rd word (5th and 6th bytes)
4th	Address of 4th word (7th and 8th bytes)
5th	Address of 5th word (9th and 10th bytes)
NEXT	Address of next instruction
EA	Effective address
VEC	Vector address

Figure A.1 shows timing waveforms for the address bus and the $\overline{RD}$, $\overline{HWR}$, and $\overline{LWR}$ s during execution of the above instruction with an 8-bit bus, using three-state access wit states.

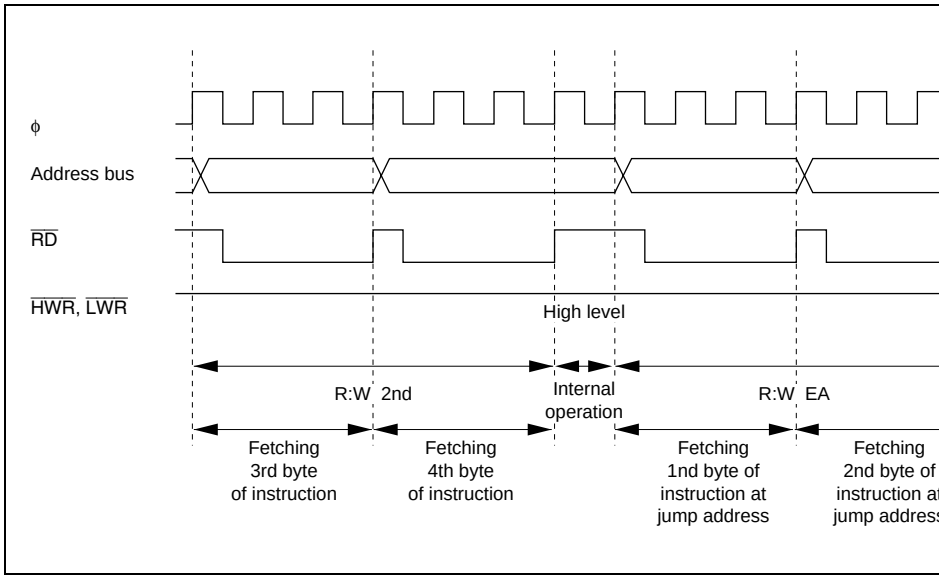


Figure A.1 Address Bus, $\overline{RD}$, $\overline{HWR}$, and $\overline{LWR}$ Timing (8-Bit Bus, Three-State Access, No Wait States)

Instruction	1	2	3	4	5	6	7
ADD.B #xx:8,Rd	R:W NEXT						
ADD.B Rs,Rd	R:W NEXT						
ADD.W #xx:16,Rd	R:W 2nd	R:W NEXT					
ADD.W Rs,Rd	R:W NEXT						
ADD.L #xx:32,ERd	R:W 2nd	R:W 3rd	R:W NEXT				
ADD.L ERs,ERd	R:W NEXT						
ADDS #1/2/4,ERd	R:W NEXT						
ADDX #xx:8,Rd	R:W NEXT						
ADDX Rs,Rd	R:W NEXT						
AND.B #xx:8,Rd	R:W NEXT						
AND.B Rs,Rd	R:W NEXT						
AND.W #xx:16,Rd	R:W 2nd	R:W NEXT					
AND.W Rs,Rd	R:W NEXT						
AND.L #xx:32,ERd	R:W 2nd	R:W 3rd	R:W NEXT				
AND.L ERs,ERd	R:W 2nd	R:W NEXT					
ANDC #xx:8,CCR	R:W NEXT						
ANDC #xx:8,EXR	R:W 2nd	R:W NEXT					
BAND #xx:3,Rd	R:W NEXT						
BAND #xx:3,@ERd	R:W 2nd	R:B EA	R:W:M NEXT				
BAND #xx:3,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT				
BAND #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT			
BAND #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT		
BRA d:8 (BT d:8)	R:W NEXT	R:W EA					
BRN d:8 (BF d:8)	R:W NEXT	R:W EA					
BHI d:8	R:W NEXT	R:W EA					
BLS d:8	R:W NEXT	R:W EA					
BCC d:8 (BHS d:8)	R:W NEXT	R:W EA					
BCS d:8 (BLO d:8)	R:W NEXT	R:W EA					
BNE d:8	R:W NEXT	R:W EA					
BEQ d:8	R:W NEXT	R:W EA					
BVC d:8	R:W NEXT	R:W EA					
BVS d:8	R:W NEXT	R:W EA					
BPL d:8	R:W NEXT	R:W EA					
BMI d:8	R:W NEXT	R:W EA					
BGE d:8	R:W NEXT	R:W EA					
BLT d:8	R:W NEXT	R:W EA					

Instruction	1	2	3	4	5	6	7
BLE d:8	R:W NEXT	R:W EA					
BRA d:16 (BT d:16)	R:W 2nd	Internal operation, 1 state	R:W EA				
BRN d:16 (BF d:16)	R:W 2nd	Internal operation, 1 state	R:W EA				
BHI d:16	R:W 2nd	Internal operation, 1 state	R:W EA				
BLS d:16	R:W 2nd	Internal operation, 1 state	R:W EA				
BCC d:16 (BHS d:16)	R:W 2nd	Internal operation, 1 state	R:W EA				
BCS d:16 (BLO d:16)	R:W 2nd	Internal operation, 1 state	R:W EA				
BNE d:16	R:W 2nd	Internal operation, 1 state	R:W EA				
BEQ d:16	R:W 2nd	Internal operation, 1 state	R:W EA				
BVC d:16	R:W 2nd	Internal operation, 1 state	R:W EA				
BVS d:16	R:W 2nd	Internal operation, 1 state	R:W EA				
BPL d:16	R:W 2nd	Internal operation, 1 state	R:W EA				
BMI d:16	R:W 2nd	Internal operation, 1 state	R:W EA				
BGE d:16	R:W 2nd	Internal operation, 1 state	R:W EA				
BLT d:16	R:W 2nd	Internal operation, 1 state	R:W EA				
BGT d:16	R:W 2nd	Internal operation, 1 state	R:W EA				
BLE d:16	R:W 2nd	Internal operation, 1 state	R:W EA				
BCLR #xx:3,Rd	R:W NEXT						
BCLR #xx:3,@ERd	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA			

Instruction	1	2	3	4	5	6	7
BCLR #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B:M EA	R:W:M NEXT	W:B EA	
BCLR Rn,Rd	R:W NEXT						
BCLR Rn,@ERd	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA			
BCLR Rn,@aa:8	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA			
BCLR Rn,@aa:16	R:W 2nd	R:W 3rd	R:B:M EA	R:W:M NEXT	W:B EA		
BCLR Rn,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B:M EA	R:W:M NEXT	W:B EA	
BIAND #xx:3,Rd	R:W NEXT						
BIAND #xx:3,@ERd	R:W 2nd	R:B EA	R:W:M NEXT				
BIAND #xx:3,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT				
BIAND #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT			
BIAND #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT		
BILD #xx:3,Rd	R:W NEXT						
BILD #xx:3,@ERd	R:W 2nd	R:B EA	R:W:M NEXT				
BILD #xx:3,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT				
BILD #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT			
BILD #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT		
BIOR #xx:3,Rd	R:W NEXT						
BIOR #xx:3,@ERd	R:W 2nd	R:B EA	R:W:M NEXT				
BIOR #xx:3,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT				
BIOR #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT			
BIOR #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT		
BIST #xx:3,Rd	R:W NEXT						
BIST #xx:3,@ERd	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA			
BIST #xx:3,@aa:8	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA			
BIST #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B:M EA	R:W:M NEXT	W:B EA		
BIST #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B:M EA	R:W:M NEXT	W:B EA	
BIXOR #xx:3,Rd	R:W NEXT						
BIXOR #xx:3,@ERd	R:W 2nd	R:B EA	R:W:M NEXT				
BIXOR #xx:3,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT				
BIXOR #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT			
BIXOR #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT		
BLD #xx:3,Rd	R:W NEXT						
BLD #xx:3,@ERd	R:W 2nd	R:B EA	R:W:M NEXT				
BLD #xx:3,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT				
BLD #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT			

RENESAS

Instruction	1	2	3	4	5	6	7
BNOT #xx:3,@ERd	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA			
BNOT #xx:3,@aa:8	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA			
BNOT #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B:M EA	R:W:M NEXT	W:B EA		
BNOT #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B:M EA	R:W:M NEXT	W:B EA	
BNOT Rn,Rd	R:W NEXT						
BNOT Rn,@ERd	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA			
BNOT Rn,@aa:8	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA			
BNOT Rn,@aa:16	R:W 2nd	R:W 3rd	R:B:M EA	R:W:M NEXT	W:B EA		
BNOT Rn,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B:M EA	R:W:M NEXT	W:B EA	
BOR #xx:3,Rd	R:W NEXT						
BOR #xx:3,@ERd	R:W 2nd	R:B EA	R:W:M NEXT				
BOR #xx:3,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT				
BOR #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT			
BOR #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT		
BSET #xx:3,Rd	R:W NEXT						
BSET #xx:3,@ERd	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA			
BSET #xx:3,@aa:8	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA			
BSET #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B:M EA	R:W:M NEXT	W:B EA		
BSET #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B:M EA	R:W:M NEXT	W:B EA	
BSET Rn,Rd	R:W NEXT						
BSET Rn,@ERd	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA			
BSET Rn,@aa:8	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA			
BSET Rn,@aa:16	R:W 2nd	R:W 3rd	R:B:M EA	R:W:M NEXT	W:B EA		
BSET Rn,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B:M EA	R:W:M NEXT	W:B EA	
BSR d:8	R:W NEXT	R:W EA	W:W:M stack (H)	W:W stack (L)			
BSR d:16	R:W 2nd	Internal operation, 1 state	R:W EA	W:W:M stack (H)	W:W stack (L)		
BST #xx:3,Rd	R:W NEXT						
BST #xx:3,@ERd	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA			
BST #xx:3,@aa:8	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA			
BST #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B:M EA	R:W:M NEXT	W:B EA		
BST #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B:M EA	R:W:M NEXT	W:B EA	
BTST #xx:3,Rd	R:W NEXT						
BTST #xx:3,@ERd	R:W 2nd	R:B EA	R:W:M NEXT				

Instruction	1	2	3	4	5	6	7
BTST #xx:3,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT				
BTST #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT			
BTST #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT		
BTST Rn,Rd	R:W NEXT						
BTST Rn,@ERd	R:W 2nd	R:B EA	R:W:M NEXT				
BTST Rn,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT				
BTST Rn,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT			
BTST Rn,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT		
BXOR #xx:3,Rd	R:W NEXT						
BXOR #xx:3,@ERd	R:W 2nd	R:B EA	R:W:M NEXT				
BXOR #xx:3,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT				
BXOR #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT			
BXOR #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT		
CLRMAC	R:W NEXT	Internal operation, 1 state					
CMP.B #xx:8,Rd	R:W NEXT						
CMP.B Rs,Rd	R:W NEXT						
CMP.W #xx:16,Rd	R:W 2nd	R:W NEXT					
CMP.W Rs,Rd	R:W NEXT						
CMP.L #xx:32,ERd	R:W 2nd	R:W 3rd	R:W NEXT				
CMP.L ERs,ERd	R:W NEXT						
DAA Rd	R:W NEXT						
DAS Rd	R:W NEXT						
DEC.B Rd	R:W NEXT						
DEC.W #1/2,Rd	R:W NEXT						
DEC.L #1/2,ERd	R:W NEXT						
DIVXS.B Rs,Rd	R:W 2nd	R:W NEXT	Internal operation, 11 states				
DIVXS.W Rs,ERd	R:W 2nd	R:W NEXT	Internal operation, 19 states				
DIVXU.B Rs,Rd	R:W NEXT	Internal operation, 11 states					
DIVXU.W Rs,ERd	R:W NEXT	Internal operation, 19 states					
EEPMOV.B	R:W 2nd	R:B EAs*1	R:B EAd*1	R:B EAs*2	W:B EAd*2	R:W NEXT	
EEPMOV.W	R:W 2nd	R:B EAs*1	R:B EAd*1	R:B EAs*2	W:B EAd*2	R:W NEXT	
EXTS.W Rd	R:W NEXT			← Repeated n times*2 →			
EXTS.L ERd	R:W NEXT						
EXTU.W Rd	R:W NEXT						

Instruction	1	2	3	4	5	6	7
INC.W #1/2,Rd	R:W NEXT						
INC.L #1/2,ERd	R:W NEXT						
JMP @ERn	R:W NEXT	R:W EA					
JMP @aa:24	R:W 2nd	Internal operation, 1 state	R:W EA				
JMP @@aa:8	R:W NEXT	R:W:M aa:8	R:W aa:8	Internal operation, 1 state	R:W EA		
JSR @ERn	R:W NEXT	R:W EA	W:W:M stack (H)	W:W stack (L)			
JSR @aa:24	R:W 2nd	Internal operation, 1 state	R:W EA	W:W:M stack (H)	W:W stack (L)		
JSR @@aa:8	R:W NEXT	R:W:M aa:8	R:W aa:8	W:W:M stack (H)	W:W stack (L)	R:W EA	
LDC #xx:8,CCR	R:W NEXT						
LDC #xx:8,EXR	R:W 2nd	R:W NEXT					
LDC Rs,CCR	R:W NEXT						
LDC Rs,EXR	R:W NEXT						
LDC @ERs,CCR	R:W 2nd	R:W NEXT	R:W EA				
LDC @ERs,EXR	R:W 2nd	R:W NEXT	R:W EA				
LDC @(d:16,ERs),CCR	R:W 2nd	R:W 3rd	R:W NEXT	R:W EA			
LDC @(d:16,ERs),EXR	R:W 2nd	R:W 3rd	R:W NEXT	R:W EA			
LDC @(d:32,ERs),CCR	R:W 2nd	R:W 3rd	R:W 4th	R:W 5th	R:W NEXT	R:W EA	
LDC @(d:32,ERs),EXR	R:W 2nd	R:W 3rd	R:W 4th	R:W 5th	R:W NEXT	R:W EA	
LDC @ERs+,CCR	R:W 2nd	R:W NEXT	Internal operation, 1 state	R:W EA			
LDC @ERs+,EXR	R:W 2nd	R:W NEXT	Internal operation, 1 state	R:W EA			
LDC @aa:16,CCR	R:W 2nd	R:W 3rd	R:W NEXT	R:W EA			
LDC @aa:16,EXR	R:W 2nd	R:W 3rd	R:W NEXT	R:W EA			
LDC @aa:32,CCR	R:W 2nd	R:W 3rd	R:W 4th	R:W NEXT	R:W EA		
LDC @aa:32,EXR	R:W 2nd	R:W 3rd	R:W 4th	R:W NEXT	R:W EA		
LDM.L @SP+, (ERn–ERn+1)	R:W 2nd	R:W:M NEXT	Internal operation, 1 state	R:W:M stack (H)*3	R:W stack (L)*3		

Instruction	1	2	3	4	5	6	7
LDM.L @SP+,(ERn–ERn+2)	R:W 2nd	R:W NEXT	Internal operation, 1 state	R:W:M stack (H)*3	R:W stack (L)*3		
LDM.L @SP+,(ERn–ERn+3)	R:W 2nd	R:W NEXT	Internal operation, 1 state	R:W:M stack (H)*3	R:W stack (L)*3		
LDMAC ERs,MACH	R:W NEXT	Internal operation, 1 state		← Repeated n times*3 →			
LDMAC ERs,MACL	R:W NEXT	Internal operation, 1 state					
MAC @ERn+,@ERm+	R:W 2nd	R:W NEXT	R:W EAn	R:W EAm			
MOV.B #xx:8,Rd	R:W NEXT						
MOV.B Rs,Rd	R:W NEXT						
MOV.B @ERs,Rd	R:W NEXT	R:B EA					
MOV.B @(d:16,ERs),Rd	R:W 2nd	R:W NEXT	R:B EA				
MOV.B @(d:32,ERs),Rd	R:W 2nd	R:W 3rd	R:W 4th	R:W NEXT	R:B EA		
MOV.B @ERs+,Rd	R:W NEXT	Internal operation, 1 state	R:B EA				
MOV.B @aa:8,Rd	R:W NEXT	R:B EA					
MOV.B @aa:16,Rd	R:W 2nd	R:W NEXT	R:B EA				
MOV.B @aa:32,Rd	R:W 2nd	R:W 3rd	R:W NEXT	R:B EA			
MOV.B Rs,@ERd	R:W NEXT	W:B EA					
MOV.B Rs,@(d:16,ERd)	R:W 2nd	R:W NEXT	W:B EA				
MOV.B Rs,@(d:32,ERd)	R:W 2nd	R:W 3rd	R:W 4th	R:W NEXT	W:B EA		
MOV.B Rs,@–ERd	R:W NEXT	Internal operation, 1 state	W:B EA				
MOV.B Rs,@aa:8	R:W NEXT	W:B EA					
MOV.B Rs,@aa:16	R:W 2nd	R:W NEXT	W:B EA				
MOV.B Rs,@aa:32	R:W 2nd	R:W 3rd	R:W NEXT	W:B EA			
MOV.W #xx:16,Rd	R:W 2nd	R:W NEXT					
MOV.W Rs,Rd	R:W NEXT						
MOV.W @ERs,Rd	R:W NEXT	R:W EA					
MOV.W @(d:16,ERs),Rd	R:W 2nd	R:W NEXT	R:W EA				
MOV.W @(d:32,ERs),Rd	R:W 2nd	R:W 3rd	R:W 4th	R:W NEXT	R:W EA		
MOV.W @ERs+, Rd	R:W NEXT	Internal operation, 1 state	R:W EA				
MOV.W @aa:16,Rd	R:W 2nd	R:W NEXT	R:W EA				

Instruction	1	2	3	4	5	6	7
MOV.W Rs,@(d:16,ERd)	R:W 2nd	R:W NEXT	W:W EA				
MOV.W Rs,@(d:32,ERd)	R:W 2nd	R:W 3rd	R:E 4th	R:W NEXT	W:W EA		
MOV.W Rs,@–ERd	R:W NEXT	Internal operation, 1 state	W:W EA				
MOV.W Rs,@aa:16	R:W 2nd	R:W NEXT	W:W EA				
MOV.W Rs,@aa:32	R:W 2nd	R:W 3rd	R:W NEXT	W:W EA			
MOV.L #xx:32,ERd	R:W 2nd	R:W 3rd	R:W NEXT				
MOV.L ERs,ERd	R:W NEXT						
MOV.L @ERs,ERd	R:W 2nd	R:W:M NEXT	R:W:M EA	R:W EA+2			
MOV.L @(d:16,ERs),ERd	R:W 2nd	R:W:M 3rd	R:W NEXT	R:W:M EA	R:W EA+2		
MOV.L @(d:32,ERs),ERd	R:W 2nd	R:W:M 3rd	R:W:M 4th	R:W 5th	R:W NEXT	R:W:M EA	R:W EA+2
MOV.L @ERs+,ERd	R:W 2nd	R:W:M NEXT	Internal operation, 1 state	R:W:M EA	R:W EA+2		
MOV.L @aa:16,ERd	R:W 2nd	R:W:M 3rd	R:W NEXT	R:W:M EA	R:W EA+2		
MOV.L @aa:32,ERd	R:W 2nd	R:W:M 3rd	R:W 4th	R:W NEXT	R:W:M EA	R:W EA+2	
MOV.L ERs,@ERd	R:W 2nd	R:W:M NEXT	W:W:M EA	W:W EA+2			
MOV.L ERs,@(d:16,ERd)	R:W 2nd	R:W:M 3rd	R:W NEXT	W:W:M EA	W:W EA+2		
MOV.L ERs,@(d:32,ERd)	R:W 2nd	R:W:M 3rd	R:W:M 4th	R:W 5th	R:W NEXT	W:W:M EA	W:W EA+2
MOV.L ERs,@–ERd	R:W 2nd	R:W:M NEXT	Internal operation, 1 state	W:W:M EA	W:W EA+2		
MOV.L ERs,@aa:16	R:W 2nd	R:W:M 3rd	R:W NEXT	W:W:M EA	W:W EA+2		
MOV.L ERs,@aa:32	R:W 2nd	R:W:M 3rd	R:W 4th	R:W NEXT	W:W:M EA	W:W EA+2	
MOVFPE @aa:16,Rd	Cannot be used in the H8S/2626 Group or H8S/2623 Group.						
MOVTPE Rs,@aa:16							
MULXS.B Rs,Rd	R:W 2nd	R:W NEXT	Internal operation, 2 states				
MULXS.W Rs,ERd	R:W 2nd	R:W NEXT	Internal operation, 3 states				
MULXU.B Rs,Rd	R:W NEXT	Internal operation, 2 states					
MULXU.W Rs,ERd	R:W NEXT	Internal operation, 3 states					
NEG.B Rd	R:W NEXT						
NEG.W Rd	R:W NEXT						
NEG.L ERd	R:W NEXT						
NOP	R:W NEXT						
NOT.B Rd	R:W NEXT						
NOT.W Rd	R:W NEXT						
NOT.L ERd	R:W NEXT						

Instruction	1	2	3	4	5	6	7
OR.W #xx:16,Rd	R:W 2nd	R:W NEXT					
OR.W Rs,Rd	R:W NEXT						
OR.L #xx:32,ERd	R:W 2nd	R:W 3rd	R:W NEXT				
OR.L ERs,ERd	R:W 2nd	R:W NEXT					
ORC #xx:8,CCR	R:W NEXT						
ORC #xx:8,EXR	R:W 2nd	R:W NEXT					
POP.W Rn	R:W NEXT	Internal operation, 1 state	R:W EA				
POP.L ERn	R:W 2nd	R:W:M NEXT	Internal operation, 1 state	R:W:M EA	R:W EA+2		
PUSH.W Rn	R:W NEXT	Internal operation, 1 state	W:W EA				
PUSH.L ERn	R:W 2nd	R:W:M NEXT	Internal operation, 1 state	W:W:M EA	W:W EA+2		
ROTL.B Rd	R:W NEXT						
ROTL.B #2,Rd	R:W NEXT						
ROTL.W Rd	R:W NEXT						
ROTL.W #2,Rd	R:W NEXT						
ROTL.L ERd	R:W NEXT						
ROTL.L #2,ERd	R:W NEXT						
ROTR.B Rd	R:W NEXT						
ROTR.B #2,Rd	R:W NEXT						
ROTR.W Rd	R:W NEXT						
ROTR.W #2,Rd	R:W NEXT						
ROTR.L ERd	R:W NEXT						
ROTR.L #2,ERd	R:W NEXT						
ROTXL.B Rd	R:W NEXT						
ROTXL.B #2,Rd	R:W NEXT						
ROTXL.W Rd	R:W NEXT						
ROTXL.W #2,Rd	R:W NEXT						
ROTXL.L ERd	R:W NEXT						
ROTXL.L #2,ERd	R:W NEXT						
ROTXR.B Rd	R:W NEXT						
ROTXR.B #2,Rd	R:W NEXT						
ROTXR.W Rd	R:W NEXT						

Instruction	1	2	3	4	5	6	7
ROTXR.L #2,ERd	R:W NEXT						
RTE	R:W NEXT	R:W stack (EXR)	R:W stack (H)	R:W stack (L)	Internal operation, 1 state	R:W*4	
RTS	R:W NEXT	R:W:M stack (H)	R:W stack (L)	Internal operation, 1 state	R:W*4		
SHAL.B Rd	R:W NEXT						
SHAL.B #2,Rd	R:W NEXT						
SHAL.W Rd	R:W NEXT						
SHAL.W #2,Rd	R:W NEXT						
SHAL.L ERd	R:W NEXT						
SHAL.L #2,ERd	R:W NEXT						
SHAR.B Rd	R:W NEXT						
SHAR.B #2,Rd	R:W NEXT						
SHAR.W Rd	R:W NEXT						
SHAR.W #2,Rd	R:W NEXT						
SHAR.L ERd	R:W NEXT						
SHAR.L #2,ERd	R:W NEXT						
SHLL.B Rd	R:W NEXT						
SHLL.B #2,Rd	R:W NEXT						
SHLL.W Rd	R:W NEXT						
SHLL.W #2,Rd	R:W NEXT						
SHLL.L ERd	R:W NEXT						
SHLL.L #2,ERd	R:W NEXT						
SHLR.B Rd	R:W NEXT						
SHLR.B #2,Rd	R:W NEXT						
SHLR.W Rd	R:W NEXT						
SHLR.W #2,Rd	R:W NEXT						
SHLR.L ERd	R:W NEXT						
SHLR.L #2,ERd	R:W NEXT						
SLEEP	R:W NEXT	Internal operation:M					
STC CCR,Rd	R:W NEXT						
STC EXR,Rd	R:W NEXT						
STC CCR,@ERd	R:W 2nd	R:W NEXT	W:W EA				
STC EXR,@ERd	R:W 2nd	R:W NEXT	W:W EA				
STC CCR,@(d:16,ERd)	R:W 2nd	R:W 3rd	R:W NEXT	W:W EA			

Instruction	1	2	3	4	5	6	7
STC EXR,@(d:16,ERd)	R:W 2nd	R:W 3rd	R:W NEXT	W:W EA			
STC CCR,@(d:32,ERd)	R:W 2nd	R:W 3rd	R:W 4th	R:W 5th	R:W NEXT	W:W EA	
STC EXR,@(d:32,ERd)	R:W 2nd	R:W 3rd	R:W 4th	R:W 5th	R:W NEXT	W:W EA	
STC CCR,@–ERd	R:W 2nd	R:W NEXT	Internal operation, 1 state	W:W EA			
STC EXR,@–ERd	R:W 2nd	R:W NEXT	Internal operation, 1 state	W:W EA			
STC CCR,@aa:16	R:W 2nd	R:W 3rd	R:W NEXT	W:W EA			
STC EXR,@aa:16	R:W 2nd	R:W 3rd	R:W NEXT	W:W EA			
STC CCR,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:W NEXT	W:W EA		
STC EXR,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:W NEXT	W:W EA		
STM.L(ERn–ERn+1),@–SP	R:W 2nd	R:W:M NEXT	Internal operation, 1 state	W:W:M stack (H)*3	W:W stack (L)*3		
STM.L(ERn–ERn+2),@–SP	R:W 2nd	R:W:M NEXT	Internal operation, 1 state	W:W:M stack (H)*3	W:W stack (L)*3		
STM.L(ERn–ERn+3),@–SP	R:W 2nd	R:W:M NEXT	Internal operation, 1 state	W:W:M stack (H)*3	W:W stack (L)*3		
STMAC MACH,ERd	R:W NEXT						
STMAC MACL,ERd	R:W NEXT						
SUB.B Rs,Rd	R:W NEXT						
SUB.W #xx:16,Rd	R:W 2nd	R:W NEXT					
SUB.W Rs,Rd	R:W NEXT						
SUB.L #xx:32,ERd	R:W 2nd	R:W 3rd	R:W NEXT				
SUB.L ERs,ERd	R:W NEXT						
SUBS #1/2/4,ERd	R:W NEXT						
SUBX #xx:8,Rd	R:W NEXT						
SUBX Rs,Rd	R:W NEXT						
TAS @ERd*8	R:W 2nd	R:W NEXT	R:B:M EA	W:B EA			
TRAPA #x:2	R:W NEXT	Internal operation, 1 state	W:W stack (L)	W:W stack (H)	W:W stack (EXR)	R:W:M VEC	R:W VEC+
XOR.B #xx8,Rd	R:W NEXT						
XOR.B Rs,Rd	R:W NEXT						
XOR.W #xx:16,Rd	R:W 2nd	R:W NEXT					
XOR.W Rs,Rd	R:W NEXT						
XOR.L #xx:32,ERd	R:W 2nd	R:W 3rd	R:W NEXT				

RENESAS

Instruction	1	2	3	4	5	6	7
XOR.L ERs,ERd	R:W 2nd	R:W NEXT					
XORC #xx:8,CCR	R:W NEXT						
XORC #xx:8,EXR	R:W 2nd	R:W NEXT					
Reset exception	R:W VEC	R:W VEC+2	Internal operation, 1 state	R:W*5			
Interrupt exception	R:W*6	Internal operation, 1 state	W:W stack (L)	W:W stack (H)	W:W stack (EXR)	R:W:M VEC	R:W VEC+

Notes: 1. EAs is the contents of ER5. EAd is the contents of ER6.
2. EAs is the contents of ER5. EAd is the contents of ER6. Both registers are incremented by 1 after execution o value of R4L or R4. If n = 0, these bus cycles are not executed.
3. Repeated two times to save or restore two registers, three times for three registers, or four times for four regis
4. Start address after return.
5. Start address of the program.
6. Prefetch address, equal to two plus the PC value pushed onto the stack. In recovery from sleep mode or softw operation is replaced by an internal operation.
7. Start address of the interrupt-handling routine.
8. Only register ER0, ER1, ER4, or ER5 should be used when using the TAS instruction.

	7 for byte operands
Si	The i-th bit of the source operand
Di	The i-th bit of the destination operand
Ri	The i-th bit of the result
Dn	The specified bit in the destination operand
—	Not affected
↕	Modified according to the result of the instruction (see definition)
0	Always cleared to 0
1	Always set to 1
*	Undetermined (no guaranteed value)
Z'	Z flag before instruction execution
C'	C flag before instruction execution

ADDS	—	—	—	—	—	
ADDX	↕	↕	↕	↕	↕	H = Sm–4 · Dm–4 + Dm–4 · $\overline{Rm–4}$ + Sm–4 · $\overline{Rm}$ N = Rm Z = Z' · $\overline{Rm}$ · ····· · $\overline{R0}$ V = Sm · Dm · $\overline{Rm}$ + $\overline{Sm}$ · $\overline{Dm}$ · Rm C = Sm · Dm + Dm · $\overline{Rm}$ + Sm · $\overline{Rm}$
AND	—	↕	↕	0	—	N = Rm Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$
ANDC	↕	↕	↕	↕	↕	Stores the corresponding bits of the result. No flags change when the operand is EXR.
BAND	—	—	—	—	↕	C = C' · Dn
Bcc	—	—	—	—	—	
BCLR	—	—	—	—	—	
BIAND	—	—	—	—	↕	C = C' · $\overline{Dn}$
BILD	—	—	—	—	↕	C = $\overline{Dn}$
BIOR	—	—	—	—	↕	C = C' + $\overline{Dn}$
BIST	—	—	—	—	—	
BIXOR	—	—	—	—	↕	C = C' · Dn + $\overline{C'}$ · $\overline{Dn}$
BLD	—	—	—	—	↕	C = Dn
BNOT	—	—	—	—	—	
BOR	—	—	—	—	↕	C = C' + Dn
BSET	—	—	—	—	—	
BSR	—	—	—	—	—	
BST	—	—	—	—	—	
BTST	—	—	↕	—	—	Z = $\overline{Dn}$
BXOR	—	—	—	—	↕	C = C' · $\overline{Dn}$ + $\overline{C'}$ · Dn

DAA	*	↕	↕	*	↕	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ C: decimal arithmetic carry
DAS	*	↕	↕	*	↕	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ C: decimal arithmetic borrow
DEC	—	↕	↕	↕	—	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ V = Dm · $\overline{Rm}$
DIVXS	—	↕	↕	—	—	N = Sm · $\overline{Dm}$ + $\overline{Sm}$ · Dm Z = $\overline{Sm} \cdot \overline{Sm-1} \cdot \cdots\cdots \cdot \overline{S0}$
DIVXU	—	↕	↕	—	—	N = Sm Z = $\overline{Sm} \cdot \overline{Sm-1} \cdot \cdots\cdots \cdot \overline{S0}$
EEPMOV	—	—	—	—	—	
EXTS	—	↕	↕	0	—	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$
EXTU	—	0	↕	0	—	Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$
INC	—	↕	↕	↕	—	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ V = $\overline{Dm}$ · Rm
JMP	—	—	—	—	—	
JSR	—	—	—	—	—	
LDC	↕	↕	↕	↕	↕	Stores the corresponding bits of the result. No flags change when the operand is EXR.
LDM	—	—	—	—	—	
LDMAC	—	—	—	—	—	
MAC	—	—	—	—	—	

						Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$
MULXU	—	—	—	—	—	
NEG	↕	↕	↕	↕	↕	H = Dm–4 + Rm–4 N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ V = Dm · Rm C = Dm + Rm
NOP	—	—	—	—	—	
NOT	—	↕	↕	0	—	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$
OR	—	↕	↕	0	—	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$
ORC	↕	↕	↕	↕	↕	Stores the corresponding bits of the result. No flags change when the operand is EXR.
POP	—	↕	↕	0	—	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$
PUSH	—	↕	↕	0	—	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$
ROTL	—	↕	↕	0	↕	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ C = Dm (1-bit shift) or C = Dm–1 (2-bit shift)
ROTR	—	↕	↕	0	↕	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ C = D0 (1-bit shift) or C = D1 (2-bit shift)

RTE	↕	↕	↕	↕	↕	Stores the corresponding bits of the result.
RTS	—	—	—	—	—	
SHAL	—	↕	↕	↕	↕	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ V = $\overline{Dm \cdot Dm-1 + \overline{Dm} \cdot \overline{Dm-1}}$ (1-bit shift) V = $\overline{Dm \cdot Dm-1 \cdot Dm-2 \cdot \overline{Dm} \cdot \overline{Dm-1} \cdot \overline{Dm-2}}$ (2 C = Dm (1-bit shift) or C = Dm–1 (2-bit shift)
SHAR	—	↕	↕	0	↕	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ C = D0 (1-bit shift) or C = D1 (2-bit shift)
SHLL	—	↕	↕	0	↕	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ C = Dm (1-bit shift) or C = Dm–1 (2-bit shift)
SHLR	—	0	↕	0	↕	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ C = D0 (1-bit shift) or C = D1 (2-bit shift)
SLEEP	—	—	—	—	—	
STC	—	—	—	—	—	
STM	—	—	—	—	—	
STMAC	—	↕	↕	↕	—	N = 1 if MAC instruction resulted in negative val register Z = 1 if MAC instruction resulted in zero value i register V = 1 if MAC instruction resulted in overflow

SUBX	↕	↕	↕	↕	↕	H = $\overline{Sm–4}$ · Dm–4 + Dm–4 · $\overline{Rm–4}$ + $\overline{Sm–4}$ · Rm N = Rm Z = Z' · $\overline{Rm}$ · ····· · $\overline{R0}$ V = $\overline{Sm}$ · Dm · $\overline{Rm}$ + Sm · $\overline{Dm}$ · Rm C = Sm · $\overline{Dm}$ + $\overline{Dm}$ · Rm + Sm · Rm
TAS	—	↕	↕	0	—	N = Dm Z = $\overline{Dm}$ · $\overline{Dm–1}$ · ····· · $\overline{D0}$
TRAPA	—	—	—	—	—	
XOR	—	↕	↕	0	—	N = Rm Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$
XORC	↕	↕	↕	↕	↕	Stores the corresponding bits of the result. No flags change when the operand is EXR.

