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/2339 Group

Hardware Manual

Renesas 16-Bit Single-Chip Microcomputer H8S Family/H8S/2300 Series

H8S/2339	HD64F2339
	HD64F2339E
H8S/2338	HD6432338
	HD64F2338
H8S/2337	HD6432337
H8S/2332	HD6412332

Renesas Electronics
www.renesas.com

Rev.4.00 2007.09

Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.

2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life

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

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

1. Handling of Unused Pins

   Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.

   — The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions may occur due to the false recognition of the pin state as an input signal. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

   The state of the product is undefined at the moment when power is supplied.

   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.

   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.

   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products

   Before changing from one product to another, i.e. to one with a different type number, confirm that the change will not lead to problems.

   — The characteristics of MPU/MCU in the same group but having different type numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different type numbers, implement a system-evaluation test for each of the products.

This LSI is a single-chip microcomputer made up of the H8S/2000 CPU with an internal 32-bit architecture as its core, and the peripheral functions required to configure a system.

This LSI is equipped with ROM, RAM, a bus controller, data transfer controller (DTC), a 16-bit timer pulse unit (TPU), a watchdog timer (WDT), a serial communication interface (SCI), a DMA controller (DMAC), a D/A converter, an A/D converter, and I/O ports as on-chip supporting modules. This LSI is suitable for use as an embedded processor for high-level control systems. Its on-chip ROM are flash memory (F-ZTAT™*) and mask ROM that provides flexibility as it can be reprogrammed in no time to cope with all situations from the early stages of mass production to full-scale mass production. This is particularly applicable to application devices with specifications that will most probably change.

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

Target Users: This manual was written for users who will be using the H8S/2339 Group in the design of application systems. Members of this audience are expected to understand the fundamentals of electrical circuits, logical circuits, and microcomputers.

Objective: This manual was written to explain the hardware functions and electrical characteristics of the H8S/2339 Group to the above audience. Refer to the H8S/2600 Series, H8S/2000 Series Software Manual for a detailed description of the instruction set.

Notes on Reading This Manual:

- In order to understand the overall functions of the chip
  Read the manual according to the contents. This manual can be roughly categorized into parts on the CPU, system control functions, peripheral functions and electrical characteristics.
- In order to understand the details of the CPU's functions
  Read the H8S/2600 Series, H8S/2000 Series Software Manual.
- In order to understand the details of a register when its name is known
  The addresses, bits, and initial values of the registers are summarized in appendix B, Internal I/O Registers.
  Example: Bit order: The MSB is on the left and the LSB is on the right.

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

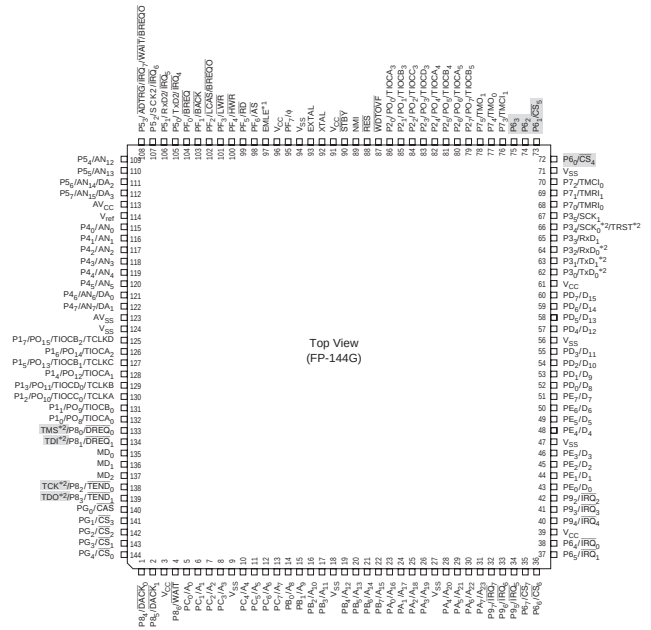
Document Title	Document No.
H8S/2339 Group Hardware Manual	This manual
H8S/2600 Series, H8S/2000 Series Software Manual	REJ09B0139

User's Manuals for Development Tools:

Document Title	Document No.
H8S, H8/300 Series C/C++ Compiler, Assembler, Optimized Linkage Editor Compiler Package Ver.6.01 User's Manual	REJ10B0161
H8S, H8/300 Series Simulator/Debugger (for Windows) User's Manual	ADE-702-037
H8S, H8/300 Series High-performance Embedded Workshop User's Manual	ADE-702-201

Application Notes:

Document Title	Document No.
H8S Family Technical Q & A	REJ05B0397

Item	Page	Revision (See Manual for Details)
1.1 Overview Table 1.1 Overview	6	Table amended Item: Specification Product lineup — Model: HD64F2339E* — Condition A: —, Condition B: O HD64F2339 — Condition A: —, Condition B: O HD6432338 — Condition A: O, Condition B: O
1.3.1 Pin Arrangement Figure 1.4 HD64F2339E Pin Arrangement (TFP-144G: Top View)	10	Figure 1.4 amended 

Signals

…Enabling or disabling of $\overline{CS}_n$ signal output is performed by setting the data direction register (DDR) bit for the port corresponding to the particular $\overline{CS}_n$ pin, the CS/67 enable bit (CS/67E), and the CS25 enable bit (CS25E).

In expanded mode with on-chip ROM disabled, the $\overline{CS}_0$ pin is placed in the output state after a reset. Pins $\overline{CS}_1$ to $\overline{CS}_7$ are placed in the input state after a reset, so the corresponding DDR bits as well as bits CS/67E and CS25E should be set to 1 when outputting signals $\overline{CS}_1$ to $\overline{CS}_7$.

In expanded mode with on-chip ROM enabled, pins $\overline{CS}_0$ to $\overline{CS}_7$ are all placed in the input state after a reset, so the corresponding DDR bits as well as bits CS/67E and CS25E should be set to 1 when outputting signals $\overline{CS}_1$ to $\overline{CS}_7$. For details, …

Section 13 Watchdog Timer — 579 to 594

Note shown below deleted

Note: The $\overline{WDTOVF}$ pin function cannot be used in the F-ZTAT version.

14.2.8 Bit Rate Register (BRR) — 616

Table 14.3 amended

Table 14.3 BRR Settings for Various Bit Rates (Asynchronous Mode)

	φ = 25 MHz		
Bit Rate (bits/s)	**n**	**N**	**Error (%)**
110	3	110	–0.02
150	3	80	0.47
300	2	162	–0.15
600	2	80	0.47
1200	1	162	–0.15
2400	1	80	0.47
4800	0	162	–0.15
9600	0	80	0.47
19200	0	40	–0.76
31250	0	24	1.00
38400	0	19	1.73

19.4.1 Features — 738

Description amended

- Reprogramming capability

The flash memory can be reprogrammed min. 100 times.

- Reprogramming capability

The flash memory can be reprogrammed min. 100 times.

22.2.6 Flash Memory Characteristics

Table 22.21 Flash Memory Characteristics

910

Table 22.21 amended

Item		Symbol	Min	Typ	Max	Unit
Erase time*1*3*6		t_E	—	50	1000	ms/block
Reprogramming count		N_{WEC}	100*7	10000*8	—	Times
Data retention time*9		t_{DRP}	10	—	—	Years
Programming	Wait time after SWE bit x setting*1		1	—	—	μs

911

Notes 7 to 9 added

Notes: 7. Minimum number of times for which all characteristics are guaranteed after rewriting (Guarantee range is 1 to minimum value).

8. Reference value for 25°C (as a guideline, rewriting should normally function up to this value).

9. Data retention characteristic when rewriting is performed within the specification range, including the minimum value.

D.1 Port States in Each Mode

1207

Note *1 deleted

(Before) $\overline{\text{WDTOVF}}$*1 → (After) $\overline{\text{WDTOVF}}$

Table D.1 I/O Port States in Each Processing State

1208

Note 1 shown below deleted

Note: 1. The $\overline{\text{WDTOVF}}$ pin function cannot be used in the F-ZTAT version.

Section 1 Overview 1
1.1 Overview 1
1.2 Block Diagram 7
1.3 Pin Description 8
1.3.1 Pin Arrangement 8
1.3.2 Pin Functions in Each Operating Mode 11
1.4 Pin Functions 17

Section 2 CPU 25
2.1 Overview 25
2.1.1 Features 25
2.1.2 Differences between H8S/2600 CPU and H8S/2000 CPU 26
2.1.3 Differences from H8/300 CPU 27
2.1.4 Differences from H8/300H CPU 27
2.2 CPU Operating Modes 28
2.3 Address Space 31
2.4 Register Configuration 32
2.4.1 Overview 32
2.4.2 General Registers 33
2.4.3 Control Registers 34
2.4.4 Initial Register Values 36
2.5 Data Formats 36
2.5.1 General Register Data Formats 37
2.5.2 Memory Data Formats 39
2.6 Instruction Set 40
2.6.1 Overview 40
2.6.2 Instructions and Addressing Modes 41
2.6.3 Table of Instructions Classified by Function 42
2.6.4 Basic Instruction Formats 52
2.7 Addressing Modes and Effective Address Calculation 53
2.7.1 Addressing Mode 53
2.7.2 Effective Address Calculation 56
2.8 Processing States 60
2.8.1 Overview 60
2.8.2 Reset State 61
2.8.3 Exception-Handling State 62
2.8.4 Program Execution State 64

2.9 Basic Timing 65
2.9.1 Overview 65
2.9.2 On-Chip Memory (ROM, RAM) 65
2.9.3 On-Chip Supporting Module Access Timing 67
2.9.4 External Address Space Access Timing 68
2.10 Usage Note 68
2.10.1 TAS Instruction 68

Section 3 MCU Operating Modes 69
3.1 Overview 69
3.1.1 Operating Mode Selection (H8S/2338 F-ZTAT) 69
3.1.2 Operating Mode Selection (Mask ROM and ROMless Versions, H8S/2339 F-ZTAT) 70
3.1.3 Register Configuration 72
3.2 Register Descriptions 72
3.2.1 Mode Control Register (MDCR) 72
3.2.2 System Control Register (SYSCR) 73
3.2.3 System Control Register 2 (SYSCR2) (F-ZTAT Version Only) 74
3.3 Operating Mode Descriptions 75
3.3.1 Mode 1 75
3.3.2 Mode 2 (H8S/2339 F-ZTAT Only) 75
3.3.3 Mode 3 (H8S/2339 F-ZTAT Only) 75
3.3.4 Mode 4 (Expanded Mode with On-Chip ROM Disabled) 75
3.3.5 Mode 5 (Expanded Mode with On-Chip ROM Disabled) 75
3.3.6 Mode 6 (Expanded Mode with On-Chip ROM Enabled) 76
3.3.7 Mode 7 (Single-Chip Mode) 76
3.3.8 Modes 8 and 9 (H8S/2338 F-ZTAT Only) 76
3.3.9 Mode 10 (H8S/2338 F-ZTAT Only) 76
3.3.10 Mode 11 (H8S/2338 F-ZTAT Only) 76
3.3.11 Modes 12 and 13 (H8S/2338 F-ZTAT Only) 76
3.3.12 Mode 14 (H8S/2338 F-ZTAT Only) 77
3.3.13 Mode 15 (H8S/2338 F-ZTAT Only) 77
3.4 Pin Functions in Each Operating Mode 78
3.5 Memory Map in Each Operating Mode 78

Section 4 Exception Handling 87
4.1 Overview 87
4.1.1 Exception Handling Types and Priority 87

4.2 Reset ... 90
4.2.1 Overview ... 90
4.2.2 Reset Sequence ... 90
4.2.3 Interrupts after Reset ... 91
4.2.4 State of On-Chip Supporting Modules after Reset Release ... 91
4.3 Traces ... 92
4.4 Interrupts ... 93
4.5 Trap Instruction ... 94
4.6 Stack Status after Exception Handling ... 94
4.7 Notes on Use of the Stack ... 95

Section 5 Interrupt Controller ... 97
5.1 Overview ... 97
5.1.1 Features ... 97
5.1.2 Block Diagram ... 98
5.1.3 Pin Configuration ... 99
5.1.4 Register Configuration ... 99
5.2 Register Descriptions ... 100
5.2.1 System Control Register (SYSCR) ... 100
5.2.2 Interrupt Priority Registers A to K (IPRA to IPRK) ... 101
5.2.3 IRQ Enable Register (IER) ... 102
5.2.4 IRQ Sense Control Registers H and L (ISCRH, ISCRL) ... 103
5.2.5 IRQ Status Register (ISR) ... 104
5.3 Interrupt Sources ... 105
5.3.1 External Interrupts ... 105
5.3.2 Internal Interrupts ... 107
5.3.3 Interrupt Exception Vector Table ... 107
5.4 Interrupt Operation ... 113
5.4.1 Interrupt Control Modes and Interrupt Operation ... 113
5.4.2 Interrupt Control Mode 0 ... 116
5.4.3 Interrupt Control Mode 2 ... 118
5.4.4 Interrupt Exception Handling Sequence ... 120
5.4.5 Interrupt Response Times ... 122
5.5 Usage Notes ... 123
5.5.1 Contention between Interrupt Generation and Disabling ... 123
5.5.2 Instructions That Disable Interrupts ... 124
5.5.3 Times when Interrupts Are Disabled ... 124
5.5.4 Interrupts during Execution of EEPMOV Instruction ... 124

5.6.2 Block Diagram 126
5.6.3 Operation 127

Section 6 Bus Controller 129
6.1 Overview 129
6.1.1 Features 129
6.1.2 Block Diagram 131
6.1.3 Pin Configuration 132
6.1.4 Register Configuration 133
6.2 Register Descriptions 134
6.2.1 Bus Width Control Register (ABWCR) 134
6.2.2 Access State Control Register (ASTCR) 135
6.2.3 Wait Control Registers H and L (WCRH, WCRL) 135
6.2.4 Bus Control Register H (BCRH) 140
6.2.5 Bus Control Register L (BCRL) 142
6.2.6 Memory Control Register (MCR) 144
6.2.7 DRAM Control Register (DRAMCR) 146
6.2.8 Refresh Timer Counter (RTCNT) 148
6.2.9 Refresh Time Control Register (RTCOR) 148
6.3 Overview of Bus Control 149
6.3.1 Area Partitioning 149
6.3.2 Bus Specifications 150
6.3.3 Memory Interfaces 151
6.3.4 Advanced Mode 152
6.3.5 Chip Select Signals 153
6.4 Basic Bus Interface 154
6.4.1 Overview 154
6.4.2 Data Size and Data Alignment 154
6.4.3 Valid Strobes 156
6.4.4 Basic Timing 157
6.4.5 Wait Control 165
6.5 DRAM Interface 167
6.5.1 Overview 167
6.5.2 Setting DRAM Space 167
6.5.3 Address Multiplexing 168
6.5.4 Data Bus 169
6.5.5 Pins Used for DRAM Interface 169
6.5.6 Basic Timing 170

6.5.9 Byte Access Control 174
6.5.10 Burst Operation 176
6.5.11 Refresh Control 179
6.6 DMAC Single Address Mode and DRAM Interface 183
6.6.1 When DDS = 1 183
6.6.2 When DDS = 0 184
6.7 Burst ROM Interface 185
6.7.1 Overview 185
6.7.2 Basic Timing 185
6.7.3 Wait Control 187
6.8 Idle Cycle 188
6.8.1 Operation 188
6.8.2 Pin States in Idle Cycle 192
6.9 Write Data Buffer Function 193
6.10 Bus Release 194
6.10.1 Overview 194
6.10.2 Operation 194
6.10.3 Pin States in External-Bus-Released State 195
6.10.4 Transition Timing 196
6.10.5 Usage Note 197
6.11 Bus Arbitration 197
6.11.1 Overview 197
6.11.2 Operation 197
6.11.3 Bus Transfer Timing 198
6.11.4 External Bus Release Usage Note 198
6.12 Resets and Bus Controller 199

Section 7 DMA Controller 201
7.1 Overview 201
7.1.1 Features 201
7.1.2 Block Diagram 202
7.1.3 Overview of Functions 203
7.1.4 Pin Configuration 205
7.1.5 Register Configuration 206
7.2 Register Descriptions (1) (Short Address Mode) 207
7.2.1 Memory Address Registers (MAR) 208
7.2.2 I/O Address Register (IOAR) 209
7.2.3 Execute Transfer Count Register (ETCR) 209

7.3 Register Descriptions (2) (Full Address Mode) ... 221
7.3.1 Memory Address Register (MAR) ... 221
7.3.2 I/O Address Register (IOAR) ... 221
7.3.3 Execute Transfer Count Register (ETCR) ... 222
7.3.4 DMA Control Register (DMACR) ... 224
7.3.5 DMA Band Control Register (DMABCR) ... 228
7.4 Register Descriptions (3) ... 234
7.4.1 DMA Write Enable Register (DMAWER) ... 234
7.4.2 DMA Terminal Control Register (DMATCR) ... 237
7.4.3 Module Stop Control Register (MSTPCR) ... 238
7.5 Operation ... 239
7.5.1 Transfer Modes ... 239
7.5.2 Sequential Mode ... 241
7.5.3 Idle Mode ... 244
7.5.4 Repeat Mode ... 247
7.5.5 Single Address Mode ... 251
7.5.6 Normal Mode ... 254
7.5.7 Block Transfer Mode ... 257
7.5.8 DMAC Activation Sources ... 263
7.5.9 Basic DMAC Bus Cycles ... 266
7.5.10 DMAC Bus Cycles (Dual Address Mode) ... 267
7.5.11 DMAC Bus Cycles (Single Address Mode) ... 275
7.5.12 Write Data Buffer Function ... 281
7.5.13 DMAC Multi-Channel Operation ... 282
7.5.14 Relation Between the DMAC and External Bus Requests, Refresh Cycles, and the DTC ... 284
7.5.15 NMI Interrupts and DMAC ... 285
7.5.16 Forced Termination of DMAC Operation ... 286
7.5.17 Clearing Full Address Mode ... 287
7.6 Interrupts ... 288
7.7 Usage Notes ... 289

Section 8 Data Transfer Controller ... 295
8.1 Overview ... 295
8.1.1 Features ... 295
8.1.2 Block Diagram ... 296
8.1.3 Register Configuration ... 297
8.2 Register Descriptions ... 298

8.2.3 DTC Source Address Register (SAR) ... 301
8.2.4 DTC Destination Address Register (DAR) ... 301
8.2.5 DTC Transfer Count Register A (CRA) ... 302
8.2.6 DTC Transfer Count Register B (CRB) ... 302
8.2.7 DTC Enable Registers (DTCER) ... 303
8.2.8 DTC Vector Register (DTVECR) ... 304
8.2.9 Module Stop Control Register (MSTPCR) ... 305
8.3 Operation ... 305
8.3.1 Overview ... 305
8.3.2 Activation Sources ... 309
8.3.3 DTC Vector Table ... 310
8.3.4 Location of Register Information in Address Space ... 313
8.3.5 Normal Mode ... 314
8.3.6 Repeat Mode ... 315
8.3.7 Block Transfer Mode ... 316
8.3.8 Chain Transfer ... 318
8.3.9 Operation Timing ... 319
8.3.10 Number of DTC Execution States ... 320
8.3.11 Procedures for Using DTC ... 322
8.3.12 Examples of Use of the DTC ... 323
8.4 Interrupts ... 327
8.5 Usage Notes ... 328

Section 9 I/O Ports ... 329
9.1 Overview ... 329
9.2 Port 1 ... 334
9.2.1 Overview ... 334
9.2.2 Register Configuration ... 335
9.2.3 Pin Functions ... 337
9.3 Port 2 ... 345
9.3.1 Overview ... 345
9.3.2 Register Configuration ... 346
9.3.3 Pin Functions ... 348
9.4 Port 3 ... 356
9.4.1 Overview ... 356
9.4.2 Register Configuration ... 356
9.4.3 Pin Functions ... 359
9.5 Port 4 ... 361

9.5.3 Pin Functions 362
9.6 Port 5 362
9.6.1 Overview 362
9.6.2 Register Configuration 364
9.6.3 Pin Functions 368
9.7 Port 6 370
9.7.1 Overview 370
9.7.2 Register Configuration 371
9.7.3 Pin Functions 374
9.8 Port 7 376
9.8.1 Overview 376
9.8.2 Register Configuration 377
9.8.3 Pin Functions 379
9.9 Port 8 381
9.9.1 Overview 381
9.9.2 Register Configuration 382
9.9.3 Pin Functions 385
9.10 Port 9 387
9.10.1 Overview 387
9.10.2 Register Configuration 388
9.10.3 Pin Functions 391
9.11 Port A 393
9.11.1 Overview 393
9.11.2 Register Configuration 394
9.11.3 Pin Functions 398
9.11.4 MOS Input Pull-Up Function 400
9.12 Port B 401
9.12.1 Overview 401
9.12.2 Register Configuration 402
9.12.3 Pin Functions 404
9.12.4 MOS Input Pull-Up Function 406
9.13 Port C 407
9.13.1 Overview 407
9.13.2 Register Configuration 408
9.13.3 Pin Functions 410
9.13.4 MOS Input Pull-Up Function 412
9.14 Port D 413
9.14.1 Overview 413

9.14.4 MOS Input Pull-Up Function ... 418
9.15 Port E ... 419
9.15.1 Overview ... 419
9.15.2 Register Configuration ... 420
9.15.3 Pin Functions ... 423
9.15.4 MOS Input Pull-Up Function ... 425
9.16 Port F ... 426
9.16.1 Overview ... 426
9.16.2 Register Configuration ... 427
9.16.3 Pin Functions ... 431
9.17 Port G ... 433
9.17.1 Overview ... 433
9.17.2 Register Configuration ... 434
9.17.3 Pin Functions ... 437

Section 10 16-Bit Timer Pulse Unit (TPU) ... 439
10.1 Overview ... 439
10.1.1 Features ... 439
10.1.2 Block Diagram ... 443
10.1.3 Pin Configuration ... 444
10.1.4 Register Configuration ... 446
10.2 Register Descriptions ... 448
10.2.1 Timer Control Registers (TCR) ... 448
10.2.2 Timer Mode Registers (TMDR) ... 453
10.2.3 Timer I/O Control Registers (TIOR) ... 455
10.2.4 Timer Interrupt Enable Registers (TIER) ... 468
10.2.5 Timer Status Registers (TSR) ... 471
10.2.6 Timer Counters (TCNT) ... 474
10.2.7 Timer General Registers (TGR) ... 475
10.2.8 Timer Start Register (TSTR) ... 476
10.2.9 Timer Synchro Register (TSYR) ... 477
10.2.10 Module Stop Control Register (MSTPCR) ... 478
10.3 Interface to Bus Master ... 479
10.3.1 16-Bit Registers ... 479
10.3.2 8-Bit Registers ... 479
10.4 Operation ... 481
10.4.1 Overview ... 481
10.4.2 Basic Functions ... 482

10.4.5 Cascaded Operation 494
10.4.6 PWM Modes 496
10.4.7 Phase Counting Mode 502
10.5 Interrupts 508
10.5.1 Interrupt Sources and Priorities 508
10.5.2 DTC/DMAC Activation 510
10.5.3 A/D Converter Activation 510
10.6 Operation Timing 511
10.6.1 Input/Output Timing 511
10.6.2 Interrupt Signal Timing 515
10.7 Usage Notes 519

Section 11 Programmable Pulse Generator (PPG) 529
11.1 Overview 529
11.1.1 Features 529
11.1.2 Block Diagram 530
11.1.3 Pin Configuration 531
11.1.4 Registers 532
11.2 Register Descriptions 533
11.2.1 Next Data Enable Registers H and L (NDERH, NDERL) 533
11.2.2 Output Data Registers H and L (PODRH, PODRL) 534
11.2.3 Next Data Registers H and L (NDRH, NDRL) 535
11.2.4 Notes on NDR Access 535
11.2.5 PPG Output Control Register (PCR) 537
11.2.6 PPG Output Mode Register (PMR) 539
11.2.7 Port 1 Data Direction Register (P1DDR) 541
11.2.8 Port 2 Data Direction Register (P2DDR) 542
11.2.9 Module Stop Control Register (MSTPCR) 542
11.3 Operation 543
11.3.1 Overview 543
11.3.2 Output Timing 544
11.3.3 Normal Pulse Output 545
11.3.4 Non-Overlapping Pulse Output 547
11.3.5 Inverted Pulse Output 550
11.3.6 Pulse Output Triggered by Input Capture 551
11.4 Usage Notes 552
11.4.1 Operation of Pulse Output Pins 552
11.4.2 Note on Non-Overlapping Output 552

12.1.1 Features ... 555
12.1.2 Block Diagram ... 556
12.1.3 Pin Configuration ... 557
12.1.4 Register Configuration ... 557
12.2 Register Descriptions ... 558
12.2.1 Timer Counters 0 and 1 (TCNT0, TCNT1) ... 558
12.2.2 Time Constant Registers A0 and A1 (TCORA0, TCORA1) ... 558
12.2.3 Time Constant Registers B0 and B1 (TCORB0, TCORB1) ... 559
12.2.4 Time Control Registers 0 and 1 (TCR0, TCR1) ... 559
12.2.5 Timer Control/Status Registers 0 and 1 (TCSR0, TCSR1) ... 561
12.2.6 Module Stop Control Register (MSTPCR) ... 564
12.3 Operation ... 565
12.3.1 TCNT Incrementation Timing ... 565
12.3.2 Compare Match Timing ... 566
12.3.3 Timing of TCNT External Reset ... 568
12.3.4 Timing of Overflow Flag (OVF) Setting ... 568
12.3.5 Operation with Cascaded Connection ... 569
12.4 Interrupts ... 570
12.4.1 Interrupt Sources and DTC Activation ... 570
12.4.2 A/D Converter Activation ... 570
12.5 Sample Application ... 571
12.6 Usage Notes ... 572
12.6.1 Contention between TCNT Write and Clear ... 572
12.6.2 Contention between TCNT Write and Increment ... 573
12.6.3 Contention between TCOR Write and Compare Match ... 574
12.6.4 Contention between Compare Matches A and B ... 575
12.6.5 Switching of Internal Clocks and TCNT Operation ... 575
12.6.6 Interrupts and Module Stop Mode ... 577

Section 13 Watchdog Timer ... 579
13.1 Overview ... 579
13.1.1 Features ... 579
13.1.2 Block Diagram ... 580
13.1.3 Pin Configuration ... 581
13.1.4 Register Configuration ... 581
13.2 Register Descriptions ... 582
13.2.1 Timer Counter (TCNT) ... 582
13.2.2 Timer Control/Status Register (TCSR) ... 583

13.3 Operation 587
13.3.1 Operation in Watchdog Timer Mode 587
13.3.2 Operation in Interval Timer Mode 589
13.3.3 Timing of Overflow Flag (OVF) Setting 590
13.3.4 Timing of Watchdog Timer Overflow Flag (WOVF) Setting 591
13.4 Interrupts 592
13.5 Usage Notes 592
13.5.1 Contention between Timer Counter (TCNT) Write and Increment 592
13.5.2 Changing Value of CKS2 to CKS0 593
13.5.3 Switching between Watchdog Timer Mode and Interval Timer Mode 593
13.5.4 System Reset by $\overline{\text{WDTOVF}}$ Signal 593
13.5.5 Internal Reset in Watchdog Timer Mode 594

Section 14 Serial Communication Interface (SCI) 595
14.1 Overview 595
14.1.1 Features 595
14.1.2 Block Diagram 597
14.1.3 Pin Configuration 598
14.1.4 Register Configuration 599
14.2 Register Descriptions 600
14.2.1 Receive Shift Register (RSR) 600
14.2.2 Receive Data Register (RDR) 600
14.2.3 Transmit Shift Register (TSR) 601
14.2.4 Transmit Data Register (TDR) 601
14.2.5 Serial Mode Register (SMR) 602
14.2.6 Serial Control Register (SCR) 605
14.2.7 Serial Status Register (SSR) 609
14.2.8 Bit Rate Register (BRR) 613
14.2.9 Smart Card Mode Register (SCMR) 621
14.2.10 Module Stop Control Register (MSTPCR) 623
14.3 Operation 624
14.3.1 Overview 624
14.3.2 Operation in Asynchronous Mode 626
14.3.3 Multiprocessor Communication Function 637
14.3.4 Operation in Synchronous Mode 645
14.4 SCI Interrupts 654
14.5 Usage Notes 656

15.1.1 Features ... 665
15.1.2 Block Diagram ... 666
15.1.3 Pin Configuration ... 667
15.1.4 Register Configuration ... 668
15.2 Register Descriptions ... 669
15.2.1 Smart Card Mode Register (SCMR) ... 669
15.2.2 Serial Status Register (SSR) ... 671
15.2.3 Serial Mode Register (SMR) ... 673
15.2.4 Serial Control Register (SCR) ... 675
15.3 Operation ... 676
15.3.1 Overview ... 676
15.3.2 Pin Connections ... 676
15.3.3 Data Format ... 678
15.3.4 Register Settings ... 680
15.3.5 Clock ... 682
15.3.6 Data Transfer Operations ... 684
15.3.7 Operation in GSM Mode ... 692
15.3.8 Operation in Block Transfer Mode ... 693
15.4 Usage Notes ... 694

Section 16 A/D Converter (12 Analog Input Channel Version) ... 699
16.1 Overview ... 699
16.1.1 Features ... 699
16.1.2 Block Diagram ... 700
16.1.3 Pin Configuration ... 701
16.1.4 Register Configuration ... 702
16.2 Register Descriptions ... 703
16.2.1 A/D Data Registers A to D (ADDRA to ADDRD) ... 703
16.2.2 A/D Control/Status Register (ADCSR) ... 704
16.2.3 A/D Control Register (ADCR) ... 706
16.2.4 Module Stop Control Register (MSTPCR) ... 707
16.3 Interface to Bus Master ... 708
16.4 Operation ... 709
16.4.1 Single Mode (SCAN = 0) ... 709
16.4.2 Scan Mode (SCAN = 1) ... 711
16.4.3 Input Sampling and A/D Conversion Time ... 713
16.4.4 External Trigger Input Timing ... 714
16.5 Interrupts ... 715

Section 17 D/A Converter 721
17.1 Overview 721
17.1.1 Features 721
17.1.2 Block Diagram 722
17.1.3 Pin Configuration 723
17.1.4 Register Configuration 723
17.2 Register Descriptions 724
17.2.1 D/A Data Registers 0 to 3 (DADR0 to DADR3) 724
17.2.2 D/A Control Registers 01 and 23 (DACR01, DACR23) 724
17.2.3 Module Stop Control Register (MSTPCR) 726
17.3 Operation 727

Section 18 RAM 729
18.1 Overview 729
18.1.1 Block Diagram 729
18.1.2 Register Configuration 730
18.2 Register Descriptions 730
18.2.1 System Control Register (SYSCR) 730
18.3 Operation 731
18.4 Usage Note 731

Section 19 ROM 733
19.1 Overview 733
19.1.1 Block Diagram 733
19.1.2 Register Configuration 734
19.2 Register Descriptions 734
19.2.1 Mode Control Register (MDCR) 734
19.2.2 Bus Control Register L (BCRL) 735
19.3 Operation 735
19.4 Overview of Flash Memory (H8S/2339 F-ZTAT) 738
19.4.1 Features 738
19.4.2 Overview 739
19.4.3 Flash Memory Operating Modes 740
19.4.4 On-Board Programming Modes 741
19.4.5 Flash Memory Emulation in RAM 743
19.4.6 Differences between Boot Mode and User Program Mode 744
19.4.7 Block Configuration 745
19.4.8 Pin Configuration 746

19.5.1 Flash Memory Control Register 1 (FLMCR1) 748
19.5.2 Flash Memory Control Register 2 (FLMCR2) 751
19.5.3 Erase Block Register 1 (EBR1) 752
19.5.4 Erase Block Registers 2 (EBR2) 752
19.5.5 System Control Register 2 (SYSCR2) 753
19.5.6 RAM Emulation Register (RAMER) 754
19.6 On-Board Programming Modes 756
19.6.1 Boot Mode 757
19.6.2 User Program Mode 761
19.7 Programming/Erasing Flash Memory 763
19.7.1 Program Mode 763
19.7.2 Program-Verify Mode 764
19.7.3 Erase Mode 766
19.7.4 Erase-Verify Mode 766
19.8 Flash Memory Protection 768
19.8.1 Hardware Protection 768
19.8.2 Software Protection 768
19.8.3 Error Protection 769
19.9 Flash Memory Emulation in RAM 771
19.9.1 Emulation in RAM 771
19.9.2 RAM Overlap 772
19.10 Interrupt Handling when Programming/Erasing Flash Memory 773
19.11 Flash Memory PROM Mode 774
19.11.1 PROM Mode Setting 774
19.11.2 Socket Adapters and Memory Map 774
19.11.3 PROM Mode Operation 776
19.11.4 Memory Read Mode 777
19.11.5 Auto-Program Mode 781
19.11.6 Auto-Erase Mode 783
19.11.7 Status Read Mode 784
19.11.8 Status Polling 786
19.11.9 PROM Mode Transition Time 786
19.11.10 Notes on Memory Programming 787
19.12 Flash Memory Programming and Erasing Precautions 787
19.13 Overview of Flash Memory (H8S/2338 F-ZTAT) 789
19.13.1 Features 789
19.13.2 Overview 790
19.13.3 Flash Memory Operating Modes 791

19.13.6 Differences between Boot Mode and User Program Mode 795
19.13.7 Block Configuration 796
19.13.8 Pin Configuration 797
19.13.9 Register Configuration 798
19.14 Register Descriptions 799
19.14.1 Flash Memory Control Register 1 (FLMCR1) 799
19.14.2 Flash Memory Control Register 2 (FLMCR2) 802
19.14.3 Erase Block Register 1 (EBR1) 803
19.14.4 Erase Block Registers 2 (EBR2) 803
19.14.5 System Control Register 2 (SYSCR2) 804
19.14.6 RAM Emulation Register (RAMER) 805
19.15 On-Board Programming Modes 807
19.15.1 Boot Mode 807
19.15.2 User Program Mode 813
19.16 Programming/Erasing Flash Memory 815
19.16.1 Program Mode 815
19.16.2 Program-Verify Mode 816
19.16.3 Erase Mode 818
19.16.4 Erase-Verify Mode 818
19.17 Flash Memory Protection 820
19.17.1 Hardware Protection 820
19.17.2 Software Protection 820
19.17.3 Error Protection 821
19.18 Flash Memory Emulation in RAM 823
19.18.1 Emulation in RAM 823
19.18.2 RAM Overlap 824
19.19 Interrupt Handling when Programming/Erasing Flash Memory 825
19.20 Flash Memory PROM Mode 826
19.20.1 PROM Mode Setting 826
19.20.2 Socket Adapters and Memory Map 827
19.20.3 PROM Mode Operation 829
19.20.4 Memory Read Mode 830
19.20.5 Auto-Program Mode 834
19.20.6 Auto-Erase Mode 836
19.20.7 Status Read Mode 838
19.20.8 Status Polling 839
19.20.9 PROM Mode Transition Time 840
19.20.10 Notes on Memory Programming 841

Section 20 Clock Pulse Generator 847
20.1 Overview 847
20.1.1 Block Diagram 847
20.1.2 Register Configuration 848
20.2 Register Descriptions 848
20.2.1 System Clock Control Register (SCKCR) 848
20.3 Oscillator 850
20.3.1 Connecting a Crystal Resonator 850
20.3.2 External Clock Input 852
20.4 Duty Adjustment Circuit 854
20.5 Medium-Speed Clock Divider 854
20.6 Bus Master Clock Selection Circuit 854

Section 21 Power-Down Modes 855
21.1 Overview 855
21.1.1 Register Configuration 856
21.2 Register Descriptions 857
21.2.1 Standby Control Register (SBYCR) 857
21.2.2 System Clock Control Register (SCKCR) 859
21.2.3 Module Stop Control Register (MSTPCR) 861
21.3 Medium-Speed Mode 861
21.4 Sleep Mode 862
21.5 Module Stop Mode 863
21.5.1 Module Stop Mode 863
21.5.2 Usage Notes 864
21.6 Software Standby Mode 865
21.6.1 Software Standby Mode 865
21.6.2 Clearing Software Standby Mode 865
21.6.3 Setting Oscillation Stabilization Time after Clearing Software Standby Mode 866
21.6.4 Software Standby Mode Application Example 866
21.6.5 Usage Notes 867
21.7 Hardware Standby Mode 868
21.7.1 Hardware Standby Mode 868
21.7.2 Hardware Standby Mode Timing 868
21.8 φ Clock Output Disabling Function 869

and ROMless Version (H8S/2332) 871
22.1.1 Absolute Maximum Ratings 871
22.1.2 DC Characteristics 872
22.1.3 AC Characteristics 874
22.1.4 A/D Conversion Characteristics 898
22.1.5 D/A Conversion Characteristics 899
22.2 Electrical Characteristics of F-ZTAT Version (H8S/2339, H8S/2339E, H8S/2338) 900
22.2.1 Absolute Maximum Ratings 900
22.2.2 DC Characteristics 901
22.2.3 AC Characteristics 904
22.2.4 A/D Conversion Characteristics 909
22.2.5 D/A Conversion Characteristics 909
22.2.6 Flash Memory Characteristics 910
22.3 Usage Note 911

Appendix A Instruction Set 913
A.1 Instruction List 913
A.2 Instruction Codes 937
A.3 Operation Code Map 952
A.4 Number of States Required for Instruction Execution 956
A.5 Bus States during Instruction Execution 970
A.6 Condition Code Modification 984

Appendix B Internal I/O Registers 990
B.1 List of Registers (Address Order) 990
B.2 List of Registers (By Module) 1001
B.3 Functions 1012

Appendix C I/O Port Block Diagrams 1156
C.1 Port 1 1156
C.2 Port 2 1159
C.3 Port 3 1160
C.4 Port 4 1163
C.5 Port 5 1164
C.6 Port 6 1169
C.7 Port 7 1173
C.8 Port 8 1176
C.9 Port 9 1180

C.12 Port C 1186
C.13 Port D 1187
C.14 Port E 1188
C.15 Port F 1189
C.16 Port G 1197

Appendix D Pin States 1201
D.1 Port States in Each Mode 1201

Appendix E Product Lineup 1209

Appendix F Package Dimensions 1210

1.1 Overview

The H8S/2339 Group is a series of microcomputers (MCUs: microcomputer units), built around the H8S/2000 CPU, employing Renesas' proprietary architecture, and equipped with peripheral functions on-chip.

The H8S/2000 CPU has an internal 32-bit architecture, is provided with sixteen 16-bit general registers and a concise, optimized instruction set designed for high-speed operation, and can address a 16-Mbyte linear address space. The instruction set is upward-compatible with H8/300 and H8/300H CPU instructions at the object-code level, facilitating migration from the H8/300, H8/300L, or H8/300H Series.

On-chip peripheral functions required for system configuration include DMA controller (DMAC) and data transfer controller (DTC) bus masters, ROM, RAM, a 16-bit timer pulse unit (TPU), programmable pulse generator (PPG), 8-bit timer, watchdog timer (WDT), serial communication interface (SCI), A/D converter, D/A converter, and I/O ports.

A high-functionality bus controller is also provided, enabling fast and easy connection of DRAM and other kinds of memory.

Single-power-supply flash memory (F-ZTAT™*) and mask ROM versions are available, providing a quick and flexible response to conditions from ramp-up through full-scale volume production, even for applications with frequently changing specifications. ROM is connected to the CPU via a 16-bit data bus, enabling both byte and word data to be accessed in one state. Instruction fetching is thus speeded up, and processing speed increased.

The features of the H8S/2339 Group are shown in table 1.1.

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

Item	Specification
CPU	• General-register machine — Sixteen 16-bit general registers (also usable as sixteen 8-bit registers or eight 32-bit registers) • High-speed operation suitable for realtime control — Maximum clock rate: 25 MHz — High-speed arithmetic operations 8/16/32-bit register-register add/subtract: 40 ns (at 25-MHz operation) 16 × 16-bit register-register multiply: 800 ns (at 25-MHz operation) 32 ÷ 16-bit register-register divide: 800 ns (at 25-MHz operation) • Instruction set suitable for high-speed operation — Sixty-five basic instructions — 8/16/32-bit data transfer, arithmetic, and logic instructions — Unsigned/signed multiply and divide instructions — Powerful bit-manipulation instructions • CPU operating mode — Advanced mode: 16-Mbyte address space
Bus controller	• Address space divided into 8 areas, with bus specifications settable independently for each area • Chip select output possible 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 area • Number of program wait states can be set for each area • Maximum 8-Mbyte DRAM directly connectable (or use of interval timer possible) • External bus release function
DMA controller (DMAC)	• Choice of short address mode or full address mode • 4 channels in short address mode 2 channels in full address mode • Transfer possible in repeat mode, block transfer mode, etc. • Single address mode transfer possible • Can be activated by internal interrupt

controller (DTC)	• Multiple transfers or multiple types of transfer possible for one activation source • Transfer possible in repeat mode, block transfer mode, etc. • Request can be sent to CPU for interrupt that activated DTC
16-bit timer pulse unit (TPU)	• 6-channel 16-bit timer on-chip • Pulse I/O processing capability for up to 16 pins • Automatic 2-phase encoder count capability
Programmable pulse generator (PPG)	• Maximum 16-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 possible
8-bit timer, 2 channels	• 8-bit up-counter (external event count capability) • Two time constant registers • Two-channel connection possible
Watchdog timer	• Watchdog timer or interval timer selectable
Serial communication interface (SCI), 3 channels	• Asynchronous mode or synchronous mode selectable • Multiprocessor communication function • Smart card interface function
A/D converter	• Resolution: 10 bits • Input: 12 channels • 6.7-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 trigger
D/A converter	• Resolution: 8 bits • Output: 4 channels
I/O ports	• 106 input/output pins, 12 input-only pins

- High-speed static RAM

Product Name	ROM	RAM
H8S/2339	384 kbytes	32 kbytes
H8S/2338	256 kbytes	8 kbytes
H8S/2337	128 kbytes	8 kbytes
H8S/2332	—	8 kbytes

Interrupt controller	• Nine external interrupt pins (NMI, $\overline{IRQ}_0$ to $\overline{IRQ}_7$) • 52 internal interrupt sources • Eight priority levels settable
Power-down state	• Medium-speed mode • Sleep mode • Module stop mode • Software standby mode • Hardware standby mode • Variable clock division ratio

Mode	CPU Operating Mode	Description	On-Chip ROM	External Data Bus	
				Initial Value	Maximum Value
1	—	—	—	—	—
2					
3					
4	Advanced	Expanded mode with on-chip ROM disabled	Disabled	16 bits	16 bits
5				8 bits	16 bits
6		Expanded mode with on-chip ROM enabled	Enabled	8 bits	16 bits
7		Single-chip mode		—	—
8	—	—	—	—	—
9					
10	Advanced	Boot mode	Enabled	8 bits	16 bits
11				—	—
12	—	—	—	—	—
13					
14	Advanced	User program mode	Enabled	8 bits	16 bits
15				—	—

- Four MCU operating modes (H8S/2339 F-ZTAT, Mask ROM, and ROMless versions)

Mode	CPU Operating Mode	Description	On-Chip ROM	External Data Bus	
				Initial Value	Maximum Value
1	—	—	—	—	—
2					
3					
4*	Advanced	Expanded mode with on-chip ROM disabled	Disabled	16 bits	16 bits
5*		Expanded mode with on-chip ROM disabled	Disabled	8 bits	16 bits
6		Expanded mode with on-chip ROM enabled	Enabled	8 bits	16 bits
7		Single-chip mode	Enabled	—	—

Note: * Only modes 4 and 5 are provided in the ROMless version.

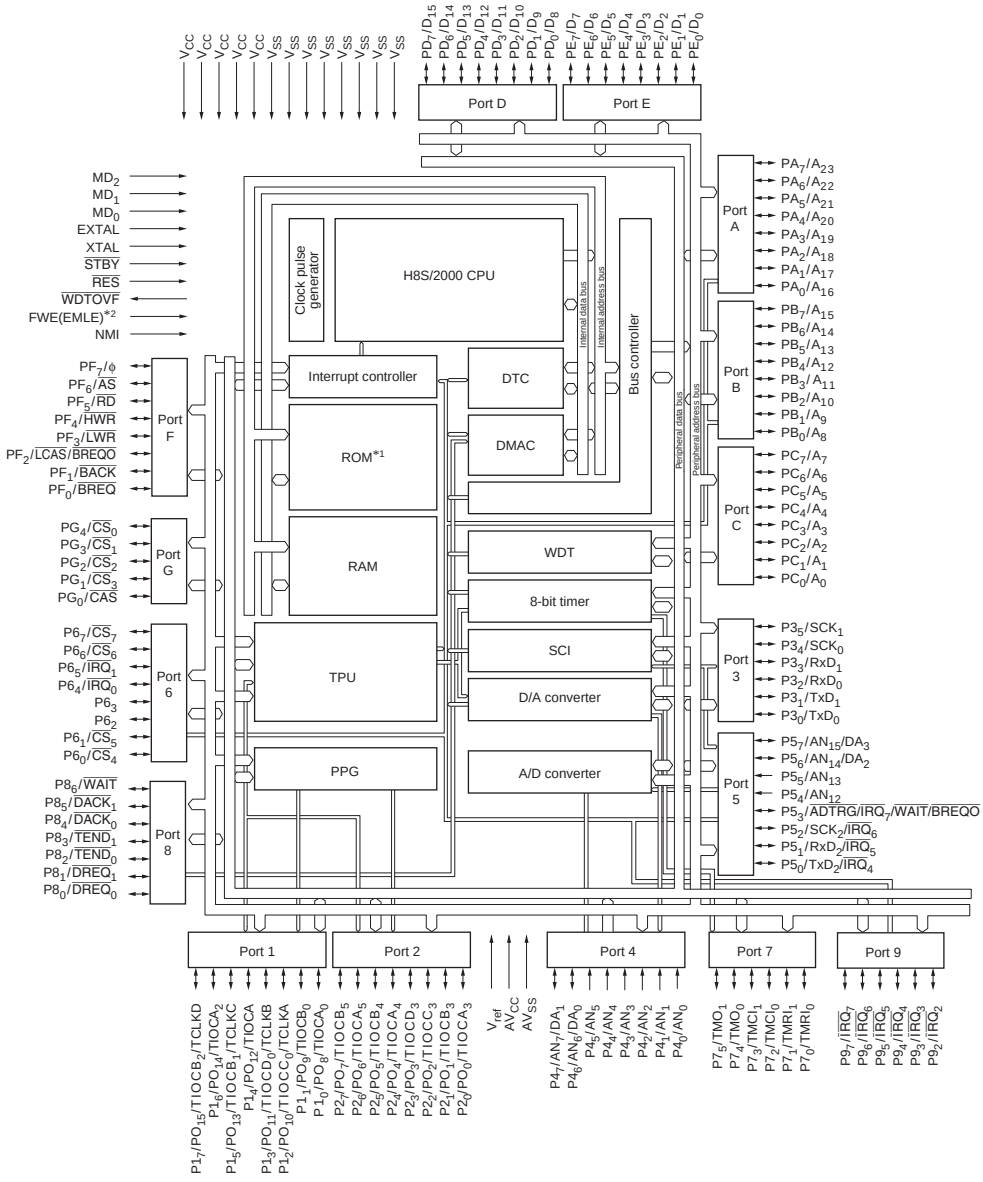
generator

Product lineup			Condition A	Condition B
	Operating power supply voltage		2.7 to 3.6 V	3.0 to 3.6 V
	Operating frequency		2 to 20 MHz	2 to 25 MHz
	Model	HD64F2339E*	—	O
		HD64F2339	—	O
		HD6432338	O	O
		HD64F2338	—	O
		HD6432337	O	O
		HD6412332	O	O

O: Products in the current lineup

Note: * The on-chip debug function can be used with the E10-A emulator (E10-A compatible version). However, some function modules and pin functions are unavailable when the on-chip debug function is in use. Refer to figure 1.2, Pin Arrangement.

Package	• 144-pin plastic QFP (FP-144G)

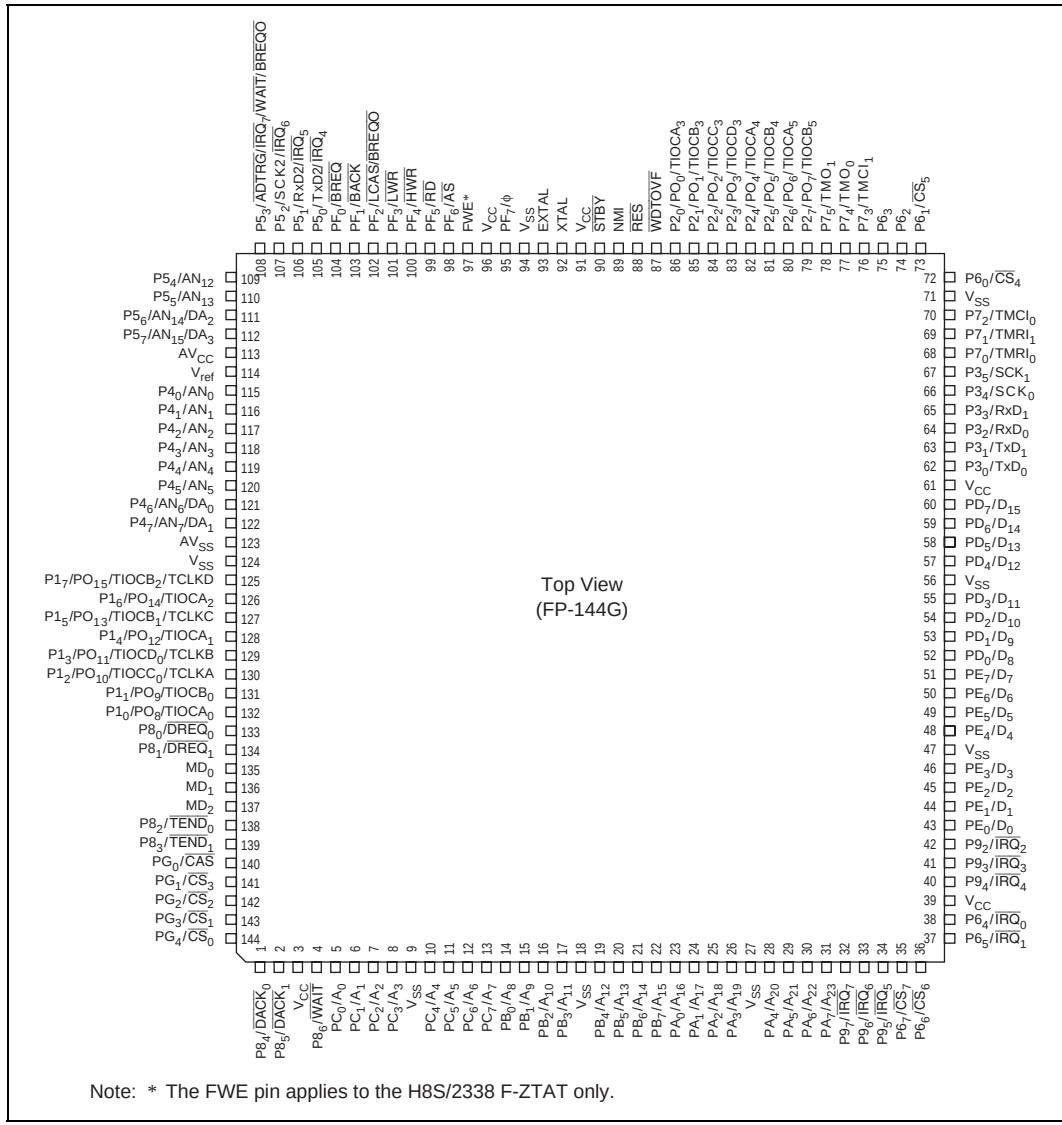


Notes: 1. ROM is not supported in the ROMless version.
2. The FWE pin applies to the H8S/2338 F-ZTAT only.
The EMLE pin applies to the H8S/2339 F-ZTAT only.

Figure 1.1 Internal Block Diagram

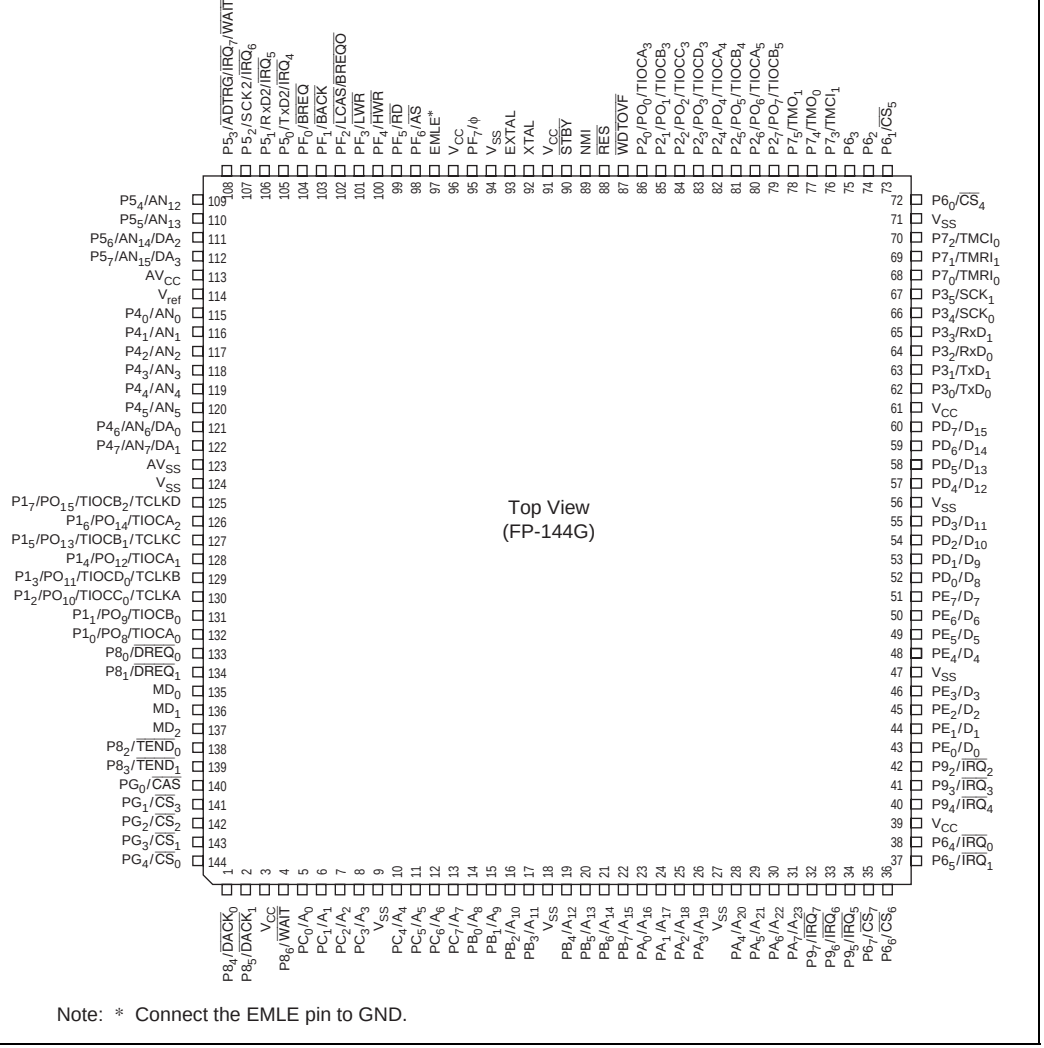
1.3.1 Pin Arrangement

Figures 1.2 and 1.3 show the pin arrangement of the H8S/2339 Group.



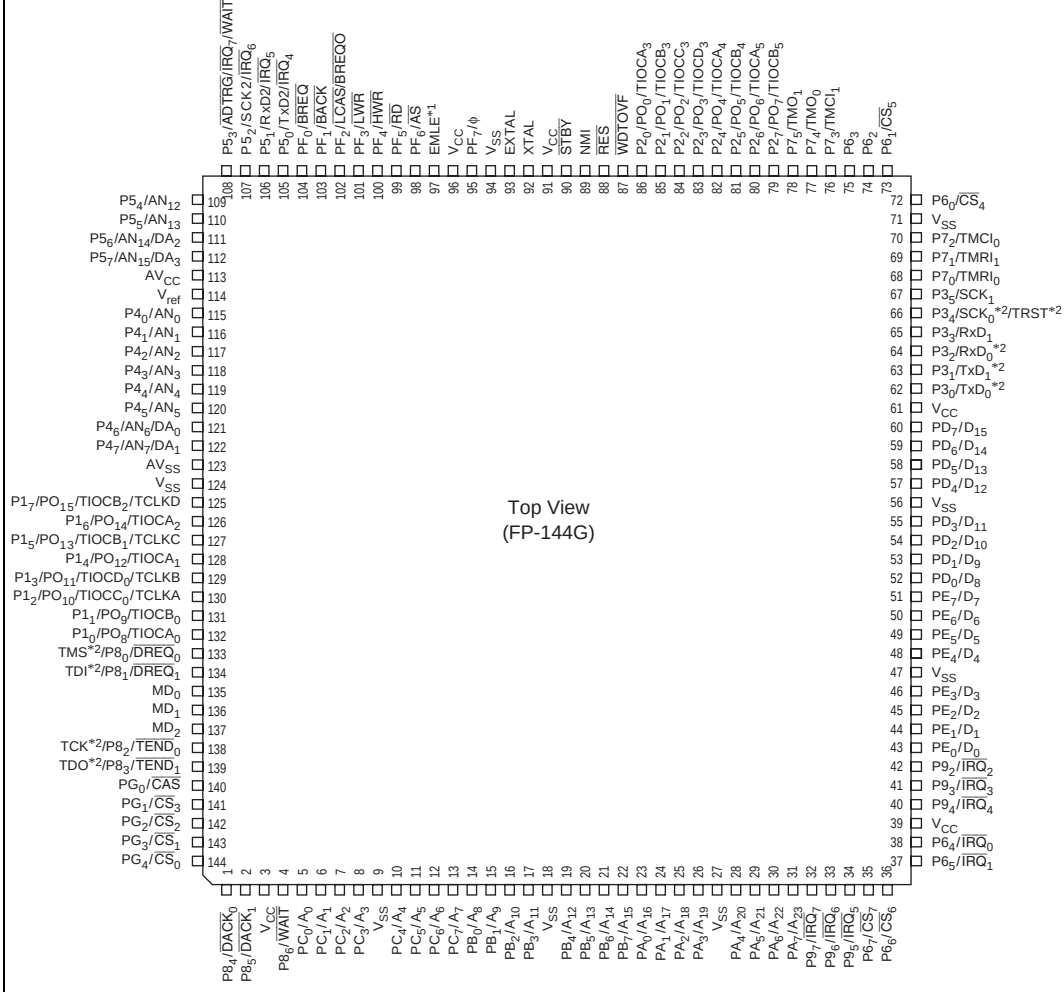
Note: * The FWE pin applies to the H8S/2338 F-ZTAT only.

Figure 1.2 H8S/2338, H8S/2337, H8S/2332 Pin Arrangement (TFP-144G: Top View)



Note: * Connect the EMLE pin to GND.

Figure 1.3 HD64F2339 Pin Arrangement (TFP-144G: Top View)



Notes: 1. The EMLE pin applies to the HD64F2339E only.

2. If an E10-A emulator is used, the TDO, TDI, TDK, TMS and $\overline{\text{TRST}}$ pins are used exclusively for the H-UDI and the functions and function modules associated with these pins are not available. SCI channel 0 is not available. Also, the watchdog timer continues to operate during break states and, if the settings specify that an internal reset is to be performed, a reset is generated if an overflow occurs. Refer to the E10-A Emulator User's Manual for E10-A emulator connection examples.

Figure 1.4 HD64F2339E Pin Arrangement (TFP-144G: Top View)

Table 1.2 Pin Functions in Each Operating Mode

Pin No.	Pin Name				
FP-144G	Mode 4*1	Mode 5*1	Mode 6	Mode 7	Flash Memory Programmer Mode
1	$P8_4/\overline{DACK_0}$	$P8_4/\overline{DACK_0}$	$P8_4/\overline{DACK_0}$	$P8_4/\overline{DACK_0}$	NC
2	$P8_5/\overline{DACK_1}$	$P8_5/\overline{DACK_1}$	$P8_5/\overline{DACK_1}$	$P8_5/\overline{DACK_1}$	NC
3	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}
4	$P8_6/\overline{WAIT}$	$P8_6/\overline{WAIT}$	$P8_6/\overline{WAIT}$	$P8_6$	NC
5	A_0	A_0	PC_0/A_0	PC_0	A_0
6	A_1	A_1	PC_1/A_1	PC_1	A_1
7	A_2	A_2	PC_2/A_2	PC_2	A_2
8	A_3	A_3	PC_3/A_3	PC_3	A_3
9	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
10	A_4	A_4	PC_4/A_4	PC_4	A_4
11	A_5	A_5	PC_5/A_5	PC_5	A_5
12	A_6	A_6	PC_6/A_6	PC_6	A_6
13	A_7	A_7	PC_7/A_7	PC_7	A_7
14	A_8	A_8	PB_0/A_8	PB_0	A_8
15	A_9	A_9	PB_1/A_9	PB_1	A_9
16	A_{10}	A_{10}	PB_2/A_{10}	PB_2	A_{10}
17	A_{11}	A_{11}	PB_3/A_{11}	PB_3	A_{11}
18	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
19	A_{12}	A_{12}	PB_4/A_{12}	PB_4	A_{12}
20	A_{13}	A_{13}	PB_5/A_{13}	PB_5	A_{13}
21	A_{14}	A_{14}	PB_6/A_{14}	PB_6	A_{14}
22	A_{15}	A_{15}	PB_7/A_{15}	PB_7	A_{15}
23	A_{16}	A_{16}	PA_0/A_{16}	PA_0	A_{16}
24	A_{17}	A_{17}	PA_1/A_{17}	PA_1	A_{17}
25	A_{18}	A_{18}	PA_2/A_{18}	PA_2	A_{18}
26	A_{19}	A_{19}	PA_3/A_{19}	PA_3	NC
27	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}

FP-144G	Mode 4*1	Mode 5*1	Mode 6	Mode 7	Programmer Mode
28	A_{20}	A_{20}	PA_4/A_{20}	PA_4	NC
29	PA_5/A_{21}	PA_5/A_{21}	PA_5/A_{21}	PA_5	NC
30	PA_6/A_{22}	PA_6/A_{22}	PA_6/A_{22}	PA_6	NC
31	PA_7/A_{23}	PA_7/A_{23}	PA_7/A_{23}	PA_7	NC
32	$P9_7/\overline{IRQ}_7$	$P9_7/\overline{IRQ}_7$	$P9_7/\overline{IRQ}_7$	$P9_7/\overline{IRQ}_7$	NC
33	$P9_6/\overline{IRQ}_6$	$P9_6/\overline{IRQ}_6$	$P9_6/\overline{IRQ}_6$	$P9_6/\overline{IRQ}_6$	NC
34	$P9_5/\overline{IRQ}_5$	$P9_5/\overline{IRQ}_5$	$P9_5/\overline{IRQ}_5$	$P9_5/\overline{IRQ}_5$	NC
35	$P6_7/\overline{CS}_7$	$P6_7/\overline{CS}_7$	$P6_7/\overline{CS}_7$	$P6_7$	NC
36	$P6_6/\overline{CS}_6$	$P6_6/\overline{CS}_6$	$P6_6/\overline{CS}_6$	$P6_6$	V_{CC}
37	$P6_5/\overline{IRQ}_1$	$P6_5/\overline{IRQ}_1$	$P6_5/\overline{IRQ}_1$	$P6_5/\overline{IRQ}_1$	V_{SS}
38	$P6_4/\overline{IRQ}_0$	$P6_4/\overline{IRQ}_0$	$P6_4/\overline{IRQ}_0$	$P6_4/\overline{IRQ}_0$	V_{SS}
39	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}
40	$P9_4/\overline{IRQ}_4$	$P9_4/\overline{IRQ}_4$	$P9_4/\overline{IRQ}_4$	$P9_4/\overline{IRQ}_4$	NC
41	$P9_3/\overline{IRQ}_3$	$P9_3/\overline{IRQ}_3$	$P9_3/\overline{IRQ}_3$	$P9_3/\overline{IRQ}_3$	NC
42	$P9_2/\overline{IRQ}_2$	$P9_2/\overline{IRQ}_2$	$P9_2/\overline{IRQ}_2$	$P9_2/\overline{IRQ}_2$	NC
43	PE_0/D_0	PE_0/D_0	PE_0/D_0	PE_0	NC
44	PE_1/D_1	PE_1/D_1	PE_1/D_1	PE_1	NC
45	PE_2/D_2	PE_2/D_2	PE_2/D_2	PE_2	NC
46	PE_3/D_3	PE_3/D_3	PE_3/D_3	PE_3	NC
47	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
48	PE_4/D_4	PE_4/D_4	PE_4/D_4	PE_4	NC
49	PE_5/D_5	PE_5/D_5	PE_5/D_5	PE_5	NC
50	PE_6/D_6	PE_6/D_6	PE_6/D_6	PE_6	NC
51	PE_7/D_7	PE_7/D_7	PE_7/D_7	PE_7	NC
52	D_8	D_8	D_8	PD_0	I/O_0
53	D_9	D_9	D_9	PD_1	I/O_1
54	D_{10}	D_{10}	D_{10}	PD_2	I/O_2
55	D_{11}	D_{11}	D_{11}	PD_3	I/O_3
56	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}

FP-144G	Mode 4*1	Mode 5*1	Mode 6	Mode 7	Programmer Mode
57	D_{12}	D_{12}	D_{12}	PD_4	I/O_4
58	D_{13}	D_{13}	D_{13}	PD_5	I/O_5
59	D_{14}	D_{14}	D_{14}	PD_6	I/O_6
60	D_{15}	D_{15}	D_{15}	PD_7	I/O_7
61	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}
62	$P3_0/TxD_0$	$P3_0/TxD_0$	$P3_0/TxD_0$	$P3_0/TxD_0$	NC
63	$P3_1/TxD_1$	$P3_1/TxD_1$	$P3_1/TxD_1$	$P3_1/TxD_1$	NC
64	$P3_2/RxD_0$	$P3_2/RxD_0$	$P3_2/RxD_0$	$P3_2/RxD_0$	V_{CC}
65	$P3_3/RxD_1$	$P3_3/RxD_1$	$P3_3/RxD_1$	$P3_3/RxD_1$	NC
66	$P3_4/SCK_0$	$P3_4/SCK_0$	$P3_4/SCK_0$	$P3_4/SCK_0$	NC
67	$P3_5/SCK_1$	$P3_5/SCK_1$	$P3_5/SCK_1$	$P3_5/SCK_1$	NC
68	$P7_0/TMRI_0$	$P7_0/TMRI_0$	$P7_0/TMRI_0$	$P7_0/TMRI_0$	NC
69	$P7_1/TMRI_1$	$P7_1/TMRI_1$	$P7_1/TMRI_1$	$P7_1/TMRI_1$	NC
70	$P7_2/TMCI_0$	$P7_2/TMCI_0$	$P7_2/TMCI_0$	$P7_2/TMCI_0$	NC
71	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
72	$P6_0/\overline{CS_4}$	$P6_0/\overline{CS_4}$	$P6_0/\overline{CS_4}$	$P6_0$	NC
73	$P6_1/\overline{CS_5}$	$P6_1/\overline{CS_5}$	$P6_1/\overline{CS_5}$	$P6_1$	NC
74	$P6_2$	$P6_2$	$P6_2$	$P6_2$	NC
75	$P6_3$	$P6_3$	$P6_3$	$P6_3$	NC
76	$P7_3/TMCI_1$	$P7_3/TMCI_1$	$P7_3/TMCI_1$	$P7_3/TMCI_1$	NC
77	$P7_4/TMO_0$	$P7_4/TMO_0$	$P7_4/TMO_0$	$P7_4/TMO_0$	NC
78	$P7_5/TMO_1$	$P7_5/TMO_1$	$P7_5/TMO_1$	$P7_5/TMO_1$	NC
79	$P2_7/PO_7/TIOCB_5$	$P2_7/PO_7/TIOCB_5$	$P2_7/PO_7/TIOCB_5$	$P2_7/PO_7/TIOCB_5$	NC
80	$P2_6/PO_6/TIOCA_5$	$P2_6/PO_6/TIOCA_5$	$P2_6/PO_6/TIOCA_5$	$P2_6/PO_6/TIOCA_5$	NC
81	$P2_5/PO_5/TIOCB_4$	$P2_5/PO_5/TIOCB_4$	$P2_5/PO_5/TIOCB_4$	$P2_5/PO_5/TIOCB_4$	V_{SS}
82	$P2_4/PO_4/TIOCA_4$	$P2_4/PO_4/TIOCA_4$	$P2_4/PO_4/TIOCA_4$	$P2_4/PO_4/TIOCA_4$	$\overline{WE}$
83	$P2_3/PO_3/TIOCD_3$	$P2_3/PO_3/TIOCD_3$	$P2_3/PO_3/TIOCD_3$	$P2_3/PO_3/TIOCD_3$	$\overline{CE}$
84	$P2_2/PO_2/TIOCC_3$	$P2_2/PO_2/TIOCC_3$	$P2_2/PO_2/TIOCC_3$	$P2_2/PO_2/TIOCC_3$	$\overline{OE}$
85	$P2_1/PO_1/TIOCB_3$	$P2_1/PO_1/TIOCB_3$	$P2_1/PO_1/TIOCB_3$	$P2_1/PO_1/TIOCB_3$	NC

FP-144G	Mode 4*1	Mode 5*1	Mode 6	Mode 7	Programmer Mode
86	$P2_0/PO_0/TIOCA_3$	$P2_0/PO_0/TIOCA_3$	$P2_0/PO_0/TIOCA_3$	$P2_0/PO_0/TIOCA_3$	NC
87	$\overline{WDTOVF}$	$\overline{WDTOVF}$	$\overline{WDTOVF}$	$\overline{WDTOVF}$	NC
88	$\overline{RES}$	$\overline{RES}$	$\overline{RES}$	$\overline{RES}$	$\overline{RES}$
89	NMI	NMI	NMI	NMI	V_{CC}
90	$\overline{STBY}$	$\overline{STBY}$	$\overline{STBY}$	$\overline{STBY}$	V_{CC}
91	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}
92	XTAL	XTAL	XTAL	XTAL	XTAL
93	EXTAL	EXTAL	EXTAL	EXTAL	EXTAL
94	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
95	PF_7/ϕ	PF_7/ϕ	PF_7/ϕ	PF_7/ϕ	NC
96	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}
97	FWE (EMLE)*2	FWE (EMLE)*2	FWE (EMLE)*2	FWE (EMLE)*2	FWE (EMLE)*2
98	$PF_6/\overline{AS}$	$PF_6/\overline{AS}$	$PF_6/\overline{AS}$	PF_6	NC
99	$\overline{RD}$	$\overline{RD}$	$\overline{RD}$	PF_5	NC
100	$\overline{HWR}$	$\overline{HWR}$	$\overline{HWR}$	PF_4	NC
101	$PF_3/\overline{LWR}$	$PF_3/\overline{LWR}$	$PF_3/\overline{LWR}$	PF_3	NC
102	$PF_2/\overline{LCAS}/\overline{BREQO}$	$PF_2/\overline{LCAS}/\overline{BREQO}$	$PF_2/\overline{LCAS}/\overline{BREQO}$	PF_2	NC
103	$PF_1/\overline{BACK}$	$PF_1/\overline{BACK}$	$PF_1/\overline{BACK}$	PF_1	NC
104	$PF_0/\overline{BREQ}$	$PF_0/\overline{BREQ}$	$PF_0/\overline{BREQ}$	PF_0	NC
105	$P5_0/TxD_2/\overline{IRQ_4}$	$P5_0/TxD_2/\overline{IRQ_4}$	$P5_0/TxD_2/\overline{IRQ_4}$	$P5_0/TxD_2/\overline{IRQ_4}$	NC
106	$P5_1/RxD_2/\overline{IRQ_5}$	$P5_1/RxD_2/\overline{IRQ_5}$	$P5_1/RxD_2/\overline{IRQ_5}$	$P5_1/RxD_2/\overline{IRQ_5}$	NC
107	$P5_2/SCK_2/\overline{IRQ_6}$	$P5_2/SCK_2/\overline{IRQ_6}$	$P5_2/SCK_2/\overline{IRQ_6}$	$P5_2/SCK_2/\overline{IRQ_6}$	NC
108	$P5_3/\overline{ADTRG}/\overline{IRQ_7}/\overline{WAIT}/\overline{BREQO}$	$P5_3/\overline{ADTRG}/\overline{IRQ_7}/\overline{WAIT}/\overline{BREQO}$	$P5_3/\overline{ADTRG}/\overline{IRQ_7}/\overline{WAIT}/\overline{BREQO}$	$P5_3/\overline{ADTRG}/\overline{IRQ_7}$	NC
109	$P5_4/AN_{12}$	$P5_4/AN_{12}$	$P5_4/AN_{12}$	$P5_4/AN_{12}$	NC
110	$P5_5/AN_{13}$	$P5_5/AN_{13}$	$P5_5/AN_{13}$	$P5_5/AN_{13}$	NC
111	$P5_6/AN_{14}/DA_2$	$P5_6/AN_{14}/DA_2$	$P5_6/AN_{14}/DA_2$	$P5_6/AN_{14}/DA_2$	NC
112	$P5_7/AN_{15}/DA_3$	$P5_7/AN_{15}/DA_3$	$P5_7/AN_{15}/DA_3$	$P5_7/AN_{15}/DA_3$	NC

FP-144G	Mode 4*1	Mode 5*1	Mode 6	Mode 7	Programmer Mode
113	AV_{CC}	AV_{CC}	AV_{CC}	AV_{CC}	V_{CC}
114	V_{ref}	V_{ref}	V_{ref}	V_{ref}	V_{CC}
115	$P4_0/AN_0$	$P4_0/AN_0$	$P4_0/AN_0$	$P4_0/AN_0$	NC
116	$P4_1/AN_1$	$P4_1/AN_1$	$P4_1/AN_1$	$P4_1/AN_1$	NC
117	$P4_2/AN_2$	$P4_2/AN_2$	$P4_2/AN_2$	$P4_2/AN_2$	NC
118	$P4_3/AN_3$	$P4_3/AN_3$	$P4_3/AN_3$	$P4_3/AN_3$	NC
119	$P4_4/AN_4$	$P4_4/AN_4$	$P4_4/AN_4$	$P4_4/AN_4$	NC
120	$P4_5/AN_5$	$P4_5/AN_5$	$P4_5/AN_5$	$P4_5/AN_5$	NC
121	$P4_6/AN_6/DA_0$	$P4_6/AN_6/DA_0$	$P4_6/AN_6/DA_0$	$P4_6/AN_6/DA_0$	NC
122	$P4_7/AN_7/DA_1$	$P4_7/AN_7/DA_1$	$P4_7/AN_7/DA_1$	$P4_7/AN_7/DA_1$	NC
123	AV_{SS}	AV_{SS}	AV_{SS}	AV_{SS}	V_{SS}
124	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
125	$P1_7/PO_{15}$/ $TIOCB_2$/TCLKD	$P1_7/PO_{15}$/ $TIOCB_2$/TCLKD	$P1_7/PO_{15}$/ $TIOCB_2$/TCLKD	$P1_7/PO_{15}$/ $TIOCB_2$/TCLKD	NC
126	$P1_6/PO_{14}$/ $TIOCA_2$	$P1_6/PO_{14}$/ $TIOCA_2$	$P1_6/PO_{14}$/ $TIOCA_2$	$P1_6/PO_{14}$/ $TIOCA_2$	NC
127	$P1_5/PO_{13}$/ $TIOCB_1$/TCLKC	$P1_5/PO_{13}$/ $TIOCB_1$/TCLKC	$P1_5/PO_{13}$/ $TIOCB_1$/TCLKC	$P1_5/PO_{13}$/ $TIOCB_1$/TCLKC	NC
128	$P1_4/PO_{12}$/ $TIOCA_1$	$P1_4/PO_{12}$/ $TIOCA_1$	$P1_4/PO_{12}$/ $TIOCA_1$	$P1_4/PO_{12}$/ $TIOCA_1$	NC
129	$P1_3/PO_{11}$/ $TIOCD_0$/TCLKB	$P1_3/PO_{11}$/ $TIOCD_0$/TCLKB	$P1_3/PO_{11}$/ $TIOCD_0$/TCLKB	$P1_3/PO_{11}$/ $TIOCD_0$/TCLKB	NC
130	$P1_2/PO_{10}$/ $TIOCC_0$/TCLKA	$P1_2/PO_{10}$/ $TIOCC_0$/TCLKA	$P1_2/PO_{10}$/ $TIOCC_0$/TCLKA	$P1_2/PO_{10}$/ $TIOCC_0$/TCLKA	NC
131	$P1_1/PO_9/TIOCB_0$	$P1_1/PO_9/TIOCB_0$	$P1_1/PO_9/TIOCB_0$	$P1_1/PO_9/TIOCB_0$	NC
132	$P1_0/PO_8/TIOCA_0$	$P1_0/PO_8/TIOCA_0$	$P1_0/PO_8/TIOCA_0$	$P1_0/PO_8/TIOCA_0$	NC
133	$P8_0/\overline{DREQ}_0$	$P8_0/\overline{DREQ}_0$	$P8_0/\overline{DREQ}_0$	$P8_0/\overline{DREQ}_0$	NC
134	$P8_1/\overline{DREQ}_1$	$P8_1/\overline{DREQ}_1$	$P8_1/\overline{DREQ}_1$	$P8_1/\overline{DREQ}_1$	NC
135	MD_0	MD_0	MD_0	MD_0	V_{SS}
136	MD_1	MD_1	MD_1	MD_1	V_{SS}

FP-144G	Mode 4*1	Mode 5*1	Mode 6	Mode 7	Programmer Mode
137	MD_2	MD_2	MD_2	MD_2	V_{SS}
138	$P8_2/\overline{TEND_0}$	$P8_2/\overline{TEND_0}$	$P8_2/\overline{TEND_0}$	$P8_2/\overline{TEND_0}$	NC
139	$P8_3/\overline{TEND_1}$	$P8_3/\overline{TEND_1}$	$P8_3/\overline{TEND_1}$	$P8_3/\overline{TEND_1}$	NC
140	$PG_0/\overline{CAS}$	$PG_0/\overline{CAS}$	$PG_0/\overline{CAS}$	PG_0	NC
141	$PG_1/\overline{CS_3}$	$PG_1/\overline{CS_3}$	$PG_1/\overline{CS_3}$	PG_1	NC
142	$PG_2/\overline{CS_2}$	$PG_2/\overline{CS_2}$	$PG_2/\overline{CS_2}$	PG_2	NC
143	$PG_3/\overline{CS_1}$	$PG_3/\overline{CS_1}$	$PG_3/\overline{CS_1}$	PG_3	NC
144	$PG_4/\overline{CS_0}$	$PG_4/\overline{CS_0}$	$PG_4/\overline{CS_0}$	PG_4	NC

Notes: 1. Only modes 4 and 5 are provided in the ROMless version.
2. The FWE pin applies to the H8S/2338 F-ZTAT only. The EMLE pin applies to the H8S/2339 F-ZTAT only.