	SAR									
	DAR									
	CRA									
	CRB									
H'F800	MCR	MCR7	—	MCR5	—	—	MCR2	MCR1	MCR0	HC
H'F801	GSR	—	—	—	—	GSR3	GSR2	GSR1	GSR0	
H'F802	BCR	BCR7	BCR6	BCR5	BCR4	BCR3	BCR2	BCR1	BCR0	
H'F803		BCR15	BCR14	BCR13	BCR12	BCR11	BCR10	BCR9	BCR8	
H'F804	MBCR	MBCR7	MBCR6	MBCR5	MBCR4	MBCR3	MBCR2	MBCR1	—	
H'F805		MBCR15	MBCR14	MBCR13	MBCR12	MBCR11	MBCR10	MBCR9	MBCR8	
H'F806	TXPR	TXPR7	TXPR6	TXPR5	TXPR4	TXPR3	TXPR2	TXPR1	—	
H'F807		TXPR15	TXPR14	TXPR13	TXPR12	TXPR11	TXPR10	TXPR9	TXPR8	
H'F808	TXCR	TXCR7	TXCR6	TXCR5	TXCR4	TXCR3	TXCR2	TXCR1	—	
H'F809		TXCR15	TXCR14	TXCR13	TXCR12	TXCR11	TXCR10	TXCR9	TXCR8	
H'F80A	TXACK	TXACK7	TXACK6	TXACK5	TXACK4	TXACK3	TXACK2	TXACK1	—	
H'F80B		TXACK15	TXACK14	TXACK13	TXACK12	TXACK11	TXACK10	TXACK9	TXACK8	
H'F80C	ABACK	ABACK7	ABACK6	ABACK5	ABACK4	ABACK3	ABACK2	ABACK1	—	
H'F80D		ABACK15	ABACK14	ABACK13	ABACK12	ABACK11	ABACK10	ABACK9	ABACK8	
H'F80E	RXPR	RXPR7	RXPR6	RXPR5	RXPR4	RXPR3	RXPR2	RXPR1	RXPR0	
H'F80F		RXPR15	RXPR14	RXPR13	RXPR12	RXPR11	RXPR10	RXPR9	RXPR8	
H'F810	RFPR	RFPR7	RFPR6	RFPR5	RFPR4	RFPR3	RFPR2	RFPR1	RFPR0	
H'F811		RFPR15	RFPR14	RFPR13	RFPR12	RFPR11	RFPR10	RFPR9	RFPR8	
H'F812	IRR	IRR7	IRR6	IRR5	IRR4	IRR3	IRR2	IRR1	IRR0	
H'F813		—	—	—	IRR12	—	—	IRR9	IRR8	

H'F81B		UMSR15	UMSR14	UMSR13	UMSR12	UMSR11	UMSR10	UMSR9	UMSR8
H'F81C	LAFML	LAFML7	LAFML6	LAFML5	LAFML4	LAFML3	LAFML2	LAFML1	LAFML0
H'F81D		LAFML15	LAFML14	LAFML13	LAFML12	LAFML11	LAFML10	LAFML9	LAFML8
H'F81E	LAFMH	LAFMH7	LAFMH6	LAFMH5	—	—	—	LAFMH1	LAFMH0
H'F81F		LAFMH15	LAFMH14	LAFMH13	LAFMH12	LAFMH11	LAFMH10	LAFMH9	LAFMH8
H'F820	MC0[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0
H'F821	MC0[2]	—	—	—	—	—	—	—	—
H'F822	MC0[3]	—	—	—	—	—	—	—	—
H'F823	MC0[4]	—	—	—	—	—	—	—	—
H'F824	MC0[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
H'F825	MC0[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
H'F826	MC0[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
H'F827	MC0[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
H'F828	MC1[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0
H'F829	MC1[2]	—	—	—	—	—	—	—	—
H'F82A	MC1[3]	—	—	—	—	—	—	—	—
H'F82B	MC1[4]	—	—	—	—	—	—	—	—
H'F82C	MC1[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
H'F82D	MC1[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
H'F82E	MC1[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
H'F82F	MC1[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
H'F830	MC2[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0
H'F831	MC2[2]	—	—	—	—	—	—	—	—
H'F832	MC2[3]	—	—	—	—	—	—	—	—
H'F833	MC2[4]	—	—	—	—	—	—	—	—
H'F834	MC2[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
H'F835	MC2[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
H'F836	MC2[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
H'F837	MC2[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
H'F838	MC3[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0
H'F839	MC3[2]	—	—	—	—	—	—	—	—

H'F841	MC4[2]	—	—	—	—	—	—	—	—
H'F842	MC4[3]	—	—	—	—	—	—	—	—
H'F843	MC4[4]	—	—	—	—	—	—	—	—
H'F844	MC4[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
H'F845	MC4[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
H'F846	MC4[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
H'F847	MC4[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
H'F848	MC5[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0
H'F849	MC5[2]	—	—	—	—	—	—	—	—
H'F84A	MC5[3]	—	—	—	—	—	—	—	—
H'F84B	MC5[4]	—	—	—	—	—	—	—	—
H'F84C	MC5[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
H'F84D	MC5[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
H'F84E	MC5[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
H'F84F	MC5[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
H'F850	MC6[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0
H'F851	MC6[2]	—	—	—	—	—	—	—	—
H'F852	MC6[3]	—	—	—	—	—	—	—	—
H'F853	MC6[4]	—	—	—	—	—	—	—	—
H'F854	MC6[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
H'F855	MC6[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
H'F856	MC6[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
H'F857	MC6[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
H'F858	MC7[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0
H'F859	MC7[2]	—	—	—	—	—	—	—	—
H'F85A	MC7[3]	—	—	—	—	—	—	—	—
H'F85B	MC7[4]	—	—	—	—	—	—	—	—
H'F85C	MC7[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
H'F85D	MC7[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
H'F85E	MC7[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0

H'F866	MC8[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
H'F867	MC8[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
H'F868	MC9[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0
H'F869	MC9[2]	—	—	—	—	—	—	—	—
H'F86A	MC9[3]	—	—	—	—	—	—	—	—
H'F86B	MC9[4]	—	—	—	—	—	—	—	—
H'F86C	MC9[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
H'F86D	MC9[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
H'F86E	MC9[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
H'F86F	MC9[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
H'F870	MC10[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0
H'F871	MC10[2]	—	—	—	—	—	—	—	—
H'F872	MC10[3]	—	—	—	—	—	—	—	—
H'F873	MC10[4]	—	—	—	—	—	—	—	—
H'F874	MC10[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
H'F875	MC10[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
H'F876	MC10[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
H'F877	MC10[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
H'F878	MC11[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0
H'F879	MC11[2]	—	—	—	—	—	—	—	—
H'F87A	MC11[3]	—	—	—	—	—	—	—	—
H'F87B	MC11[4]	—	—	—	—	—	—	—	—
H'F87C	MC11[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
H'F87D	MC11[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
H'F87E	MC11[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
H'F87F	MC11[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
H'F880	MC12[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0
H'F881	MC12[2]	—	—	—	—	—	—	—	—
H'F882	MC12[3]	—	—	—	—	—	—	—	—
H'F883	MC12[4]	—	—	—	—	—	—	—	—

H'F88B	MC13[4]	—	—	—	—	—	—	—	—
H'F88C	MC13[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
H'F88D	MC13[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
H'F88E	MC13[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
H'F88F	MC13[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
H'F890	MC14[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0
H'F891	MC14[2]	—	—	—	—	—	—	—	—
H'F892	MC14[3]	—	—	—	—	—	—	—	—
H'F893	MC14[4]	—	—	—	—	—	—	—	—
H'F894	MC14[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
H'F895	MC14[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
H'F896	MC14[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
H'F897	MC14[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
H'F898	MC15[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0
H'F899	MC15[2]	—	—	—	—	—	—	—	—
H'F89A	MC15[3]	—	—	—	—	—	—	—	—
H'F89B	MC15[4]	—	—	—	—	—	—	—	—
H'F89C	MC15[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
H'F89D	MC15[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
H'F89E	MC15[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
H'F89F	MC15[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
H'F8B0	MD0[1]								
H'F8B1	MD0[2]								
H'F8B2	MD0[3]								
H'F8B3	MD0[4]								
H'F8B4	MD0[5]								
H'F8B5	MD0[6]								
H'F8B6	MD0[7]								
H'F8B7	MD0[8]								
H'F8B8	MD1[1]								

H'F8C0	MD2[1]
H'F8C1	MD2[2]
H'F8C2	MD2[3]
H'F8C3	MD2[4]
H'F8C4	MD2[5]
H'F8C5	MD2[6]
H'F8C6	MD2[7]
H'F8C7	MD2[8]
H'F8C8	MD3[1]
H'F8C9	MD3[2]
H'F8CA	MD3[3]
H'F8CB	MD3[4]
H'F8CC	MD3[5]
H'F8CD	MD3[6]
H'F8CE	MD3[7]
H'F8CF	MD3[8]
H'F8D0	MD4[1]
H'F8D1	MD4[2]
H'F8D2	MD4[3]
H'F8D3	MD4[4]
H'F8D4	MD4[5]
H'F8D5	MD4[6]
H'F8D6	MD4[7]
H'F8D7	MD4[8]
H'F8D8	MD5[1]
H'F8D9	MD5[2]
H'F8DA	MD5[3]
H'F8DB	MD5[4]
H'F8DC	MD5[5]
H'F8DD	MD5[6]

H'F8E5	MD6[6]
H'F8E6	MD6[7]
H'F8E7	MD6[8]
H'F8E8	MD7[1]
H'F8E9	MD7[2]
H'F8EA	MD7[3]
H'F8EB	MD7[4]
H'F8EC	MD7[5]
H'F8ED	MD7[6]
H'F8EE	MD7[7]
H'F8EF	MD7[8]
H'F8F0	MD8[1]
H'F8F1	MD8[2]
H'F8F2	MD8[3]
H'F8F3	MD8[4]
H'F8F4	MD8[5]
H'F8F5	MD8[6]
H'F8F6	MD8[7]
H'F8F7	MD8[8]
H'F8F8	MD9[1]
H'F8F9	MD9[2]
H'F8FA	MD9[3]
H'F8FB	MD9[4]
H'F8FC	MD9[5]
H'F8FD	MD9[6]
H'F8FE	MD9[7]
H'F8FF	MD9[8]
H'F900	MD10[1]
H'F901	MD10[2]
H'F902	MD10[3]

H'F90A	MD11[3]
H'F90B	MD11[4]
H'F90C	MD11[5]
H'F90D	MD11[6]
H'F90E	MD11[7]
H'F90F	MD11[8]
H'F910	MD12[1]
H'F911	MD12[2]
H'F912	MD12[3]
H'F913	MD12[4]
H'F914	MD12[5]
H'F915	MD12[6]
H'F916	MD12[7]
H'F917	MD12[8]
H'F918	MD13[1]
H'F919	MD13[2]
H'F91A	MD13[3]
H'F91B	MD13[4]
H'F91C	MD13[5]
H'F91D	MD13[6]
H'F91E	MD13[7]
H'F91F	MD13[8]
H'F920	MD14[1]
H'F921	MD14[2]
H'F922	MD14[3]
H'F923	MD14[4]
H'F924	MD14[5]
H'F925	MD14[6]
H'F926	MD14[7]
H'F927	MD14[8]
H'F928	MD15[1]

H'FDAC	DADR2*6	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	D/A
H'FDAD	DADR3*6	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	co
H'FDAE	DACR23*6	DAOE1	DAOE0	DAE	—	—	—	—	—	
H'FDB4	SCRX	—	—	—	—	FLSHE	—	—	—	RC
H'FDE4	SBYCR	SSBY	STS2	STS1	STS0	OPE	—	—	—	Po do
H'FDE5	SYSCR	MACS	—	INTM1	INTM0	NMIEG	—	—	RAME	MC RA int co
H'FDE6	SCKCR	PSTOP	—	—	—	STCS	SCK2	SCK1	SCK0	Clo pul ge po do
H'FDE7	MDCR	—	—	—	—	—	MDS2	MDS1	MDS0	MC
H'FDE8	MSTPCRA	MSTPA7	MSTPA6	MSTPA5	MSTPA4	MSTPA3	MSTPA2	MSTPA1	MSTPA0	Po do
H'FDE9	MSTPCRB	MSTPB7	MSTPB6	MSTPB5	MSTPB4	MSTPB3	MSTPB2	MSTPB1	MSTPB0	
H'FDEA	MSTPCRC	MSTPC7	MSTPC6	MSTPC5	MSTPC4	MSTPC3	MSTPC2	MSTPC1	MSTPC0	
H'FDEB	PFCR	—	—	BUZZE*4	—	AE3	AE2	AE1	AE0	MC co
H'FDEC	LPWRCR	DTON*4	LSON*4	NESEL*4	SUBSTP*4	RFCUT	—	STC1	STC0	Clo pul ge
H'FE00	BARA	—	—	—	—	—	—	—	—	PC co
		BAA23	BAA22	BAA21	BAA20	BAA19	BAA18	BAA17	BAA16	
		BAA15	BAA14	BAA13	BAA12	BAA11	BAA10	BAA9	BAA8	
		BAA7	BAA6	BAA5	BAA4	BAA3	BAA2	BAA1	BAA0	
H'FE04	BARB	—	—	—	—	—	—	—	—	
		BAA23	BAA22	BAA21	BAA20	BAA19	BAA18	BAA17	BAA16	
		BAA15	BAA14	BAA13	BAA12	BAA11	BAA10	BAA9	BAA8	
		BAA7	BAA6	BAA5	BAA4	BAA3	BAA2	BAA1	BAA0	
H'FE08	BCRA	CMFA	CDA	BAMRA2	BAMRA1	BAMRA0	CSELA1	CSELA0	BIEA	
H'FE09	BCRB	CMFB	CDB	BAMRB2	BAMRB1	BAMRA0	CSELB1	CSELB0	BIEB	

H'FE19	DTCERD	DTCED7	DTCED6	DTCED5	DTCED4	DTCED3	DTCED2	DTCED1	DTCED0	
H'FE1A	DTCERE	DTCEE7	DTCEE6	DTCEE5	DTCEE4	DTCEE3	DTCEE2	DTCEE1	DTCEE0	
H'FE1B	DTCERF	DTCEF7	DTCEF6	DTCEF5	DTCEF4	DTCEF3	DTCEF2	DTCEF1	DTCEF0	
H'FE1C	DTCERG	DTCEG7	DTCEG6	DTCEG5	DTCEG4	DTCEG3	DTCEG2	DTCEG1	DTCEG0	
H'FE1F	DTVECR	SWDTE	DTVEC6	DTVEC5	DTVEC4	DTVEC3	DTVEC2	DTVEC1	DTVEC0	
H'FE26	PCR	G3CMS1	G3CMS0	G2CMS1	G2CMS0	G1CMS1	G1CMS0	G0CMS1	G0CMS0	PPG
H'FE27	PMR	G3INV	G2INV	G1INV	G0INV	G3NOV	G2NOV	G1NOV	G0NOV	
H'FE28	NDERH	NDER15	NDER14	NDER13	NDER12	NDER11	NDER10	NDER9	NDER8	
H'FE29	NDERL	NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1	NDER0	
H'FE2A	PODRH	POD15	POD14	POD13	POD12	POD11	POD10	POD9	POD8	
H'FE2B	PODRL	POD7	POD6	POD5	POD4	POD3	POD2	POD1	POD0	
H'FE2C	NDRH*2	NDR15	NDR14	NDR13	NDR12	NDR11	NDR10	NDR9	NDR8	
H'FE2D	NDRL*2	NDR7	NDR6	NDR5	NDR4	NDR3	NDR2	NDR1	NDR0	
H'FE2E	NDRH*2	—	—	—	—	NDR11	NDR10	NDR9	NDR8	
H'FE2F	NDRL*2	—	—	—	—	NDR3	NDR2	NDR1	NDR0	
H'FE30	P1DDR	P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DDR	I/O p
H'FE39	PADDR	—	—	PA5DDR*5	PA4DDR*5	PA3DDR	PA2DDR	PA1DDR	PA0DDR	
H'FE3A	PBDDR	PB7DDR	PB6DDR	PB5DDR	PB4DDR	PB3DDR	PB2DDR	PB1DDR	PB0DDR	
H'FE3B	PCDDR	PC7DDR	PC6DDR	PC5DDR	PC4DDR	PC3DDR	PC2DDR	PC1DDR	PC0DDR	
H'FE3C	PDDDR	PD7DDR	PD6DDR	PD5DDR	PD4DDR	PD3DDR	PD2DDR	PD1DDR	PD0DDR	
H'FE3D	PEDDR	PE7DDR	PE6DDR	PE5DDR	PE4DDR	PE3DDR	PE2DDR	PE1DDR	PE0DDR	
H'FE3E	PFDDR	PF7DDR	PF6DDR	PF5DDR	PF4DDR	PF3DDR	PF2DDR	PF1DDR	PF0DDR	
H'FE40	PAPCR	—	—	PA5PCR*5	PA4PCR*5	PA3PCR	PA2PCR	PA1PCR	PA0PCR	
H'FE41	PBPCR	PB7PCR	PB6PCR	PB5PCR	PB4PCR	PB3PCR	PB2PCR	PB1PCR	PB0PCR	
H'FE42	PCPCR	PC7PCR	PC6PCR	PC5PCR	PC4PCR	PC3PCR	PC2PCR	PC1PCR	PC0PCR	
H'FE43	PDPCR	PD7PCR	PD6PCR	PD5PCR	PD4PCR	PD3PCR	PD2PCR	PD1PCR	PD0PCR	
H'FE44	PEPCR	PE7PCR	PE6PCR	PE5PCR	PE4PCR	PE3PCR	PE2PCR	PE1PCR	PE0PCR	
H'FE47	PAODR	—	—	PA5ODR*5	PA4ODR*5	PA3ODR	PA2ODR	PA1ODR	PA0ODR	
H'FE48	PBODR	PB7ODR	PB6ODR	PB5ODR	PB4ODR	PB3ODR	PB2ODR	PB1ODR	PB0ODR	
H'FE49	PCODR	PC7ODR	PC6ODR	PC5ODR	PC4ODR	PC3ODR	PC2ODR	PC1ODR	PC0ODR	

H'FE87										
H'FE88	TGR3A									
H'FE89										
H'FE8A	TGR3B									
H'FE8B										
H'FE8C	TGR3C									
H'FE8D										
H'FE8E	TGR3D									
H'FE8F										
H'FE90	TCR4	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TP
H'FE91	TMDR4	—	—	—	—	MD3	MD2	MD1	MD0	
H'FE92	TIOR4	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
H'FE94	TIER4	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
H'FE95	TSR4	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
H'FE96	TCNT4									
H'FE97										
H'FE98	TGR4A									
H'FE99										
H'FE9A	TGR4B									
H'FE9B										
H'FEA0	TCR5	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TP
H'FEA1	TMDR5	—	—	—	—	MD3	MD2	MD1	MD0	
H'FEA2	TIOR5	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
H'FEA4	TIER5	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
H'FEA5	TSR5	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
H'FEA6	TCNT5									
H'FEA7										
H'FEA8	TGR5A									
H'FEA9										
H'FEAA	TGR5B									
H'FEAB										

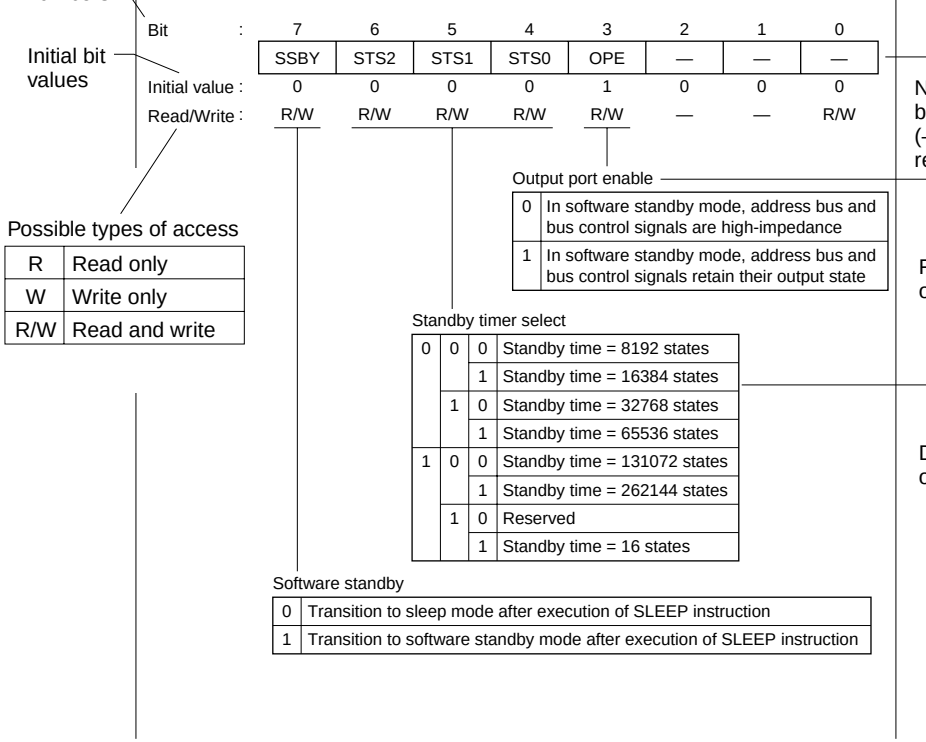
H'FEC5	IPRF	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
H'FEC6	IPRG	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
H'FEC7	IPRH	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
H'FEC8	IPRI	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
H'FEC9	IPRJ	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
H'FECA	IPRK	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
H'FECC	IPRM	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
H'FED0	ABWCR	ABW7	ABW6	ABW5	ABW4	ABW3	ABW2	ABW1	ABW0	Bus
H'FED1	ASTCR	AST7	AST6	AST5	AST4	AST3	AST2	AST1	AST0	contr
H'FED2	WCRH	W71	W70	W61	W60	W51	W50	W41	W40	
H'FED3	WCRL	W31	W30	W21	W20	W11	W10	W01	W00	
H'FED4	BCRH	ICIS1	ICIS0	BRSTRM	BRSTS1	BRSTS0	—	—	—	
H'FED5	BCRL	BRLE	BREQOE	—	—	—	—	WDBE	WAITE	
H'FEDB	RAMER*3	—	—	—	—	RAMS	RAM2	RAM1	RAM0	ROM
H'FF00	P1DR	P17DR	P16DR	P15DR	P14DR	P13DR	P12DR	P11DR	P10DR	I/O p
H'FF09	PADR	—	—	PA5DR*5	PA4DR*5	PA3DR	PA2DR	PA1DR	PA0DR	
H'FF0A	PBDR	PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR	
H'FF0B	PCDR	PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR	
H'FF0C	PDDR	PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR	
H'FF0D	PEDR	PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR	
H'FF0E	PFDR	PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1DR	PF0DR	
H'FF10	TCR0	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU
H'FF11	TMDR0	—	—	BFB	BFA	MD3	MD2	MD1	MD0	
H'FF12	TIOR0H	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
H'FF13	TIOR0L	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0	
H'FF14	TIER0	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	
H'FF15	TSR0	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
H'FF16	TCNT0									
H'FF17										

H'FF1F										
H'FF20	TCR1	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TP
H'FF21	TMDR1	—	—	—	—	MD3	MD2	MD1	MD0	
H'FF22	TIOR1	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
H'FF24	TIER1	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
H'FF25	TSR1	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
H'FF26	TCNT1									
H'FF27										
H'FF28	TGR1A									
H'FF29										
H'FF2A	TGR1B									
H'FF2B										
H'FF30	TCR2	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TP
H'FF31	TMDR2	—	—	—	—	MD3	MD2	MD1	MD0	
H'FF32	TIOR2	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
H'FF34	TIER2	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
H'FF35	TSR2	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
H'FF36	TCNT2									
H'FF37										
H'FF38	TGR2A									
H'FF39										
H'FF3A	TGR2B									
H'FF3B										
H'FF74	TCSR0	OVF	WT/$\overline{\text{IT}}$	TME	—	—	CKS2	CKS1	CKS0	WD
(Write)	TCNT0									
H'FF75 (Read)	TCNT0									
H'FF76 (Write)	RSTCSR	WOVF	RSTE	RSTS	—	—	—	—	—	
H'FF77 (Read)	RSTCSR	WOVF	RSTE	RSTS	—	—	—	—	—	

										inter
H'FF7D	RDR0									SCI0
H'FF7E	SCMR0	—	—	—	—	SDIR	SINV	—	SMIF	smar inter
H'FF80	SMR1	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI1
		GM	BLK	PE	O/$\overline{E}$	BCP1	BCP0	CKS1	CKS0	Sma inter
H'FF81	BRR1									SCI1
H'FF82	SCR1	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	smar inter
H'FF83	TDR1									
H'FF84	SSR1	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	SCI1
	SSR1	TDRE	RDRF	ORER	ERS	PER	TEND	MPB	MPBT	Sma inter
H'FF85	RDR1									SCI1
H'FF86	SCMR1	—	—	—	—	SDIR	SINV	—	SMIF	smar inter
H'FF88	SMR2	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI2
		GM	BLK	PE	O/$\overline{E}$	BCP1	BCP0	CKS1	CKS0	Sma inter
H'FF89	BRR2									SCI2
H'FF8A	SCR2	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	smar inter
H'FF8B	TDR2									
H'FF8C	SSR2	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	SCI2
	SSR2	TDRE	RDRF	ORER	ERS	PER	TEND	MPB	MPBT	Sma inter
H'FF8D	RDR2									SCI2
H'FF8E	SCMR2	—	—	—	—	SDIR	SINV	—	SMIF	smar inter

H'FF97		AD1	AD0	—	—	—	—	—	—	
H'FF98	ADCSR	ADF	ADIE	ADST	SCAN	CH3	CH2	CH1	CH0	
H'FF99	ADCR	TRGS1	TRGS0	—	—	CKS1	CKS0	—	—	
H'FFA2	TCSR1*6	OVF	WT/$\overline{IT}$	TME	PSS	RST/$\overline{NMI}$	CKS2	CKS1	CKS0	WD
(Write)	TCNT1*6									
H'FFA3 (Read)	TCNT1*6									
H'FFA8	FLMCR1*3	FWE	SWE1	ESU1	PSU1	EV1	PV1	E1	P1	RO
H'FFA9	FLMCR2*3	FLER	—	—	—	—	—	—	—	
H'FFAA	EBR1*3	EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0	
H'FFAB	EBR2*3	—	—	—	—	EB11	EB10	EB9	EB8	
H'FFAC	FLPWCR	PDWND	—	—	—	—	—	—	—	
H'FFB0	PORT1	P17	P16	P15	P14	P13	P12	P11	P10	I/O
H'FFB3	PORT4	P47	P46	P45	P44	P43	P42	P41	P40	
H'FFB8	PORT9	P97	P96	P95	P94	P93	P92	P91	P90	
H'FFB9	PORTA	—	—	PA5	PA4	PA3	PA2	PA1	PA0	
H'FFBA	PORTB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	
H'FFBB	PORTC	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	
H'FFBC	PORTD	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0	
H'FFBD	PORTE	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0	
H'FFBE	PORTF	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0	

Notes: 1. Located in on-chip RAM. The bus width is 32 bits when the DTC accesses register information, and 16 bits otherwise.
2. The address depends on the output trigger setting.
3. These registers are present in the F-ZTAT version, but not in the mask RO An undefined value will be returned if these registers are read in the mask version.
4. Valid only in the H8S/2626 Group; reserved bits in the H8S/2623 Group. Fo handling of these bits in register writes, see the individual register descripti respective sections.
5. Valid only in the H8S/2623 Group; reserved bits in the H8S/2626 Group. Fo handling of these bits in register writes, see the individual register descripti respective sections.
6. These registers are not available, and must not be accessed, in the H8S/26


Bit :
7 6 5 4 3 2 1 0
SSBY STS2 STS1 STS0 OPE — — —
Initial bit values
Initial value :
0 0 0 0 1 0 0 0
Read/Write :
R/W R/W R/W R/W R/W — — R/W
Possible types of access
R Read only
W Write only
R/W Read and write
Output port enable
0 In software standby mode, address bus and bus control signals are high-impedance
1 In software standby mode, address bus and bus control signals retain their output state
Standby timer select
0 0 0 Standby time = 8192 states
1 Standby time = 16384 states
1 0 Standby time = 32768 states
1 Standby time = 65536 states
1 0 0 Standby time = 131072 states
1 Standby time = 262144 states
1 0 Reserved
1 Standby time = 16 states
Software standby
0 Transition to sleep mode after execution of SLEEP instruction
1 Transition to software standby mode after execution of SLEEP instruction

DTC data tr

0	Byte-si
1	Word-s

DTC transfer mode

0	Destination sid area or block a
1	Source side is area or block a

DTC mode

0	0	Normal mode
	1	Repeat mode
1	0	Block transfer mo
	1	—

Destination address mode

0	—	DAR is fixed
1	0	DAR is incremented after a transfe (by +1 when Sz = 0; by +2 when Sz
	1	DAR is decremented after a transfe (by –1 when Sz = 0; by –2 when Sz

Source address mode

0	—	SAR is fixed
1	0	SAR is incremented after a transfer (by +1 when Sz = 0; by +2 when Sz = 1)
	1	SAR is decremented after a transfer (by –1 when Sz = 0; by –2 when Sz = 1)

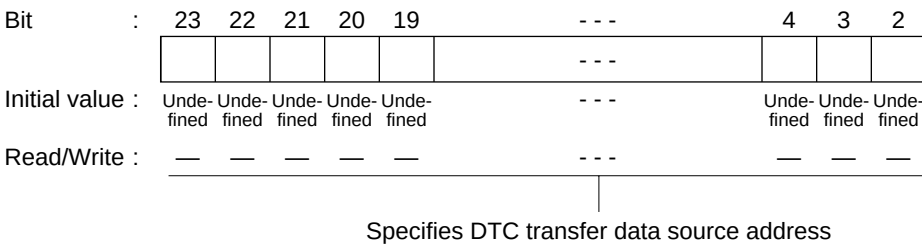
DTC interrupt select

0	After a data transfer ends, the CPU interrupt is disabl unless the transfer counter is 0
1	After a data transfer ends, the CPU interrupt is enable

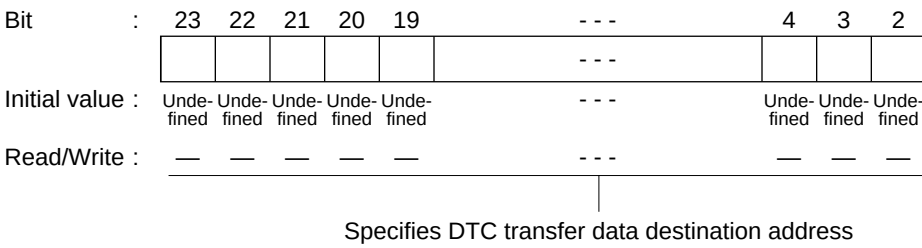
DTC chain transfer enable

0	End of DTC data transfer
1	DTC chain transfer

SAR—DTC Source Address Register **H'EBC0–H'EFBF**



DAR—DTC Destination Address Register **H'EBC0–H'EFBF**



Specifies the number of DTC data transfers

CRB—DTC Transfer Count Register B **H'EBC0–H'EFBF**

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
Initial value	:	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fine
Read/Write	:	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Specifies the number of DTC block data transfers

Reset request

0	Normal operating m (MCR0 = 0 and GS [Setting condition] When 0 is written a an HCAN reset
1	HCAN reset mode request

Halt request

0	HCAN normal operating
1	HCAN halt mode transitio

Message transmission method

0	Transmission order determined message identifier priority
1	Transmission order determined mailbox (buffer) number priority (TXPR1 > TXPR15)

HCAN sleep mode

0	HCAN sleep mode released
1	Transition to HCAN sleep mode enabled

HCAN sleep mode release

0	HCAN sleep mode release by CAN bus operation disabled
1	HCAN sleep mode release by CAN bus operation enabled

Bus off fla

0	[Rese Recov bus o
1	When (bus c

Transmit/receive wa

0	[Reset condition When TEC < 96 REC < 96 or TE
1	When TEC ≥ 96 REC ≥ 96

Message transmission status fla

0	Message transmission peri
1	[Reset condition] Idle period

Reset status bit

0	Normal operating state [Setting condition] After an HCAN internal reset
1	Configuration mode [Reset condition] MCR0 reset mode and sleep mo

Baud rate prescale

0	0	0	0	0	0	2 × system
0	0	0	0	0	1	4 × system
0	0	0	0	1	0	6 × system
:	:	:	:	:	:	
1	1	1	1	1	1	128 × syst

Resynchronization jump width

0	0	Max. bit synchronization width = 1 time quantum
	1	Max. bit synchronization width = 2 time quanta
1	0	Max. bit synchronization width = 3 time quanta
	1	Max. bit synchronization width = 4 time quanta

0	0	0	0	Setting prohibite
0	0	0	1	Setting prohibite
0	0	1	0	Setting prohibite
0	0	1	1	TSEG1 = 4 time
0	1	0	0	TSEG1 = 5 time
:	:	:	:	:
1	1	1	1	TSEG1 = 16 tim

Time segment 2

0	0	0	Setting prohibited
		1	TSEG2 = 2 time quanta
	1	0	TSEG2 = 3 time quanta
		1	TSEG2 = 4 time quanta
1	0	0	TSEG2 = 5 time quanta
		1	TSEG2 = 6 time quanta
	1	0	TSEG2 = 7 time quanta
		1	TSEG2 = 8 time quanta

Bit sample point

0	Bit sampling at one point (end of time segment 1 (TSEG1))
1	Bit sampling at three points (end of time segment 1 (TSEG1), a quantum before and after)

Note: For details, see section 15.2.3, Bit Configuration Register (BCR).

Bit	:	7	6	5	4	3	2	1
		MBCR15	MBCR14	MBCR13	MBCR12	MBCR11	MBCR10	MBCR9
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Mailbox setting register

0	Corresponding mailbox is set for trans
1	Corresponding mailbox is set for recep

Bit	:	7	6	5	4	3	2	1
		TXPR15	TXPR14	TXPR13	TXPR12	TXPR11	TXPR10	TXPR9
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Transmit wait register

0	Transmit message idle state in correspondi [Clearing condition] Message transmission completion and canc completion
1	Transmit wait for transmit message in corres mailbox (CAN bus arbitration)

Bit	:	7	6	5	4	3	2	1
		TXCR15	TXCR14	TXCR13	TXCR12	TXCR11	TXCR10	TXCR9
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Transmit wait cancel register

0	Transmit message cancellation idle s corresponding mailbox [Clearing condition] Completion of TXPR clearing (when message is canceled normally)
1	TXPR cleared for corresponding mai (transmit message cancellation)

Bit	:	7	6	5	4	3	2	1
		TXACK15	TXACK14	TXACK13	TXACK12	TXACK11	TXACK10	TXACK9
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Transmit acknowledge register

0	[Clearing condition] Writing 1
1	Completion of message transmiss for corresponding mailbox

Note: * Can only be written with 1 for flag clearing.

Bit	:	7	6	5	4	3	2	1
		ABACK15	ABACK14	ABACK13	ABACK12	ABACK11	ABACK10	ABACK9
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Abort acknowledge register

0	[Clearing condition] Writing 1
1	Completion of transmit message can for corresponding mailbox

Note: * Can only be written with 1 for flag clearing.

Bit	:	7	6	5	4	3	2	1
		RXPR15	RXPR14	RXPR13	RXPR12	RXPR11	RXPR10	RXPR9
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Receive complete register

0	[Clearing condition] Writing 1
1	Completion of message (data frame or rem reception in corresponding mailbox

Note: * Can only be written with 1 for flag clearing.

Bit	:	7	6	5	4	3	2	1
		RFPR15	RFPR14	RFPR13	RFPR12	RFPR11	RFPR10	RFPR9
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Remote request wait register

0	[Clearing condition] Writing 1
1	Completion of remote frame r in corresponding mailbox

Note: * Can only be written with 1 for flag clearing.

Reset interrupt flag

0	[Clearing condition] Writing 1
1	Hardware reset (HCAN modul software standby) [Setting condition] When reset processing is com a hardware reset (HCAN mod software standby)

Receive message interrupt flag

0	[Clearing condition] Clearing of all bits in RXPR (receive register) of mailbox for which receiv requests are enabled MBIMR
1	Data frame or remote frame received and stored in mailbox [Setting conditions] When data frame or remote frame reception is completed When corresponding MBIMR = 0

Remote frame request interrupt flag

0	[Clearing condition] Clearing of all bits in RFPR (remote requ wait register) of mailbox for which receive requests are enabled MBIMR
1	Remote frame received and stored in ma [Setting conditions] When remote frame reception is complet When corresponding MBIMR = 0

Bus off interrupt flag

0	[Clearing condition] Writing 1
1	Bus off state caused by transmit error [Setting condition] When TEC ≥ 256

Transmit overload warning interrupt flag

0	[Clearing condition] Writing 1
1	Error warning state caused by transmit error [Setting condition] When TEC ≥ 96

Receive overload warning interrupt flag

0	[Clearing condition] Writing 1
1	Error warning state caused by receive error [Setting condition] When REC ≥ 96

Overload frame/bus off recovery interrupt flag

0	[Clearing condition] Writing 1
1	Overload frame transmission or recovery from bus off state [Setting conditions] • Error active/passive state — When overload frame is transmitted • Bus off state — When 11 recessive bits are received 128 times (REC ≥ 128)

Error passive interrupt flag

0	[Clearing condition] Writing 1
1	Error passive state caused by transmit/receive error [Setting condition] When TEC ≥ 128 or REC ≥ 128

Note: * Can only be written with 1 for flag clearing.

	Writing 1
1	Transmit message has been or aborted, and new messag stored [Setting condition] When TXPR (transmit wait re cleared by completion of tran completion of transmission a

Unread interrupt flag

0	[Clearing condition] Clearing of all bits in UMSR (unread status register)
1	Unread message overwrite [Setting condition] When UMSR (unread message statu is set

Bus operation interrupt flag

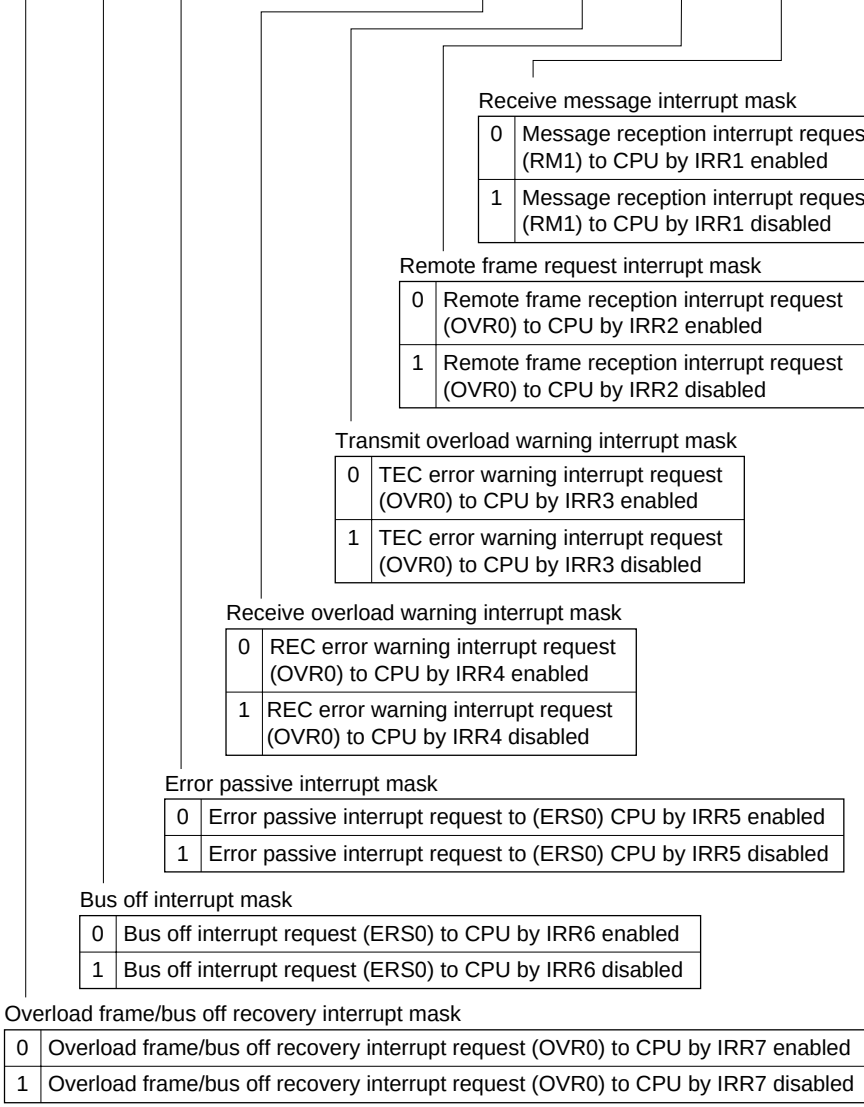
0	CAN bus idle state [Clearing condition] Writing 1
1	CAN bus operation in HCAN sleep mode [Setting condition] Bus operation (dominant bit detection) in HCAN

Note: * Can only be written with 1 for flag clearing.

Bit	:	7	6	5	4	3	2	1
		MBIMR15	MBIMR14	MBIMR13	MBIMR12	MBIMR11	MBIMR10	MBIMR9
Initial value	:	1	1	1	1	1	1	1
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Mailbox interrupt mask

0	[Transmitting] Interrupt request to CPU due to TXPI [Receiving] Interrupt request to CPU due to RXP
1	Interrupt requests to CPU disabled


Receive message interrupt mask
0 Message reception interrupt reques (RM1) to CPU by IRR1 enabled
1 Message reception interrupt reques (RM1) to CPU by IRR1 disabled
Remote frame request interrupt mask
0 Remote frame reception interrupt request (OVR0) to CPU by IRR2 enabled
1 Remote frame reception interrupt request (OVR0) to CPU by IRR2 disabled
Transmit overload warning interrupt mask
0 TEC error warning interrupt request (OVR0) to CPU by IRR3 enabled
1 TEC error warning interrupt request (OVR0) to CPU by IRR3 disabled
Receive overload warning interrupt mask
0 REC error warning interrupt request (OVR0) to CPU by IRR4 enabled
1 REC error warning interrupt request (OVR0) to CPU by IRR4 disabled
Error passive interrupt mask
0 Error passive interrupt request to (ERS0) CPU by IRR5 enabled
1 Error passive interrupt request to (ERS0) CPU by IRR5 disabled
Bus off interrupt mask
0 Bus off interrupt request (ERS0) to CPU by IRR6 enabled
1 Bus off interrupt request (ERS0) to CPU by IRR6 disabled
Overload frame/bus off recovery interrupt mask
0 Overload frame/bus off recovery interrupt request (OVR0) to CPU by IRR7 enabled
1 Overload frame/bus off recovery interrupt request (OVR0) to CPU by IRR7 disabled

Mailbox empty interrupt mask

0	Mailbox empty interrupt request to CPU by IRR8 enabled
1	Mailbox empty interrupt request to CPU by IRR8 disabled

Unread interrupt mask

0	Unread message overwrite interrupt request (OVR0) to CPU by IRR9 enabled
1	Unread message overwrite interrupt request (OVR0) to CPU by IRR9 disabled

Bus operation interrupt mask

0	Bus operation interrupt request (OVR0) to CPU by IRR12 enabled
1	Bus operation interrupt request (OVR0) to CPU by IRR12 disabled

TEC—Transmit Error Counter H'F819

TEC

Bit	:	7	6	5	4	3	2	1
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	R	R	R	R	R	R	R

UMSR—Unread Message Status Register H'F81A

UMSR

Bit	:	15	14	13	12	11	10	9
		UMSR7	UMSR6	UMSR5	UMSR4	UMSR3	UMSR2	UMSR1
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Bit	:	7	6	5	4	3	2	1
		UMSR15	UMSR14	UMSR13	UMSR12	UMSR11	UMSR10	UMSR9
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Unread message status flags

0	[Clearing condition] Writing 1
1	Unread receive message is overwritten by a new [Setting condition] When a new message is received before RXPR is

Note: * Can only be written with 1 for flag clearing.

Bit	:	7	6	5	4	3	2	1
		LAFML15	LAFML14	LAFML13	LAFML12	LAFML11	LAFML10	LAFML9
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

LAFMH

Bit	:	15	14	13	12	11	10	9
		LAFMH7	LAFMH6	LAFMH5	—	—	—	LAFMH1
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R	R	R	R/W

Bit	:	7	6	5	4	3	2	1
		LAFMH15	LAFMH14	LAFMH13	LAFMH12	LAFMH11	LAFMH10	LAFMH9
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

LAFMH Bits 7 to 0 and 15 to 13—11-bit identifier filter

0	Stored in MC0, MD0 (receive-only mailbox) depending on bit matc MC0 message identifier and receive message identifier (Care)
1	Stored in MC0, MD0 (receive-only mailbox) regardless of bit match MC0 message identifier and receive message identifier (Don't Car

LAFMH bits 9 and 8, LAFML Bits 15 to 0—18-bit identifier filter

0	Stored in MC0 (receive-only mailbox) depending on bit match betw message identifier and receive message identifier (Care)
1	Stored in MC0 (receive-only mailbox) regardless of bit match betw message identifier and receive message identifier (Don't Care)

		1	0	Data length = 6 bytes
			1	Data length = 7 bytes
1	0/1	0/1	0/1	Data length = 8 bytes

MCx[2] Bit :

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[3] Bit :

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[4] Bit :

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[5] Bit :

7	6	5	4	3	2	1	0
STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

Identifier extension

0	Standard format
1	Extended format

Extended identifier
These bits set the identifie
(extended identifier) of dat
frames and remote frames

Standard identifier
These bits set the identifier
(standard identifier) of data
frames and remote frames

Remote transmission request

0	Data frame
1	Remote frame

MCx[6] Bit :

7	6	5	4	3	2	1	0
STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

Standard identifier
These bits set the identifier (standard identifier) of data frames and remote frame

MCx[7] Bit :

7	6	5	4	3	2	1	0
EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[8] Bit :

7	6	5	4	3	2	1	0
EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

Extended identifier
These bits set the identifier (extended identifier) of data frames and remote frame

x = 0

		1	0	Data length = 6 byte
			1	Data length = 7 byte
1	0/1	0/1	0/1	Data length = 8 byte

MCx[2] Bit :

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[3] Bit :

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[4] Bit :

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[5] Bit :

7	6	5	4	3	2	1	0
STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID1
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

Identifier extension

0	Standard format
1	Extended format

Extended identifier
These bits set the identi
(extended identifier) of d
frames and remote fram

Standard identifier
These bits set the identifier
(standard identifier) of data
frames and remote frames

Remote transmission request

0	Data frame
1	Remote frame

MCx[6] Bit :

7	6	5	4	3	2	1	0
STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

Standard identifier
These bits set the identifier (standard identifier) of data frames and remote fram

MCx[7] Bit :

7	6	5	4	3	2	1	0
EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[8] Bit :

7	6	5	4	3	2	1	0
EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

Extended identifier
These bits set the identifier (extended identifier) of data frames and remote fram

x =

RENESAS

		1	0	Data length = 6 bytes
			1	Data length = 7 bytes
1	0/1	0/1	0/1	Data length = 8 bytes

MCx[2] Bit :

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[3] Bit :

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[4] Bit :

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[5] Bit :

7	6	5	4	3	2	1	0
STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

Identifier extension

0	Standard format
1	Extended format

Extended identifier
These bits set the identifie
(extended identifier) of dat
frames and remote frames

Standard identifier
These bits set the identifier
(standard identifier) of data
frames and remote frames

Remote transmission request

0	Data frame
1	Remote frame

MCx[6] Bit :

7	6	5	4	3	2	1	0
STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

Standard identifier
These bits set the identifier (standard identifier) of data frames and remote frame

MCx[7] Bit :

7	6	5	4	3	2	1	0
EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[8] Bit :

7	6	5	4	3	2	1	0
EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

Extended identifier
These bits set the identifier (extended identifier) of data frames and remote frame

x = 2

		1	0	Data length = 6 byte
			1	Data length = 7 byte
1	0/1	0/1	0/1	Data length = 8 byte

MCx[2] Bit :

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[3] Bit :

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[4] Bit :

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[5] Bit :

7	6	5	4	3	2	1	0
STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID1
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

Identifier extension

0	Standard format
1	Extended format

Extended identifier
These bits set the identi
(extended identifier) of d
frames and remote fram

Standard identifier
These bits set the identifier
(standard identifier) of data
frames and remote frames

Remote transmission request

0	Data frame
1	Remote frame

MCx[6] Bit :

7	6	5	4	3	2	1	0
STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

Standard identifier
These bits set the identifier (standard identifier) of data frames and remote fram

MCx[7] Bit :

7	6	5	4	3	2	1	0
EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[8] Bit :

7	6	5	4	3	2	1	0
EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

Extended identifier
These bits set the identifier (extended identifier) of data frames and remote fram

x =

RENESAS

		1	0	Data length = 6 bytes
			1	Data length = 7 bytes
1	0/1	0/1	0/1	Data length = 8 bytes

MCx[2] Bit :

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[3] Bit :

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[4] Bit :

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[5] Bit :

7	6	5	4	3	2	1	0
STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

Identifier extension

0	Standard format
1	Extended format

Extended identifier
These bits set the identifie
(extended identifier) of dat
frames and remote frames

Standard identifier
These bits set the identifier
(standard identifier) of data
frames and remote frames

Remote transmission request

0	Data frame
1	Remote frame

MCx[6] Bit :

7	6	5	4	3	2	1	0
STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

Standard identifier
These bits set the identifier (standard identifier) of data frames and remote frame

MCx[7] Bit :

7	6	5	4	3	2	1	0
EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[8] Bit :

7	6	5	4	3	2	1	0
EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

Extended identifier
These bits set the identifier (extended identifier) of data frames and remote frame

x = 4

		1	0	Data length = 6 byte
			1	Data length = 7 byte
1	0/1	0/1	0/1	Data length = 8 byte

MCx[2] Bit :

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[3] Bit :

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[4] Bit :

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[5] Bit :

7	6	5	4	3	2	1	0
STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID1
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

Identifier extension

0	Standard format
1	Extended format

Extended identifier
These bits set the identi
(extended identifier) of d
frames and remote fram

Standard identifier
These bits set the identifier
(standard identifier) of data
frames and remote frames

Remote transmission request

0	Data frame
1	Remote frame

MCx[6] Bit :

7	6	5	4	3	2	1	0
STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

Standard identifier
These bits set the identifier (standard identifier) of data frames and remote fram

MCx[7] Bit :

7	6	5	4	3	2	1	0
EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[8] Bit :

7	6	5	4	3	2	1	0
EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

Extended identifier
These bits set the identifier (extended identifier) of data frames and remote fram

x =

RENESAS

		1	0	Data length = 6 bytes
			1	Data length = 7 bytes
1	0/1	0/1	0/1	Data length = 8 bytes

MCx[2] Bit :

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[3] Bit :

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[4] Bit :

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[5] Bit :

7	6	5	4	3	2	1	0
STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

Identifier extension

0	Standard format
1	Extended format

Extended identifier
These bits set the identifie
(extended identifier) of dat
frames and remote frames

Standard identifier
These bits set the identifier
(standard identifier) of data
frames and remote frames

Remote transmission request

0	Data frame
1	Remote frame

MCx[6] Bit :

7	6	5	4	3	2	1	0
STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

Standard identifier
These bits set the identifier (standard identifier) of data frames and remote frame

MCx[7] Bit :

7	6	5	4	3	2	1	0
EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[8] Bit :

7	6	5	4	3	2	1	0
EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

Extended identifier
These bits set the identifier (extended identifier) of data frames and remote frame

x = 6

		1	0	Data length = 6 byte
			1	Data length = 7 byte
1	0/1	0/1	0/1	Data length = 8 byte

MCx[2] Bit :

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[3] Bit :

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[4] Bit :

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[5] Bit :

7	6	5	4	3	2	1	0
STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID1
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

Identifier extension

0	Standard format
1	Extended format

Extended identifier
These bits set the identi
(extended identifier) of d
frames and remote fram

Standard identifier
These bits set the identifier
(standard identifier) of data
frames and remote frames

Remote transmission request

0	Data frame
1	Remote frame

MCx[6] Bit :

7	6	5	4	3	2	1	0
STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

Standard identifier
These bits set the identifier (standard identifier) of data frames and remote fram

MCx[7] Bit :

7	6	5	4	3	2	1	0
EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[8] Bit :

7	6	5	4	3	2	1	0
EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

Extended identifier
These bits set the identifier (extended identifier) of data frames and remote fram

x =

RENESAS

		1	0	Data length = 6 bytes
			1	Data length = 7 bytes
1	0/1	0/1	0/1	Data length = 8 bytes

MCx[2] Bit :

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[3] Bit :

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[4] Bit :

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[5] Bit :

7	6	5	4	3	2	1	0
STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

Identifier extension

0	Standard format
1	Extended format

Extended identifier
These bits set the identifie
(extended identifier) of dat
frames and remote frames

Standard identifier
These bits set the identifier
(standard identifier) of data
frames and remote frames

Remote transmission request

0	Data frame
1	Remote frame

MCx[6] Bit :

7	6	5	4	3	2	1	0
STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

Standard identifier
These bits set the identifier (standard identifier) of data frames and remote frame

MCx[7] Bit :

7	6	5	4	3	2	1	0
EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[8] Bit :

7	6	5	4	3	2	1	0
EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

Extended identifier
These bits set the identifier (extended identifier) of data frames and remote frame

x = 8

		1	0	Data length = 6 byte
			1	Data length = 7 byte
1	0/1	0/1	0/1	Data length = 8 byte

MCx[2] Bit :

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[3] Bit :

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[4] Bit :

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[5] Bit :

7	6	5	4	3	2	1	0
STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID1
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

Identifier extension

0	Standard format
1	Extended format

Extended identifier
These bits set the identi
(extended identifier) of d
frames and remote fram

Standard identifier
These bits set the identifier
(standard identifier) of data
frames and remote frames

Remote transmission request

0	Data frame
1	Remote frame

MCx[6] Bit :

7	6	5	4	3	2	1	0
STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

Standard identifier
These bits set the identifier (standard identifier) of data frames and remote fram

MCx[7] Bit :

7	6	5	4	3	2	1	0
EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[8] Bit :

7	6	5	4	3	2	1	0
EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

Extended identifier
These bits set the identifier (extended identifier) of data frames and remote fram

x =

RENESAS

		1	0	Data length = 6 bytes
			1	Data length = 7 bytes
1	0/1	0/1	0/1	Data length = 8 bytes

MCx[2] Bit :

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[3] Bit :

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[4] Bit :

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[5] Bit :

7	6	5	4	3	2	1	0
STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

Identifier extension

0	Standard format
1	Extended format

Extended identifier
These bits set the identifie
(extended identifier) of dat
frames and remote frames

Standard identifier
These bits set the identifier
(standard identifier) of data
frames and remote frames

Remote transmission request

0	Data frame
1	Remote frame

MCx[6] Bit :

7	6	5	4	3	2	1	0
STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

Standard identifier
These bits set the identifier (standard identifier) of data frames and remote frame

MCx[7] Bit :

7	6	5	4	3	2	1	0
EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[8] Bit :

7	6	5	4	3	2	1	0
EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

Extended identifier
These bits set the identifier (extended identifier) of data frames and remote frame

x = 1

		1	0	Data length = 6 byte
			1	Data length = 7 byte
1	0/1	0/1	0/1	Data length = 8 byte

MCx[2] Bit :

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[3] Bit :

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[4] Bit :

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[5] Bit :

7	6	5	4	3	2	1	0
STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID1
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

Identifier extension

0	Standard format
1	Extended format

Extended identifier
These bits set the identi
(extended identifier) of d
frames and remote fram

Standard identifier
These bits set the identifier
(standard identifier) of data
frames and remote frames

Remote transmission request

0	Data frame
1	Remote frame

MCx[6] Bit :

7	6	5	4	3	2	1	0
STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

Standard identifier
These bits set the identifier (standard identifier) of data frames and remote fram

MCx[7] Bit :

7	6	5	4	3	2	1	0
EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[8] Bit :

7	6	5	4	3	2	1	0
EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

Extended identifier
These bits set the identifier (extended identifier) of data frames and remote fram

x =

RENESAS

		1	0	Data length = 6 bytes
			1	Data length = 7 bytes
1	0/1	0/1	0/1	Data length = 8 bytes

MCx[2] Bit :

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[3] Bit :

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[4] Bit :

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[5] Bit :

7	6	5	4	3	2	1	0
STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

Identifier extension

0	Standard format
1	Extended format

Extended identifier
These bits set the identifie
(extended identifier) of dat
frames and remote frames

Standard identifier
These bits set the identifier
(standard identifier) of data
frames and remote frames

Remote transmission request

0	Data frame
1	Remote frame

MCx[6] Bit :

7	6	5	4	3	2	1	0
STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

Standard identifier
These bits set the identifier (standard identifier) of data frames and remote frame

MCx[7] Bit :

7	6	5	4	3	2	1	0
EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[8] Bit :

7	6	5	4	3	2	1	0
EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

Extended identifier
These bits set the identifier (extended identifier) of data frames and remote frame

x = 1

		1	0	Data length = 6 byte
			1	Data length = 7 byte
1	0/1	0/1	0/1	Data length = 8 byte

MCx[2] Bit :

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[3] Bit :

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[4] Bit :

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[5] Bit :

7	6	5	4	3	2	1	0
STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID1
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

Identifier extension

0	Standard format
1	Extended format

Extended identifier
These bits set the identi
(extended identifier) of d
frames and remote fram

Standard identifier
These bits set the identifier
(standard identifier) of data
frames and remote frames

Remote transmission request

0	Data frame
1	Remote frame

MCx[6] Bit :

7	6	5	4	3	2	1	0
STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

Standard identifier
These bits set the identifier (standard identifier) of data frames and remote fram

MCx[7] Bit :

7	6	5	4	3	2	1	0
EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[8] Bit :

7	6	5	4	3	2	1	0
EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

Extended identifier
These bits set the identifier (extended identifier) of data frames and remote fram

x =

RENESAS

		1	0	Data length = 6 bytes
			1	Data length = 7 bytes
1	0/1	0/1	0/1	Data length = 8 bytes

MCx[2] Bit :

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[3] Bit :

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[4] Bit :

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[5] Bit :

7	6	5	4	3	2	1	0
STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

Identifier extension

0	Standard format
1	Extended format

Extended identifier
These bits set the identifie
(extended identifier) of dat
frames and remote frames

Standard identifier
These bits set the identifier
(standard identifier) of data
frames and remote frames

Remote transmission request

0	Data frame
1	Remote frame

MCx[6] Bit :

7	6	5	4	3	2	1	0
STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

Standard identifier
These bits set the identifier (standard identifier) of data frames and remote frame

MCx[7] Bit :

7	6	5	4	3	2	1	0
EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[8] Bit :

7	6	5	4	3	2	1	0
EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

Extended identifier
These bits set the identifier (extended identifier) of data frames and remote frame

x = 1

		1	0	Data length = 6 byte
			1	Data length = 7 byte
1	0/1	0/1	0/1	Data length = 8 byte

MCx[2] Bit :

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[3] Bit :

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[4] Bit :

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[5] Bit :

7	6	5	4	3	2	1	0
STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID1
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

Identifier extension

0	Standard format
1	Extended format

Extended identifier
These bits set the identi
(extended identifier) of d
frames and remote fram

Standard identifier
These bits set the identifier
(standard identifier) of data
frames and remote frames

Remote transmission request

0	Data frame
1	Remote frame

MCx[6] Bit :

7	6	5	4	3	2	1	0
STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

Standard identifier
These bits set the identifier (standard identifier) of data frames and remote fram

MCx[7] Bit :

7	6	5	4	3	2	1	0
EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

MCx[8] Bit :

7	6	5	4	3	2	1	0
EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID
Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefine
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value :
Read/Write :

Extended identifier
These bits set the identifier (extended identifier) of data frames and remote fram

x =

RENESAS

MD8—Message Data	**H'F8F0**
MD9—Message Data	**H'F8F8**
MD10—Message Data	**H'F900**
MD11—Message Data	**H'F908**
MD12—Message Data	**H'F910**
MD13—Message Data	**H'F918**
MD14—Message Data	**H'F920**
MD15—Message Data	**H'F928**

MDx [1]

Bit:	7	6	5	4	3	2	1
Initial value:	*	*	*	*	*	*	*
Read/Write:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx [2]

Bit:	7	6	5	4	3	2	1
Initial value:	*	*	*	*	*	*	*
Read/Write:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx [3]

Bit:	7	6	5	4	3	2	1
Initial value:	*	*	*	*	*	*	*
Read/Write:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1
Initial value:	*	*	*	*	*	*	*
Read/Write:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx [6]

Bit:	7	6	5	4	3	2	1
Initial value:	*	*	*	*	*	*	*
Read/Write:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx [7]

Bit:	7	6	5	4	3	2	1
Initial value:	*	*	*	*	*	*	*
Read/Write:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx [8]

Bit:	7	6	5	4	3	2	1
Initial value:	*	*	*	*	*	*	*
Read/Write:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	:	7	6	5	4	3	2	1
		DAOE1	DAOE0	DAE	—	—	—	—
Initial value	:	0	0	0	1	1	1	1
Read/Write	:	R/W	R/W	R/W	—	—	—	—

D/A enable

DAOE1	DAOE0	DAE	Description
0	0	*	Disables channel 2, 3 D/A conversio
	1	0	Enables channel 2 D/A conversion
			Disables channel 3 D/A conversion
		1	Enables channel 2, 3 D/A conversio
1	0	0	Disables channel 2 D/A conversion
			Enables channel 3 D/A conversion
		1	Enables channel 2, 3 D/A conversio
	1	*	Enables channel 2, 3 D/A conversio

*: Don't

D/A output enable 0

0	Disables analog output DA2
1	Enables channel 2 D/A conversion. Also enables analog output D

D/A output enable 1

0	Disables analog output DA3
1	Enables channel 3 D/A conversion. Also enables analog output DA3

Flash memory control register enable

0	Area H'FFFFA8 to H'FFFFAC flash control registers are not
1	Area H'FFFFA8 to H'FFFFAC flash control registers are sel

Output port enable

0	In software standby mode, addres bus control signals are high-imped
1	In software standby mode, addres bus control signals retain their outp

Standby timer select

0	0	0	Standby time = 8192 states
		1	Standby time = 16384 states
	1	0	Standby time = 32768 states
		1	Standby time = 65536 states
1	0	0	Standby time = 131072 states
		1	Standby time = 262144 states
	1	0	Reserved
		1	Standby time = 16 states

Software standby

0	Transition to sleep mode after execution of SLEEP instruction
1	Transition to software standby mode after execution of SLEEP inst

0	On-chip RAM
1	On-chip RAM

Note: When the D
the RAME b
set to 1.

NMI interrupt input edge se

0	Falling edge
1	Rising edge

Interrupt control mode select

0	0	Interrupt control mode 0
	1	Setting prohibited
1	0	Interrupt control mode 2
	1	Setting prohibited

Note: For details, see section 5.4.1 Interrupt Control Modes and Interrupt Operation.

Mac saturation

0	Non-saturating calculation for MAC instruction
1	Saturating calculation for MAC instruction

0	0	0	Bus master is in high-spee
		1	Medium-speed clock is φ/2
	1	0	Medium-speed clock is φ/4
		1	Medium-speed clock is φ/8
1	0	1	Medium-speed clock is φ/1
		1	Medium-speed clock is φ/3
	1	—	—

Frequency multiplication factor switching mode

0	Specified multiplication factor is valid after transition to software standby mode
1	Specified multiplication factor is valid imm after STC bits are rewritten

φ clock output control

PSTOP	High-Speed Mode, Medium-Speed Mode, Sub-Active Mode*	Sleep Mode, Sub-Sleep Mode*	Software Standby Mode, Watch Mode*, Direct Transition*	Hard Standb
0	φ output	φ output	Fixed high	High im
1	Fixed high	Fixed high	Fixed high	High im

Note: * Subclock functions (subactive mode, subsleep mode, and watch mode) direct transition are not available in the H8S/2623 Group, but are availab in the H8S/2626 Group.

Note: * Determined by pins MD_2 to MD_0.

Bit	:	7	6	5	4	3	2	1	0
		MSTPB7	MSTPB6	MSTPB5	MSTPB4	MSTPB3	MSTPB2	MSTPB1	MSTPB0
Initial value	:	1	1	1	1	1	1	1	1
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MSTPCRC

Bit	:	7	6	5	4	3	2	1	0
		MSTPC7	MSTPC6	MSTPC5	MSTPC4	MSTPC3	MSTPC2	MSTPC1	MSTPC0
Initial value	:	1	1	1	1	1	1	1	1
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Module stop mode specification

0	Module stop mode is cleared (initial value of MSTPA7 and MSTPA6)
1	Module stop mode is set (initial value of MSTPA5–0, MSTPB7–0, and MSTPC7–0)

MSTP bits and corresponding on-chip supporting modules

Register	Bit	Module
MSTPCRA	MSTPA7*	—
	MSTPA6	Data transfer controller (DTC)
	MSTPA5	16-bit timer pulse unit (TPU)
	MSTPA4*	—
	MSTPA3	Programmable pulse generator (PPG)
	MSTPA2*	—
	MSTPA1	A/D converter
	MSTPA0*	—
MSTPCRB	MSTPB7	Serial communication interface 0 (SCI0)
	MSTPB6	Serial communication interface 1 (SCI1)
	MSTPB5	Serial communication interface 2 (SCI2)
	MSTPB4*	—
	MSTPB3*	—
	MSTPB2*	—
	MSTPB1*	—
	MSTPB0*	—
MSTPCRC	MSTPC7*	—
	MSTPC6*	—
	MSTPC5	D/A converter (channels 2, 3)
	MSTPC4	PC break controller (PBC)
	MSTPC3	HCAN
	MSTPC2*	—
	MSTPC1*	—
	MSTPC0*	—

Notes: * MSTPA7 is a readable/writable bit with an initial value of 0.
MSTPA4, MSTPA2, MSTPA0, MSTPB4 to MSTPB0, MSTPC7 to MSTPC4, and MSTPC2 to MSTPC0 are readable/writable bits with an initial value of 1 and should always be written with

Address output enable

0	0	0	0	A8–A23 address output disabled
			1	A8 address output enabled; A9–A23 address output disa
		1	0	A8, A9 address output enabled; A10–A23 address outpu
			1	A8–A10 address output enabled; A11–A23 address outp
	1	0	0	A8–A11 address output enabled; A12–A23 address outp
			1	A8–A12 address output enabled; A13–A23 address outp
		1	0	A8–A13 address output enabled; A14–A23 address outp
			1	A8–A14 address output enabled; A15–A23 address outp
1	0	0	0	A8–A15 address output enabled; A16–A23 address outp
			1	A8–A16 address output enabled; A17–A23 address outp
		1	0	A8–A17 address output enabled; A18–A23 address outp
			1	A8–A18 address output enabled; A19–A23 address outp
	1	0	0	A8–A19 address output enabled; A20–A23 address outp
			1	A8–A20 address output enabled; A21–A23 address outp
		1	0	A8–A21 address output enabled; A22, A23 address outp
			1	A8–A23 address output enabled

BUZZ output enable

0	Functions as PF1 I/O pin
1	Functions as BUZZ output pin

Notes: 1. In expanded mode with ROM, bits AE3 to AE0 are initialized to B'0000
In ROMless expanded mode, bits AE3 to AE0 are initialized to B'1101
Address pins A0 to A7 are made address outputs by setting the corres
DDR bits to 1.
2. This bit is valid only in the H8S/2626 Group; in the H8S/2623 Group, 0
written to this bit.

	1	× 2
1	0	× 4
	1	Do not se

Note: A system clock frequ
by the multiplication f
and STC0) should no
maximum operating f
defined in section 22
Characteristics.

Oscillator circuit feedback resistor control

0	Feedback resistor ON when main clo OFF when not operation
1	Feedback resistor OFF

Subclock enable

0	Subclock generation enabled
1	Subclock generation disabled

Note: This bit is valid only in the H8S/2626 Group; in t
H8S/2623 Group, 0 must be written to this bit.

Noise elimination sampling frequency select

0	Sampling uses ϕ/32 clock
1	Sampling uses ϕ/4 clock

Note: This bit is valid only in the H8S/2626 Group; in the
H8S/2623 Group, 0 must be written to this bit.

Low-speed ON flag

0	• When the SLEEP command is executed in high-speed mode or medium-speed mode transfers to sleep mode, software standby mode, or watch mode* • When the SLEEP command is executed in sub-active mode, operation transfers to w or directly to high-speed mode • Operation transfers to high-speed mode after watch mode is canceled
1	• When the SLEEP command is executed in high-speed mode, operation transfers to or sub-active mode • When the SLEEP command is executed in sub-active mode, operation transfers to s mode or watch mode • Operation transfers to sub-active mode immediately watch mode is canceled

Notes: This bit is valid only in the H8S/2626 Group; in the H8S/2623 Group, 0 must be writt
* Always select high-speed mode when transferring to watch mode or sub-active m

Direct transfer ON flag

0	• When the SLEEP command is executed in high-speed mode or medium-speed mode, operation transfers to sleep mode, software standby mode, or watch mode* • When the SLEEP command is executed in sub-active mode, operation transfers to sub-sleep mode or watch mode
1	• When the SLEEP command is executed in high-speed mode or medium-speed mode, operation transfers directly to sub-active mode, or transfers to sleep mode or software standby mode • When the SLEEP command is executed in sub-active mode, operation transfers directly to high-speed mode or transfers to sub-sleep mode

Notes: This bit is valid only in the H8S/2626 Group; in the H8S/2623 Group, 0 must be written to this bit.
* Always select high-speed mode when transferring to watch mode or sub-active mode.

Break address specification

Break interrupt enable

0	PC break interrupts ar
1	PC break interrupts ar

Break condition select

0	0	Instruction fetch is used as break cond
	1	Data read cycle is used as break cond
1	0	Data write cycle is used as break cond
	1	Data read/write cycle is used as break

Break address mask register

0	0	0	All BARA bits are unmasked and included in b conditions
		1	BAA0 (lowest bit) is masked, and not included conditions
	1	0	BAA1–0 (lower 2 bits) are masked, and not inc in break conditions
		1	BAA2–0 (lower 3 bits) are masked, and not inc in break conditions
1	0	0	BAA3–0 (lower 4 bits) are masked, and not inc in break conditions
		1	BAA7–0 (lower 8 bits) are masked, and not inc in break conditions
	1	0	BAA11–0 (lower 12 bits) are masked, and not in break conditions
		1	BAA15–0 (lower 16 bits) are masked, and not in break conditions

CPU cycle/DTC cycle select A

0	PC break is performed when CPU is bus master
1	PC break is performed when CPU or DTC is bus master

Condition match flag

0	[Clearing condition] When 0 is written to CMFA after reading CMFA = 1
1	[Setting condition] When a condition set for channel A is satisfied

Notes: The bit configuration of BCRB is the same as that of BCRA, except that BCRB performs break for channel B.