Table 1.3 Pin Functions

Type	Symbol	Pin No. FP-144G	I/O	Name and Function
Power	V_{CC}	3, 39, 61, 91, 96	Input	**Power supply:** For connection to the power supply. All V_{CC} pins should be connected to the system power supply.
	V_{SS}	9, 18, 27, 47, 56, 71, 94, 124	Input	**Ground:** For connection to ground (0 V). All V_{SS} pins should be connected to the system power supply (0 V).
Clock	XTAL	92	Input	Connects to a crystal resonator. See section 20, Clock Pulse Generator, for typical connection diagrams for a crystal resonator and external clock input.
	EXTAL	93	Input	Connects to a crystal resonator. The EXTAL pin can also input an external clock. See section 20, Clock Pulse Generator, for typical connection diagrams for a crystal resonator and external clock input.
	ϕ	95	Output	**System clock:** Supplies the system clock to an external device.

Operating mode control	MD_2 to MD_0	137 to 135	Input	**Mode pins:** These pins set the operating mode. The relation between the settings of pins MD_2 to MD_0 and the operating mode is shown below. These pins should not be changed while the chip is operating.

H8S/2338 F-ZTAT:

FWE	MD_2	MD_1	MD_0	Operating Mode
0	0	0	1	—
		1	0	—
			1	—
	1	0	0	Mode 4
			1	Mode 5
		1	0	Mode 6
			1	Mode 7
1	0	0	0	—
			1	—
		1	0	Mode 10
			1	Mode 11
	1	0	0	—
			1	—
		1	0	Mode 14
			1	Mode 15

Operating mode control	MD_2 to MD_0	137 to 135	Input	H8S/2339 F-ZTAT, Mask ROM, and ROMless versions:

MD_2	MD_1	MD_0	Operating Mode
0	0	1	—
	1	0	—
		1	—
1	0	0	Mode 4
		1	Mode 5
	1	0	Mode 6*
		1	Mode 7*

Note: * Modes 6 and 7 are not provided in the ROMless version.

System control	$\overline{RES}$	88	Input	**Reset input:** When this pin is driven low, the chip is reset.
	$\overline{STBY}$	90	Input	**Standby:** When this pin is driven low, a transition is made to hardware standby mode.
	$\overline{BREQ}$	104	Input	**Bus request:** Used by an external bus master to issue a bus request to the chip.
	$\overline{BREQO}$	102,108	Output	**Bus request output:** The external bus request signal used when an internal bus master accesses external space in the external bus-released state.
	$\overline{BACK}$	103	Output	**Bus request acknowledge:** Indicates that the bus has been released to an external bus master.
	FWE*1	97	Input	**Flash write enable:** Enables/disables flash memory programming. In the mask ROM version and ROMless version, connect this pin to ground.
	EMLE*2	97	Input	**Emulator enable:** For connection to ground (0 V).

Interrupts	NMI	89	Input	**Nonmaskable interrupt:** Requests a nonmaskable interrupt. When this pin is not used, it should be fixed high.
	$\overline{IRQ_7}$ to $\overline{IRQ_0}$	32 to 34, 40 to 42, 37, 38 108 to 105	Input	**Interrupt request 7 to 0:** These pins request a maskable interrupt.
Address bus	A_{23} to A_0	31 to 28, 26 to 19, 17 to 10, 8 to 5	Output	**Address bus:** These pins output an address.
Data bus	D_{15} to D_0	60 to 57, 55 to 48, 46 to 43	I/O	**Data bus:** These pins constitute a bidirectional data bus.
Bus control	$\overline{CS_7}$ to $\overline{CS_0}$	35, 36, 61, 60, 141 to 144	Output	**Chip select:** Signals for selecting areas 7 to 0.
	$\overline{AS}$	98	Output	**Address strobe:** When this pin is low, it indicates that address output on the address bus is enabled.
	$\overline{RD}$	99	Output	**Read:** When this pin is low, it indicates that the external address space can be read.
	$\overline{HWR}$	100	Output	**High write/write enable:** A strobe signal that writes to external space and indicates that the upper half (D_{15} to D_8) of the data bus is enabled. The 2-CAS type DRAM write enable signal.
	$\overline{LWR}$	101	Output	**Low write:** A strobe signal that writes to external space and indicates that the lower half (D_7 to D_0) of the data bus is enabled.
	$\overline{CAS}$	140	Output	**Upper column address strobe/column address strobe:** The 2-CAS type DRAM upper column address strobe signal.
	$\overline{LCAS}$	102	Output	**Lower column address strobe:** The 2-CAS type DRAM lower column address strobe signal.

Bus control	$\overline{WAIT}$	4,108	Input	**Wait:** Requests insertion of a wait state in the bus cycle when accessing external 3-state address space.
DMA controller (DMAC)	$\overline{DREQ}_1$, $\overline{DREQ}_0$	134,133	Input	**DMA request 1 and 0:** These pins request DMAC activation.
	$\overline{TEND}_1$, $\overline{TEND}_0$	139,138	Output	**DMA transfer end 1 and 0:** These pins indicate the end of DMAC data transfer.
	$\overline{DACK}_1$, $\overline{DACK}_0$	2,1	Output	**DMA transfer acknowledge 1 and 0:** These are the DMAC single address transfer acknowledge pins.
16-bit timer pulse unit (TPU)	TCLKD to TCLKA	125, 127, 129, 130	Input	**Clock input D to A:** These pins input an external clock.
	$TIOCA_0$, $TIOCB_0$, $TIOCC_0$, $TIOCD_0$	132 to 129	I/O	**Input capture/output compare match A0 to D0:** The TGR0A to TGR0D input capture input or output compare output, or PWM output pins.
	$TIOCA_1$, $TIOCB_1$	128, 127	I/O	**Input capture/output compare match A1 and B1:** The TGR1A and TGR1B input capture input or output compare output, or PWM output pins.
	$TIOCA_2$, $TIOCB_2$	126, 125	I/O	**Input capture/output compare match A2 and B2:** The TGR2A and TGR2B input capture input or output compare output, or PWM output pins.
	$TIOCA_3$, $TIOCB_3$, $TIOCC_3$, $TIOCD_3$	86 to 83	I/O	**Input capture/output compare match A3 to D3:** The TGR3A to TGR3D input capture input or output compare output, or PWM output pins.
	$TIOCA_4$, $TIOCB_4$	82, 81	I/O	**Input capture/output compare match A4 and B4:** The TGR4A and TGR4B input capture input or output compare output, or PWM output pins.
	$TIOCA_5$, $TIOCB_5$	80, 79	I/O	**Input capture/output compare match A5 and B5:** The TGR5A and TGR5B input capture input or output compare output, or PWM output pins.

Programmable pulse generator (PPG)	PO_{15} to PO_0	125 to 132, 79 to 86	Output	**Pulse output 15 to 0:** Pulse output pins.
8-bit timer	TMO_0, TMO_1	77, 78	Output	**Compare match output:** The compare match output pins.
	$TMCI_0$, $TMCI_1$	70, 76	Input	**Counter external clock input:** Input pins for the external clock input to the counter.
	$TMRI_0$, $TMRI_1$	68, 69	Input	**Counter external reset input:** The counter reset input pins.
Watchdog timer (WDT)	$\overline{WDTOVF}$	87	Output	**Watchdog timer overflow:** The counter overflow signal output pin in watchdog timer mode.
Serial communication interface (SCI)/ smart card interface	TxD_2, TxD_1, TxD_0	105, 63, 62	Output	**Transmit data (channel 0, 1, 2):** Data output pins.
	RxD_2, RxD_1, RxD_0	106, 65, 64	Input	**Receive data (channel 0, 1, 2):** Data input pins.
	SCK_2, SCK_1, SCK_0	107, 67, 66	I/O	**Serial clock (channel 0, 1, 2):** Clock I/O pins.
A/D converter	AN_{15} to AN_{12}, AN_7 to AN_0	112 to 109, 122 to 115	Input	**Analog 15 to 12, and 7 to 0:** Analog input pins.
	$\overline{ADTRG}$	108	Input	**A/D conversion external trigger input:** Pin for input of an external trigger to start A/D conversion.
D/A converter	DA_3, DA_2, DA_1, DA_0	112, 111, 122, 121	Output	**Analog output:** D/A converter analog output pins.

A/D converter and D/A converter	AV_{CC}	113	Input	This is the power supply pin for the A/D converter and D/A converter. When the A/D converter and D/A converter are not used, this pin should be connected to the system power supply (+3 V).
	AV_{SS}	123	Input	This is the ground pin for the A/D converter and D/A converter. This pin should be connected to the system power supply (0 V).
	V_{ref}	114	Input	This is the reference voltage input pin for the A/D converter and D/A converter. When the A/D converter and D/A converter are not used, this pin should be connected to the system power supply (+3 V).
I/O ports	$P1_7$ to $P1_0$	125 to 132	I/O	**Port 1:** An 8-bit I/O port. Input or output can be designated for each bit by means of the port 1 data direction register (P1DDR).
	$P2_7$ to $P2_0$	79 to 86	I/O	**Port 2:** An 8-bit I/O port. Input or output can be designated for each bit by means of the port 2 data direction register (P2DDR).
	$P3_5$ to $P3_0$	67 to 62	I/O	**Port 3:** A 6-bit I/O port. Input or output can be designated for each bit by means of the port 3 data direction register (P3DDR).
	$P4_7$ to $P4_0$	122 to 115	Input	**Port 4:** An 8-bit input port.
	$P5_7$ to $P5_0$	112 to 109, 108 to 105	Input I/O	**Port 5:** A 4-bit input port and a 4-bit I/O port. For $P5_3$ to $P5_0$, input or output can be designated for each bit by means of the port 5 data direction register (P5DDR).
	$P6_7$ to $P6_0$	35 to 38, 75 to 72	I/O	**Port 6:** An 8-bit I/O port. Input or output can be designated for each bit by means of the port 6 data direction register (P6DDR).
	$P7_5$ to $P7_0$	78 to 76, 70 to 68	I/O	**Port 7:** A 6-bit I/O port. Input or output can be designated for each bit by means of the port 7 data direction register (P7DDR).
	$P8_6$ to $P8_0$	4, 2, 1, 139, 138, 134, 133	I/O	**Port 8:** A 7-bit I/O port. Input or output can be designated for each bit by means of the port 8 data direction register (P8DDR).

I/O ports	$P9_7$ to $P9_2$	32 to 34, 40 to 42	I/O	**Port 9:** A 6-bit I/O port. Input or output can be designated for each bit by means of the port 9 data direction register (P9DDR).
	PA_7 to PA_0	31 to 28, 26 to 23	I/O	**Port A:** An 8-bit I/O port. Input or output can be designated for each bit by means of the port A data direction register (PADDR).
	PB_7 to PB_0	22 to 19, 17 to 14	I/O	**Port B:** An 8-bit I/O port. Input or output can be designated for each bit by means of the port B data direction register (PBDDR).
	PC_7 to PC_0	13 to 10, 8 to 5	I/O	**Port C:** An 8-bit I/O port. Input or output can be designated for each bit by means of the port C data direction register (PCDDR).
	PD_7 to PD_0	60 to 57, 55 to 52	I/O	**Port D:** An 8-bit I/O port. Input or output can be designated for each bit by means of the port D data direction register (PDDDR).
	PE_7 to PE_0	51 to 48, 46 to 43	I/O	**Port E:** An 8-bit I/O port. Input or output can be designated for each bit by means of the port E data direction register (PEDDR).
	PF_7 to PF_0	95, 98 to 104	I/O	**Port F:** An 8-bit I/O port. Input or output can be designated for each bit by means of the port F data direction register (PFDDR).
	PG_4 to PG_0	144 to 140	I/O	**Port G:** A 5-bit I/O port. Input or output can be designated for each bit by means of the port G data direction register (PGDDR).

Notes: 1. Applies to the H8S/2338 F-ZTAT only.
2. Applies to the H8S/2339 F-ZTAT only.

2.1 Overview

The H8S/2000 CPU is a high-speed central processing unit with an internal 32-bit architecture that is upward-compatible with the H8/300 and H8/300H CPUs. The H8S/2000 CPU has sixteen 16-bit general registers, can address a 16-Mbyte (architecturally 4-Gbyte) linear address space, and is ideal for realtime control.

2.1.1 Features

The H8S/2000 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 32-bit registers)
- Sixty-five basic instructions
  - 8/16/32-bit arithmetic and logic instructions
  - Multiply and divide instructions
  - Powerful bit-manipulation instructions
- Eight addressing modes
  - Register direct [Rn]
  - Register indirect [@ERn]
  - Register indirect with displacement [@(d:16,ERn) or @(d:32,ERn)]
  - Register indirect with post-increment or pre-decrement [@ERn+ or @–ERn]
  - Absolute address [@aa:8, @aa:16, @aa:24, or @aa:32]
  - Immediate [#xx:8, #xx:16, or #xx:32]
  - Program-counter relative [@(d:8,PC) or @(d:16,PC)]
  - Memory indirect [@@aa:8]
- 16-Mbyte address space
  - Program: 16 Mbytes
  - Data: 16 Mbytes (4 Gbytes architecturally)

— Maximum clock rate: 25 MHz
— 8/16/32-bit register-register add/subtract: 40 ns
— 8 × 8-bit register-register multiply: 480 ns
— 16 ÷ 8-bit register-register divide: 480 ns
— 16 × 16-bit register-register multiply: 800 ns
— 32 ÷ 16-bit register-register divide: 800 ns

- CPU operating mode
  — Advanced mode
- 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 below.

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

		Internal Operation	
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

There are also differences in the address space, CCR and EXR functions, power-down state, etc., depending on the product.

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

- More general registers and control registers
  - Eight 16-bit expanded registers, and one 8-bit control register, have been added.
- Expanded address space
  - Advanced mode supports a maximum 16-Mbyte address space.
- Enhanced addressing
  - The addressing modes have been enhanced to make effective use of the 16-Mbyte address space.
- Enhanced instructions
  - Addressing modes of bit-manipulation instructions have been enhanced.
  - Signed multiply and divide instructions have been added.
  - 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/2000 CPU has the following enhancements.

- Additional control register
  - One 8-bit control register has been added.
- Enhanced instructions
  - Addressing modes of bit-manipulation instructions have been enhanced.
  - Two-bit shift 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.

The H8S/2339 Group CPU has advanced operating mode. Advanced mode supports a maximum 16-Mbyte total address space (architecturally a maximum 16-Mbyte program area and a maximum of 4 Gbytes for program and data areas combined). The mode is selected by the mode pins of the microcontroller.

Advanced Mode

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

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

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

bits, the upper 8 bits are ignored and a branch address is stored in the lower 24 bits (figure 2.1). For details of the exception vector table, see section 4, Exception Handling.

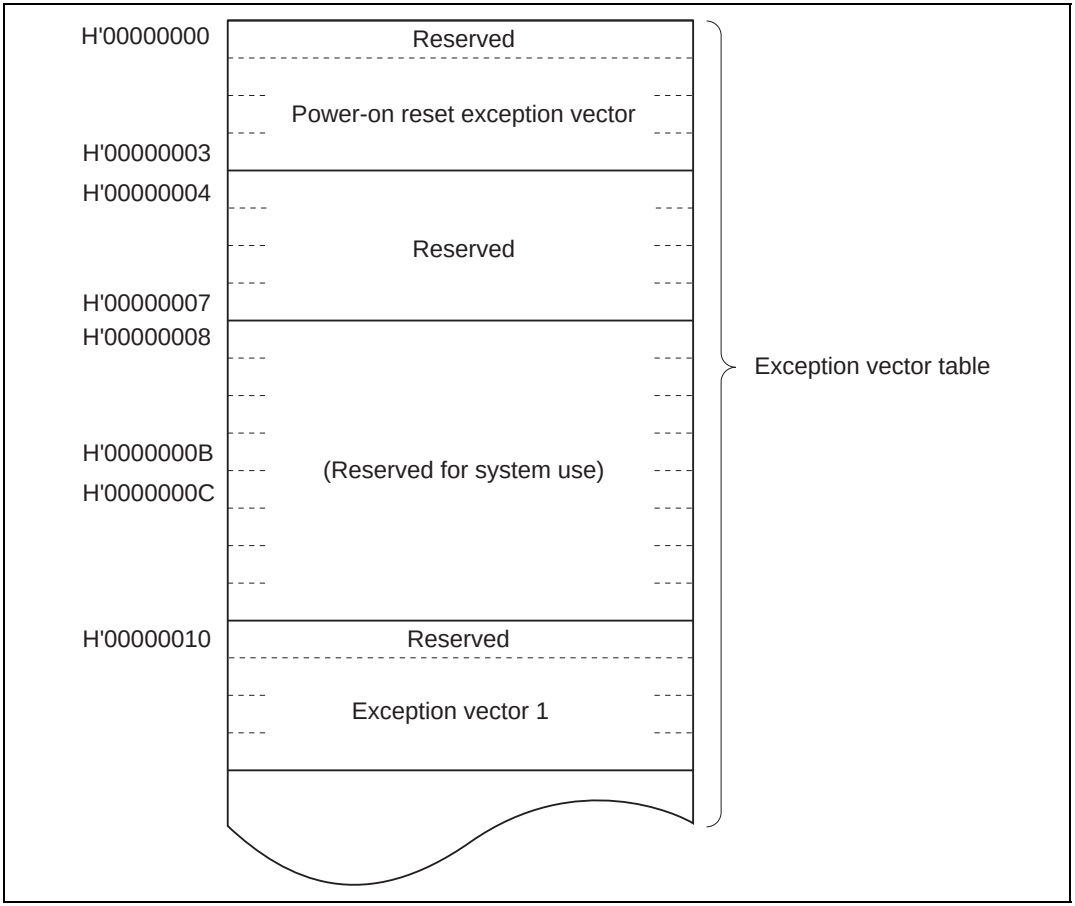


Figure 2.1 Exception Vector Table (Advanced Mode)

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

are pushed onto the stack in exception handling, they are stored as shown in figure 2.2. When EXR is invalid, it is not pushed onto the stack. For details, see section 4, Exception Handling.

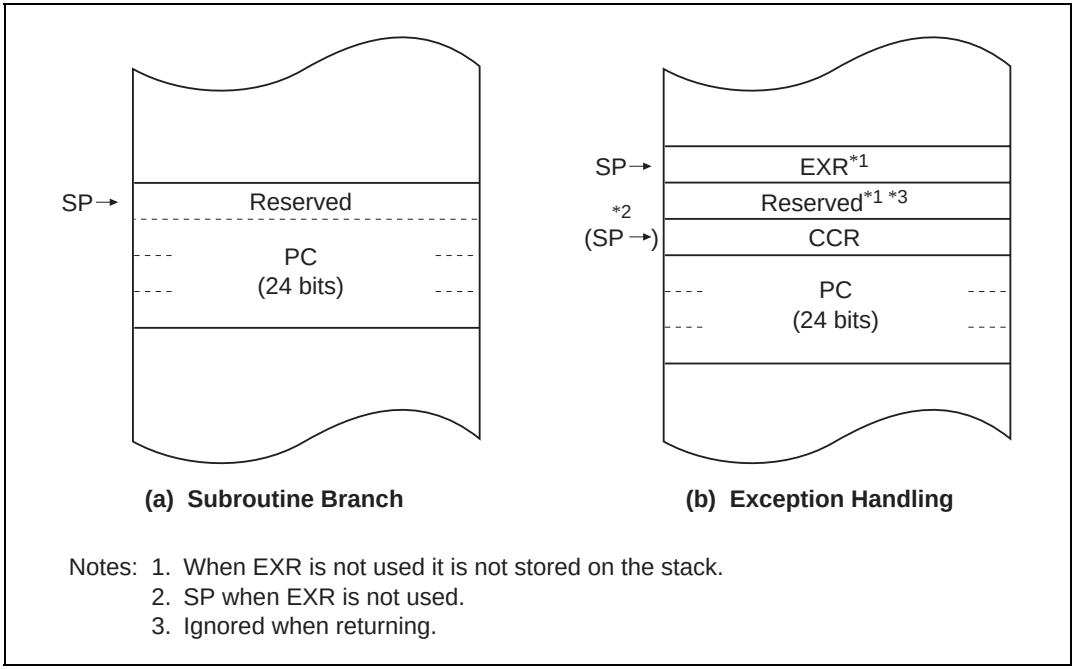


Figure 2.2 Stack Structure in Advanced Mode

Figure 2.3 shows a memory map of the H8S/2000 CPU. The H8S/2000 CPU provides linear access to a maximum 16-Mbyte (architecturally 4-Gbyte) address space in advanced mode.

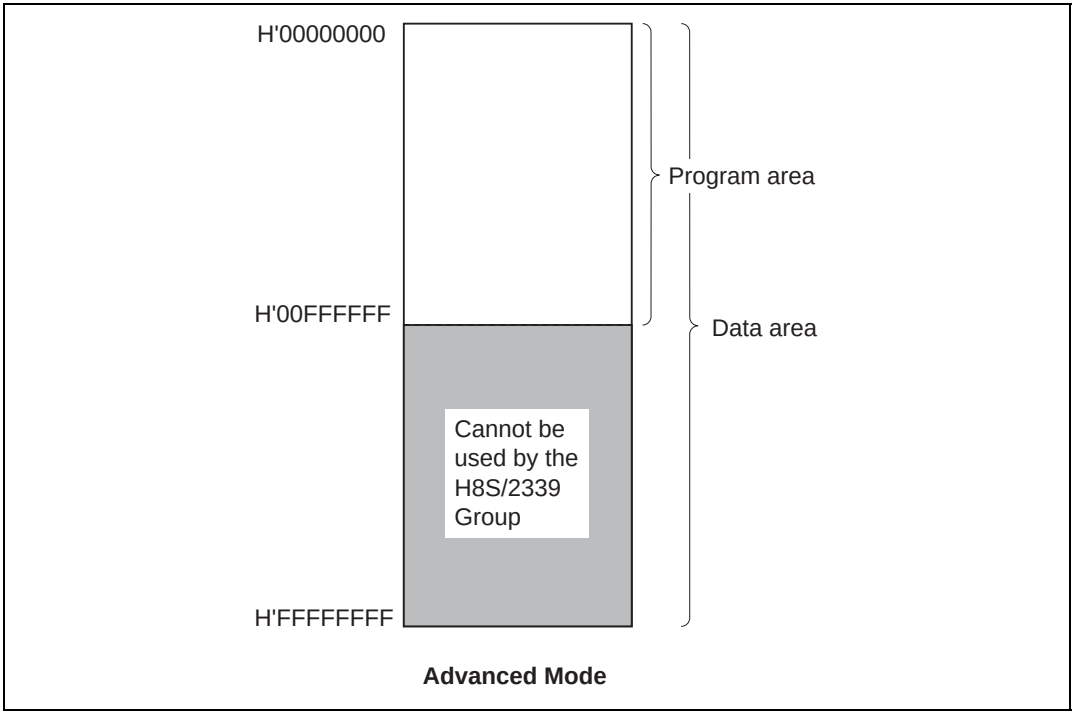


Figure 2.3 Memory Map

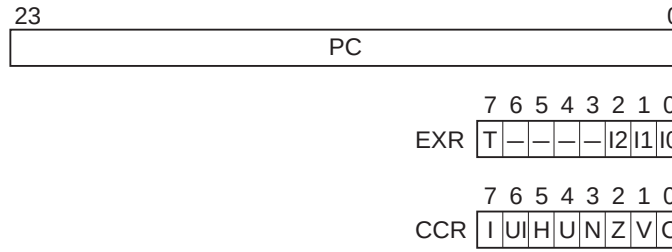
2.4.1 Overview

The CPU has the internal registers shown in figure 2.4. There are two types of registers: general registers and control registers.

General Registers (Rn) and Extended Registers (En)

	15 07	07 0	
ER0	E0	R0H	R0L
ER1	E1	R1H	R1L
ER2	E2	R2H	R2L
ER3	E3	R3H	R3L
ER4	E4	R4H	R4L
ER5	E5	R5H	R5L
ER6	E6	R6H	R6L
ER7 (SP)	E7	R7H	R7L

Control Registers (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		
UI:	User bit or interrupt mask bit*		

Note: * In the H8S/2339 Group, this bit cannot be used as an interrupt mask.

Figure 2.4 CPU Registers

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

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

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

Figure 2.5 illustrates the usage of the general registers. The usage of each register can be selected independently.

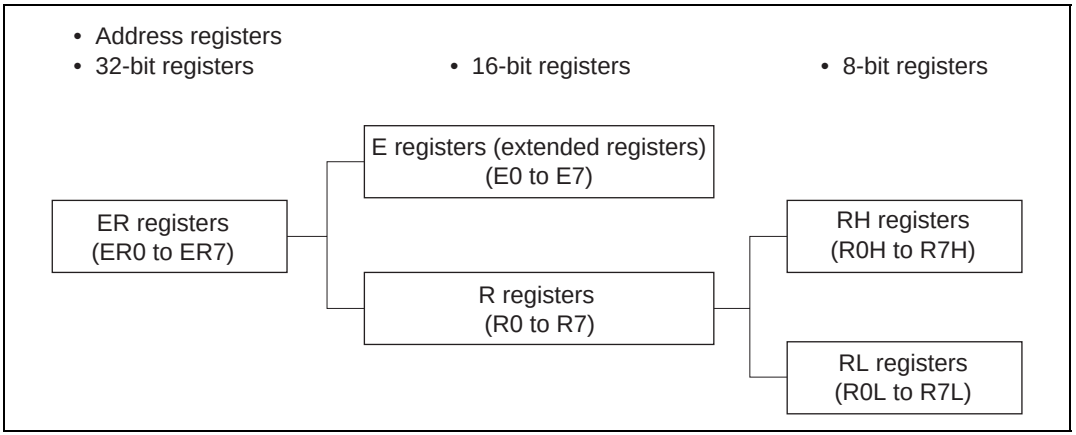


Figure 2.5 Usage of General Registers

General register ER7 has the function of stack pointer (SP) in addition to its general-register function, and is used implicitly in exception handling and subroutine calls. Figure 2.6 shows the stack.

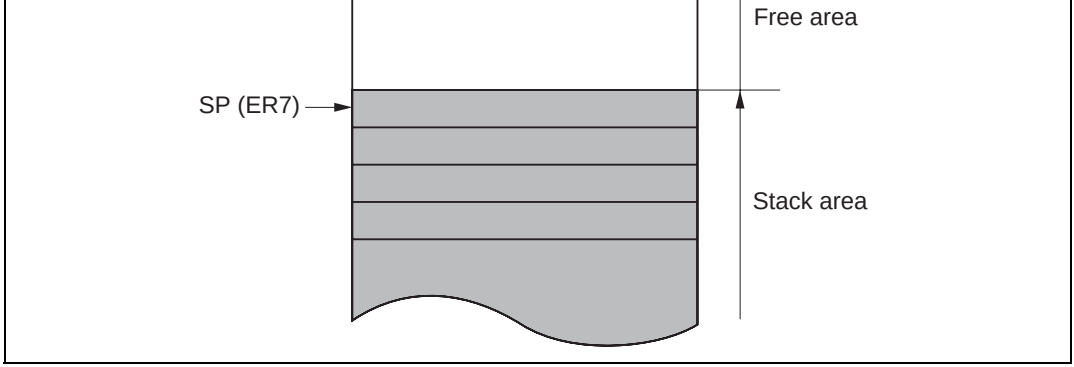


Figure 2.6 Stack

2.4.3 Control Registers

The control registers are the 24-bit program counter (PC), 8-bit extended control register (EXR), and 8-bit condition-code register (CCR).

(1) Program Counter (PC)

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

(2) 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 executed in sequence. When this bit is set to 1, a trace exception is generated each time an instruction is executed.

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

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

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

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

Bit 7—Interrupt Mask Bit (I): Masks interrupts other than NMI when set to 1. (NMI is accepted regardless of the I bit setting.) The I bit is set to 1 by hardware at the start of an exception-handling sequence. For details, refer to section 5, Interrupt Controller.

Bit 6—User Bit or Interrupt Mask Bit (UI): Can be written and read by software using the LDC, STC, ANDC, ORC, and XORC instructions. With the H8S/2339 Group, this bit cannot be used as an interrupt mask bit.

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

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

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

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

Bit 1—Overflow Flag (V): Set to 1 when an arithmetic overflow occurs, and cleared to 0 at other times.

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

- Add instructions, to indicate a carry
- Subtract instructions, to indicate a borrow
- Shift and rotate instructions, to 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 instruction on the flag bits, refer to appendix A.1, Instruction List.

(Bcc) instructions.

2.4.4 Initial Register Values

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

2.5 Data Formats

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

Figure 2.7 shows the data formats in general registers.

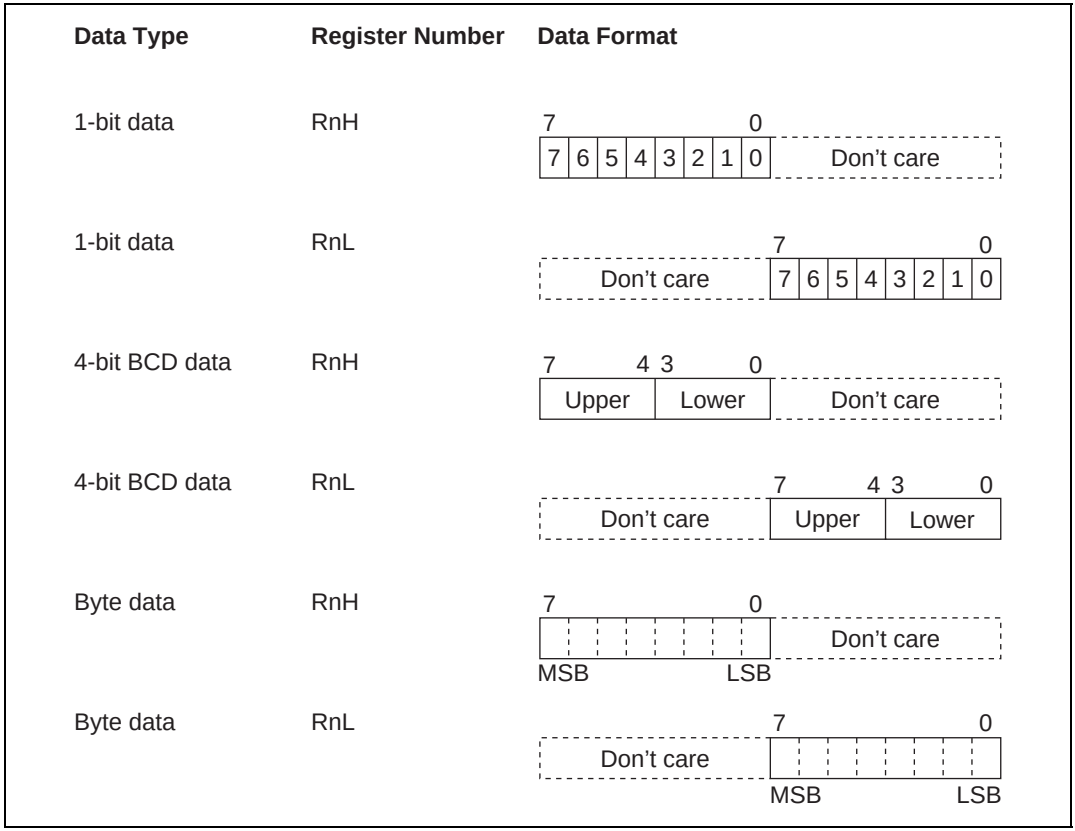
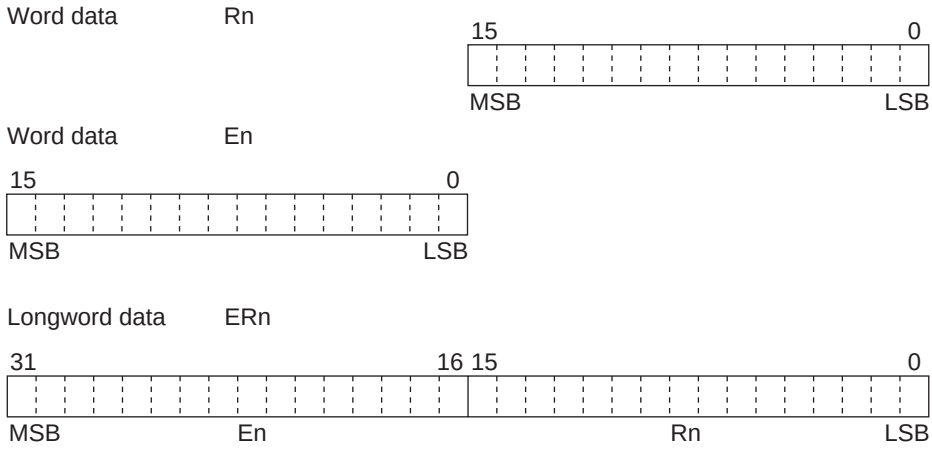


Figure 2.7 General Register Data Formats



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 2.7 General Register Data Formats (cont)

Figure 2.8 shows the data formats in memory. The CPU can access word data and longword data in memory, but word or longword data must begin at an even address. If an attempt is made to access word or longword data at an odd address, no address error occurs but the least significant bit of the address is regarded as 0, so the access starts at the preceding address. This also applies to instruction fetches.

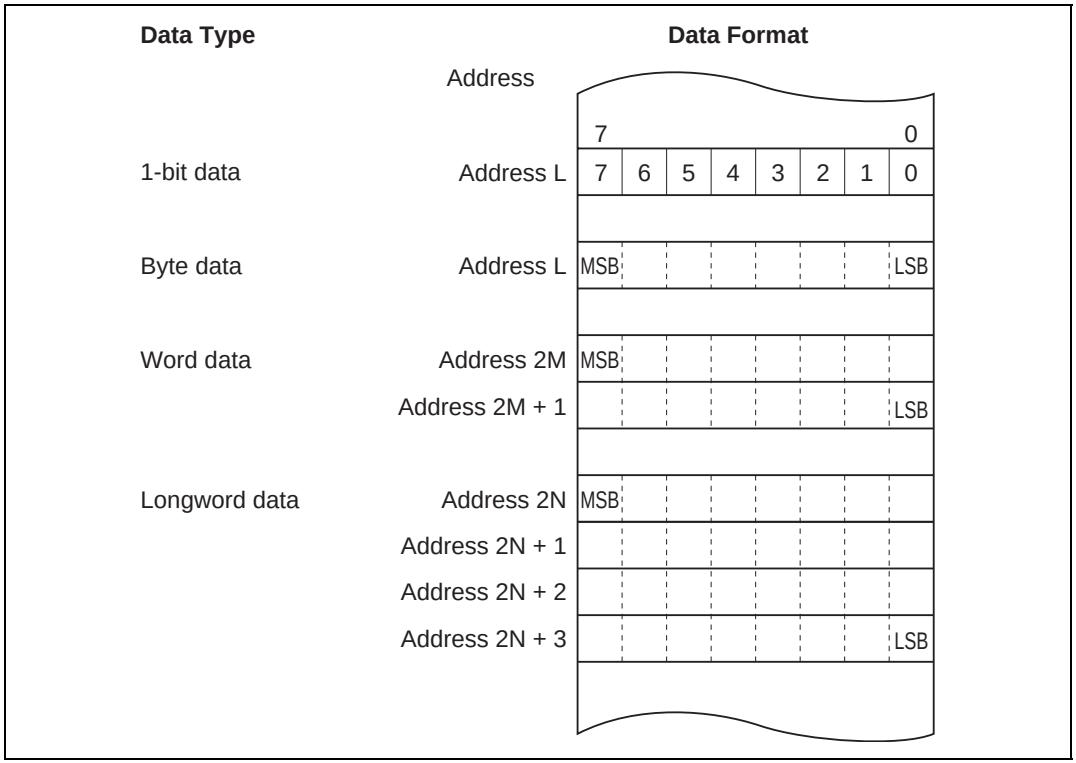


Figure 2.8 Memory Data Formats

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

2.6.1 Overview

The H8S/2000 CPU has 65 types of instructions. The instructions are classified by function in table 2.1.

Table 2.1 Instruction Classification

Function	Instructions	Size	Types
Data transfer	MOV	BWL	5
	POP*1, PUSH*1	WL	
	LDM, STM	L	
	MOVFPE, MOVTPE*3	B	
Arithmetic operations	ADD, SUB, CMP, NEG	BWL	19
	ADDX, SUBX, DAA, DAS	B	
	INC, DEC	BWL	
	ADDS, SUBS	L	
	MULXU, DIVXU, MULXS, DIVXS	BW	
	EXTU, EXTS	WL	
	TAS*4	B	
Logic operations	AND, OR, XOR, NOT	BWL	4
Shift	SHAL, SHAR, SHLL, SHLR, ROTL, ROTR, ROTXL, ROTXR	BWL	8
Bit manipulation	BSET, BCLR, BNOT, BTST, BLD, BILD, BST, BIST, BAND, BIAND, BOR, BIOR, BXOR, BIXOR	B	14
Branch	Bcc*2, JMP, BSR, JSR, RTS	—	5
System control	TRAPA, RTE, SLEEP, LDC, STC, ANDC, ORC, XORC, NOP	—	9
Block data transfer	EEPMOV	—	1
			Total: 65

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 Rn, @-SP. POP.L ERn and PUSH.L ERn are identical to MOV.L @SP+, ERn and MOV.L ERn, @-SP.
2. Bcc is the general name for conditional branch instructions.
3. Cannot be used in the H8S/2339 Group.
4. Only register ER0, ER1, ER4, or ER5 should be used when using the TAS instruction.

Table 2.2 indicates the combinations of instructions and addressing modes that the H8S/2600 CPU can use.

Table 2.2 Combinations of Instructions and Addressing Modes

Function	Instruction	Addressing Modes													
		#xx	Rn	@ERn	@(d:16,ERn)	@(d:32,ERn)	@–ERn/@ERn+	@aa:8	@aa:16	@aa:24	@aa:32	@(d:8,PC)	@(d:16,PC)	@@aa:8	|
Data transfer	MOV	BWL	BWL	BWL	BWL	BWL	BWL	B	BWL	—	BWL	—	—	—	—
	POP, PUSH	—	—	—	—	—	—	—	—	—	—	—	—	—	WL
	LDM, STM	—	—	—	—	—	—	—	—	—	—	—	—	—	L
	MOVFPE, 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	—	—	—	—	—	—	—	—	—	—	—
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		—	—	—	—	—	—	—	—	—	—	—	—	—	BW

Legend:
B: Byte
W: Word
L: Longword

Notes: 1. Cannot be used in the H8S/2339 Group.
2. Only register ER0, ER1, ER4, or ER5 should be used when using the TAS instruction.

Table 2.3 summarizes the instructions in each functional category. The notation used in table 2.3 is defined below.

Operation Notation

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

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

Type	Instruction	Size	Function
Data transfer	MOV	B/W/L	(EAs) → Rd, Rs → (Ead) Moves data between two general registers or between a general register and memory, or moves immediate data to a general register.
	MOVFPE	B	Cannot be used in the H8S/2339 Group.
	MOVTPE	B	Cannot be used in the H8S/2339 Group.
	POP	W/L	@SP+ → Rn Pops a register from the stack. POP.W Rn is identical to MOV.W @SP+, Rn. POP.L ERn is identical to MOV.L @SP+, ERn.
	PUSH	W/L	Rn → @–SP Pushes a register onto the stack. PUSH.W Rn is identical to MOV.W Rn, @–SP. PUSH.L ERn is identical to MOV.L ERn, @–SP.
	LDM	L	@SP+ → Rn (register list) Pops two or more general registers from the stack.
	STM	L	Rn (register list) → @–SP Pushes two or more general registers onto the stack.

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

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

operations			Performs a logical AND operation on a general register and another general register or immediate data.
	OR	B/W/L	Rd ∨ Rs → Rd, Rd ∨ #IMM → Rd Performs a logical OR operation on a general register and another general register or immediate data.
	XOR	B/W/L	Rd ⊕ Rs → Rd, Rd ⊕ #IMM → Rd Performs a logical exclusive OR operation on a general register and another general register or immediate data.
	NOT	B/W/L	¬ (Rd) → (Rd) Takes the one's complement of general register contents.
Shift operations	SHAL SHAR	B/W/L	Rd (shift) → Rd Performs an arithmetic shift on general register contents. 1-bit or 2-bit shift is possible.
	SHLL SHLR	B/W/L	Rd (shift) → Rd Performs a logical shift on general register contents. 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 carry flag. 1-bit or 2-bit rotation is possible.

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

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

instructions			is true. The branching conditions are listed below.

Mnemonic	Description	Condition
BRA(BT)	Always (true)	Always
BRN(BF)	Never (false)	Never
BHI	High	$C \vee Z = 0$
BLS	Low or same	$C \vee Z = 1$
BCC(BHS)	Carry clear (high or same)	$C = 0$
BCS(BLO)	Carry set (low)	$C = 1$
BNE	Not equal	$Z = 0$
BEQ	Equal	$Z = 1$
BVC	Overflow clear	$V = 0$
BVS	Overflow set	$V = 1$
BPL	Plus	$N = 0$
BMI	Minus	$N = 1$
BGE	Greater or equal	$N \oplus V = 0$
BLT	Less than	$N \oplus V = 1$
BGT	Greater than	$Z \vee (N \oplus V) = 0$
BLE	Less or equal	$Z \vee (N \oplus V) = 1$

JMP	—	Branches unconditionally to a specified address.
BSR	—	Branches to a subroutine at a specified address.
JSR	—	Branches to a subroutine at a specified address.
RTS	—	Returns from a subroutine.

instructions

RTE	—	Returns from an exception-handling routine.
SLEEP	—	Causes a transition to a power-down state.
LDC	B/W	(EAs) → CCR, (EAs) → EXR Moves the source operand contents or immediate data to CCR or EXR. Although CCR and EXR are 8-bit registers, word-size transfers are performed between them and memory. The upper 8 bits are valid.
STC	B/W	CCR → (EAd), EXR → (EAd) Transfers CCR or EXR contents to a general register or memory. Although CCR and EXR are 8-bit registers, word-size transfers are performed between them and memory. The upper 8 bits are valid.
ANDC	B	CCR ∧ #IMM → CCR, EXR ∧ #IMM → EXR Logically ANDs the CCR or EXR contents with immediate data.
ORC	B	CCR ∨ #IMM → CCR, EXR ∨ #IMM → EXR Logically ORs the CCR or EXR contents with immediate data.
XORC	B	CCR ⊕ #IMM → CCR, EXR ⊕ #IMM → EXR Logically exclusive-ORs the CCR or EXR contents with immediate data.
NOP	—	PC + 2 → PC Only increments the program counter.

transfer instruction			Repeat @ER5+ → @ER6+ R4L–1 → R4L Until R4L = 0 else next;
	EEPMOV.W	—	if R4 ≠ 0 then Repeat @ER5+ → @ER6+ R4–1 → R4 Until R4 = 0 else next; Transfers a data block according to parameters set in 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 soon as the 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 instruction.

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

Figure 2.9 shows examples of instruction formats.

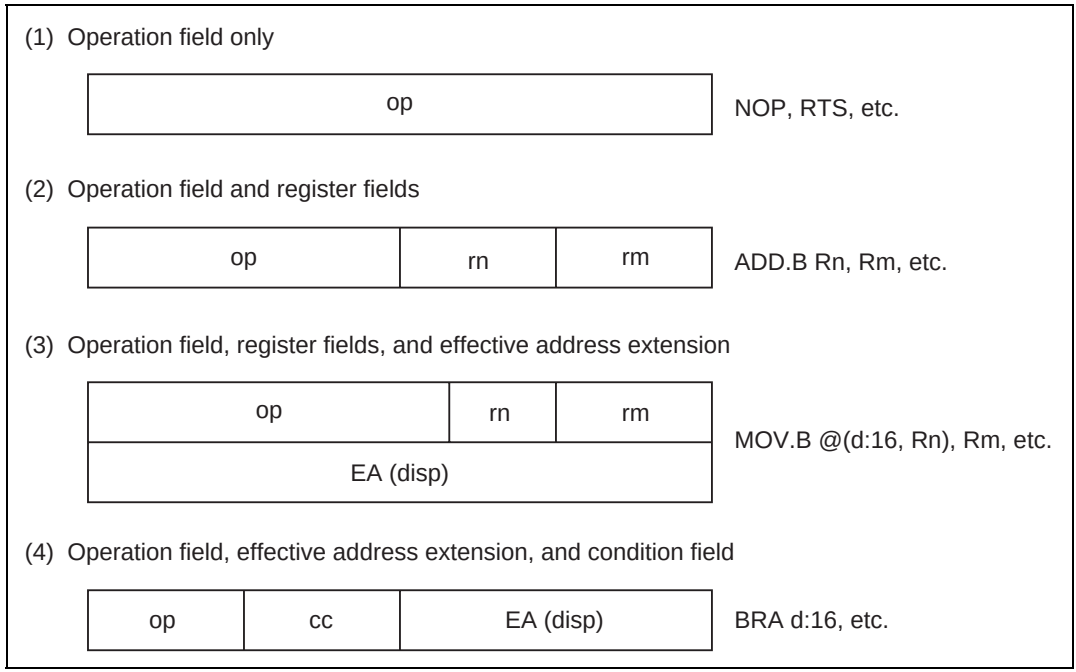


Figure 2.9 Instruction Formats (Examples)

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

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

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

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

2.7.1 Addressing Mode

The CPU supports the eight addressing modes listed in table 2.4. Each instruction uses a subset of these addressing modes. Arithmetic and logic instructions can use the register direct and immediate modes. Data transfer instructions can use all addressing modes except program-counter relative and memory indirect. Bit manipulation instructions use register direct, register indirect, or absolute addressing mode to specify an operand, and register direct (BSET, BCLR, BNOT, and BTST instructions) or immediate (3-bit) addressing mode to specify a bit number in the operand.

Table 2.4 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/@aa:32
6	Immediate	#xx:8/#xx:16/#xx:32
7	Program-counter relative	@(d:8,PC)/@(d:16,PC)
8	Memory indirect	@@aa:8

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

(2) Register Indirect—@ERn: The register field of the instruction code specifies an address register (ERn) which contains the address of the operand on memory. If the address is a program instruction address, the lower 24 bits are valid and the upper 8 bits are all assumed to be 0 (H'00).

(3) Register Indirect with Displacement—@(d:16, ERn) or @(d:32, ERn): A 16-bit or 32-bit displacement contained in the instruction is added to an address register (ERn) specified by the register field of the instruction, and the sum gives the address of a memory operand. A 16-bit displacement is sign-extended when added.

- Register indirect with post-increment—@ERn+

  The register field of the instruction code specifies an address register (ERn) which contains the address of a memory operand. After the operand is accessed, 1, 2, or 4 is added to the address register contents and the sum is stored in the address register. The value added is 1 for byte access, 2 for word transfer instruction, or 4 for longword transfer instruction. For word or 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 register field in the instruction code, and the result becomes the address of a memory operand. The result is also stored in the address register. The value subtracted is 1 for byte access, 2 for word transfer instruction, or 4 for longword transfer instruction. For word or longword transfer instruction, 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 address may be 8 bits long (@aa:8), 16 bits long (@aa:16), 24 bits long (@aa:24), or 32 bits long (@aa:32).

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

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

Table 2.5 indicates the accessible absolute address ranges.

Table 2.5 Absolute Address Access Ranges

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

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

(7) Program-Counter Relative—@(d:8, PC) or @(d:16, PC): This mode is used in the Bcc and BSR instructions. An 8-bit or 16-bit displacement contained in the instruction is sign-extended and added to the 24-bit PC contents to generate a branch address. Only the lower 24 bits of this branch address are valid; the upper 8 bits are all assumed to be 0 (H'00). The PC value to which the displacement is added is the address of the first byte of the next instruction, so the possible branching range is –126 to +128 bytes (–63 to +64 words) or –32766 to +32768 bytes (–16383 to +16384 words) from the branch instruction. The resulting value should be an even number.

(8) Memory Indirect—@@aa:8: This mode can be used by the JMP and JSR instructions. The instruction code contains an 8-bit absolute address specifying a memory operand. This memory operand contains a branch address. The upper bits of the absolute address are all assumed to be 0, so the address range is 0 to 255 (H'000000 to H'0000FF).

In advanced mode the memory operand is a longword operand, the first byte of which is assumed to be all 0 (H'00).

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

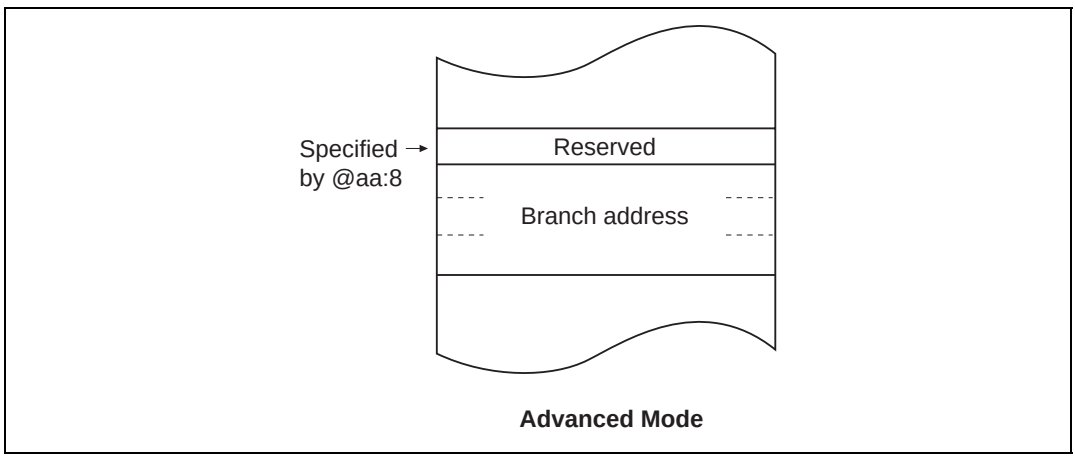
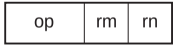
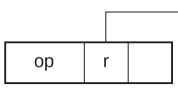
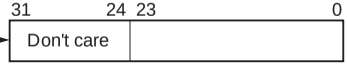
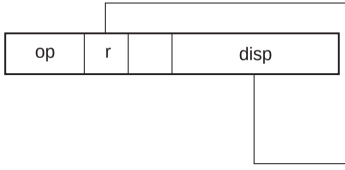
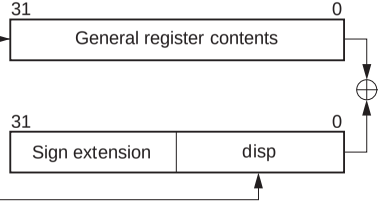
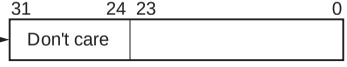
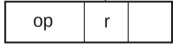
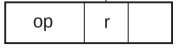
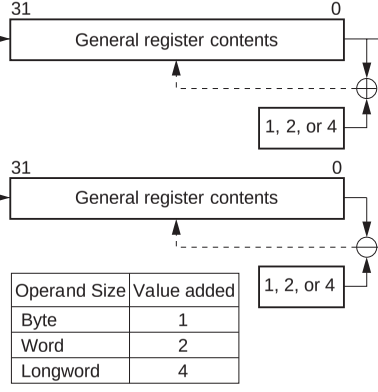
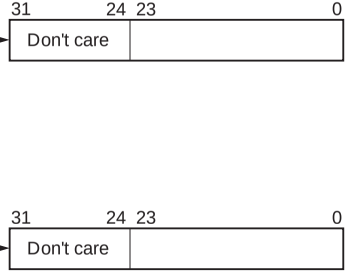


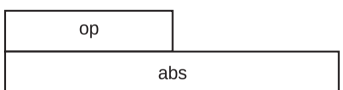
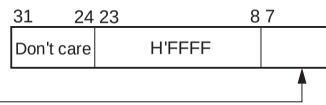
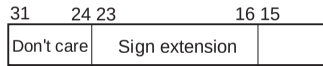
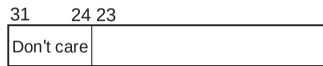
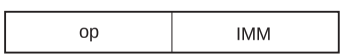
Figure 2.10 Branch Address Specification in Memory Indirect Mode

at the address preceding the specified address. (For further information, see section 2.5.2, Memory Data Formats.)

2.7.2 Effective Address Calculation

Table 2.6 indicates how effective addresses are calculated in each addressing mode.

No.	Addressing Mode and Instruction Format	Effective Address Calculation	Effective Address (EA)
1	Register direct (Rn) 		Operand is general register contents.
2	Register indirect (@ERn) 		
3	Register indirect with displacement @(d:16, ERn) or @(d:32, ERn) 		
4	Register indirect with post-increment or pre-decrement • Register indirect with post-increment @ERn+  • Register indirect with pre-decrement @-ERn 	 Operand Size / Value added Byte: 1 Word: 2 Longword: 4	

No.	Addressing Mode and Instruction Format	Effective Address Calculation	Effective Address (EA)
5	Absolute address @aa:8  @aa:16  @aa:24  @aa:32  		   
6	Immediate #xx:8/#xx:16/#xx:32  		Operand is immediate data.

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

2.8.1 Overview

The CPU has five main processing states: the reset state, exception handling state, program execution state, bus-released state, and power-down state. Figure 2.11 shows a diagram of the processing states. Figure 2.12 indicates the state transitions.

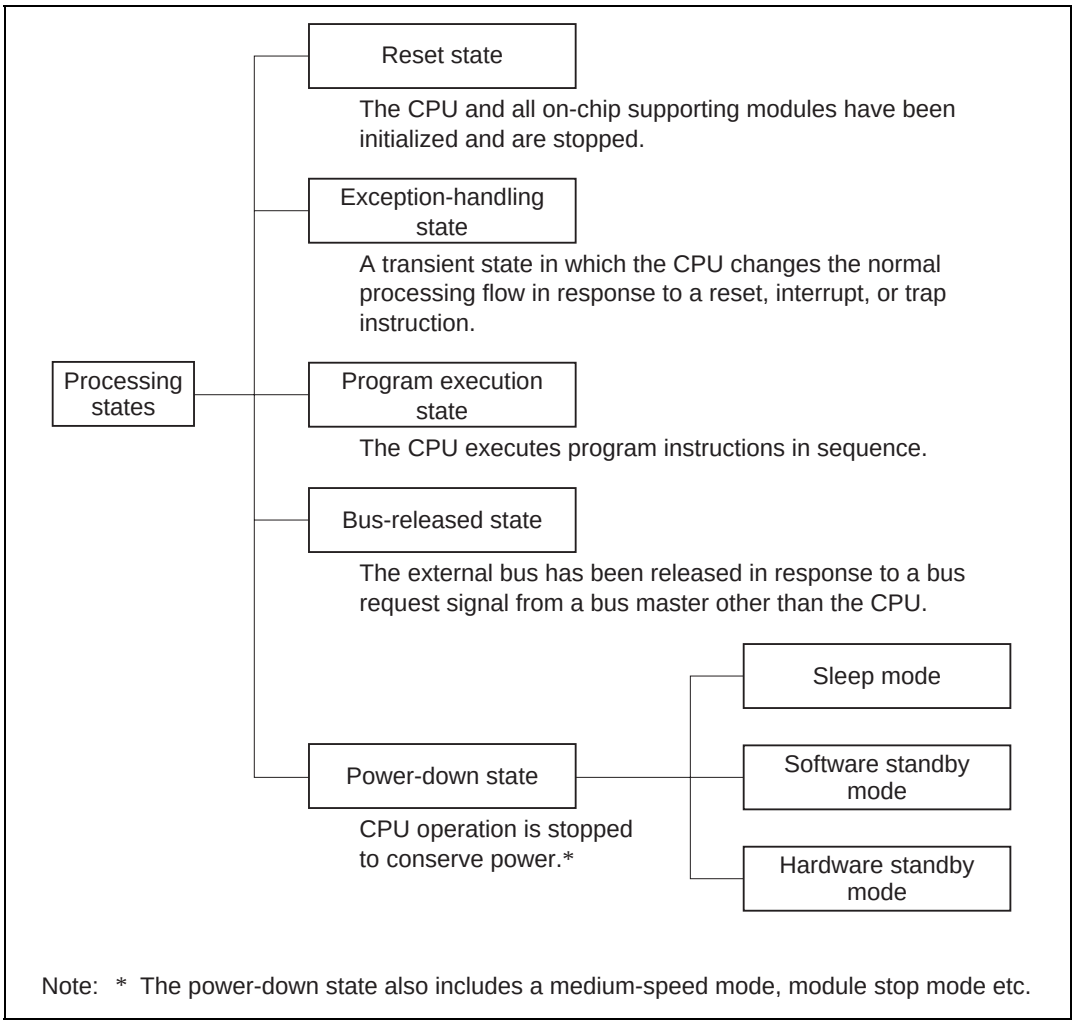
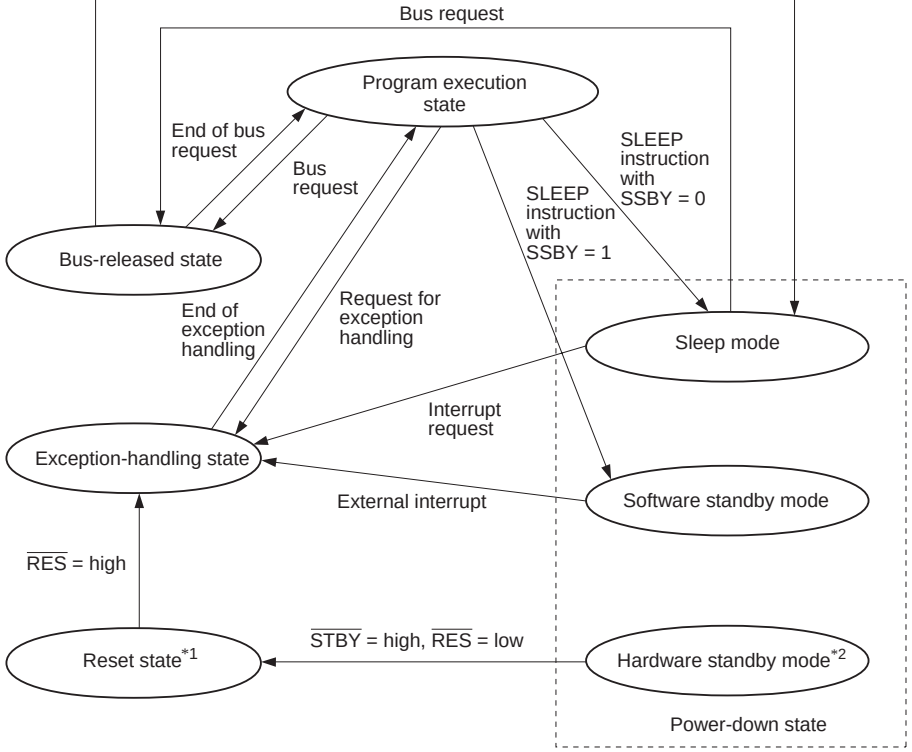


Figure 2.11 Processing States



Notes: 1. From any state except hardware standby mode, a transition to the reset state occurs whenever $\overline{RES}$ goes low. A transition can also be made to the reset state when the watchdog timer overflows.
2. From any state, a transition to hardware standby mode occurs when $\overline{STBY}$ goes low.

Figure 2.12 State Transitions

2.8.2 Reset State

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

The reset state can also be entered by a watchdog timer overflow. For details, refer to section 13, Watchdog Timer.

The exception-handling state is a transient state that occurs when the CPU alters the normal processing flow due to a reset, interrupt, or trap instruction. The CPU fetches a start address (vector) from the exception vector table and branches to that address.

(1) Types of Exception Handling and Their Priority

Exception handling is performed for traces, resets, interrupts, and trap instructions. Table 2.7 indicates the types of exception handling and their priority. Trap instruction exception handling is always accepted, in the program execution state.

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

Table 2.7 Exception Handling Types and Priority

Priority	Type of Exception	Detection Timing	Start of Exception Handling
High ↑ Low	Reset	Synchronized with clock	Exception handling starts immediately after a low-to-high transition at the $\overline{\text{RES}}$ pin, or when the watchdog timer overflows
	Trace	End of instruction execution or end of exception-handling sequence*1	When the trace (T) bit is set to 1, the trace starts at the end of the current instruction or current exception-handling sequence
	Interrupt	End of instruction execution or end of exception-handling sequence*2	When an interrupt is requested, exception handling starts at the end of the current instruction or current exception-handling sequence
	Trap instruction	When TRAPA instruction is executed	Exception handling starts when a trap (TRAPA) instruction is executed*3

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

After the RES pin has gone low and the reset state has been entered, when RES goes high again, reset exception handling starts. When reset exception handling starts the CPU fetches a start address (vector) from the exception vector table and starts program execution from that address. All interrupts, including NMI, are disabled during reset exception handling and after it ends.