* Can only be written with 0 for flag clearing.

IRQ5 and IRQ4 sense

ISCRL

Bit :	7	6	5	4	3	2	1
	IRQ3SCB	IRQ3SCA	IRQ2SCB	IRQ2SCA	IRQ1SCB	IRQ1SCA	IRQ0SCB
Initial value :	0	0	0	0	0	0	0
Read/Write :	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IRQ3 to IRQ0 sense control

IRQnSCB	IRQnSCA	Interrupt Request Gener
0	0	Low level of $\overline{\text{IRQn}}$ input
	1	Falling edge of $\overline{\text{IRQn}}$ input
1	0	Rising edge of $\overline{\text{IRQn}}$ input
	1	Rising and falling edges of $\overline{\text{IRQ}}$

IRQn enable

0	IRQn interrupt is disal
1	IRQn interrupt is enab

IRQ5 to IRQ0 interrupt request status indication

0	[Clearing conditions] • Cleared by reading IRQnF flag when IRQnF = 1, then to IRQnF flag • When interrupt exception handling is executed when I detection is set (IRQnSCB = IRQnSCA = 0) and $\overline{\text{IRQn}}$ high • When IRQn interrupt exception handling is executed w falling, rising, or both-edge detection is set (IRQnSCB IRQnSCA = 1) • When the DTC is activated by an IRQn interrupt, and bit in MRB of the DTC is cleared to 0
1	[Setting conditions] • When $\overline{\text{IRQn}}$ input goes low when low-level detection is (IRQnSCB = IRQnSCA = 0) • When a falling edge occurs in $\overline{\text{IRQn}}$ input when falling detection is set (IRQnSCB = 0, IRQnSCA = 1) • When a rising edge occurs in $\overline{\text{IRQn}}$ input when rising detection is set (IRQnSCB = 1, IRQnSCA = 0) • When a falling or rising edge occurs in $\overline{\text{IRQn}}$ input wh edge detection is set (IRQnSCB = IRQnSCA = 1)

Note: * Can only be written with 0 for flag clearing.

DTC activation enable

0	DTC activation by interrupt is disabled [Clearing conditions] • When data transfer ends while the DISEL bit i • When the specified number of transfers are co
1	DTC activation by interrupt is enabled [Maintenance condition] When the DISEL bit is 0 and the specified numb of transfers have not been completed

Interrupt Sources and DTCER Bits

Register	Bit						
	7	6	5	4	3	2	1
DTCERA	IRQ0	IRQ1	IRQ2	IRQ3	IRQ4	IRQ5	—
DTCERB	—	ADI	TGI0A	TGI0B	TGI0C	TGI0D	TGI1A
DTCERC	TGI2A	TGI2B	TGI3A	TGI3B	TGI3C	TGI3D	TGI4A
DTCERD	—	—	TGI5A	TGI5B	—	—	—
DTCERE	—	—	—	—	RXI0	TXI0	RXI1
DTCERF	RXI2	TXI2	—	—	—	—	—
DTCERG	—	—	RM0	—	—	—	—

DTC software activation enable

0	DTC software activation is disabled [Clearing conditions] • When the DISEL bit is 0 and the specified number of transfers h been completed • When 0 is written after a software-activated data transfer interru (SWDTEND) request has been sent to the CPU
1	DTC software activation is enabled [Maintenance conditions] • When data transfer ends while the DISEL bit is 1 • When the specified number of transfers are completed • During data transfer activated by software

Notes: 1. Only 1 can be written to the SWDTE bit.
2. Bits DTVEC6 to DTVEC0 can be written to when SWDTE = 0.

Group 0 compare match select

0	0	Compare match in TPU
	1	Compare match in TPU
1	0	Compare match in TPU
	1	Compare match in TPU

Group 1 compare match select

0	0	Compare match in TPU char
	1	Compare match in TPU char
1	0	Compare match in TPU char
	1	Compare match in TPU char

Group 2 compare match select

0	0	Compare match in TPU channel 0
	1	Compare match in TPU channel 1
1	0	Compare match in TPU channel 2
	1	Compare match in TPU channel 3

Group 3 compare match select

0	0	Compare match in TPU channel 0
	1	Compare match in TPU channel 1
1	0	Compare match in TPU channel 2
	1	Compare match in TPU channel 3

0	Normal operation in pul 0 (output values update match A in the selected
1	Non-overlapping opera output group 0 (1 outpu can be performed indep compare match A and TPU channel)

Group 1 non-overlap

0	Normal operation in pulse ou 1 (output values updated at match A in the selected TPU
1	Non-overlapping operation in output group 1 (1 output and can be performed independe compare match A and B in th TPU channel)

Group 2 non-overlap

0	Normal operation in pulse output 2 (output values updated at com match A in the selected TPU cha
1	Non-overlapping operation in pu output group 2 (1 output and 0 o can be performed independently compare match A and B in the s TPU channel)

Group 3 non-overlap

0	Normal operation in pulse output group 3 (output values updated at compare match A in the selected TPU channel)
1	Non-overlapping operation in pulse output group 3 (1 output and 0 output can be performed independently at compare match A and B in the selected TPU channel)

Group 0 invert

0	Inverted output for pulse output group 0 (low-level output at pin fo
1	Direct output for pulse output group 0 (high-level output at pin for

Group 1 invert

0	Inverted output for pulse output group 1 (low-level output at pin for a 1 in
1	Direct output for pulse output group 1 (high-level output at pin for a 1 in P

Group 2 invert

0	Inverted output for pulse output group 2 (low-level output at pin for a 1 in PODR
1	Direct output for pulse output group 2 (high-level output at pin for a 1 in PODRH

Group 3 invert

0	Inverted output for pulse output group 3 (low-level output at pin for a 1 in PODRH)
1	Direct output for pulse output group 3 (high-level output at pin for a 1 in PODRH)

Next data enable

0	Pulse outputs PO15 to PO8 are disabled (transfer from NDR15–NDR8 to POD15–POD8 is disable
1	Pulse outputs PO15 to PO8 are enabled (transfer from NDR15–NDR8 to POD15–POD8 is enable

NDERL—Next Data Enable Register L **H'FE29**

Bit :	7	6	5	4	3	2	1
	NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1
Initial value :	0	0	0	0	0	0	0
Read/Write :	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Next data enable

0	Pulse outputs PO7 to PO0 are disabled (transfer from NDR7–NDR0 to POD7–POD0 is disabled)
1	Pulse outputs PO7 to PO0 are enabled (transfer from NDR7–NDR0 to POD7–POD0 is enabled)

PODRL—Output Data Register L **H'FE2B**

Bit	:	7	6	5	4	3	2	1
		POD7	POD6	POD5	POD4	POD3	POD2	POD1
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * A bit that has been set for pulse output by NDER is read-only.

Initial value :	0	0	0	0	0	0	0
Read/Write :	R/W	R/W	R/W	R/W	R/W	R/W	R/W

H'FE2E

Bit :	7	6	5	4	3	2	1
	—	—	—	—	—	—	—
Initial value :	1	1	1	1	1	1	1
Read/Write :	—	—	—	—	—	—	—

When pulse output group output triggers are different

H'FE2C

Bit :	7	6	5	4	3	2	1
	NDR15	NDR14	NDR13	NDR12	—	—	—
Initial value :	0	0	0	0	1	1	1
Read/Write :	R/W	R/W	R/W	R/W	—	—	—

H'FE2E

Bit :	7	6	5	4	3	2	1
	—	—	—	—	NDR11	NDR10	NDR9
Initial value :	1	1	1	1	0	0	0
Read/Write :	—	—	—	—	R/W	R/W	R/W

Note: For details see section 11.2.4, Notes on NDR Access.

Initial value :	0	0	0	0	0	0	0
Read/Write :	R/W	R/W	R/W	R/W	R/W	R/W	R/W

H'FE2F

Bit :	7	6	5	4	3	2	1
	—	—	—	—	—	—	—
Initial value :	1	1	1	1	1	1	1
Read/Write :	—	—	—	—	—	—	—

When pulse output group output triggers are different

H'FE2D

Bit :	7	6	5	4	3	2	1
	NDR7	NDR6	NDR5	NDR4	—	—	—
Initial value :	0	0	0	0	1	1	1
Read/Write :	R/W	R/W	R/W	R/W	—	—	—

H'FE2F

Bit :	7	6	5	4	3	2	1
	—	—	—	—	NDR3	NDR2	NDR1
Initial value :	1	1	1	1	0	0	0
Read/Write :	—	—	—	—	R/W	R/W	R/W

Note: For details see section 11.2.4, Notes on NDR Access.

PADDR—Port A Data Direction Register H'FE39

Bit :	7	6	5	4	3	2	1
	—	—	PA5DDR*	PA4DDR*	PA3DDR	PA2DDR	PA1DDR
Initial value :	Undefined	Undefined	0	0	0	0	0
Read/Write :	—	—	W	W	W	W	W

Specification of input or output for port A pi

Note: * Reserved bits in the H8S/2626 Group.

PBDDR—Port B Data Direction Register H'FE3A

Bit :	7	6	5	4	3	2	1
	PB7DDR	PB6DDR	PB5DDR	PB4DDR	PB3DDR	PB2DDR	PB1DDR
Initial value :	0	0	0	0	0	0	0
Read/Write :	W	W	W	W	W	W	W

Specification of input or output for port B pins

Specification of input or output for port C pins

PDDDR—Port D Data Direction Register H'FE3C

Bit	:	7	6	5	4	3	2	1
		PD7DDR	PD6DDR	PD5DDR	PD4DDR	PD3DDR	PD2DDR	PD1DDR
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	W	W	W	W	W	W	W

Specification of input or output for port D pins

PEDDR—Port E Data Direction Register H'FE3D

Bit	:	7	6	5	4	3	2	1
		PE7DDR	PE6DDR	PE5DDR	PE4DDR	PE3DDR	PE2DDR	PE1DDR
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	W	W	W	W	W	W	W

Specification of input or output for port E pins

Initial value :	0	0	0	0	0	0	0
Read/Write :	W	W	W	W	W	W	W

Specification of input or output for port F pins

PAPCR—Port A MOS Pull-Up Control Register H'FE40

Bit :	7	6	5	4	3	2	1
	—	—	PA5PCR*	PA4PCR*	PA3PCR	PA2PCR	PA1PCR
Initial value :	Undefined	Undefined	0	0	0	0	0
Read/Write :	—	—	R/W	R/W	R/W	R/W	R/W

Bit-by-bit control of port A MOS input pull-ups

Note: * Reserved bits in the H8S/2626 Group.

PBPCR—Port B MOS Pull-Up Control Register H'FE41

Bit :	7	6	5	4	3	2	1
	PB7PCR	PB6PCR	PB5PCR	PB4PCR	PB3PCR	PB2PCR	PB1PCR
Initial value :	0	0	0	0	0	0	0
Read/Write :	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit-by-bit control of port B MOS input pull-ups

Bit-by-bit control of port C MOS input pull-ups

PDPCR—Port D MOS Pull-Up Control Register H'FE43

Bit	:	7	6	5	4	3	2	1
		PD7PCR	PD6PCR	PD5PCR	PD4PCR	PD3PCR	PD2PCR	PD1PCR
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit-by-bit control of port D MOS input pull-ups

PEPCR—Port E MOS Pull-Up Control Register H'FE44

Bit	:	7	6	5	4	3	2	1
		PE7PCR	PE6PCR	PE5PCR	PE4PCR	PE3PCR	PE2PCR	PE1PCR
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit-by-bit control of port E MOS input pull-ups

PMOS on/off control for port A pins (PA3 to PA0)

Note: * Reserved bits in the H8S/2626 Group.

PBODR—Port B Open Drain Control Register H'FE48

Bit	:	7	6	5	4	3	2	1
		PB7ODR	PB6ODR	PB5ODR	PB4ODR	PB3ODR	PB2ODR	PB1ODR
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PMOS on/off control for port B pins (PB7 to PB0)

PCODR—Port C Open Drain Control Register H'FE49

Bit	:	7	6	5	4	3	2	1
		PC7ODR	PC6ODR	PC5ODR	PC4ODR	PC3ODR	PC2ODR	PC1ODR
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PMOS on/off control for port C pins (PC7 to PC0)

Timer prescaler

0	0	0	Internal clock: counts on ϕ/1
		1	Internal clock: counts on ϕ/4
	1	0	Internal clock: counts on ϕ/16
		1	Internal clock: counts on ϕ/64
1	0	0	External clock: counts on TCLK
		1	Internal clock: counts on ϕ/1024
	1	0	Internal clock: counts on ϕ/256
		1	Internal clock: counts on ϕ/4096

Input clock edge select

0	0	Count at rising edge
	1	Count at falling edge
1	—	Count at both edges

Note: Internal clock edge selection is valid when th
input clock is ϕ/4 or slower. This setting is ig
if is ϕ/1 is selected as the input clock.

Counter clear

0	0	0	TCNT clearing disabled
		1	TCNT cleared by TGRA compare match/input capture
	1	0	TCNT cleared by TGRB compare match/input capture
		1	TCNT cleared by counter clearing for another channel performing synchronous clearing/synchronous operation*1
1	0	0	TCNT clearing disabled
		1	TCNT cleared by TGRC compare match/input capture*2
	1	0	TCNT cleared by TGRD compare match/input capture*2
		1	TCNT cleared by counter clearing for another channel performing synchronous clearing/synchronous operation*1

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

0	0	0	0	Normal operati
			1	Reserved
		1	0	PWM mode 1
			1	PWM mode 2
	1	0	0	Phase counting
			1	Phase counting
		1	0	Phase counting
			1	Phase counting
1	*	*	*	—

*:

Notes: 1. MD3 is a reserved b
In a write, it should
written with 0.
2. Phase counting mo
be set for channel 3
case, 0 should alwa
written to MD2.

Buffer operation setting A

0	TGRA operates normally
1	TGRA and TGRC used together f operation

Buffer operation setting B

0	TGRB operates normally
1	TGRB and TGRD used together for buffer operation

0	0	0	0	TGR3A is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare n
		1	0			1 output at compare n
			1			Toggle output at comp
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare n
		1	0			1 output at compare n
			1			Toggle output at comp
1	0	0	0	TGR3A is input capture register	Capture input source is TIOCA3 pin	Input capture at rising
			1			Input capture at falling
		1	*			Input capture at both e
	1	*	*		Capture input source is channel 4/count clock	Input capture at TCNT count-down

TGR3B I/O control

0	0	0	0	TGR3B is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	0	0	0	TGR3B is input capture register	Capture input source is TIOCB3 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Capture input source is channel 4/count clock	Input capture at TCNT4 count-up/ count-down*1

*: Don't care

Note: 1. When bits TPSC2 to TPSC0 in TCR4 are set to B'000 and ø/1 is used as the TCNT4 count clock, this setting is invalid and input capture is not generated.

			1	compare register	Initial output is 0 output	0 output at compare match	
		1	0			1 output at compare match	
			1			Toggle output at compare ma	
	1	0	0		Output disabled		
			1		Initial output is 1 output	0 output at compare match	
		1	0			1 output at compare match	
			1			Toggle output at compare ma	
1	0	0	0	TGR3C is input capture register	Capture input source is TIOCC3 pin	Input capture at rising edge	
			1			Input capture at falling edge	
		1	*			Input capture at both edges	
	1	*	*		Capture input source is channel 4/count clock	Input capture at TCNT4 coun count-down	

TGR3D I/O control

0	0	0	0	TGR3D is output compare register*2	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	0	0	0	TGR3D is input capture register*2	Capture input source is TIOCD3 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Capture input source is channel 4/count clock	Input capture at TCNT4 count-up/ count-down*1

*: Don't care

Notes: 1. When bits TPSC2 to TPSC0 in TCR4 are set to B'000 and ϕ/1 is used as the TCNT4 count clock, this setting is invalid and input capture is not generated.
2. When the BFB bit in TMDR3 is set to 1 and TGR3D is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Note: When TGRC or TGRD is designated for buffer operation, this setting is invalid and the register operates register.

TGR interrupt enal

0	Interrupt requ by TGFA bit d
1	Interrupt requ by TGFA bit e

TGR interrupt enable

0	Interrupt requests by TGFB bit disal
1	Interrupt requests by TGFB bit enab

TGR interrupt enable C

0	Interrupt requests (T by TGFC bit disabled
1	Interrupt requests (T by TGFC bit enabled

TGR interrupt enable D

0	Interrupt requests (TGID) by TGFD bit disabled
1	Interrupt requests (TGID) by TGFD bit enabled

Overflow interrupt enable

0	Interrupt requests (TCIV) by TCFV
1	Interrupt requests (TCIV) by TCFV

A/D conversion start request enable

0	A/D conversion start request generation disabled
1	A/D conversion start request generation enabled

TGR input capture/output compare flag A

0	[Clearing conditions] • When DTC is activated by TGIA interrupt bit in DTC's MRB is 0 • When 0 is written to TGFA after reading
1	[Setting conditions] • When TCNT = TGRA while TGRA is func output compare register • When TCNT value is transferred to TGR capture signal while TGRA is functioning capture register

TGR input capture/output compare flag B

0	[Clearing conditions] • When DTC is activated by TGIB interrupt, and DISE bit in DTC's MRB is 0 • When 0 is written to TGFB after reading TGFB = 1
1	[Setting conditions] • When TCNT = TGRB while TGRB is functioning as output compare register • When TCNT value is transferred to TGRB by input capture signal while TGRB is functioning as input capture register

TGR input capture/output compare flag C

0	[Clearing conditions] • When DTC is activated by TGIC interrupt, and DISEL bit in DTC • When 0 is written to TGFC after reading TGFC = 1
1	[Setting conditions] • When TCNT = TGRC while TGRC is functioning as output comp • When TCNT value is transferred to TGRC by input capture sign TGRC is functioning as input capture register

TGR input capture/output compare flag D

0	[Clearing conditions] • When DTC is activated by TGID interrupt, and DISEL bit in DTC's MRB • When 0 is written to TGFD after reading TGFD = 1
1	[Setting conditions] • When TCNT = TGRD while TGRD is functioning as output compare regi • When TCNT value is transferred to TGRD by input capture signal while TGRD is functioning as input capture register

Overflow flag

0	[Clearing condition] When 0 is written to TCFV after reading TCFV = 1
1	[Setting condition] When the TCNT value overflows (changes from H'FFFF to H'0000)

Note: * Can only be written with 0 for flag clearing.

Up counter

TGR3A—Timer General Register 3A	**H'FE88**
TGR3B—Timer General Register 3B	**H'FE8A**
TGR3C—Timer General Register 3C	**H'FE8C**
TGR3D—Timer General Register 3D	**H'FE8E**

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
Initial value	:	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Time prescaler

0	0	0	Internal clock: counts on φ/1
		1	Internal clock: counts on φ/4
	1	0	Internal clock: counts on φ/16
		1	Internal clock: counts on φ/64
1	0	0	External clock: counts on TCLKA
		1	External clock: counts on TCLKC
	1	0	Internal clock: counts on φ/1024
		1	Counts on TCNT5 overflow/unde

Note: This setting is invalid when channel 4 counting mode.

Input clock edge select

0	0	Count at rising edge
	1	Count at falling edge
1	—	Count at both edges

Note: This setting is invalid when channel 4 is in pha counting mode.
This setting is ignored if the input clock is φ/1, overflow/underflow of another channel is sele

Counter clear

0	0	0	TCNT clearing disabled
		1	TCNT cleared by TGRA compare match/input ca
	1	0	TCNT cleared by TGRB compare match/input ca
		1	TCNT cleared by counter clearing for another ch performing synchronous clearing/synchronous o

Note: * Synchronous operation setting is performed by setti SYNC bit in TSYR to 1.

0	0	0	0	Normal opera
			1	Reserved
		1	0	PWM mode 1
			1	PWM mode 2
	1	0	0	Phase countir
			1	Phase countir
		1	0	Phase countir
			1	Phase countir
1	*	*	*	—

*

Note: MD3 is a reserved bit.
it should always be wri

0	0	0	0	TGR4A is output compare register	Output disabled		
			1		Initial output is 0 output	0 output at compare ma	
		1	0			1 output at compare ma	
			1			Toggle output at compa	
	1	0	0		Output disabled		
			1		Initial output is 1 output	0 output at compare ma	
		1	0			1 output at compare ma	
			1			Toggle output at compa	
1	0	0	0	TGR4A is input capture register	Capture input source is TIOCA4 pin	Input capture at rising e	
			1			Input capture at falling e	
		1	*			Input capture at both ed	
	1	*	*		Capture input source is TGR3A compare match/ input capture	Input capture at genera TGR3A compare match capture	

TGR4B I/O control

0	0	0	0	TGR4B is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	0	0	0	TGR4B is input capture register	Capture input source is TIOCB4 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Capture input source is TGR3C compare match/ input capture	Input capture at generation of TGR3C compare match/input capture

*: Don't care

TGI interrupt enab

0	Interrupt requ by TGFA bit d
1	Interrupt requ by TGFA bit e

TGI interrupt enable B

0	Interrupt requests (T by TGFB bit disabled
1	Interrupt requests (T by TGFB bit enabled

Overflow interrupt enable

0	Interrupt requests (TCIV) by TCF
1	Interrupt requests (TCIV) by TCF

Underflow interrupt enable

0	Interrupt requests (TCIU) by TCFU disabled
1	Interrupt requests (TCIU) by TCFU enabled

A/D conversion start request enable

0	A/D conversion start request generation disabled
1	A/D conversion start request generation enabled

TGR input capture/output compare fla

0	[Clearing conditions] • When DTC is activated by TGIA and DISEL bit in DTC's MRB is • When 0 is written to TGFA after TGFA = 1
1	[Setting conditions] • When TCNT = TGRA while TGI functioning as output compare r • When TCNT value is transferre TGRA by input capture signal w is functioning as input capture r

TGR input capture/output compare flag B

0	[Clearing conditions] • When DTC is activated by TGIB in and DISEL bit in DTC's MRB is 0 • When 0 is written to TGFB after re TGFB = 1
1	[Setting conditions] • When TCNT = TGRB while TGRB functioning as output compare reg • When TCNT value is transferred to TGRB by input capture signal whil is functioning as input capture regi

Overflow flag

0	[Clearing condition] When 0 is written to TCFV after reading TCFV = 1
1	[Setting condition] When the TCNT value overflows (changes from H'FFFF

Underflow flag

0	[Clearing condition] When 0 is written to TCFU after reading TCFU = 1
1	[Setting condition] When the TCNT value underflows (changes from H'0000 to H'FF

Counter direction flag

0	TCNT counts down
1	TCNT counts up

Note: * Can only be written with 0 for flag clearing.

Up/down-counter

Note: * This timer counter can be used as an up/down-counter only in phase countin
when performing overflow/underflow counting on another channel. In other c
functions as an up-counter.

TGR4A—Timer General Register 4A **H'FE98**
TGR4B—Timer General Register 4B **H'FE9A**

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
Initial value	:	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Time prescaler

0	0	0	Internal clock: counts on φ/1
		1	Internal clock: counts on φ/4
	1	0	Internal clock: counts on φ/16
		1	Internal clock: counts on φ/64
1	0	0	External clock: counts on TCLKA
		1	External clock: counts on TCLKC
	1	0	Internal clock: counts on φ/256
		1	External clock: counts on TCLKD

Note: This setting is invalid when channel 5 counting mode.

Input clock edge select

0	0	Count at rising edge
	1	Count at falling edge
1	—	Count at both edges

Note: This setting is invalid when channel 5 is in phase counting mode, and also when φ/1 is selected as the input clock.

Counter clear

0	0	0	TCNT clearing disabled
		1	TCNT cleared by TGRA compare match/input ca
	1	0	TCNT cleared by TGRB compare match/input ca
		1	TCNT cleared by counter clearing for another ch performing synchronous clearing/synchronous op

Note: * Synchronous operation setting is performed by setti SYNC bit in TSYR to 1.

0	0	0	0	Normal opera
			1	Reserved
		1	0	PWM mode 1
			1	PWM mode 2
	1	0	0	Phase countir
			1	Phase countir
		1	0	Phase countir
			1	Phase countir
1	*	*	*	—

*

Note: MD3 is a reserved bit.
it should always be wri

0	0	0	0	TGR5A is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare ma
		1	0			1 output at compare ma
			1			Toggle output at compa
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare ma
		1	0			1 output at compare ma
			1			Toggle output at compa
1	*	0	0	TGR5A is input capture register	Capture input source is TIOCA5 pin	Input capture at rising e
			1			Input capture at falling e
		1	*			Input capture at both ed

TGR5B I/O control

0	0	0	0	TGR5B is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	*	0	0	TGR5B is input capture register	Capture input source is TIOCB5 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges

*: Don't care

TGI interrupt enab

0	Interrupt requ by TGFA bit d
1	Interrupt requ by TGFA bit e

TGI interrupt enable B

0	Interrupt requests (T by TGFB bit disabled
1	Interrupt requests (T by TGFB bit enabled

Overflow interrupt enable

0	Interrupt requests (TCIV) by TCF
1	Interrupt requests (TCIV) by TCF

Underflow interrupt enable

0	Interrupt requests (TCIU) by TCFU disabled
1	Interrupt requests (TCIU) by TCFU enabled

A/D conversion start request enable

0	A/D conversion start request generation disabled
1	A/D conversion start request generation enabled

TGR input capture/output compare flag

0	[Clearing conditions] • When DTC is activated by TGI and DISEL bit in DTC's MRB is • When 0 is written to TGFA afte TGFA = 1
1	[Setting conditions] • When TCNT = TGRA while TG functioning as output compare • When TCNT value is transferr TGRA by input capture signal is functioning as input capture

TGR input capture/output compare flag B

0	[Clearing conditions] • When DTC is activated by TGIB in and DISEL bit in DTC's MRB is 0 • When 0 is written to TGFB after re TGFB = 1
1	[Setting conditions] • When TCNT = TGRB while TGRB functioning as output compare reg • When TCNT value is transferred t TGRB by input capture signal whi is functioning as input capture reg

Overflow flag

0	[Clearing condition] When 0 is written to TCFV after reading TCFV = 1
1	[Setting condition] When the TCNT value overflows (changes from H'FFFF

Underflow flag

0	[Clearing condition] When 0 is written to TCFU after reading TCFU = 1
1	[Setting condition] When the TCNT value underflows (changes from H'0000 to H'FF

Counter direction flag

0	TCNT counts down
1	TCNT counts up

Note: * Can only be written with 0 for flag clearing.

Up/down-counter

Note: * This timer counter can be used as an up/down-counter only in phase countin
when performing overflow/underflow counting on another channel. In other c
functions as an up-counter.

TGR5A—Timer General Register 5A **H'FEA8**
TGR5B—Timer General Register 5B **H'FEAA**

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
Initial value	:	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TSTR—Timer Start Register **H'FEB0**

Bit	:	7	6	5	4	3	2	1
		—	—	CST5	CST4	CST3	CST2	CST1
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	—	—	R/W	R/W	R/W	R/W	R/W

Counter start

0	TCNTn count operation
1	TCNTn performs count

Note: If 0 is written to the CST bit during operation with the TIOC pin designated for
the counter stops but the TIOC pin output compare output level is retained. If T
written to when the CST bit is cleared to 0, the pin output level will be changed
the set initial output value.

Timer Synchronization

0	TCNTn operates independently (TCNT presetting/clearing is unrelated to other channel
1	TCNTn performs synchronous operation TCNT synchronous presetting/synchronous clearing is

(r

Notes: 1. To set synchronous operation, the SYNC bits for at least two channels must
2. To set synchronous clearing, in addition to the SYNC bit, the TCNT clearing must also be set by means of bits CCLR2 to CCLR0 in TCR.

Bit	:	7	6	5	4	3	2	1
		—	IPR6	IPR5	IPR4	—	IPR2	IPR1
Initial value	:	0	1	1	1	0	1	1
Read/Write	:	—	R/W	R/W	R/W	—	R/W	R/W

Interrupt Sources and IPR Settings

Register	Bits	
	6 to 4	2 to 0
IPRA	IRQ0	IRQ1
IPRB	IRQ2	IRQ4
	IRQ3	IRQ5
IPRC	—*1	DTC
IPRD	WDT0	—*1
IPRE	PC break	A/D converter, WDT1*2
IPRF	TPU channel 0	TPU channel 1
IPRG	TPU channel 2	TPU channel 3
IPRH	TPU channel 4	TPU channel 5
IPRI	—*1	—*1
IPRJ	—*1	SCI channel 0
IPRK	SCI channel 1	SCI channel 2
IPRM	HCAN	—*1

Notes: 1. These bits are reserved. They are always read as 1 and can
modified.
2. Valid only in the H8S/2626 Group.

Initial value :	0	0	0	0	0	0	0
Read/Write :	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Area 7 to 0 bus width control

0	Area n is designated for 16-bit
1	Area n is designated for 8-bit a

(

ASTCR—Access State Control Register **H'FED1** **Bus**

Bit :	7	6	5	4	3	2	1
	AST7	AST6	AST5	AST4	AST3	AST2	AST1
Initial value :	1	1	1	1	1	1	1
Read/Write :	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Area 7 to 0 access state control

0	Area n is designated for 2-state access Wait state insertion in area n external space access is
1	Area n is designated for 3-state access Wait state insertion in area n external space access is

(

Area 4 wait control

0	0	Program wait not ins
	1	1 program wait state
1	0	2 program wait state
	1	3 program wait state

Area 5 wait control

0	0	Program wait not inserte
	1	1 program wait state ins
1	0	2 program wait states in
	1	3 program wait states in

Area 6 wait control

0	0	Program wait not inserted
	1	1 program wait state inserted
1	0	2 program wait states inserted
	1	3 program wait states inserted

Area 7 wait control

0	0	Program wait not inserted
	1	1 program wait state inserted
1	0	2 program wait states inserted
	1	3 program wait states inserted

Area 0 wait control

0	0	Program wait not inse
	1	1 program wait state i
1	0	2 program wait states
	1	3 program wait states

Area 1 wait control

0	0	Program wait not inserted
	1	1 program wait state inser
1	0	2 program wait states inse
	1	3 program wait states inse

Area 2 wait control

0	0	Program wait not inserted
	1	1 program wait state inserted
1	0	2 program wait states inserted
	1	3 program wait states inserted

Area 3 wait control

0	0	Program wait not inserted
	1	1 program wait state inserted
1	0	2 program wait states inserted
	1	3 program wait states inserted

Burst cycle select 0

0	Max. 4 words in burst a
1	Max. 8 words in burst a

Burst cycle select 1

0	Burst cycle comprises 1 state
1	Burst cycle comprises 2 states

Area 0 burst ROM enable

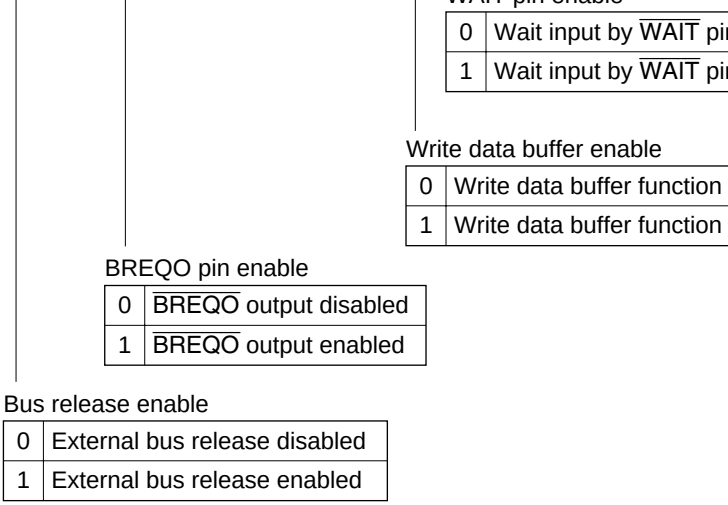
0	Basic bus interface
1	Burst ROM interface

Idle cycle insert 0

0	Idle cycle not inserted in case of successive external and external write cycles
1	Idle cycle inserted in case of successive external rea and external write cycles

Idle cycle insert 1

0	Idle cycle not inserted in case of successive external read cycle in different areas
1	Idle cycle inserted in case of successive external read cycles in different areas


WAIT pin enable
0 Wait input by WAIT pi
1 Wait input by WAIT pi
Write data buffer enable
0 Write data buffer function
1 Write data buffer function
BREQO pin enable
0 BREQO output disabled
1 BREQO output enabled
Bus release enable
0 External bus release disabled
1 External bus release enabled

RAM select

0	Emulation not selected Program/erase-protection of all flash memory blocks is disable
1	Emulation selected Program/erase-protection of all flash memory blocks is enabled

Flash memory area divisions

Addresses	Block Name	RAMS	RAM2	RA
H'FFD000–H'FFDFFF	RAM area 4 kbytes	0	*	*
H'000000–H'000FFF	EB0 (4 kbytes)	1	0	0
H'001000–H'001FFF	EB1 (4 kbytes)	1	0	0
H'002000–H'002FFF	EB2 (4 kbytes)	1	0	1
H'003000–H'003FFF	EB3 (4 kbytes)	1	0	1
H'004000–H'004FFF	EB4 (4 kbytes)	1	1	0
H'005000–H'005FFF	EB5 (4 kbytes)	1	1	0
H'006000–H'006FFF	EB6 (4 kbytes)	1	1	1
H'007000–H'007FFF	EB7 (4 kbytes)	1	1	1

*

Note: 1. This register is present only in the F-ZTAT version; it is not provided in the version.

Stores output data for port 1 pins (P17 to P10)

PADR—Port A Data Register **H'FF09**

Bit	:	7	6	5	4	3	2	1
		—	—	PA5DR*	PA4DR*	PA3DR	PA2DR	PA1DR
Initial value	:	Undefined	Undefined	0	0	0	0	0
Read/Write	:	—	—	R/W	R/W	R/W	R/W	R/W

Stores output data for port A pins (PA5 to P

Note: * Reserved bits in the H8S/2626 Group.

PBDR—Port B Data Register **H'FF0A**

Bit	:	7	6	5	4	3	2	1
		PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Stores output data for port B pins (PB7 to PB0)

Stores output data for port C pins (PC7 to PC0)

PDDR—Port D Data Register **H'FF0C**

Bit	:	7	6	5	4	3	2	1
		PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Stores output data for port D pins (PD7 to PD0)

PEDR—Port E Data Register **H'FF0D**

Bit	:	7	6	5	4	3	2	1
		PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Stores output data for port E pins (PE7 to PE0)

PFDR—Port F Data Register **H'FF0E**

Bit	:	7	6	5	4	3	2	1
		—	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1DR
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Stores output data for port F pins (PF6 to PF0)

0	0	0	Internal clock: counts on φ/1
		1	Internal clock: counts on φ/4
	1	0	Internal clock: counts on φ/16
		1	Internal clock: counts on φ/64
1	0	0	External clock: counts on TCLKA
		1	External clock: counts on TCLKB
	1	0	External clock: counts on TCLKC
		1	External clock: counts on TCLKD

Input clock edge select

0	0	Count at rising edge
	1	Count at falling edge
1	—	Count at both edges

Note: Internal clock edge selection is valid when the i
clock is φ/4 or slower. This setting is ignored if is
selected as the input clock.

Counter clear

0	0	0	TCNT clearing disabled
		1	TCNT cleared by TGRA compare match/input capture
	1	0	TCNT cleared by TGRB compare match/input capture
		1	TCNT cleared by counter clearing for another channel performing synchronous clearing/synchronous operation*1
1	0	0	TCNT clearing disabled
		1	TCNT cleared by TGRC compare match/input capture*2
	1	0	TCNT cleared by TGRD compare match/input capture*2
		1	TCNT cleared by counter clearing for another channel performing synchronous clearing/synchronous operation*1

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

0	0	0	0	Normal opera
			1	Reserved
		1	0	PWM mode 1
			1	PWM mode 2
	1	0	0	Phase countin
			1	Phase countin
		1	0	Phase countin
			1	Phase countin
1	*	*	*	—

*

Notes: 1. MD3 is a reserved
In a write, it should
written with 0.
2. Phase counting m
be set for channel
case, 0 should alw
written to MD2.

Buffer operation setting A

0	TGRA operates normally
1	TGRA and TGRC used together operation

Buffer operation setting B

0	TGRB operates normally
1	TGRB and TGRD used together for buffer operation

0	0	0	0	TGR0A is output compare register	Output disabled		
			1		Initial output is 0 output	0 output at compare ma	
		1	0			1 output at compare ma	
			1			Toggle output at compa	
	1	0	0		Output disabled		
			1		Initial output is 1 output	0 output at compare ma	
		1	0			1 output at compare ma	
			1			Toggle output at compa	
1	0	0	0	TGR0A is input capture register	Capture input source is TIOCA0 pin	Input capture at rising e	
			1			Input capture at falling e	
		1	*			Input capture at both ed	
	1	*	*		Capture input source is channel 1/count clock	Input capture at TCNT1 count-down	

TGR0B I/O control

0	0	0	0	TGR0B is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	0	0	0	TGR0B is input capture register	Capture input source is TIOCB0 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Capture input source is channel 1/count clock	Input capture at TCNT1 count-up/ count-down[*1]

*: Don't care

Note: 1. When bits TPSC2 to TPSC0 in TCR1 are set to B'000 and ø/1 is used as the TCNT1 count clock, this setting is invalid and input capture is not generated.

				TGR0C is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare m
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare m
1	0	0	0	TGR0C is input capture register	Capture input source is TIOCC0 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Capture input source is channel 1/count clock	Input capture at TCNT1 cou count-down

TGR0D I/O control

0	0	0	0	TGR0D is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	0	0	0	TGR0D is input capture register	Capture input source is TIOCD0 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Capture input source is channel 1/count clock	Input capture at TCNT1 count-up/ count-down[*1]

*: Don't care

Note: 1. When bits TPSC2 to TPSC0 in TCR1 are set to B'000 and ϕ/1 is used as the TCNT1 count clock, this setting is invalid and input capture is not generated.

Note: When TGR0C or TGR0D is designated for buffer operation, this setting is invalid and the register opera register.

TGR interrupt enabl

0	Interrupt reques by TGFA bit dis
1	Interrupt reques by TGFA bit en

TGR interrupt enable B

0	Interrupt requests by TGFB bit disabl
1	Interrupt requests by TGFB bit enabl

TGR interrupt enable C

0	Interrupt requests (TG by TGFC bit disabled
1	Interrupt requests (TG by TGFC bit enabled

TGR interrupt enable D

0	Interrupt requests (TGID) by TGFD bit disabled
1	Interrupt requests (TGID) by TGFD bit enabled

Overflow interrupt enable

0	Interrupt requests (TCIV) by TCFV d
1	Interrupt requests (TCIV) by TCFV e

A/D conversion start request enable

0	A/D conversion start request generation disabled
1	A/D conversion start request generation enabled

TGRA input capture/output compare flag

0	[Clearing conditions] • When DTC is activated by TGIA interru bit in DTC's MRB is 0 • When 0 is written to TGFA after readin
1	[Setting conditions] • When TCNT = TGRA while TGRA is fu output compare register • When TCNT value is transferred to TG capture signal while TGRA is functioni capture register

TGRB input capture/output compare flag

0	[Clearing conditions] • When DTC is activated by TGIB interrupt, and DI bit in DTC's MRB is 0 • When 0 is written to TGFB after reading TGFB =
1	[Setting conditions] • When TCNT = TGRB while TGRB is functioning a output compare register • When TCNT value is transferred to TGRB by inpu capture signal while TGRB is functioning as input capture register

TGRC input capture/output compare flag

0	[Clearing conditions] • When DTC is activated by TGIC interrupt, and DISEL bit in D • When 0 is written to TGFC after reading TGFC = 1
1	[Setting conditions] • When TCNT = TGRC while TGRC is functioning as output co • When TCNT value is transferred to TGRC by input capture si TGRC is functioning as input capture register

TGRD input capture/output compare flag

0	[Clearing conditions] • When DTC is activated by TGID interrupt, and DISEL bit in DTC's MR • When 0 is written to TGFD after reading TGFD = 1
1	[Setting conditions] • When TCNT = TGRD while TGRD is functioning as output compare re • When TCNT value is transferred to TGRD by input capture signal whil TGRD is functioning as input capture register

Overflow flag

0	[Clearing condition] When 0 is written to TCFV after reading TCFV = 1
1	[Setting condition] When the TCNT value overflows (changes from H'FFFF to H'0000)

Note: * Can only be written with 0 for flag clearing.

TGR0A—Timer General Register 0A	**H'FF18**
TGR0B—Timer General Register 0B	**H'FF1A**
TGR0C—Timer General Register 0C	**H'FF1C**
TGR0D—Timer General Register 0D	**H'FF1E**

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
Initial value	:	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Time prescaler

0	0	0	Internal clock: counts on ϕ/1
		1	Internal clock: counts on ϕ/4
	1	0	Internal clock: counts on ϕ/16
		1	Internal clock: counts on ϕ/64
1	0	0	External clock: counts on TCLK
		1	External clock: counts on TCLK
	1	0	Internal clock: counts on ϕ/256
		1	Counts on TCNT2 overflow/und

Note: This setting is invalid when channel 1 counting mode.

Input clock edge select

0	0	Count at rising edge
	1	Count at falling edge
1	—	Count at both edges

Note: This setting is invalid when channel 1 is in pha mode, and also when ϕ/1 or overflow/underflo channel is selected as the input clock.

Counter clear

0	0	TCNT clearing disabled
	1	TCNT cleared by TGRA compare match/input cap
1	0	TCNT cleared by TGRB compare match/input cap
	1	TCNT cleared by counter clearing for another cha performing synchronous clearing/synchronous op

Note: * Synchronous operation setting is performed by s SYNC bit in TSYR to 1.

0	0	0	0	Normal operati
			1	Reserved
		1	0	PWM mode 1
			1	PWM mode 2
	1	0	0	Phase counting
			1	Phase counting
		1	0	Phase counting
			1	Phase counting
1	*	*	*	—

*:

Note: MD3 is a reserved bit. In
it should always be writt

0	0	0	0	TGR1A is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare n
		1	0			1 output at compare n
			1			Toggle output at comp
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare n
		1	0			1 output at compare n
			1			Toggle output at comp
1	0	0	0	TGR1A is input capture register	Capture input source is TIOCA1 pin	Input capture at rising
			1			Input capture at falling
		1	*			Input capture at both e
	1	*	*		Capture input source is TGR0A compare match/ input capture	Input capture at gener channel 0/TGR0A cor input capture

TGR1B I/O control

0	0	0	0	TGR1B is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	0	0	0	TGR1B is input capture register	Capture input source is TIOCB1 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Capture input source is TGR0C compare match/ input capture	Input capture at generation of TGR0C compare match/input capture

*: Don't care

TGI interrupt enable

0	Interrupt reques by TGFA bit dis
1	Interrupt reques by TGFA bit en

TGI interrupt enable B

0	Interrupt requests (TG by TGFB bit disabled
1	Interrupt requests (TG by TGFB bit enabled

Overflow interrupt enable

0	Interrupt requests (TCIV) by TCFV
1	Interrupt requests (TCIV) by TCFV

Underflow interrupt enable

0	Interrupt requests (TCIU) by TCFU disabled
1	Interrupt requests (TCIU) by TCFU enabled

A/D conversion start request enable

0	A/D conversion start request generation disabled
1	A/D conversion start request generation enabled

TGR input capture/output compare f

0	[Clearing conditions] • When DTC is activated by TG and DISEL bit in DTC's MRB • When 0 is written to TGFA af TGFA = 1
1	[Setting conditions] • When TCNT = TGRA while T functioning as output compar • When TCNT value is transfer TGRA by input capture signal is functioning as input capture

TGR input capture/output compare flag

0	[Clearing conditions] • When DTC is activated by TGIB and DISEL bit in DTC's MRB is 0 • When 0 is written to TGFB after r TGFB = 1
1	[Setting conditions] • When TCNT = TGRB while TGR functioning as output compare re • When TCNT value is transferred TGRB by input capture signal wh is functioning as input capture re

Overflow flag

0	[Clearing condition] When 0 is written to TCFV after reading TCFV = 1
1	[Setting condition] When the TCNT value overflows (changes from H'FFF

Underflow flag

0	[Clearing condition] When 0 is written to TCFU after reading TCFU = 1
1	[Setting condition] When the TCNT value underflows (changes from H'0000 to H'F

Counter direction flag

0	TCNT counts down
1	TCNT counts up

Note: * Can only be written with 0 for flag clearing.

Up/down-counter

Note: * This timer counter can be used as an up/down-counter only in phase counting when performing overflow/underflow counting on another channel. In other ca functions as an up-counter.

TGR1A—Timer General Register 1A **H'FF28**
TGR1B—Timer General Register 1B **H'FF2A**

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
Initial value	:	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Time prescaler

0	0	0	Internal clock: counts on φ/1
		1	Internal clock: counts on φ/4
	1	0	Internal clock: counts on φ/16
		1	Internal clock: counts on φ/64
1	0	0	External clock: counts on TCLK
		1	External clock: counts on TCLK
	1	0	External clock: counts on TCLK
		1	Internal clock: counts on φ/1024

Note: This setting is invalid when channel 2 counting mode.

Input clock edge select

0	0	Count at rising edge
	1	Count at falling edge
1	—	Count at both edges

Note: This setting is invalid when channel 2 is in ph counting mode, and also when φ/1 is selecte the input clock.

Counter clear

0	0	TCNT clearing disabled
	1	TCNT cleared by TGRA compare match/input cap
1	0	TCNT cleared by TGRB compare match/input cap
	1	TCNT cleared by counter clearing for another cha performing synchronous clearing/synchronous op

Note: * Synchronous operation setting is performed by s SYNC bit in TSYR to 1.

0	0	0	0	Normal operati
			1	Reserved
		1	0	PWM mode 1
			1	PWM mode 2
	1	0	0	Phase counting
			1	Phase counting
		1	0	Phase counting
			1	Phase counting
1	*	*	*	—

*:

Note: MD3 is a reserved bit. In
it should always be writt

0	0	0	0	TGR2A is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare n
		1	0			1 output at compare n
			1			Toggle output at comp
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare n
		1	0			1 output at compare n
			1			Toggle output at comp
1	*	0	0	TGR2A is input capture register	Capture input source is TIOCA2 pin	Input capture at rising
			1			Input capture at falling
		1	*			Input capture at both e

TGR2B I/O control

0	0	0	0	TGR2B is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	*	0	0	TGR2B is input capture register	Capture input source is TIOCB2 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges

*: Don't care

TGI interrupt enable

0	Interrupt reques by TGFA bit dis
1	Interrupt reques by TGFA bit en

TGI interrupt enable B

0	Interrupt requests (TG by TGFB bit disabled
1	Interrupt requests (TG by TGFB bit enabled

Overflow interrupt enable

0	Interrupt requests (TCIV) by TCFV
1	Interrupt requests (TCIV) by TCFV

Underflow interrupt enable

0	Interrupt requests (TCIU) by TCFU disabled
1	Interrupt requests (TCIU) by TCFU enabled

A/D conversion start request enable

0	A/D conversion start request generation disabled
1	A/D conversion start request generation enabled

TGR input capture/output compare f

0	[Clearing conditions] • When DTC is activated by TG and DISEL bit in DTC's MRB • When 0 is written to TGFA af TGFA = 1
1	[Setting conditions] • When TCNT = TGRA while T functioning as output compar • When TCNT value is transfer TGRA by input capture signal is functioning as input capture

TGR input capture/output compare flag

0	[Clearing conditions] • When DTC is activated by TGIB and DISEL bit in DTC's MRB is 0 • When 0 is written to TGFB after r TGFB = 1
1	[Setting conditions] • When TCNT = TGRB while TGR functioning as output compare re • When TCNT value is transferred TGRB by input capture signal wh is functioning as input capture re

Overflow flag

0	[Clearing condition] When 0 is written to TCFV after reading TCFV = 1
1	[Setting condition] When the TCNT value overflows (changes from H'FFF

Underflow flag

0	[Clearing condition] When 0 is written to TCFU after reading TCFU = 1
1	[Setting condition] When the TCNT value underflows (changes from H'0000 to H'F

Counter direction flag

0	TCNT counts down
1	TCNT counts up

Note: * Can only be written with 0 for flag clearing.

Up/down-counter

Note: * This timer counter can be used as an up/down-counter only in phase counting when performing overflow/underflow counting on another channel. In other ca functions as an up-counter.

TGR2A—Timer General Register 2A **H'FF38**
TGR2B—Timer General Register 2B **H'FF3A**

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
Initial value	:	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CKS2	CKS2	CKS2	Clock	Overfl (when
0	0	0	φ/2 (Initial value)	25.6 μs
		1	φ/64	819.2 μ
	1	0	φ/128	1.6 ms
		1	φ/512	6.6 ms
1	0	0	φ/2048	26.2 ms
		1	φ/8192	104.9 m
	1	0	φ/32768	419.4 m
		1	φ/131072	1.68 s

Note: * The overflow period is the time from wh
starts counting up until overflow occurs.

Timer enable

0	TCNT is initialized to H'00 and halted
1	TCNT counts

Timer mode select

0	Interval timer mode: Sends the CPU an interval timer interrupt requ (WOVI) when TCNT overflows
1	Watchdog timer mode: Generates the $\overline{\text{WDTOVF}}$ signal when TCNT

Note: * For details of the case where TCNT overflows in watchdog tim
see section 12.2.3, Reset Control/Status Register (RSTCSR).

Overflow flag

0	[Clearing conditions] When 0 is written to OVF after reading TCSR when OVF = 1
1	[Setting condition] When TCNT overflows (changes from H'FF to H'00) When internal reset request generation is selected in watchdog timer mode, C cleared automatically by the internal reset

Notes: TCSR is write-protected by a password to prevent accidental overwriting. For details
section 12.2.5, Notes on register access.

* Can only be written with 0 for flag clearing.

section 12.2.5, Notes on Register Access.

RSTCSR—Reset Control/Status Register H'FF76 (W), H'FF77 (R)

Bit	:	7	6	5	4	3	2	1
		WOVF	RSTE	RSTS	—	—	—	—
Initial value	:	0	0	0	1	1	1	1
Read/Write	:	R/(W)*	R/W	R/W	—	—	—	—

Reset select

0	Power-on reset
1	Setting prohibited

Reset enable

0	Internal reset is not performed when TCNT overflows*
1	Internal reset is performed when TCNT overflows

Note: * The modules within the chip are not reset, but TCNT TCSR within the WDT are reset.

Watchdog overflow flag

0	[Clearing condition] When 0 is written to WOVF after reading TCSR when WOVF = 1
1	[Setting condition] When TCNT overflows (changes from H'FF to H'00) in watchdog time

Notes: RSTCSR is write-protected by a password to prevent accidental overwriting. Fo see section 12.2.5, Notes on Register Access.

* Can only be written with 0 for flag clearing.

0	0	φ
	1	φ
1	0	φ
	1	φ

Multiprocessor mode

0	Multiprocessor functio
1	Multiprocessor format

Stop bit length

0	1 stop bit
1	2 stop bits

Parity mode

0	Even parity*1
1	Odd parity*2

Notes: 1. When even parity is set, parity bit addition is
in transmission so that the total number of 1
transmit character plus the parity bit is even.
In reception, a check is performed to see if t
number of 1 bits in the receive character plu
bit is even.

2. When odd parity is set, parity bit addition is
in transmission so that the total number of 1
transmit character plus the parity bit is odd.
In reception, a check is performed to see if t
number of 1 bits in the receive character plu
bit is odd.

Parity enable

0	Parity bit addition and checking disabled
1	Parity bit addition and checking enabled*

Note: * When the PE bit is set to 1, the parity (even or odd) specified
O/$\overline{E}$ bit is added to transmit data before transmission. In recep
parity bit is checked for the parity (even or odd) specified by t

Character length

0	8-bit data
1	7-bit data*

Note: * When 7-bit data is selected, the MSB (bit 7) of TDR is not transmitted.

Asynchronous mode/synchronous mode select

0	Asynchronous mode
1	Synchronous mode

	1	c
1	0	c
	1	c

Basic clock pulse

0	0	32 clock periods
	1	64 clock periods
1	0	372 clock periods
	1	256 clock periods

Parity mode

0	Even parity*1
1	Odd parity*2

Notes: 1. When even parity is set, parity bit addition is perf
transmission so that the total number of 1 bits in
character plus the parity bit is even.
In reception, a check is performed to see if the to
of 1 bits in the receive character plus the parity b
2. When odd parity is set, parity bit addition is perfo
transmission so that the total number of 1 bits in
character plus the parity bit is odd.
In reception, a check is performed to see if the to
of 1 bits in the receive character plus the parity b

Parity enable

0	Parity bit addition and checking disabled
1	Parity bit addition and checking enabled

Note: When the smart card interface is used, be sure to make the 1

Block transfer mode

0	Normal smart card interface mode operation • Error signal transmission/detection and automatic data retransmission perform • TXI interrupt generated by TEND flag • TEND flag set 12.5 etu after start of transmission (11.0 etu in GSM mode)
1	Block transfer mode operation • Error signal transmission/detection and automatic data retransmission not perf • TXI interrupt generated by TDRE flag • TEND flag set 11.5 etu after start of transmission (11.0 etu in GSM mode)

GSM mode

0	Normal smart card interface mode operation • TEND flag generation 12.5 etu (11.5 etu in block transfer mode) after beginning of start bit • Clock output on/off control only
1	GSM mode smart card interface mode operation • TEND flag generation 11.0 etu after beginning of start bit • High/low fixing control possible in addition to clock output on/off control (set by SCR)

Note: etu: Elementary Time Unit (time for transfer of 1 bit)

Sets the serial transmit/receive bit rate

Note: For details see section 13.2.8, Bit Rate Register (BRR).

Clock enable

0	0	Asynchronous mode	Internal clock/SCK pin funct I/O port
		Synchronous mode	Internal clock/SCK pin funct serial clock output
	1	Asynchronous mode	Internal clock/SCK pin funct clock output*1
		Synchronous mode	Internal clock/SCK pin funct serial clock output
1	0	Asynchronous mode	External clock/SCK pin func clock input*2
		Synchronous mode	External clock/SCK pin func serial clock input
	1	Asynchronous mode	External clock/SCK pin func clock input*2
		Synchronous mode	External clock/SCK pin func serial clock input

Notes: 1. Outputs a clock of the same frequency as
2. Inputs a clock with a frequency 16 times t

Transmit end interrupt enable

0	Transmit end interrupt (TEI) request disabled
1	Transmit end interrupt (TEI) request enabled

Multiprocessor interrupt enable

0	Multiprocessor interrupts disabled [Clearing conditions] • When the MPIE bit is cleared to 0 • When data with MPB = 1 is received
1	Multiprocessor interrupts enabled Receive interrupt (RXI) requests, receive error interrupt (ER requests, and setting of the RDRF, FER, and ORER flags in disabled until data with the multiprocessor bit set to 1 is rec

Receive enable

0	Reception disabled
1	Reception enabled

Transmit enable

0	Transmission disabled
1	Transmission enabled

Receive interrupt enable

0	Receive data full interrupt (RXI) request and receive error interrupt (ERI) request disa
1	Receive data full interrupt (RXI) request and receive error interrupt (ERI) request ena

Transmit interrupt enable

0	Transmit data empty interrupt (TXI) request disabled
1	Transmit data empty interrupt (TXI) request enabled

Note: For details of how to clear interrupt requests, see section 13.2.6, Serial Control Register (SCR).

Clock enable

SCMR	SMR	SCR Setting		SCK Pin
SMIF	C/$\overline{A}$, GM	CKE1	CKE0	
0				See the SCI
1	0	0	0	Operates as
			1	Outputs clock output pin
	1		0	Operates as pin, with outp
			1	Outputs clock output pin
		1	0	Operates as pin, with outp
			1	Outputs clock output pin

Transmit end interrupt enable

0	Transmit end interrupt (TEI) request disabled
1	Transmit end interrupt (TEI) request enabled

Multiprocessor interrupt enable

0	Multiprocessor interrupts disabled [Clearing conditions] • When the MPIE bit is cleared to 0 • When data with MPB = 1 is received
1	Multiprocessor interrupts enabled Receive interrupt (RXI) requests, receive error int requests, and setting of the RDRF, FER, and OR SSR are disabled until data with the multiprocess 1 is received

Receive enable

0	Reception disabled
1	Reception enabled

Transmit enable

0	Transmission disabled
1	Transmission enabled

Receive interrupt enable

0	Receive data full interrupt (RXI) request and receive error interrupt (ERI) reques
1	Receive data full interrupt (RXI) request and receive error interrupt (ERI) reques

Transmit interrupt enable

0	Transmit data empty interrupt (TXI) request disabled
1	Transmit data empty interrupt (TXI) request enabled

Note: For details of how to clear interrupt requests, see section 13.2.6, Serial Control Register (SCR).

Stores data for serial transmission

1	Data with a 1 bit is transmit

Multiprocessor bit

0	[Clearing condition] When data with a 0 multiprocesso
1	[Setting condition] When data with a 1 multiprocesso

Transmit end

0	[Clearing conditions] • When 0 is written to TDRE after rea • When the DTC is activated by a TX writes data to TDR
1	[Setting conditions] • When the TE bit in SCR is 0 • When TDRE = 1 at transmission of a 1-byte serial transmit character

Parity error

0	[Clearing condition] When 0 is written to PER after reading PER
1	[Setting condition] When, in reception, the number of 1 bits in t plus the parity bit does not match the parity odd) specified by the O/$\overline{E}$ bit in SMR

Framing error

0	[Clearing condition] When 0 is written to FER after reading FER = 1
1	[Setting condition] When the SCI checks whether the stop bit at the end data is 1 when reception ends, and the stop bit is 0

Overrun error

0	[Clearing condition] When 0 is written to ORER after reading ORER = 1
1	[Setting condition] When the next serial reception is completed while RDRF = 1

Receive data register full

0	[Clearing conditions] • When 0 is written to RDRF after reading RDRF = 1 • When the DTC is activated by an RXI interrupt and reads data from RDR
1	[Setting condition] When serial reception ends normally and receive data is transferred from R

Transmit data register empty

0	[Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DTC is activated by a TXI interrupt and writes data to TDR
1	[Setting conditions] • When the TE bit in SCR is 0 • When data is transferred from TDR to TSR and data can be written to TDR

Notes: For details, see section 13.2.7, Serial Status Register (SSR).
* Can only be written with 0 for flag clearing.

Multiprocessor bit

0	[Clearing condition] When data with a 0 multiprocesso
1	[Setting condition] When data with a 1 multiprocesso

Transmit end

0	[Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DTC is activated by a TXI interrupt and writ
1	[Setting conditions] • Upon reset, and in standby mode or module stop mod • When the TE bit in SCR is 0 and the ERS bit is also 0 • When TDRE = 1 ERS = 0 (normal transmission) 2.5 e transmission of a 1-byte serial character when GM = • When TDRE = 1 ERS = 0 (normal transmission) 1.5 e transmission of a 1-byte serial character when GM = • When TDRE = 1 ERS = 0 (normal transmission) 1.0 e transmission of a 1-byte serial character when GM = • When TDRE = 1 ERS = 0 (normal transmission) 1.0 e transmission of a 1-byte serial character when GM =

Note: etu: Elementary Time Unit (time for transfer of 1 bit)

Parity error

0	[Clearing condition] When 0 is written to PER after reading PER = 1
1	[Setting condition] When, in reception, the number of 1 bits in the receive data parity bit does not match the parity setting (even or odd) sp the O/E bit in SMR

Error signal status

0	[Clearing conditions] • Upon reset, and in standby mode or module stop mode • When 0 is written to ERS after reading ERS = 1
1	[Setting condition] When the low level of the error signal is sampled

Note: Clearing the TE bit in SCR to 0 does not affect the ERS flag, which retains its previous state.

Overrun error

0	[Clearing condition] When 0 is written to ORER after reading ORER = 1
1	[Setting condition] When the next serial reception is completed while RDRF = 1

Receive data register full

0	[Clearing conditions] • When 0 is written to RDRF after reading RDRF = 1 • When the DTC is activated by an RXI interrupt and reads data from RDR
1	[Setting condition] When serial reception ends normally and receive data is transferred from RSR to RDR

Transmit data register empty

0	[Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DTC is activated by a TXI interrupt and writes data to TDR
1	[Setting conditions] • When the TE bit in SCR is 0 • When data is transferred from TDR to TSR and data can be written to TDR

Notes: For details, see section 14.2.2, Serial Status Register (SSR).
* Can only be written with 0 for flag clearing.

Stores received serial data

SCMR0—Smart Card Mode Register 0 **H'FF7E** **SCI0, Smart Ca**

Bit	:	7	6	5	4	3	2	1
		—	—	—	—	SDIR	SINV	—
Initial value	:	1	1	1	1	0	0	1
Read/Write	:	—	—	—	—	R/W	R/W	—

Smart card in
mode select

0	Smart ca function
1	Smart ca function

Smart card data invert

0	TDR contents are transmitted as they ar Receive data is stored as it is in RDR
1	TDR contents are inverted before being Receive data is stored in inverted form i

Smart card data transfer direction

0	TDR contents are transmitted LSB-first Receive data is stored in RDR LSB-first
1	TDR contents are transmitted MSB-first Receive data is stored in RDR MSB-first

0	0	φ/
	1	φ/4
1	0	φ/
	1	φ/

Multiprocessor mode

0	Multiprocessor function
1	Multiprocessor format s

Stop bit length

0	1 stop bit
1	2 stop bits

Parity mode

0	Even parity*1
1	Odd parity*2

Notes: 1. When even parity is set, parity bit addition is p
in transmission so that the total number of 1 b
transmit character plus the parity bit is even.
In reception, a check is performed to see if th
number of 1 bits in the receive character plus
bit is even.

2. When odd parity is set, parity bit addition is pe
in transmission so that the total number of 1 b
transmit character plus the parity bit is odd.
In reception, a check is performed to see if th
number of 1 bits in the receive character plus
bit is odd.

Parity enable

0	Parity bit addition and checking disabled
1	Parity bit addition and checking enabled*

Note: * When the PE bit is set to 1, the parity (even or odd) specified b
O/$\overline{E}$ bit is added to transmit data before transmission. In recept
parity bit is checked for the parity (even or odd) specified by th

Character length

0	8-bit data
1	7-bit data*

Note: * When 7-bit data is selected, the MSB (bit 7) of TDR is not transmitted.

Asynchronous mode/synchronous mode select

0	Asynchronous mode
1	Synchronous mode

	1	
1	0	
	1	

Basic clock pulse

0	0	32 clock periods
	1	64 clock periods
1	0	372 clock periods
	1	256 clock periods

Parity mode

0	Even parity*1
1	Odd parity*2

Notes: 1. When even parity is set, parity bit addition is pe
transmission so that the total number of 1 bits i
character plus the parity bit is even.
In reception, a check is performed to see if the
of 1 bits in the receive character plus the parity
2. When odd parity is set, parity bit addition is per
transmission so that the total number of 1 bits i
character plus the parity bit is odd.
In reception, a check is performed to see if the
of 1 bits in the receive character plus the parity

Parity enable

0	Parity bit addition and checking disabled
1	Parity bit addition and checking enabled

Note: When the smart card interface is used, be sure to make the

Block transfer mode

0	Normal smart card interface mode operation • Error signal transmission/detection and automatic data retransmission perfor • TXI interrupt generated by TEND flag • TEND flag set 12.5 etu after start of transmission (11.0 etu in GSM mode)
1	Block transfer mode operation • Error signal transmission/detection and automatic data retransmission not pe • TXI interrupt generated by TDRE flag • TEND flag set 11.5 etu after start of transmission (11.0 etu in GSM mode)

GSM mode

0	Normal smart card interface mode operation • TEND flag generation 12.5 etu (11.5 etu in block transfer mode) after beginning of start bit • Clock output on/off control only
1	GSM mode smart card interface mode operation • TEND flag generation 11.0 etu after beginning of start bit • High/low fixing control possible in addition to clock output on/off control (set by SCR)

Note: etu: Elementary Time Unit (time for transfer of 1 bit)

Sets the serial transmit/receive bit rate

Note: For details see section 13.2.8, Bit Rate Register (BRR).

0	0	Asynchronous mode	Internal clock/SCK pin fu I/O port
		Synchronous mode	Internal clock/SCK pin fu serial clock output
	1	Asynchronous mode	Internal clock/SCK pin fu clock output*1
		Synchronous mode	Internal clock/SCK pin fu serial clock output
1	0	Asynchronous mode	External clock/SCK pin fu clock input*2
		Synchronous mode	External clock/SCK pin fu serial clock input
	1	Asynchronous mode	External clock/SCK pin fu clock input*2
		Synchronous mode	External clock/SCK pin fu serial clock input

Notes: 1. Outputs a clock of the same frequency
2. Inputs a clock with a frequency 16 time

Transmit end interrupt enable

0	Transmit end interrupt (TEI) request disabled
1	Transmit end interrupt (TEI) request enabled

Multiprocessor interrupt enable

0	Multiprocessor interrupts disabled [Clearing conditions] • When the MPIE bit is cleared to 0 • When data with MPB = 1 is received
1	Multiprocessor interrupts enabled Receive interrupt (RXI) requests, receive error interrupt (requests, and setting of the RDRF, FER, and ORER flag disabled until data with the multiprocessor bit set to 1 is r

Receive enable

0	Reception disabled
1	Reception enabled

Transmit enable

0	Transmission disabled
1	Transmission enabled

Receive interrupt enable

0	Receive data full interrupt (RXI) request and receive error interrupt (ERI) request d
1	Receive data full interrupt (RXI) request and receive error interrupt (ERI) request e

Transmit interrupt enable

0	Transmit data empty interrupt (TXI) request disabled
1	Transmit data empty interrupt (TXI) request enabled

Note: For details of how to clear interrupt requests, see section 13.2.6, Serial Control Register (SCR).

SCMR	SMR	SCR Setting		SCK Pin
SMIF	C/$\overline{A}$, GM	CKE1	CKE0	
0				See the SCI
1	0	0	0	Operates as
			1	Outputs cloc output pin
	1		0	Operates as pin, with outp
			1	Outputs cloc output pin
		1	0	Operates as pin, with outp
			1	Outputs cloc output pin

Transmit end interrupt enable

0	Transmit end interrupt (TEI) request disabled
1	Transmit end interrupt (TEI) request enabled

Multiprocessor interrupt enable

0	Multiprocessor interrupts disabled [Clearing conditions] • When the MPIE bit is cleared to 0 • When data with MPB = 1 is received
1	Multiprocessor interrupts enabled Receive interrupt (RXI) requests, receive error in requests, and setting of the RDRF, FER, and OR SSR are disabled until data with the multiprocess to 1 is received

Receive enable

0	Reception disabled
1	Reception enabled

Transmit enable

0	Transmission disabled
1	Transmission enabled

Receive interrupt enable

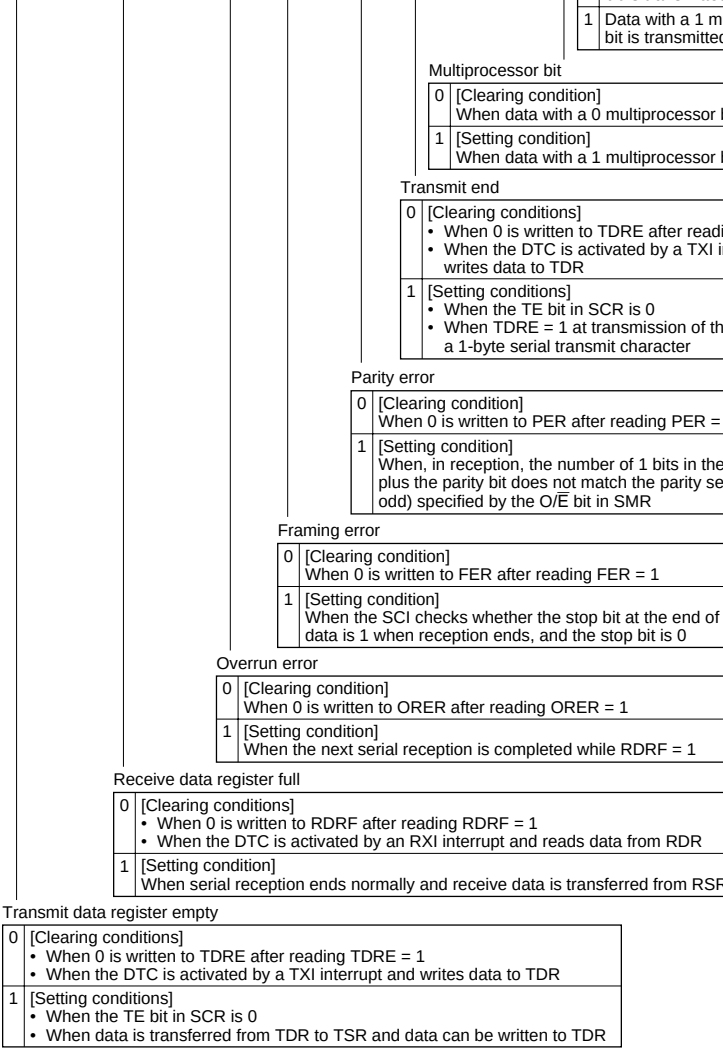
0	Receive data full interrupt (RXI) request and receive error interrupt (ERI) reques
1	Receive data full interrupt (RXI) request and receive error interrupt (ERI) reques

Transmit interrupt enable

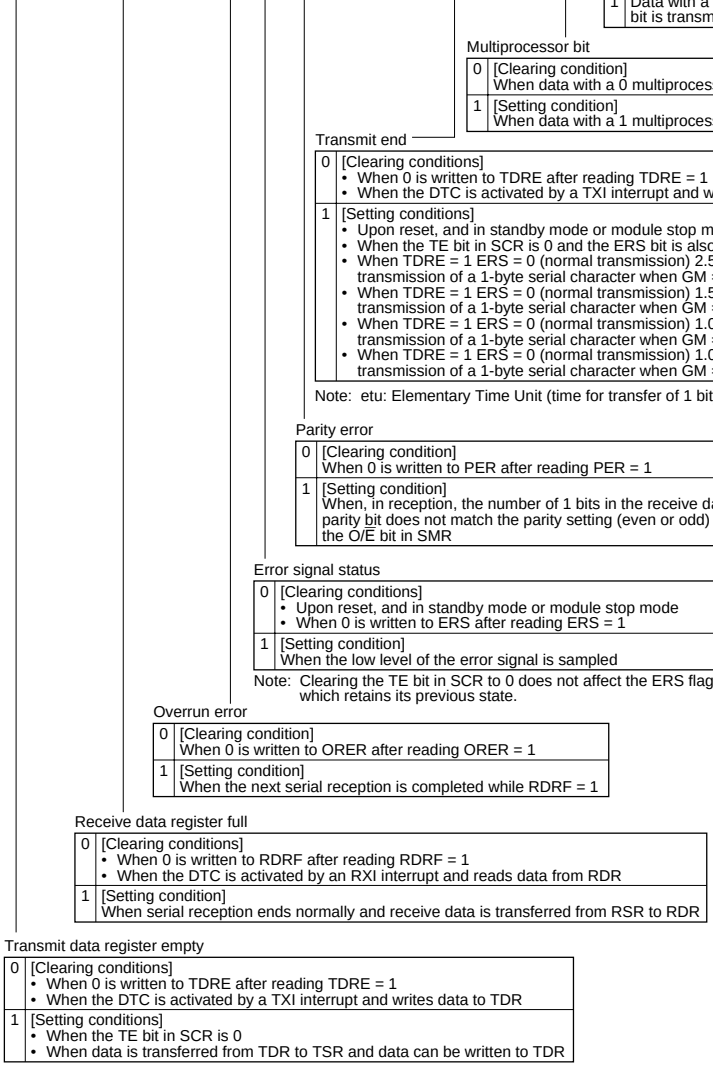
0	Transmit data empty interrupt (TXI) request disabled
1	Transmit data empty interrupt (TXI) request enabled

Note: For details of how to clear interrupt requests, see section 13.2.6, Serial Control Register (SCR).