(3) Traces

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

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

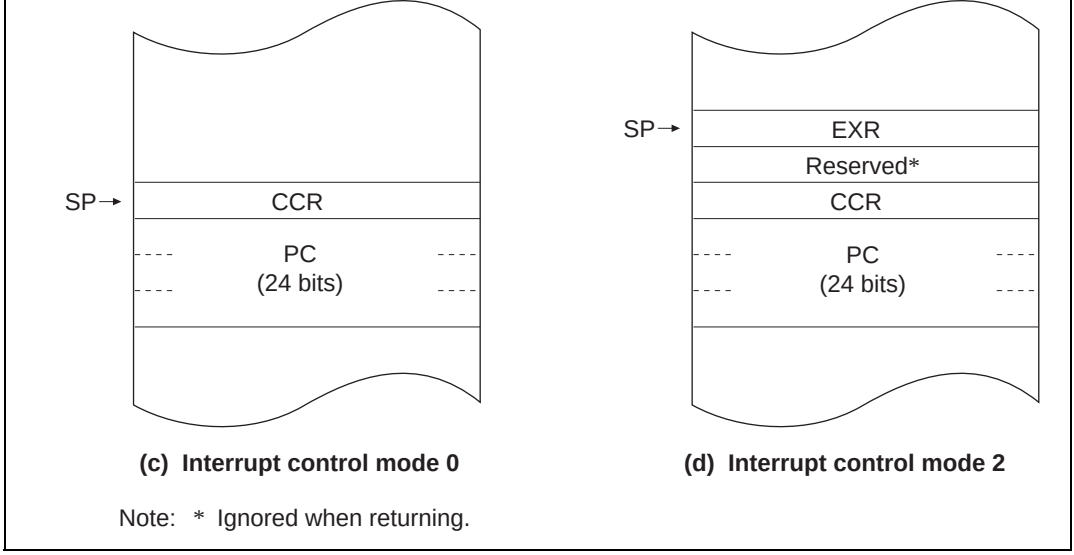
The T bit saved on the stack retains its value of 1, and when the RTE instruction is executed to return from the trace exception-handling routine, trace mode is entered again. Trace exception-handling is not executed at the end of the RTE instruction.

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

(4) Interrupt Exception Handling and Trap Instruction Exception Handling

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

Figure 2.13 shows the stack after exception handling ends.



(c) Interrupt control mode 0 **(d) Interrupt control mode 2**

Note: * Ignored when returning.

Figure 2.13 Stack Structure after Exception Handling (Examples)

2.8.4 Program Execution State

In this state the CPU executes program instructions in sequence.

2.8.5 Bus-Released State

This is a state in which the bus has been released in response to a bus request from a bus master other than the CPU. While the bus is released, the CPU halts.

There is one other bus master in addition to the CPU: the DMA controller (DMAC) and data transfer controller (DTC).

For further details, refer to section 6, Bus Controller.

The power-down state includes both modes in which the CPU stops operating and modes in which the CPU does not stop. There are three modes in which the CPU stops operating: sleep mode, software standby mode, and hardware standby mode. There are also two other power-down modes: medium-speed mode, and module stop mode. In medium-speed mode the CPU and other bus masters operate on a medium-speed clock. Module stop mode permits halting of the operation of individual modules, other than the CPU. For details, refer to section 21, Power-Down Modes.

(1) Sleep Mode: A transition to sleep mode is made if the SLEEP instruction is executed while the software standby bit (SSBY) in the standby control register (SBYCR) is cleared to 0. In sleep mode, CPU operations stop immediately after execution of the SLEEP instruction. The contents of CPU registers are retained.

(2) Software Standby Mode: A transition to software standby mode is made if the SLEEP instruction is executed while the SSBY bit in SBYCR is set to 1. In software standby mode, the CPU and clock halt and all MCU operations stop. As long as a specified voltage is supplied, the contents of CPU registers and on-chip RAM are retained. The I/O ports also remain in their existing states.

(3) Hardware Standby Mode: A transition to hardware standby mode is made when the $\overline{\text{STBY}}$ pin goes low. In hardware standby mode, the CPU and clock halt and all MCU operations stop. The on-chip supporting modules are reset, but as long as a specified voltage is supplied, on-chip RAM contents are retained.

2.9 Basic Timing

2.9.1 Overview

The CPU is driven by a system clock, denoted by the symbol ϕ. The period from one rising edge of ϕ to the next is referred to as a "state." The memory cycle or bus cycle consists of one, two, or three states. Different methods are used to access on-chip memory, on-chip supporting modules, and the external address space.

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 byte and word transfer instruction. Figure 2.14 shows the on-chip memory access cycle. Figure 2.15 shows the pin states.

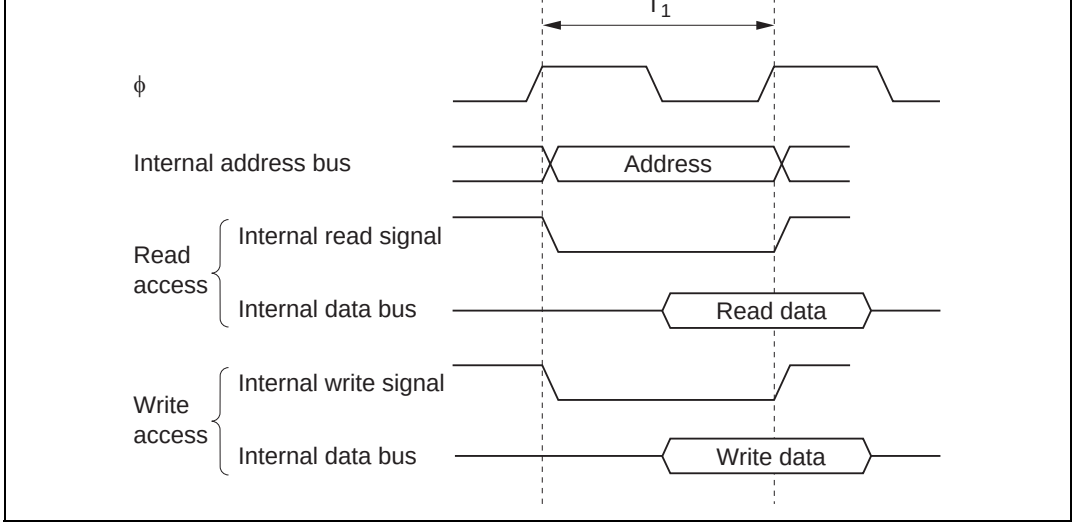


Figure 2.14 On-Chip Memory Access Cycle

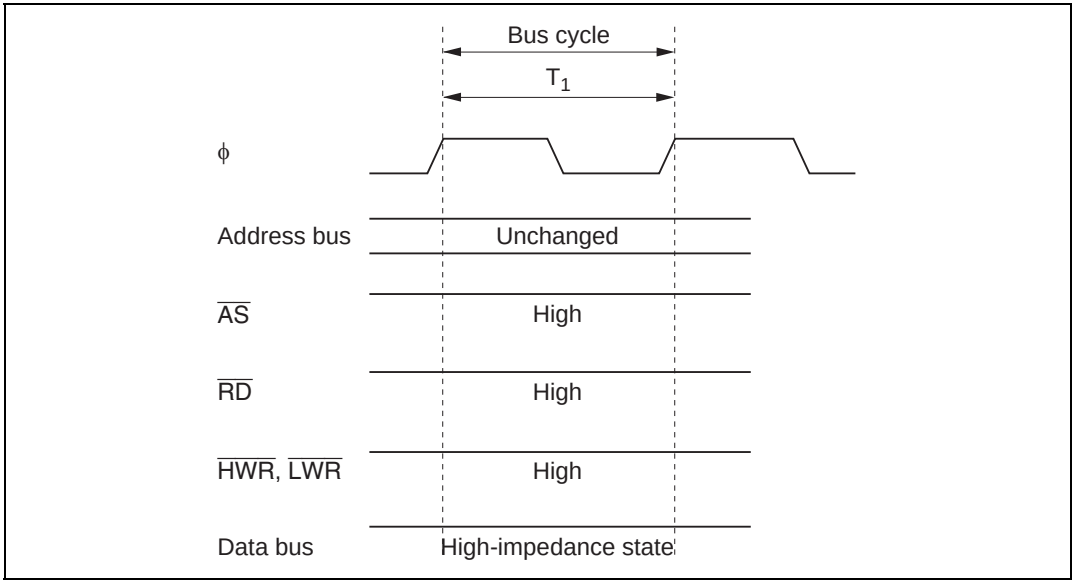


Figure 2.15 Pin States during On-Chip Memory Access

The on-chip supporting modules are accessed in two states. The data bus is either 8 bits or 16 bits wide, depending on the particular internal I/O register being accessed. Figure 2.16 shows the access timing for the on-chip supporting modules. Figure 2.17 shows the pin states.

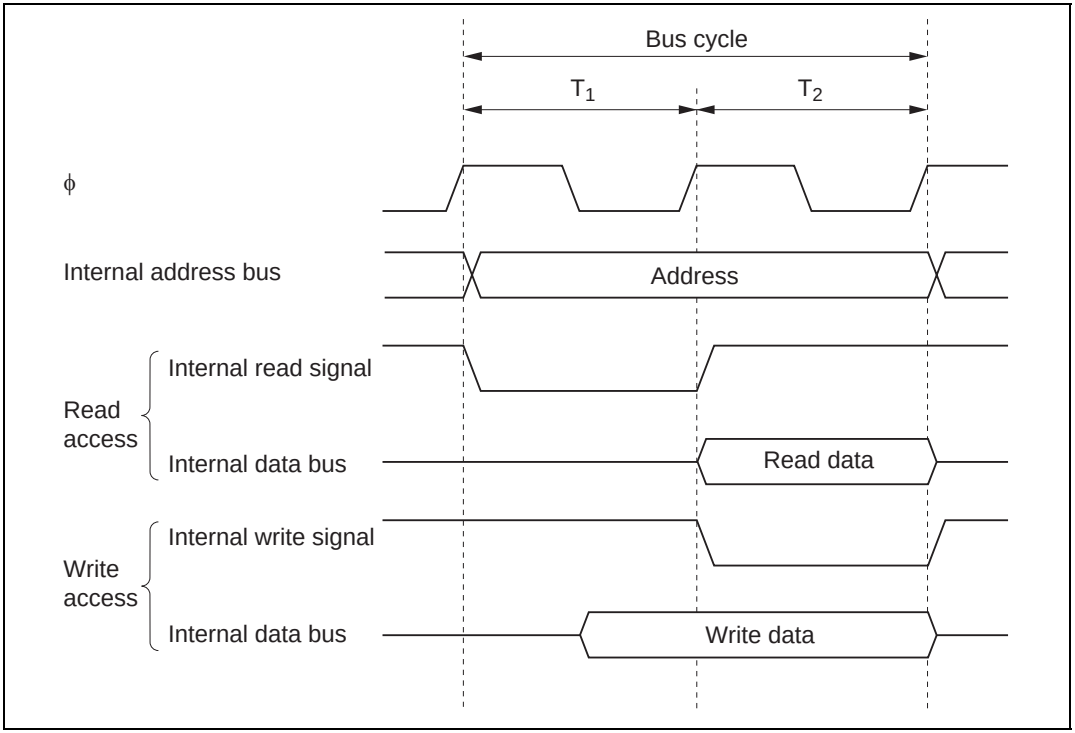


Figure 2.16 On-Chip Supporting Module Access Cycle

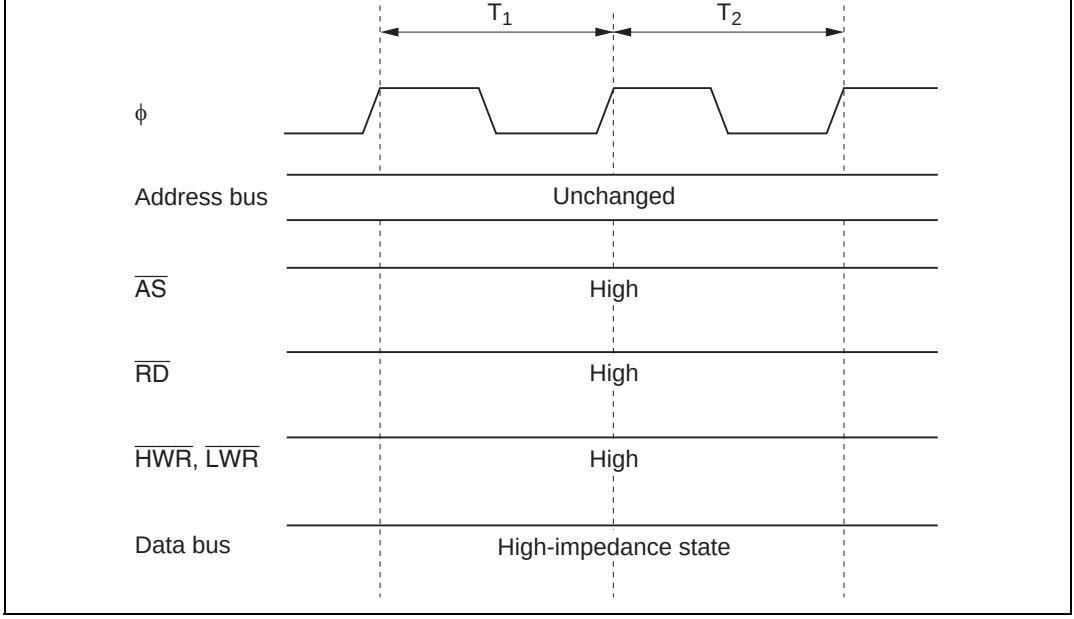


Figure 2.17 Pin States during On-Chip Supporting Module Access

2.9.4 External Address Space Access Timing

The external address space is accessed with an 8-bit or 16-bit data bus width in a two-state or three-state bus cycle. In three-state access, wait states can be inserted. For further details, refer to section 6, Bus Controller.

2.10 Usage Note

2.10.1 TAS Instruction

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

3.1 Overview

3.1.1 Operating Mode Selection (H8S/2338 F-ZTAT)

This version has eight operating modes (modes 4 to 7, 10, 11, 14, and 15). These modes are determined by the mode pin (MD_2 to MD_0) and flash write enable pin (FWE) settings. The CPU operating mode and initial bus width can be selected as shown in table 3.1.

Table 3.1 lists the MCU operating modes.

Table 3.1 MCU Operating Mode Selection (H8S/2338 F-ZTAT)

MCU Operating Mode	FWE	MD_2	MD_1	MD_0	CPU Operating Mode	Description	On-Chip ROM	External Data Bus	
								Initial Value	Max Value
1	0	0	0	1	—	—	—	—	—
2			1	0					
3				1					
4		1	0	0	Advanced	Expanded mode with on-chip ROM disabled	Disabled	16 bits	16 bits
5				1				8 bits	16 bits
6			1	0		Expanded mode with on-chip ROM enabled	Enabled	8 bits	16 bits
7				1		Single-chip mode		—	—
8	1	0	0	0	—	—	—	—	—
9				1					
10			1	0	Advanced	Boot mode	Enabled	8 bits	16 bits
11				1				—	—
12		1	0	0	—	—	—	—	—
13				1					
14			1	0	Advanced	User program mode	Enabled	8 bits	16 bits
15				1				—	—

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

The external expansion modes allow switching between 8-bit and 16-bit bus modes. After program execution starts, an 8-bit or 16-bit address space can be set for each area, depending on the bus controller setting. If 16-bit access is selected for any one area, 16-bit bus mode is set; if 8-bit access is selected for all areas, 8-bit bus mode is set. Note that the functions of each pin depend on the operating mode.

Modes 10, 11, 14, and 15 are boot modes and user program modes in which the flash memory can be programmed and erased. For details, see section 19, ROM.

This version can only be used in modes 4 to 7, 10, 11, 14, and 15. This means that the flash write enable pin and mode pins must be set to select one of these modes.

Do not change the inputs at the mode pins during operation.

3.1.2 Operating Mode Selection (Mask ROM and ROMless Versions, H8S/2339 F-ZTAT)

The ROMless and Mask ROM versions have four operating modes (modes 4 to 7). H8S/2339 F-ZTAT has six operating modes (modes 2 to 7). The operating mode is determined by the mode pins (MD_2 to MD_0). The CPU operating mode, enabling or disabling of on-chip ROM, and the initial bus width setting can be selected as shown in table 3.2.

Table 3.2 lists the MCU operating modes.

MCU Operating Mode	MD_2	MD_1	MD_0	CPU Operating Mode	Description	On-Chip ROM	External Data Bus Initial Value	External Data Bus Max Value
1	0	0	1	—	—	—	—	—
2		1	0					
3			1					
4*	1	0	0	Advanced	Expanded mode with on-chip ROM disabled	Disabled	16 bits	16 bits
5*			1				8 bits	16 bits
6		1	0		Expanded mode with on-chip ROM enabled	Enabled	8 bits	16 bits
7			1		Single-chip mode		—	—

Note: * Only modes 4 and 5 are provided in the ROMless version.

The CPU's architecture allows for 4 Gbytes of address space, but these versions actually access a maximum of 16 Mbytes.

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

The external expansion modes allow switching between 8-bit and 16-bit bus modes. After program execution starts, an 8-bit or 16-bit address space can be set for each area, depending on the bus controller setting. If 16-bit access is selected for any one area, 16-bit bus mode is set; if 8-bit access is selected for all areas, 8-bit bus mode is set. Note that the functions of each pin depend on the operating mode.

The ROMless and Mask ROM versions can only be used in modes 4 to 7. This means that the mode pins must be set to select one of these modes. However, note that only mode 4 or 5 can be set for the ROMless version.

H8S/2339 F-ZTAT can only be used in modes 2 to 7. This means that the mode pins must be set to select one of these modes.

Do not change the inputs at the mode pins during operation.

The chip has a mode control register (MDCR) that indicates the inputs at the mode pins (MD_2 to MD_0), and a system control register (SYSCR) and system control register 2 (SYSCR2)*2 that control the operation of the chip. Table 3.3 summarizes these registers.

Table 3.3 Registers

Name	Abbreviation	R/W	Initial Value	Address*1
Mode control register	MDCR	R	Undefined	H'FF3B
System control register	SYSCR	R/W	H'01	H'FF39
System control register 2*2	SYSCR2	R/W	H'00	H'FF42

Notes: 1. Lower 16 bits of the address.
2. The SYSCR2 register can only be used in the F-ZTAT version. In the mask ROM and ROMless versions this register will return an undefined value if read, and cannot be modified.

3.2 Register Descriptions

3.2.1 Mode Control Register (MDCR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	—	MDS2	MDS1	MDS0
Initial value	:	1	0	0	0	0	—*	—*	—*
R/W	:	—	—	—	—	—	R	R	R

Note: * Determined by pins MD_2 to MD_0.

MDCR is an 8-bit read-only register that indicates the current operating mode of the H8S/2339 Group chip.

Bit 7—Reserved: This bit is always read as 1, and cannot be modified.

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 levels at pins MD_2 to MD_0 (the current operating mode). Bits MDS2 to MDS0 correspond to pins MD_2 to MD_0. MDS2 to MDS0 are read-only bits, and cannot be written to. The mode pin (MD_2 to MD_0) input levels are latched into these bits when MDCR is read. These latches are canceled by a reset.

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

Bit 7—Reserved: Only 0 should be written to this bit.

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 the control mode of the interrupt controller. For details of the interrupt control modes, see section 5.4.1, Interrupt Control Modes and Interrupt Operation.

Bit 5 INTM1	Bit 4 INTM0	Interrupt Control Mode	Description	
0	0	0	Control of interrupts by I bit	(Initial value)
	1	—	Setting prohibited	
1	0	2	Control of interrupts by I2 to I0 bits and IPR	
	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	(Initial value)
1	An interrupt is requested at the rising edge of NMI input	

Bit 2—LWR Output Disable (LWROD): Enables or disables $\overline{\text{LWR}}$ output.

Bit 2 LWROD	Description	
0	PF_3 is designated as $\overline{\text{LWR}}$ output pin	(Initial value)
1	PF_3 is designated as I/O port, and does not function as $\overline{\text{LWR}}$ output pin	

Bit 1—IRQ Port Switching Select (IRQPAS): Selects switching of input pins for $\overline{\text{IRQ}}_4$ to $\overline{\text{IRQ}}_7$. $\overline{\text{IRQ}}_4$ to $\overline{\text{IRQ}}_7$ input is always performed from one of the ports.

0	$P9_4$ to $P9_7$ are used for $\overline{IRQ}_4$ to $\overline{IRQ}_7$ input	(Initial value)
1	$P5_3$ to $P5_0$ are used for $\overline{IRQ}_4$ to $\overline{IRQ}_7$ input	

Bit 0—RAM Enable (RAME): Enables or disables the on-chip RAM. The RAME bit is initialized when the reset state is released. It is not initialized in software standby mode.

Bit 0 RAME	Description	
0	On-chip RAM is disabled	
1	On-chip RAM is enabled	(Initial value)

3.2.3 System Control Register 2 (SYSCR2) (F-ZTAT Version Only)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	FLSHE	—	—	—
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	—	—	—	—	R/W	—	—	— (R/W)*

Note: * R/W in the H8S/2339 F-ZTAT.

SYSCR2 is an 8-bit readable/writable register that performs on-chip flash memory control.

SYSCR2 is initialized to H'00 by a reset, and in hardware standby mode.

Bits 7 to 4—Reserved: These bits are always read as 0, and cannot be modified.

Bit 3—Flash Memory Control Register Enable (FLSHE): Controls CPU access to the flash memory control registers (FLMCR1, FLMCR2, EBR1, and EBR2). For details, see section 19, ROM.

Bit 3 FLSHE	Description	
0	Flash control registers are not selected for addresses H'FFFFC8 to H'FFFFCB	(Initial value)
1	Flash control registers are selected for addresses H'FFFFC8 to H'FFFFCB	

Bits 2 and 1—Reserved: These bits are always read as 0, and cannot be modified.

3.3 Operating Mode Descriptions

3.3.1 Mode 1

Mode 1 is not supported in the H8S/2339 Group, and must not be set.

3.3.2 Mode 2 (H8S/2339 F-ZTAT Only)

This is a flash memory boot mode. See section 19, ROM, for details. This is the same as advanced on-chip ROM enabled expansion mode, except when erasing and reprogramming flash memory.

3.3.3 Mode 3 (H8S/2339 F-ZTAT Only)

This is a flash memory boot mode. See section 19, ROM, for details. This is the same as advanced single-chip ROM mode, except when erasing and reprogramming flash memory.

3.3.4 Mode 4 (Expanded Mode with On-Chip ROM Disabled)

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

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

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

3.3.5 Mode 5 (Expanded Mode with On-Chip ROM Disabled)

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

Ports A, B, and C function as an address bus, port D functions as a data bus, and part of port F carries bus control signals.

The initial bus mode after a reset is 8 bits, with 8-bit access to all areas. However, note that if at least one area is designated for 16-bit access by the bus controller, the bus mode switches to 16 bits and port E becomes a data bus.

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

Ports A, B, and C function as input ports immediately after a reset. These pins can be set to output addresses by setting the corresponding data direction register (DDR) bits to 1. Port D functions as a data bus, and part of port F carries bus control signals.

The initial bus mode after a reset is 8 bits, with 8-bit access to all areas. However, note that if at least one area is designated for 16-bit access by the bus controller, the bus mode switches to 16 bits and port E becomes a data bus.

3.3.7 Mode 7 (Single-Chip Mode)

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

All I/O ports are available for use as input/output ports.

3.3.8 Modes 8 and 9 (H8S/2338 F-ZTAT Only)

Modes 8 and 9 are not supported and must not be set.

3.3.9 Mode 10 (H8S/2338 F-ZTAT Only)

This is a flash memory boot mode. For details, see section 19, ROM.

Except for the fact that flash memory programming and erasing can be performed, operation in this mode is the same as in advanced expanded mode with on-chip ROM enabled.

3.3.10 Mode 11 (H8S/2338 F-ZTAT Only)

This is a flash memory boot mode. For details, see section 19, ROM.

Except for the fact that flash memory programming and erasing can be performed, operation in this mode is the same as in advanced single-chip mode.

3.3.11 Modes 12 and 13 (H8S/2338 F-ZTAT Only)

Modes 12 and 13 are not supported and must not be set.

This is a flash memory user program mode. For details, see section 19, ROM.

Except for the fact that flash memory programming and erasing can be performed, operation in this mode is the same as in advanced expanded mode with on-chip ROM enabled.

3.3.13 Mode 15 (H8S/2338 F-ZTAT Only)

This is a flash memory user program mode. For details, see section 19, ROM.

Except for the fact that flash memory programming and erasing can be performed, operation in this mode is the same as in advanced single-chip mode.

The pin functions of ports A to F vary depending on the operating mode. Table 3.4 shows their functions in each operating mode.

Table 3.4 Pin Functions in Each Mode

Port		Mode 2*4	Mode 3*4	Mode 4	Mode 5	Mode 6*2	Mode 7*2	Mode 10*3	Mode 11*3	Mode 14*3	Mode 15*3
Port A	PA_7 to PA_5	P*1/A	P	P*1/A	P*1/A	P*1/A	P	P*1/A	P	P*1/A	P
	PA_4 to PA_0			A	A						
Port B		P*1/A	P	A	A	P*1/A	P	P*1/A	P	P*1/A	P
Port C		P*1/A	P	A	A	P*1/A	P	P*1/A	P	P*1/A	P
Port D		D	P	D	D	D	P	D	P	D	P
Port E		P*1/D	P	P/D*1	P*1/D	P*1/D	P	P*1/D	P	P*1/D	P
Port F	PF_7	P*1/C	P*1/C	P/C*1	P/C*1	P/C*1	P*1/C	P/C*1	P*1/C	P/C*1	P*1/C
	PF_6		P				P		P		P
	PF_5 to PF_4	C		C	C	C		C		C	
	PF_3	P/C*1		P/C*1	P/C*1	P/C*1		P/C*1		P/C*1	
	PF_2 to PF_0	P*1/C		P*1/C	P*1/C	P*1/C		P*1/C		P*1/C	

Legend

P: I/O port

A: Address bus output

D: Data bus I/O

C: Control signals, clock I/O

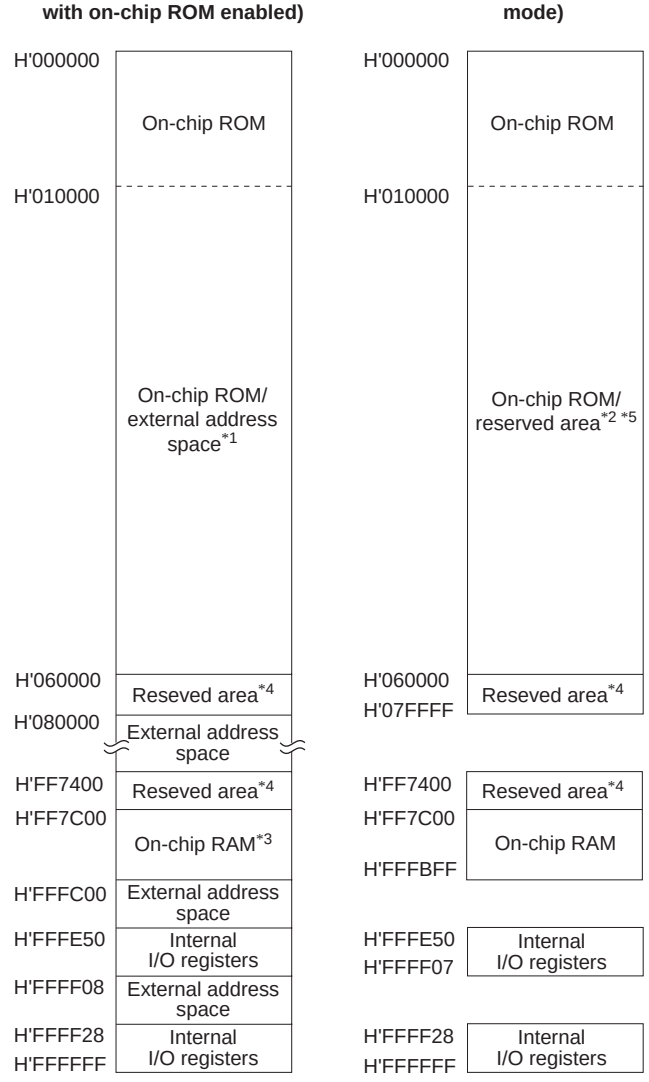
Notes: 1. After reset.
2. Setting is prohibited in the ROMless versions.
3. Setting prohibited except in case of the H8S/2338 F-ZTAT.
4. Valid only in the H8S/2339 F-ZTAT.

3.5 Memory Map in Each Operating Mode

Figures 3.1 to 3.4 show memory maps for each of the operating modes.

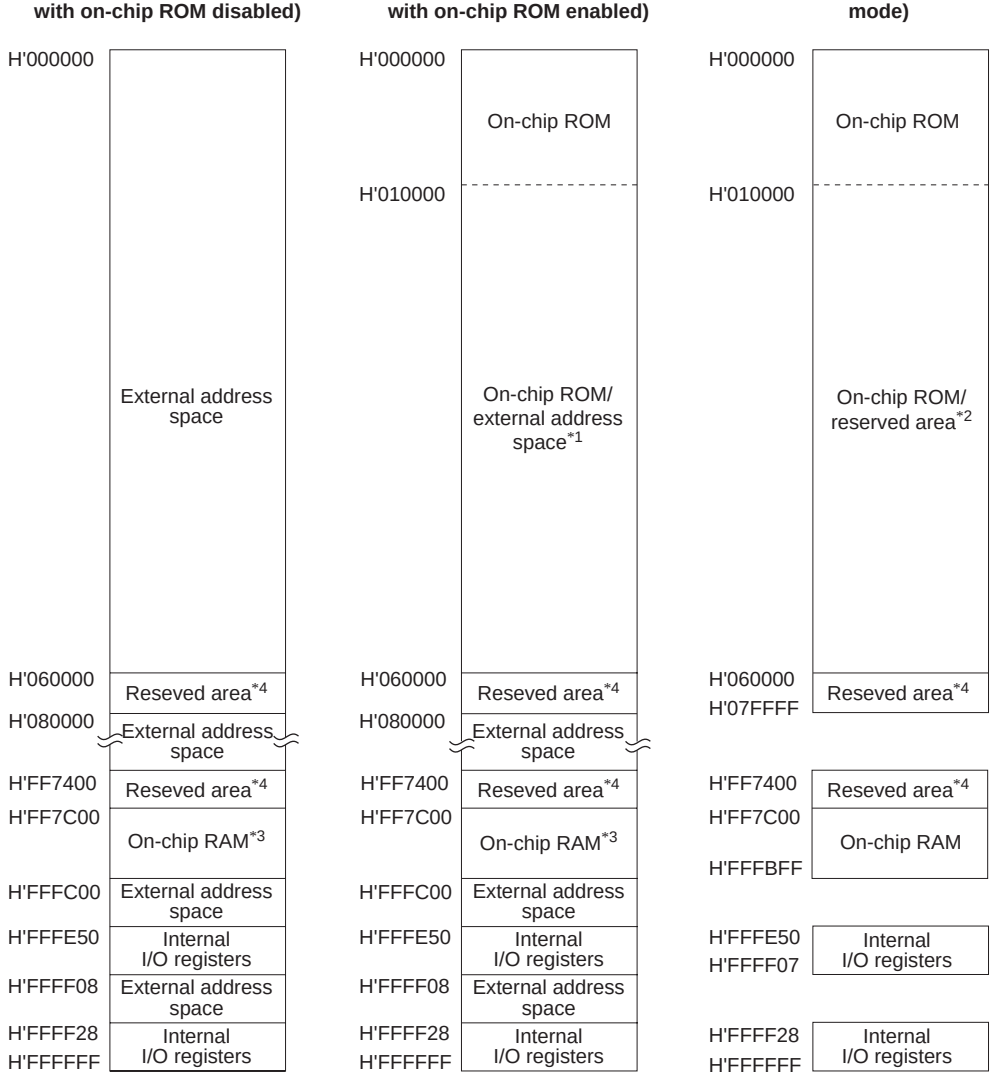
The address space is 16 Mbytes.

The address space is divided into eight areas.



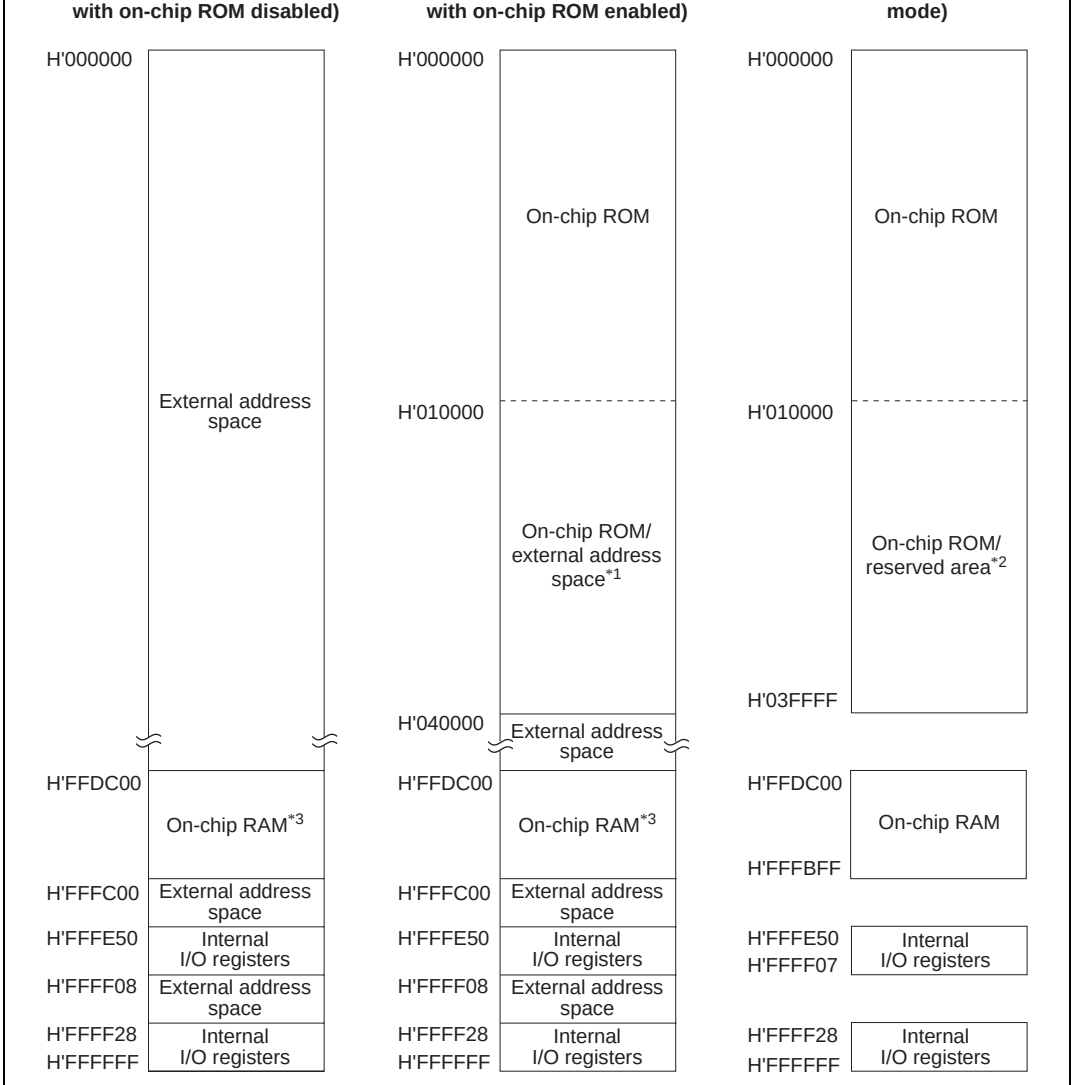
Notes: 1. External addresses when EAE = 1 in BCRL; on-chip ROM when EAE = 0.
2. Reserved area when EAE = 1 in BCRL; on-chip ROM when EAE = 0.
3. On-chip RAM is used for flash memory programming. Do not clear the RAME bit in SYSCR to 0.
4. Access to the reserved areas H'060000 to H'07FFFF and H'FF7400 to H'FF7BFF is prohibited.
5. Do not access a reserved area.

Figure 3.1 (a) H8S/2339 Memory Map in Each Operating Mode



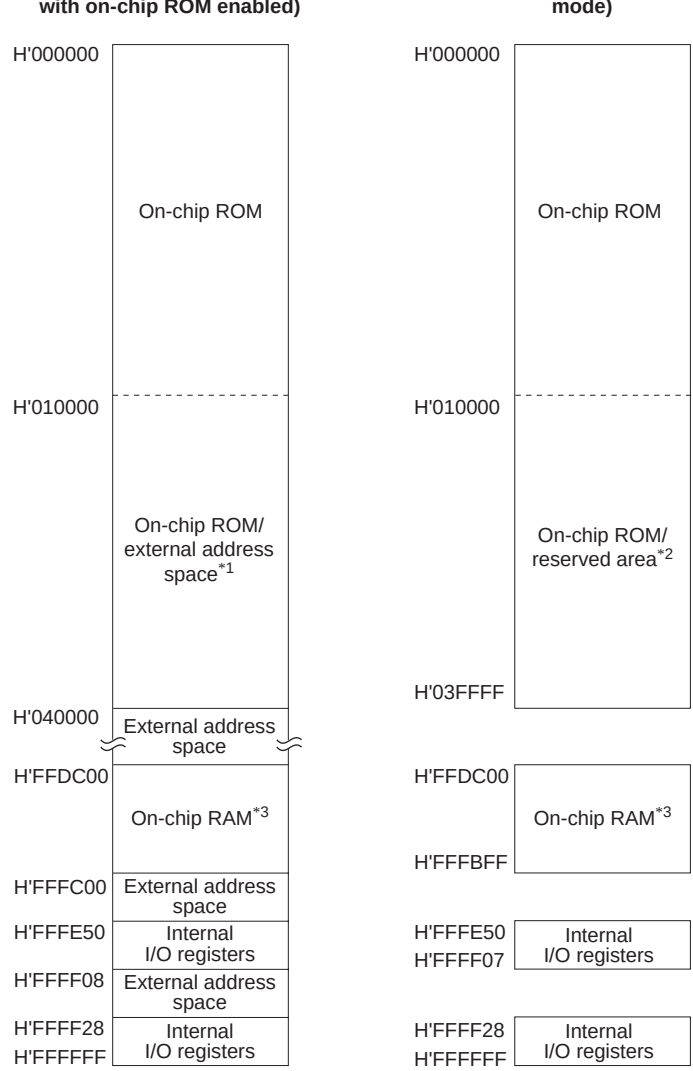
Notes: 1. External addresses when EAE = 1 in BCRL; on-chip ROM when EAE = 0.
2. Reserved area when EAE = 1 in BCRL; on-chip ROM when EAE = 0.
3. External addresses can be accessed by clearing the RAME bit in SYSCR to 0.
4. Access to the reserved areas H'060000 to H'07FFFF and H'FF7400 to H'FF7BFF is prohibited.

Figure 3.1 (b) H8S/2339 Memory Map in Each Operating Mode



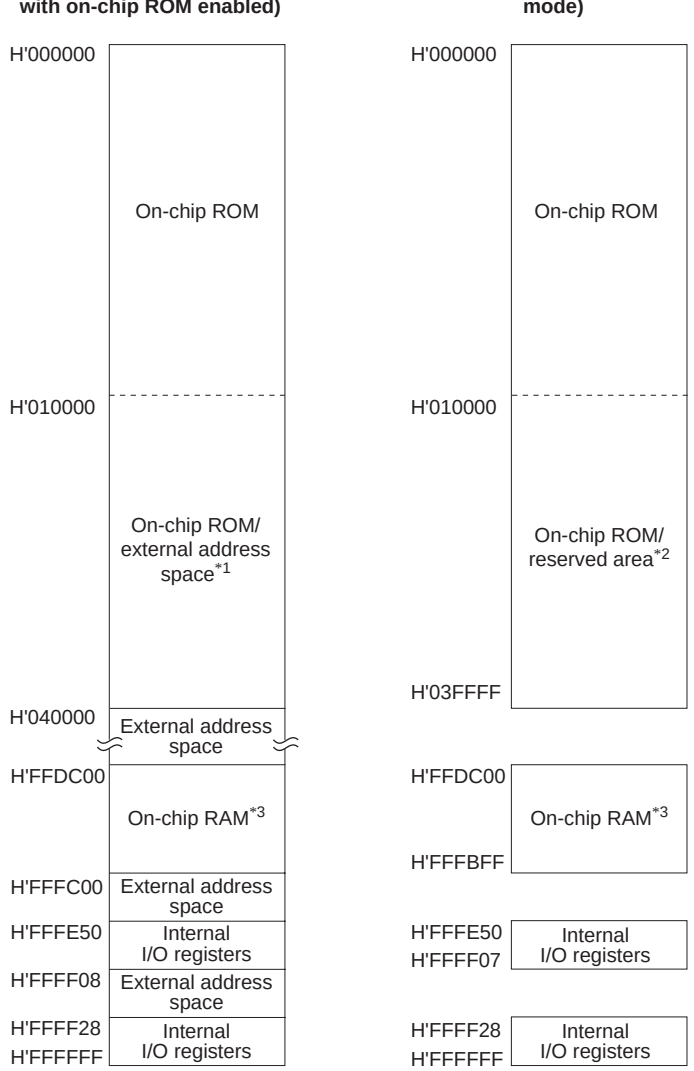
Notes: 1. External addresses when EAE = 1 in BCRL; on-chip ROM when EAE = 0.
2. Reserved area when EAE = 1 in BCRL; on-chip ROM when EAE = 0.
3. External addresses can be accessed by clearing the RAME bit in SYSCR to 0.

Figure 3.2 H8S/2338 Memory Map in Each Operating Mode



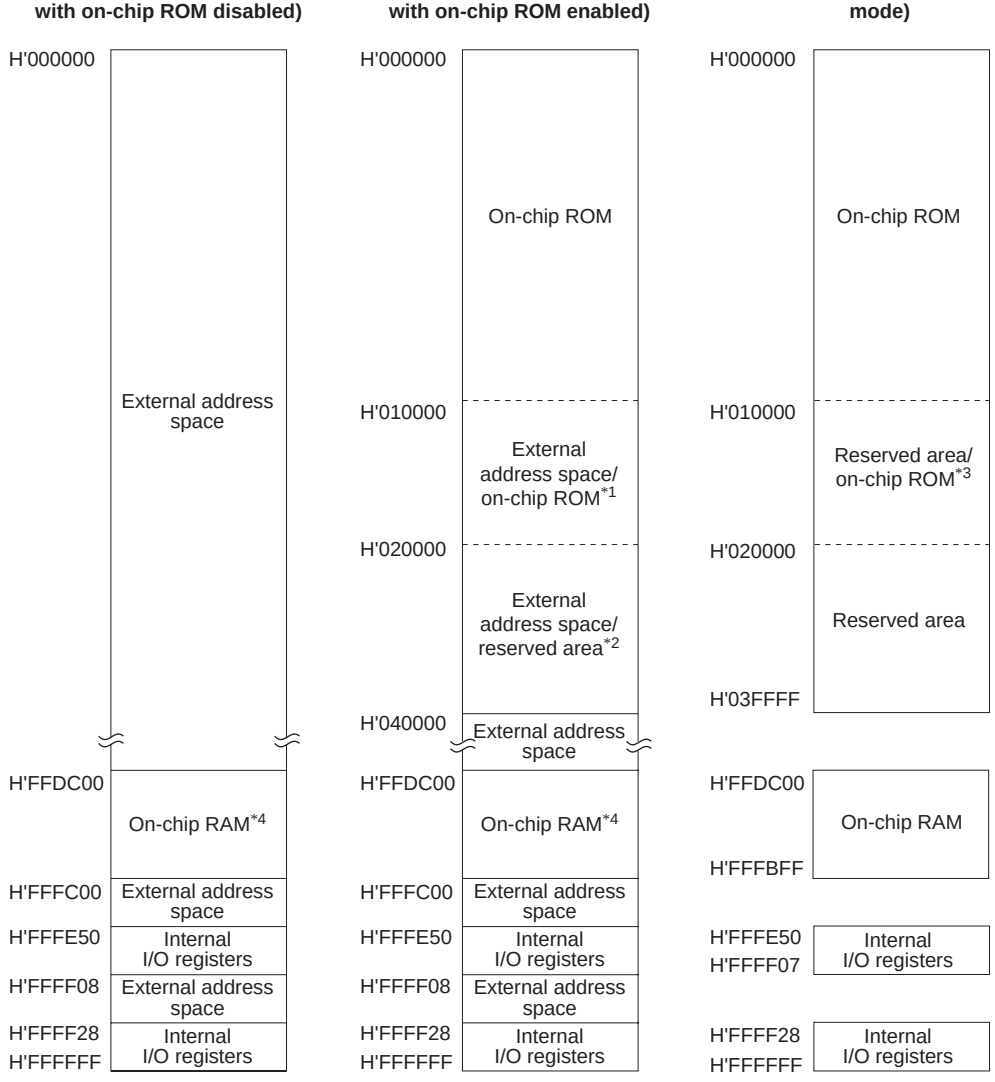
Notes: 1. External addresses when EAE = 1 in BCRL; on-chip ROM when EAE = 0.
2. Reserved area when EAE = 1 in BCRL; on-chip ROM when EAE = 0.
3. On-chip RAM is used for flash memory programming. Do not clear the RAME bit in SYSCR to 0.

Figure 3.2 H8S/2338 Memory Map in Each Operating Mode (cont)
(F-ZTAT Version Only)



Notes: 1. External addresses when EAE = 1 in BCRL; on-chip ROM when EAE = 0.
2. Reserved area when EAE = 1 in BCRL; on-chip ROM when EAE = 0.
3. On-chip RAM is used for flash memory programming. Do not clear the RAME bit in SYSCR to 0.

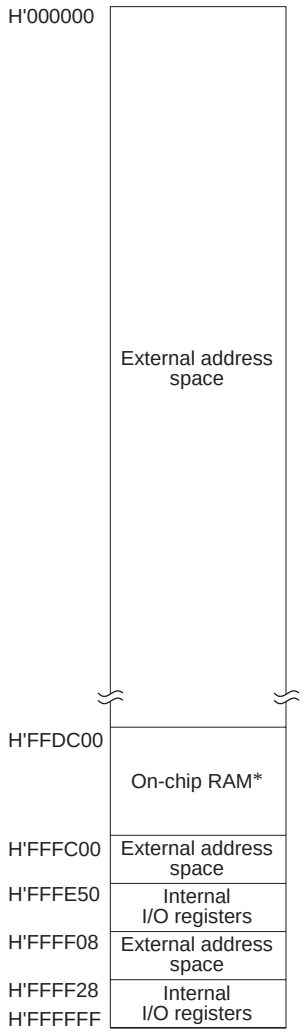
Figure 3.2 H8S/2338 Memory Map in Each Operating Mode (cont)
(F-ZTAT Version Only)



Notes: 1. External addresses when EAE = 1 in BCRL;on-chip ROM when EAE = 0.
2. External addresses when EAE = 1 in BCRL; reserved area when EAE = 0.
3. Reserved area when EAE = 1 in BCRL; on-chip ROM when EAE = 0.
4. External addresses can be accessed by clearing the RAME bit in SYSCR to 0.

Figure 3.3 H8S/2337 Memory Map in Each Operating Mode

with on-chip ROM disabled)



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

Figure 3.4 H8S/2332 Memory Map in Each Operating Mode

4.1 Overview

4.1.1 Exception Handling Types and Priority

As table 4.1 indicates, exception handling may be caused by a reset, trap instruction, or interrupt. Exception handling is prioritized as shown in table 4.1. If two or more exceptions occur simultaneously, they are accepted and processed in order of priority. Trap instruction exceptions are accepted at all times in the program execution state.

Exception handling sources, the stack structure, and the operation of the CPU vary depending on 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 $\overline{\text{RES}}$ pin, or when the watchdog timer overflows
	Trace*1	Starts when execution of the current instruction or exception handling ends, if the trace (T) bit is set to 1
	Interrupt	Starts when execution of the current instruction or exception handling ends, if an interrupt request has been issued*2
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 handling is not executed after execution of an RTE instruction.
2. Interrupt detection is not performed on completion of ANDC, ORC, XORC, or LDC instruction execution, or on completion of reset exception handling.
3. Trap instruction exception handling requests are accepted at all times in the program execution state.

Exceptions originate from various sources. Trap instructions and interrupts are handled as follows:

1. The program counter (PC), condition code register (CCR), and extend register (EXR) are pushed onto the stack.
2. The interrupt mask bits are updated. The T bit is cleared to 0.
3. A vector address corresponding to the exception source is generated, and program execution 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 are assigned to different exception sources.

Table 4.2 lists the exception sources and their vector addresses.

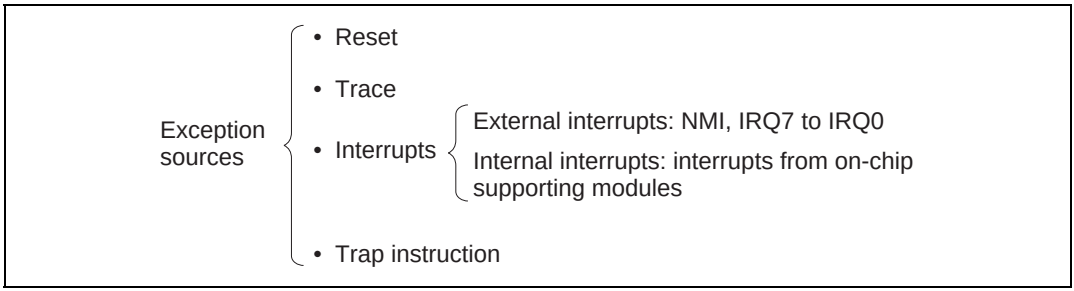


Figure 4.1 Exception Sources

In modes 6 and 7, the on-chip ROM available for use after a power-on reset is the 64-kbyte area comprising addresses H'000000 to H'00FFFF. Care is required when setting vector addresses. In this case, clearing the EAE bit in BCRL enables the 256-kbyte (128 kbytes/384 kbytes/512 kbytes)* area comprising addresses H'000000 to H'03FFFF (to H'01FFFF/H'05FFFF/H'07FFFF) to be used. For details, see section 6.2.5, Bus Control Register L (BCRL).

Note: * The amount of on-chip ROM differs depending on the product.

Exception Source		Vector Number	Vector Address Advanced Mode
Reset		0	H'0000 to H'0003
Reserved		1	H'0004 to H'0007
Reserved for system use		2	H'0008 to H'000B
		3	H'000C to H'000F
		4	H'0010 to H'0013
Trace		5	H'0014 to H'0017
Reserved for system use		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 for system use		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
	IRQ6	22	H'0058 to H'005B
	IRQ7	23	H'005C to H'005F
Internal interrupt*2		24 ⎮ 91	H'0060 to H'0063 ⎮ H'016C to H'016F

Notes: 1. Lower 16 bits of the address.

2. For details of internal interrupt vectors, see section 5.3.3, Interrupt Exception Vector Table.

4.2.1 Overview

A reset has the highest exception priority.

When the $\overline{\text{RES}}$ pin goes low, all processing halts and the chip enters the reset state. A reset initializes the internal state of the CPU and the registers of on-chip 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.

A reset can also be caused by watchdog timer overflow. For details see section 13, Watchdog Timer.

4.2.2 Reset Sequence

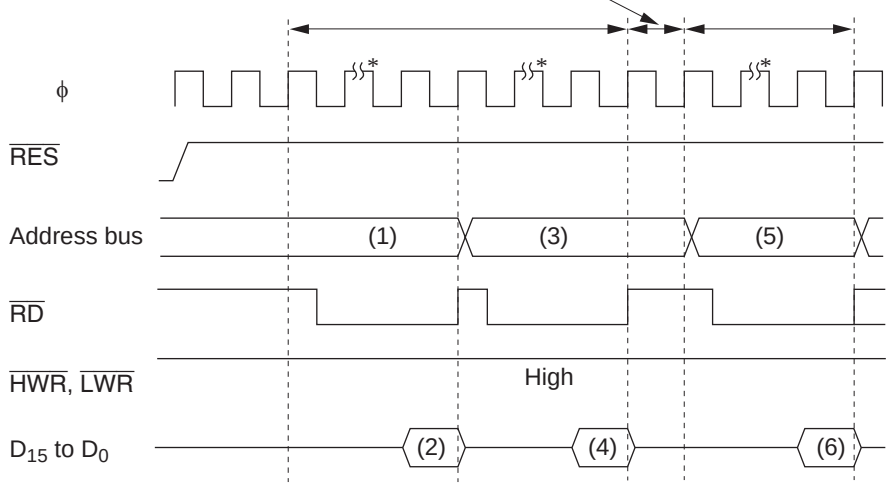
The chip enters the reset state when the $\overline{\text{RES}}$ pin goes low.

To ensure that the chip is reset, hold the $\overline{\text{RES}}$ pin low for at least 20 ms at power-up. To reset the 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 reset exception handling as follows:

1. The internal state of the CPU and the registers of the on-chip supporting modules are 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 vector address is read and transferred to the PC, and program execution starts from the address indicated by the PC.

Figure 4.2 shows an example of the reset sequence.



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

Note: * 3 program wait states are inserted.

Figure 4.2 Reset Sequence (Mode 4)

4.2.3 Interrupts after Reset

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

4.2.4 State of On-Chip Supporting Modules after Reset Release

After reset release, MSTPCR is initialized to H'3FFF and all modules except the DMAC and DTC enter module stop mode. Consequently, on-chip supporting module registers cannot be read or written to. Register reading and writing is enabled when module stop mode is exited.

Traces are enabled in interrupt control mode 2. Trace mode is not activated in interrupt control mode 0, irrespective of the state of the T bit. For details of interrupt control modes, see section 5, Interrupt Controller.

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

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

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 the trace 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	T
0	Trace exception handling cannot be used.			
2	1	—	—	0

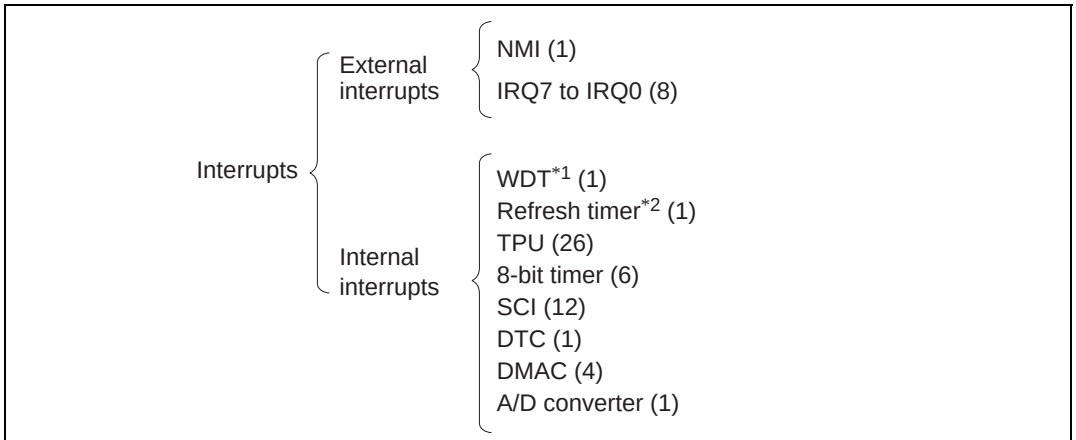
Legend:
1: Set to 1
0: Cleared to 0
—: Retains value prior to execution.

Interrupt exception handling can be requested by nine external sources (NMI, IRQ7 to IRQ0) and 52 internal sources in the on-chip supporting modules. Figure 4.3 classifies the interrupt sources and the number of interrupts of each type.

The on-chip supporting modules that can request interrupts include the watchdog timer (WDT), refresh timer, 16-bit timer-pulse unit (TPU), 8-bit timer, serial communication interface (SCI), data transfer controller (DTC), DMA controller (DMAC), and A/D converter. Each interrupt source has a separate vector address.

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

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



Notes: Numbers in parentheses are the numbers of interrupt sources.
1. When the watchdog timer is used as an interval timer, it generates an interrupt request at each counter overflow.
2. When the refresh timer is used as an interval timer, it generates an interrupt request at each compare match.

Figure 4.3 Interrupt Sources and Number of Interrupts

Trap instruction exception handling starts when a TRAPA instruction is executed. Trap instruction exception handling can be executed at all times in the program execution state.

The TRAPA instruction fetches a start address from a vector table entry corresponding to a vector number from 0 to 3, as specified in the instruction code.

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

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

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

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

4.6 Stack Status after Exception Handling

Figure 4.4 shows the stack after completion of trap instruction exception handling and interrupt exception handling.

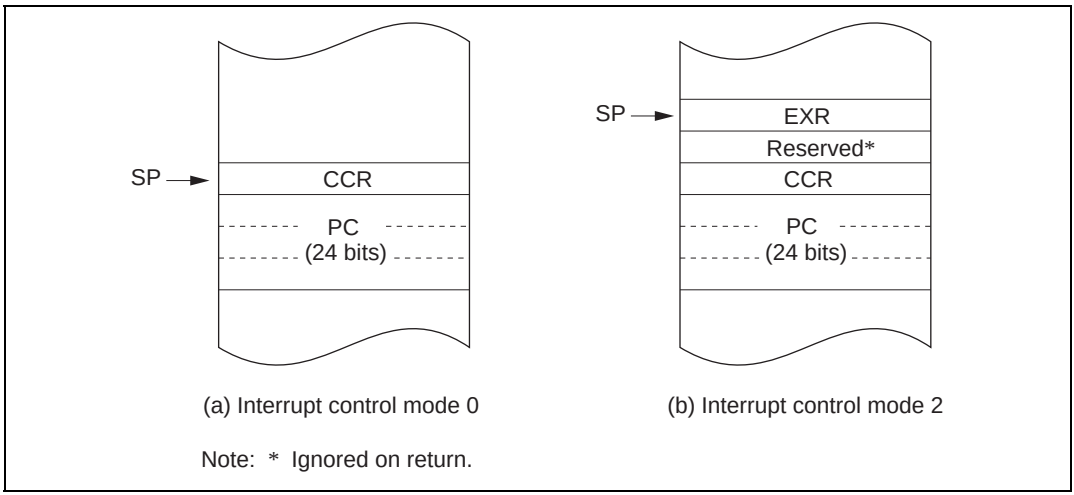


Figure 4.4 Stack Status after Exception Handling (Advanced Modes)

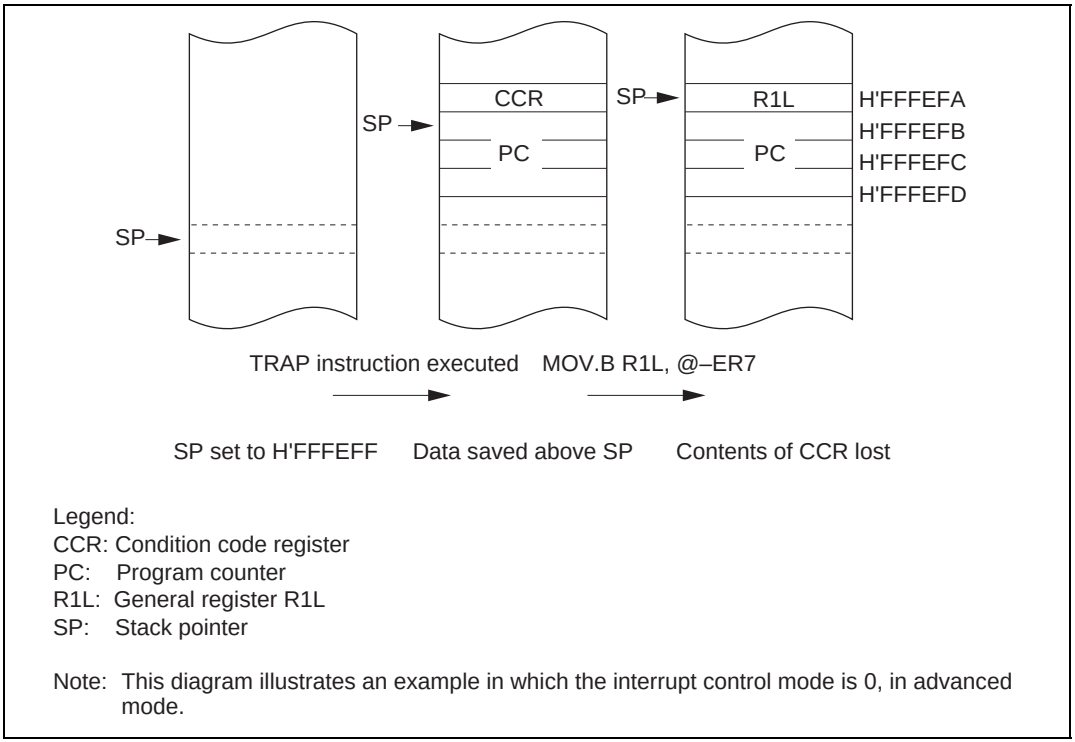
When accessing word data or longword data, the chip assumes that the lowest address bit is 0. The stack should always be accessed by word transfer instruction or longword transfer instruction, and the value of the stack pointer (SP, ER7) should always be kept even. Use the following instructions to save registers:

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

Use the following instructions to restore registers:

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

Setting SP to an odd value may lead to a malfunction. Figure 4.5 shows an example of what 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 4.5 Operation when SP Value is Odd

5.1 Overview

5.1.1 Features

The chip controls interrupts by means of an interrupt controller. The interrupt controller has the following features. This chapter assumes the maximum number of interrupt sources available in these series—nine external interrupts and 52 internal interrupts.

- Two interrupt control modes
  - Either of two interrupt control modes can be set by means of the INTM1 and INTM0 bits in the system control register (SYSCR)
- Priorities settable with IPRs
  - Interrupt priority registers (IPRs) are provided for setting interrupt priorities. Eight priority 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 unnecessary for the source to be identified in the interrupt handling routine
- Nine external interrupt pins
  - NMI is the highest-priority interrupt, and is accepted at all times. Rising edge or falling edge can be selected for NMI
  - Falling edge, rising edge, or both edge detection, or level sensing, can be selected for IRQ7 to IRQ0
- DTC and DMAC control
  - DTC and DMAC activation is controlled by means of interrupts

A block diagram of the interrupt controller is shown in figure 5.1.

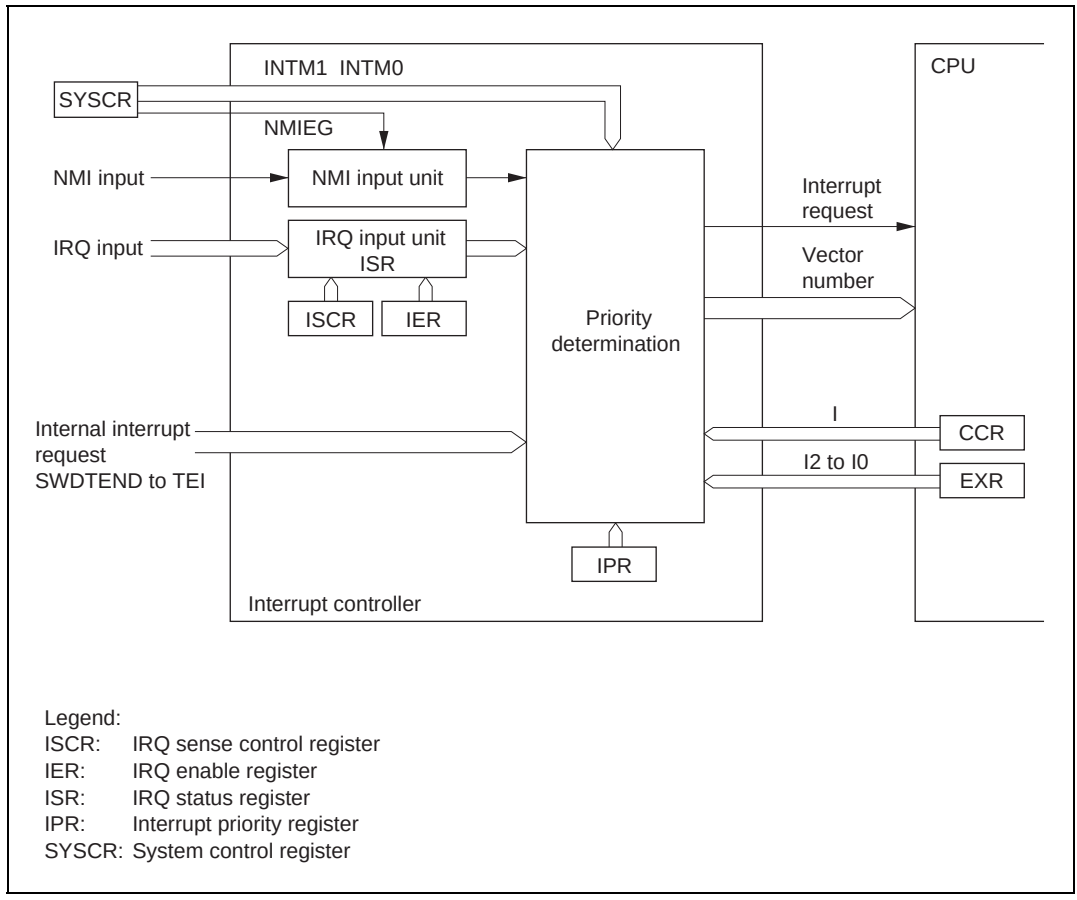


Figure 5.1 Block Diagram of Interrupt Controller

Table 5.1 summarizes the pins of the interrupt controller.

Table 5.1 Interrupt Controller Pins

Name	Symbol	I/O	Function
Nonmaskable interrupt	NMI	Input	Nonmaskable external interrupt; rising or falling edge can be selected
External interrupt requests 7 to 0	$\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$	Input	Maskable external interrupts; rising, falling, or both edges, or level sensing, can be selected

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	Address*1
System control register	SYSCR	R/W	H'01	H'FF39
IRQ sense control register H	ISCRH	R/W	H'00	H'FF2C
IRQ sense control register L	ISCRL	R/W	H'00	H'FF2D
IRQ enable register	IER	R/W	H'00	H'FF2E
IRQ status register	ISR	R/(W)*2	H'00	H'FF2F
Interrupt priority register A	IPRA	R/W	H'77	H'FEC4
Interrupt priority register B	IPRB	R/W	H'77	H'FEC5
Interrupt priority register C	IPRC	R/W	H'77	H'FEC6
Interrupt priority register D	IPRD	R/W	H'77	H'FEC7
Interrupt priority register E	IPRE	R/W	H'77	H'FEC8
Interrupt priority register F	IPRF	R/W	H'77	H'FEC9
Interrupt priority register G	IPRG	R/W	H'77	H'FECA
Interrupt priority register H	IPRH	R/W	H'77	H'FECB
Interrupt priority register I	IPRI	R/W	H'77	H'FECC
Interrupt priority register J	IPRJ	R/W	H'77	H'FECD
Interrupt priority register K	IPRK	R/W	H'77	H'FECE

Notes: 1. Lower 16 bits of the address.
2. Can only be written with 0 for flag clearing.

5.2.1 System Control Register (SYSCR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	INTM1	INTM0	NMIEG	LWROD	IRQPAS	RAME
Initial value	:	0	0	0	0	0	0	0	1
R/W	:	R/W	—	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 the detected edge for NMI.

Only bits 5 to 3, and 1 are described here; for details of the other bits, see section 3, MCU Operating Modes.

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

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

Bit 5 INTM1	Bit 4 INTM0	Interrupt Control Mode	Description	
0	0	0	Interrupts are controlled by I bit	(Initial value)
	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	(Initial value)
1	Interrupt request generated at rising edge of NMI input	

Bit 1—IRQ Input Pin Select (IRQPAS): Selects switching of the pins that can be used for input of $\overline{\text{IRQ4}}$ to $\overline{\text{IRQ7}}$. $\overline{\text{IRQ4}}$ to $\overline{\text{IRQ7}}$ input is always performed from one of the ports.

Bit	:	7	6	5	4	3	2	1	0
		—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0
Initial value	:	0	1	1	1	0	1	1	1
R/W	:	—	R/W	R/W	R/W	—	R/W	R/W	R/W

The IPR registers are eleven 8-bit readable/writable registers that set priorities (level 7 to 0) for 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.

Bits 7 and 3—Reserved: Read-only bits, always read as 0.

Table 5.3 Correspondence between Interrupt Sources and IPR Settings

	Bits	
Register	**6 to 4**	**2 to 0**
IPRA	IRQ0	IRQ1
IPRB	IRQ2 IRQ3	IRQ4 IRQ5
IPRC	IRQ6 IRQ7	DTC
IPRD	Watchdog timer	Refresh timer
IPRE	—*	A/D converter
IPRF	TPU channel 0	TPU channel 1
IPRG	TPU channel 2	TPU channel 3
IPRH	TPU channel 4	TPU channel 5
IPRI	8-bit timer channel 0	8-bit timer channel 1
IPRJ	DMAC	SCI channel 0
IPRK	SCI channel 1	SCI channel 2

Note: * Reserved bits.

interrupt. The lowest priority level, level 0, is assigned by setting H'0, and the highest priority level, level 7, by setting H'7.

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

5.2.3 IRQ Enable Register (IER)

Bit	:	7	6	5	4	3	2	1	0
		IRQ7E	IRQ6E	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IRQ0E
Initial value	:	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

IER is an 8-bit readable/writable register that controls enabling and disabling of interrupt requests IRQ7 to IRQ0.

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

Bits 7 to 0—IRQ7 to IRQ0 Enable (IRQ7E to IRQ0E): These bits select whether IRQ7 to IRQ0 are enabled or disabled.