Stores data for serial transmission



Notes: For details, see section 13.2.7, Serial Status Register (SSR).
* Can only be written with 0 for flag clearing.



Notes: For details, see section 14.2.2, Serial Status Register (SSR).
* Can only be written with 0 for flag clearing.

Stores received serial data

SCMR1—Smart Card Mode Register 1 **H'FF86** **SCI, Smart Car**

Bit	:	7	6	5	4	3	2	1
		—	—	—	—	SDIR	SINV	—
Initial value	:	1	1	1	1	0	0	1
Read/Write	:	—	—	—	—	R/W	R/W	—

Smart card int
mode select

0	Smart car function is
1	Smart car function is

Smart card data invert

0	TDR contents are transmitted as they are Receive data is stored as it is in RDR
1	TDR contents are inverted before being t Receive data is stored in inverted form in

Smart card data transfer direction

0	TDR contents are transmitted LSB-first Receive data is stored in RDR LSB-first
1	TDR contents are transmitted MSB-first Receive data is stored in RDR MSB-first

0	0	φ
	1	φ
1	0	φ
	1	φ

Multiprocessor mode

0	Multiprocessor functio
1	Multiprocessor format

Stop bit length

0	1 stop bit
1	2 stop bits

Parity mode

0	Even parity*1
1	Odd parity*2

Notes: 1. When even parity is set, parity bit addition is
in transmission so that the total number of 1
transmit character plus the parity bit is even.
In reception, a check is performed to see if t
number of 1 bits in the receive character plu
bit is even.

2. When odd parity is set, parity bit addition is
in transmission so that the total number of 1
transmit character plus the parity bit is odd.
In reception, a check is performed to see if t
number of 1 bits in the receive character plu
bit is odd.

Parity enable

0	Parity bit addition and checking disabled
1	Parity bit addition and checking enabled*

Note: * When the PE bit is set to 1, the parity (even or odd) specified
O/$\overline{E}$ bit is added to transmit data before transmission. In recep
parity bit is checked for the parity (even or odd) specified by t

Character length

0	8-bit data
1	7-bit data*

Note: * When 7-bit data is selected, the MSB (bit 7) of TDR is not transmitted.

Asynchronous mode/synchronous mode select

0	Asynchronous mode
1	Synchronous mode

	1	
1	0	
	1	

Basic clock pulse

0	0	32 clock periods
	1	64 clock periods
1	0	372 clock periods
	1	256 clock periods

Parity mode

0	Even parity*1
1	Odd parity*2

Notes: 1. When even parity is set, parity bit addition is per
transmission so that the total number of 1 bits in
character plus the parity bit is even.
In reception, a check is performed to see if the to
of 1 bits in the receive character plus the parity b
2. When odd parity is set, parity bit addition is perfo
transmission so that the total number of 1 bits in
character plus the parity bit is odd.
In reception, a check is performed to see if the to
of 1 bits in the receive character plus the parity b

Parity enable

0	Parity bit addition and checking disabled
1	Parity bit addition and checking enabled

Note: When the smart card interface is used, be sure to make the 1

Block transfer mode

0	Normal smart card interface mode operation • Error signal transmission/detection and automatic data retransmission perform • TXI interrupt generated by TEND flag • TEND flag set 12.5 etu after start of transmission (11.0 etu in GSM mode)
1	Block transfer mode operation • Error signal transmission/detection and automatic data retransmission not per • TXI interrupt generated by TDRE flag • TEND flag set 11.5 etu after start of transmission (11.0 etu in GSM mode)

GSM mode

0	Normal smart card interface mode operation • TEND flag generation 12.5 etu (11.5 etu in block transfer mode) after beginning of start bit • Clock output on/off control only
1	GSM mode smart card interface mode operation • TEND flag generation 11.0 etu after beginning of start bit • High/low fixing control possible in addition to clock output on/off control (set by SCR)

Note: etu: Elementary Time Unit (time for transfer of 1 bit)

Sets the serial transmit/receive bit rate

Note: For details see section 13.2.8, Bit Rate Register (BRR).

0	0	Asynchronous mode	Internal clock/SCK pin func I/O port
		Synchronous mode	Internal clock/SCK pin func serial clock output
	1	Asynchronous mode	Internal clock/SCK pin func clock output*1
		Synchronous mode	Internal clock/SCK pin func serial clock output
1	0	Asynchronous mode	External clock/SCK pin fun clock input*2
		Synchronous mode	External clock/SCK pin fun serial clock input
	1	Asynchronous mode	External clock/SCK pin fun clock input*2
		Synchronous mode	External clock/SCK pin fun serial clock input

Notes: 1. Outputs a clock of the same frequency a
2. Inputs a clock with a frequency 16 times

Transmit end interrupt enable

0	Transmit end interrupt (TEI) request disabled
1	Transmit end interrupt (TEI) request enabled

Multiprocessor interrupt enable

0	Multiprocessor interrupts disabled [Clearing conditions] • When the MPIE bit is cleared to 0 • When data with MPB = 1 is received
1	Multiprocessor interrupts enabled Receive interrupt (RXI) requests, receive error interrupt (EI requests, and setting of the RDRF, FER, and ORER flags i disabled until data with the multiprocessor bit set to 1 is rec

Receive enable

0	Reception disabled
1	Reception enabled

Transmit enable

0	Transmission disabled
1	Transmission enabled

Receive interrupt enable

0	Receive data full interrupt (RXI) request and receive error interrupt (ERI) request dis
1	Receive data full interrupt (RXI) request and receive error interrupt (ERI) request en

Transmit interrupt enable

0	Transmit data empty interrupt (TXI) request disabled
1	Transmit data empty interrupt (TXI) request enabled

Note: For details of how to clear interrupt requests, see section 13.2.6, Serial Control Register (SCR).

SMIF	C/$\overline{A}$, GM	CKE1	CKE0	SCK Pin
0				See the SC
1	0	0	0	Operates as
			1	Outputs clo output pin
	1		0	Operates as pin, with ou
			1	Outputs clo output pin
		1	0	Operates as pin, with ou
			1	Outputs clo output pin

Transmit end interrupt enable

0	Transmit end interrupt (TEI) request disable
1	Transmit end interrupt (TEI) request enable

Multiprocessor interrupt enable

0	Multiprocessor interrupts disabled [Clearing conditions] • When the MPIE bit is cleared to 0 • When data with MPB = 1 is received
1	Multiprocessor interrupts enabled Receive interrupt (RXI) requests, receive error i requests, and setting of the RDRF, FER, and O SSR are disabled until data with the multiproces to 1 is received

Receive enable

0	Reception disabled
1	Reception enabled

Transmit enable

0	Transmission disabled
1	Transmission enabled

Receive interrupt enable

0	Receive data full interrupt (RXI) request and receive error interrupt (ERI) reque
1	Receive data full interrupt (RXI) request and receive error interrupt (ERI) reque

Transmit interrupt enable

0	Transmit data empty interrupt (TXI) request disabled
1	Transmit data empty interrupt (TXI) request enabled

Note: For details of how to clear interrupt requests, see section 13.2.6, Serial Control Register (SCR).

Stores data for serial transmission

1	Data with a 1 bit is transmit

Multiprocessor bit

0	[Clearing condition] When data with a 0 multiprocesso
1	[Setting condition] When data with a 1 multiprocesso

Transmit end

0	[Clearing conditions] • When 0 is written to TDRE after rea • When the DTC is activated by a TX writes data to TDR
1	[Setting conditions] • When the TE bit in SCR is 0 • When TDRE = 1 at transmission of a 1-byte serial transmit character

Parity error

0	[Clearing condition] When 0 is written to PER after reading PER
1	[Setting condition] When, in reception, the number of 1 bits in t data plus the parity bit does not match the p (even or odd) specified by the O/$\overline{E}$ bit in SM

Framing error

0	[Clearing condition] When 0 is written to FER after reading FER = 1
1	[Setting condition] When the SCI checks whether the stop bit at the end data is 1 when reception ends, and the stop bit is 0

Overrun error

0	[Clearing condition] When 0 is written to ORER after reading ORER = 1
1	[Setting condition] When the next serial reception is completed while RDRF = 1

Receive data register full

0	[Clearing conditions] • When 0 is written to RDRF after reading RDRF = 1 • When the DTC is activated by an RXI interrupt and reads data from RDR
1	[Setting condition] When serial reception ends normally and receive data is transferred from R

Transmit data register empty

0	[Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DTC is activated by a TXI interrupt and writes data to TDR
1	[Setting conditions] • When the TE bit in SCR is 0 • When data is transferred from TDR to TSR and data can be written to TDR

Notes: For details, see section 13.2.7, Serial Status Register (SSR).
* Can only be written with 0 for flag clearing.

1	Data with a 1 bit is transmitt

Multiprocessor bit

0	[Clearing condition] When data with a 0 multiprocesso
1	[Setting condition] When data with a 1 multiprocesso

Transmit end

0	[Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DTC is activated by a TXI interrupt and writ
1	[Setting conditions] • Upon reset, and in standby mode or module stop mod • When the TE bit in SCR is 0 and the ERS bit is also 0 • When TDRE = 1 ERS = 0 (normal transmission) 2.5 e transmission of a 1-byte serial character when GM = 0 • When TDRE = 1 ERS = 0 (normal transmission) 1.5 e transmission of a 1-byte serial character when GM = 0 • When TDRE = 1 ERS = 0 (normal transmission) 1.0 e transmission of a 1-byte serial character when GM = 1 • When TDRE = 1 ERS = 0 (normal transmission) 1.0 e transmission of a 1-byte serial character when GM = 1

Note: etu: Elementary Time Unit (time for transfer of 1 bit)

Parity error

0	[Clearing condition] When 0 is written to PER after reading PER = 1
1	[Setting condition] When, in reception, the number of 1 bits in the receive data parity bit does not match the parity setting (even or odd) sp the O/E bit in SMR

Error signal status

0	[Clearing conditions] • Upon reset, and in standby mode or module stop mode • When 0 is written to ERS after reading ERS = 1
1	[Setting condition] When the low level of the error signal is sampled

Note: Clearing the TE bit in SCR to 0 does not affect the ERS flag, which retains its previous state.

Overrun error

0	[Clearing condition] When 0 is written to ORER after reading ORER = 1
1	[Setting condition] When the next serial reception is completed while RDRF = 1

Receive data register full

0	[Clearing conditions] • When 0 is written to RDRF after reading RDRF = 1 • When the DTC is activated by an RXI interrupt and reads data from RDR
1	[Setting condition] When serial reception ends normally and receive data is transferred from RSR to RDR

Transmit data register empty

0	[Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DTC is activated by a TXI interrupt and writes data to TDR
1	[Setting conditions] • When the TE bit in SCR is 0 • When data is transferred from TDR to TSR and data can be written to TDR

Notes: For details, see section 14.2.2, Serial Status Register (SSR).
* Can only be written with 0 for flag clearing.

Stores received serial data

SCMR2—Smart Card Mode Register 2 **H'FF8E** **SCI, Smart Ca**

Bit	:	7	6	5	4	3	2	1
		—	—	—	—	SDIR	SINV	—
Initial value	:	1	1	1	1	0	0	1
Read/Write	:	—	—	—	—	R/W	R/W	—

Smart card in
mode select

0	Smart ca function
1	Smart ca function

Smart card data invert

0	TDR contents are transmitted as they ar Receive data is stored as it is in RDR
1	TDR contents are inverted before being Receive data is stored in inverted form i

Smart card data transfer direction

0	TDR contents are transmitted LSB-first Receive data is stored in RDR LSB-first
1	TDR contents are transmitted MSB-first Receive data is stored in RDR MSB-first

Read/Write :	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Analog input channels and corresponding ADDR registers

Analog Input Channel				A/D Data Register
Channel Set 0 (CH3 = 0)		Channel Set 1 (CH3 = 1)		
Group 0	Group 1	Group 0	Group 1	
AN0	AN4	AN8	AN12	ADDRA
AN1	AN5	AN9	AN13	ADDRB
AN2	AN6	AN10	AN14	ADDRC
AN3	AN7	AN11	AN15	ADDRD

Note: The upper byte can be read directly, but for the lower byte, data transfer is performed via a temporary register (TEMP). For details see section 16.3, Interface to Bus Master.

CH3	CH2	CH1	CH0	Single Mode (SCAN = 0)	Sc (S
0	0	0	0	AN0 (Initial value)	AN0
			1	AN1	AN0,
		1	0	AN2	AN0
			1	AN3	AN0
	1	0	0	AN4	AN4
			1	AN5	AN4,
		1	0	AN6	AN4
			1	AN7	AN4
1	0	0	0	AN8	AN8
			1	AN9	AN8,
		1	0	AN10	AN8
			1	AN11	AN8
	1	0	0	AN12	AN1
			1	AN13	AN1
		1	0	AN14	AN1
			1	AN15	AN1

Channel select

0	AN8 to AN11 are group 0 analog input pins, AN are group 1 analog input pins
1	AN0 to AN3 are group 0 analog input pins, AN4 are group 1 analog input pins

Scan mode

0	Single mode
1	Scan mode

A/D start

0	A/D conversion stopped
1	• Single mode: A/D conversion is started. Cleared to 0 automati conversion on the specified channel ends • Scan mode: A/D conversion is started. Conversion continues c on the selected channels until ADST is cleared to 0 by softwar or a transition to standby mode or module stop mode

A/D interrupt enable

0	A/D conversion end interrupt (ADI) request disabled
1	A/D conversion end interrupt (ADI) request enabled

A/D end flag

0	[Clearing conditions] • When 0 is written to ADF after reading ADF = 1 • When the DTC is activated by an ADI interrupt and ADDR is read
1	[Setting conditions] • Single mode: When A/D conversion ends • Scan mode: When A/D conversion ends on all specified channels

Note: * Can only be written with 0 for flag clearing.

Clock select

0	0	Conversion time = 530 states (
	1	Conversion time = 266 states (
1	0	Conversion time = 134 states (
	1	Conversion time = 68 states (n

Timer trigger select

0	0	A/D conversion start by software is enabled
	1	A/D conversion start by TPU conversion start trigger is enab
1	0	Setting prohibited
	1	A/D conversion start by external trigger pin ($\overline{\text{ADTRG}}$) is ena

					(when ϕSU
0	0	0	0	ϕ/2	25.
			1	ϕ/64	819
		1	0	ϕ/128	1.6
			1	ϕ/512	6.6
	1	0	0	ϕ/2048	26.
			1	ϕ/8192	104
		1	0	ϕ/32768	419
			1	ϕ/131072	1.6
1	0	0	0	ϕSUB/2	15.
			1	ϕSUB/4	31.
		1	0	ϕSUB/8	62.
			1	ϕSUB/16	125
	1	0	0	ϕSUB/32	250
			1	ϕSUB/64	500
		1	0	ϕSUB/128	1 s
			1	ϕSUB/256	2 s

Note: * The overflow cycle starts when TCNT starts c
H'00 and ends when an overflow occurs.

Reset or NMI

0	NMI interrupt request
1	Internal reset request

Prescaler select

0	TCNT counts the divided clock output by the ø-based prescaler
1	TCNT counts the divided clock output by the øSUB-based prescaler

Timer enable

0	Initializes TCNT to H'00 and disables the counting operation
1	TCNT performs counting operation

Timer mode select

0	Interval timer mode: Interval timer interrupt (WOVI) request sent to CPU when overflow occurs at TCNT
1	Watchdog timer mode: Reset or NMI interrupt request sent to CPU when overflow occurs at TCNT

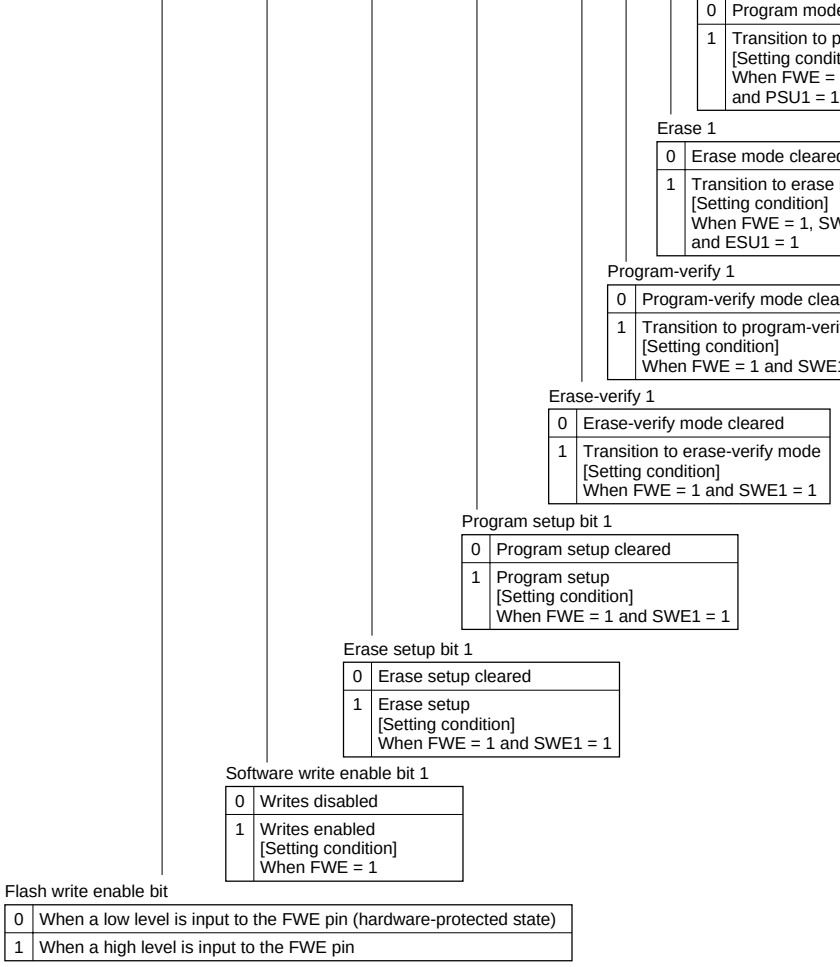
Overflow flag

0	[Clearing] (1) When 0 is written to TME bit; (2) When 0 is written to OVF bit after reading TCSR when OVF=1.
1	[Setting] When TCNT overflows (H'FF→ H'00). When internal reset request generation is selected in watchdog timer mode, OVF is cleared automatically by the internal reset.

Notes: TCSR is write-protected by a password to prevent accidental overwriting. For details see section 12.2.5, Notes on Register
1. Only 0 can be written to these bits (to clear these flags).
2. This register is not available, and must not be accessed, in the H8S/2623 Group.

For details see section 12.2.5, Notes on Register Access.

* This register is not available, and must not be accessed, in the H8S/2623 Gr



Program mode	
0	Program mode
1	Transition to p [Setting condit When FWE = and PSU1 = 1

Erase 1

Erase mode cleared	
0	Erase mode cleared
1	Transition to erase [Setting condition] When FWE = 1, SW and ESU1 = 1

Program-verify 1

Program-verify mode clea	
0	Program-verify mode clea
1	Transition to program-veri [Setting condition] When FWE = 1 and SWE1

Erase-verify 1

Erase-verify mode cleared	
0	Erase-verify mode cleared
1	Transition to erase-verify mode [Setting condition] When FWE = 1 and SWE1 = 1

Program setup bit 1

Program setup cleared	
0	Program setup cleared
1	Program setup [Setting condition] When FWE = 1 and SWE1 = 1

Erase setup bit 1

Erase setup cleared	
0	Erase setup cleared
1	Erase setup [Setting condition] When FWE = 1 and SWE1 = 1

Software write enable bit 1

Writes disabled	
0	Writes disabled
1	Writes enabled [Setting condition] When FWE = 1

Flash write enable bit

0	When a low level is input to the FWE pin (hardware-protected state)
1	When a high level is input to the FWE pin

Notes: 1. This register is not present in the mask ROM version, and an attempt to read it will return an undefi
2. To access this register, set the FLSHE bit to 1 in serial control register X (SCRX). Even if FLSHE = is in a mode in which the on-chip flash memory is disabled, a read will return H'00 and writes are in to this register are also invalid when the FWE bit in FLMCR1 is not set to 1.

Note: * Determined by the state of the FWE pin.

Flash memory error

0	Flash memory is operating normally Flash memory program/erase protection (error protection) is disable [Clearing condition] Reset or hardware standby mode
1	An error has occurred during flash memory programming/erasing Flash memory program/erase protection (error protection) is enable [Setting condition] See section 19.8.3, Error Protection

Notes: 1. This register is not present in the mask ROM version, and ar attempt to read it will return an undefined value.
2. To access this register, set the FLSHE bit to 1 in serial contr register X (SCRX). Even if FLSHE = 1, if the chip is in a mod in which the on-chip flash memory is disabled, a read will ret H'00 and writes are invalid. Writes to this register are also invalid when the FWE bit in FLMCR1 is not set to 1.

EBR1—Erase Block Register 1 **H'FFAA**

Bit :	7	6	5	4	3	2	1
	EB7	EB6	EB5	EB4	EB3	EB2	EB1
Initial value :	0	0	0	0	0	0	0
Read/Write :	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Sets flash memory erase area block by block

Flash memory erase blocks

Block (Size)	Addresses
EB0 (4 kbytes)	H'000000–H'000FFF
EB1 (4 kbytes)	H'001000–H'001FFF
EB2 (4 kbytes)	H'002000–H'002FFF
EB3 (4 kbytes)	H'003000–H'003FFF
EB4 (4 kbytes)	H'004000–H'004FFF
EB5 (4 kbytes)	H'005000–H'005FFF
EB6 (4 kbytes)	H'006000–H'006FFF
EB7 (4 kbytes)	H'007000–H'007FFF
EB8 (32 kbytes)	H'008000–H'00FFFF
EB9 (64 kbytes)	H'010000–H'01FFFF
EB10 (64 kbytes)	H'020000–H'02FFFF
EB11 (64 kbytes)	H'030000–H'03FFFF

Notes: 1. This register is not present in the mask ROM version, and a attempt to read it will return an undefined value.
2. To access this register, set the FLSHE bit to 1 in serial cont register X (SCRX). Even if FLSHE = 1, if the chip is in a mo in which the on-chip flash memory is disabled, a read will re H'00 and writes are invalid. Writes to this register are also invalid when the FWE bit in FLMCR1 is not set to 1.

0	Transition to flash memory power-down mode enabled
1	Transition to flash memory power-down mode disabled

PORT1—Port 1 Register **H'FFB0**

Bit	:	7	6	5	4	3	2	1
		P17	P16	P15	P14	P13	P12	P11
Initial value	:	—*	—*	—*	—*	—*	—*	—*
Read/Write	:	R	R	R	R	R	R	R

State of port 1 pins

Note: * Determined by the state of pins P17 to P10.

PORT4—Port 4 Register **H'FFB3**

Bit	:	7	6	5	4	3	2	1
		P47	P46	P45	P44	P43	P42	P41
Initial value	:	—*	—*	—*	—*	—*	—*	—*
Read/Write	:	R	R	R	R	R	R	R

State of port 4 pins

Note: * Determined by the state of pins P47 to P40.

State of port 9 pins

Note: * Determined by the state of pins P97 to P90.

PORTA—Port A Register **H'FFB9**

Bit	:	7	6	5	4	3	2	1
		—	—	PA5*2	PA4*2	PA3	PA2	PA1
Initial value	:	Undefined	Undefined	—*1	—*1	—*1	—*1	—*1
Read/Write	:	—	—	R	R	R	R	R

State of port A pins

Notes: 1. Determined by the state of pins PA5 to PA0.
2. Reserved bits in the H8S/2626 Group.

PORTB—Port B Register **H'FFBA**

Bit	:	7	6	5	4	3	2	1
		PB7	PB6	PB5	PB4	PB3	PB2	PB1
Initial value	:	—*	—*	—*	—*	—*	—*	—*
Read/Write	:	R	R	R	R	R	R	R

State of port B pins

Note: * Determined by the state of pins PB7 to PB0.

State of port C pins

Note: * Determined by the state of pins PC7 to PC0.

PORTD—Port D Register **H'FFBC**

Bit	:	7	6	5	4	3	2	1
		PD7	PD6	PD5	PD4	PD3	PD2	PD1
Initial value	:	—*	—*	—*	—*	—*	—*	—*
Read/Write	:	R	R	R	R	R	R	R

State of port D pins

Note: * Determined by the state of pins PD7 to PD0.

PORTE—Port E Register **H'FFBD**

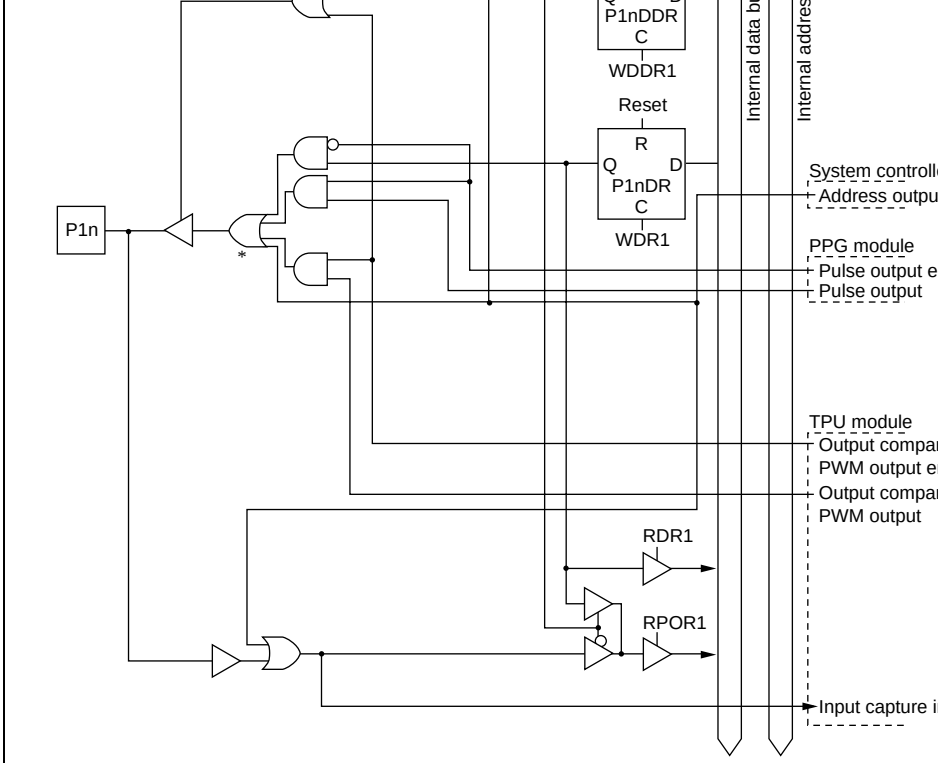
Bit	:	7	6	5	4	3	2	1
		PE7	PE6	PE5	PE4	PE3	PE2	PE1
Initial value	:	—*	—*	—*	—*	—*	—*	—*
Read/Write	:	R	R	R	R	R	R	R

State of port E pins

Note: * Determined by the state of pins PE7 to PE0.

State of port F pins

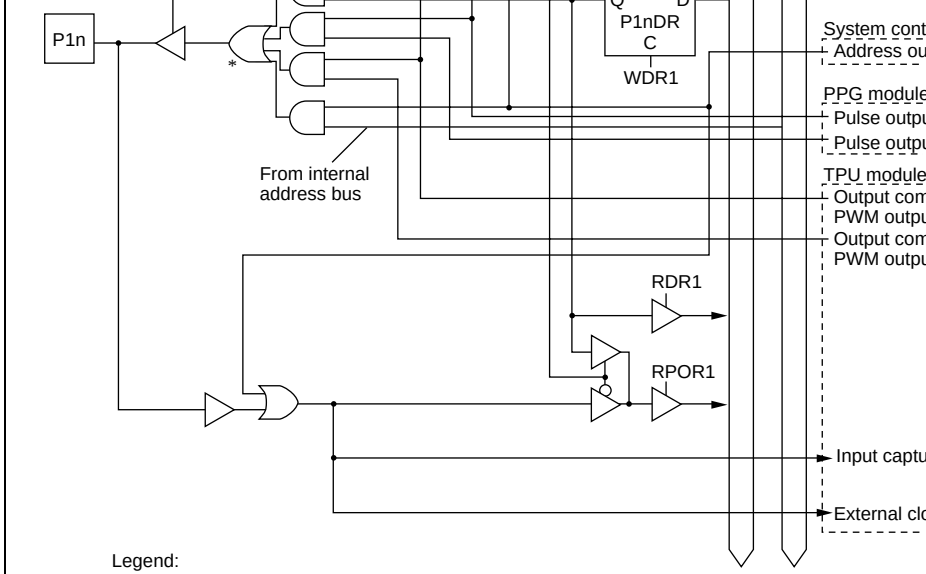
Note: * Determined by the state of pins PF7 to PF0.