Bit n IRQnE	Description	
0	IRQn interrupts disabled	(Initial value)
1	IRQn interrupts enabled	

(n = 7 to 0)

ISCRH

Bit	:	15	14	13	12	11	10	9	8
		IRQ7SCB	IRQ7SCA	IRQ6SCB	IRQ6SCA	IRQ5SCB	IRQ5SCA	IRQ4SCB	IRQ4SCA
Initial value	:	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

ISCRL

Bit	:	7	6	5	4	3	2	1	0
		IRQ3SCB	IRQ3SCA	IRQ2SCB	IRQ2SCA	IRQ1SCB	IRQ1SCA	IRQ0SCB	IRQ0SCA
Initial value	:	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

ISCR (composed of ISCRH and ISCRL) is a 16-bit readable/writable register that selects rising edge, falling edge, or both edge detection, or level sensing, for the input at pins $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$.

ISCR is initialized to H'0000 by a reset and in hardware standby mode.

Bits 15 to 0—IRQ7 Sense Control A and B (IRQ7SCA, IRQ7SCB) to IRQ0 Sense Control A and B (IRQ0SCA, IRQ0SCB)

Bits 15 to 0

IRQ7SCB to IRQ0SCB	IRQ7SCA to IRQ0SCA	Description
0	0	Interrupt request generated at $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$ input low level (Initial value)
	1	Interrupt request generated at falling edge of $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$ input
1	0	Interrupt request generated at rising edge of $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$ input
	1	Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$ input

Bit	:	7	6	5	4	3	2	1	0
		IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F
Initial value	:	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)*

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

ISR is an 8-bit readable/writable register that indicates the status of IRQ7 to IRQ0 interrupt requests.

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

Bits 7 to 0—IRQ7 to IRQ0 flags (IRQ7F to IRQ0F): These bits indicate the status of IRQ7 to IRQ0 interrupt requests.

Bit n IRQnF	Description
0	[Clearing conditions] (Initial value) • Cleared by reading IRQnF flag when IRQnF = 1, then writing 0 to IRQnF flag • When interrupt exception handling including other interrupt exception handling is executed when low-level detection is set (IRQnSCB = IRQnSCA = 0) and $\overline{\text{IRQn}}$ input is high • When IRQn interrupt exception handling is executed when falling, rising, or both-edge detection is set (IRQnSCB = 1 or IRQnSCA = 1) • When the DTC is activated by an IRQn interrupt, and the DISEL 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 set (IRQnSCB = IRQnSCA = 0) • When a falling edge occurs in $\overline{\text{IRQn}}$ input when falling edge detection is set (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 detection is set (IRQnSCB = IRQnSCA = 1)

(n = 7 to 0)

Interrupt sources comprise external interrupts (NMI and IRQ7 to IRQ0) and internal interrupts (52 sources).

5.3.1 External Interrupts

There are nine external interrupts: NMI and IRQ7 to IRQ0. NMI and IRQ7 to IRQ0 can be used to restore the chip from software standby mode. (IRQ7 to IRQ3 can be designated for use as software standby mode clearing sources by setting the IRQ37S bit in SBYCR to 1.)

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

The vector number for NMI interrupt exception handling is 7.

IRQ7 to IRQ0 Interrupts: Interrupts IRQ7 to IRQ0 are requested by an input signal at pins $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$. Interrupts IRQ7 to IRQ0 have the following features:

- Using ISCR, it is possible to select whether an interrupt is generated by a low level, falling edge, rising edge, or both edges, at pins $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$.
- Enabling or disabling of interrupt requests IRQ7 to IRQ0 can be selected with IER.
- The interrupt priority level can be set with IPR.
- The status of interrupt requests IRQ7 to IRQ0 is indicated in ISR. ISR flags can be cleared to 0 by software.

A block diagram of interrupts IRQ7 to IRQ0 is shown in figure 5.2.

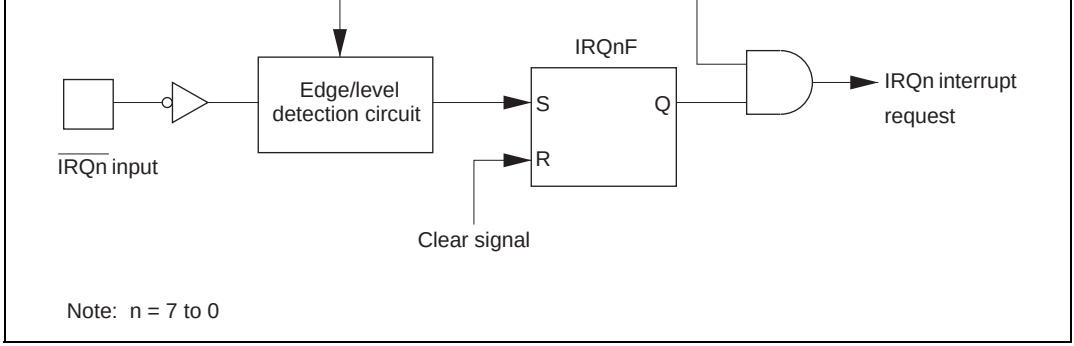


Figure 5.2 Block Diagram of Interrupts IRQ7 to IRQ0

Figure 5.3 shows the timing of setting IRQnF.

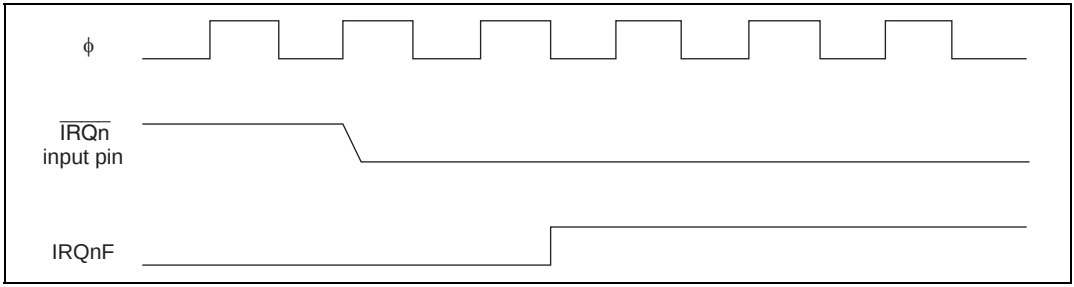


Figure 5.3 Timing of Setting IRQnF

The vector numbers for IRQ7 to IRQ0 interrupt exception handling are 23 to 16.

Detection of IRQ7 to IRQ0 interrupts does not depend on whether the relevant pin has been set for input or output. Therefore, when a pin is used as an external interrupt input pin, do not clear the corresponding DDR bit to 0 and use the pin as an I/O pin for another function. The pins that can be used for IRQ4 to IRQ7 interrupt input can be switched by means of the IRQPAS bit in SYSCR.

There are 52 sources for internal interrupts from on-chip supporting modules.

- For each on-chip supporting module there are flags that indicate the interrupt request status, and enable bits that select enabling or disabling of these interrupts. If both of these are set to 1 for a particular interrupt source, an interrupt request is issued to the interrupt controller.
- The interrupt priority level can be set by means of IPR.
- The DMAC and DTC can be activated by a TPU, SCI, or other interrupt request. When the DMAC or DTC is activated by an interrupt, the interrupt control mode and interrupt mask bits have no effect.

5.3.3 Interrupt Exception Vector Table

Table 5.4 shows interrupt exception handling sources, vector addresses, and interrupt priorities. For default priorities, the lower the vector number, the higher the priority. Interrupt sources can also be used to activate the DTC and DMAC.

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

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address*	IPR	Priority	DTC Activation	DMAC Activation
Power-on reset		0	H'0000	—	High	—	—
Reserved		1	H'0004				
Reserved for system use		2	H'0008				
		3	H'000C				
		4	H'0010				
Trace		5	H'0014				
Reserved for system use		6	H'0018				
NMI	External pin	7	H'001C				
Trap instruction (4 sources)		8	H'0020				
		9	H'0024				
		10	H'0028				
		11	H'002C				
Reserved for system use		12	H'0030				
		13	H'0034				
		14	H'0038				
		15	H'003C				
IRQ_0	External pin	16	H'0040	IPRA6 to IPRA4		○	—
IRQ_1		17	H'0044	IPRA2 to IPRA0		○	—
IRQ_2		18	H'0048	IPRB6 to IPRB4		○	—
IRQ_3		19	H'004C			○	—
IRQ_4		20	H'0050	IPRB2 to IPRB0		○	—
IRQ_5		21	H'0054			○	—
IRQ_6		22	H'0058	IPRC6 to IPRC4		○	—
IRQ_7		23	H'005C		Low	○	—

Interrupt Source	Source	Number	Address	IPR	Priority	tion	tion
SWDTEND (software-activated data transfer end)	DTC	24	H'0060	IPRC2 to IPRC0	High	○	—
WOVI (interval timer)	Watchdog timer	25	H'0064	IPRD6 to IPRD4		—	—
CMI (compare match)	Refresh controller	26	H'0068	IPRD2 to IPRD0		—	—
Reserved	—	27	H'006C	IPRE6 to IPRE4		—	—
ADI (A/D conversion end)	A/D	28	H'0070	IPRE2 to IPRE0		○	○
Reserved	—	29	H'0074			—	—
		30	H'0078				
		31	H'007C				
TGI0A (TGR0A input capture/compare match)	TPU channel 0	32	H'0080	IPRF6 to IPRF4		○	○
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		Low		

Interrupt Source	Source	Number	Address	IPR	Priority	tion	tion
TGI1A (TGR1A input capture/compare match)	TPU channel 1	40	H'00A0	IPRF2 to IPRF0	High	○	○
TGI1B (TGR1B input capture/compare match)		41	H'00A4			○	—
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 IPRG4		○	○
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 IPRG0		○	○
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	H'00D4			—	—
		54	H'00D8				
		55	H'00DC		Low		

Interrupt Source	Source	Number	Address	IPR	Priority	tion	tion
TGI4A (TGR4A input capture/compare match)	TPU channel 4	56	H'00E0	IPRH6 to IPRH4	High	○	○
TGI4B (TGR4B input capture/compare match)		57	H'00E4			○	—
TCI4V (overflow 4)		58	H'00E8			—	—
TCI4U (underflow 4)		59	H'00EC			—	—
TGI5A (TGR5A input capture/compare match)	TPU channel 5	60	H'00F0	IPRH2 to IPRH0		○	○
TGI5B (TGR5B input capture/compare match)		61	H'00F4			○	—
TCI5V (overflow 5)		62	H'00F8			—	—
TCI5U (underflow 5)		63	H'00FC			—	—
CMIA0 (compare match A)	8-bit timer channel 0	64	H'0100	IPRI6 to IPRI4		○	—
CMIB0 (compare match B)		65	H'0104			○	—
OVI0 (overflow 0)		66	H'0108			—	—
Reserved	—	67	H'010C			—	—
CMIA1 (compare match A)	8-bit timer channel 1	68	H'0110	IPRI2 to IPRI0		○	—
CMIB1 (compare match B)		69	H'0114			○	—
OVI1 (overflow 1)		70	H'0118			—	—
Reserved	—	71	H'011C		Low	—	—

Interrupt Source	Source	Number	Address	IPR	Priority	tion	tion
DEND0A (channel 0/channel 0A transfer end)	DMAC	72	H'0120	IPRJ6 to IPRJ4	High	○	—
DEND0B (channel 0B transfer end)		73	H'0124			○	—
DEND1A (channel 1/channel 1A transfer end)		74	H'0128			○	—
DEND1B (channel 1B transfer end)		75	H'012C			○	—
Reserved	—	76	H'0130			—	—
		77	H'0134				
		78	H'0138				
		79	H'013C				
ERI0 (receive error 0)	SCI channel 0	80	H'0140	IPRJ2 to IPRJ0		—	—
RXI0 (receive-data-full 0)		81	H'0144			○	○
TXI0 (transmit-data-empty 0)		82	H'0148			○	○
TEI0 (transmit end 0)		83	H'014C			—	—
ERI1 (receive error 1)	SCI channel 1	84	H'0150	IPRK6 to IPRK4		—	—
RXI1 (receive-data-full 1)		85	H'0154			○	○
TXI1 (transmit data empty 1)		86	H'0158			○	○
TEI1 (transmit end 1)		87	H'015C			—	—
ERI2 (receive error 2)	SCI channel 2	88	H'0160	IPRK2 to IPRK0		—	—
RXI2 (receive-data-full 2)		89	H'0164			○	—
TXI2 (transmit-data-empty 2)		90	H'0168			○	—
TEI2 (transmit end 2)		91	H'016C		Low	—	—

Note: * Lower 16 bits of the start address.

5.4.1 Interrupt Control Modes and Interrupt Operation

Interrupt operations in the chip differ depending on the interrupt control mode.

NMI interrupts are accepted at all times except in the reset state and the hardware standby state. In the case of IRQ interrupts and on-chip supporting module interrupts, an enable bit is provided for each interrupt. Clearing an enable bit to 0 disables the corresponding interrupt request. Interrupt 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 mode set by the INTM1 and INTM0 bits in SYSCR, the priorities set in IPR, and the masking state indicated by the I bit in the CPU's CCR, and bits I2 to I0 in EXR.

Table 5.5 Interrupt Control Modes

Interrupt Control Mode	SYSCR		Priority Setting Registers	Interrupt Mask Bits	Description
	INTM1	INTM0			
0	0	0	—	I	Interrupt mask control is performed by the I bit.
—		1	—	—	Setting prohibited
2	1	0	IPR	I2 to I0	8-level interrupt mask control is performed by bits I2 to I0. 8 priority levels can be set with IPR.
—		1	—	—	Setting prohibited

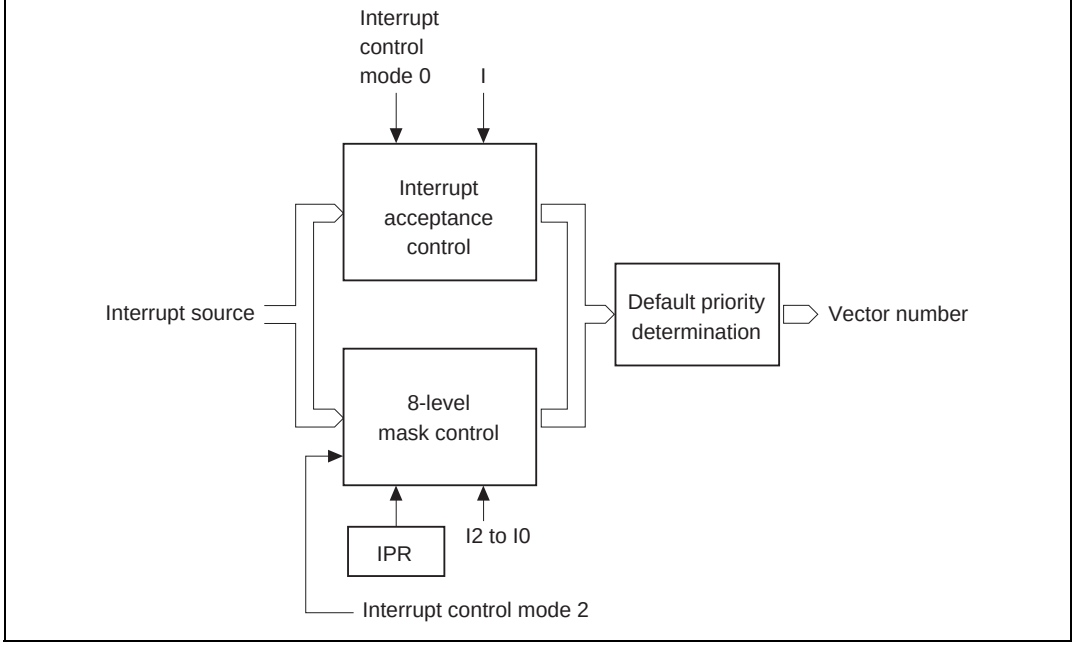


Figure 5.4 Block Diagram of Interrupt Control Operation

Interrupt Acceptance Control: In interrupt control mode 0, interrupt acceptance is controlled by the I bit in CCR.

Table 5.6 shows the interrupts selected in each interrupt control mode.

Table 5.6 Interrupts Selected in Each Interrupt Control Mode (1)

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

*: Don't care

(IPR).

The interrupt source selected is the interrupt with the highest priority level, and whose priority 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 greater than the mask level (IPR > I2 to I0)

Default Priority Determination: When an interrupt is selected by 8-level control, its priority is determined and a vector number is generated.

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

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

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

Table 5.8 Operations and Control Signal Functions in Each Interrupt Control Mode

Interrupt Control Mode	Setting		Interrupt Acceptance Control		8-Level Control			Default Priority Determination	T (Trace)
	INTM1	INTM0		I		I2 to I0	IPR		
0	0	0	○	IM	X	—	—*2	○	—
2	1	0	X	—*1	○	IM	PR	○	T

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.

Enabling and disabling of IRQ interrupts and on-chip supporting module interrupts can be set by means of the I bit in the CPU's CCR. Interrupts are enabled when the I bit is cleared to 0, and disabled when set to 1.

Figure 5.5 shows a flowchart of the interrupt acceptance operation in this case.

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

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

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

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

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

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

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

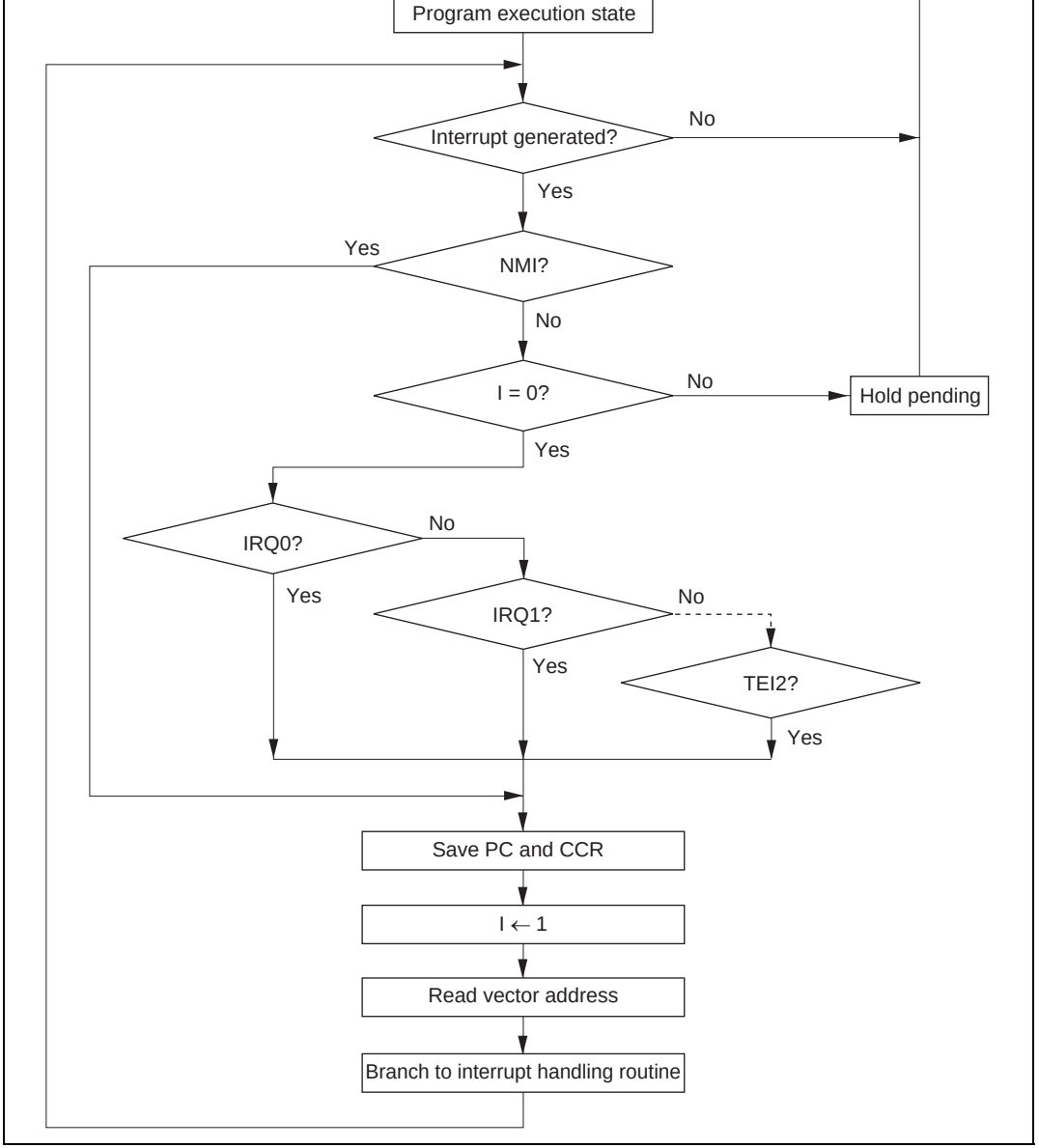


Figure 5.5 Flowchart of Procedure Up to Interrupt Acceptance in Interrupt Control Mode 0

Eight-level masking is implemented for IRQ interrupts and on-chip supporting module interrupts by comparing the interrupt mask level set by bits I2 to I0 of EXR in the CPU with IPR.

Figure 5.6 shows a flowchart of the interrupt acceptance operation in this case.

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

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

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

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

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

[6] The T bit in EXR is cleared to 0. The interrupt mask level is rewritten with the priority level of 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 interrupt handling routine starts at the address indicated by the contents of that vector address.

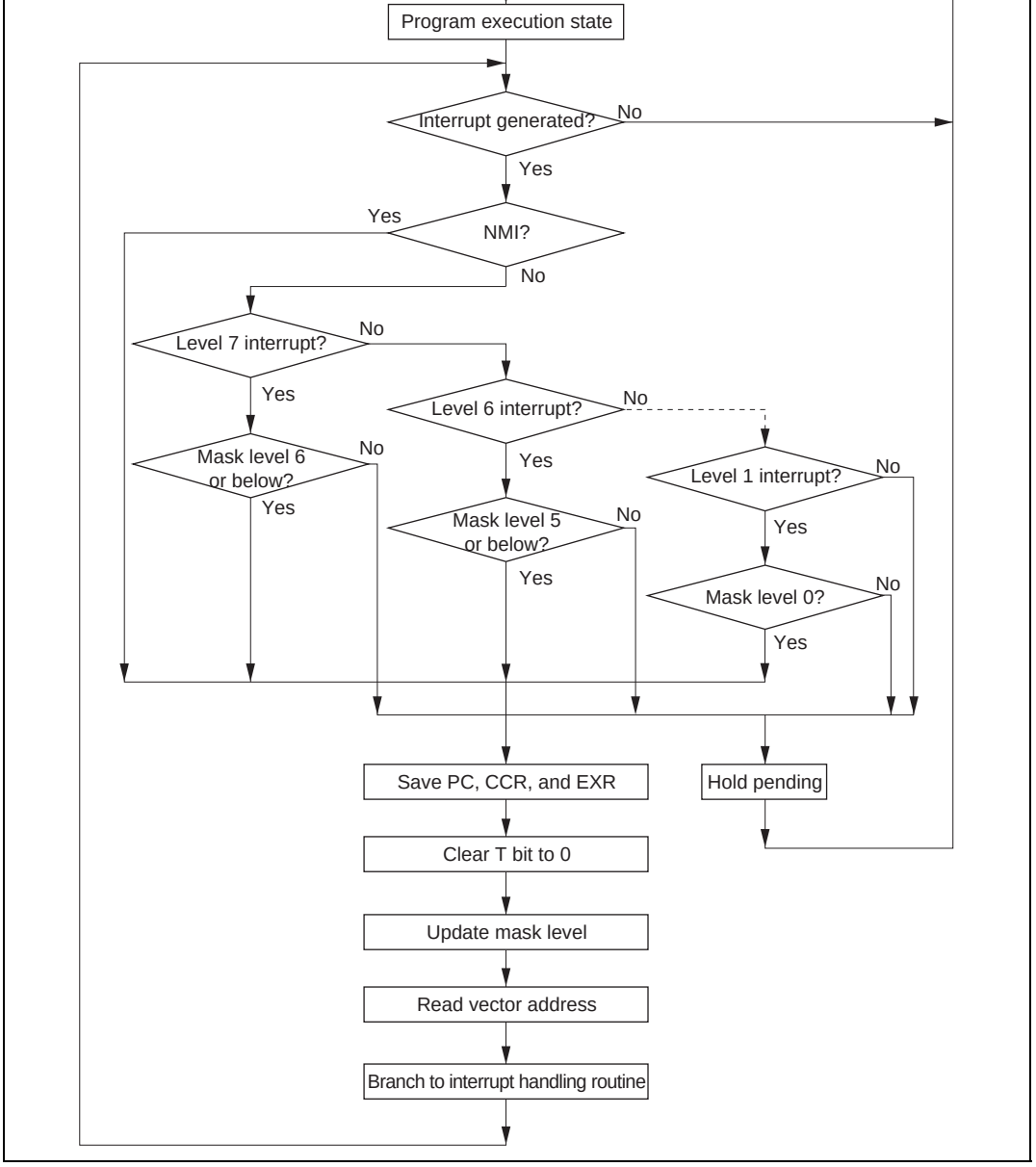


Figure 5.6 Flowchart of Procedure Up to Interrupt Acceptance in Interrupt Control Mode 2

Figure 5.7 shows the interrupt exception handling sequence. The example shown is for the case where interrupt control mode 0 is set in advanced mode, and the program area and stack area are in on-chip memory.

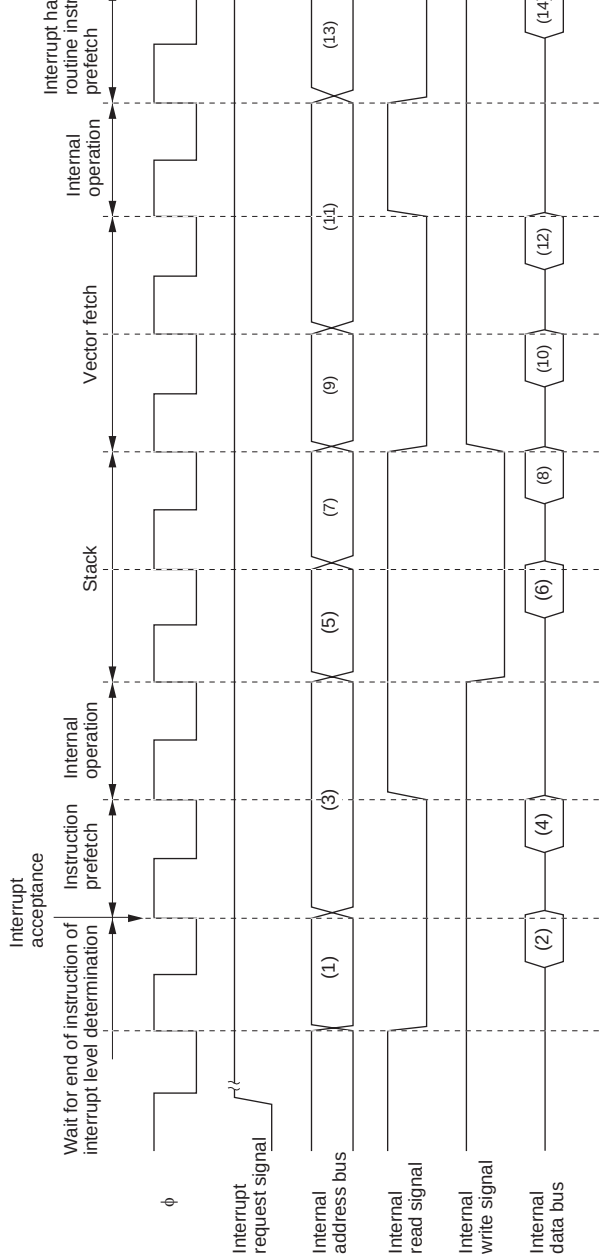


Figure 5.7 Interrupt Exception Handling

The chip is capable of fast word transfer instruction to on-chip memory, and the program area is provided in on-chip ROM and the stack area in on-chip RAM, enabling high-speed processing.

Table 5.9 shows interrupt response times—the interval between generation of an interrupt request and execution of the first instruction in the interrupt handling routine. The execution status symbols used in table 5.9 are explained in table 5.10.

Table 5.9 Interrupt Response Times

		Advanced Mode	
No.	**Item**	**INTM1 = 0**	**INTM1 = 1**
1	Interrupt priority determination*1	3	3
2	Number of wait states until executing instruction ends*2	1 to (19 + 2·S_I)	1 to (19 + 2·S_I)
3	PC, CCR, EXR stack save	2·S_K	3·S_K
4	Vector fetch	2·S_I	2·S_I
5	Instruction fetch*3	2·S_I	2·S_I
6	Internal processing*4	2	2
Total (using on-chip memory)		12 to 32	13 to 33

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 vector fetch.

Table 5.10 Number of States in Interrupt Handling Routine Execution

		Object of Access				
			External Device			
			8-Bit Bus		16-Bit Bus	
Symbol		**Internal Memory**	**2-State Access**	**3-State Access**	**2-State Access**	**3-State Access**
Instruction fetch	S_I	1	4	6 + 2m	2	3 + m
Branch address read	S_J					
Stack manipulation	S_K					

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

5.5.1 Contention between Interrupt Generation and Disabling

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

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

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

Figure 5.8 shows an example in which the TGIEA bit in the TPU's TIER0 register is cleared to 0.

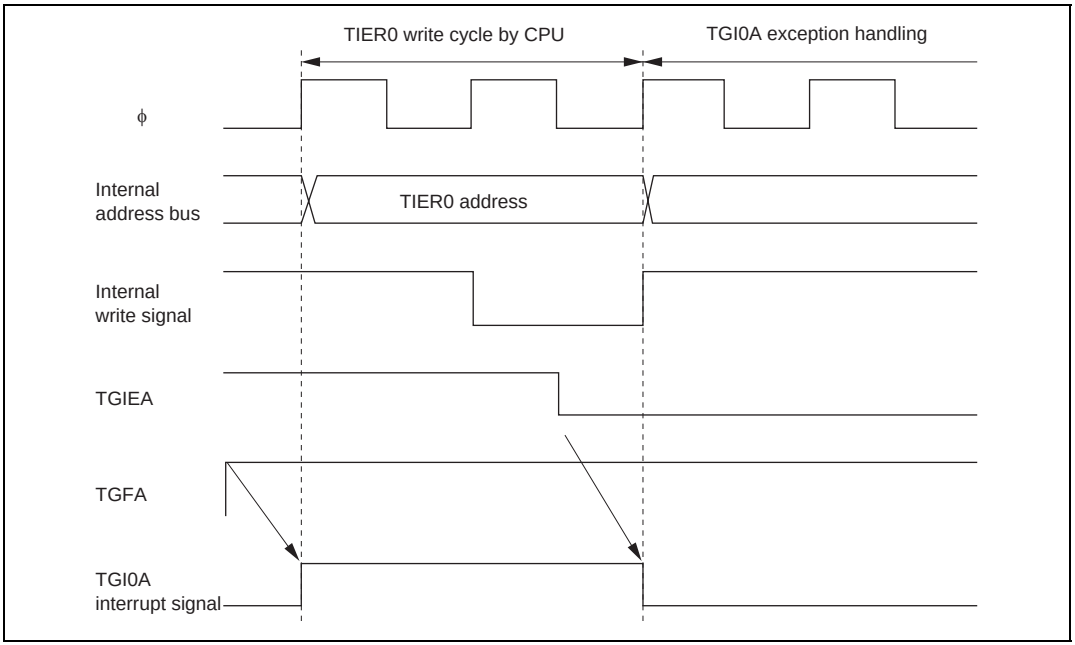


Figure 5.8 Contention between Interrupt Generation and Disabling

The above contention will not occur if an enable bit or interrupt source flag is cleared to 0 while the interrupt is masked.

Instructions that disable interrupts are LDC, ANDC, ORC, and XORC. After any of these instructions is executed, all interrupts including NMI are disabled and the next instruction is always executed. When the I bit is set by one of these instructions, the new value becomes valid 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 has updated the mask level with an LDC, ANDC, ORC, or XORC instruction.

5.5.4 Interrupts during Execution of EEPMOV Instruction

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

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

With the EEPMOV.W instruction, if an interrupt request is issued during the transfer, interrupt exception handling starts at a break in the transfer cycle. The PC value saved on the stack in this case is the address of the next instruction.

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

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

5.6.1 Overview

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

1. Interrupt request to CPU
2. Activation request to DTC
3. Activation request to DMAC
4. Selection of a number of the above

For details of interrupt requests that can be used with to activate the DTC or DMAC, see section 8, Data Transfer Controller, and section 7, DMA Controller.

Figure 5.9 shows a block diagram of the DTC, DMAC, and interrupt controller.

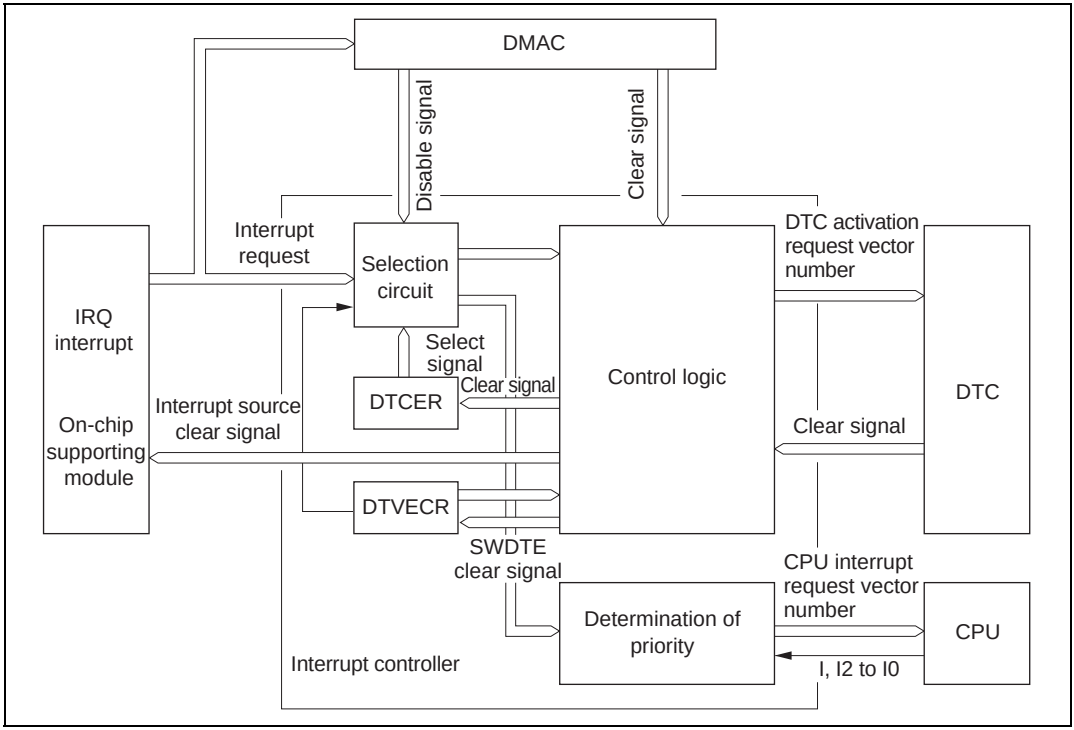


Figure 5.9 Interrupt Control for DTC and DMAC

The interrupt controller has three main functions in DTC and DMAC control.

Selection of Interrupt Source: With the DMAC, the activation source is input directly to each channel. The activation source for each DMAC channel is selected with bits DTF3 to DTF0 in DMACR. Whether the selected activation source is to be managed by the DMAC can be selected with the DTA bit of DMABCR. When the DTA bit is set to 1, the interrupt source constituting that DMAC activation source is not a DTC activation source or CPU interrupt source.

For interrupt sources other than interrupts managed by the DMAC, it is possible to select DTC activation request or CPU interrupt request with the DTCE bit of DTCERA to DTCERF in the DTC.

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

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

Determination of Priority: The DTC activation source is selected in accordance with the default priority order, and is not affected by mask or priority levels. See section 7.6, Interrupts, and section 8.3.3, DTC Vector Table, for the respective priorities.

With the DMAC, the activation source is input directly to each channel.

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

If the same interrupt is selected as a DMAC activation source and a DTC activation source or CPU interrupt source, operations are performed for them independently according to their respective operating statuses and bus mastership priorities.

Table 5.11 summarizes interrupt source selection and interrupt source clearance control according to the settings of the DTA bit of DMABCR in the DMAC, the DTCE bit of DTCERA to DTCERF in the DTC, and the DISEL bit of MRB in the DTC.

Settings					
DMAC	**DTC**		**Interrupt Source Selection/Clearing Control**		
DTA	**DTCE**	**DISEL**	**DMAC**	**DTC**	**CPU**
0	0	*	○	X	◎
	1	0	○	◎	X
		1	○	○	◎
1	*	*	◎	X	X

Legend:

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

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

X: The relevant interrupt cannot be used.

*: Don't care

Usage Note: SCI and A/D converter interrupt sources are cleared when the DMAC or DTC reads or writes to the prescribed register, and are not dependent upon the DTA bit or DISEL bit.

6.1 Overview

The chip has an on-chip bus controller (BSC) that manages the external address space divided into eight areas. The bus specifications, such as bus width and number of access states, can be set independently for each area, enabling multiple memories to be connected easily.

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

6.1.1 Features

The features of the bus controller are listed below.

- Manages external address space in area units
  - In advanced mode, manages the external space as 8 areas of 2 Mbytes
  - Bus specifications can be set independently for each area
  - DRAM and burst ROM interfaces can be set
- Basic bus interface
  - Chip select signals ($\overline{CS}_0$ to $\overline{CS}_7$) can be output for areas 0 to 7
  - 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
- DRAM interface
  - DRAM interface can be set for areas 2 to 5 (in advanced mode)
  - Row address/column address multiplexed output (8/9/10 bits)
  - 2-CAS access method
  - Burst operation (fast page mode)
  - TP cycle insertion to secure RAS precharging time
  - Selection of CAS-before-RAS refreshing or self-refreshing
- Burst ROM interface
  - Burst ROM interface can be set for area 0
  - Selection of 1- or 2-state burst access

— An idle cycle can be inserted in case of an external write cycle immediately after an external read cycle

- Write buffer function
  — External write cycle and internal access can be executed in parallel
  — DMAC single address mode and internal access can be executed in parallel
- Bus arbitration function
  — Includes a bus arbiter that arbitrates bus mastership between the CPU, DMAC, and DTC
- Other features
  — Refresh counter (refresh timer) can be used as an interval timer
  — External bus release function

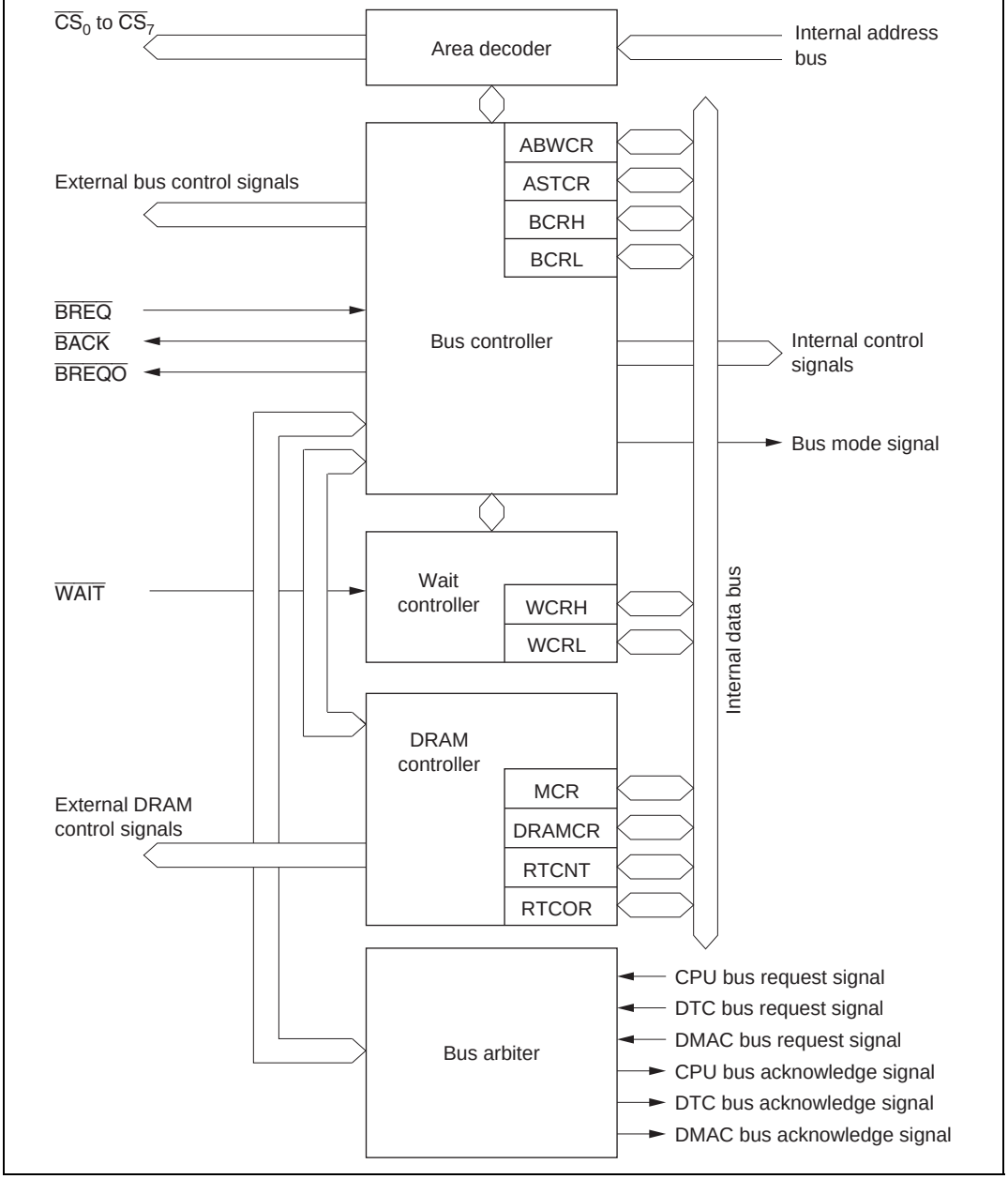


Figure 6.1 Block Diagram of Bus Controller

Table 6.1 summarizes the pins of the bus controller.

Table 6.1 Bus Controller Pins

Name	Symbol	I/O	Function
Address strobe	$\overline{AS}$	Output	Strobe signal indicating that address output on address bus is enabled.
Read	$\overline{RD}$	Output	Strobe signal indicating that external space is being read.
High write/write enable	$\overline{HWR}$	Output	Strobe signal indicating that external space is to be written, and upper half (D_{15} to D_8) of data bus is enabled. 2-CAS DRAM write enable signal.
Low write	$\overline{LWR}$	Output	Strobe signal indicating that external space is to be written, and lower half (D_7 to D_0) of data bus is enabled.
Chip select 0	$\overline{CS_0}$	Output	Strobe signal indicating that area 0 is selected.
Chip select 1	$\overline{CS_1}$	Output	Strobe signal indicating that area 1 is selected.
Chip select 2/row address strobe 2	$\overline{CS_2}$	Output	Strobe signal indicating that area 2 is selected. DRAM row address strobe signal when area 2 is in DRAM space.
Chip select 3/row address strobe 3	$\overline{CS_3}$	Output	Strobe signal indicating that area 3 is selected. DRAM row address strobe signal when area 3 is in DRAM space.
Chip select 4/row address strobe 4	$\overline{CS_4}$	Output	Strobe signal indicating that area 4 is selected. DRAM row address strobe signal when area 4 is in DRAM space.
Chip select 5/row address strobe 5	$\overline{CS_5}$	Output	Strobe signal indicating that area 5 is selected. DRAM row address strobe signal when area 5 is in DRAM space.
Chip select 6	$\overline{CS_6}$	Output	Strobe signal indicating that area 6 is selected.
Chip select 7	$\overline{CS_7}$	Output	Strobe signal indicating that area 7 is selected.
Upper column address strobe	$\overline{CAS}$	Output	2-CAS DRAM upper column address strobe signal.
Lower column address strobe	$\overline{LCAS}$	Output	DRAM lower column address strobe signal.

			state access space.
Bus request	$\overline{\text{BREQ}}$	Input	Request signal for release of bus to external device.
Bus request acknowledge	$\overline{\text{BACK}}$	Output	Acknowledge signal indicating that bus has been released.
Bus request output	$\overline{\text{BREQO}}$	Output	External bus request signal used when internal bus master accesses external space when external bus is released.

6.1.4 Register Configuration

Table 6.2 summarizes the registers of the bus controller.

Table 6.2 Bus Controller Registers

Name	Abbreviation	R/W	Initial Value Reset	Address*1
Bus width control register	ABWCR	R/W	H'FF/H'00*2	H'FED0
Access state control register	ASTCR	R/W	H'FF	H'FED1
Wait control register H	WCRH	R/W	H'FF	H'FED2
Wait control register L	WCRL	R/W	H'FF	H'FED3
Bus control register H	BCRH	R/W	H'D0	H'FED4
Bus control register L	BCRL	R/W	H'3C	H'FED5
Memory control register	MCR	R/W	H'00	H'FED6
DRAM control register	DRAMCR	R/W	H'00	H'FED7
Refresh timer counter	RTCNT	R/W	H'00	H'FED8
Refresh time constant register	RTCOR	R/W	H'FF	H'FED9

Notes: 1. Lower 16 bits of the address.
2. Determined by the MCU operating mode.

6.2.1 Bus Width Control Register (ABWCR)

Bit	:	7	6	5	4	3	2	1	0
		ABW7	ABW6	ABW5	ABW4	ABW3	ABW2	ABW1	ABW0
Modes 5 to 7									
Initial value	:	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
Mode 4									
Initial value	:	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

ABWCR is an 8-bit readable/writable register that designates each area as either 8-bit access space or 16-bit access space.

ABWCR sets the data bus width for the external memory space. The bus width for on-chip 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 to 7*, and to H'00 in mode 4. It is not initialized in software standby mode.

Note: * Modes 6 and 7 cannot be used in the ROMless version.

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

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

(n = 7 to 0)

Bit	:	7	6	5	4	3	2	1	0
		AST7	AST6	AST5	AST4	AST3	AST2	AST1	AST0
Initial value	:	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

ASTCR is an 8-bit readable/writable register that designates each area as either 2-state access space or 3-state access space.

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

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

Bits 7 to 0—Area 7 to 0 Access State Control (AST7 to AST0): These bits select whether the corresponding area is to be designated as 2-state access space or 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 access is disabled	
1	Area n is designated for 3-state access Wait state insertion in area n external space access is enabled	(Initial value)

(n = 7 to 0)

6.2.3 Wait Control Registers H and L (WCRH, WCRL)

WCRH and WCRL are 8-bit readable/writable registers that select the number of program wait states for each area.

Program waits are not inserted in on-chip memory or internal I/O register access.

WCRH and WCRL are initialized to H'FF by a reset, and in hardware standby mode. They are not initialized in software standby mode.

Bit	:	7	6	5	4	3	2	1	0
		W71	W70	W61	W60	W51	W50	W41	W40
Initial value	:	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

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

Bit 7 W71	Bit 6 W70	Description
0	0	Program wait not inserted when external space area 7 is accessed
	1	1 program wait state inserted when external space area 7 is accessed
1	0	2 program wait states inserted when external space area 7 is accessed
	1	3 program wait states inserted when external space area 7 is accessed (Initial value)

Bits 5 and 4—Area 6 Wait Control 1 and 0 (W61, W60): These bits select the number of program wait states when area 6 in external space is accessed while the AST6 bit in ASTCR is set to 1.

Bit 5 W61	Bit 4 W60	Description
0	0	Program wait not inserted when external space area 6 is accessed
	1	1 program wait state inserted when external space area 6 is accessed
1	0	2 program wait states inserted when external space area 6 is accessed
	1	3 program wait states inserted when external space area 6 is accessed (Initial value)

to 1.

Bit 3 W51	Bit 2 W50	Description
0	0	Program wait not inserted when external space area 5 is accessed
	1	1 program wait state inserted when external space area 5 is accessed
1	0	2 program wait states inserted when external space area 5 is accessed
	1	3 program wait states inserted when external space area 5 is accessed (Initial value)

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

Bit 1 W41	Bit 0 W40	Description
0	0	Program wait not inserted when external space area 4 is accessed
	1	1 program wait state inserted when external space area 4 is accessed
1	0	2 program wait states inserted when external space area 4 is accessed
	1	3 program wait states inserted when external space area 4 is accessed (Initial value)

Bit	:	7	6	5	4	3	2	1	0
		W31	W30	W21	W20	W11	W10	W01	W00
Initial value	:	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

Bits 7 and 6—Area 3 Wait Control 1 and 0 (W31, W30): These bits select the number of program wait states when area 3 in external space is accessed while the AST3 bit in ASTCR is set to 1.

Bit 7 W31	Bit 6 W30	Description
0	0	Program wait not inserted when external space area 3 is accessed
	1	1 program wait state inserted when external space area 3 is accessed
1	0	2 program wait states inserted when external space area 3 is accessed
	1	3 program wait states inserted when external space area 3 is accessed (Initial value)

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

Bit 5 W21	Bit 4 W20	Description
0	0	Program wait not inserted when external space area 2 is accessed
	1	1 program wait state inserted when external space area 2 is accessed
1	0	2 program wait states inserted when external space area 2 is accessed
	1	3 program wait states inserted when external space area 2 is accessed (Initial value)

to 1.

Bit 3 W11	Bit 2 W10	Description
0	0	Program wait not inserted when external space area 1 is accessed
	1	1 program wait state inserted when external space area 1 is accessed
1	0	2 program wait states inserted when external space area 1 is accessed
	1	3 program wait states inserted when external space area 1 is accessed (Initial value)

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

Bit 1 W01	Bit 0 W00	Description
0	0	Program wait not inserted when external space area 0 is accessed
	1	1 program wait state inserted when external space area 0 is accessed
1	0	2 program wait states inserted when external space area 0 is accessed
	1	3 program wait states inserted when external space area 0 is accessed (Initial value)

Bit	:	7	6	5	4	3	2	1	0
		ICIS1	ICIS0	BRSTRM	BRSTS1	BRSTS0	RMTS2	RMTS1	RMTS0
Initial value	:	1	1	0	1	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BCRH is an 8-bit readable/writable register that selects enabling or disabling of idle cycle insertion, and the memory interface for areas 2 to 5 and area 0.

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

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

Bit 7 ICIS1	Description
0	Idle cycle not inserted in case of successive external read cycles in different areas.
1	Idle cycle inserted in case of successive external read cycles in different areas. (Initial value)

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

Bit 6 ICIS0	Description
0	Idle cycle not inserted in case of successive external read and external write cycles.
1	Idle cycle inserted in case of successive external read and external write cycles. (Initial value)

Bit 5—Burst ROM Enable (BRSTRM): Selects whether area 0 is used as a burst ROM interface area.

Bit 5 BRSTRM	Description
0	Area 0 is basic bus interface area (Initial value)
1	Area 0 is burst ROM interface area

Bit 4 BRSTS1	Description	
0	Burst cycle comprises 1 state	
1	Burst cycle comprises 2 states	(Initial value)

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

Bit 3 BRSTS0	Description	
0	Max. 4 words in burst access	(Initial value)
1	Max. 8 words in burst access	

Bits 2 to 0—RAM Type Select (RMTS2 to RMTS0): These bits select the memory interface for areas 2 to 5 in advanced mode.

When DRAM space is selected, the relevant area is designated as a DRAM interface area.

Bit 2	Bit 1	Bit 0	Description			
RMTS2	RMTS1	RMTS0	Area 5	Area 4	Area 3	Area 2
0	0	0	Normal space	Normal space	Normal space	Normal space
		1	Normal space	Normal space	Normal space	DRAM space
	1	0	Normal space	Normal space	DRAM space	DRAM space
		1	DRAM space	DRAM space	DRAM space	DRAM space
1	—	—	—	—	—	—

The $\overline{\text{LCAS}}$ pin is used for the $\overline{\text{LCAS}}$ signal on the 2-CAS DRAM interface. If it is wished to use $\overline{\text{BREQO}}$ output when using the $\overline{\text{LCAS}}$ signal, it is possible to switch to the P53 pin by means of the BREQOPS bit in PFCR2. For details, see section 9.6, Port 5 and section 9.16, Port F.

Bit	:	7	6	5	4	3	2	1	0
		BRLE	BREQOE	EAE	—	DDS	—	WDBE	WAITE
Initial value	:	0	0	1	1	1	1	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BCRL is an 8-bit readable/writable register that performs selection of the external bus-released state protocol, selection of the area partition unit, enabling or disabling of the write data buffer function, and enabling or disabling of $\overline{\text{WAIT}}$ pin input.

BCRL is initialized to H'3C by a reset, and in hardware standby mode. It is not initialized in software standby mode.

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

Bit 7 BRLE	Description
0	External bus release disabled. $\overline{\text{BREQ}}$, $\overline{\text{BACK}}$, and $\overline{\text{BREQO}}$ pins can be used as I/O ports (Initial value)
1	External bus release enabled

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

Bit 6 BREQOE	Description
0	$\overline{\text{BREQO}}$ output disabled. $\overline{\text{BREQO}}$ pin can be used as I/O port (Initial value)
1	$\overline{\text{BREQO}}$ output enabled

Bit 5—External Address Enable (EAE): Designates addresses H'010000 to H'03FFFF*2 as either internal or external addresses.

0	On-chip ROM	Addresses H'010000 to H'01FFFF are on-chip ROM or addresses H'020000 to H'03FFFF are reserved area*1 Reserved area*1
1	Addresses H'010000 to H'03FFFF are external addresses in external expanded mode or reserved area*1 in single-chip mode	

Notes: 1. Do not access a reserved area.
2. Addresses H'010000 to H'05FFFF in the H8S/2339.

Bit 4—Reserved: Only 1 should be written to this bit.

Bit 3—DACK Timing Select (DDS): Selects the DMAC single address transfer bus timing for the DRAM interface.

Bit 3 DDS	Description	
0	When DMAC single address transfer is performed in DRAM space, full access is always executed. $\overline{\text{DACK}}$ signal goes low from Tr or T1 cycle	
1	Burst access is possible when DMAC single address transfer is performed in DRAM space. $\overline{\text{DACK}}$ signal goes low from Tc1 or T2 cycle	(Initial value)

Bit 2—Reserved: Only 1 should be written to this bit.

Bit 1—Write Data Buffer Enable (WDBE): Selects whether or not the write buffer function is used for an external write cycle or DMAC single address cycle.

Bit 1 WDBE	Description	
0	Write data buffer function not used	(Initial value)
1	Write data buffer function used	

Bit 0—WAIT Pin Enable (WAITE): Selects enabling or disabling of wait input by the $\overline{\text{WAIT}}$ 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	(Initial value)
1	Wait input by $\overline{\text{WAIT}}$ pin enabled	

Bit	:	7	6	5	4	3	2	1	0
		TPC	BE	RCDM	—	MXC1	MXC0	RLW1	RLW0
Initial value	:	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

MCR is an 8-bit readable/writable register that selects the DRAM strobe control method, number of precharge cycles, access mode, address multiplexing shift size, and the number of wait states inserted during refreshing, when areas 2 to 5 are designated as DRAM interface areas.

MCR is initialized to H'00 by a reset, and in hardware standby mode. It is not initialized in software standby mode.

Bit 7—TP Cycle Control (TPC): Selects whether a 1-state or 2-state precharge cycle (T_P) is to be used when areas 2 to 5 designated as DRAM space are accessed.

Bit 7 TPC	Description	
0	1-state precharge cycle is inserted	(Initial value)
1	2-state precharge cycle is inserted	

Bit 6—Burst Access Enable (BE): Selects enabling or disabling of burst access to areas 2 to 5 designated as DRAM space. DRAM space burst access is performed in fast page mode.

Bit 6 BE	Description	
0	Burst disabled (always full access)	(Initial value)
1	DRAM space access performed in fast page mode	

Bit 5—RAS Down Mode (RCDM): When areas 2 to 5 are designated as DRAM space and access to DRAM is interrupted, RCDM selects whether the $\overline{RAS}$ signal is held low while waiting for the next DRAM access (RAS down mode), or is driven high again (RAS up mode).

Bit 5 RCDM	Description	
0	RAS up mode selected for DRAM interface	(Initial value)
1	RAS down mode selected for DRAM interface	

Bits 3 and 2—Multiplex Shift Count 1 and 0 (MXC1, MXC0): These bits select the size of the shift toward the lower half of the row address in row address/column address multiplexing for the DRAM interface. In burst operation on the DRAM interface, these bits also select the row address bits to be used for comparison.

Bit 3 MXC1	Bit 2 MXC0	Description
0	0	8-bit shift (Initial value) • When 8-bit access space is designated: Row address bits A_{23} to A_8 used for comparison • When 16-bit access space is designated: Row address bits A_{23} to A_9 used for comparison
	1	9-bit shift • When 8-bit access space is designated: Row address bits A_{23} to A_9 used for comparison • When 16-bit access space is designated: Row address bits A_{23} to A_{10} used for comparison
1	0	10-bit shift • When 8-bit access space is designated: Row address bits A_{23} to A_{10} used for comparison • When 16-bit access space is designated: Row address bits A_{23} to A_{11} used for comparison
	1	—

Bits 1 and 0—Refresh Cycle Wait Control 1 and 0 (RLW1, RLW0): These bits select the number of wait states to be inserted in a DRAM interface CAS-before-RAS refresh cycle. This setting is used for all areas designated as DRAM space. Wait input by the $\overline{\text{WAIT}}$ pin is disabled.

Bit 1 RLW1	Bit 0 RLW0	Description
0	0	No wait state inserted (Initial value)
	1	1 wait state inserted
1	0	2 wait states inserted
	1	3 wait states inserted

Bit	:	7	6	5	4	3	2	1	0
		RFSHE	RCW	RMODE	CMF	CMIE	CKS2	CKS1	CKS0
Initial value	:	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

DRAMCR is an 8-bit readable/writable register that selects the DRAM refresh mode and refresh counter clock and controls the refresh timer.

DRAMCR is initialized to H'00 by a reset, and in hardware standby mode. It is not initialized in software standby mode.

Bit 7—Refresh Control (RFSHE): Selects whether or not refresh control is performed. When refresh control is not performed, the refresh timer can be used as an interval timer.

Bit 7 RFSHE	Description	
0	Refresh control is not performed	(Initial value)
1	Refresh control is performed	

Bit 6—RAS-CAS Wait (RCW): Controls wait state insertion in DRAM interface CAS-before-RAS refreshing.

Bit 6 RCW	Description	
0	Wait state insertion in CAS-before-RAS refreshing disabled $\overline{\text{RAS}}$ falls in Tr cycle	(Initial value)
1	One wait state inserted in CAS-before-RAS refreshing $\overline{\text{RAS}}$ falls in Tc1 cycle	

Bit 5—Refresh Mode (RMODE): Selects whether self-refreshing is performed in software standby mode.

Bit 5 RMODE	Description	
0	Self-refreshing is not performed in software standby mode	(Initial value)
1	Self-refreshing is performed in software standby mode	

When refresh control is performed (RFSHE = 1), 1 should be written to the CMF bit when writing to DRAMCR.

Bit 4 CMF	Description	
0	[Clearing condition] When 0 is written to CMF after reading CMF = 1	(Initial value)
1	[Setting condition] When RTCNT = RTCOR	

Bit 3—Compare Match Interrupt Enable (CMIE): Enables or disables interrupt requests (CMI) by the CMF flag when the CMF flag in DRAMCR is set to 1.

When refresh control is performed (RFSHE = 1), the CMIE bit is always cleared to 0.

Bit 3 CMIE	Description	
0	Interrupt request (CMI) by CMF flag disabled	(Initial value)
1	Interrupt request (CMI) by CMF flag enabled	

Bits 2 to 0—Refresh Counter Clock Select (CKS2 to CKS0): These bits select the clock to be input to RTCNT from among seven clocks obtained by dividing the system clock (ϕ). When the input clock is selected with bits CKS2 to CKS0, RTCNT begins counting up.

Bit 2 CKS2	Bit 1 CKS1	Bit 0 CKS0	Description	
0	0	0	Count operation disabled	(Initial value)
		1	Count uses ϕ/2	
	1	0	Count uses ϕ/8	
		1	Count uses ϕ/32	
1	0	0	Count uses ϕ/128	
		1	Count uses ϕ/512	
	1	0	Count uses ϕ/2048	
		1	Count uses ϕ/4096	

Bit	:	7	6	5	4	3	2	1	0
Initial value	:	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

RTCNT is an 8-bit readable/writable up-counter.

RTCNT counts up using the internal clock selected by bits CKS2 to CKS0 in DRAMCR.

When RTCNT matches RTCOR (compare match), the CMF flag in DRAMCR is set to 1 and RTCNT is cleared to H'00. If the RFSHE bit in DRAMCR is set to 1 at this time, a refresh cycle is started. Also, if the CMIE bit in DRAMCR is set to 1, a compare match interrupt (CMI) is generated.

RTCNT is initialized to H'00 by a reset, and in hardware standby mode. It is not initialized in software standby mode.

6.2.9 Refresh Time Control Register (RTCOR)

Bit	:	7	6	5	4	3	2	1	0
Initial value	:	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

RTCOR is an 8-bit readable/writable register that sets the period for compare match operations with RTCNT.

The values of RTCOR and RTCNT are constantly compared, and if they match, the CMF flag in DRAMCR is set to 1 and RTCNT is cleared to H'00.

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

6.3.1 Area Partitioning

In advanced mode, the bus controller partitions the 16-Mbyte address space into eight areas, 0 to 7, in 2-Mbyte units, and performs bus control for external space in area units. Figure 6.2 shows an outline of the memory map.

Chip select signals ($\overline{CS_0}$ to $\overline{CS_7}$) can be output for each area.

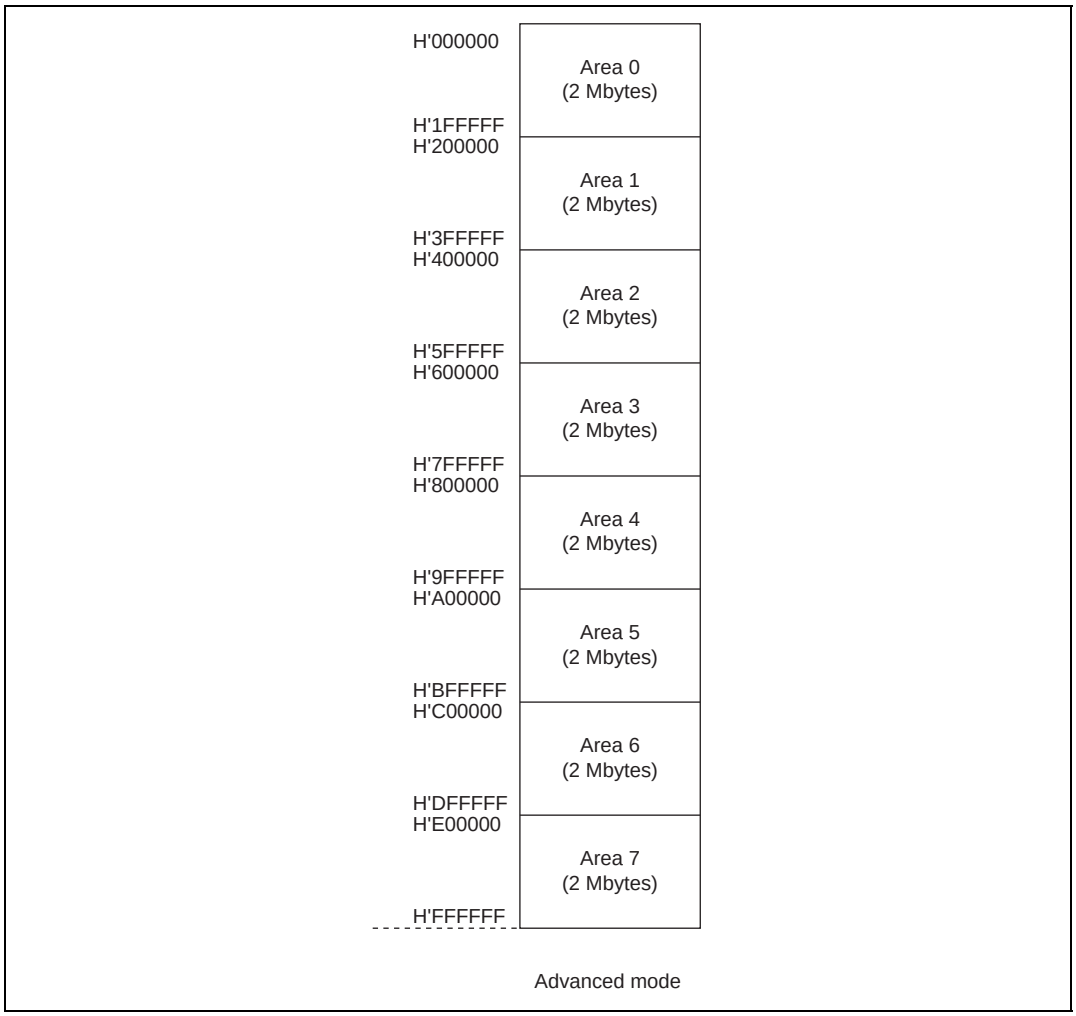


Figure 6.2 Area Partitioning

The external space bus specifications consist of three elements: (1) bus width, (2) number of access states, and (3) number of program wait states.

The bus width and number of access states for on-chip memory and internal I/O registers are fixed, and are not affected by the bus controller.

Bus Width: A bus width of 8 or 16 bits can be selected with ABWCR. An area for which an 8-bit bus is selected functions as an 8-bit access space, and an area for which a 16-bit bus is 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 designated for 16-bit access, 16-bit bus mode is always set. When the burst ROM interface is selected, 16-bit bus mode is always set.

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

With the DRAM interface and burst ROM interface, the number of access states may be determined without regard to ASTCR.

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

Number of Program Wait States: When 3-state access space is designated by ASTCR, the number of program wait states to be inserted automatically is selected with WCRH and WCRL. From 0 to 3 program wait states can be selected.

Table 6.3 shows the bus specifications for each basic bus interface area.

ABWCR	ASTCR	WCRH, WCRL		Bus Specifications (Basic Bus Interface)		
ABWn	**ASTn**	**Wn1**	**Wn0**	**Bus Width**	**Access States**	**Program Wait States**
0	0	—	—	16	2	0
	1	0	0		3	0
			1			1
		1	0			2
			1			3
1	0	—	—	8	2	0
	1	0	0		3	0
			1			1
		1	0			2
			1			3

6.3.3 Memory Interfaces

The chip's interfaces comprise a basic bus interface that allows direct connection of ROM, SRAM, and so on; a DRAM interface that allows direct connection of DRAM; and a burst ROM interface that allows direct connection of burst ROM. The interface can be selected independently for each area.

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

The initial state of each area is basic bus interface, 3-state access space. The initial bus width is selected according to the operating mode. The bus specifications described here cover basic items only, and the sections on each memory interface (section 6.4, Basic Bus Interface, section 6.5, DRAM Interface, and section 6.7, Burst ROM Interface) should be referred to for further details.

Area 0: Area 0 includes on-chip ROM, and in expanded mode with on-chip ROM disabled, all of area 0 is external space. In expanded mode with on-chip ROM enabled, the space excluding on-chip ROM is external space.

When area 0 external space is accessed, the $\overline{CS_0}$ signal can be output.

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

Areas 1 and 6: In external expanded mode, all of area 1 and area 6 is external space.

When area 1 and 6 external space is accessed, the $\overline{CS_1}$ and $\overline{CS_6}$ pin signals can be output, respectively.

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

Areas 2 to 5: In external expanded mode, areas 2 to 5 are all external space.

When area 2 to 5 external space is accessed, signals $\overline{CS_2}$ to $\overline{CS_5}$ can be output.

Basic bus interface or DRAM interface can be selected for areas 2 to 5. With the DRAM interface, signals $\overline{CS_2}$ to $\overline{CS_5}$ are used as $\overline{RAS}$ signals.

Area 7: Area 7 includes the on-chip RAM and internal I/O registers. In external expanded mode, the space excluding the on-chip RAM and internal I/O registers is external space. The on-chip RAM is enabled when the RAME bit in the system control register (SYSCR) is set to 1; when the RAME bit is cleared to 0, the on-chip RAM is disabled and the corresponding space becomes external space.

When area 7 external space is accessed, the $\overline{CS_7}$ signal can be output.

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

The chip can output chip select signals ($\overline{CS_0}$ to $\overline{CS_7}$) to areas 0 to 7, the signal being driven low when the corresponding external space area is accessed.

Figure 6.3 shows an example of $\overline{CS_n}$ (n = 0 to 7) output timing.

Enabling or disabling of $\overline{CS_n}$ signal output is performed by setting the data direction register (DDR) bit for the port corresponding to the particular $\overline{CS_n}$ pin, the CS/67 enable bit (CS/67E), and the CS25 enable bit (CS25E).

In expanded mode with on-chip ROM disabled, the CS0 pin is placed in the output state after a reset. Pins $\overline{CS_1}$ to $\overline{CS_7}$ are placed in the input state after a reset, so the corresponding DDR bits as well as bits CS/67E and CS25E should be set to 1 when outputting signals $\overline{CS_1}$ to $\overline{CS_7}$.

In expanded mode with on-chip ROM enabled, pins $\overline{CS_0}$ to $\overline{CS_7}$ are all placed in the input state after a reset, so the corresponding DDR bits as well as bits CS/67E and CS25E should be set to 1 when outputting signals $\overline{CS_1}$ to $\overline{CS_7}$.

For details, see section 9, I/O Ports.

When areas 2 to 5 are designated as DRAM space, outputs $\overline{CS_2}$ to $\overline{CS_5}$ are used as $\overline{RAS}$ signals.

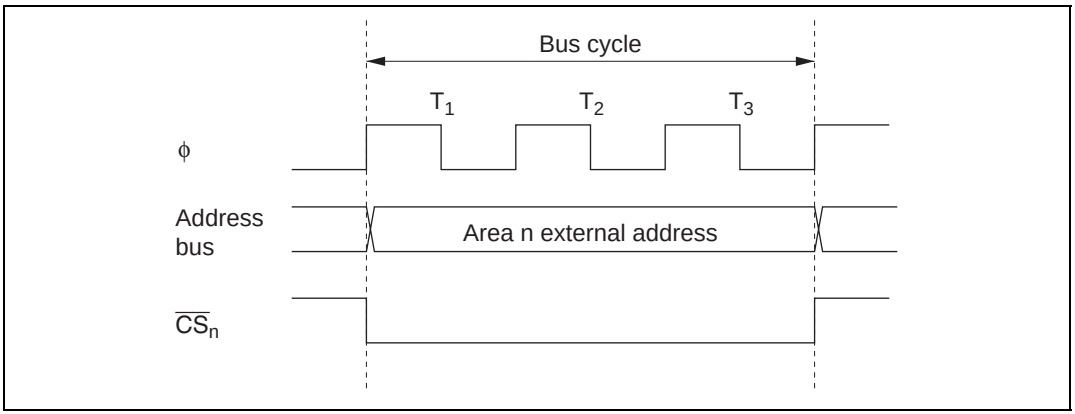


Figure 6.3 $\overline{CS_n}$ Signal Output Timing (n = 0 to 7)

6.4.1 Overview

The basic bus interface enables direct connection of ROM, SRAM, and so on.

The bus specifications can be selected with ABWCR, ASTCR, WCRH, and WCRL. (See table 6.3.)

6.4.2 Data Size and Data Alignment

Data sizes for the CPU and other internal bus masters are byte, word, and longword. The bus controller has a data alignment function, and when accessing external space, controls whether the upper data bus (D_{15} to D_8) or lower data bus (D_7 to D_0) is used according to the bus specifications for the area being accessed (8-bit access space or 16-bit access space) and the data size.

8-Bit Access Space: Figure 6.4 illustrates data alignment control for the 8-bit access space. With the 8-bit access space, the upper data bus (D_{15} to D_8) is always used for accesses. The amount of data that can be accessed at one time is one byte: a word transfer instruction is performed as two byte accesses, and a longword transfer instruction, as four byte accesses.

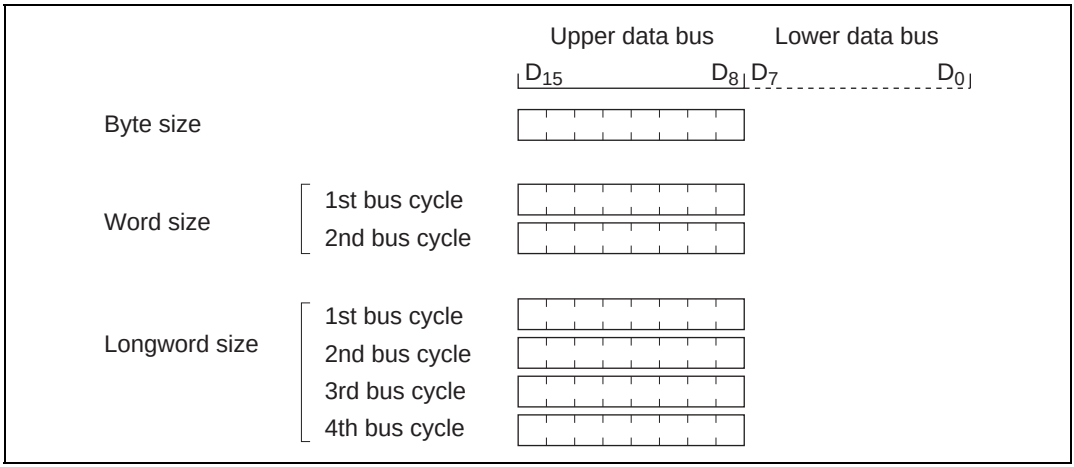


Figure 6.4 Access Sizes and Data Alignment Control (8-Bit Access Space)

for accesses. The amount of data that can be accessed at one time is one byte or one word, and a longword transfer instruction is executed as two word transfer instructions.

In byte access, whether the upper or lower data bus is used is determined by whether the address is even or odd. The upper data bus is used for an even address, and the lower data bus for an odd address.

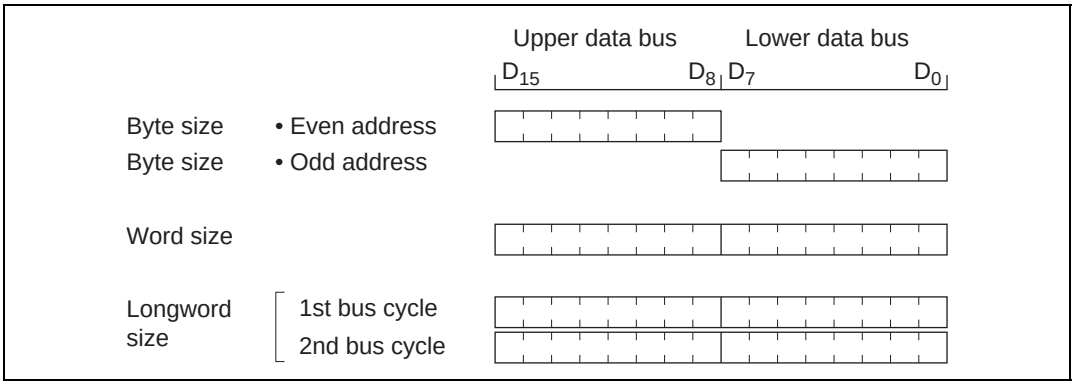


Figure 6.5 Access Sizes and Data Alignment Control (16-Bit Access Space)

Table 6.4 shows the data buses used and valid strobes for the access spaces.

In a read, the $\overline{RD}$ signal is valid without discrimination between the upper and lower halves of the data bus.

In a write, the $\overline{HWR}$ signal is valid for the upper half of the data bus, and the $\overline{LWR}$ signal for the lower half.

Table 6.4 Data Buses Used and Valid Strobes

Area	Access Size	Read/ Write	Address	Valid Strobe	Upper Data Bus (D_{15} to D_8)	Lower Data Bus (D_7 to D_0)
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

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

8-Bit 2-State Access Space: Figure 6.6 shows the bus timing for an 8-bit 2-state access space. When an 8-bit access space is accessed, the upper half (D_{15} to D_8) of the data bus is used.

The $\overline{\text{LWR}}$ pin is fixed high. Wait states cannot be inserted.

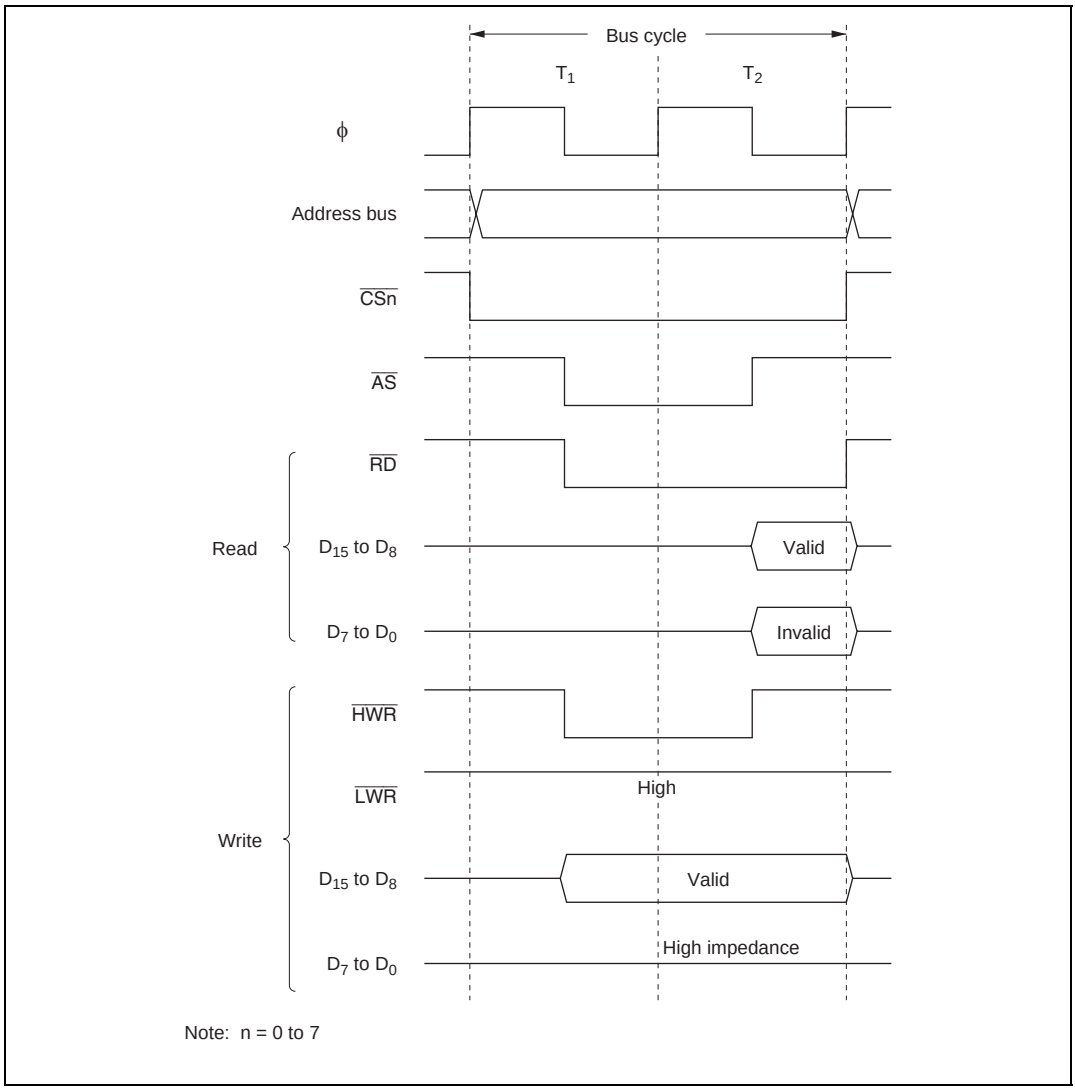


Figure 6.6 Bus Timing for 8-Bit 2-State Access Space

The $\overline{\text{LWR}}$ pin is fixed high. Wait states can be inserted.

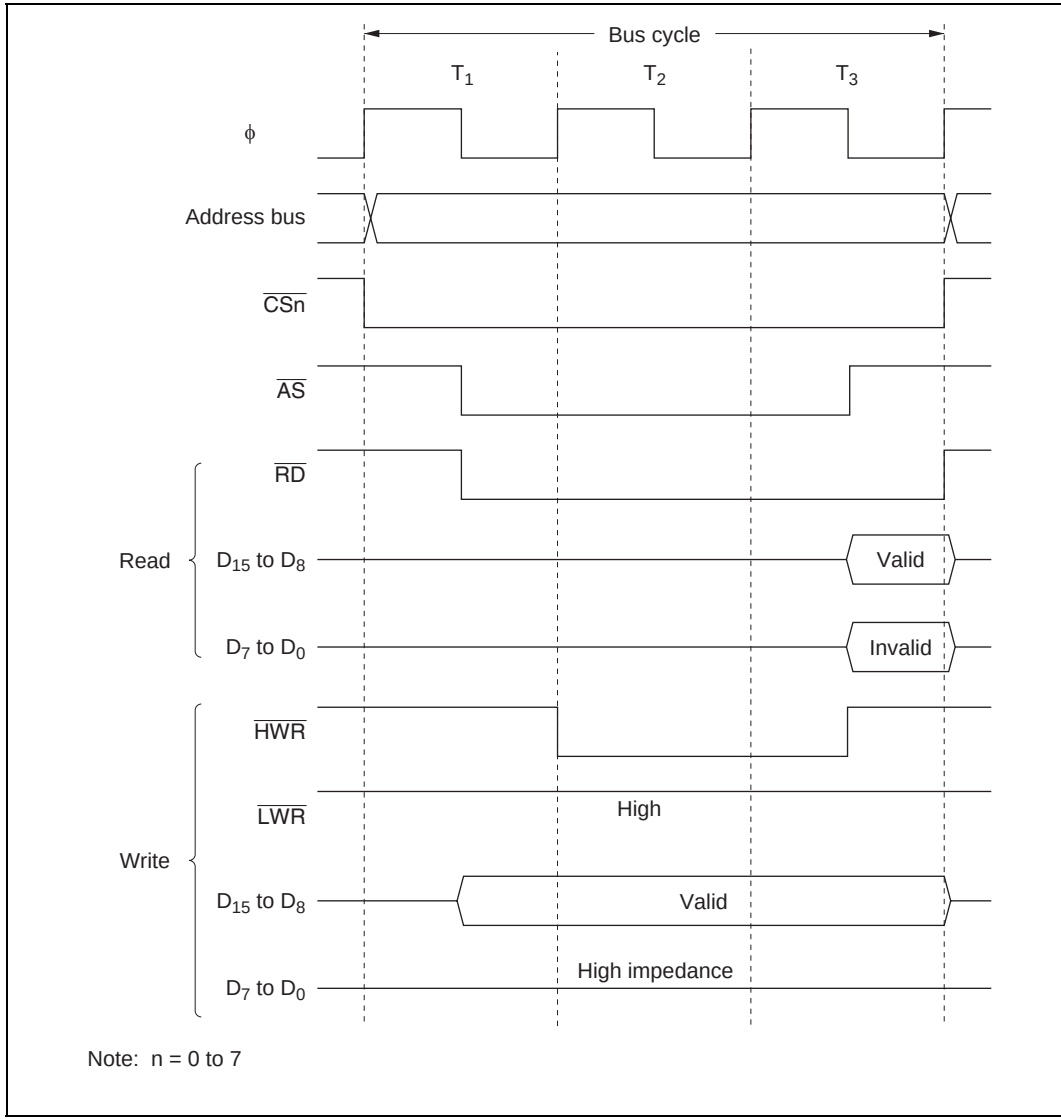


Figure 6.7 Bus Timing for 8-Bit 3-State Access Space

the even address, and the lower half (D_7 to D_0) for the odd address.

Wait states cannot be inserted.

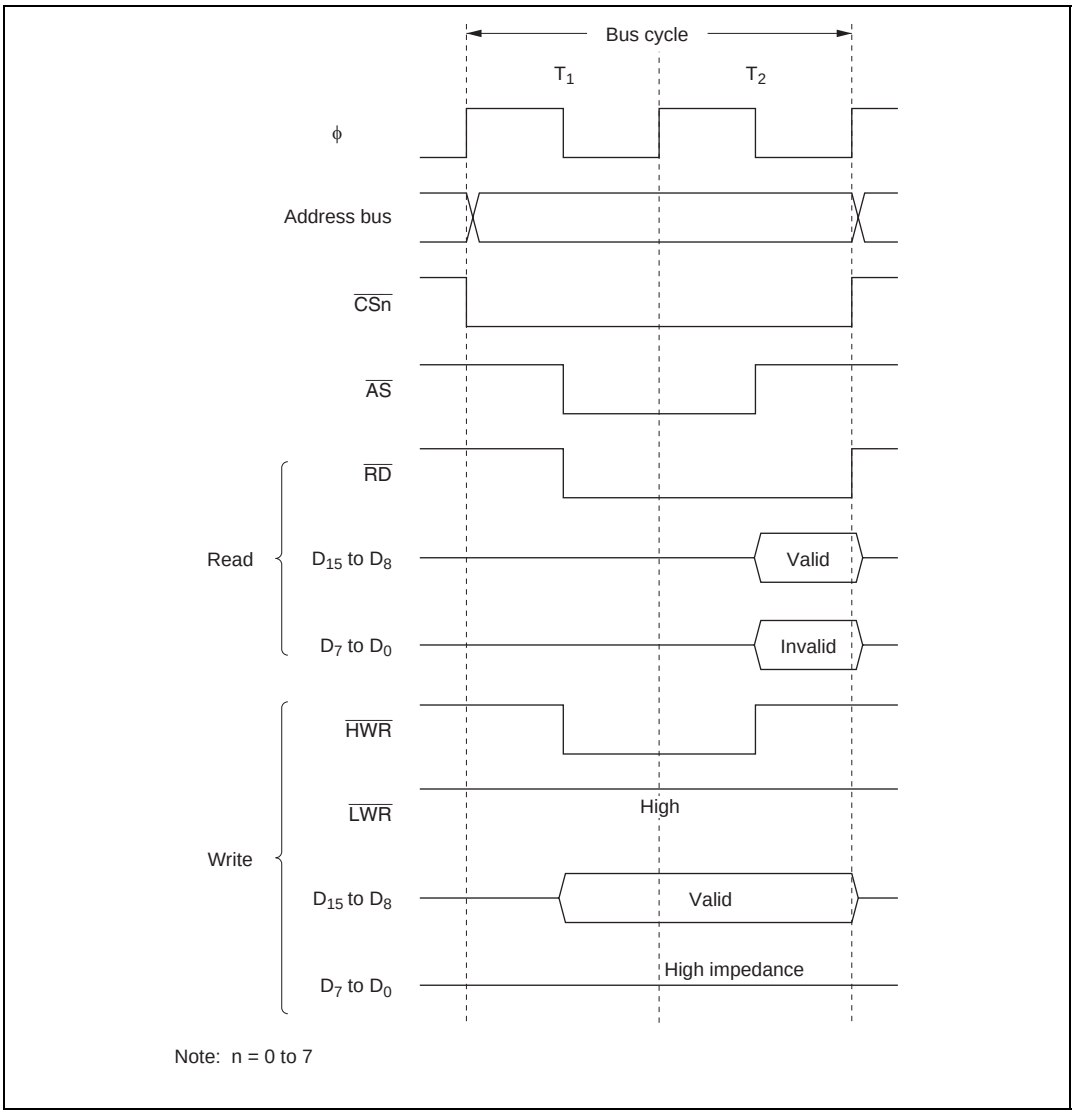


Figure 6.8 Bus Timing for 16-Bit 2-State Access Space (1) (Even Address Byte Access)

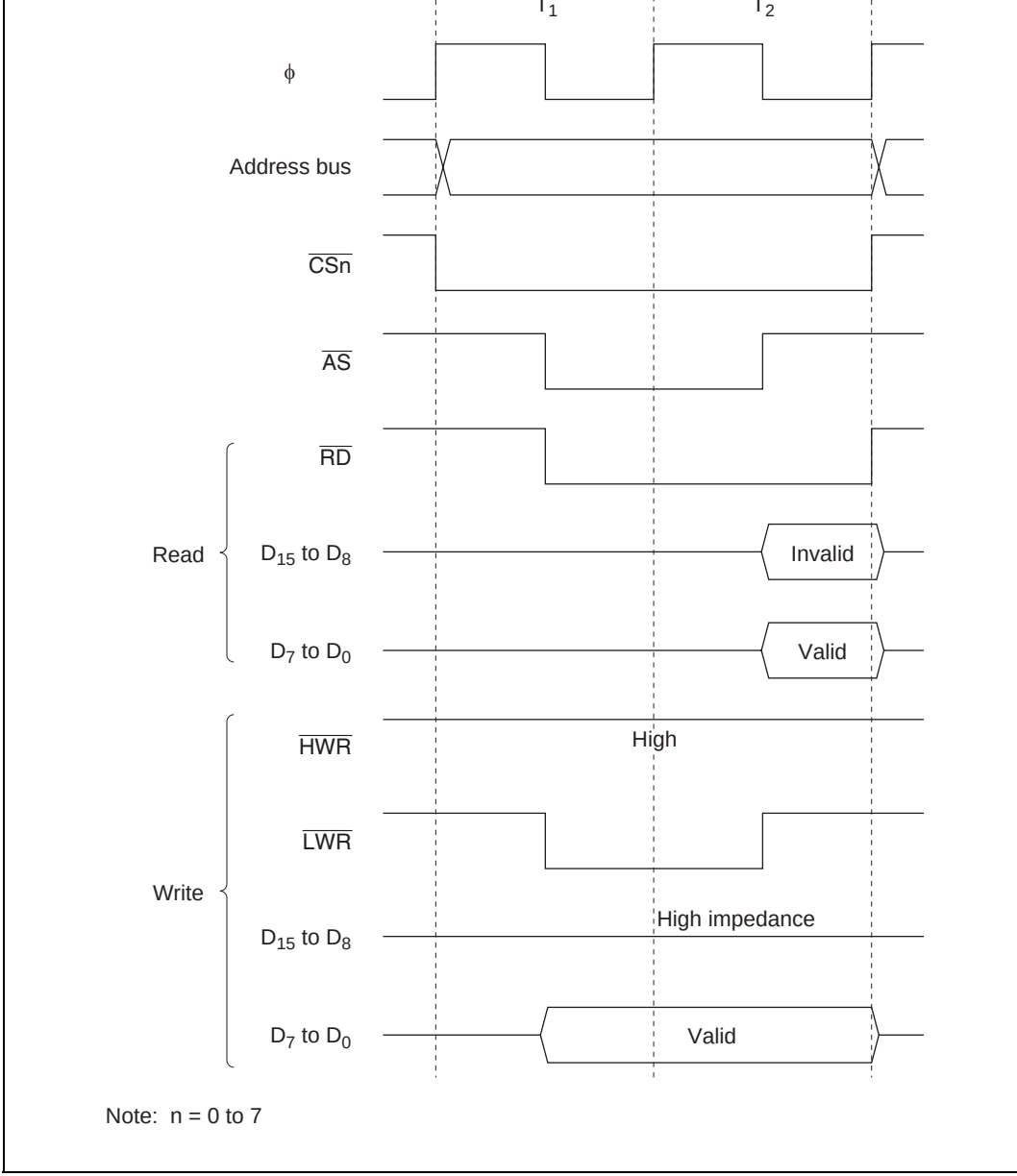
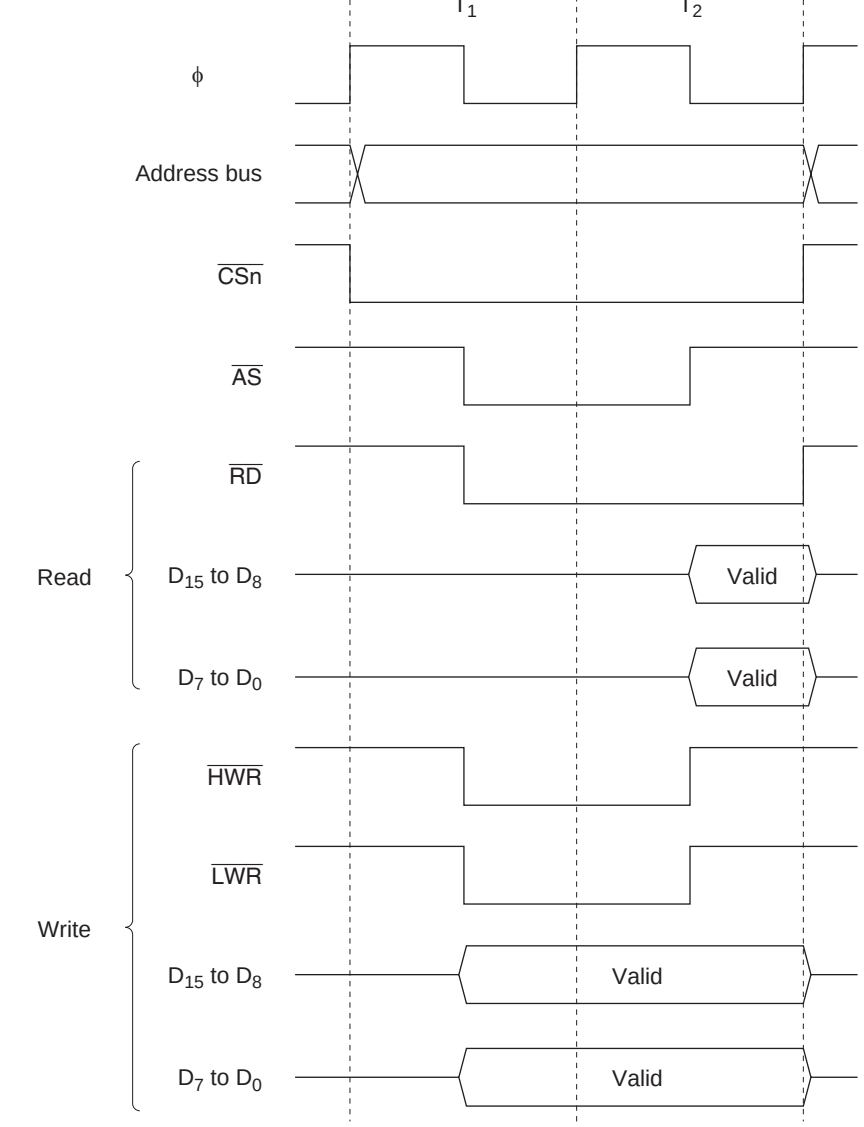


Figure 6.9 Bus Timing for 16-Bit 2-State Access Space (2) (Odd Address Byte Access)



Note: n = 0 to 7

Figure 6.10 Bus Timing for 16-Bit 2-State Access Space (3) (Word Access)

for the even address, and the lower half (D_7 to D_0) for the odd address.

Wait states can be inserted.

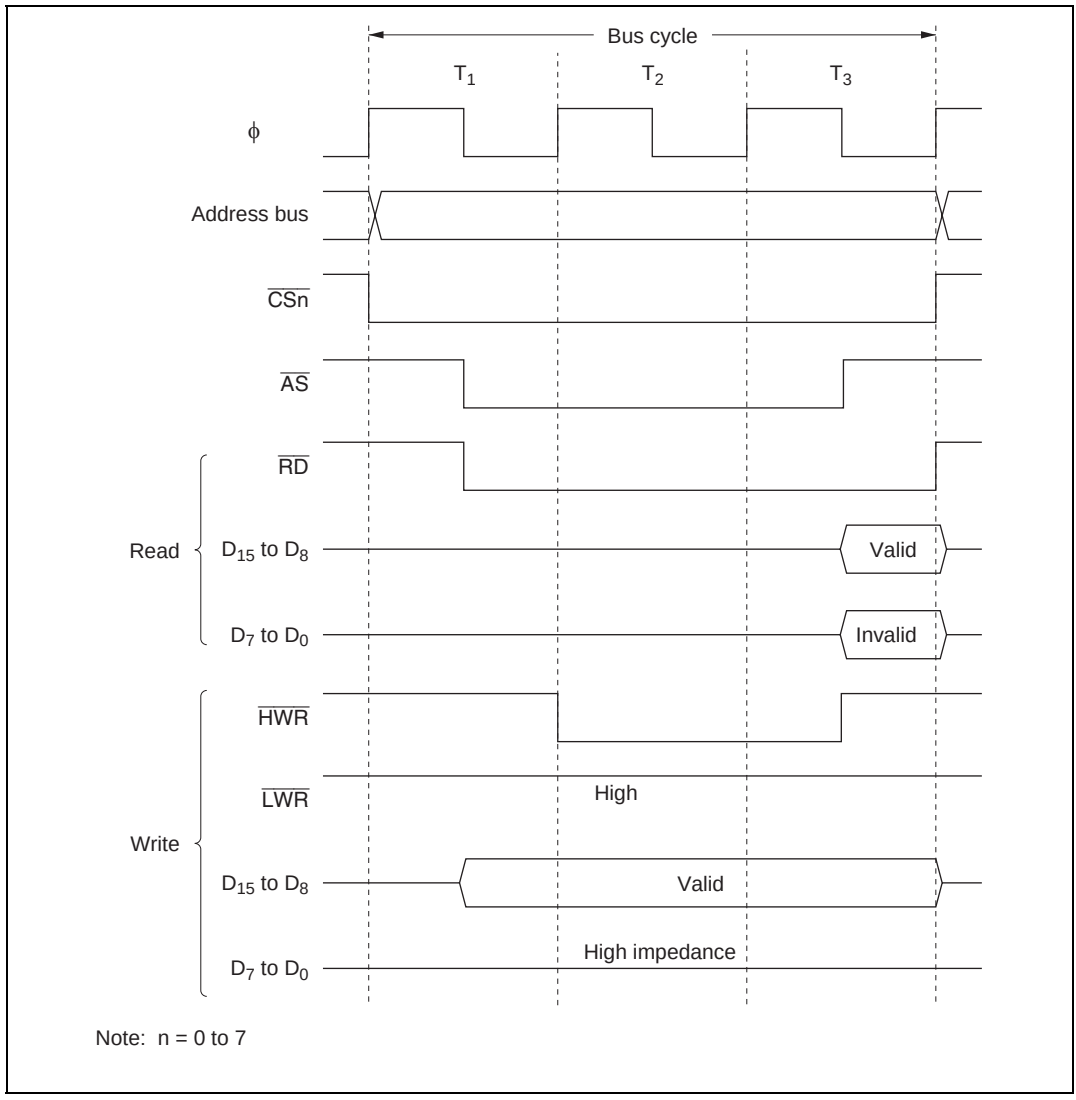


Figure 6.11 Bus Timing for 16-Bit 3-State Access Space (1) (Even Address Byte Access)

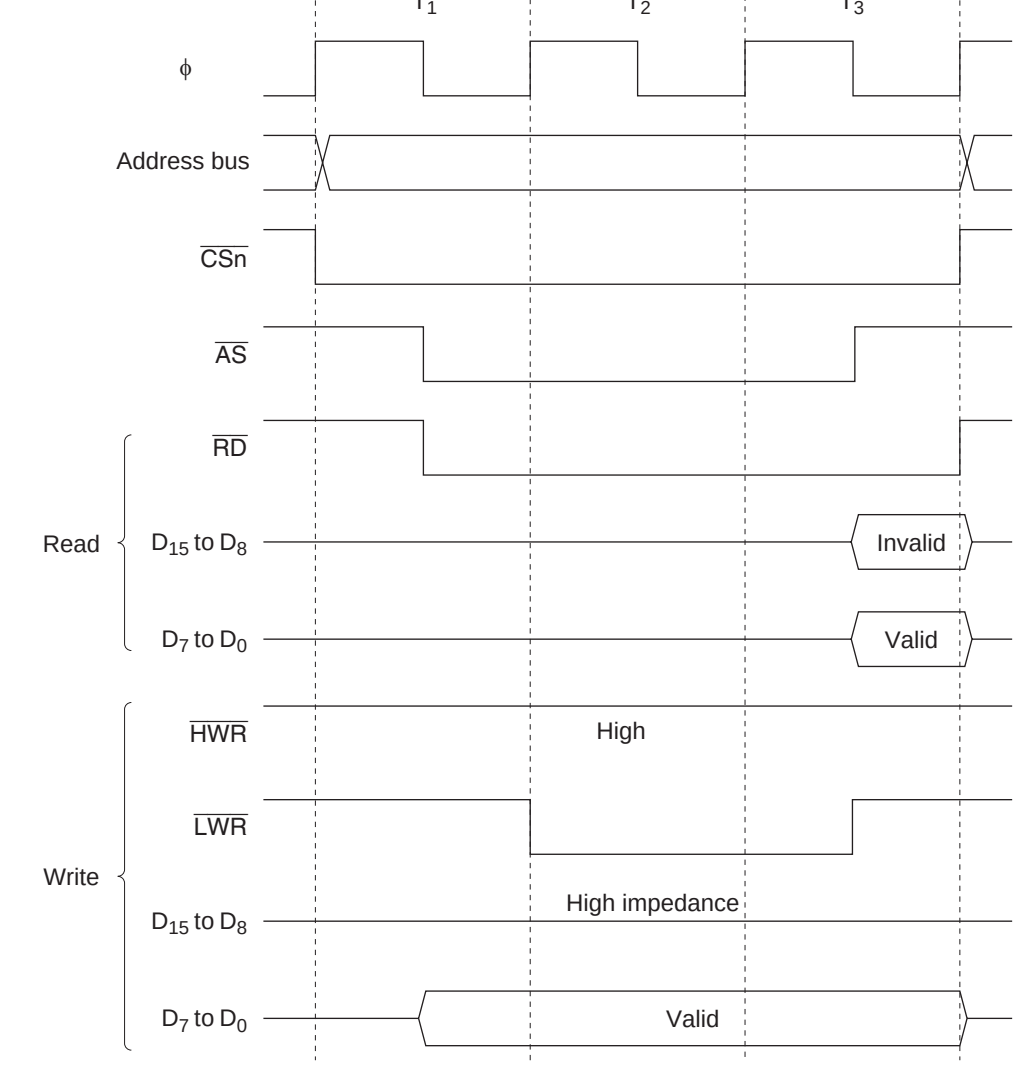


Figure 6.12 Bus Timing for 16-Bit 3-State Access Space (2) (Odd Address Byte Access)

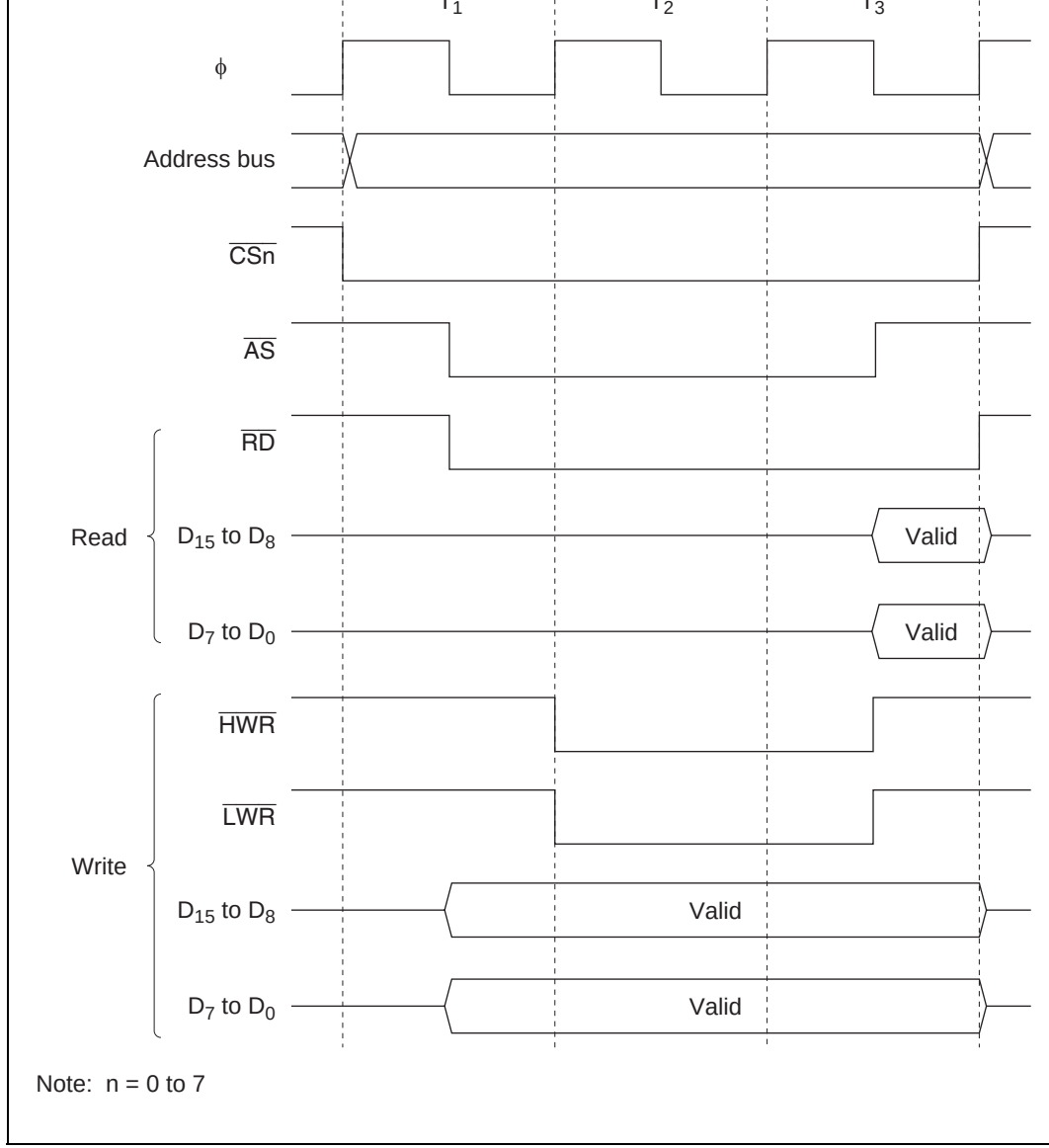


Figure 6.13 Bus Timing for 16-Bit 3-State Access Space (3) (Word Access)

When accessing external space, the chip can extend the bus cycle by inserting one or more wait states (T_w). There are two ways of inserting wait states: program wait insertion and pin wait insertion using the $\overline{\text{WAIT}}$ pin.

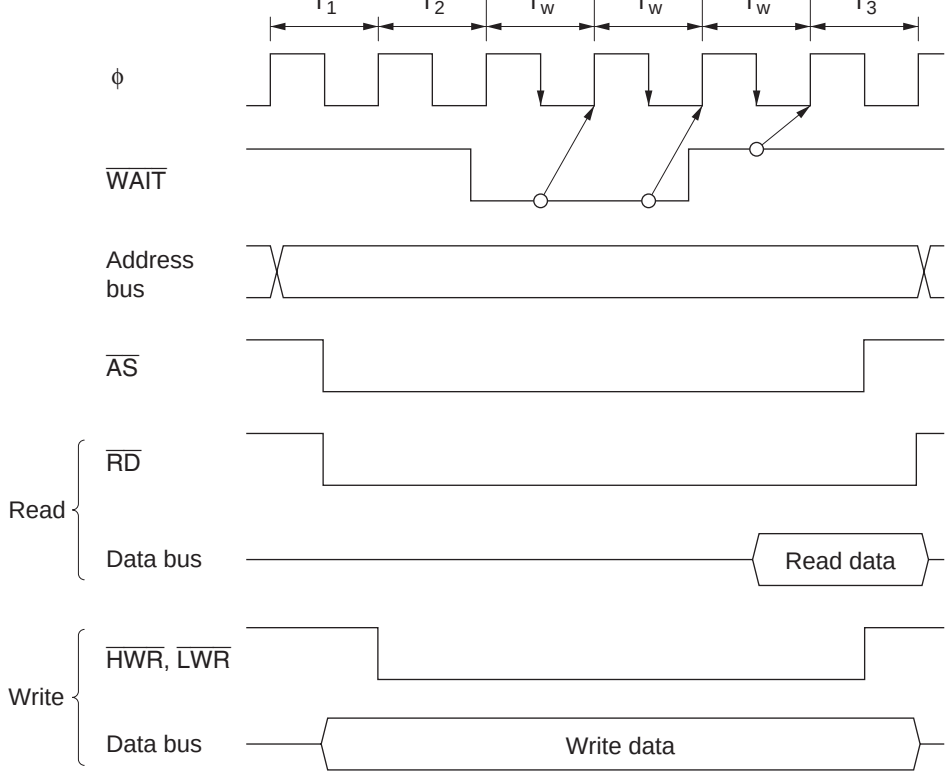
Program Wait Insertion: From 0 to 3 wait states can be inserted automatically between the T_2 state and T_3 state on an individual area basis in 3-state access space, according to the settings in WCRH and WCRL.

Pin Wait Insertion: Setting the WAITE bit in BCRL to 1 enables wait input by means of the $\overline{\text{WAIT}}$ pin. When external space is accessed in this state, a program wait is first inserted in accordance with the settings in WCRH and WCRL. Then, if the $\overline{\text{WAIT}}$ pin is low at the falling edge of ϕ in the last T_2 or T_w state, another T_w state is inserted. If the $\overline{\text{WAIT}}$ pin is held low, T_w states are inserted it goes high.

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

The WAITE bit setting applies to all areas. The WAITPS bit can be used to change the $\overline{\text{WAIT}}$ input pin from $P8_6$ to $P5_3$. To make this change, select the input pin with the WAITPS bit, then set the WAITE bit.

Figure 6.14 shows an example of wait state insertion timing.



Note: Downward arrows indicates the timing of $\overline{\text{WAIT}}$ pin sampling.

Figure 6.14 Example of Wait State Insertion Timing

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

6.5.1 Overview

When the chip is in advanced mode, external space areas 2 to 5 can be designated as DRAM space, and DRAM interfacing performed. With the DRAM interface, DRAM can be directly connected to the chip. A DRAM space of 2, 4, or 8 Mbytes can be set by means of bits RMTS2 to RMTS0 in BCRH. Burst operation is also possible, using fast page mode.

6.5.2 Setting DRAM Space

Areas 2 to 5 are designated as DRAM space by setting bits RMTS2 to RMTS0 in BCRH. The relation between the settings of bits RMTS2 to RMTS0 and DRAM space is shown in table 6.4. Possible DRAM space settings are: one area (area 2), two areas (areas 2 and 3), and four areas (areas 2 to 5).

Table 6.4 DRAM Space Settings by Bits RMTS2 to RMTS0

RMTS2	RMTS1	RMTS0	Area 5	Area 4	Area 3	Area 2
0	0	1	Normal space	Normal space	Normal space	DRAM space
	1	0	Normal space	Normal space	DRAM space	DRAM space
		1	DRAM space	DRAM space	DRAM space	DRAM space

With DRAM space, the row address and column address are multiplexed. In address multiplexing, the size of the shift of the row address is selected with bits MXC1 and MXC0 in MCR. Table 6.5 shows the relation between the settings of MXC1 and MXC0 and the shift size.

Table 6.5 Address Multiplexing Settings by Bits MXC1 and MXC0

	MCR			Address Pins													
	MXC1	MXC0	Shift Size	A_{23} to A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
Row address	0	0	8 bits	A_{23} to A_{13}	A_{20}	A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8
		1	9 bits	A_{23} to A_{13}	A_{12}	A_{20}	A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9
	1	0	10 bits	A_{23} to A_{13}	A_{12}	A_{11}	A_{20}	A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}
		1	Setting pro-hibited	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Column address	—	—	—	A_{23} to A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0

If the bit in ABWCR corresponding to an area designated as DRAM space is set to 1, that area is designated as 8-bit DRAM space; if the bit is cleared to 0, the area is designated as 16-bit DRAM space. In 16-bit DRAM space, ×16-bit configuration DRAM can be connected directly.

In 8-bit DRAM space the upper half of the data bus, D_{15} to D_8, is enabled, while in 16-bit DRAM space both the upper and lower halves of the data bus, D_{15} to D_0, are enabled.

Access sizes and data alignment are the same as for the basic bus interface. For details, see section 6.4.2, Data Size and Data Alignment.

6.5.5 Pins Used for DRAM Interface

Table 6.6 shows the pins used for DRAM interfacing and their functions.

Table 6.6 DRAM Interface Pins

Pin	With DRAM Setting	Name	I/O	Function
$\overline{\text{HWR}}$	$\overline{\text{WE}}$	Write enable	Output	Write enable for DRAM space access when 2-CAS access is set
$\overline{\text{LCAS}}$	$\overline{\text{LCAS}}$	Lower column address strobe	Output	Lower column address strobe signal for 16-bit DRAM space access
$\overline{\text{CS}_2}$	$\overline{\text{RAS2}}$	Row address strobe 2	Output	Row address strobe when area 2 is designated as DRAM space
$\overline{\text{CS}_3}$	$\overline{\text{RAS3}}$	Row address strobe 3	Output	Row address strobe when area 3 is designated as DRAM space
$\overline{\text{CS}_4}$	$\overline{\text{RAS4}}$	Row address strobe 4	Output	Row address strobe when area 4 is designated as DRAM space
$\overline{\text{CS}_5}$	$\overline{\text{RAS5}}$	Row address strobe 5	Output	Row address strobe when area 5 is designated as DRAM space
$\overline{\text{CAS}}$	$\overline{\text{UCAS}}$	Upper column address strobe	Output	Upper column address strobe for DRAM space access
$\overline{\text{WAIT}}$	$\overline{\text{WAIT}}$	Wait	Input	Wait request signal
A_{12} to A_0	A_{12} to A_0	Address pins	Output	Row address/column address multiplexed output
D_{15} to D_0	D_{15} to D_0	Data pins	I/O	Data input/output pins

Figure 6.15 shows the basic access timing for DRAM space. The basic DRAM access timing is four states. Unlike the basic bus interface, the corresponding bits in ASTCR control only enabling or disabling of wait insertion, and do not affect the number of access states. When the corresponding bit in ASTCR is cleared to 0, wait states cannot be inserted in the DRAM access cycle.

The four states of the basic timing consist of one T_p (precharge cycle) state, one T_r (row address output cycle) state, and the T_{c1} and T_{c2} (column address output cycle) states.

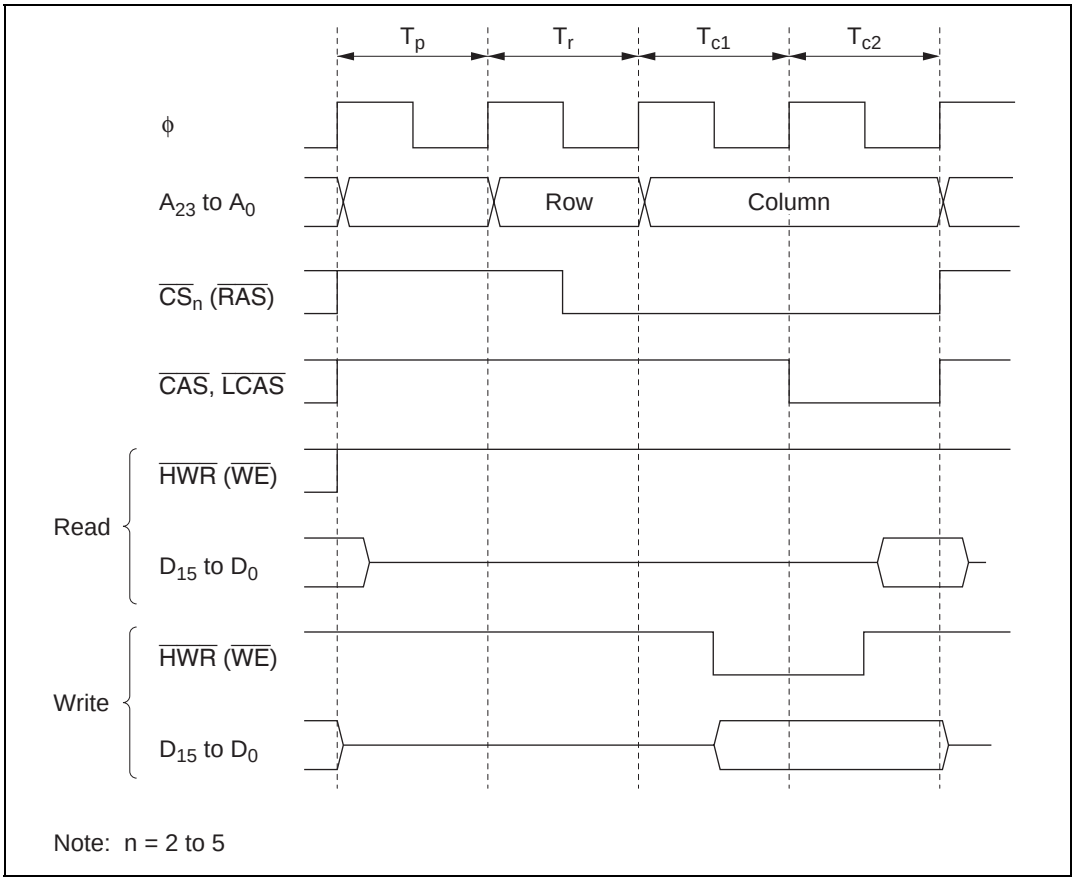


Figure 6.15 Basic Access Timing

When DRAM is accessed, RAS precharging time must be secured. With the chip, one T_p state is always inserted when DRAM space is accessed. This can be changed to two T_p states by setting the TPC bit in MCR to 1. Set the appropriate number of T_p cycles according to the DRAM connected and the operating frequency of the chip. Figure 6.16 shows the timing when two T_p states are inserted.

When the TCP bit is set to 1, two T_p states are also used for refresh cycles.

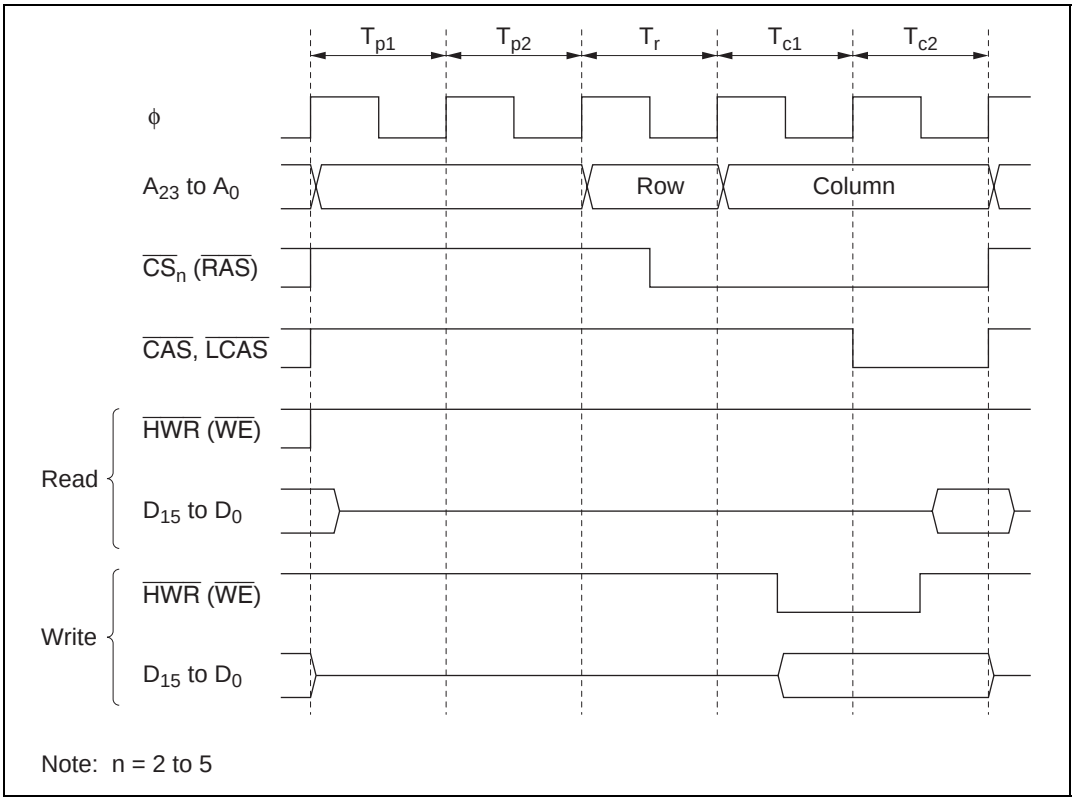


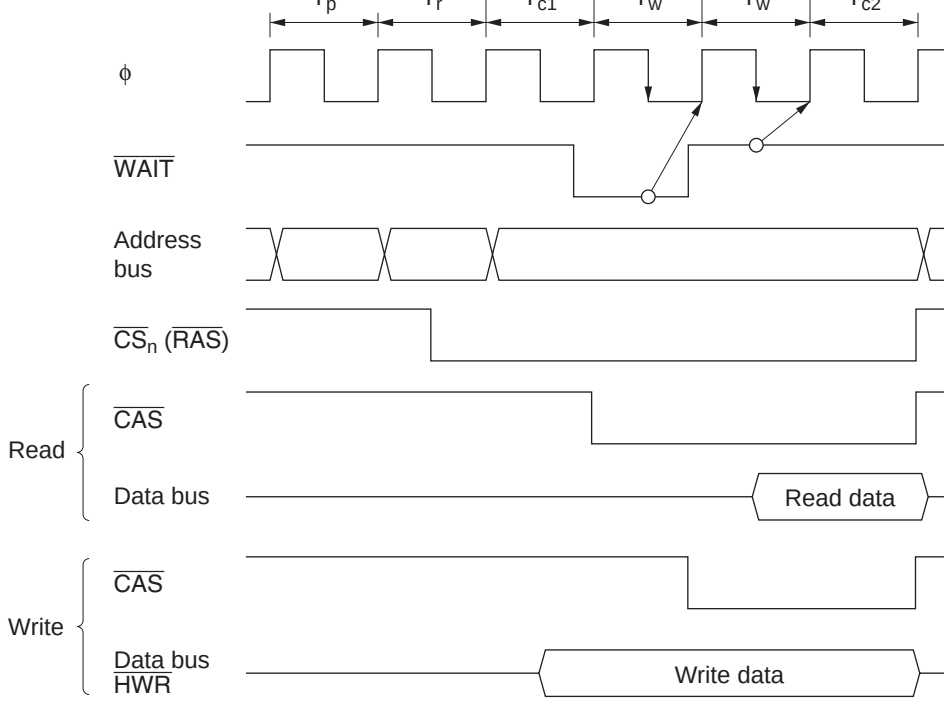
Figure 6.16 Timing with Two-State Precharge Cycle

There are two ways of inserting wait states in a DRAM access cycle: program wait insertion and pin wait insertion using the $\overline{\text{WAIT}}$ pin.

Program Wait Insertion: When the bit in ASTCR corresponding to an area designated as DRAM space is set to 1, from 0 to 3 wait states can be inserted automatically between the T_{c1} state and T_{c2} state, according to the settings of WCRH and WCRL.

Pin Wait Insertion: When the WAITE bit in BCRH is set to 1, wait input by means of the $\overline{\text{WAIT}}$ pin is enabled. When DRAM space is accessed in this state, a program wait is first inserted. If the $\overline{\text{WAIT}}$ pin is low at the falling edge of ϕ in the last T_{c1} or T_w state, another T_w state is inserted. If the $\overline{\text{WAIT}}$ pin is held low, T_w states are inserted until it goes high.

Figure 6.17 shows an example of wait state insertion timing.



Note: Downward arrows indicates the timing of $\overline{WAIT}$ pin sampling.
n = 2 to 5

Figure 6.17 Example of Wait State Insertion Timing

When DRAM with a ×16-bit configuration is connected, the 2-CAS access method is used for the control signals needed for byte access.

Figure 6.18 shows the control timing for 2-CAS access, and figure 6.19 shows an example of 2-CAS DRAM connection.

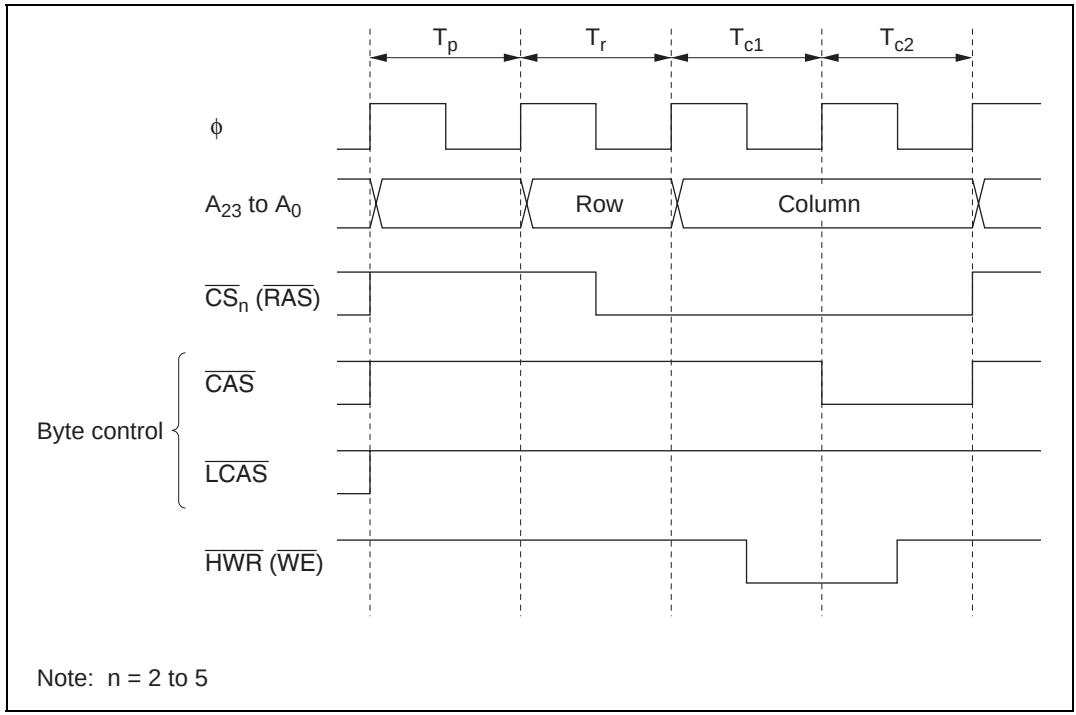


Figure 6.18 2-CAS Control Timing (Upper Byte Write Access)

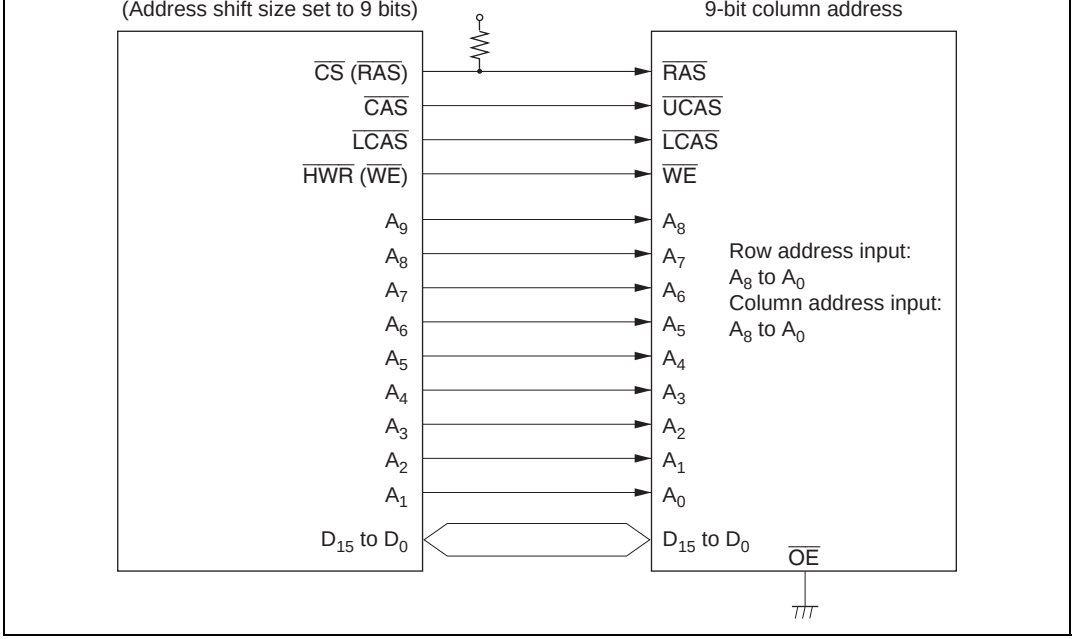


Figure 6.19 Example of 2-CAS DRAM Connection

With DRAM, in addition to full access (normal access) in which data is accessed by outputting a row address for each access, a fast page mode is also provided which can be used when making a number of consecutive accesses to the same row address. This mode enables fast (burst) access of data by simply changing the column address after the row address has been output. Burst access can be selected by setting the BE bit in MCR to 1.

Burst Access (Fast Page Mode) Operation Timing: Figure 6.20 shows the operation timing for burst access. When there are consecutive access cycles for DRAM space, the $\overline{CAS}$ signal and column address output cycles (two states) continue as long as the row address is the same for consecutive access cycles. The row address used for the comparison is set with bits MXC1 and MXC0 in MCR.

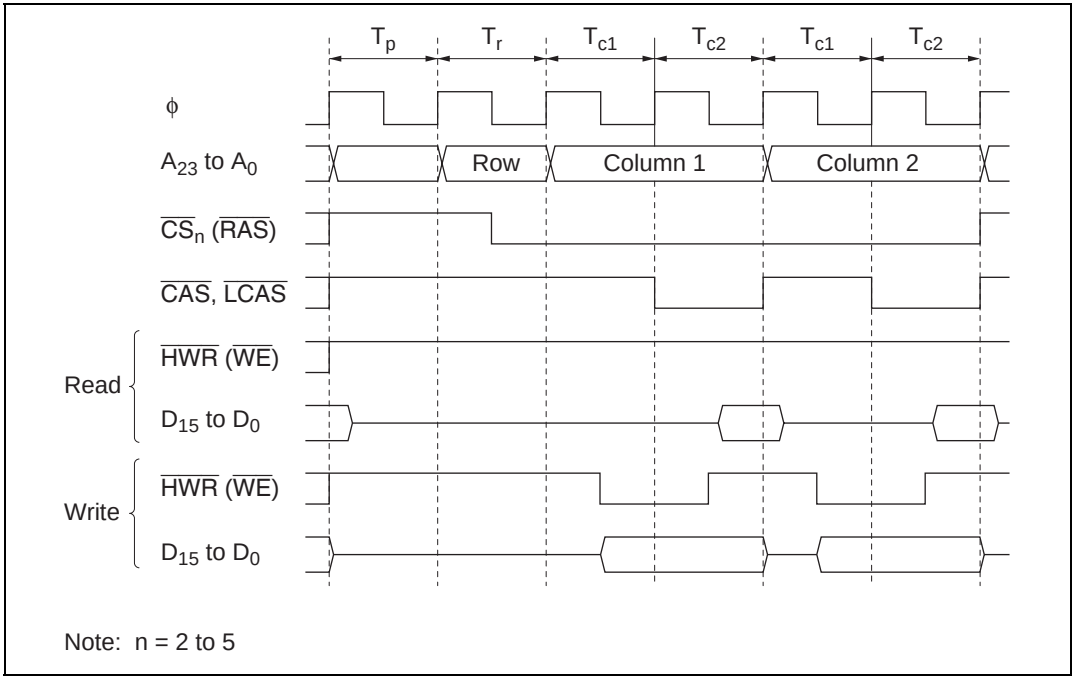


Figure 6.20 Operation Timing in Fast Page Mode

The bus cycle can also be extended in burst access by inserting wait states. The wait state insertion method and timing are the same as for full access. For details, see section 6.4.5, Wait Control.

case, if the $\overline{RAS}$ signal is held low during the access to the other space, burst operation can be resumed when the same row address in DRAM space is accessed again.

- RAS down mode

  To select RAS down mode, set the RCDM bit in MCR to 1. If access to DRAM space is interrupted and another space is accessed, the $\overline{RAS}$ signal is held low during the access to the other space, and burst access is performed when the row address of the next DRAM space access is the same as the row address of the previous DRAM space access. Figure 6.21 shows an example of the timing in RAS down mode.

  Note, however, that the $\overline{RAS}$ signal will go high if a refresh operation occurs during RAS down mode.

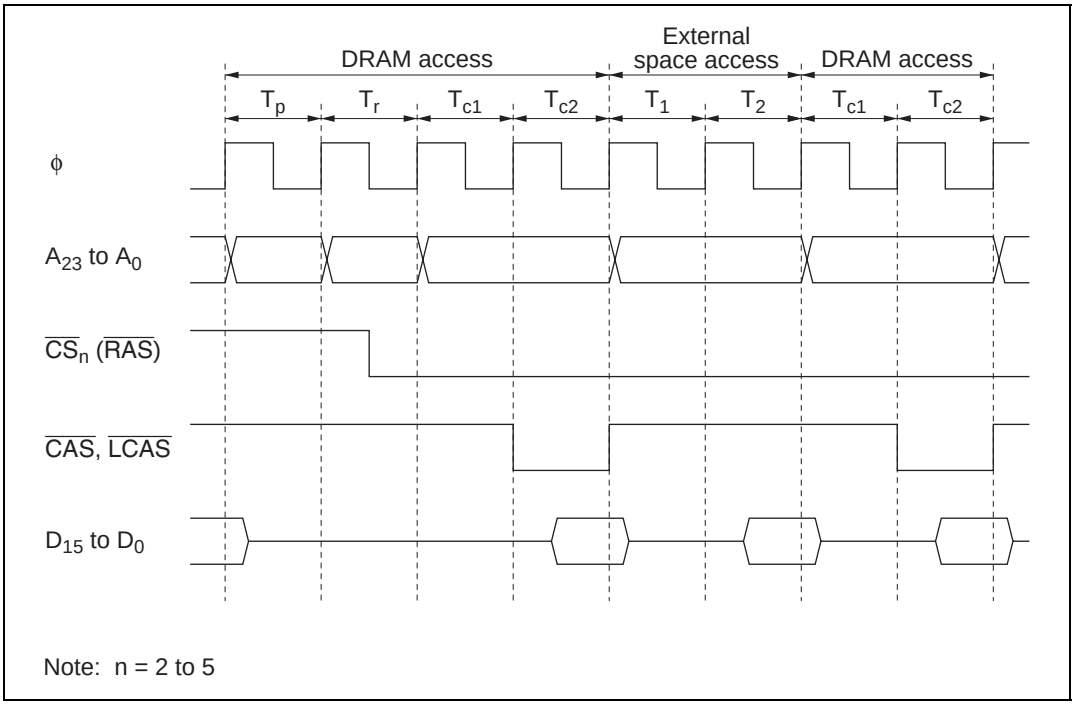


Figure 6.21 Example of Operation Timing in RAS Down Mode

is interrupted and another space is accessed, the $\overline{RAS}$ signal goes high again. Burst operation is only performed if DRAM space is continuous. Figure 6.22 shows an example of the timing in RAS up mode. Note that in burst ROM space access, the $\overline{RAS}$ signal does not return to the high level.

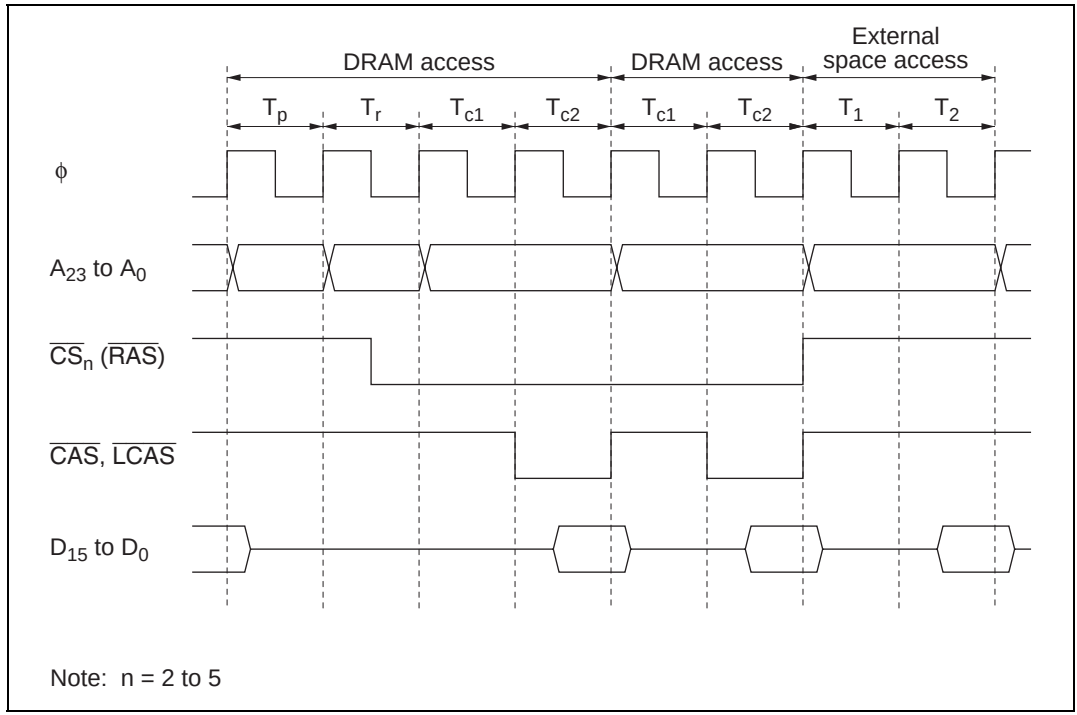


Figure 6.22 Example of Operation Timing in RAS Up Mode

The chip is provided with a DRAM refresh control function. Either of two refreshing methods can be selected: CAS-before-RAS (CBR) refreshing, or self-refreshing.

CAS-before-RAS (CBR) Refreshing: To select CBR refreshing, set the RFSHE bit in DRAMCR to 1, and clear the RMODE bit to 0.

With CBR refreshing, RTCNT counts up using the input clock selected by bits CKS2 to CKS0 in DRAMCR, and when the count matches the value set in RTCOR (compare match), refresh control is performed. At the same time, RTCNT is reset and starts counting again from H'00. Refreshing is thus repeated at fixed intervals determined by RTCOR and bits CKS2 to CKS0. Set a value in bits CKS2 to CKS0 in RTCOR that will meet the refreshing interval specification for the DRAM used.

When bits CKS2 to CKS0 are set, RTCNT starts counting up. RTCNT and RTCOR settings should therefore be completed before setting bits CKS2 to CKS0.

Do not clear the CMF flag when refresh control is performed (RFSHE = 1).

RTCNT operation is shown in figure 6.23, compare match timing in figure 6.24, and CBR refresh timing in figure 6.25.

Another normal space access can be performed during the CBR refresh interval.