Legend:
WDDR1: Write to P1DDR
WDR1: Write to P1DR
RDR1: Read P1DR
RPOR1: Read port 1

Notes: n = 0 or 1
* Priority order: address output > output compare output/PWM output > pulse output > DR

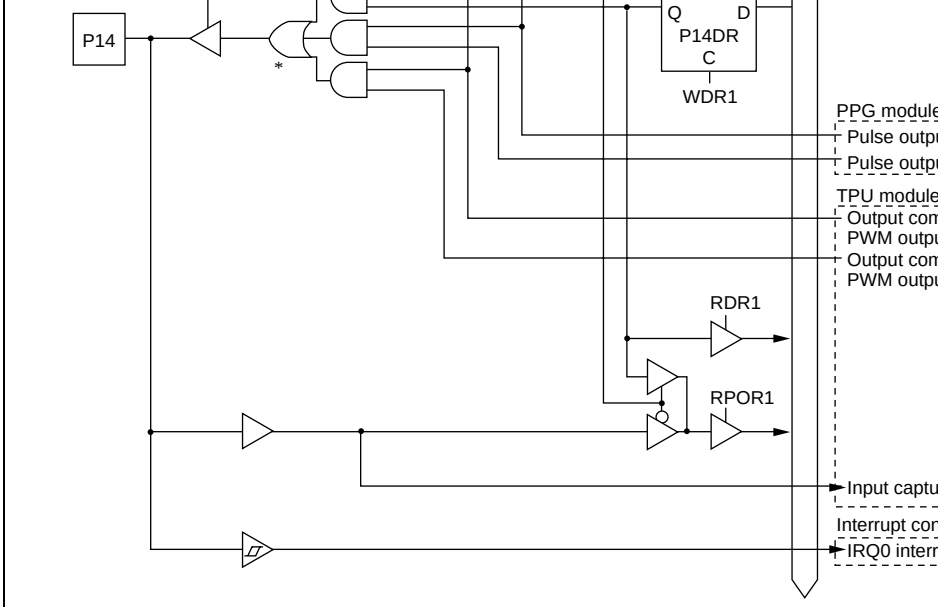
Figure C.1 (a) Port 1 Block Diagram (Pins P10 and P11)



Legend:
WDDR1: Write to P1DDR
WDR1: Write to P1DR
RDR1: Read P1DR
RPOR1: Read port 1

Notes: n = 2 or 3
* Priority order: address output > output compare output/PWM output > pulse output > DR

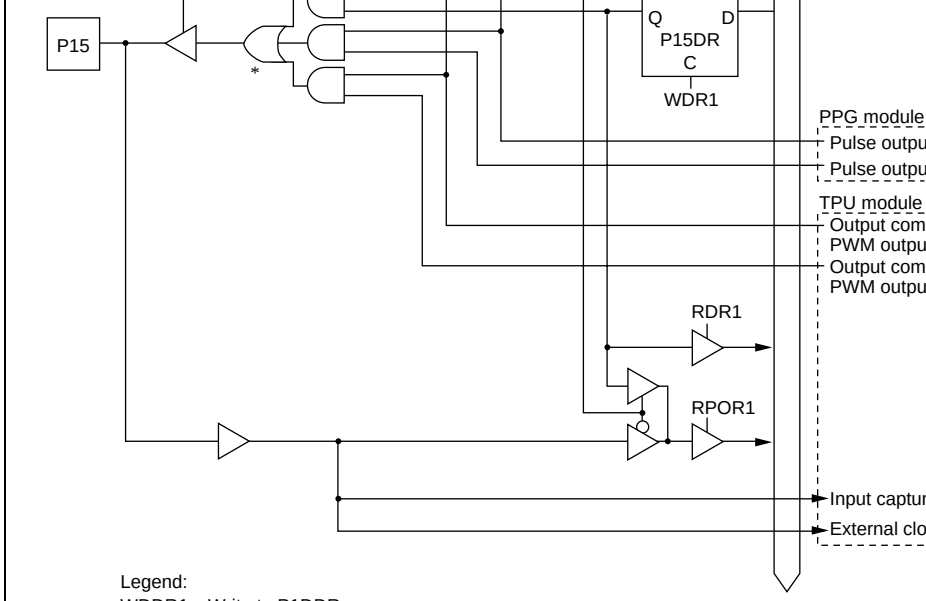
Figure C.1 (b) Port 1 Block Diagram (Pins P12 and P13)



Legend:
WDDR1: Write to P1DDR
WDR1: Write to P1DR
RDR1: Read P1DR
RPOR1: Read port 1

Note: * Priority order: output compare output/PWM output > pulse output > DR output

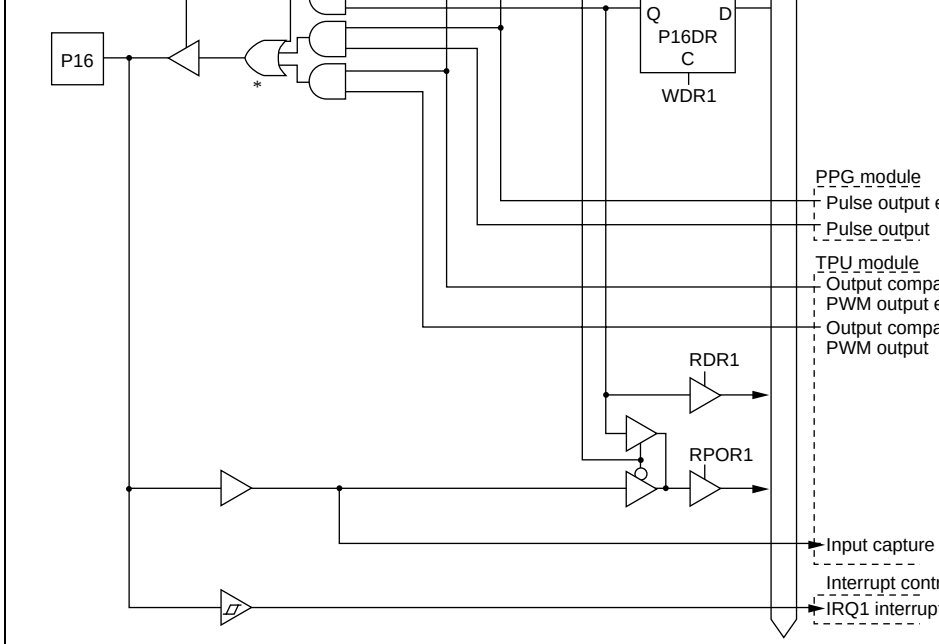
Figure C.1 (c) Port 1 Block Diagram (Pin P14)



Legend:
WDDR1: Write to P1DDR
WDR1: Write to P1DR
RDR1: Read P1DR
RPOR1: Read port 1

Note: * Priority order: output compare output/PWM output > pulse output > DR output

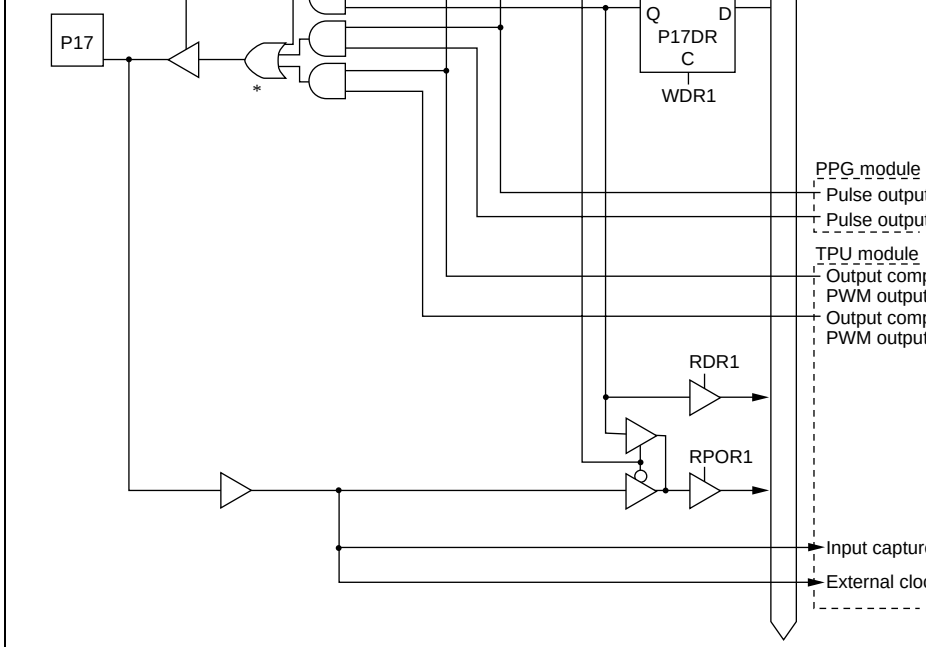
Figure C.1 (d) Port 1 Block Diagram (Pin P15)



Legend:
WDDR1: Write to P1DDR
WDR1: Write to P1DR
RDR1: Read P1DR
RPOR1: Read port 1

Note: * Priority order: output compare output/PWM output > pulse output > DR output

Figure C.1 (e) Port 1 Block Diagram (Pin P16)



Legend:
WDDR1: Write to P1DDR
WDR1: Write to P1DR
RDR1: Read P1DR
RPOR1: Read port 1

Note: * Priority order: output compare output/PWM output > pulse output > DR output

Figure C.1 (f) Port 1 Block Diagram (Pin P17)

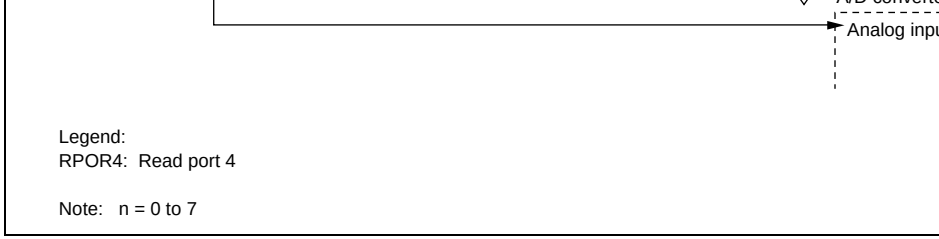


Figure C.2 Port 4 Block Diagram (Pins P40 to P47)

C.3 Port 9 Block Diagram

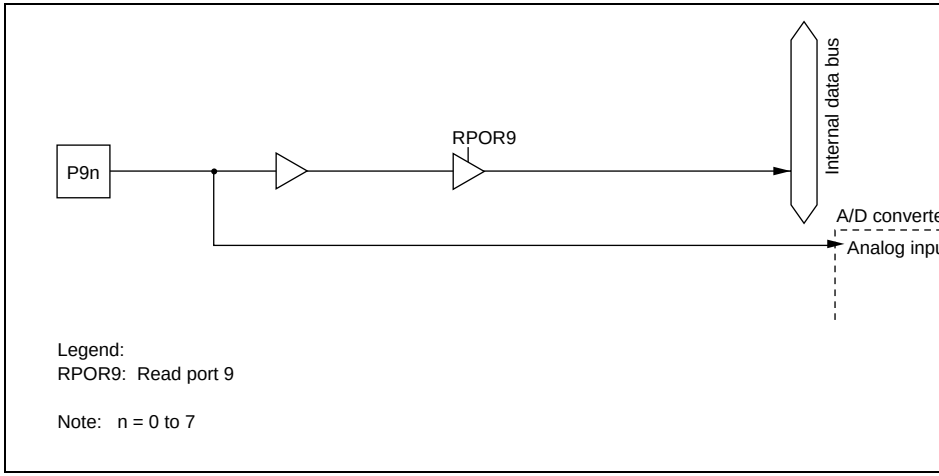


Figure C.3 Port 9 Block Diagram (Pins P90 to P97)

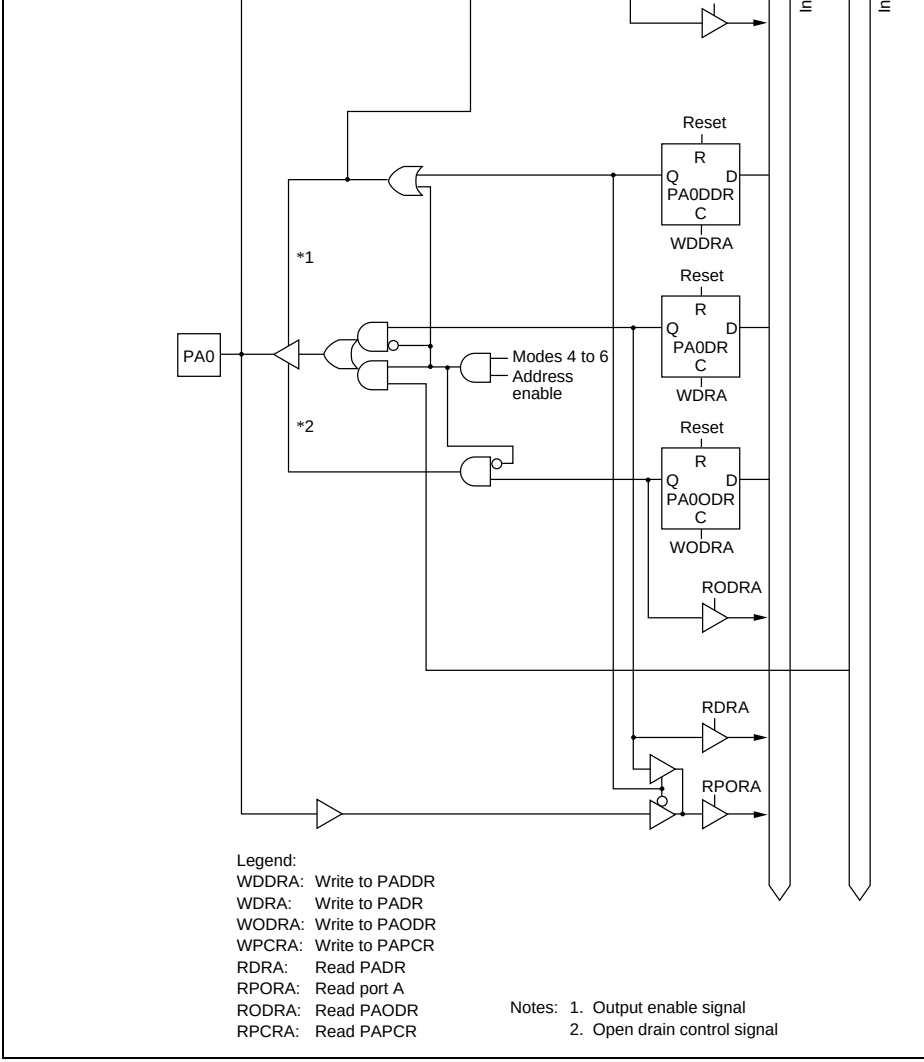


Figure C.4 (a) Port A Block Diagram (Pin PA0)

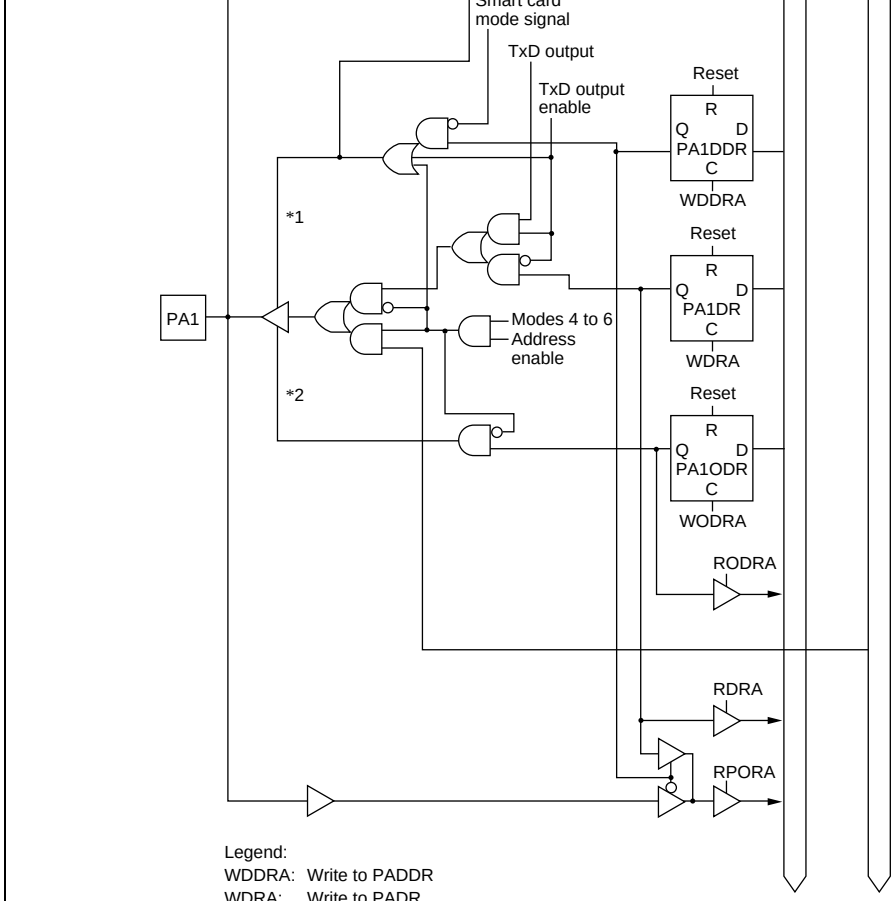


Figure C.4 (b) Port A Block Diagram (Pin PA1)

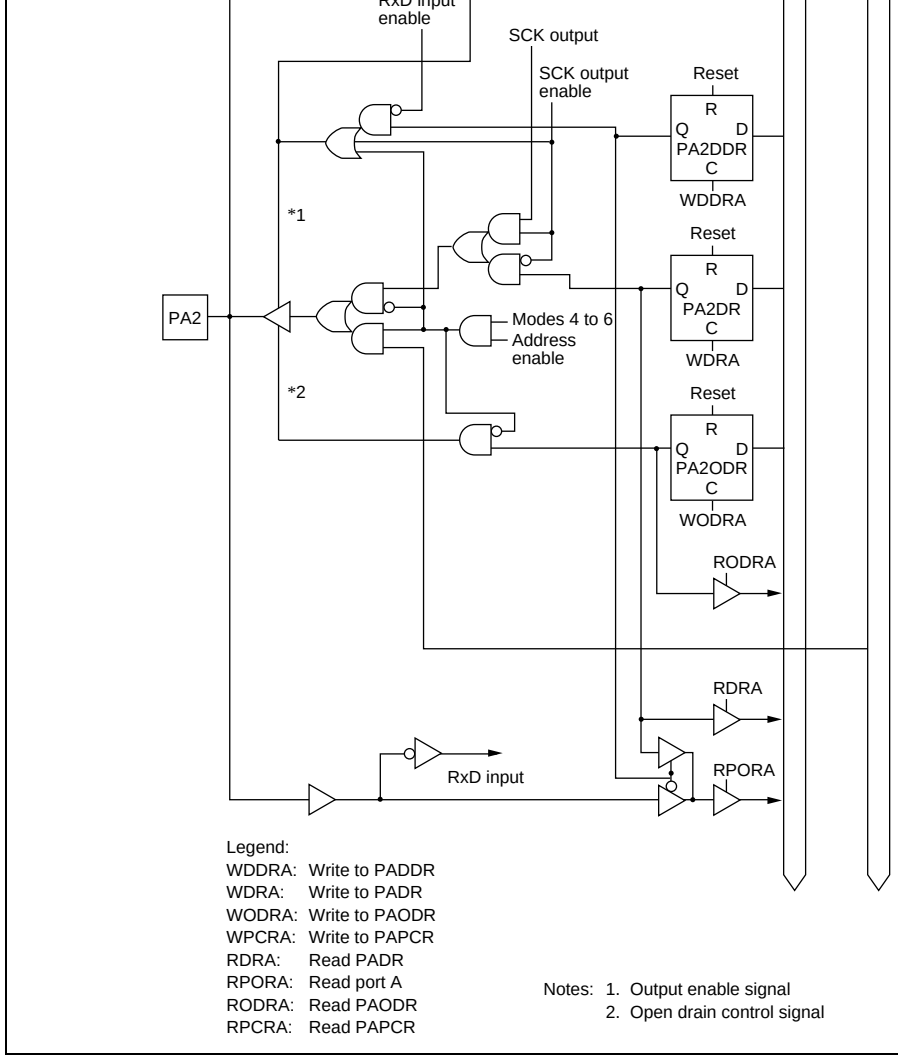


Figure C.4 (c) Port A Block Diagram (Pin PA2)

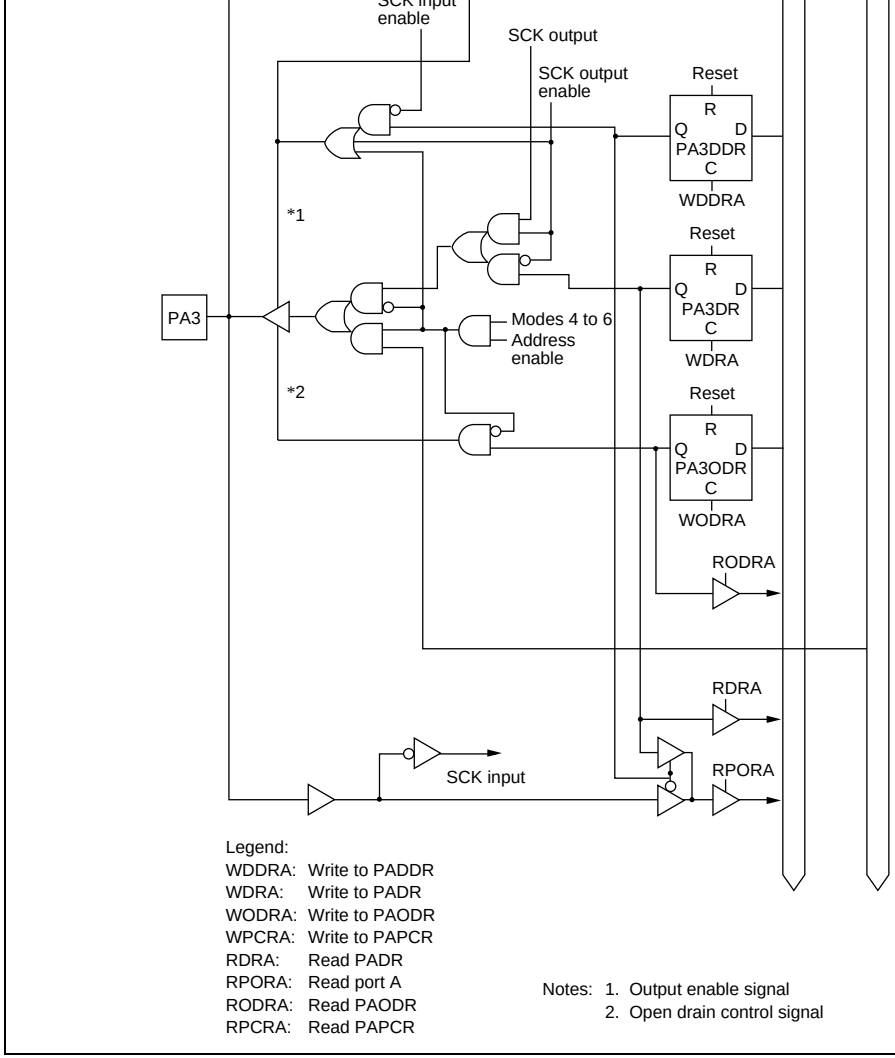


Figure C.4 (d) Port A Block Diagram (Pin PA3)

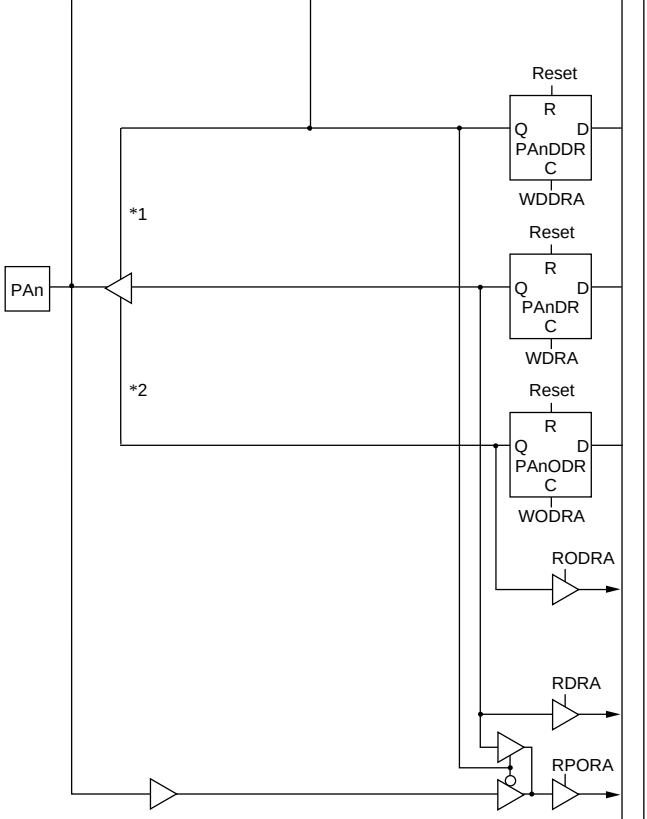


Figure C.4 (e) Port A Block Diagram (Pins PA4 and PA5)

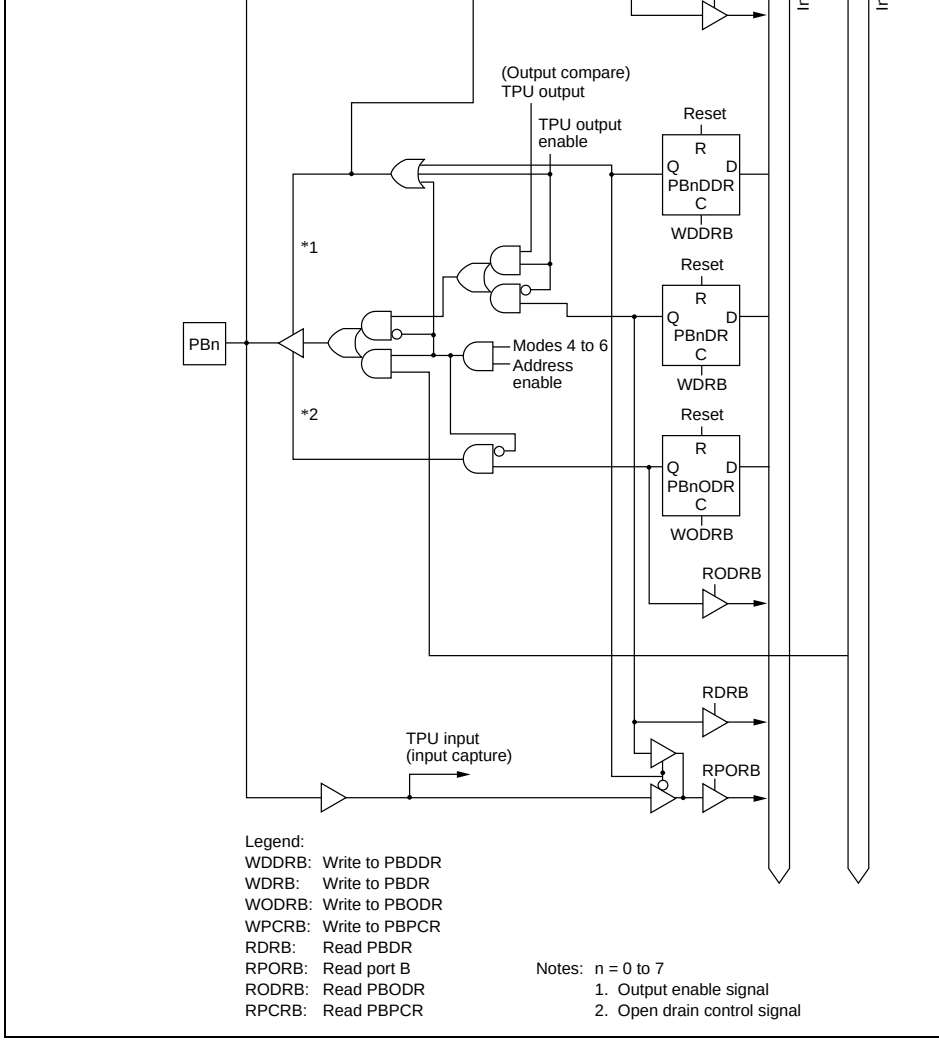


Figure C.5 Port B Block Diagram (Pins PB0 to PB7)

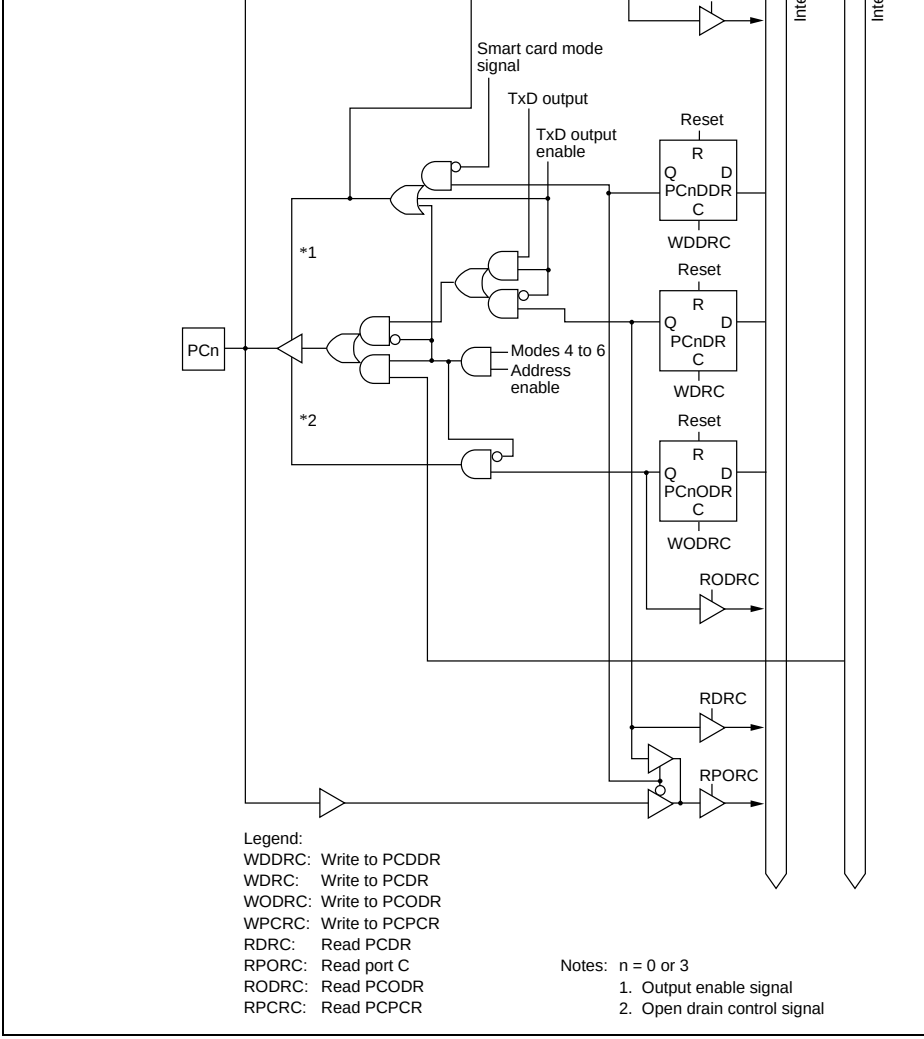


Figure C.6 (a) Port C Block Diagram (Pins PC0 and PC3)

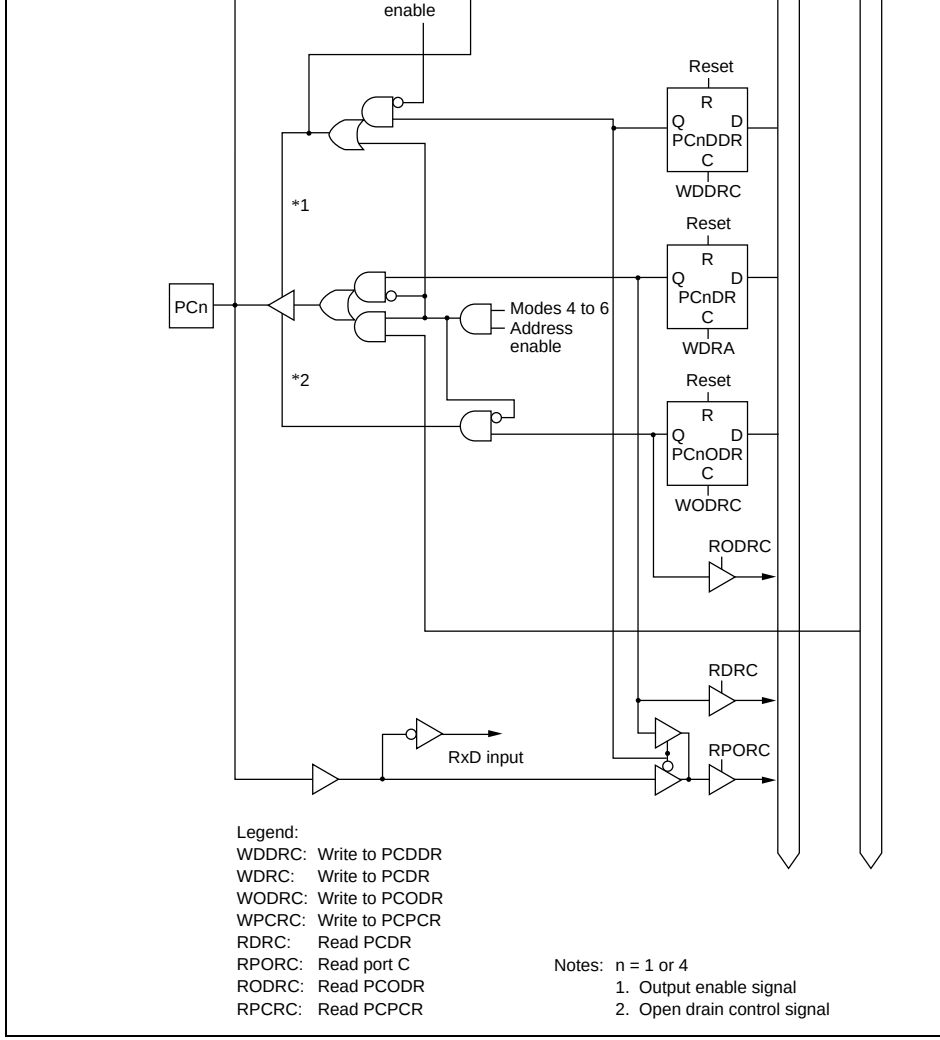


Figure C.6 (b) Port C Block Diagram (Pins PC1 and PC4)

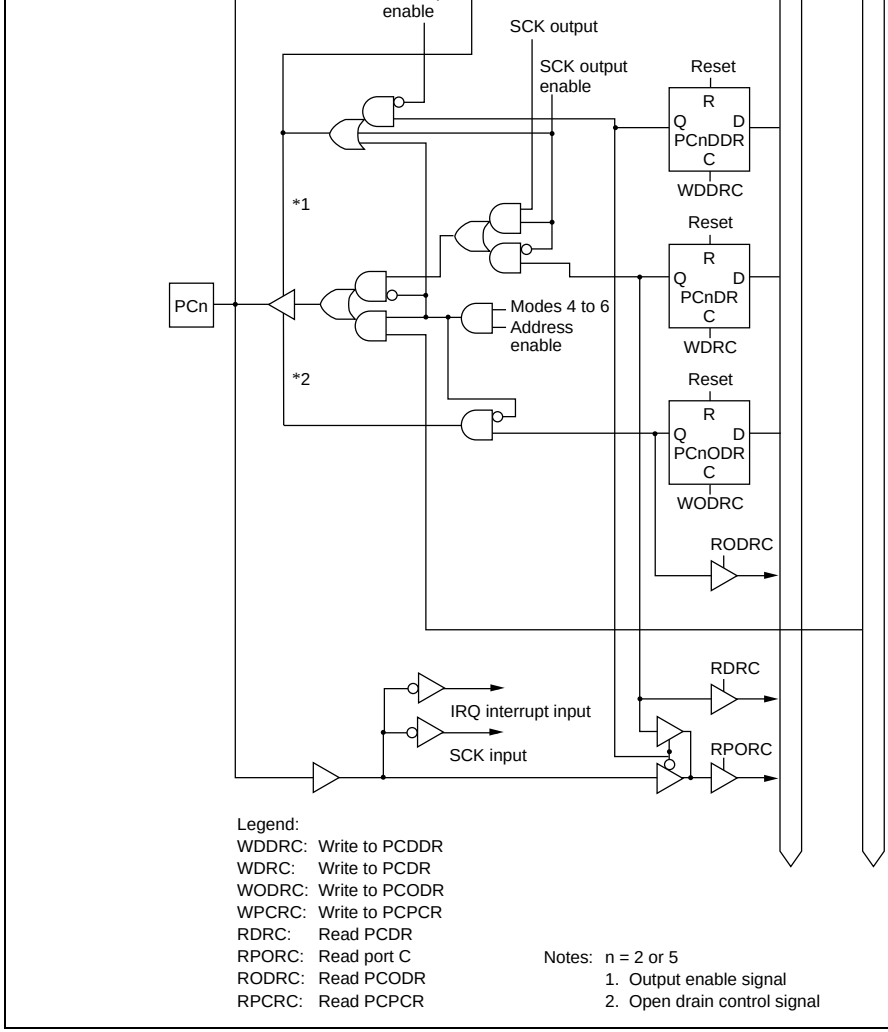


Figure C.6 (c) Port C Block Diagram (Pins PC2 and PC5)

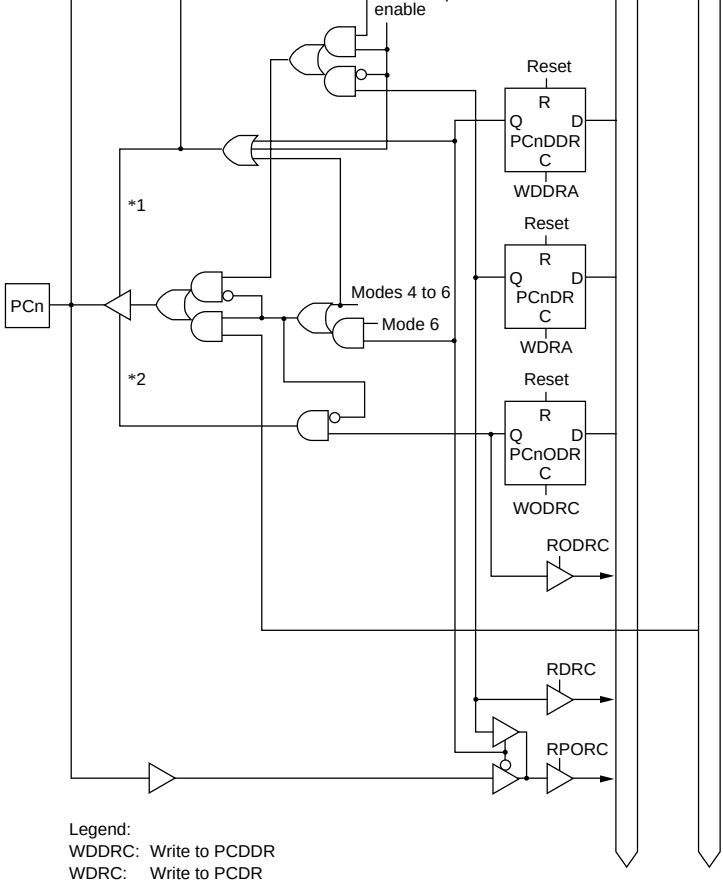


Figure C.6 (d) Port C Block Diagram (Pins PC6 and PC7)