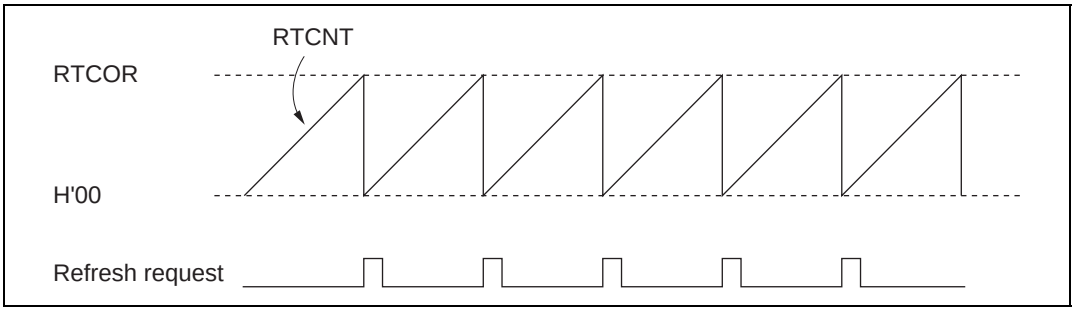


Figure 6.23 RTCNT Operation

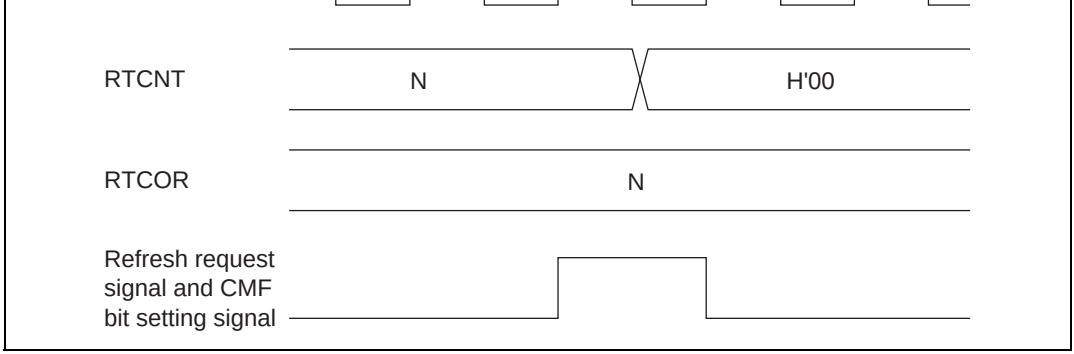


Figure 6.24 Compare Match Timing

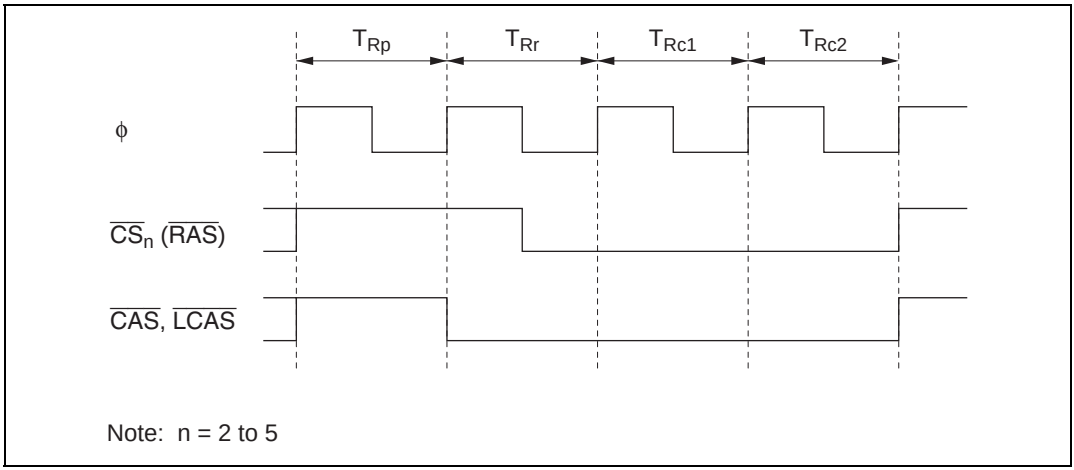
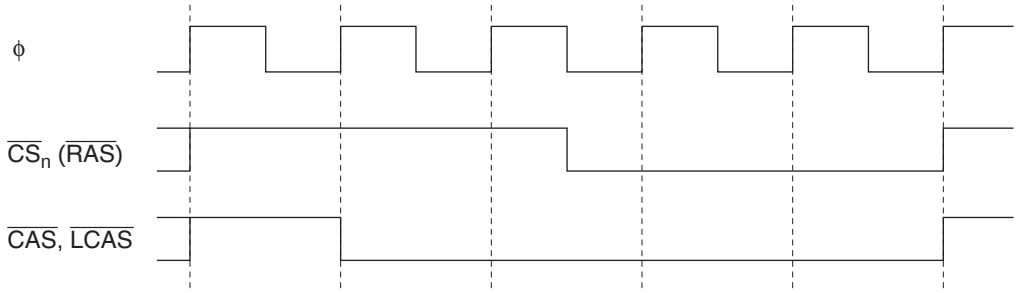


Figure 6.25 CBR Refresh Timing

When the RCW bit is set to 1, $\overline{RAS}$ signal output is delayed by one cycle. Use bits RLW1 and RLW0 to adjust the width of the $\overline{RAS}$ signal. These bits are only enabled in refresh operations.

Figure 6.26 shows the timing when the RCW bit is set to 1.



Note: n = 2 to 5

Figure 6.26 CBR Refresh Timing (When RCW = 1, RLW1 = 0, RLW0 = 1)

To select self-refreshing, set the RFSHE bit and RMODE bit in DRAMCR to 1. When a SLEEP instruction is executed to enter software standby mode, the $\overline{CAS}$ and $\overline{RAS}$ signals are output and DRAM enters self-refresh mode, as shown in figure 6.27.

When software standby mode is exited, the RMODE bit is cleared to 0 and self-refresh mode is exited.

If a CBR refresh request occurs when making a transition to software standby mode, CBR refreshing is executed, then self-refresh mode is entered.

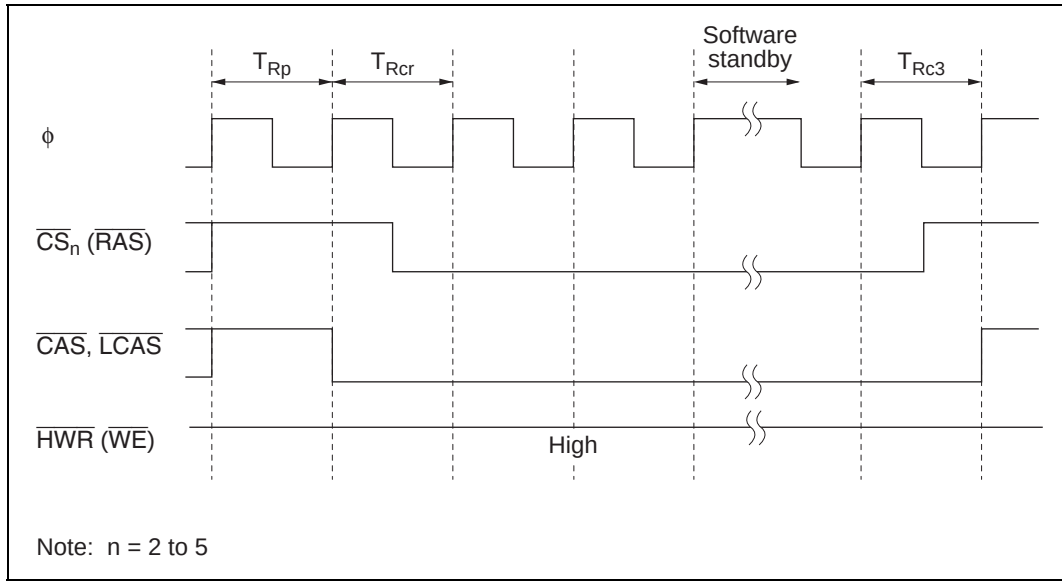


Figure 6.27 Self-Refresh Timing

When burst mode is selected with the DRAM interface, the DACK output timing can be selected with the DDS bit. When DRAM space is accessed in DMAC single address mode at the same time, the DDS bit selects whether or not burst access is to be performed.

6.6.1 When DDS = 1

Burst access is performed by determining the address only, irrespective of the bus master. With the DRAM interface, the $\overline{\text{DACK}}$ output goes low from the T_{c1} state.

Figure 6.28 shows the $\overline{\text{DACK}}$ output timing for the DRAM interface when DDS = 1.

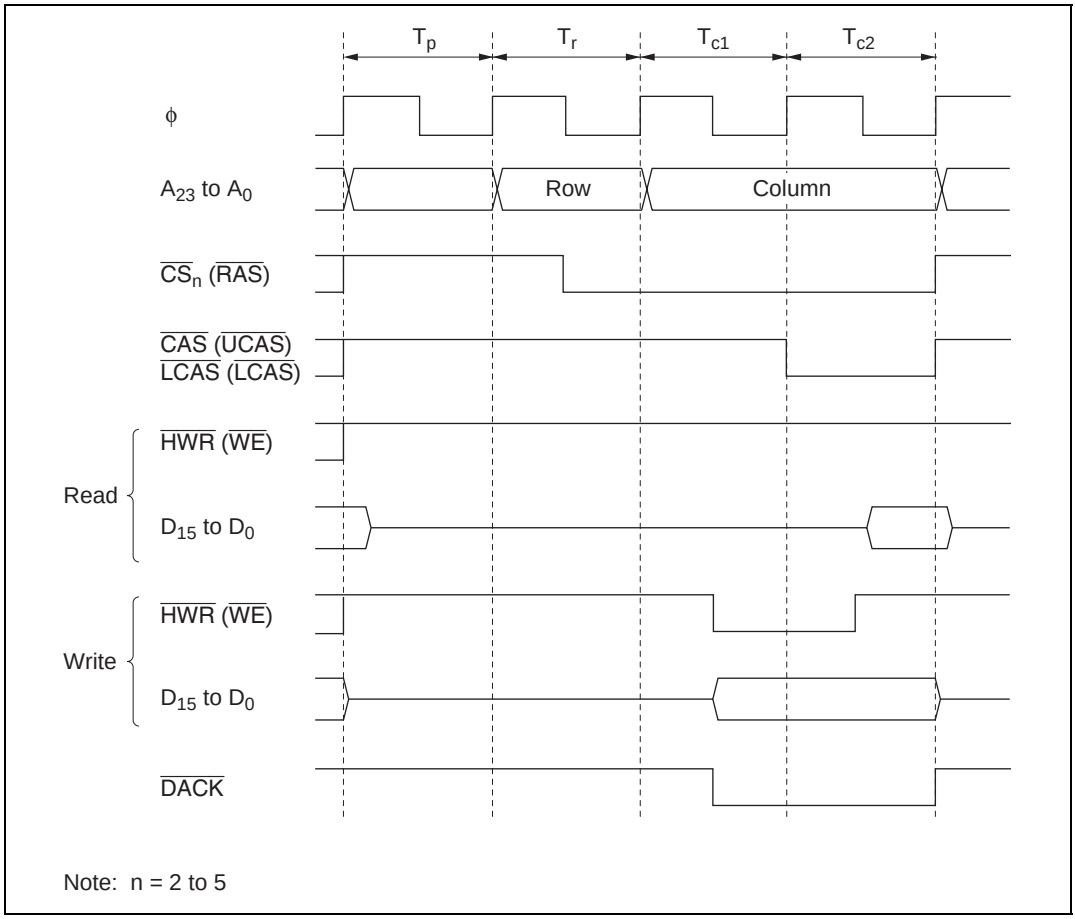


Figure 6.28 $\overline{\text{DACK}}$ Output Timing when DDS = 1 (Example of DRAM Access)

When DRAM space is accessed in DMAC single address mode, full access (normal access) is always performed. With the DRAM interface, the $\overline{\text{DACK}}$ output goes low from the T_r state.

In modes other than DMAC single address mode, burst access can be used when accessing DRAM space.

Figure 6.29 shows the $\overline{\text{DACK}}$ output timing for the DRAM interface when DDS = 0.

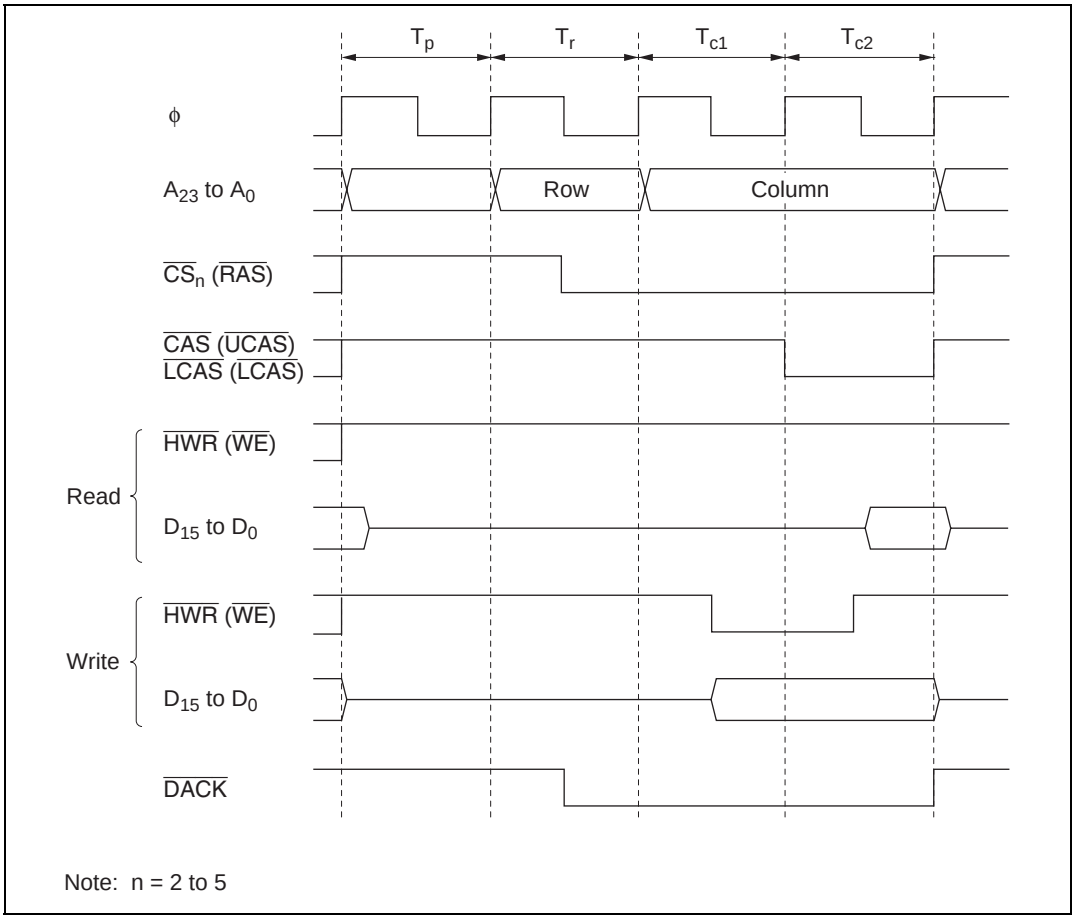


Figure 6.29 $\overline{\text{DACK}}$ Output Timing when DDS = 0 (Example of DRAM Access)

6.7.1 Overview

With the chip, external space area 0 can be designated as burst ROM space, and burst ROM interfacing performed. The burst ROM space interface enables 16-bit ROM with burst access capability to be accessed at high speed.

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

6.7.2 Basic Timing

The number of states in the initial cycle (full access) of the burst ROM interface is determined by the setting of the AST0 bit in ASTCR. When the AST0 bit is set to 1, wait state insertion is also 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 functions as 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 6.30 (a) and (b). The timing shown in figure 6.30 (a) is for the case where the AST0 and BRSTS1 bits are both set to 1, and that in figure 6.30 (b) is for the case where both these bits are cleared to 0.

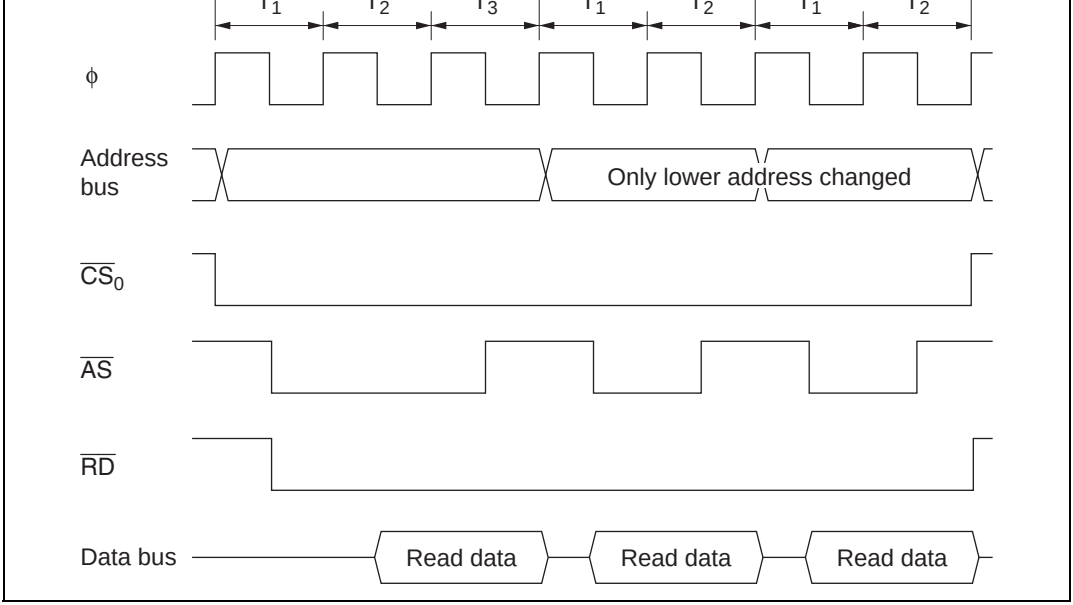


Figure 6.30 (a) Example of Burst ROM Access Timing (When AST0 = BRSTS1= 1)

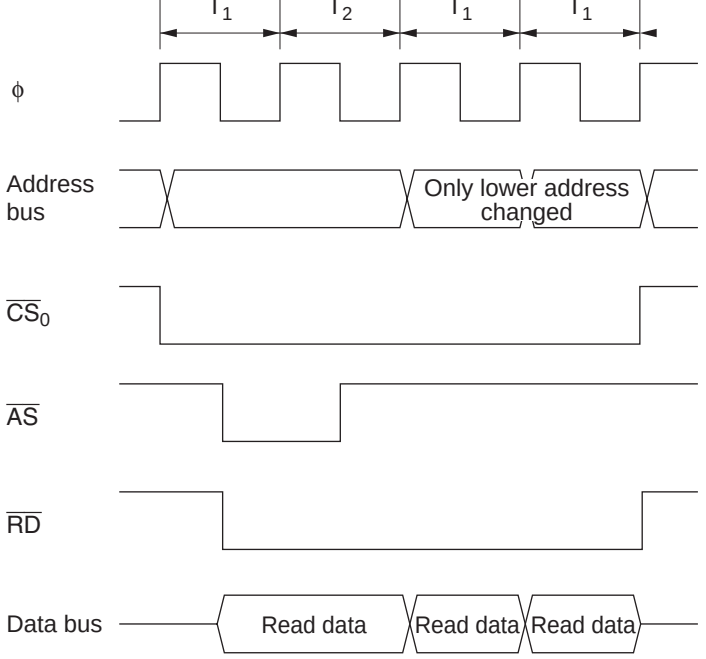


Figure 6.30 (b) Example of Burst ROM Access Timing (When AST0 = BRSTS1 = 0)

6.7.3 Wait Control

As with the basic bus interface, either program wait insertion or pin wait insertion using the $\overline{\text{WAIT}}$ pin can be used in the initial cycle (full access) on the burst ROM interface. See section 6.4.5, Wait Control.

Wait states cannot be inserted in a burst cycle.

6.8.1 Operation

When the chip accesses external space, it can insert a 1-state idle cycle (T_I) between bus cycles in the following two cases: (1) when read accesses in different areas occur consecutively, and (2) when a write cycle occurs immediately after a read cycle. By inserting an idle cycle it is possible, for example, to avoid data collisions between ROM, etc., with a long output floating time, and high-speed memory, I/O interfaces, and so on.

Consecutive Reads in Different Areas: If consecutive reads in different areas occur while the ICIS1 bit in BCRH is set to 1, an idle cycle is inserted at the start of the second read cycle. This is enabled in advanced mode.

Figure 6.31 shows an example of the operation in this case. In this example, bus cycle A is a read cycle for ROM with a long output floating time, and bus cycle B is a read cycle for SRAM, each being located in a different area. In (a), an idle cycle is not inserted, and a collision occurs in bus cycle B between the read data from ROM and that from SRAM. In (b), an idle cycle is inserted, and a data collision is prevented.

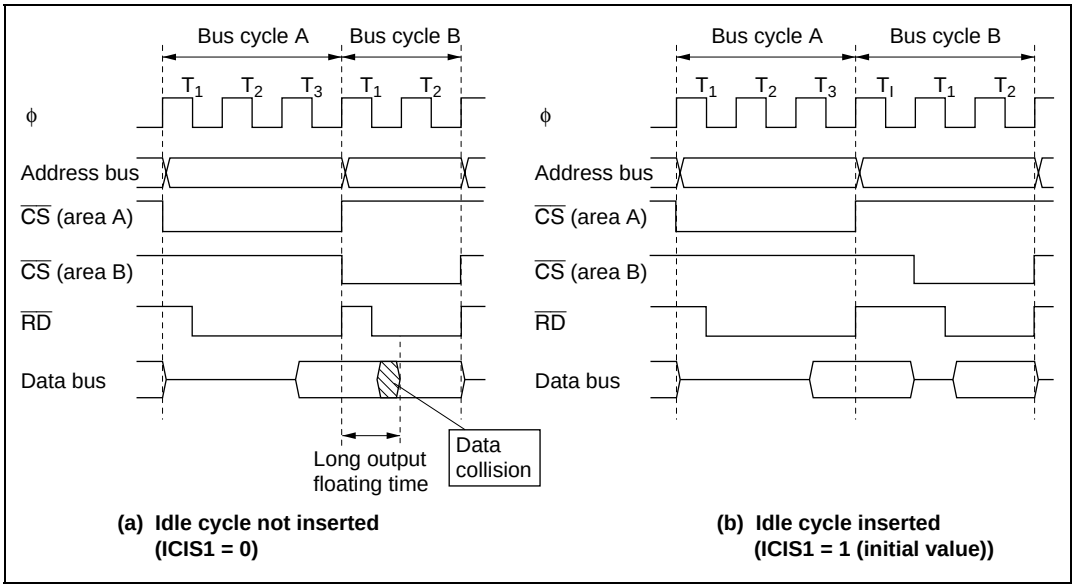


Figure 6.31 Example of Idle Cycle Operation (1)

Figure 6.32 shows an example of the operation in this case. In this example, bus cycle A is a read cycle for ROM with a long output floating time, and bus cycle B is a CPU write cycle. In (a), an idle cycle is not inserted, and a collision occurs in bus cycle B between the read data from ROM and the CPU write data. In (b), an idle cycle is inserted, and a data collision is prevented.

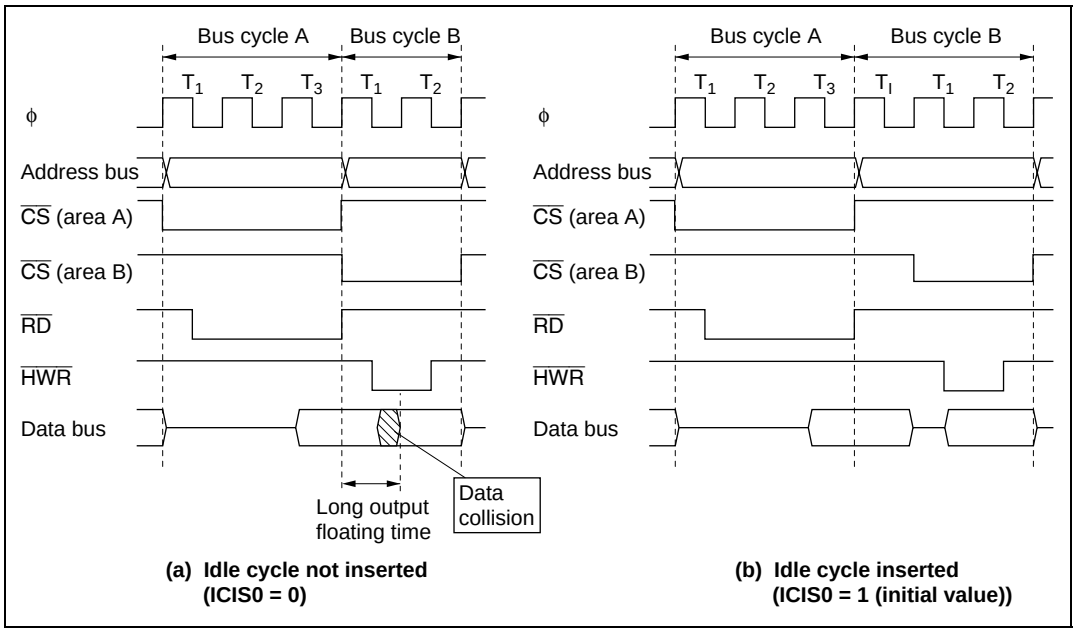


Figure 6.32 Example of Idle Cycle Operation (2)

figure 6.33.

In this case, with the setting for no idle cycle insertion (a), there may be a period of overlap between the bus cycle A $\overline{RD}$ signal and the bus cycle B $\overline{CS}$ signal.

Setting idle cycle insertion, as in (b), however, will prevent any overlap between the $\overline{RD}$ and $\overline{CS}$ signals.

In the initial state after reset release, idle cycle insertion (b) is set.

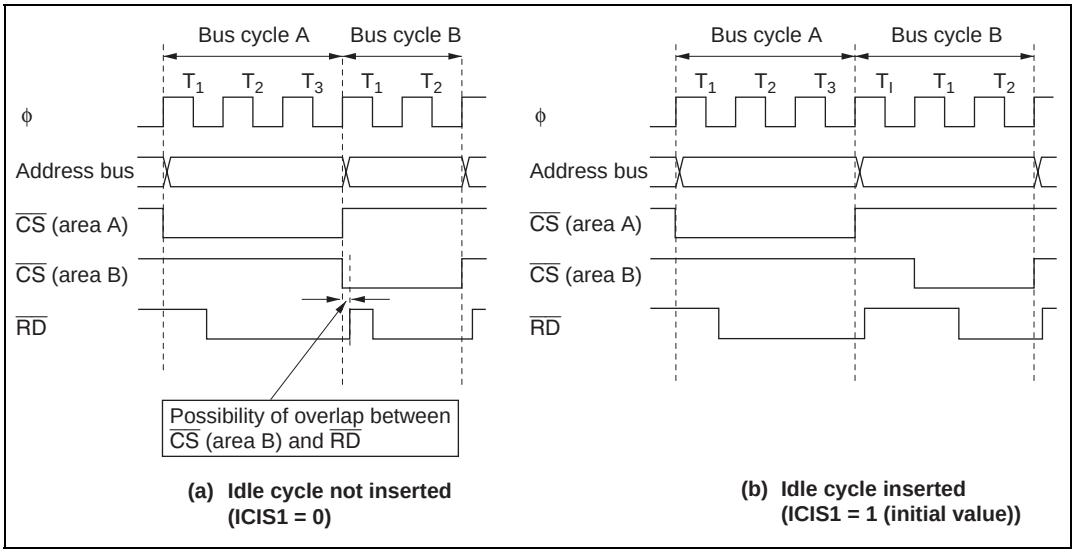


Figure 6.33 Relationship between Chip Select ($\overline{CS}$) and Read ($\overline{RD}$)

access, only a T_p cycle is inserted, and a T_I cycle is not. The timing in this case is shown in figure 6.34. However, in burst access in RAS down mode, the settings of these bits are enabled and an idle cycle is inserted. The timing in this case is shown in figures 6.35 (a) and (b).

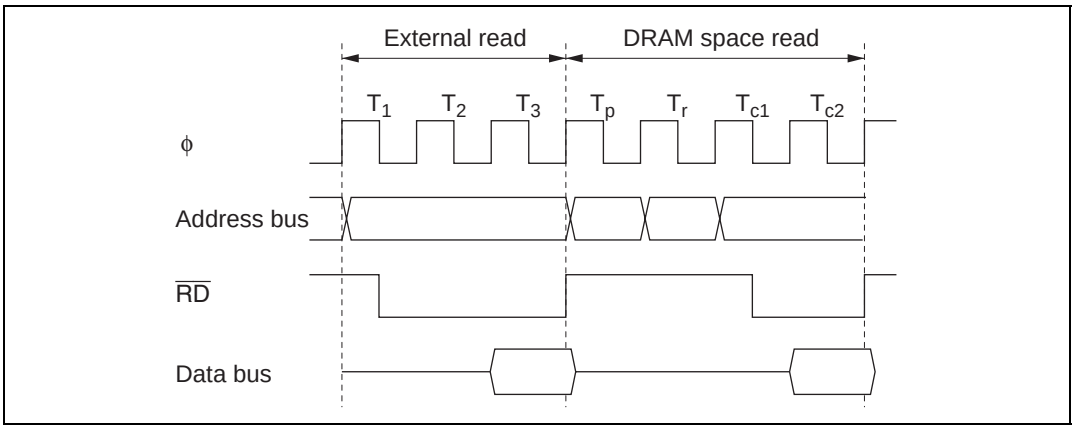


Figure 6.34 Example of DRAM Access after External Read

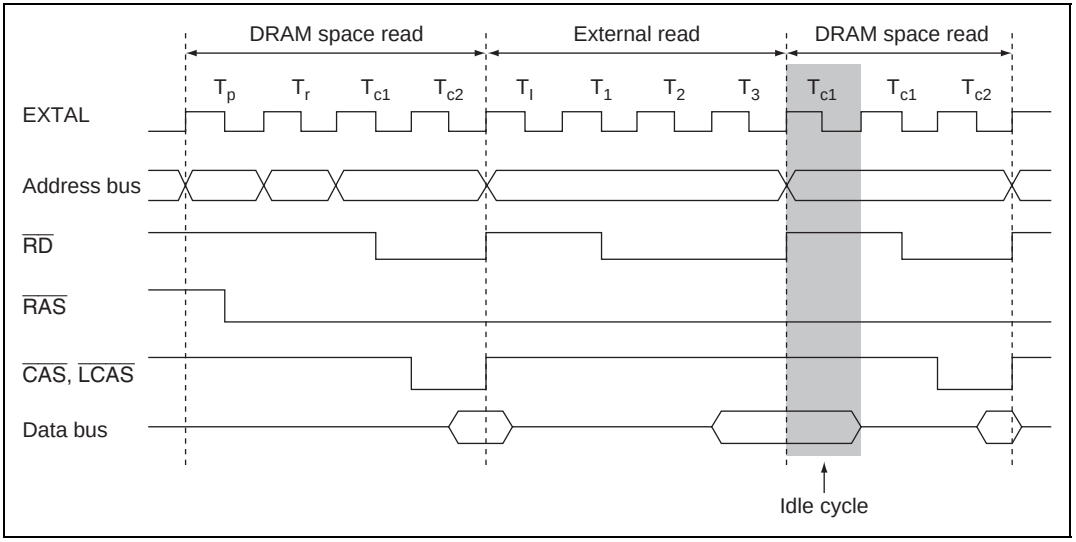


Figure 6.35 (a) Example of Idle Cycle Insertion in RAS Down Mode (ICIS1 = 1)

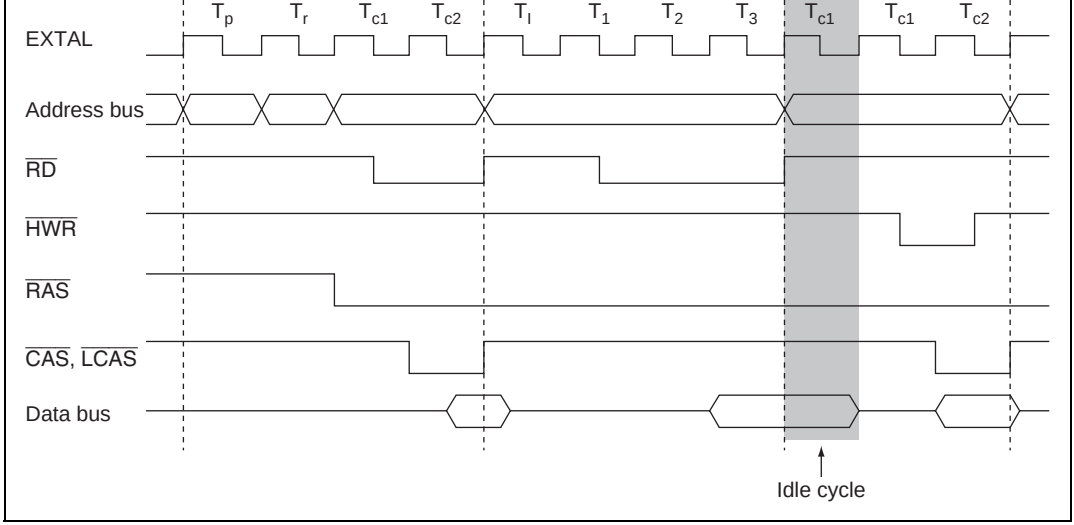


Figure 6.35 (b) Example of Idle Cycle Insertion in RAS Down Mode (ICIS0 = 1)

6.8.2 Pin States in Idle Cycle

Table 6.7 shows the pin states in an idle cycle.

Table 6.7 Pin States in Idle Cycle

Pins	Pin State
A_{23} to A_0	Contents of following bus cycle
D_{15} to D_0	High impedance
$\overline{CS_n}$ *2	High*1
$\overline{CAS}$	High
$\overline{AS}$	High
$\overline{RD}$	High
$\overline{HWR}$	High
$\overline{LWR}$	High
$\overline{DACK_m}$ *3	High

Notes: 1. Remains low in DRAM space RAS down mode or a refresh cycle.
2. n = 0 to 7
3. m = 0 or 1

The chip has a write data buffer function for the external data bus. Using the write data buffer function enables external writes and DMA single address mode transfers to be executed in parallel with internal accesses. The write data buffer function is made available by setting the WDBE bit in BCRL to 1.

Figure 6.36 shows an example of the timing when the write data buffer function is used. When this function is used, if an external write or DMA single address mode transfer continues for two states or longer, and there is an internal access next, an external write only is executed in the first state, but from the next state onward an internal access (on-chip memory or internal I/O register read/write) is executed in parallel with the external write rather than waiting until it ends.

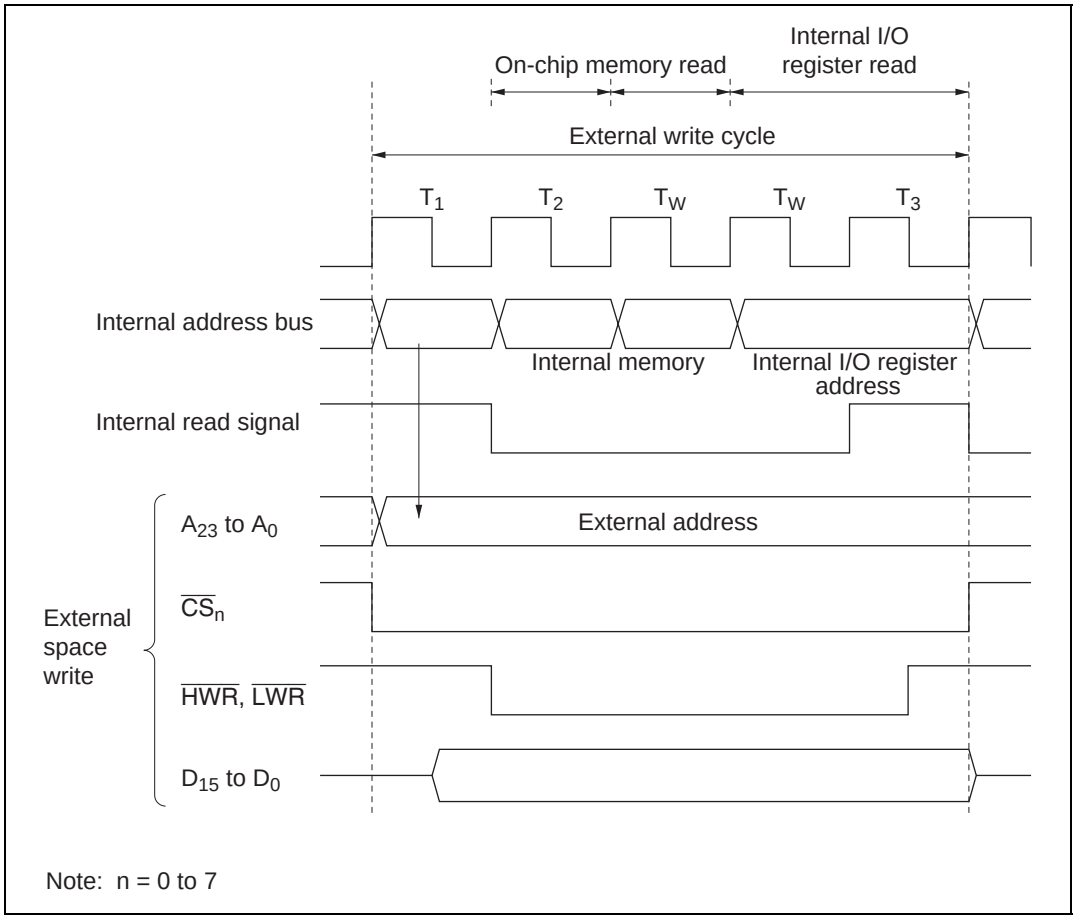


Figure 6.36 Example of Timing when Write Data Buffer Function is Used

6.10.1 Overview

The chip can release the external bus in response to a bus request from an external device. In the external bus-released state, the internal bus master continues to operate as long as there is no external access.

If an internal bus master wants to make an external access in the external bus-released state, or if a refresh request is generated, it can issue a request off-chip for the bus request to be dropped.

The BREQOPS bit can be used to change the $\overline{\text{BREQO}}$ output pin from PF_2 to $P5_3$.

6.10.2 Operation

In external expanded mode, the bus can be released to an external device by setting the BRLE bit in BCRL to 1. Driving the $\overline{\text{BREQ}}$ pin low issues an external bus request to the chip. When the $\overline{\text{BREQ}}$ pin is sampled, at the prescribed timing the $\overline{\text{BACK}}$ pin is 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 internal bus. When an internal bus master wants to make an external access, it temporarily defers activation of the bus cycle, and waits for the bus request from the external bus master to be dropped. If a refresh request is generated in the external bus-released state, refresh control is deferred until the external bus master drops the bus request.

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

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

If an external bus release request and external access occur simultaneously, the order of priority is as follows:

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

If a refresh request and external bus release request occur simultaneously, the order of priority is as follows:

(High) Refresh > External bus release (Low)

6.10.3 Pin States in External-Bus-Released State

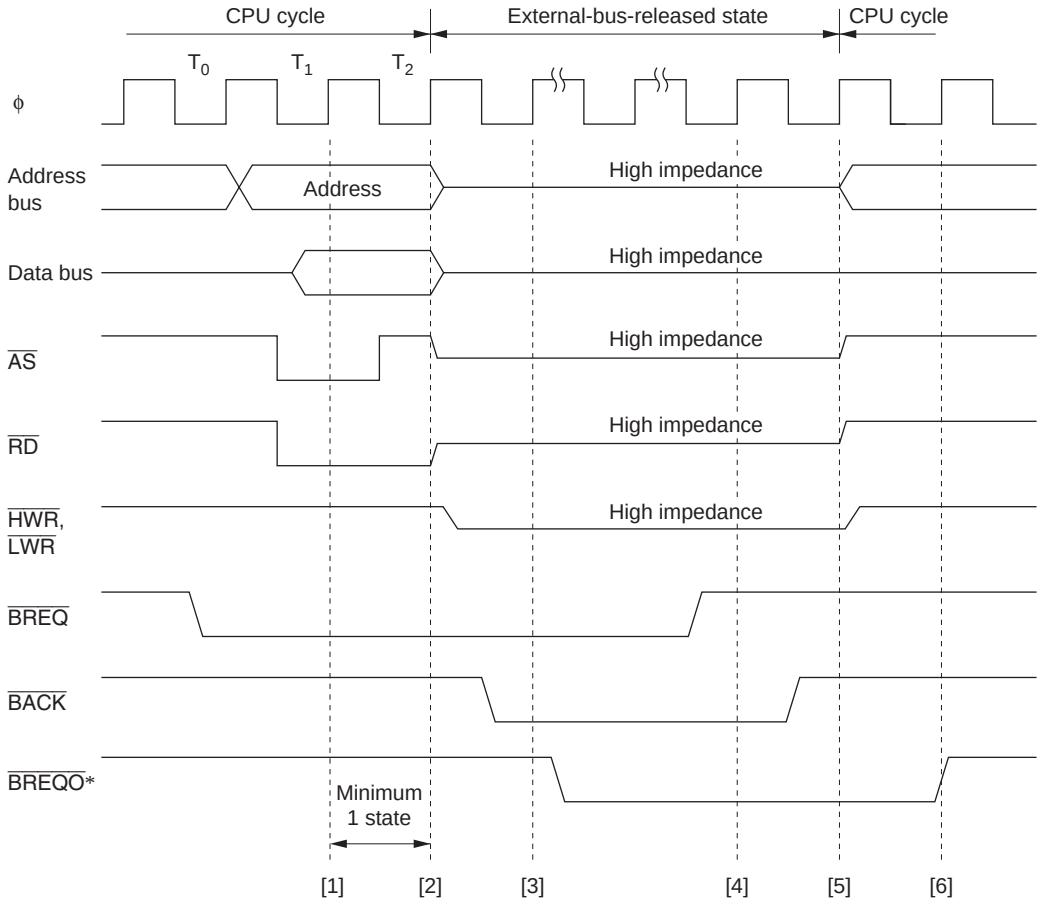
Table 6.8 shows pin states in the external-bus-released state.

Table 6.8 Pin States in Bus-Released State

Pins	Pin State
A_{23} to A_0	High impedance
D_{15} to D_0	High impedance
$\overline{CS_n}$*1	High impedance
$\overline{CAS}$	High impedance
$\overline{AS}$	High impedance
$\overline{RD}$	High impedance
$\overline{HWR}$	High impedance
$\overline{LWR}$	High impedance
$\overline{DACK_m}$*2	High

Notes: 1. n = 0 to 7
2. m = 0 or 1

Figure 6.37 shows the timing for transition to the bus-released state.



[1] Low level of $\overline{\text{BREQ}}$ pin is sampled at fall 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 rise of $\overline{\text{BACK}}$ signal.

Note: * Output only when BREQOE = 1.

Figure 6.37 Bus-Released State Transition Timing

If MSTPCR is set to H'FFFF or H'EFFF and a transition is made to sleep mode, the external bus release function will halt. Therefore, these settings should not be used.

6.11 Bus Arbitration

6.11.1 Overview

The chip has a bus arbiter that arbitrates bus master operations.

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

6.11.2 Operation

The bus arbiter monitors the bus masters' bus request signals, and if the bus is requested, sends a bus request acknowledge signal to the bus master making the request. If there are bus requests from more than one bus master, the bus request acknowledge signal is sent to the one with the highest priority. When a bus master receives the bus request acknowledge signal, it takes possession of the bus until that signal is canceled.

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

(High) DMAC > DTC > CPU (Low)

An external access by an internal bus master, external bus release, and a refresh can be executed in parallel.

If an external bus release request, a refresh request, and an external access by an internal bus master occur simultaneously, the order of priority is as follows:

(High) Refresh > External bus release (Low)

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

As a refresh and an external access by an internal bus master can be executed simultaneously, there is no relative order of priority for these two operations.

Even if a bus request is received from a bus master with a higher priority than that of the bus master that has acquired the bus and is currently operating, the bus is not necessarily transferred immediately. There are specific times at which each bus master can relinquish the bus.

CPU: The CPU is the lowest-priority bus master, and if a bus request is received from the DTC or DMAC, the bus arbiter transfers the bus to the bus master that issued the request. The timing for transfer of the bus is as follows:

- The bus is transferred at a break between bus cycles. However, if a bus cycle is executed in discrete operations, as in the case of a longword-size access, the bus is not transferred between the component operations. For details of times when the bus is not transferred, see appendix A.5, Bus States during Instruction Execution.
- 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 is generated.

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

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

In the case of an external request in short address mode or normal mode, and in cycle steal mode, the DMAC releases the bus after a single transfer.

In block transfer mode, it releases the bus after transfer of one block, and in burst mode, after completion of the transfer.

6.11.4 External Bus Release Usage Note

External bus release can be performed on completion of an external bus cycle. The $\overline{\text{RD}}$ signal and the DRAM interface $\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ signals remain low until the end of the external bus cycle. Therefore, when external bus release is performed, the $\overline{\text{RD}}$, $\overline{\text{RAS}}$, and $\overline{\text{CAS}}$ signals may change from the low level to the high-impedance state.

In a reset, the chip, including the bus controller, enters the reset state immediately, and any executing bus cycle is aborted.

7.1 Overview

The chip has a built-in DMA controller (DMAC) which can carry out data transfer on up to 4 channels.

7.1.1 Features

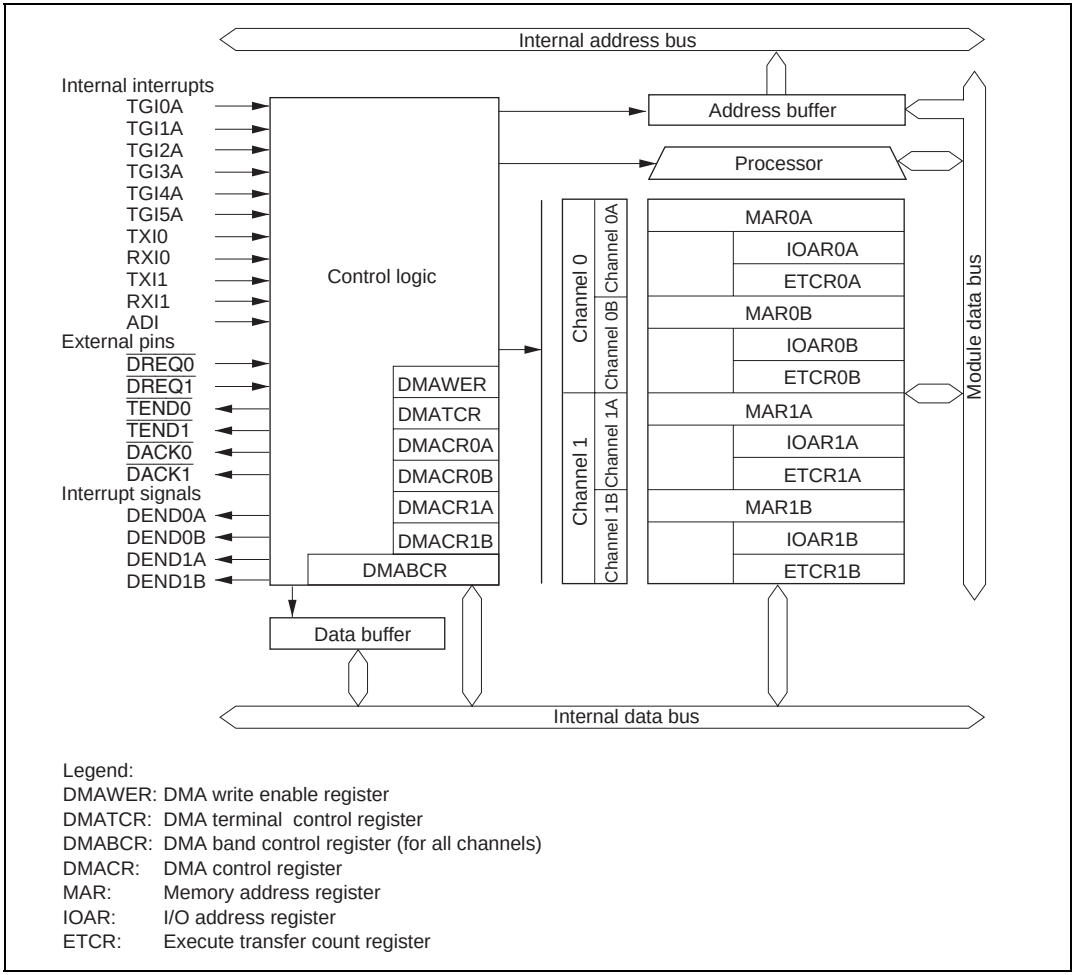
The features of the DMAC are listed below.

- Choice of short address mode or full address mode
  Short address mode
  - Maximum of 4 channels can be used
  - Choice of dual address mode or single address mode
  - In dual address mode, one of the two addresses, transfer source and transfer destination, is specified as 24 bits and the other as 16 bits
  - In single address mode, transfer source or transfer destination address only is specified as 24 bits
  - In single address mode, transfer can be performed in one bus cycle
  - Choice of sequential mode, idle mode, or repeat mode for dual address mode and single address mode

  Full address mode
  - Maximum of 2 channels can be used
  - Transfer source and transfer destination address specified as 24 bits
  - Choice of normal mode or block transfer mode
- 16-Mbyte address space can be specified directly
- Byte or word can be set as the transfer unit
- Activation sources: internal interrupt, external request, auto-request (depending on transfer mode)
  - Six 16-bit timer-pulse unit (TPU) compare match/input capture interrupts
  - Serial communication interface (SCI0, SCI1) transmit-data-empty interrupt, receive-data-full interrupt
  - A/D converter conversion end interrupt
  - External request
  - Auto-request
- Module stop mode can be set

7.1.2 Block Diagram

A block diagram of the DMAC is shown in figure 7.1.



Legend:
DMAWER: DMA write enable register
DMATCR: DMA terminal control register
DMABCR: DMA band control register (for all channels)
DMACR: DMA control register
MAR: Memory address register
IOAR: I/O address register
ETCR: Execute transfer count register

Figure 7.1 Block Diagram of DMAC

Tables 7.1 (1) and (2) summarize DMAC functions in short address mode and full address mode, respectively.

Table 7.1 (1) Overview of DMAC Functions (Short Address Mode)

		Address Register Bit Length	
Transfer Mode	**Transfer Source**	**Source**	**Destination**
Dual address mode • Sequential mode — 1-byte or 1-word transfer executed for one transfer request — Memory address incremented/decremented by 1 or 2 — 1 to 65,536 transfers • Idle mode — 1-byte or 1-word transfer executed for one transfer request — Memory address fixed — 1 to 65,536 transfers • Repeat mode — 1-byte or 1-word transfer executed for one transfer request — Memory address incremented/ decremented by 1 or 2 — After specified number of transfers (1 to 256), initial state is restored and operation continues	• TPU channel 0 to 5 compare match/input capture A interrupt • SCI transmit-data-empty interrupt • SCI receive-data-full interrupt • A/D converter conversion end interrupt • External request	24/16	16/24
Single address mode • 1-byte or 1-word transfer executed for one transfer request • Transfer in 1 bus cycle using $\overline{\text{DACK}}$ pin in place of address specifying I/O • Specifiable for sequential, idle, and repeat modes	• External request	24/$\overline{\text{DACK}}$	$\overline{\text{DACK}}$/24

Transfer Mode	Transfer Source	Address Register Bit Length	
		Source	Destination
• Normal mode Auto-request — Transfer request retained internally — Transfers continue for the specified number of times (1 to 65,536) — Choice of burst or cycle steal transfer	• Auto-request	24	24
External request — 1-byte or 1-word transfer executed for one transfer request — 1 to 65,536 transfers	• External request		
• Block transfer mode — Specified block size transfer executed for one transfer request — 1 to 65,536 transfers — Either source or destination specifiable as block area — Block size: 1 to 256 bytes or words	• TPU channel 0 to 5 compare match/input capture A interrupt • SCI transmit-data-empty interrupt • SCI receive-data-full interrupt • External request • A/D converter conversion end interrupt	24	24

Table 7.2 summarizes the DMAC pins.

In short address mode, external request transfer, single address transfer, and transfer end output are not performed for channel A.

The DMA transfer acknowledge function is used in channel B single address mode in short address mode.

When the $\overline{\text{DREQ}}$ pin is used, do not designate the corresponding port for output.

With regard to the $\overline{\text{DACK}}$ pins, setting single address transfer automatically sets the corresponding port to output, functioning as a $\overline{\text{DACK}}$ pin.

With regard to the $\overline{\text{TEND}}$ pins, whether or not the corresponding port is used as a $\overline{\text{TEND}}$ pin can be specified by means of a register setting.

Table 7.2 DMAC Pins

Channel	Pin Name	Symbol	I/O	Function
0	DMA request 0	$\overline{\text{DREQ0}}$	Input	DMAC channel 0 external request
	DMA transfer acknowledge 0	$\overline{\text{DACK0}}$	Output	DMAC channel 0 single address transfer acknowledge
	DMA transfer end 0	$\overline{\text{TEND0}}$	Output	DMAC channel 0 transfer end
1	DMA request 1	$\overline{\text{DREQ1}}$	Input	DMAC channel 1 external request
	DMA transfer acknowledge 1	$\overline{\text{DACK1}}$	Output	DMAC channel 1 single address transfer acknowledge
	DMA transfer end 1	$\overline{\text{TEND1}}$	Output	DMAC channel 1 transfer end

Table 7.3 summarizes the DMAC registers.

Table 7.3 DMAC Registers

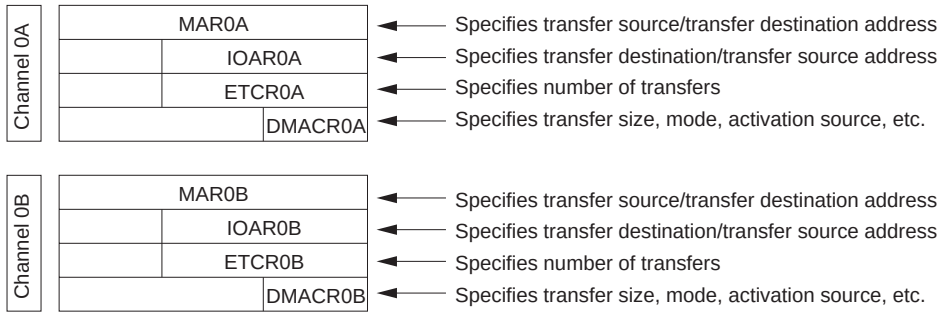
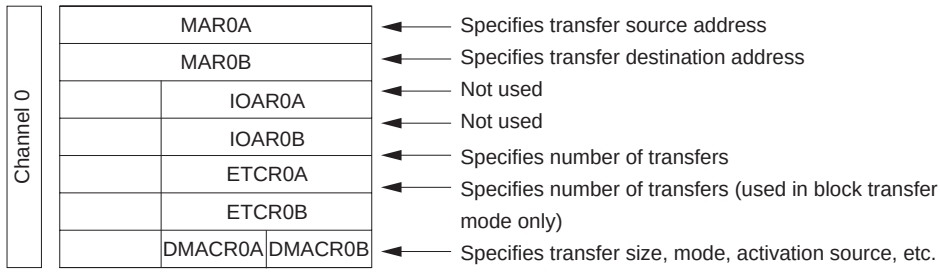
Channel	Name	Abbreviation	R/W	Initial Value	Address*	Bus Width
0	Memory address register 0A	MAR0A	R/W	Undefined	H'FEE0	16 bits
	I/O address register 0A	IOAR0A	R/W	Undefined	H'FEE4	16 bits
	Transfer count register 0A	ETCR0A	R/W	Undefined	H'FEE6	16 bits
	Memory address register 0B	MAR0B	R/W	Undefined	H'FEE8	16 bits
	I/O address register 0B	IOAR0B	R/W	Undefined	H'FEEC	16 bits
	Transfer count register 0B	ETCR0B	R/W	Undefined	H'FEEE	16 bits
1	Memory address register 1A	MAR1A	R/W	Undefined	H'FEF0	16 bits
	I/O address register 1A	IOAR1A	R/W	Undefined	H'FEF4	16 bits
	Transfer count register 1A	ETCR1A	R/W	Undefined	H'FEF6	16 bits
	Memory address register 1B	MAR1B	R/W	Undefined	H'FEF8	16 bits
	I/O address register 1B	IOAR1B	R/W	Undefined	H'FEFC	16 bits
	Transfer count register 1B	ETCR1B	R/W	Undefined	H'FEFE	16 bits
0, 1	DMA write enable register	DMAWER	R/W	H'00	H'FF00	8 bits
	DMA terminal control register	DMATCR	R/W	H'00	H'FF01	8 bits
	DMA control register 0A	DMACR0A	R/W	H'00	H'FF02	16 bits
	DMA control register 0B	DMACR0B	R/W	H'00	H'FF03	16 bits
	DMA control register 1A	DMACR1A	R/W	H'00	H'FF04	16 bits
	DMA control register 1B	DMACR1B	R/W	H'00	H'FF05	16 bits
	DMA band control register	DMABCR	R/W	H'0000	H'FF06	16 bits
	Module stop control register	MSTPCR	R/W	H'3FFF	H'FF3C	8 bits

Note: * Lower 16 bits of the address.

Short address mode transfer can be performed for channels A and B independently.

Short address mode transfer is specified for each channel by clearing the FAE bit in DMABCR to 0, as shown in table 7.4. Short address mode or full address mode can be selected for channels 1 and 0 independently by means of bits FAE1 and FAE0.

Table 7.4 Short Address Mode and Full Address Mode (For 1 Channel: Example of Channel 0)

FAE0	Description
0	Short address mode specified (channels A and B operate independently) 
1	Full address mode specified (channels A and B operate in combination) 

Bit	:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MAR	:	—	—	—	—	—	—	—	—								
Initial value	:	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

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAR	:																
Initial value	:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
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/W	R/W

*: Undefined

MAR is a 32-bit readable/writable register that specifies the transfer source address or destination address.

The upper 8 bits of MAR are reserved: they are always read as 0, and cannot be modified.

Whether MAR functions as the source address register or as the destination address register can be selected by means of the DTDIR bit in DMACR.

MAR is incremented or decremented each time a byte or word transfer is executed, so that the address specified by MAR is constantly updated. For details, see section 7.2.4, DMA Control Register (DMACR).

MAR is not initialized by a reset or in standby mode.

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IOAR	:																
Initial value	:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
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/W	R/W

*: Undefined

IOAR is a 16-bit readable/writable register that specifies the lower 16 bits of the transfer source address or destination address. The upper 8 bits of the transfer address are automatically set to H'FF.

Whether IOAR functions as the source address register or as the destination address register can be selected by means of the DTDIR bit in DMACR.

IOAR is invalid in single address mode.

IOAR is not incremented or decremented each time a transfer is executed, so the address specified by IOAR is fixed.

IOAR is not initialized by a reset or in standby mode.

7.2.3 Execute Transfer Count Register (ETCR)

ETCR is a 16-bit readable/writable register that specifies the number of transfers. The setting of this register is different for sequential mode and idle mode on the one hand, and for repeat mode on the other.

Sequential Mode and Idle Mode

Transfer Counter (ETCR)

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	:																
Initial value	:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
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/W	R/W

*: Undefined

reaches H'0000, the DTE bit in DMABCR is cleared, and transfer ends.

Repeat Mode

Transfer Number Storage (ETCRH)

Bit	:	15	14	13	12	11	10	9	8
Initial value	:	*	*	*	*	*	*	*	*
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Transfer Counter (ETCRL)

Bit	:	7	6	5	4	3	2	1	0
Initial value	:	*	*	*	*	*	*	*	*
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

*: Undefined

In repeat mode, ETCR functions as transfer counter ETCRL (with a count range of 1 to 256) and transfer number storage register ETCRH. ETCRL is decremented by 1 each time a transfer is performed, and when the count reaches H'00, ETCRL is loaded with the value in ETCRH. At this point, MAR is automatically restored to the value it had when the count was started. The DTE bit in DMABCR is not cleared, and so transfers can be performed repeatedly until the DTE bit is cleared by the user.

ETCR is not initialized by a reset or in standby mode.

Bit	:	7	6	5	4	3	2	1	0
		DTSZ	DTID5	RPE	DTDIR	DTF3	DTF2	DTF1	DTF0
Initial value	:	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

DMACR is an 8-bit readable/writable register that controls the operation of each DMAC channel.

DMACR is initialized to H'00 by a reset, and in standby mode.

Bit 7—Data Transfer Size (DTSZ): Selects the size of data to be transferred at one time.

Bit 7 DTSZ	Description	
0	Byte-size transfer	(Initial value)
1	Word-size transfer	

Bit 6—Data Transfer Increment/Decrement (DTID): Selects incrementing or decrementing of MAR after every data transfer in sequential mode or repeat mode.

In idle mode, MAR is neither incremented nor decremented.

Bit 6 DTID	Description	
0	MAR is incremented after a data transfer	(Initial value)
	• When DTSZ = 0, MAR is incremented by 1 after a transfer	
	• When DTSZ = 1, MAR is incremented by 2 after a transfer	
1	MAR is decremented after a data transfer	
	• When DTSZ = 0, MAR is decremented by 1 after a transfer	
	• When DTSZ = 1, MAR is decremented by 2 after a transfer	

Bit 5 RPE	DMABCR DTIE	Description	
0	0	Transfer in sequential mode (no transfer end interrupt)	(Initial value)
	1	Transfer in sequential mode (with transfer end interrupt)	
1	0	Transfer in repeat mode (no transfer end interrupt)	
	1	Transfer in idle mode (with transfer end interrupt)	

For details of operation in sequential, idle, and repeat mode, see section 7.5.2, Sequential Mode, section 7.5.3, Idle Mode, and section 7.5.4, Repeat Mode.

Bit 4—Data Transfer Direction (DTDIR): Used in combination with the SAE bit in DMABCR to specify the data transfer direction (source or destination). The function of this bit is therefore different in dual address mode and single address mode.

DMABCR SAE	Bit 4 DTDIR	Description	
0	0	Transfer with MAR as source address and IOAR as destination address	(Initial value)
	1	Transfer with IOAR as source address and MAR as destination address	
1	0	Transfer with MAR as source address and $\overline{\text{DACK}}$ pin as write strobe	
	1	Transfer with $\overline{\text{DACK}}$ pin as read strobe and MAR as destination address	

B.

Channel A

Bit 3 DTF3	Bit 2 DTF2	Bit 1 DTF1	Bit 0 DTF0	Description
0	0	0	0	— (Initial value)
			1	Activated by A/D converter conversion end interrupt
		1	0	—
			1	—
	1	0	0	Activated by SCI channel 0 transmit-data-empty interrupt
			1	Activated by SCI channel 0 receive-data-full interrupt
		1	0	Activated by SCI channel 1 transmit-data-empty interrupt
			1	Activated by SCI channel 1 receive-data-full interrupt
1	0	0	0	Activated by TPU channel 0 compare match/input capture A interrupt
			1	Activated by TPU channel 1 compare match/input capture A interrupt
		1	0	Activated by TPU channel 2 compare match/input capture A interrupt
			1	Activated by TPU channel 3 compare match/input capture A interrupt
	1	0	0	Activated by TPU channel 4 compare match/input capture A interrupt
			1	Activated by TPU channel 5 compare match/input capture A interrupt
		1	0	—
			1	—

Bit 3 DTF3	Bit 2 DTF2	Bit 1 DTF1	Bit 0 DTF0	Description
0	0	0	0	— (Initial value)
			1	Activated by A/D converter conversion end interrupt
		1	0	Activated by $\overline{\text{DREQ}}$ pin falling edge input*
			1	Activated by $\overline{\text{DREQ}}$ pin low-level input
	1	0	0	Activated by SCI channel 0 transmit-data-empty interrupt
			1	Activated by SCI channel 0 receive-data-full interrupt
		1	0	Activated by SCI channel 1 transmit-data-empty interrupt
			1	Activated by SCI channel 1 receive-data-full interrupt
1	0	0	0	Activated by TPU channel 0 compare match/input capture A interrupt
			1	Activated by TPU channel 1 compare match/input capture A interrupt
		1	0	Activated by TPU channel 2 compare match/input capture A interrupt
			1	Activated by TPU channel 3 compare match/input capture A interrupt
	1	0	0	Activated by TPU channel 4 compare match/input capture A interrupt
			1	Activated by TPU channel 5 compare match/input capture A interrupt
		1	0	—
			1	—

Note: * Detected as a low level in the first transfer after transfer is enabled.

The same factor can be selected for more than one channel. In this case, activation starts with the highest-priority channel according to the relative channel priorities. For relative channel priorities, see section 7.5.13, DMAC Multi-Channel Operation.

DMABCRH

Bit	:	15	14	13	12	11	10	9	8
		FAE1	FAE0	SAE1	SAE0	DTA1B	DTA1A	DTA0B	DTA0A
Initial value	:	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

DMABCRL

Bit	:	7	6	5	4	3	2	1	0
		DTE1B	DTE1A	DTE0B	DTE0A	DTIE1B	DTIE1A	DTIE0B	DTIE0A
Initial value	:	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

DMABCR is a 16-bit readable/writable register that controls the operation of each DMAC channel.

DMABCR is initialized to H'0000 by a reset, and in hardware standby mode.

Bit 15—Full Address Enable 1 (FAE1): Specifies whether channel 1 is to be used in short address mode or full address mode.

In short address mode, channels 1A and 1B can be used as independent channels.

Bit 15 FAE1	**Description**	
0	Short address mode	(Initial value)
1	Full address mode	

In short address mode, channels 0A and 0B can be used as independent channels.

Bit 14 FAE0	Description	
0	Short address mode	(Initial value)
1	Full address mode	

Bit 13—Single Address Enable 1 (SAE1): Specifies whether channel 1B is to be used for transfer in dual address mode or single address mode.

This bit is invalid in full address mode.

Bit 13 SAE1	Description	
0	Transfer in dual address mode	(Initial value)
1	Transfer in single address mode	

Bit 12—Single Address Enable 0 (SAE0): Specifies whether channel 0B is to be used for transfer in dual address mode or single address mode.

This bit is invalid in full address mode.

Bit 12 SAE0	Description	
0	Transfer in dual address mode	(Initial value)
1	Transfer in single address mode	

Bits 11 to 8—Data Transfer Acknowledge (DTA): These bits enable or disable clearing, when DMA transfer is performed, of the internal interrupt source selected by the data transfer factor setting.

When DTE = 1 and DTA = 1, the internal interrupt source selected by the data transfer factor setting is cleared automatically by DMA transfer. When DTE = 1 and DTA = 1, the internal interrupt source selected by the data transfer factor setting does not issue an interrupt request to the CPU or DTC.

or DTC in parallel. In this case, the interrupt source should be cleared by the CPU or DTC transfer.

When DTE = 0, the internal interrupt source selected by the data transfer factor setting issues an interrupt request to the CPU or DTC regardless of the DTA bit setting.

Bit 11—Data Transfer Acknowledge 1B (DTA1B): Enables or disables clearing, when DMA transfer is performed, of the internal interrupt source selected by the channel 1B data transfer factor setting.

Bit 11 DTA1B	Description
0	Clearing of selected internal interrupt source at time of DMA transfer is disabled (Initial value)
1	Clearing of selected internal interrupt source at time of DMA transfer is enabled

Bit 10—Data Transfer Acknowledge 1A (DTA1A): Enables or disables clearing, when DMA transfer is performed, of the internal interrupt source selected by the channel 1A data transfer factor setting.

Bit 10 DTA1A	Description
0	Clearing of selected internal interrupt source at time of DMA transfer is disabled (Initial value)
1	Clearing of selected internal interrupt source at time of DMA transfer is enabled

Bit 9—Data Transfer Acknowledge 0B (DTA0B): Enables or disables clearing, when DMA transfer is performed, of the internal interrupt source selected by the channel 0B data transfer factor setting.

Bit 9 DTA0B	Description
0	Clearing of selected internal interrupt source at time of DMA transfer is disabled (Initial value)
1	Clearing of selected internal interrupt source at time of DMA transfer is enabled

factor setting.

Bit 8 DTA0A	Description
0	Clearing of selected internal interrupt source at time of DMA transfer is disabled (Initial value)
1	Clearing of selected internal interrupt source at time of DMA transfer is enabled

Bits 7 to 4—Data Transfer Enable (DTE): When DTE = 0, data transfer is disabled and the activation source selected by the data transfer factor setting is ignored. If the activation source is an internal interrupt, an interrupt request is issued to the CPU or DTC. If the DTIE bit is set to 1 when DTE = 0, the DMAC regards this as indicating the end of a transfer, and issues a transfer end interrupt request to the CPU or DTC.

The conditions for the DTE bit being cleared to 0 are as follows:

- When initialization is performed
- When the specified number of transfers have been completed in a transfer mode other than repeat mode
- When 0 is written to the DTE bit to forcibly abort the transfer, or for a similar reason

When DTE = 1, data transfer is enabled and the DMAC waits for a request by the activation source selected by the data transfer factor setting. When a request is issued by the activation source, DMA transfer is executed.

The condition for the DTE bit being set to 1 is as follows:

- When 1 is written to the DTE bit after the DTE bit is read as 0

Bit 7—Data Transfer Enable 1B (DTE1B): Enables or disables data transfer on channel 1B.

Bit 7 DTE1B	Description
0	Data transfer disabled (Initial value)
1	Data transfer enabled

Bit 6 DTE1A	Description	
0	Data transfer disabled	(Initial value)
1	Data transfer enabled	

Bit 5—Data Transfer Enable 0B (DTE0B): Enables or disables data transfer on channel 0B.

Bit 5 DTE0B	Description	
0	Data transfer disabled	(Initial value)
1	Data transfer enabled	

Bit 4—Data Transfer Enable 0A (DTE0A): Enables or disables data transfer on channel 0A.

Bit 4 DTE0A	Description	
0	Data transfer disabled	(Initial value)
1	Data transfer enabled	

Bits 3 to 0—Data Transfer End Interrupt Enable (DTIE): These bits enable or disable an interrupt to the CPU or DTC when transfer ends. If the DTIE bit is set to 1 when DTE = 0, the DMAC regards this as indicating the end of a transfer, and issues a transfer end interrupt request to the CPU or DTC.

A transfer end interrupt can be canceled either by clearing the DTIE bit to 0 in the interrupt handling routine, or by performing processing to continue transfer by setting the transfer counter and address register again, and then setting the DTE bit to 1.

Bit 3—Data Transfer Interrupt Enable 1B (DTIE1B): Enables or disables the channel 1B transfer end interrupt.

Bit 3 DTIE1B	Description	
0	Transfer end interrupt disabled	(Initial value)
1	Transfer end interrupt enabled	

Bit 2 DTIE1A	Description	
0	Transfer end interrupt disabled	(Initial value)
1	Transfer end interrupt enabled	

Bit 1—Data Transfer Interrupt Enable 0B (DTIE0B): Enables or disables the channel 0B transfer end interrupt.

Bit 1 DTIE0B	Description	
0	Transfer end interrupt disabled	(Initial value)
1	Transfer end interrupt enabled	

Bit 0—Data Transfer Interrupt Enable 0A (DTIE0A): Enables or disables the channel 0A transfer end interrupt.

Bit 0 DTIE0A	Description	
0	Transfer end interrupt disabled	(Initial value)
1	Transfer end interrupt enabled	

Full address mode transfer is performed with channels A and B together. For details of full address mode setting, see table 7.4.

7.3.1 Memory Address Register (MAR)

Bit	:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		—	—	—	—	—	—	—	—								
Initial value	:	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

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
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/W	R/W

*: Undefined

MAR is a 32-bit readable/writable register; MARA functions as the transfer source address register, and MARB as the destination address register.

MAR is composed of two 16-bit registers, MARH and MARL. The upper 8 bits of MARH are reserved: they are always read as 0, and cannot be modified.

MAR is incremented or decremented each time a byte or word transfer is executed, so that the source or destination memory address can be updated automatically. For details, see section 7.3.4, DMA Control Register (DMACR).

MAR is not initialized by a reset or in standby mode.

7.3.2 I/O Address Register (IOAR)

IOAR is not used in full address transfer.

ETCR is a 16-bit readable/writable register that specifies the number of transfers. The function of this register is different in normal mode and in block transfer mode.

ETCR is not initialized by a reset or in standby mode.

Normal Mode

ETCRA

Transfer Counter

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
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/W	R/W

*: Undefined

In normal mode, ETCRA functions as a 16-bit transfer counter. ETCRA is decremented by 1 each time a transfer is performed, and transfer ends when the count reaches H'0000. ETCRB is not used at this time.

ETCRB

ETCRB is not used in normal mode.

ETCRA

Block Size Storage (ETCRAH)

Bit	:	15	14	13	12	11	10	9	8
Initial value	:	*	*	*	*	*	*	*	*
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Block Size Counter (ETCRAL)

Bit	:	7	6	5	4	3	2	1	0
Initial value	:	*	*	*	*	*	*	*	*
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

*: Undefined

ETCRB

Block Transfer Counter

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
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/W	R/W

*: Undefined

In block transfer mode, ETCRAL functions as an 8-bit block size counter and ETCRAH holds the block size. ETCRAL is decremented each time a 1-byte or 1-word transfer is performed, and when the count reaches H'00, ETCRAL is loaded with the value in ETCRAH. So by setting the block size in ETCRAH and ETCRAL, it is possible to repeatedly transfer blocks consisting of any desired number of bytes or words.

ETCRB functions in block transfer mode, as a 16-bit block transfer counter. ETCRB is decremented by 1 each time a block is transferred, and transfer ends when the count reaches H'0000.

DMACR is a 16-bit readable/writable register that controls the operation of each DMAC channel. In full address mode, DMACRA and DMACRB have different functions.

DMACR is initialized to H'0000 by a reset, and in hardware standby mode.

DMACRA

Bit	:	15	14	13	12	11	10	9	8
		DTSZ	SAID	SAIDE	BLKDIR	BLKE	—	—	—
Initial value	:	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

DMACRB

Bit	:	7	6	5	4	3	2	1	0
		—	DAID	DAIDE	—	DTF3	DTF2	DTF1	DTF0
Initial value	:	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

Bit 15—Data Transfer Size (DTSZ): Selects the size of data to be transferred at one time.

Bit 15 DTSZ	Description	
0	Byte-size transfer	(Initial value)
1	Word-size transfer	

source address register MARA is to be incremented, decremented, or left unchanged, when data transfer is performed.

Bit 14 SAID	Bit 13 SAIDE	Description	
0	0	MARA is fixed	(Initial value)
	1	MARA is incremented after a data transfer • When DTSZ = 0, MARA is incremented by 1 after a transfer • When DTSZ = 1, MARA is incremented by 2 after a transfer	
1	0	MARA is fixed	
	1	MARA is decremented after a data transfer • When DTSZ = 0, MARA is decremented by 1 after a transfer • When DTSZ = 1, MARA is decremented by 2 after a transfer	

Bit 12—Block Direction (BLKDIR)

Bit 11—Block Enable (BLKE): These bits specify whether normal mode or block transfer mode is to be used. If block transfer mode is specified, the BLKDIR bit specifies whether the source side or the destination side is to be the block area.

Bit 12 BLKDIR	Bit 11 BLKE	Description	
0	0	Transfer in normal mode	(Initial value)
	1	Transfer in block transfer mode, destination side is block area	
1	0	Transfer in normal mode	
	1	Transfer in block transfer mode, source side is block area	

For operation in normal mode and block transfer mode, see section 7.5, Operation.

Bits 10 to 7—Reserved: Can be read or written to. Only 0 should be written to these bits.

whether destination address register MARB is to be incremented, decremented, or left unchanged, when data transfer is performed.

Bit 6 DAID	Bit 5 DAIDE	Description	
0	0	MARB is fixed	(Initial value)
	1	MARB is incremented after a data transfer • When DTSZ = 0, MARB is incremented by 1 after a transfer • When DTSZ = 1, MARB is incremented by 2 after a transfer	
1	0	MARB is fixed	
	1	MARB is decremented after a data transfer • When DTSZ = 0, MARB is decremented by 1 after a transfer • When DTSZ = 1, MARB is decremented by 2 after a transfer	

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

Bits 3 to 0—Data Transfer Factor (DTF3 to DTF0): These bits select the data transfer factor (activation source). The factors that can be specified differ between normal mode and block transfer mode.

- Normal Mode

Bit 3 DTF3	Bit 2 DTF2	Bit 1 DTF1	Bit 0 DTF0	Description	
0	0	0	0	—	(Initial value)
			1	—	
		1	0	Activated by $\overline{\text{DREQ}}$ pin falling edge input	
			1	Activated by $\overline{\text{DREQ}}$ pin low-level input	
	1	0	*	—	
		1	0	Auto-request (cycle steal)	
			1	Auto-request (burst)	
1	*	*	*	—	

*: Don't care

Bit 3 DTF3	Bit 2 DTF2	Bit 1 DTF1	Bit 0 DTF0	Description
0	0	0	0	— (Initial value)
			1	Activated by A/D converter conversion end interrupt
		1	0	Activated by $\overline{\text{DREQ}}$ pin falling edge input*
			1	Activated by $\overline{\text{DREQ}}$ pin low-level input
	1	0	0	Activated by SCI channel 0 transmit-data-empty interrupt
			1	Activated by SCI channel 0 receive-data-full interrupt
		1	0	Activated by SCI channel 1 transmit-data-empty interrupt
			1	Activated by SCI channel 1 receive-data-full interrupt
1	0	0	0	Activated by TPU channel 0 compare match/input capture A interrupt
			1	Activated by TPU channel 1 compare match/input capture A interrupt
		1	0	Activated by TPU channel 2 compare match/input capture A interrupt
			1	Activated by TPU channel 3 compare match/input capture A interrupt
	1	0	0	Activated by TPU channel 4 compare match/input capture A interrupt
			1	Activated by TPU channel 5 compare match/input capture A interrupt
		1	0	—
			1	—

Note: * Detected as a low level in the first transfer after transfer is enabled.

The same factor can be selected for more than one channel. In this case, activation starts with the highest-priority channel according to the relative channel priorities. For relative channel priorities, see section 7.5.13, DMAC Multi-Channel Operation.

DMABCRH:

Bit	:	15	14	13	12	11	10	9	8
		FAE1	FAE0	—	—	DTA1	—	DTA0	—
Initial value	:	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

DMABCRL:

Bit	:	7	6	5	4	3	2	1	0
	:	DTME1	DTE1	DTME0	DTE0	DTIE1B	DTIE1A	DTIE0B	DTIE0A
Initial value	:	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

DMABCR is a 16-bit readable/writable register that controls the operation of each DMAC channel.

DMABCR is initialized to H'0000 by a reset, and in hardware standby mode.

Bit 15—Full Address Enable 1 (FAE1): Specifies whether channel 1 is to be used in short address mode or full address mode.

In full address mode, channels 1A and 1B are used together as a single channel.

Bit 15 FAE1	**Description**	
0	Short address mode	(Initial value)
1	Full address mode	

In full address mode, channels 0A and 0B are used together as a single channel.

Bit 14 FAE0	Description	
0	Short address mode	(Initial value)
1	Full address mode	

Bits 13 and 12—Reserved: Can be read or written to. Only 0 should be written to these bits.

Bits 11 and 9—Data Transfer Acknowledge (DTA): These bits enable or disable clearing, when DMA transfer is performed, of the internal interrupt source selected by the data transfer factor setting.

When DTE = 1 and DTA = 1, the internal interrupt source selected by the data transfer factor setting is cleared automatically by DMA transfer. When DTE = 1 and DTA = 1, the internal interrupt source selected by the data transfer factor setting does not issue an interrupt request to the CPU or DTC.

When DTE = 1 and DTA = 0, the internal interrupt source selected by the data transfer factor setting is not cleared when a transfer is performed, and can issue an interrupt request to the CPU or DTC in parallel. In this case, the interrupt source should be cleared by the CPU or DTC transfer.

When DTE = 0, the internal interrupt source selected by the data transfer factor setting issues an interrupt request to the CPU or DTC regardless of the DTA bit setting.

The state of the DTME bit does not affect the above operations.

Bit 11—Data Transfer Acknowledge 1 (DTA1): Enables or disables clearing, when DMA transfer is performed, of the internal interrupt source selected by the channel 1 data transfer factor setting.

Bit 11 DTA1	Description	
0	Clearing of selected internal interrupt source at time of DMA transfer is disabled	(Initial value)
1	Clearing of selected internal interrupt source at time of DMA transfer is enabled	

setting.

Bit 9 DTA0	Description
0	Clearing of selected internal interrupt source at time of DMA transfer is disabled (Initial value)
1	Clearing of selected internal interrupt source at time of DMA transfer is enabled

Bits 10 and 8—Reserved: Can be read or written to. Only 0 should be written to these bits.

Bits 7 and 5—Data Transfer Master Enable (DTME): Together with the DTE bit, these bits control enabling or disabling of data transfer on the relevant channel. When both the DTME bit and the DTE bit are set to 1, transfer is enabled for the channel.

If the relevant channel is in the middle of a burst mode transfer when an NMI interrupt is generated, the DTME bit is cleared, the transfer is interrupted, and bus mastership passes to the CPU. When the DTME bit is subsequently set to 1 again, the interrupted transfer is resumed. In block transfer mode, however, the DTME bit is not cleared by an NMI interrupt, and transfer is not interrupted.

The conditions for the DTME bit being cleared to 0 are as follows:

- When initialization is performed
- When NMI is input in burst mode
- When 0 is written to the DTME bit

The condition for DTME being set to 1 is as follows:

- When 1 is written to DTME after DTME is read as 0

Bit 7—Data Transfer Master Enable 1 (DTME1): Enables or disables data transfer on channel 1.

Bit 7 DTME1	Description
0	Data transfer disabled. In burst mode, cleared to 0 by an NMI interrupt (Initial value)
1	Data transfer enabled

Bit 5 DTME0	Description	
0	Data transfer disabled. In normal mode, cleared to 0 by an NMI interrupt	(Initial value)
1	Data transfer enabled	

Bits 6 and 4—Data Transfer Enable (DTE): When DTE = 0, data transfer is disabled and the activation source selected by the data transfer factor setting is ignored. If the activation source is an internal interrupt, an interrupt request is issued to the CPU or DTC. If the DTIE bit is set to 1 when DTE = 0, the DMAC regards this as indicating the end of a transfer, and issues a transfer end interrupt request to the CPU.

The conditions for the DTE bit being cleared to 0 are as follows:

- When initialization is performed
- When the specified number of transfers have been completed
- When 0 is written to the DTE bit to forcibly abort the transfer, or for a similar reason

When DTE = 1 and DTME = 1, data transfer is enabled and the DMAC waits for a request by the activation source selected by the data transfer factor setting. When a request is issued by the activation source, DMA transfer is executed.

The condition for the DTE bit being set to 1 is as follows:

- When 1 is written to the DTE bit after the DTE bit is read as 0

Bit 6—Data Transfer Enable 1 (DTE1): Enables or disables data transfer on channel 1.

Bit 6 DTE1	Description	
0	Data transfer disabled	(Initial value)
1	Data transfer enabled	

Bit 4 DTE0	Description	
0	Data transfer disabled	(Initial value)
1	Data transfer enabled	

Bits 3 and 1—Data Transfer Interrupt Enable B (DTIEB): These bits enable or disable an interrupt to the CPU or DTC when transfer is interrupted. If the DTIEB bit is set to 1 when DTME = 0, the DMAC regards this as indicating a break in the transfer, and issues a transfer break interrupt request to the CPU or DTC.

A transfer break interrupt can be canceled either by clearing the DTIEB bit to 0 in the interrupt handling routine, or by performing processing to continue transfer by setting the DTME bit to 1.

Bit 3—Data Transfer Interrupt Enable 1B (DTIE1B): Enables or disables the channel 1 transfer break interrupt.

Bit 3 DTIE1B	Description	
0	Transfer break interrupt disabled	(Initial value)
1	Transfer break interrupt enabled	

Bit 1—Data Transfer Interrupt Enable 0B (DTIE0B): Enables or disables the channel 0 transfer break interrupt.

Bit 1 DTIE0B	Description	
0	Transfer break interrupt disabled	(Initial value)
1	Transfer break interrupt enabled	

Bits 2 and 0—Data Transfer End Interrupt Enable A (DTIEA): These bits enable or disable an interrupt to the CPU or DTC when transfer ends. If the DTIEA bit is set to 1 when DTE = 0, the DMAC regards this as indicating the end of a transfer, and issues a transfer end interrupt request to the CPU or DTC.

A transfer end interrupt can be canceled either by clearing the DTIEA bit to 0 in the interrupt handling routine, or by performing processing to continue transfer by setting the transfer counter and address register again, and then setting the DTE bit to 1.

Bit 2 DTIE1A	Description	
0	Transfer end interrupt disabled	(Initial value)
1	Transfer end interrupt enabled	

Bit 0—Data Transfer Interrupt Enable 0A (DTIE0A): Enables or disables the channel 0 transfer end interrupt.

Bit 0 DTIE0A	Description	
0	Transfer end interrupt disabled	(Initial value)
1	Transfer end interrupt enabled	

7.4.1 DMA Write Enable Register (DMAWER)

The DMAC can activate the DTC with a transfer end interrupt, rewrite the channel on which the transfer ended using a DTC chain transfer, and reactivate the DTC. DMAWER applies restrictions so that specific bits of DMACR for the specific channel, and also DMATCR and DMABCR, can be changed to prevent inadvertent rewriting of registers other than those for the channel concerned. The restrictions applied by DMAWER are valid for the DTC.

Figure 7.2 shows the transfer areas for activating the DTC with a channel 0A transfer end interrupt, and reactivating channel 0A. The address register and count register area is re-set by the first DTC transfer, then the control register area is re-set by the second DTC chain transfer.

When re-setting the control register area, perform masking by setting bits in DMAWER to prevent modification of the contents of the other channels.

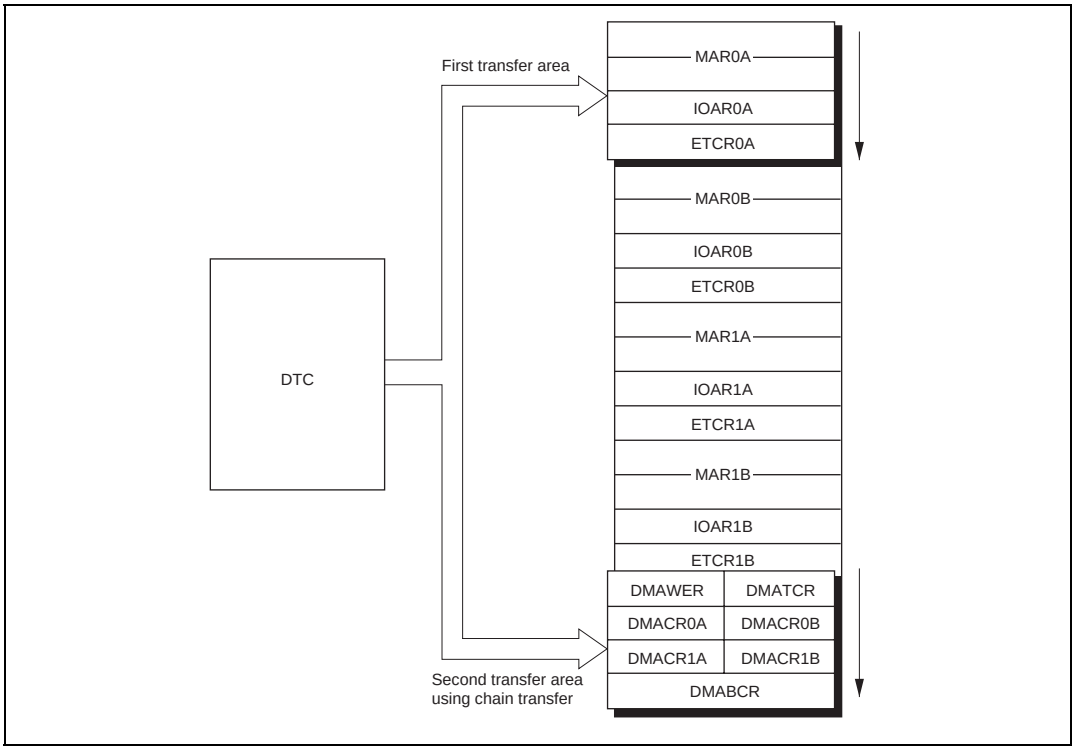


Figure 7.2 Areas for Register Re-Setting by DTC (Example: Channel 0A)

Initial value :	0	0	0	0	0	0	0	0
R/W :	—	—	—	—	R/W	R/W	R/W	R/W

DMAWER is an 8-bit readable/writable register that controls enabling or disabling of writes to DMACR, DMABCR, and DMATCR by the DTC.

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

Bits 7 to 4—Reserved: Read-only bits, always read as 0.

Bit 3—Write Enable 1B (WE1B): Enables or disables writes to all bits in DMACR1B, bits 11, 7, and 3 in DMABCR, and bit 5 in DMATCR, by the DTC.

Bit 3 WE1B	Description
0	Writes to all bits in DMACR1B, bits 11, 7, and 3 in DMABCR, and bit 5 in DMATCR are disabled (Initial value)
1	Writes to all bits in DMACR1B, bits 11, 7, and 3 in DMABCR, and bit 5 in DMATCR are enabled

Bit 2—Write Enable 1A (WE1A): Enables or disables writes to all bits in DMACR1A, and bits 10, 6, and 2 in DMABCR, by the DTC.

Bit 2 WE1A	Description
0	Writes to all bits in DMACR1A, and bits 10, 6, and 2 in DMABCR are disabled (Initial value)
1	Writes to all bits in DMACR1A, and bits 10, 6, and 2 in DMABCR are enabled

Bit 1 WE0B	Description
0	Writes to all bits in DMACR0B, bits 9, 5, and 1 in DMABCR, and bit 4 in DMATCR are disabled (Initial value)
1	Writes to all bits in DMACR0B, bits 9, 5, and 1 in DMABCR, and bit 4 in DMATCR are enabled

Bit 0—Write Enable 0A (WE0A): Enables or disables writes to all bits in DMACR0A, and bits 8, 4, and 0 in DMABCR, by the DTC.

Bit 0 WE0A	Description
0	Writes to all bits in DMACR0A, and bits 8, 4, and 0 in DMABCR are disabled (Initial value)
1	Writes to all bits in DMACR0A, and bits 8, 4, and 0 in DMABCR are enabled

Writes by the DTC to bits 15 to 12 (FAE and SAE) in DMABCR are invalid regardless of the DMAWER settings. These bits should be changed, if necessary, by CPU processing.

In writes by the DTC to bits 7 to 4 (DTE) in DMABCR, 1 can be written without first reading 0. To reactivate a channel set to full address mode, write 1 to both Write Enable A and Write Enable B for the channel to be reactivated.

MAR, IOAR, and ETCR are always write-enabled regardless of the DMAWER settings. When modifying these registers, the channel for which the modification is to be made should be halted.

Bit	:	7	6	5	4	3	2	1	0
		—	—	TEE1	TEE0	—	—	—	—
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	—	—	R/W	R/W	—	—	—	—

DMATCR is an 8-bit readable/writable register that controls enabling or disabling of DMAC transfer end pin output. A port can be set for output automatically, and a transfer end signal output, by setting the appropriate bit.

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

Bits 7 and 6—Reserved: Read-only bits, always read as 0.

Bit 5—Transfer End Enable 1 (TEE1): Enables or disables transfer end pin 1 ($\overline{\text{TEND1}}$) output.

Bit 5 TEE1	Description	
0	$\overline{\text{TEND1}}$ pin output disabled	(Initial value)
1	$\overline{\text{TEND1}}$ pin output enabled	

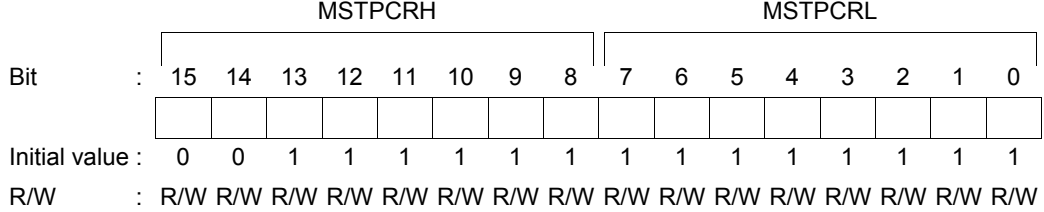
Bit 4—Transfer End Enable 0 (TEE0): Enables or disables transfer end pin 0 ($\overline{\text{TEND0}}$) output.

Bit 4 TEE0	Description	
0	$\overline{\text{TEND0}}$ pin output disabled	(Initial value)
1	$\overline{\text{TEND0}}$ pin output enabled	

The $\overline{\text{TEND}}$ pins are assigned only to channel B in short address mode.

The transfer end signal indicates the transfer cycle in which the transfer counter reached 0, regardless of the transfer source. An exception is block transfer mode, in which the transfer end signal indicates the transfer cycle in which the block counter reached 0.

Bits 3 to 0—Reserved: Read-only bits, always read as 0.



MSTPCR is a 16-bit readable/writable register that performs module stop mode control.

When the MSTP15 bit in MSTPCR is set to 1, the DMAC operation stops at the end of the bus cycle and a transition is made to module stop mode. For details, see section 21.5, Module Stop Mode.

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

Bit 15—Module Stop (MSTP15): Specifies the DMAC module stop mode.

Bit 15 MSTP15	Description	
0	DMAC module stop mode cleared	(Initial value)
1	DMAC module stop mode set	

7.5.1 Transfer Modes

Table 7.5 lists the DMAC modes.

Table 7.5 DMAC Transfer Modes

Transfer Mode			Transfer Source	Remarks
Short address mode	Dual address mode	(1) Sequential mode (2) Idle mode (3) Repeat mode	• TPU channel 0 to 5 compare match/input capture A interrupt • SCI transmit-data-empty interrupt • SCI receive-data-full interrupt • A/D converter conversion end interrupt • External request	• Up to 4 channels can operate independently • External request applies to channel B only • Single address mode applies to channel B only • Modes (1), (2), and (3) can also be specified for single address mode
	(4) Single address mode			
Full address mode	(5) Normal mode		• External request • Auto-request	• Max. 2-channel operation, combining channels A and B
	(6) Block transfer mode		• TPU channel 0 to 5 compare match/input capture A interrupt • SCI transmit-data-empty interrupt • SCI receive-data-full interrupt • A/D converter conversion end interrupt • External request	• With auto-request, burst mode transfer or cycle steal transfer can be selected

Sequential Mode: In response to a single transfer request, the specified number of transfers are carried out, one byte or one word at a time. An interrupt request can be sent to the CPU or DTC when the specified number of transfers have been completed. One address is specified as 24 bits, and the other as 16 bits. The transfer direction is programmable.

Idle Mode: In response to a single transfer request, the specified number of transfers are carried out, one byte or one word at a time. An interrupt request can be sent to the CPU or DTC when the specified number of transfers have been completed. One address is specified as 24 bits, and the other as 16 bits. The transfer source address and transfer destination address are fixed. The transfer direction is programmable.

Repeat Mode: In response to a single transfer request, the specified number of transfers are carried out, one byte or one word at a time. When the specified number of transfers have been completed, the addresses and transfer counter are restored to their original settings, and operation is continued. No interrupt request is sent to the CPU or DTC. One address is specified as 24 bits, and the other as 16 bits. The transfer direction is programmable.

Single Address Mode: In response to a single transfer request, the specified number of transfers are carried out between external memory and an external device, one byte or one word at a time. Unlike dual address mode, source and destination accesses are performed in parallel. Therefore, either the source or the destination is an external device which can be accessed with a strobe alone, using the $\overline{\text{DACK}}$ pin. One address is specified as 24 bits, and for the other, the pin is set automatically. The transfer direction is programmable.

Sequential mode, idle mode, and repeat mode can also be specified for single address mode.

Normal Mode

- Auto-request

  By means of register settings only, the DMAC is activated, and transfer continues until the specified number of transfers have been completed. An interrupt request can be sent to the CPU or DTC when transfer is completed. Both addresses are specified as 24 bits.

  — Cycle steal mode

  The bus is released to another bus master after each byte or word transfer.

  — Burst mode

  The bus is held and transfer continued until the specified number of transfers have been completed.

byte or one word at a time. An interrupt request can be sent to the CPU or DTC when the specified number of transfers have been completed. Both addresses are specified as 24 bits.

Block Transfer Mode: In response to a single transfer request, a block transfer of the specified block size is carried out. This is repeated the specified number of times, once each time there is a transfer request. At the end of each single block transfer, one address is restored to its original setting. An interrupt request can be sent to the CPU or DTC when the specified number of block transfers have been completed. Both addresses are specified as 24 bits.

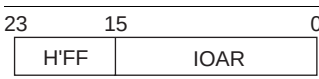
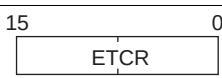
7.5.2 Sequential Mode

Sequential mode can be specified by clearing the RPE bit in DMACR to 0. In sequential mode, MAR is updated after each byte or word transfer in response to a single transfer request, and this is executed the number of times specified in ETCR.

One address is specified by MAR, and the other by IOAR. The transfer direction can be specified by the DTDIR bit in DMACR.

Table 7.6 summarizes register functions in sequential mode.

Table 7.6 Register Functions in Sequential Mode

Register	Function		Initial Setting	Operation
	DTDIR = 0	DTDIR = 1		
	Source address register	Destination address register	Start address of transfer destination or transfer source	Incremented/ decremented every transfer
	Destination address register	Source address register	Start address of transfer source or transfer destination	Fixed
	Transfer counter		Number of transfers	Decremented every transfer; transfer ends when count reaches H'0000

Legend:
MAR: Memory address register
IOAR: I/O address register
ETCR: Execute transfer count register
DTDIR: Data transfer direction bit

IOAR specifies the lower 16 bits of the other address. The 8 bits above IOAR have a value of H'FF.

Figure 7.3 illustrates operation in sequential mode.

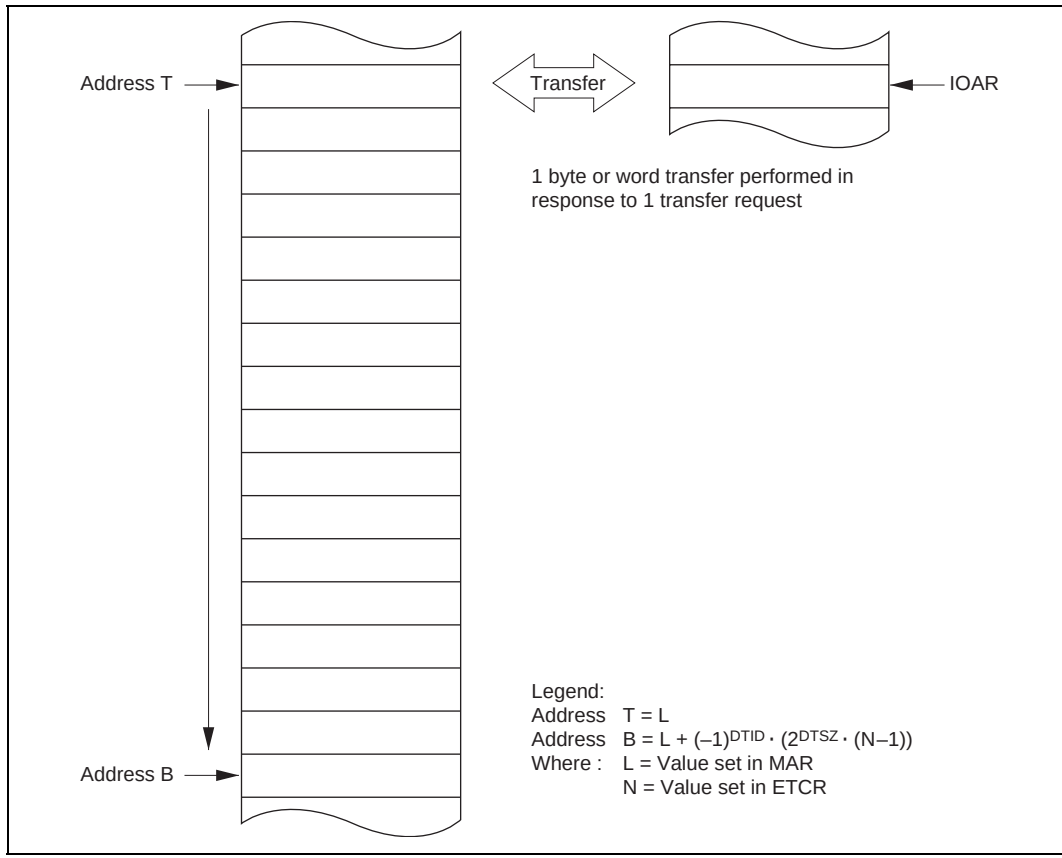


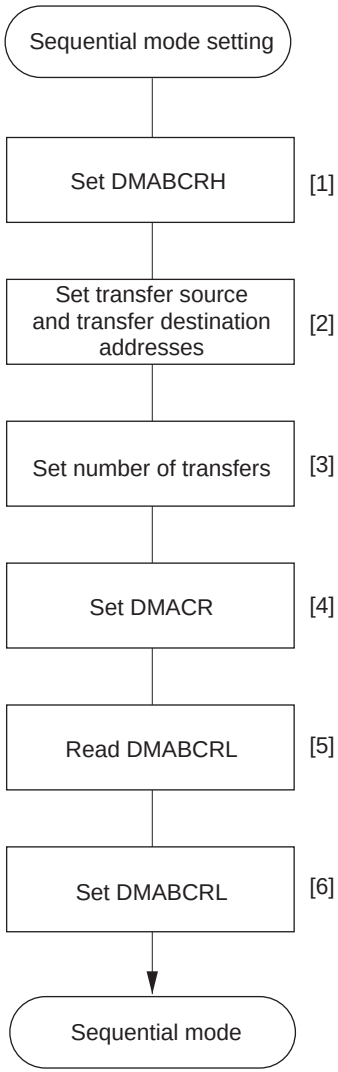
Figure 7.3 Operation in Sequential Mode

The number of transfers is specified as 16 bits in ETCR. ETCR is decremented by 1 each time a transfer is executed, and when its value reaches H'0000, the DTE bit is cleared and transfer ends. If the DTIE bit is set to 1 at this time, an interrupt request is sent to the CPU or DTC.

The maximum number of transfers, when H'0000 is set in ETCR, is 65,536.

match/input capture A interrupts. External requests can be set for channel B only.

Figure 7.4 shows an example of the setting procedure for sequential mode.



[1] Set each bit in DMABCRH.
- Clear the FAE bit to 0 to select short address mode.
- Specify enabling or disabling of internal interrupt clearing with the DTA bit.

[2] Set the transfer source address and transfer destination address in MAR and IOAR.

[3] Set the number of transfers in ETCR.

[4] Set each bit in DMACR.
- Set the transfer data size with the DTSZ bit.
- Specify whether MAR is to be incremented or decremented with the DTID bit.
- Clear the RPE bit to 0 to select sequential mode.
- Specify the transfer direction with the DTDIR bit.
- Select the activation source with bits DTF3 to DTF0.

[5] Read the DTE bit in DMABCRL as 0.

[6] Set each bit in DMABCRL.
- Specify enabling or disabling of transfer end interrupts with the DTIE bit.
- Set the DTE bit to 1 to enable transfer.

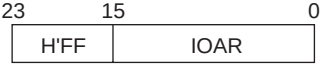
Figure 7.4 Example of Sequential Mode Setting Procedure

Idle mode can be specified by setting the RPE bit and DTIE bit in DMACR to 1. In idle mode, one byte or word is transferred in response to a single transfer request, and this is executed the number of times specified in ETCR.

One address is specified by MAR, and the other by IOAR. The transfer direction can be specified by the DTDIR bit in DMACR.

Table 7.7 summarizes register functions in idle mode.

Table 7.7 Register Functions in Idle Mode

Register	Function		Initial Setting	Operation
	DTDIR = 0	DTDIR = 1		
	Source address register	Destination address register	Start address of transfer destination or transfer source	Fixed
	Destination address register	Source address register	Start address of transfer source or transfer destination	Fixed
	Transfer counter		Number of transfers	Decremented every transfer; transfer ends when count reaches H'0000

Legend:
MAR: Memory address register
IOAR: I/O address register
ETCR: Execute transfer count register
DTDIR: Data transfer direction bit

MAR specifies the start address of the transfer source or transfer destination as 24 bits. MAR is neither incremented nor decremented each time a byte or word is transferred.

IOAR specifies the lower 16 bits of the other address. The 8 bits above IOAR have a value of H'FF.

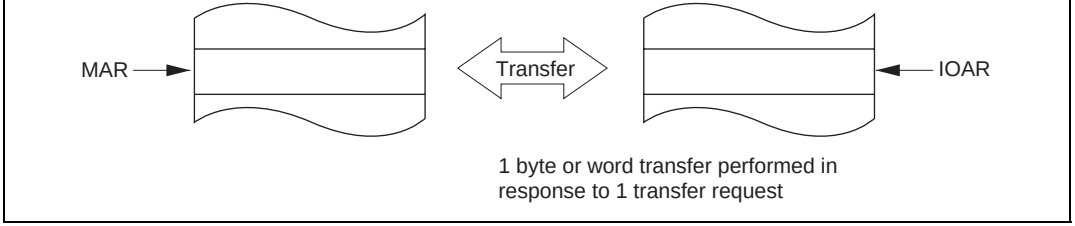


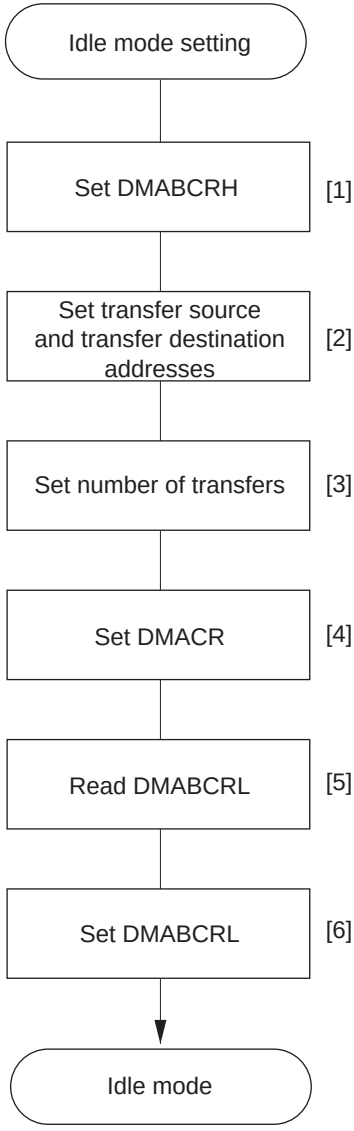
Figure 7.5 Operation in Idle Mode

The number of transfers is specified as 16 bits in ETCR. ETCR is decremented by 1 each time a transfer is executed, and when its value reaches H'0000, the DTE bit is cleared and transfer ends. If the DTIE bit is set to 1 at this time, an interrupt request is sent to the CPU or DTC.

The maximum number of transfers, when H'0000 is set in ETCR, is 65,536.

Transfer requests (activation sources) consist of A/D converter conversion end interrupts, external requests, SCI transmit-data-empty and receive-data-full interrupts, and TPU channel 0 to 5 compare match/input capture A interrupts. External requests can be set for channel B only.

When the DMAC is used in single address mode, only channel B can be set.



[1] Set each bit in DMABCRH.
- Clear the FAE bit to 0 to select short address mode.
- Specify enabling or disabling of internal interrupt clearing with the DTA bit.

[2] Set the transfer source address and transfer destination address in MAR and IOAR.

[3] Set the number of transfers in ETCR.

[4] Set each bit in DMACR.
- Set the transfer data size with the DTSZ bit.
- Specify whether MAR is to be incremented or decremented with the DTID bit.
- Set the RPE bit to 1.
- Specify the transfer direction with the DTDIR bit.
- Select the activation source with bits DTF3 to DTF0.

[5] Read the DTE bit in DMABCRL as 0.

[6] Set each bit in DMABCRL.
- Set the DTIE bit to 1.
- Set the DTE bit to 1 to enable transfer.

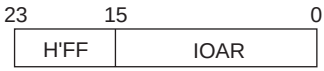
Figure 7.6 Example of Idle Mode Setting Procedure

Repeat mode can be specified by setting the RPE bit in DMACR to 1, and clearing the DTIE bit to 0. In repeat mode, MAR is updated after each byte or word transfer in response to a single transfer request, and this is executed the number of times specified in ETCRL. On completion of the specified number of transfers, MAR and ETCRL are automatically restored to their original settings and operation continues.

One address is specified by MAR, and the other by IOAR. The transfer direction can be specified by the DTDIR bit in DMACR.

Table 7.8 summarizes register functions in repeat mode.

Table 7.8 Register Functions in Repeat Mode

Register	Function		Initial Setting	Operation
	DTDIR = 0	DTDIR = 1		
	Source address register	Destination address register	Start address of transfer destination or transfer source	Incremented/ decremented every transfer. Initial setting is restored when value reaches H'0000
	Destination address register	Source address register	Start address of transfer source or transfer destination	Fixed
	Holds number of transfers		Number of transfers	Fixed
	Transfer counter		Number of transfers	Decremented every transfer. Loaded with ETCRH value when count reaches H'00

Legend:
MAR: Memory address register
IOAR: I/O address register
ETCR: Execute transfer count register
DTDIR: Data transfer direction bit

IOAR specifies the lower 16 bits of the other address. The 8 bits above IOAR have a value of H'FF.

The number of transfers is specified as 8 bits by ETCRH and ETCRL. The maximum number of transfers, when H'00 is set in both ETCRH and ETCRL, is 256.

In repeat mode, ETCRL functions as the transfer counter, and ETCRH is used to hold the number of transfers. ETCRL is decremented by 1 each time a transfer is executed, and when its value reaches H'00, it is loaded with the value in ETCRH. At the same time, the value set in MAR is restored in accordance with the values of the DTSZ and DTID bits in DMACR. The MAR restoration operation is as shown below.

$$MAR = MAR - (-1)^{DTID} \cdot 2^{DTSZ} \cdot ETCRH$$

The same value should be set in ETCRH and ETCRL.

In repeat mode, operation continues until the DTE bit is cleared. To end the transfer operation, therefore, the DTE bit should be cleared to 0. A transfer end interrupt request is not sent to the CPU or DTC.

By setting the DTE bit to 1 again after it has been cleared, the operation can be restarted from the transfer after that terminated when the DTE bit was cleared.

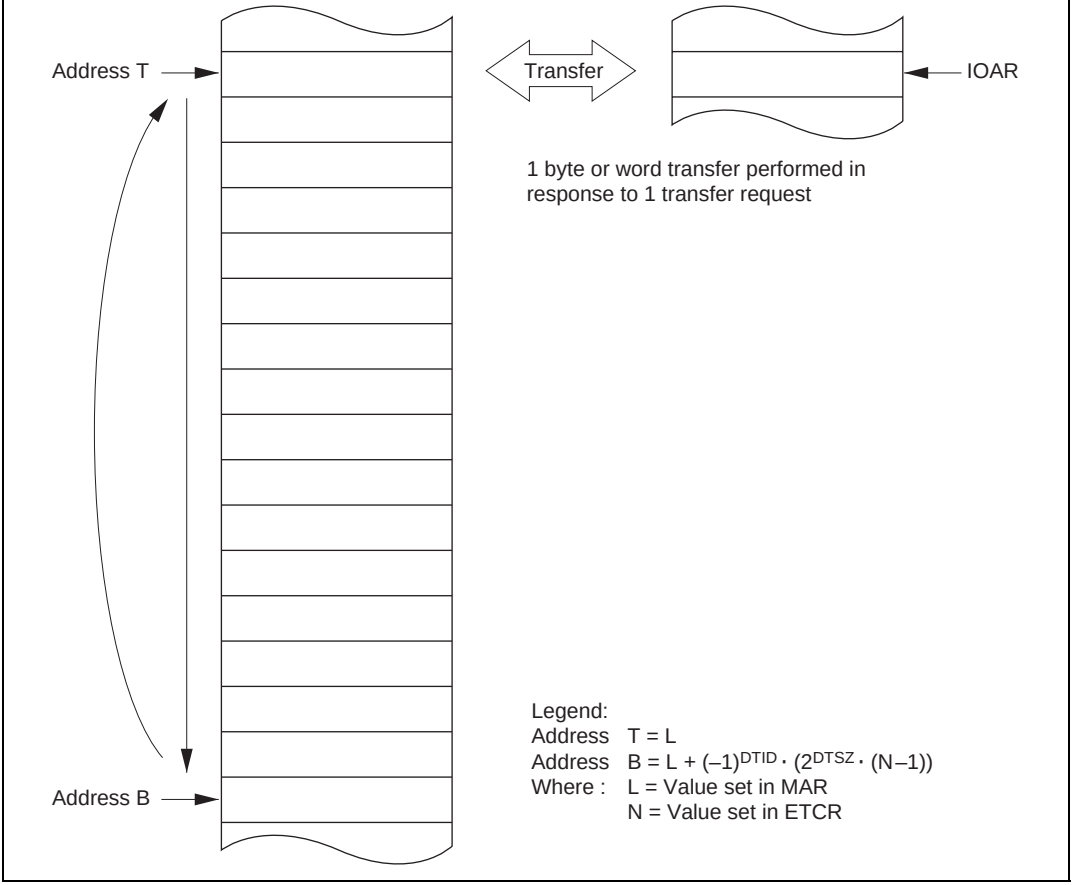
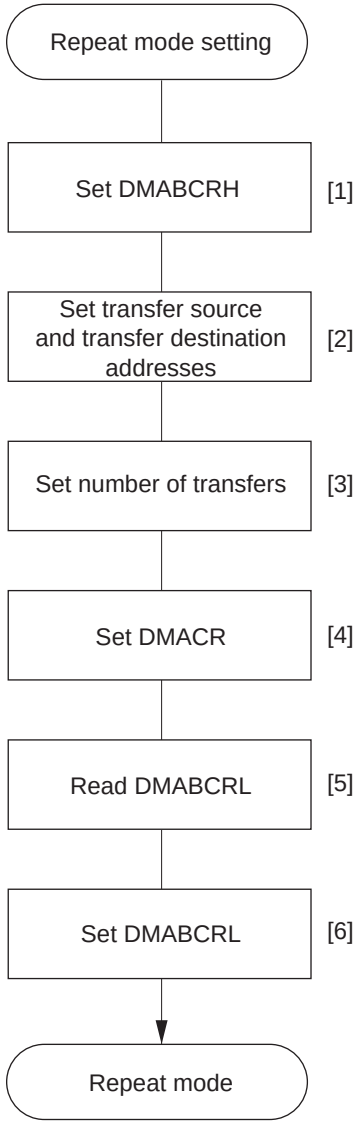


Figure 7.7 Operation in Repeat mode

Transfer requests (activation sources) consist of A/D converter conversion end interrupts, external requests, SCI transmit-data-empty and receive-data-full interrupts, and TPU channel 0 to 5 compare match/input capture A interrupts. External requests can be set for channel B only.



[1] Set each bit in DMABCRH.
- Clear the FAE bit to 0 to select short address mode.
- Specify enabling or disabling of internal interrupt clearing with the DTA bit.

[2] Set the transfer source address and transfer destination address in MAR and IOAR.

[3] Set the number of transfers in both ETCRH and ETCRL.

[4] Set each bit in DMACR.
- Set the transfer data size with the DTSZ bit.
- Specify whether MAR is to be incremented or decremented with the DTID bit.
- Set the RPE bit to 1.
- Specify the transfer direction with the DTDIR bit.
- Select the activation source with bits DTF3 to DTF0.

[5] Read the DTE bit in DMABCRL as 0.

[6] Set each bit in DMABCRL.
- Clear the DTIE bit to 0.
- Set the DTE bit to 1 to enable transfer.

Figure 7.8 Example of Repeat Mode Setting Procedure

Single address mode can only be specified for channel B. This mode can be specified by setting the SAE bit in DMABCR to 1 in short address mode.

One address is specified by MAR, and the other is set automatically to the data transfer acknowledge pin ($\overline{\text{DACK}}$). The transfer direction can be specified by the DTDIR bit in DMACR.

Table 7.9 summarizes register functions in single address mode.

Table 7.9 Register Functions in Single Address Mode

Register	Function		Initial Setting	Operation
	DTDIR = 0	DTDIR = 1		
23 MAR 0	Source address register	Destination address register	Start address of transfer destination or transfer source	*
$\overline{\text{DACK}}$ pin	Write strobe	Read strobe	(Set automatically by SAE bit; IOAR is invalid)	Strobe for external device
15 ETCR 0	Transfer counter		Number of transfers	*

Legend:
MAR: Memory address register
IOAR: I/O address register
ETCR: Execute transfer register
DTDIR: Data transfer direction bit
$\overline{\text{DACK}}$: Data transfer acknowledge

Note: * See the operation descriptions in sections 7.5.2, Sequential Mode, 7.5.3, Idle Mode, and 7.5.4, Repeat Mode.

MAR specifies the start address of the transfer source or transfer destination as 24 bits.

IOAR is invalid; in its place the strobe for external devices ($\overline{\text{DACK}}$) is output.

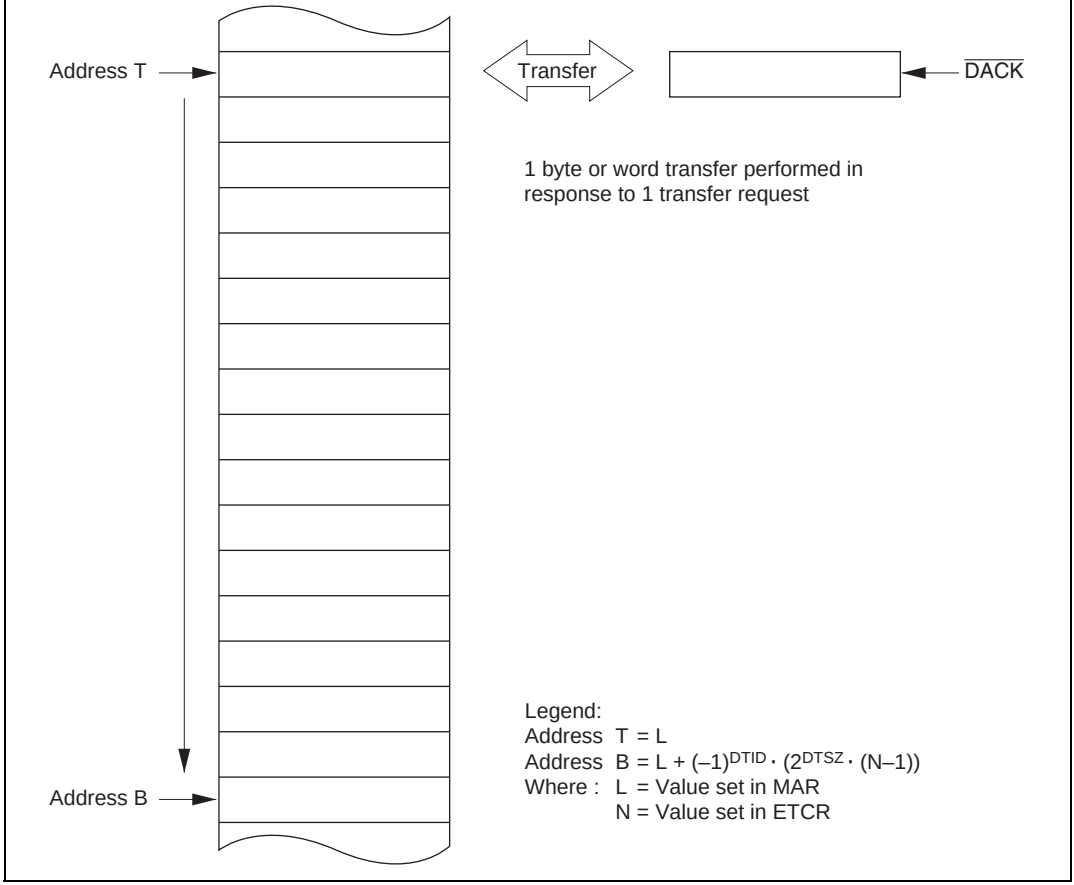
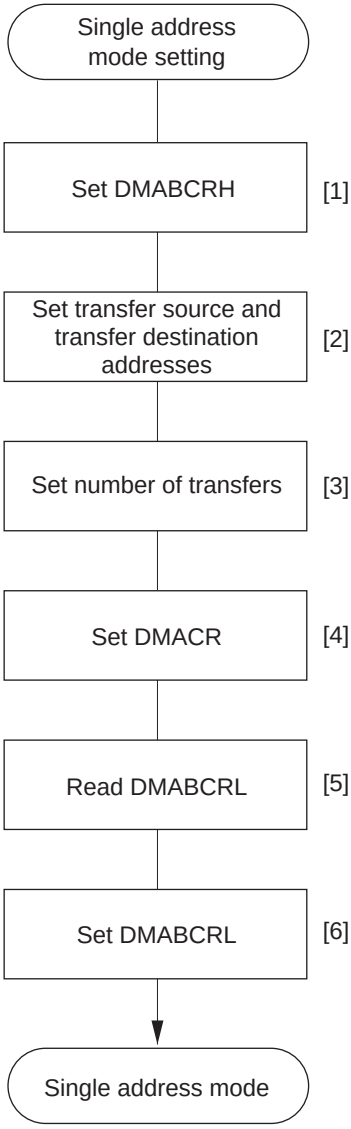


Figure 7.9 Operation in Single Address Mode (When Sequential Mode is Specified)



[1] Set each bit in DMABCRH.
- Clear the FAE bit to 0 to select short address mode.
- Set the SAE bit to 1 to select single address mode.
- Specify enabling or disabling of internal interrupt clearing with the DTA bit.

[2] Set the transfer source address/transfer destination address in MAR.

[3] Set the number of transfers in ETCR.

[4] Set each bit in DMACR.
- Set the transfer data size with the DTSZ bit.
- Specify whether MAR is to be incremented or decremented with the DTID bit.
- Clear the RPE bit to 0 to select sequential mode.
- Specify the transfer direction with the DTDIR bit.
- Select the activation source with bits DTF3 to DTF0.

[5] Read the DTE bit in DMABCRL as 0.

[6] Set each bit in DMABCRL.
- Specify enabling or disabling of transfer end interrupts with the DTIE bit.
- Set the DTE bit to 1 to enable transfer.

Figure 7.10 Example of Single Address Mode Setting Procedure (When Sequential Mode is Specified)

In normal mode, transfer is performed with channels A and B used in combination. Normal mode can be specified by setting the FAE bit in DMABCR to 1 and clearing the BLKE bit in DMACRA to 0.

In normal mode, MAR is updated after each byte or word transfer in response to a single transfer request, and this is executed the number of times specified in ETCRA. The transfer source is specified by MARA, and the transfer destination by MARB.

Table 7.10 summarizes register functions in normal mode.

Table 7.10 Register Functions in Normal Mode

Register	Function	Initial Setting	Operation
	Source address register	Start address of transfer source	Incremented/decremented every transfer, or fixed
	Destination address register	Start address of transfer destination	Incremented/decremented every transfer, or fixed
	Transfer counter	Number of transfers	Decremented every transfer; transfer ends when count reaches H'0000

Legend:
MARA: Memory address register A
MARB: Memory address register B
ETCRA: Execute transfer count register A

MARA and MARB specify the start addresses of the transfer source and transfer destination, respectively, as 24 bits. MAR can be incremented or decremented by 1 or 2 each time a byte or word is transferred, or can be fixed.

Incrementing, decrementing, or holding a fixed value can be set separately for MARA and MARB.

The number of transfers is specified by ETCRA as 16 bits. ETCRA is decremented each time a transfer is performed, and when its value reaches H'0000 the DTE bit is cleared and transfer ends. If the DTIE bit is set to 1 at this time, an interrupt request is sent to the CPU or DTC.

The maximum number of transfers, when H'0000 is set in ETCRA, is 65,536.

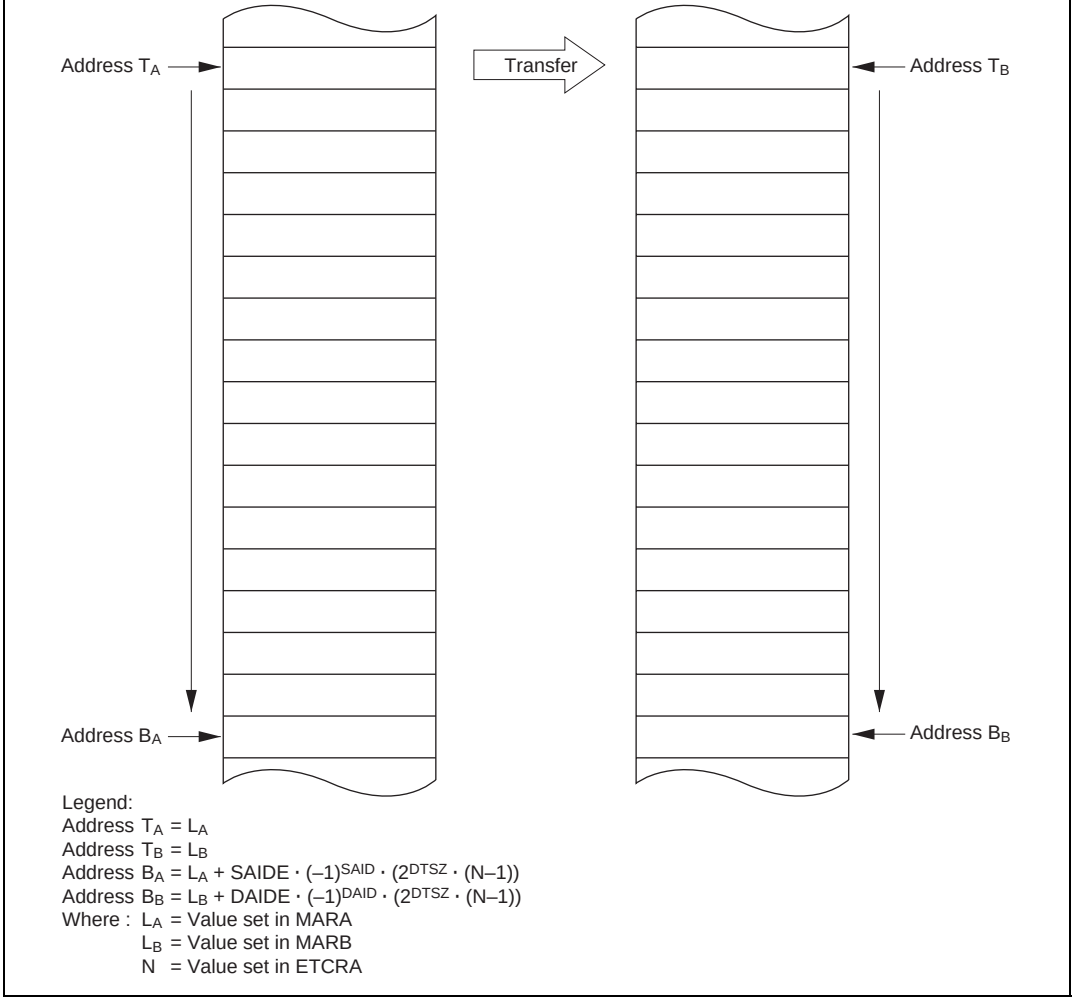
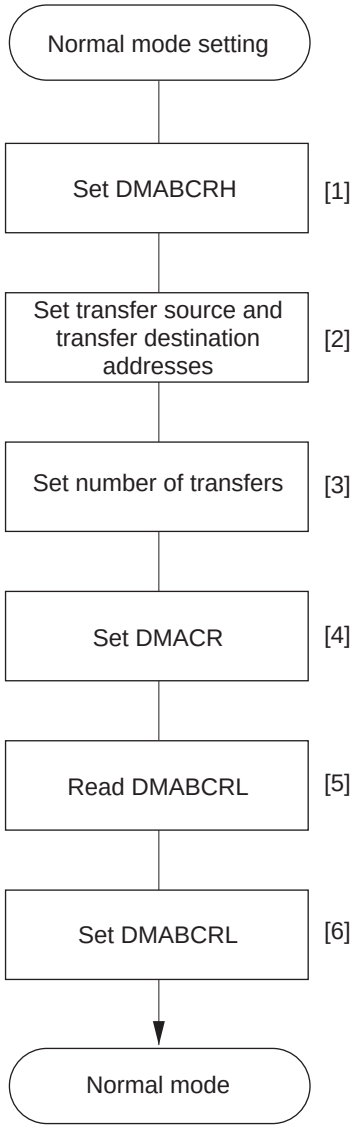


Figure 7.11 Operation in Normal Mode

Transfer requests (activation sources) are external requests and auto-requests.

With auto-request, the DMAC is only activated by register setting, and the specified number of transfers are performed automatically. With auto-request, cycle steal mode or burst mode can be selected. In cycle steal mode, the bus is released to another bus master each time a transfer is performed. In burst mode, the bus is held continuously until transfer ends.

Figure 7.12 shows an example of the setting procedure for normal mode.



[1] Set each bit in DMABCRH.
- Set the FAE bit to 1 to select full address mode.
- Specify enabling or disabling of internal interrupt clearing with the DTA bit.

[2] Set the transfer source address in MARA, and the transfer destination address in MARB.

[3] Set the number of transfers in ETCRA.

[4] Set each bit in DMACRA and DMACRB.
- Set the transfer data size with the DTSZ bit.
- Specify whether MARA is to be incremented, decremented, or fixed, with the SAID and SAIDE bits.
- Clear the BLKE bit to 0 to select normal mode.
- Specify whether MARB is to be incremented, decremented, or fixed, with the DAID and DAIDE bits.
- Select the activation source with bits DTF3 to DTF0.

[5] Read DTE = 0 and DTME = 0 in DMABCRL.

[6] Set each bit in DMABCRL.
- Specify enabling or disabling of transfer end interrupts with the DTIE bit.
- Set both the DTME bit and the DTE bit to 1 to enable transfer.

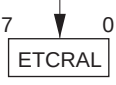
Figure 7.12 Example of Normal Mode Setting Procedure

In block transfer mode, transfer is performed with channels A and B used in combination. Block transfer mode can be specified by setting the FAE bit in DMABCR and the BLKE bit in DMACRA to 1.

In block transfer mode, a transfer of the specified block size is carried out in response to a single transfer request, and this is executed the specified number of times. The transfer source is specified by MARA, and the transfer destination by MARB. Either the transfer source or the transfer destination can be selected as a block area (an area composed of a number of bytes or words).

Table 7.11 summarizes register functions in block transfer mode.

Table 7.11 Register Functions in Block Transfer Mode

Register	Function	Initial Setting	Operation
	Source address register	Start address of transfer source	Incremented/decremented every transfer, or fixed
	Destination address register	Start address of transfer destination	Incremented/decremented every transfer, or fixed
	Holds block size	Block size	Fixed
	Block size counter	Block size	Decremented every transfer; ETCRH value copied when count reaches H'00
	Block transfer counter	Number of block transfers	Decremented every block transfer; transfer ends when count reaches H'0000

Legend:

MARA: Memory address register A
MARB: Memory address register B
ETCRA: Execute transfer count register A
ETCRB: Execute transfer count register B

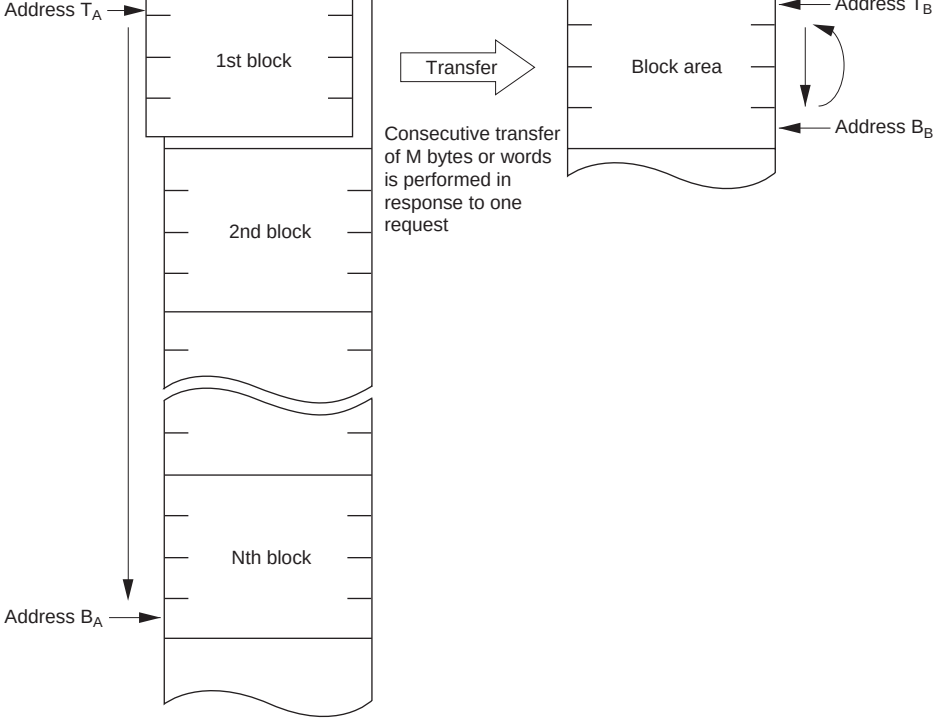
word is transferred, or can be fixed.

Incrementing, decrementing, or holding a fixed value can be set separately for MARA and MARB.

Whether a block is to be designated for MARA or for MARB is specified by the BLKDIR bit in DMACRA.

To specify the number of transfers, if M is the size of one block (where M = 1 to 256) and N transfers are to be performed (where N = 1 to 65,536), M is set in both ETCRAH and ETCRAL, and N in ETCRB.

Figure 7.13 illustrates operation in block transfer mode when MARB is designated as a block area.



Legend:
Address $T_A = L_A$
Address $T_B = L_B$
Address $B_A = L_A + SAIDE \cdot (-1)^{SAID} \cdot (2^{DTSZ} \cdot (M \cdot N - 1))$
Address $B_B = L_B + DAIDE \cdot (-1)^{DAID} \cdot (2^{DTSZ} \cdot (N - 1))$
Where : L_A = Value set in MARA
L_B = Value set in MARB
N = Value set in ETCRB
M = Value set in ETCRAH and ETCRAL

Figure 7.13 Operation in Block Transfer Mode (BLKDIR = 0)

Figure 7.14 illustrates operation in block transfer mode when MARA is designated as a block area.

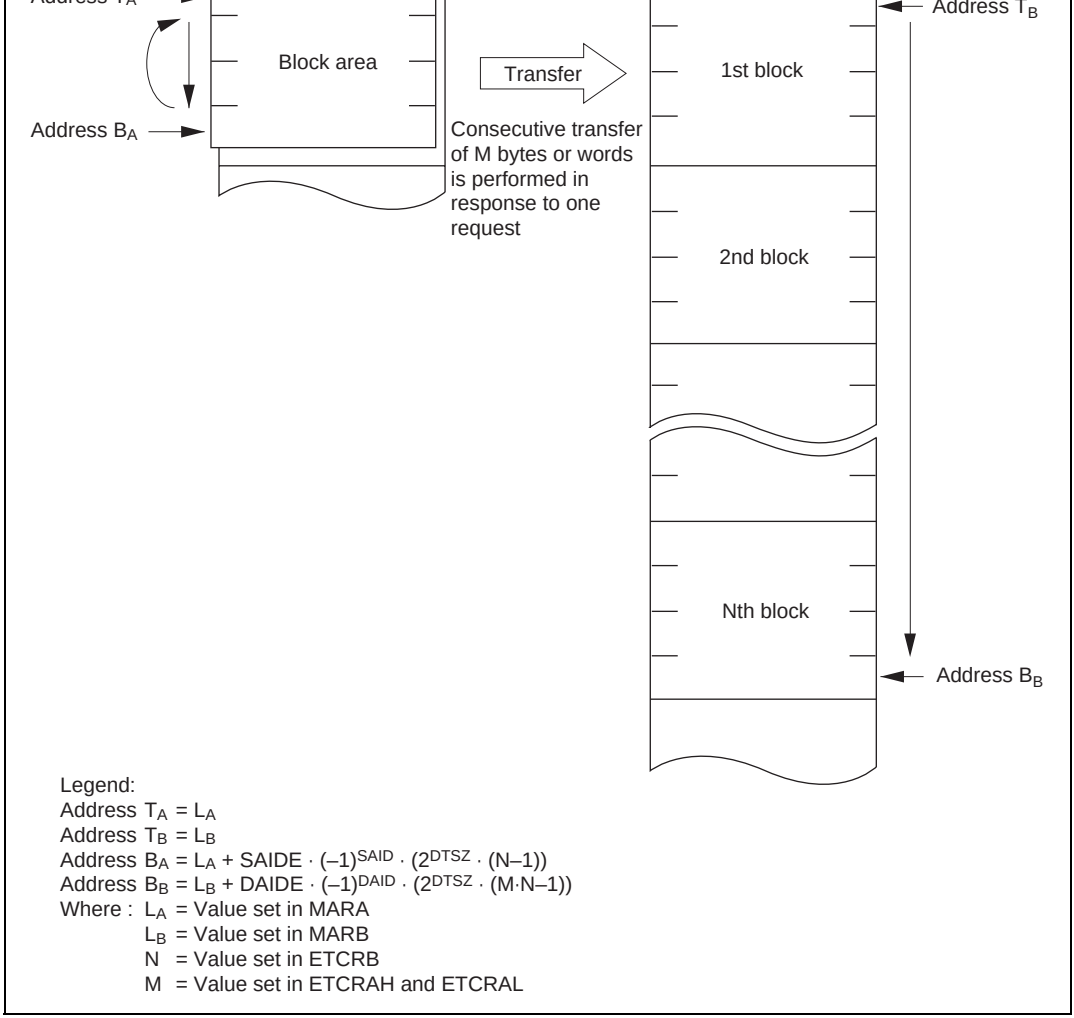


Figure 7.14 Operation in Block Transfer Mode (BLKDIR = 1)

ETCRAL is decremented by 1 each time a byte or word transfer is performed. In response to a single transfer request, burst transfer is performed until the value in ETCRAL reaches H'00. ETCRAL is then loaded with the value in ETCRAH. At this time, the value in the MAR register for which a block designation has been given by the BLKDIR bit in DMACRA is restored in accordance with the DTSZ, SAID/DAID, and SAIDE/DAIDE bits in DMACR.

sent to the CPU or DTC.

Figure 7.15 shows the operation flow in block transfer mode.