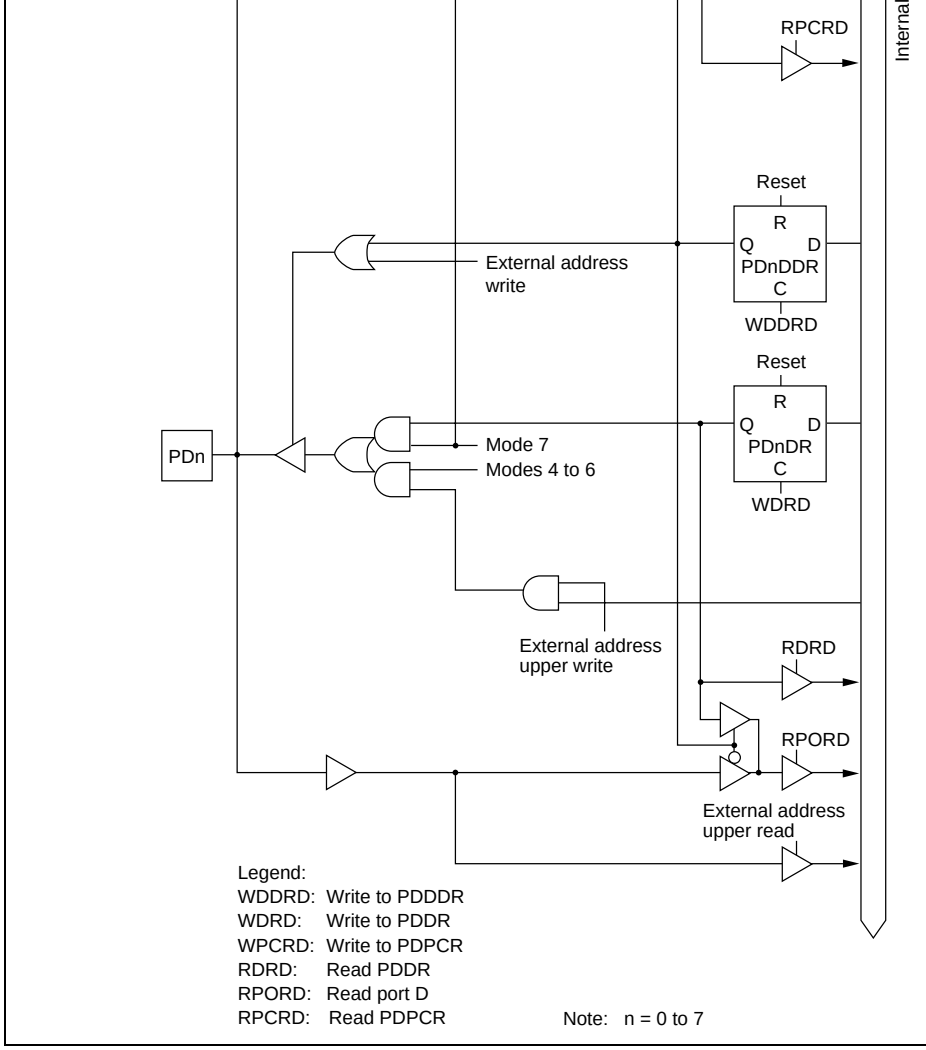


Figure C.7 Port D Block Diagram (Pins PD0 to PD7)

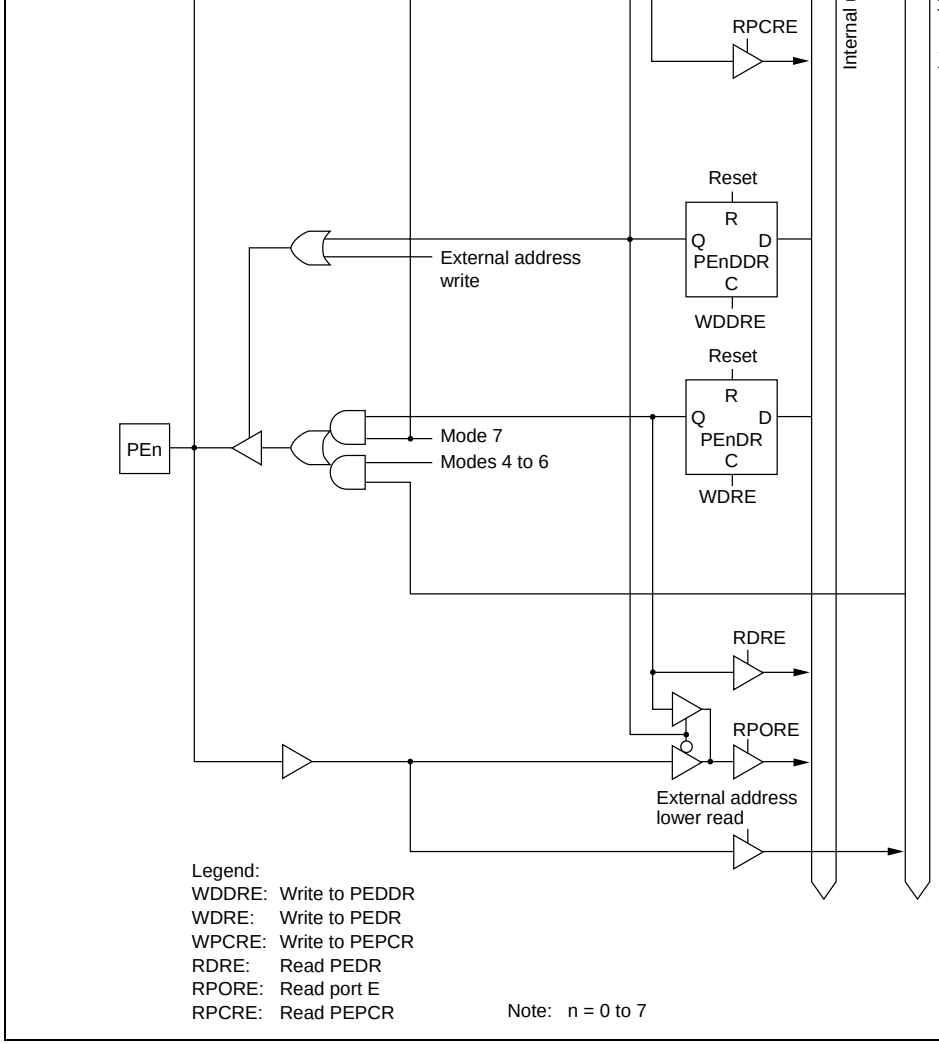


Figure C.8 Port E Block Diagram (Pins PE0 to PE7)

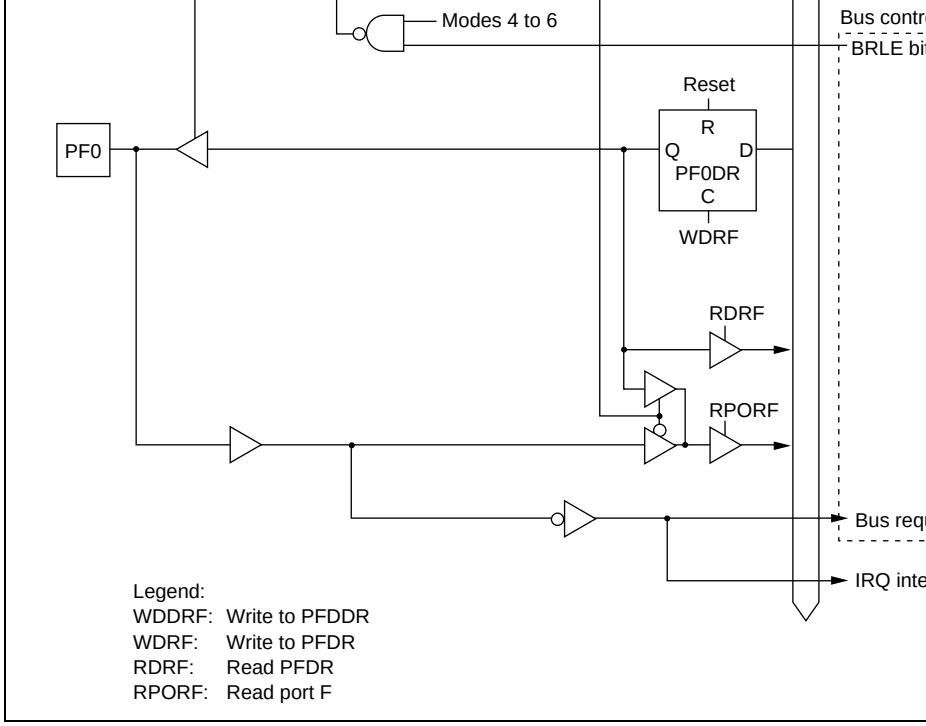


Figure C.9 (a) Port F Block Diagram (Pin PF0)

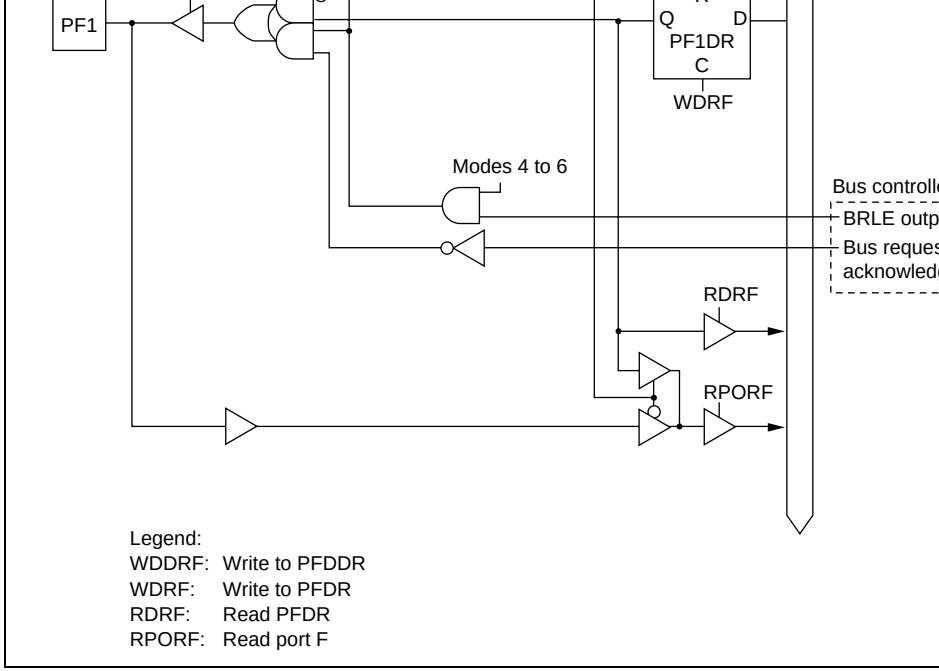


Figure C.9 (b) Port F Block Diagram in the H8S/2623 Group (Pin PF1

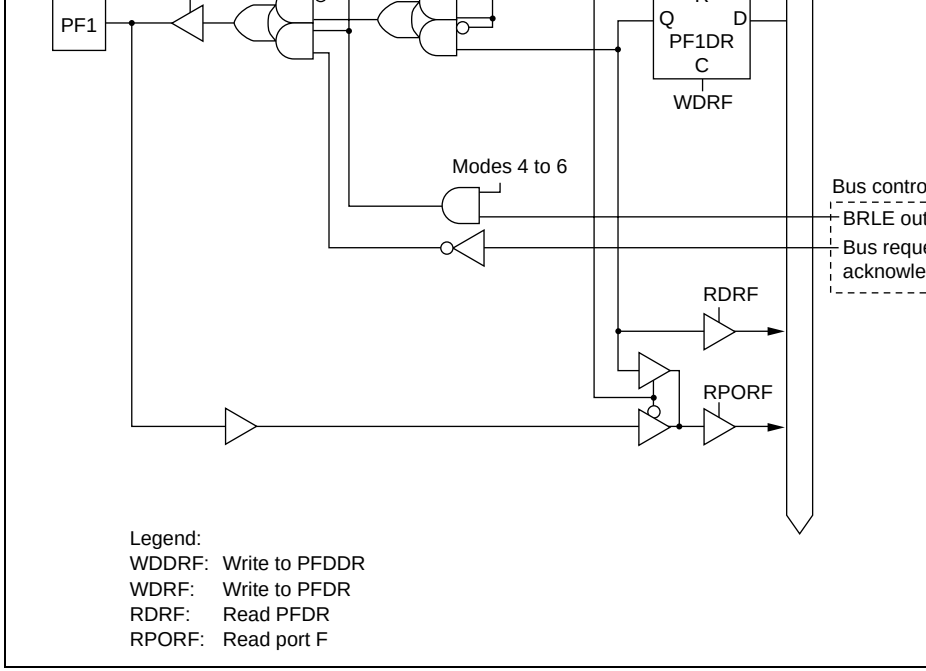


Figure C.9 (c) Port F Block Diagram in the H8S/2626 Group (Pin PF

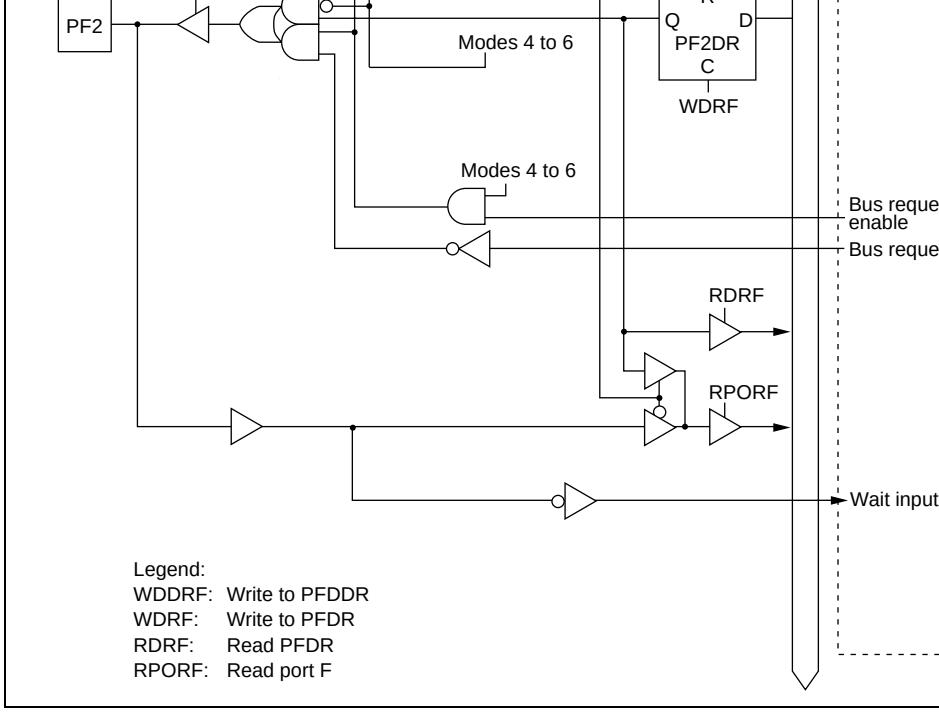


Figure C.9 (d) Port F Block Diagram (Pin PF2)

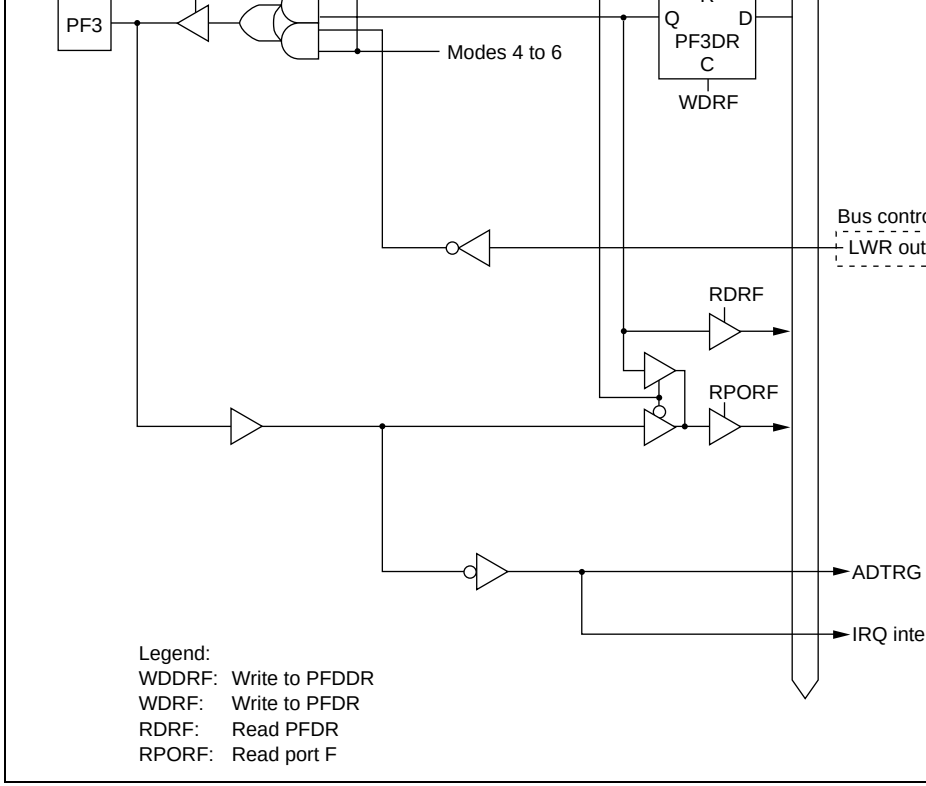


Figure C.9 (e) Port F Block Diagram (Pin PF3)

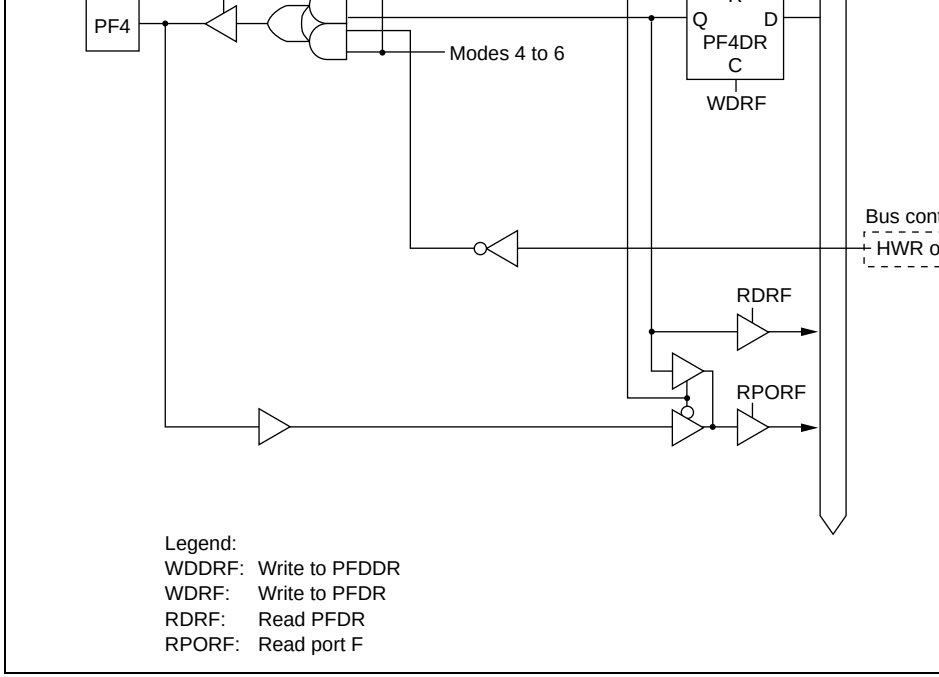


Figure C.9 (f) Port F Block Diagram (Pin PF4)

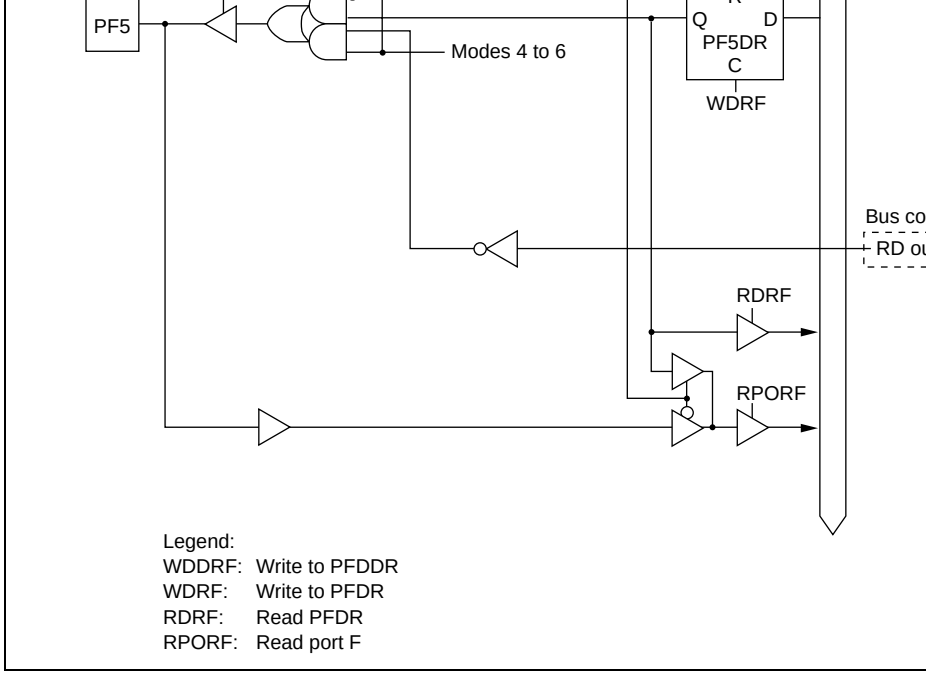


Figure C.9 (g) Port F Block Diagram (Pin PF5)

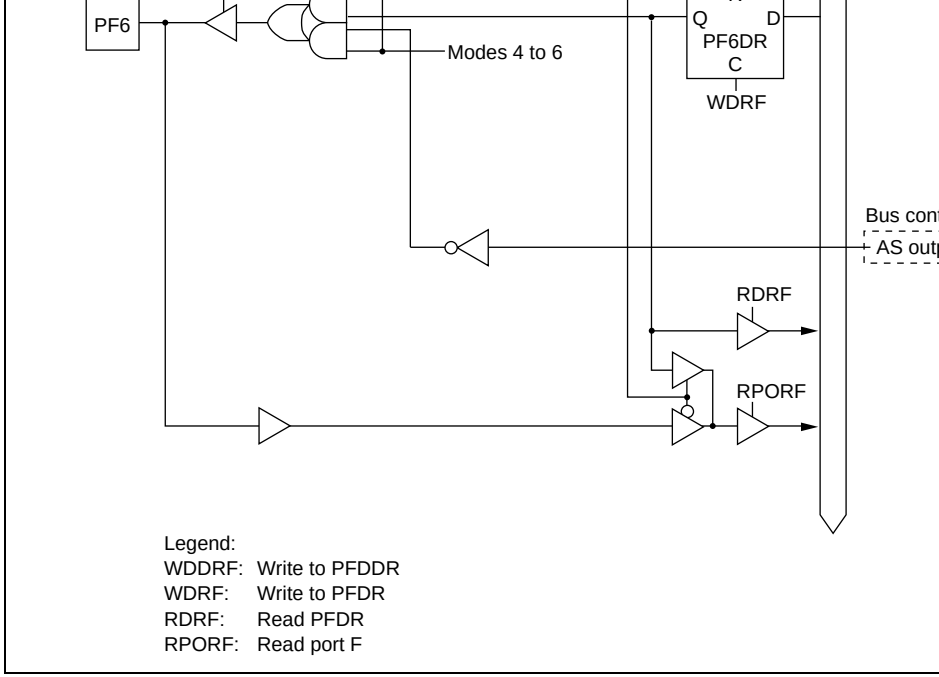
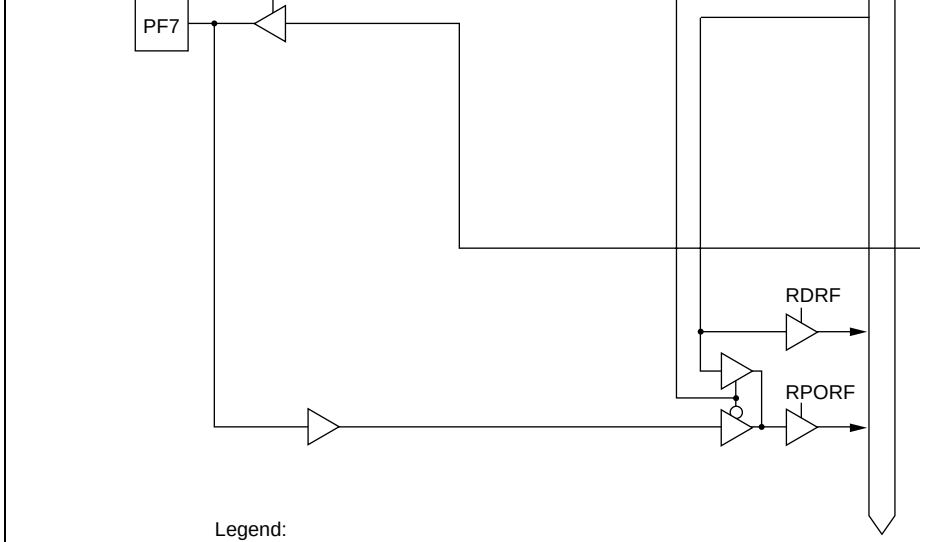


Figure C.9 (h) Port F Block Diagram (Pin PF6)



Legend:
WDDRF: Write to PFDDR
WDRF: Write to PFDR
RDRF: Read PFDR
RPORF: Read port F

Note: * Set priority

Figure C.9 (i) Port F Block Diagram (Pin PF7)

Pin Name	Mode	Reset	Mode	Standby Mode	State	Sleep
Port 1	4, 5	L	T	[Address output, OPE = 0] T	[Address output] T	[Addr A23 t
	6	T		[Address output, OPE = 1] kept [Otherwise] kept	[Otherwise] kept	[Othe I/O p
	7	T	T	kept	kept	I/O p
Port 4	4 to 7	T	T	T	T	Input
Port 9	4 to 7	T	T	T	T	Input
PA5 PA4	4 to 7	T	T	kept	kept	I/O p
PA3/A19 PA2/A18 PA1/A17 PA0/A16	4, 5	L	T	[Address output, OPE = 0] T	[Address output] T	[Addr A19 t
	6	T		[Address output, OPE = 1] kept [Otherwise] kept	[Otherwise] kept	[Othe I/O p
	7	T	T	kept	kept	I/O p
Port B	4, 5	L	T	[Address output, OPE = 0] T	[Address output] T	[Addr A15 t
	6	T		[Address output, OPE = 1] kept [Otherwise] kept	[Otherwise] kept	[Othe I/O p
	7	T	T	kept	kept	I/O p

					[DDR = 1, OPE = 1] kept [DDR = 0] kept		[DDI I/O p
	7		T	T	kept	kept	I/O p
Port D	4 to 6		T	T	T	T	Data
	7		T	T	kept	kept	I/O p
Port E	4 to 6	8-bit bus	T	T	kept	kept	I/O p
		16-bit bus	T	T	T	T	Data
	7		T	T	kept	kept	I/O p
PF7/ϕ	4 to 6		Clock output	T	[DDR = 0] T	kept	[DDI T
	7		T		[DDR = 1] H		[DDI Cloc
PF6/$\overline{AS}$	4 to 6		H	T	[OPE = 0] T [OPE = 1] H	T	$\overline{AS}$
	7		T	T	kept	kept	I/O p
PF5/$\overline{RD}$ PF4/$\overline{HWR}$ PF3/$\overline{LWR}$/ $\overline{ADTRG}$/ $\overline{IRQ3}$	4 to 6		H	T	[OPE = 0] T [OPE = 1] H	T	$\overline{RD}$,
	7		T	T	kept	kept	I/O p

				I/O port	I/O port	I/O po
				[BRLE = 1] H	[BRLE = 1] L	[BRLE $\overline{\text{BACK}}$
	7	T	T	kept	kept	I/O po
PF0/$\overline{\text{BREQ}}$/ $\overline{\text{IRQ2}}$	4 to 6	T	T	[BRLE = 0] kept	T	[BRLE I/O po
				[BRLE = 1] T		[BRLE $\overline{\text{BREQ}}$
	7	T	T	kept	kept	I/O po
HTxD	4 to 7	H	T	H	H	Outpu
HRxD	4 to 7	Input	T	T	Input	Input

Legend:

H : High level
L : Low level
T : High impedance
kept : Input port becomes high-impedance, output port retains state
DDR : Data direction register
OPE : Output port enable
WAITE : Wait input enable
BRLE : Bus release enable
BREQOE : BREQO pin enable

STBY signal goes low (delay from STBY low to RES high: 0 ns or more).

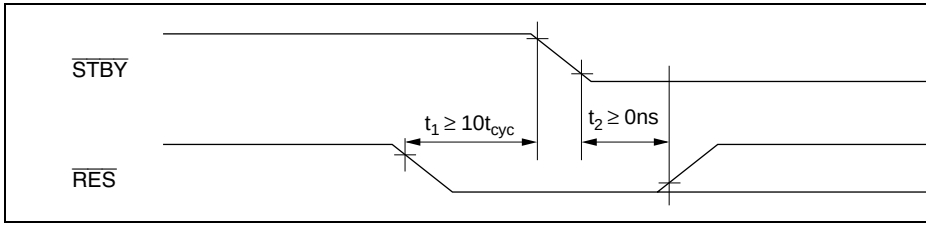


Figure E.1 Timing of Transition to Hardware Standby Mode

(2) To retain RAM contents with the RAME bit cleared to 0 in SYSCR, or when RAM not need to be retained, $\overline{RES}$ does not have to be driven low as in (1).

Timing of Recovery from Hardware Standby Mode

Drive the $\overline{RES}$ signal low and the NMI signal high approximately 100 ns or more befo goes high to execute a power-on reset.

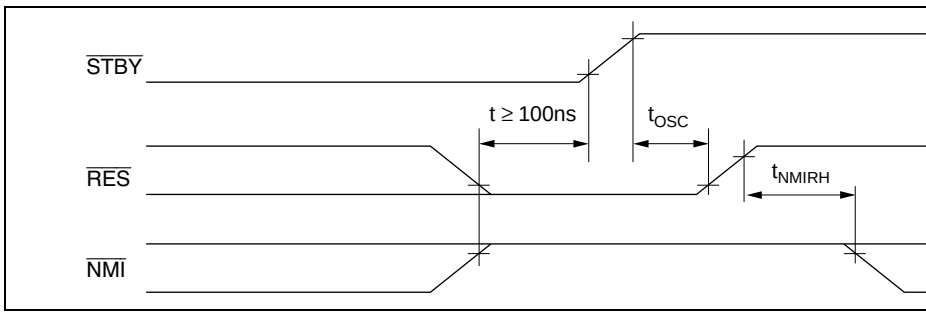


Figure E.2 Timing of Recovery from Hardware Standby Mode

H8S/2625		HD6432625	HD6432625FA	100-pin QFP (
H8S/2624		HD6432624	HD6432624FA	100-pin QFP (
H8S/2623	F-ZTAT version	HD64F2623	HD64F2623FA	100-pin QFP (
	Mask ROM version	HD6432623	HD6432623FA	100-pin QFP (
H8S/2622		HD6432622	HD6432622FA	100-pin QFP (
H8S/2621		HD6432621	HD6432621FA	100-pin QFP (

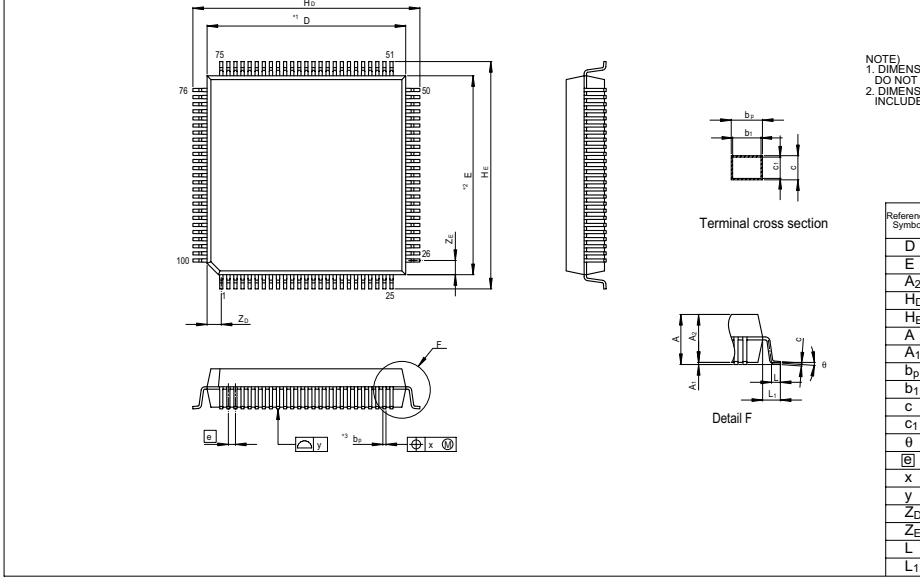


Figure G.1 FP-100B Package Dimensions

Renesas 16Bit Single-Chip Microcomputer
Hardware Manual
H8S/2626 Group, H8S/2623 Group,
H8S/2626F-ZTAT™, H8S/2623F-ZTAT™

Publication Date:	1st Edition, December 1998
	Rev.5.00, January 10, 2006
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, U.K.
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

Renesas Technology (Shanghai) Co., Ltd.
Unit 205, AZIA Center, No.133 Yincheng Rd (n), Pudong District, Shanghai 200120, China
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7898

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

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

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

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

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

H8S/2626 Group, H8S/2623 Group, H8S/2626F-ZTAT™, H8S/2623F-ZTAT™

Hardware Manual

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

REJ09B0275-0500

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for renesas ***manufacturer:***

Other Similar products are found below :

EL4511CUZ-T7 PYB15-Q24-S5-H-U PQA30-D24-S24-DH PQA30-D48-S12-TH PYB30-Q24-T312-H-U PYB15-Q24-S5-H-T PYB15-Q24-S12-H-T V7815-500-SMT PYB20-Q48-S12-H-T PQZ6-Q24-S15-D PYB20-Q48-S5-H-T PYB20-Q24-S12-H-T VLED15-120-350 VGS-75-12 PYB15-Q24-S12-H-U VGS-50-15 VGS-50-24 R5F100GFAFB#V0 VGS-25-24 VGS-50-5 VGS-100-12 M30620FCAFP#U3 ETSA120500UD-P5P-SZ PDQ2-D24-S12-S PDS1-S12-D12-M PDS1-S12-D15-M PYB15-Q24-S12-T PYB15-Q24-S24 PYB20-Q24-S24-DIN PYB20-Q48-S12 R0K33062PS000BE R0K505220S000BE R0K561664S000BE R0K570865S000BE HC55185AIMZ R5F52108CDFP#30 R5F72145BDFA#V1 R5S72631P200FP R7S721001VCBGAC0 EPSA050250UB-P5P-EJ HS0005PUU01H IS82C55A-5 ISL29035EVAL1Z ISL55110IVZ ISL6730AEVAL1Z ISL68200DEMO1Z ISL78235EVAL2Z ISL78268EVAL1Z ISL91107IRA-EVZ ISL9220IRTZEVAL1Z