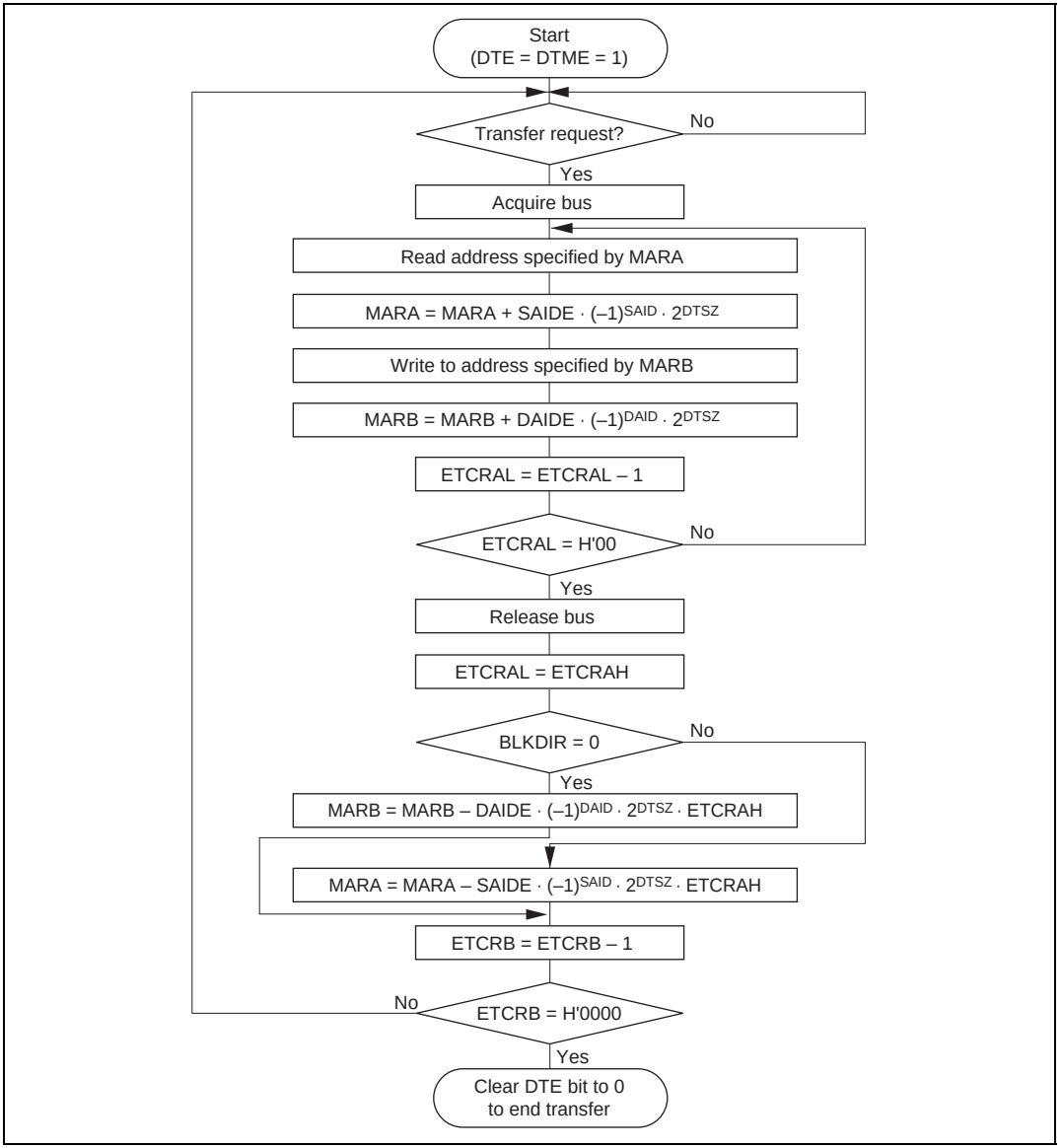


Figure 7.15 Operation Flow in Block Transfer Mode

compare match/input capture A interrupts.

For details, see section 7.3.4, DMA Control Register (DMACR).

Figure 7.16 shows an example of the setting procedure for block transfer mode.

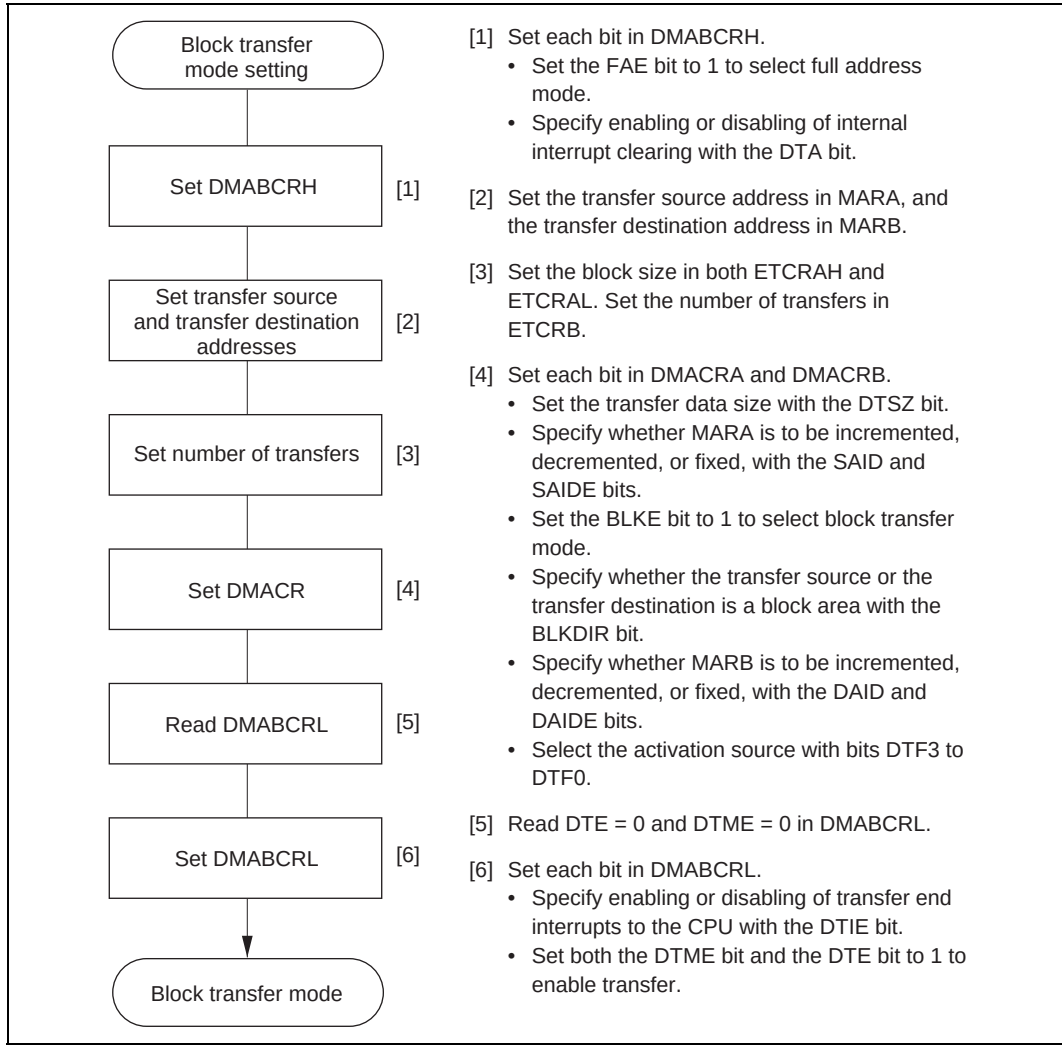


Figure 7.16 Example of Block Transfer Mode Setting Procedure

DMAC activation sources consist of internal interrupts, external requests, and auto-requests. The activation sources that can be specified depend on the transfer mode and the channel, as shown in table 7.12.

Table 7.12 DMAC Activation Sources

		Short Address Mode		Full Address Mode	
Activation Source		**Channels 0A and 1A**	**Channels 0B and 1B**	**Normal Mode**	**Block Transfer Mode**
Internal Interrupts	ADI	○	○	X	○
	TXI0	○	○	X	○
	RXI0	○	○	X	○
	TXI1	○	○	X	○
	RXI1	○	○	X	○
	TGI0A	○	○	X	○
	TGI1A	○	○	X	○
	TGI2A	○	○	X	○
	TGI3A	○	○	X	○
	TGI4A	○	○	X	○
	TGI5A	○	○	X	○
External Requests	$\overline{\text{DREQ}}$ pin falling edge input	X	○	○	○
	$\overline{\text{DREQ}}$ pin low-level input	X	○	○	○
Auto-request		X	X	○	X

Legend:

○ : Can be specified

X : Cannot be specified

Activation by Internal Interrupt: An interrupt request selected as a DMAC activation source can be sent simultaneously to the CPU and DTC. For details, see section 5, Interrupt Controller.

With activation by an internal interrupt, the DMAC accepts the request independently of the interrupt controller. Consequently, interrupt controller priority settings are irrelevant.

If the DMAC is activated by a CPU interrupt source or an interrupt source that is not used as a DTC activation source (DTA = 1), the interrupt source flag is cleared automatically by the DMA transfer. With ADI, TXI, and RXI interrupts, however, the interrupt source flag is not cleared

priority channel is activated first. Transfer requests for other channels are held pending in the DMAC, and activation is carried out in order of priority.

When DTE = 0, such as after completion of a transfer, a request from the selected activation source is not sent to the DMAC, regardless of the DTA bit. In this case, the relevant interrupt request is sent to the CPU or DTC.

In case of overlap with a CPU interrupt source or DTC activation source (DTA = 0), the interrupt request flag is not cleared by the DMAC.

Activation by External Request: If an external request ($\overline{\text{DREQ}}$ pin) is specified as an activation source, the relevant port should be set to input mode in advance.

Level sensing or edge sensing can be used for external requests.

External request operation in normal mode (short address mode or full address mode) is described below.

When edge sensing is selected, a 1-byte or 1-word transfer is executed each time a high-to-low transition is detected on the $\overline{\text{DREQ}}$ pin. The next transfer may not be performed if the next edge is input before transfer is completed.

When level sensing is selected, the DMAC stands by for a transfer request while the $\overline{\text{DREQ}}$ pin is held high. While the $\overline{\text{DREQ}}$ pin is held low, transfers continue in succession, with the bus being released each time a byte or word is transferred. If the $\overline{\text{DREQ}}$ pin goes high in the middle of a transfer, the transfer is interrupted and the DMAC stands by for a transfer request.

Activation by Auto-Request: Auto-request activation is performed by register setting only, and transfer continues to the end.

With auto-request activation, cycle steal mode or burst mode can be selected.

In cycle steal mode, the DMAC releases the bus to another bus master each time a byte or word is transferred. DMA and CPU cycles usually alternate.

In burst mode, the DMAC keeps possession of the bus until the end of the transfer, and transfer is performed continuously.

Single Address Mode: The DMAC can operate in dual address mode in which read cycles and write cycles are separate cycles, or single address mode in which read and write cycles are executed in parallel.

In single address mode, on the other hand, transfer is performed between external space in which either the transfer source or the transfer destination is specified by an address, and an external device for which selection is performed by means of the $\overline{\text{DACK}}$ strobe, without regard to the address. Figure 7.16 shows the data bus in single address mode.

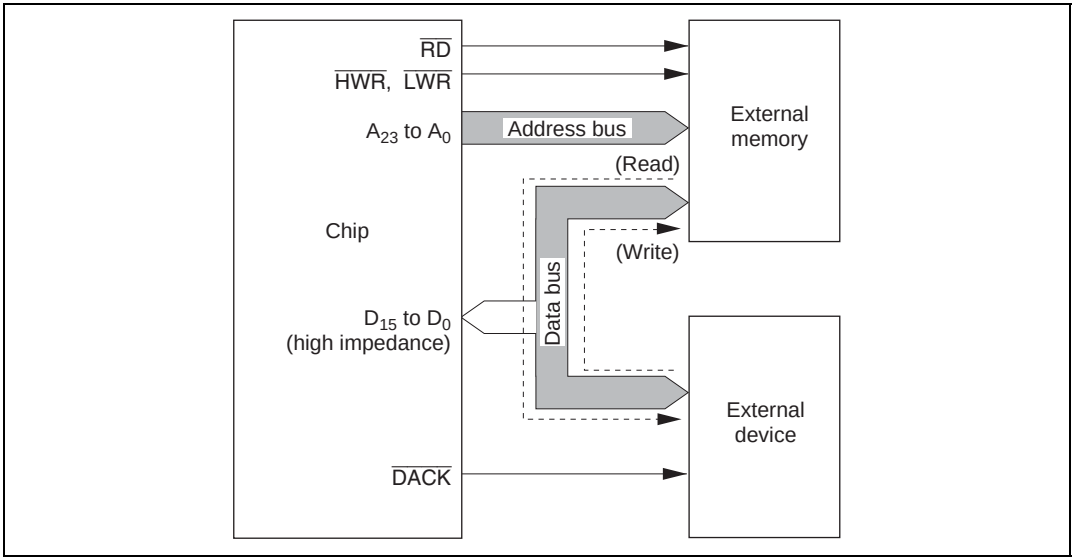


Figure 7.17 Data Bus in Single Address Mode

When using the DMAC for single address mode reading, transfer is performed from external memory to the external device, and the $\overline{\text{DACK}}$ pin functions as a write strobe for the external device. When using the DMAC for single address mode writing, transfer is performed from the external device to external memory, and the $\overline{\text{DACK}}$ pin functions as a read strobe for the external device. Since there is no directional control for the external device, one or other of the above single directions should be used.

Bus cycles in single address mode are in accordance with the settings of the bus controller for the external memory area. On the external device side, $\overline{\text{DACK}}$ is output in synchronization with the address strobe. For details of bus cycles, see section 7.5.11, DMAC Bus Cycles (Single Address Mode).

Do not specify internal space for transfer addresses in single address mode.

An example of the basic DMAC bus cycle timing is shown in figure 7.18. In this example, word-size transfer is performed from 16-bit, 2-state access space to 8-bit, 3-state access space. When the bus is transferred from the CPU to the DMAC, a source address read and destination address write are performed. The bus is not released in response to another bus request, etc., between these read and write operations. As with CPU cycles, DMA cycles conform to the bus controller settings.

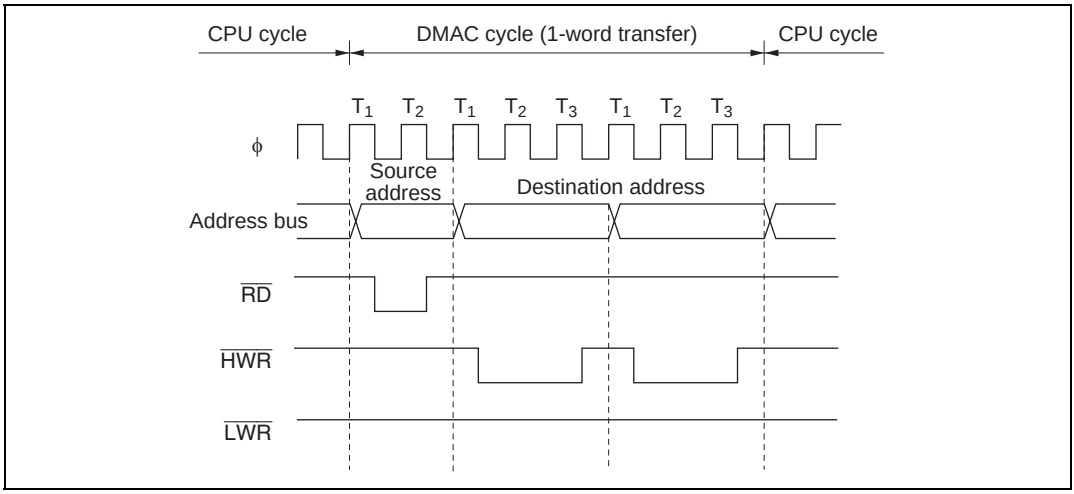


Figure 7.18 Example of DMA Transfer Bus Timing

The address is not output to the external address bus in an access to on-chip memory or an internal I/O register.

Short Address Mode: Figure 7.19 shows a transfer example in which TEND output is enabled and byte-size short address mode transfer (sequential/idle/repeat mode) is performed from external 8-bit, 2-state access space to internal I/O space.

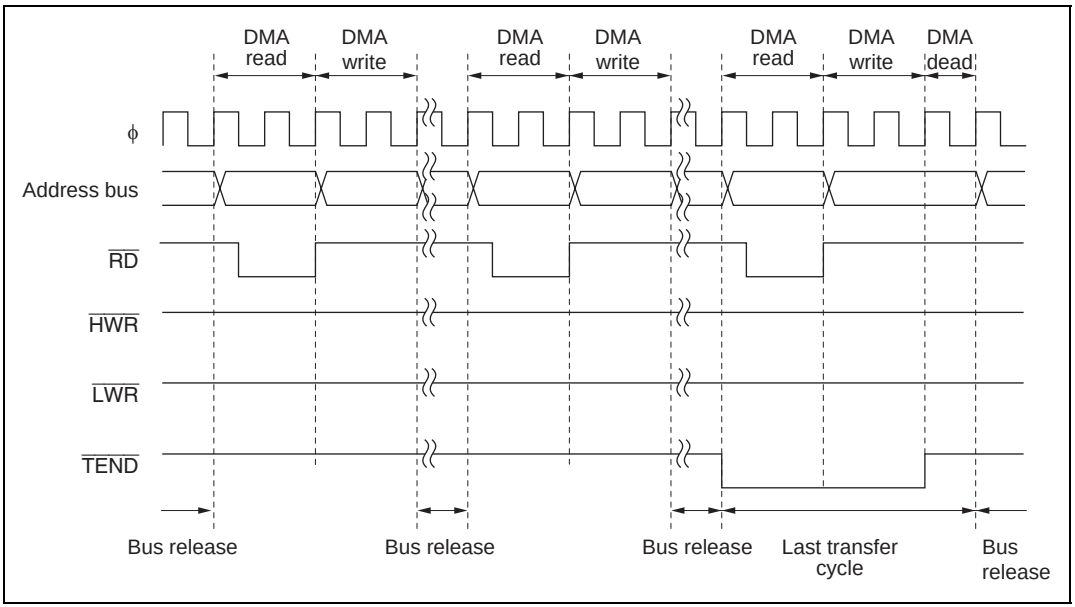


Figure 7.19 Example of Short Address Mode Transfer

A one-byte or one-word transfer is performed for one transfer request, and after the transfer the bus is released. While the bus is released one or more bus cycles are executed by the CPU or DTC.

In the transfer end cycle (the cycle in which the transfer counter reaches 0), a one-state DMA dead cycle is inserted after the DMA write cycle.

In repeat mode, when $\overline{\text{TEND}}$ output is enabled, $\overline{\text{TEND}}$ output goes low in the transfer cycle in which the transfer counter reaches 0.

external 16-bit, 2-state access space to external 16-bit, 2-state access space.

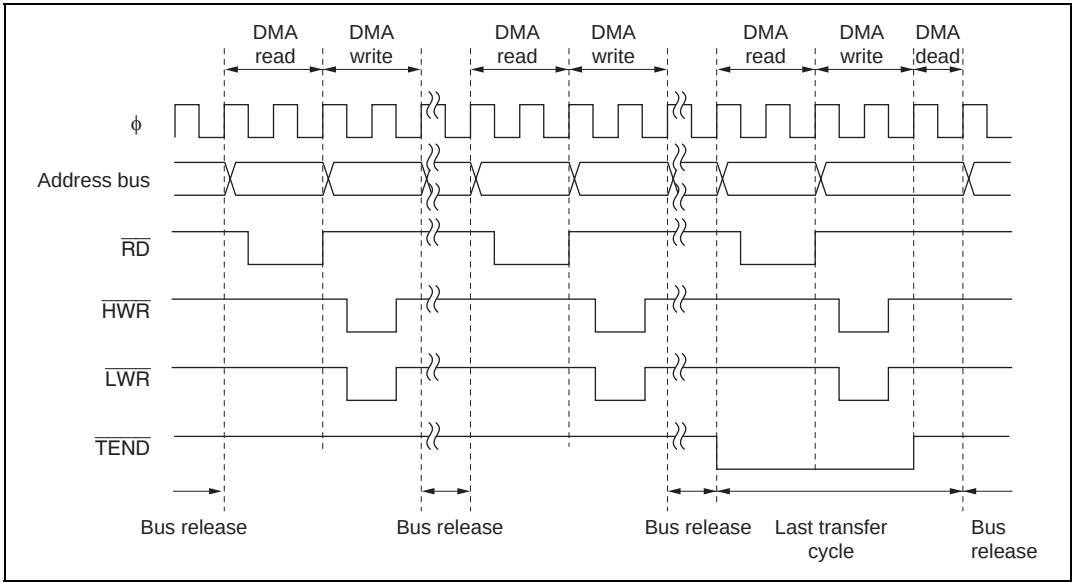


Figure 7.20 Example of Full Address Mode (Cycle Steal) Transfer

A one-byte or one-word transfer is performed, and after the transfer the bus is released. While the bus is released one bus cycle is executed by the CPU or DTC.

In the transfer end cycle (the cycle in which the transfer counter reaches 0), a one-state DMA dead cycle is inserted after the DMA write cycle.

bit, 2-state access space to external 16-bit, 2-state access space.

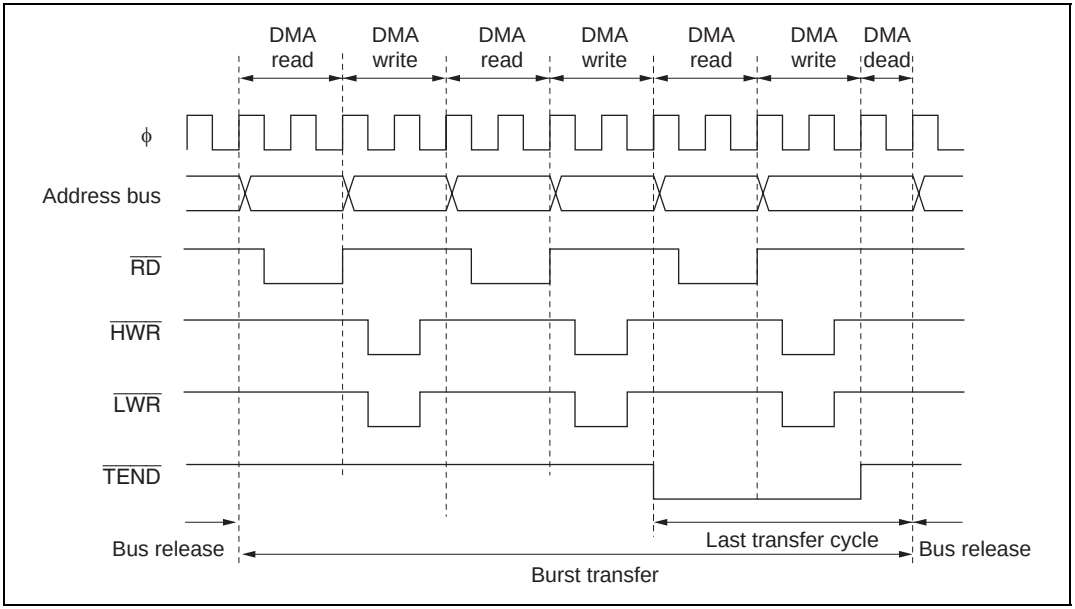


Figure 7.21 Example of Full Address Mode (Burst Mode) Transfer

In burst mode, one-byte or one-word transfers are executed consecutively until transfer ends.

In the transfer end cycle (the cycle in which the transfer counter reaches 0), a one-state DMA dead cycle is inserted after the DMA write cycle.

If a request from another higher-priority channel is generated after burst transfer starts, that channel has to wait until the burst transfer ends.

If an NMI is generated while a channel designated for burst transfer is in the transfer enabled state, the DTME bit is cleared and the channel is placed in the transfer disabled state. If burst transfer has already been activated inside the DMAC, the bus is released on completion of a one-byte or one-word transfer within the burst transfer, and burst transfer is suspended. If the last transfer cycle of the burst transfer has already been activated inside the DMAC, execution continues to the end of the transfer even if the DTME bit is cleared.

performed from internal 16-bit, 1-state access space to external 16-bit, 2-state access space.

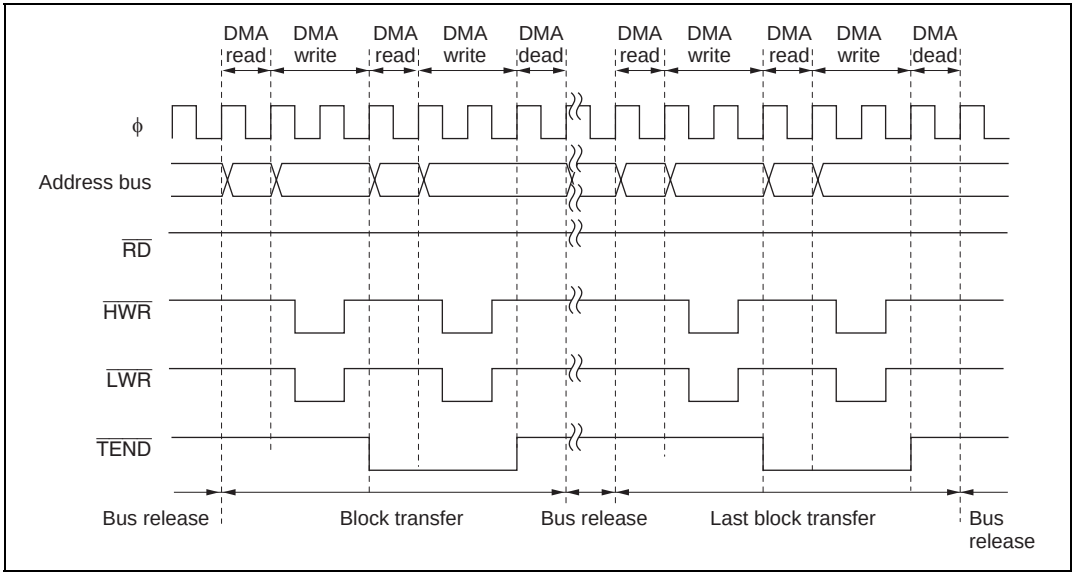


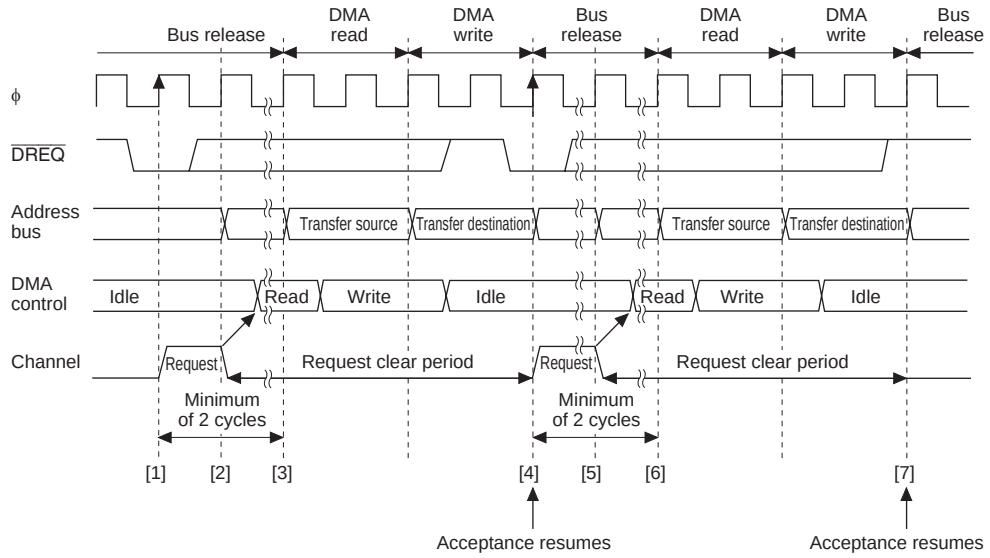
Figure 7.22 Example of Full Address Mode (Block Transfer Mode) Transfer

A one-block transfer is performed for one transfer request, and after the transfer the bus is released. While the bus is released, one or more bus cycles are executed by the CPU or DTC.

In the transfer end cycle of each block (the cycle in which the transfer counter reaches 0), a one-state DMA dead cycle is inserted after the DMA write cycle.

One block is transmitted without interruption. NMI generation does not affect block transfer operation.

Figure 7.23 shows an example of $\overline{\text{DREQ}}$ pin falling edge activated normal mode transfer.



[1] Acceptance after transfer enabling; the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of φ, and the request is held.

[2] [5] The request is cleared at the next bus break, and activation is started in the DMAC.

[3] [6] Start of DMA cycle; $\overline{\text{DREQ}}$ pin high level sampling on the rising edge of φ starts.

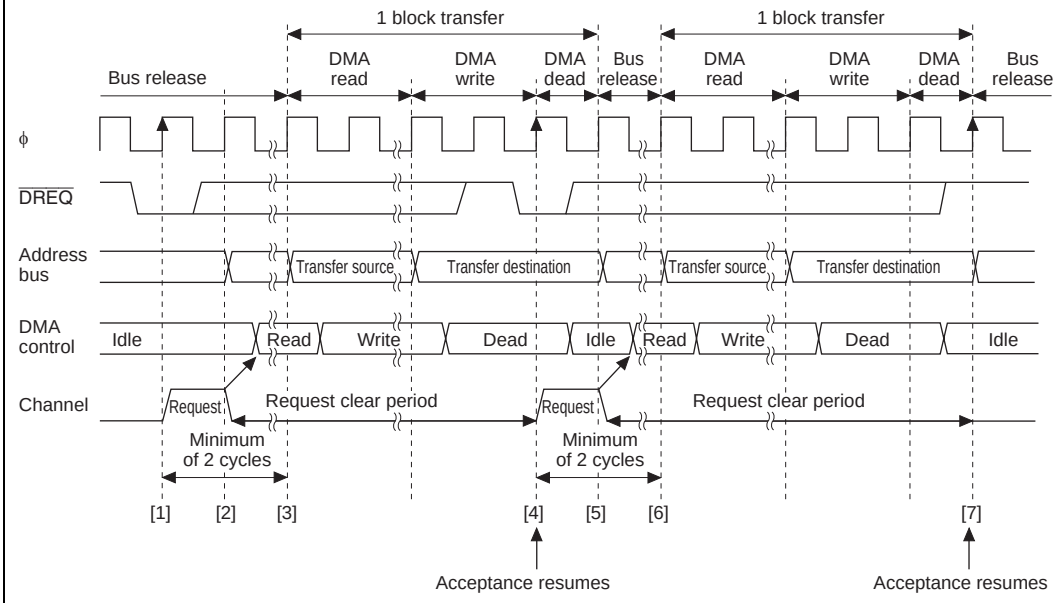
[4] [7] When the $\overline{\text{DREQ}}$ pin high level has been sampled, acceptance is resumed after the write cycle is completed.
(As in [1], the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of φ, and the request is held.)

Note: In write data buffer mode, bus breaks from [2] to [7] may be hidden, and not visible.

Figure 7.23 Example of $\overline{\text{DREQ}}$ Pin Falling Edge Activated Normal Mode Transfer

$\overline{\text{DREQ}}$ pin sampling is performed every cycle, with the rising edge of the next φ cycle after the end of the DMABCR write cycle for setting the transfer enabled state as the starting point.

When the $\overline{\text{DREQ}}$ pin low level is sampled while acceptance by means of the $\overline{\text{DREQ}}$ pin is possible, the request is held in the DMAC. Then, when activation is initiated in the DMAC, the request is cleared, and $\overline{\text{DREQ}}$ pin high level sampling for edge detection is started. If $\overline{\text{DREQ}}$ pin high level sampling has been completed by the time the DMA write cycle ends, acceptance resumes after the end of the write cycle, $\overline{\text{DREQ}}$ pin low level sampling is performed again, and this operation is repeated until the transfer ends.



[1] Acceptance after transfer enabling; the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of φ, and the request is held.

[2] [5] The request is cleared at the next bus break, and activation is started in the DMAC.

[3] [6] Start of DMA cycle; $\overline{\text{DREQ}}$ pin high level sampling on the rising edge of φ starts.

[4] [7] When the $\overline{\text{DREQ}}$ pin high level has been sampled, acceptance is resumed after the dead cycle is completed.
(As in [1], the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of φ, and the request is held.)

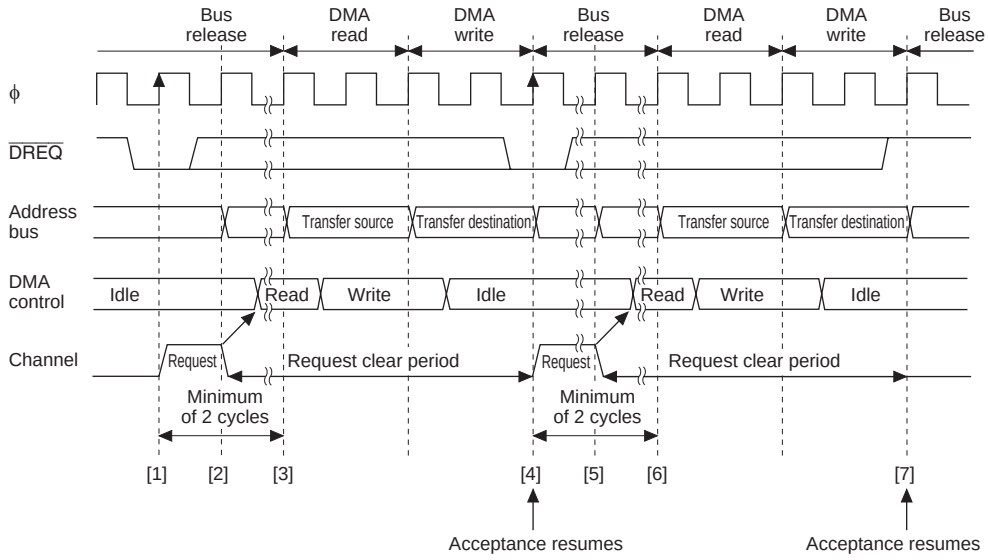
Note: In write data buffer mode, bus breaks from [2] to [7] may be hidden, and not visible.

Figure 7.24 Example of $\overline{\text{DREQ}}$ Pin Falling Edge Activated Block Transfer Mode Transfer

$\overline{\text{DREQ}}$ pin sampling is performed every cycle, with the rising edge of the next φ cycle after the end of the DMABCR write cycle for setting the transfer enabled state as the starting point.

When the $\overline{\text{DREQ}}$ pin low level is sampled while acceptance by means of the $\overline{\text{DREQ}}$ pin is possible, the request is held in the DMAC. Then, when activation is initiated in the DMAC, the request is cleared, and $\overline{\text{DREQ}}$ pin high level sampling for edge detection is started. If $\overline{\text{DREQ}}$ pin high level sampling has been completed by the time the DMA dead cycle ends, acceptance resumes after the end of the dead cycle, $\overline{\text{DREQ}}$ pin low level sampling is performed again, and this operation is repeated until the transfer ends.

Figure 7.25 shows an example of $\overline{\text{DREQ}}$ level activated normal mode transfer.



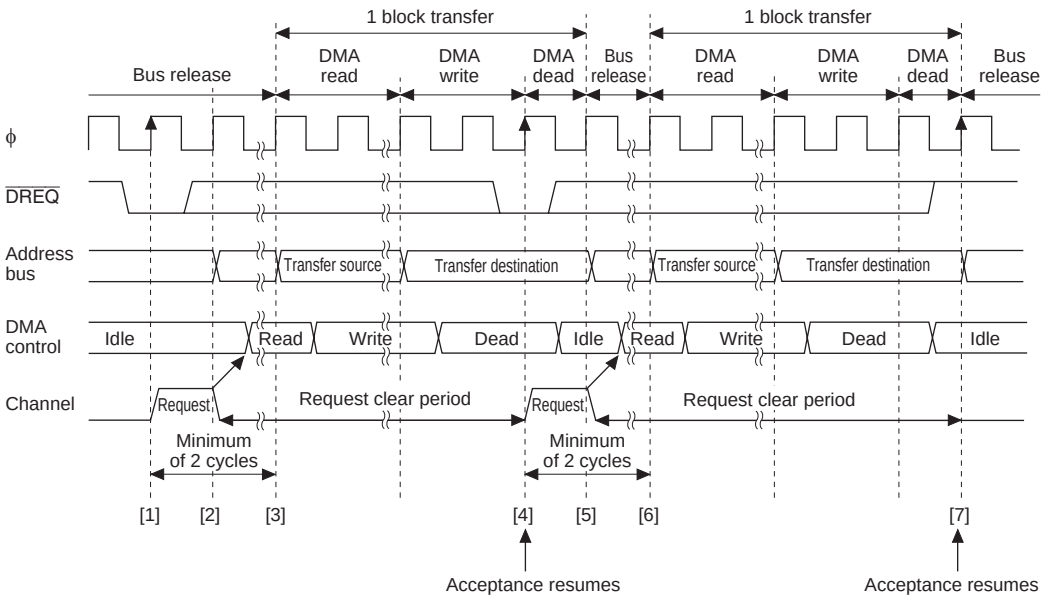
[1] Acceptance after transfer enabling; the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of ϕ, and the request is held.
[2] [5] The request is cleared at the next bus break, and activation is started in the DMAC.
[3] [6] The DMA cycle is started.
[4] [7] Acceptance is resumed after the write cycle is completed.
(As in [1], the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of ϕ, and the request is held.)

Note: In write data buffer mode, bus breaks from [2] to [7] may be hidden, and not visible.

Figure 7.25 Example of $\overline{\text{DREQ}}$ Level Activated Normal Mode Transfer

$\overline{\text{DREQ}}$ pin sampling is performed every cycle, with the rising edge of the next ϕ cycle after the end of the DMABCR write cycle for setting the transfer enabled state as the starting point.

When the $\overline{\text{DREQ}}$ pin low level is sampled while acceptance by means of the $\overline{\text{DREQ}}$ pin is possible, the request is held in the DMAC. Then, when activation is initiated in the DMAC, the request is cleared. After the end of the write cycle, acceptance resumes, $\overline{\text{DREQ}}$ pin low level sampling is performed again, and this operation is repeated until the transfer ends.



[1] Acceptance after transfer enabling; the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of φ, and the request is held.

[2] [5] The request is cleared at the next bus break, and activation is started in the DMAC.

[3] [6] The DMA cycle is started.

[4] [7] Acceptance is resumed after the dead cycle is completed. (As in [1], the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of φ, and the request is held.)

Note: In write data buffer mode, bus breaks from [2] to [7] may be hidden, and not visible.

Figure 7.26 Example of $\overline{\text{DREQ}}$ Level Activated Block Transfer Mode Transfer

$\overline{\text{DREQ}}$ pin sampling is performed every cycle, with the rising edge of the next φ cycle after the end of the DMABCR write cycle for setting the transfer enabled state as the starting point.

When the $\overline{\text{DREQ}}$ pin low level is sampled while acceptance by means of the $\overline{\text{DREQ}}$ pin is possible, the request is held in the DMAC. Then, when activation is initiated in the DMAC, the request is cleared. After the end of the dead cycle, acceptance resumes, $\overline{\text{DREQ}}$ pin low level sampling is performed again, and this operation is repeated until the transfer ends.

Single Address Mode (Read): Figure 7.27 shows a transfer example in which TEND output is enabled and byte-size single address mode transfer (read) is performed from external 8-bit, 2-state access space to an external device.

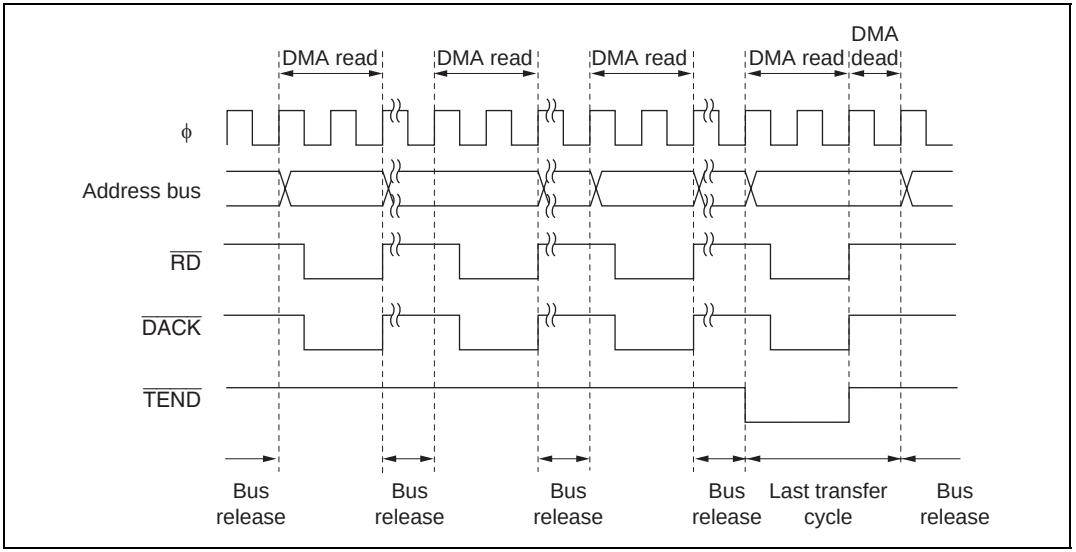


Figure 7.27 Example of Single Address Mode (Byte Read) Transfer

device.

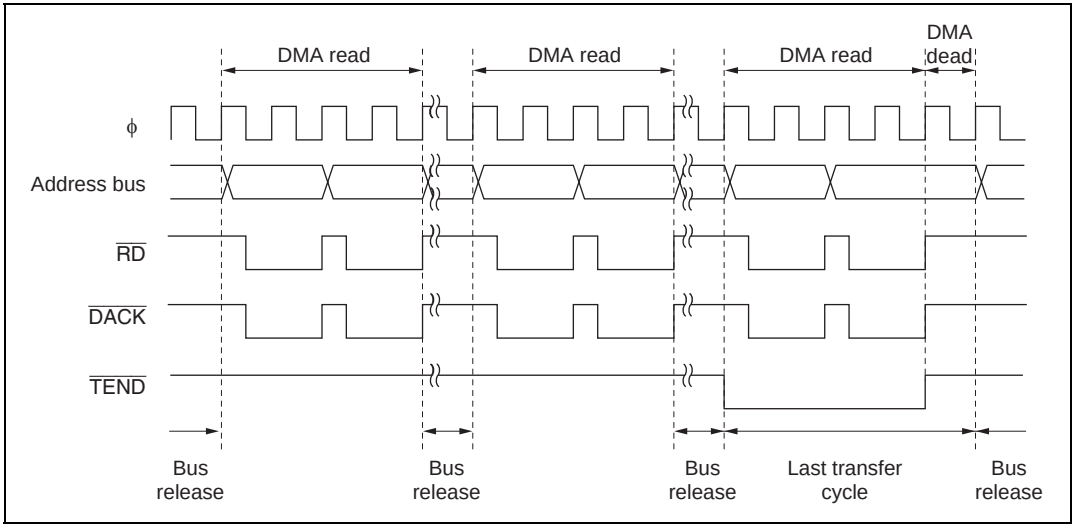


Figure 7.28 Example of Single Address Mode (Word Read) Transfer

A one-byte or one-word transfer is performed for one transfer request, and after the transfer the bus is released. While the bus is released, one or more bus cycles are executed by the CPU or DTC.

In the transfer end cycle (the cycle in which the transfer counter reaches 0), a one-state DMA dead cycle is inserted after the DMA write cycle.

external 8-bit, 2-state access space.

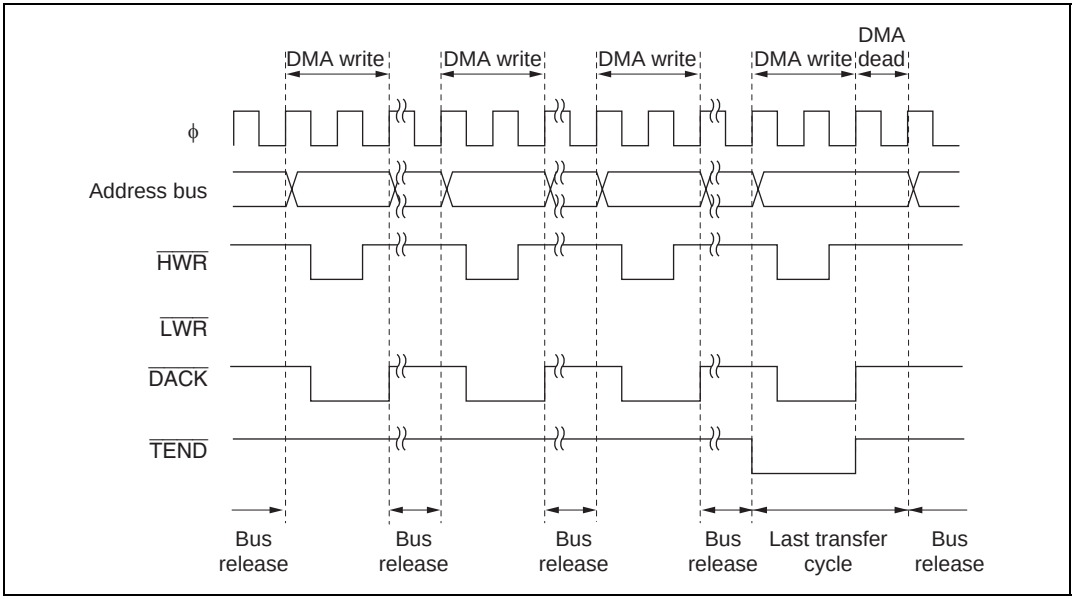


Figure 7.29 Example of Single Address Mode (Byte Write) Transfer

space.

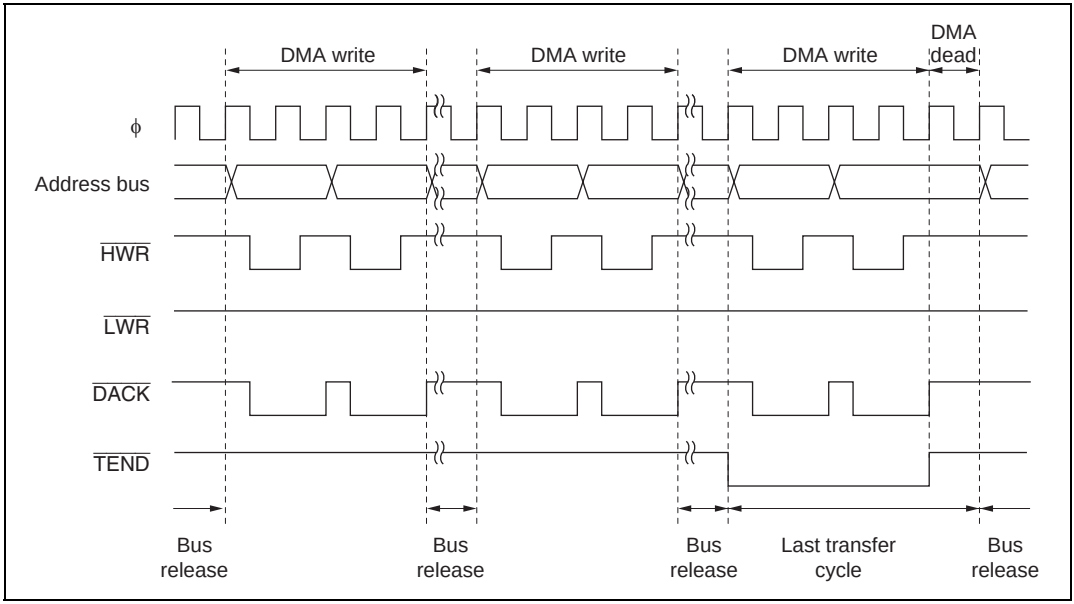
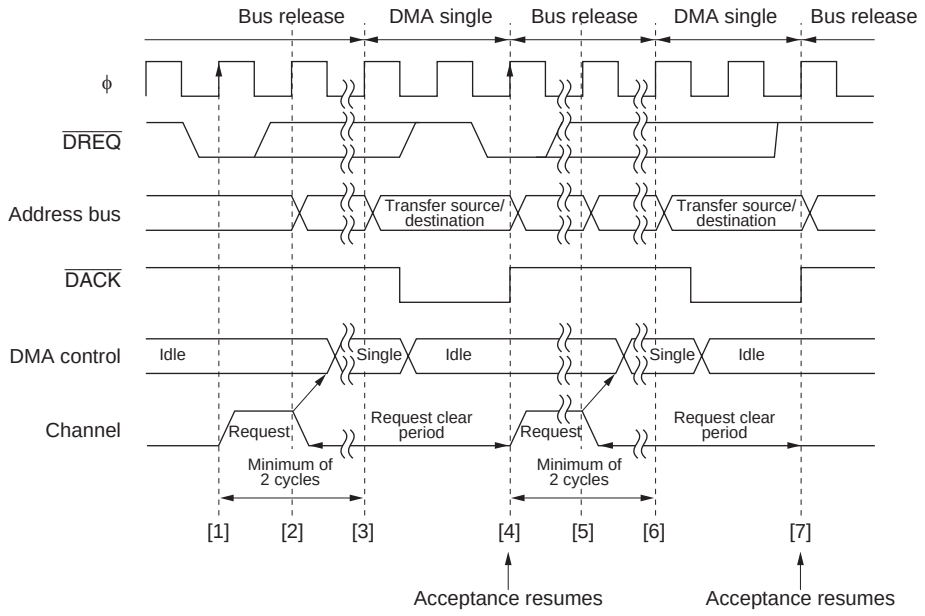


Figure 7.30 Example of Single Address Mode (Word Write) Transfer

A one-byte or one-word transfer is performed for one transfer request, and after the transfer the bus is released. While the bus is released one or more bus cycles are executed by the CPU or DTC.

In the transfer end cycle (the cycle in which the transfer counter reaches 0), a one-state DMA dead cycle is inserted after the DMA write cycle.

Figure 7.31 shows an example of $\overline{\text{DREQ}}$ pin falling edge activated single address mode transfer.



[1] Acceptance after transfer enabling; the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of φ, and the request is held.

[2] [5] The request is cleared at the next bus break, and activation is started in the DMAC.

[3] [6] Start of DMA cycle; $\overline{\text{DREQ}}$ pin high level sampling on the rising edge of φ starts.

[4] [7] When the $\overline{\text{DREQ}}$ pin high level has been sampled, acceptance is resumed after the single cycle is completed. (As in [1], the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of φ, and the request is held.)

Note: In write data buffer mode, bus breaks from [2] to [7] may be hidden, and not visible.

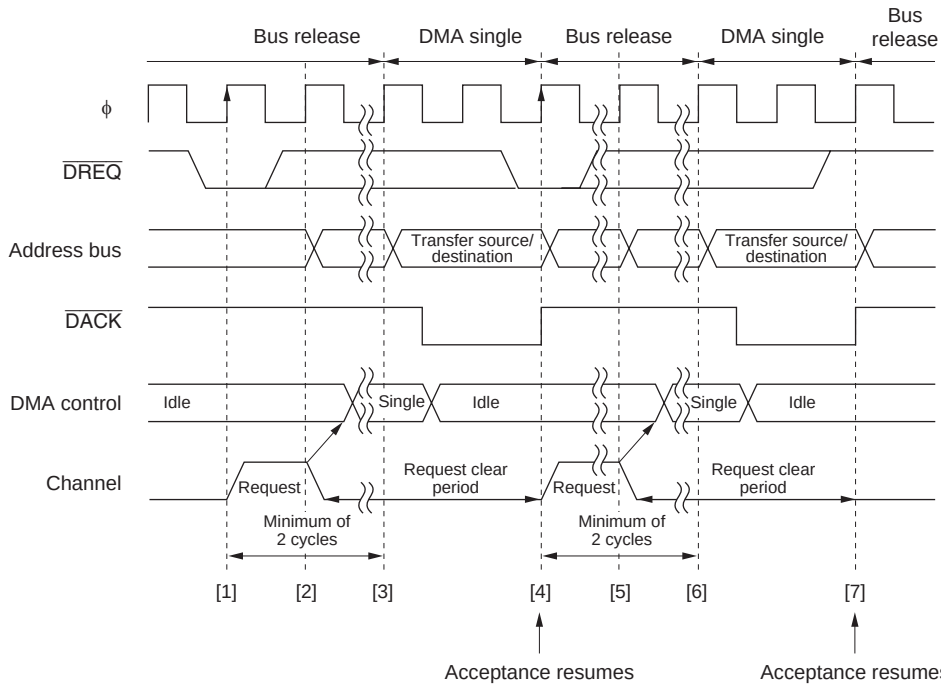
Figure 7.31 Example of $\overline{\text{DREQ}}$ Pin Falling Edge Activated Single Address Mode Transfer

$\overline{\text{DREQ}}$ pin sampling is performed every cycle, with the rising edge of the next φ cycle after the end of the DMABCR write cycle for setting the transfer enabled state as the starting point.

When the $\overline{\text{DREQ}}$ pin low level is sampled while acceptance by means of the $\overline{\text{DREQ}}$ pin is possible, the request is held in the DMAC. Then, when activation is initiated in the DMAC, the request is cleared, and $\overline{\text{DREQ}}$ pin high level sampling for edge detection is started. If $\overline{\text{DREQ}}$ pin high level sampling has been completed by the time the DMA single cycle ends, acceptance

$\overline{\text{DREQ}}$ Pin Low Level Activation Timing: Set the DTA bit for the channel for which the $\overline{\text{DREQ}}$ pin is selected to 1.

Figure 7.32 shows an example of $\overline{\text{DREQ}}$ pin low level activated single address mode transfer.



[1] Acceptance after transfer enabling; the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of φ, and the request is held.
[2] [5] The request is cleared at the next bus break, and activation is started in the DMAC.
[3] [6] The DMAC cycle is started.
[4] [7] Acceptance is resumed after the single cycle is completed.
(As in [1], the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of φ, and the request is held.)

Note: In write data buffer mode, bus breaks from [2] to [7] may be hidden, and not visible.

Figure 7.32 Example of $\overline{\text{DREQ}}$ Pin Low Level Activated Single Address Mode Transfer

$\overline{\text{DREQ}}$ pin sampling is performed every cycle, with the rising edge of the next φ cycle after the end of the DMABCR write cycle for setting the transfer enabled state as the starting point.

request is cleared. After the end of the single cycle, acceptance resumes, $\overline{\text{DREQ}}$ pin low level sampling is performed again, and this operation is repeated until the transfer ends.

7.5.12 Write Data Buffer Function

DMAC internal-to-external dual address transfers and single address transfers can be executed at high speed using the write data buffer function, enabling system throughput to be improved.

When the WDBE bit of BCRL in the bus controller is set to 1, enabling the write data buffer function, dual address transfer external write cycles or single address transfers and internal accesses (on-chip memory or internal I/O registers) are executed in parallel. Internal accesses are independent of the bus master, and DMAC dead cycles are regarded as internal accesses.

A low level can always be output from the $\overline{\text{TEND}}$ pin if the bus cycle in which a low level is to be output is an external bus cycle. However, a low level is not output from the $\overline{\text{TEND}}$ pin if the bus cycle in which a low level is to be output from the $\overline{\text{TEND}}$ pin is an internal bus cycle, and an external write cycle is executed in parallel with this cycle.

Figure 7.33 shows an example of burst mode transfer from on-chip RAM to external memory using the write data buffer function.

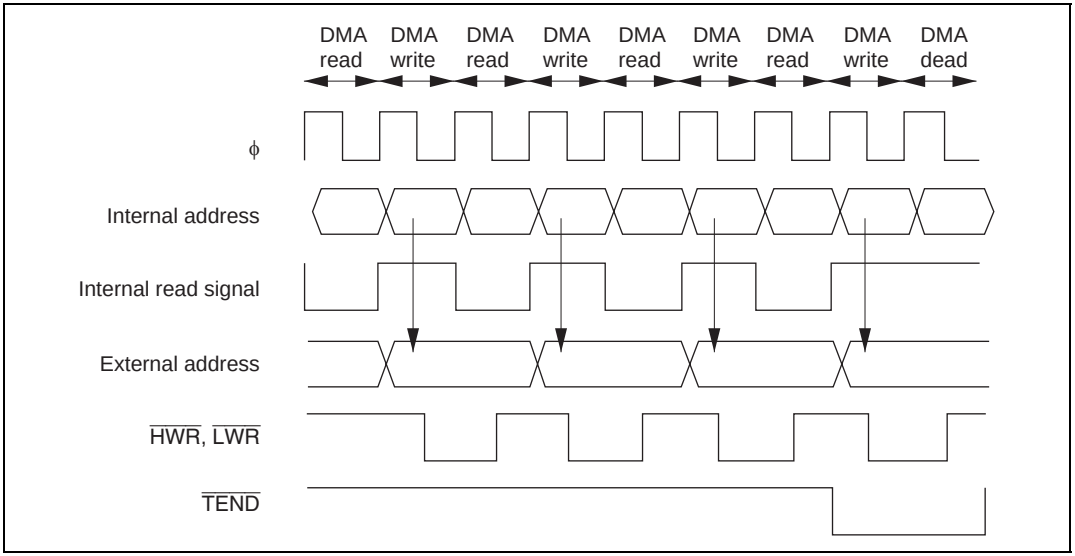


Figure 7.33 Example of Dual Address Transfer Using Write Data Buffer Function

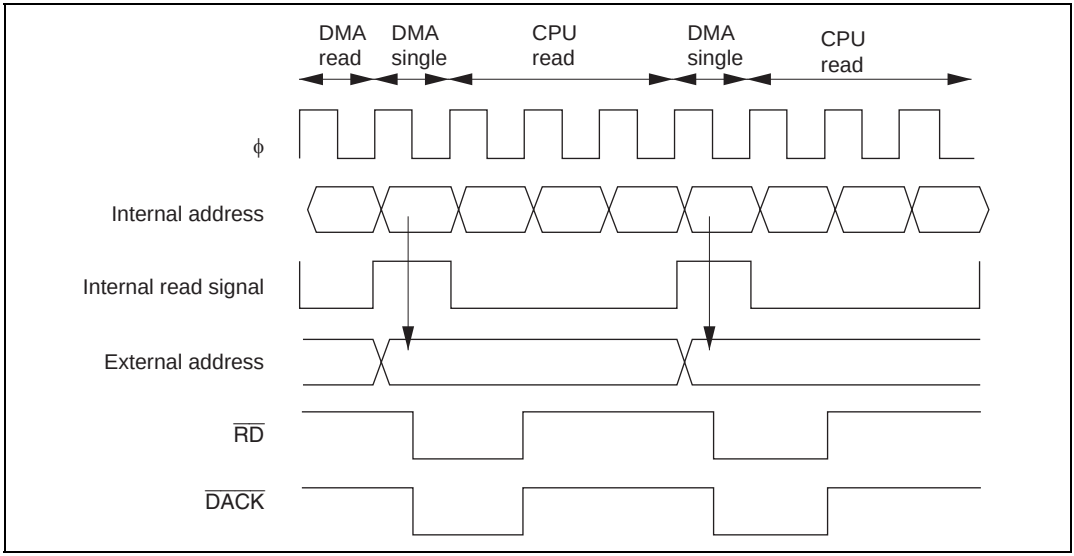


Figure 7.34 Example of Single Address Transfer Using Write Data Buffer Function

When the write data buffer function is activated, the DMAC recognizes that the bus cycle concerned has ended, and starts the next operation. Therefore, $\overline{\text{DREQ}}$ pin sampling is started one state after the start of the DMA write cycle or single address transfer.

7.5.13 DMAC Multi-Channel Operation

The DMAC channel priority order is: channel 0 > channel 1, and channel A > channel B. Table 7.13 summarizes the priority order for DMAC channels.

Table 7.13 DMAC Channel Priority Order

Short Address Mode	Full Address Mode	Priority
Channel 0A	Channel 0	High
Channel 0B		↑
Channel 1A	Channel 1	
Channel 1B		Low

highest-priority channel from among those issuing a request according to the priority order shown in table 7.13.

During burst transfer, or when one block is being transferred in block transfer, the channel will not be changed until the end of the transfer.

Figure 7.35 shows a transfer example in which transfer requests are issued simultaneously for channels 0A, 0B, and 1.

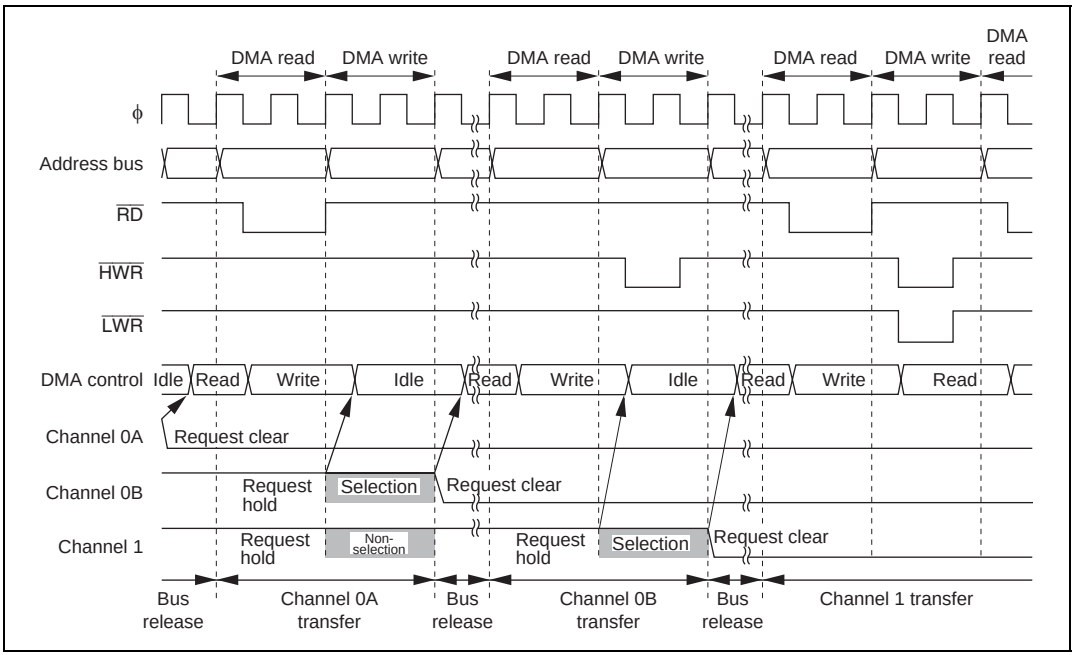


Figure 7.35 Example of Multi-Channel Transfer

There can be no break between a DMA cycle read and a DMA cycle write. This means that a refresh cycle, external bus release cycle, or DTC cycle is not generated between the external read and external write in a DMA cycle.

In the case of successive read and write cycles, such as in burst transfer or block transfer, a refresh or external bus released state may be inserted after a write cycle. Since the DTC has a lower priority than the DMAC, the DTC does not operate until the DMAC releases the bus.

When DMA cycle reads or writes are accesses to on-chip memory or internal I/O registers, these DMA cycles can be executed at the same time as refresh cycles or external bus release. However, simultaneous operation may not be possible when a write buffer is used.

When an NMI interrupt is requested, burst mode transfer in full address mode is interrupted. An NMI interrupt does not affect the operation of the DMAC in other modes.

In full address mode, transfer is enabled for a channel when both the DTE bit and the DTME bit are set to 1. With burst mode setting, the DTME bit is cleared when an NMI interrupt is requested.

If the DTME bit is cleared during burst mode transfer, the DMAC discontinues transfer on completion of the 1-byte or 1-word transfer in progress, then releases the bus, which passes to the CPU.

The channel on which transfer was interrupted can be restarted by setting the DTME bit to 1 again. Figure 7.36 shows the procedure for continuing transfer when it has been interrupted by an NMI interrupt on a channel designated for burst mode transfer.

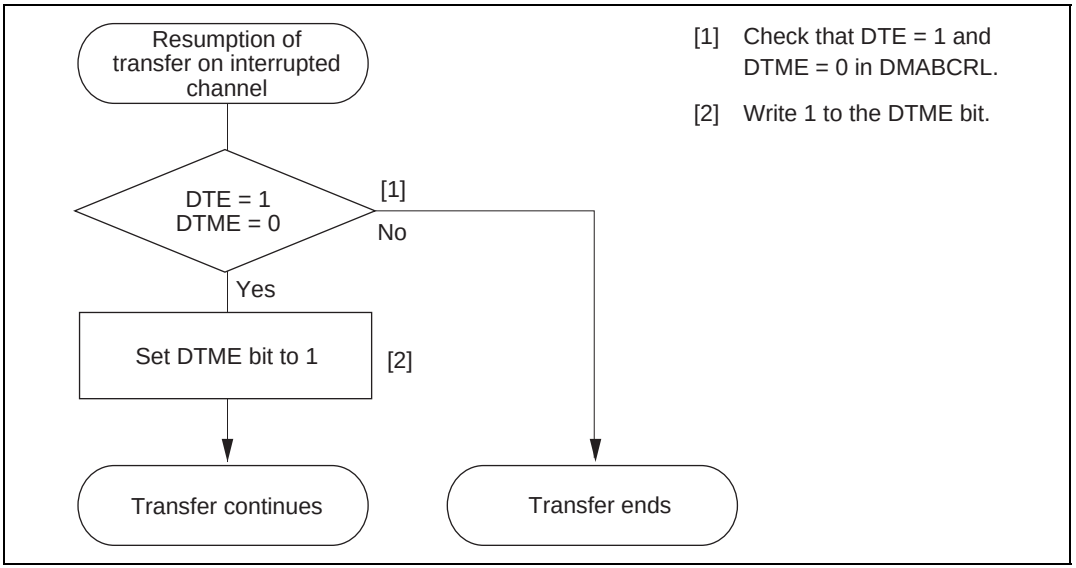


Figure 7.36 Example of Procedure for Continuing Transfer on Channel Interrupted by NMI Interrupt

If the DTE bit for the channel currently operating is cleared to 0, the DMAC stops on completion of the 1-byte or 1-word transfer in progress. DMAC operation resumes when the DTE bit is set to 1 again.

In full address mode, the same applies to the DTME bit.

Figure 7.37 shows the procedure for forcibly terminating DMAC operation by software.

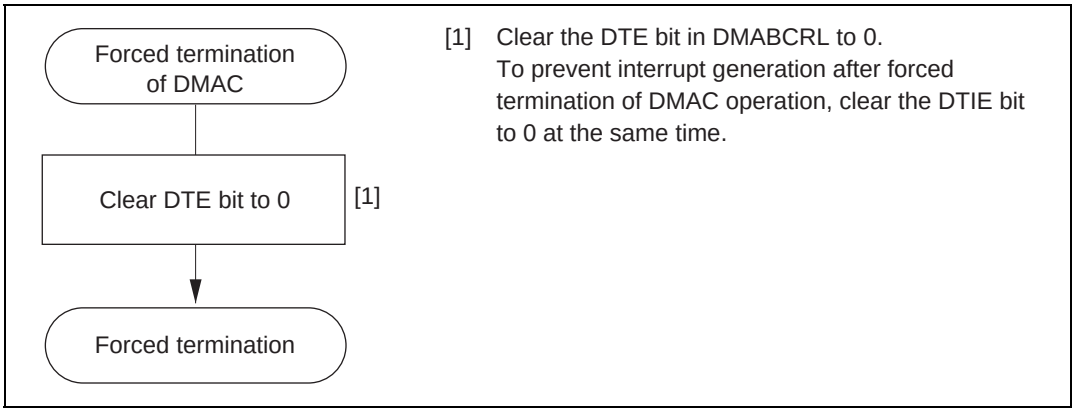


Figure 7.37 Example of Procedure for Forcibly Terminating DMAC Operation

Figure 7.38 shows the procedure for releasing and initializing a channel designated for full address mode. After full address mode has been cleared, the channel can be set to another transfer mode using the appropriate setting procedure.

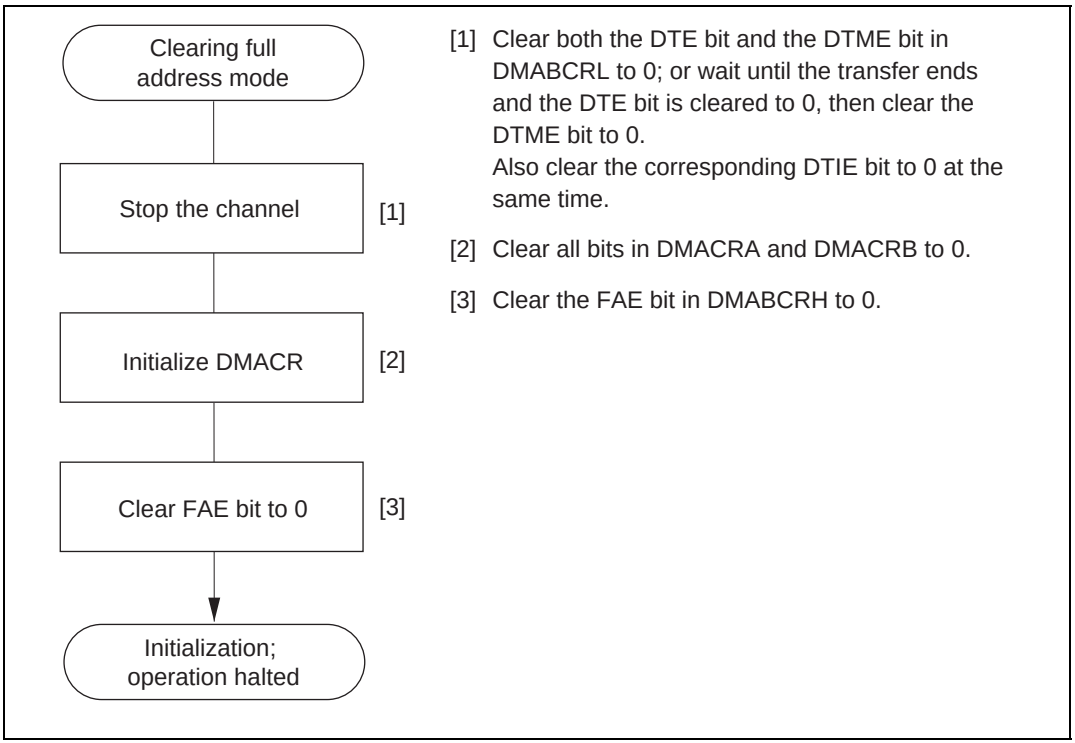


Figure 7.38 Example of Procedure for Clearing Full Address Mode

The sources of interrupts generated by the DMAC are transfer end and transfer break. Table 7.14 shows the interrupt sources and their priority order.

Table 7.14 Interrupt Source Priority Order

Interrupt Name	Interrupt Source		Interrupt Priority Order
	Short Address Mode	Full Address Mode	
DEND0A	Interrupt due to end of transfer on channel 0A	Interrupt due to end of transfer on channel 0	High
DEND0B	Interrupt due to end of transfer on channel 0B	Interrupt due to break in transfer on channel 0	
DEND1A	Interrupt due to end of transfer on channel 1A	Interrupt due to end of transfer on channel 1	
DEND1B	Interrupt due to end of transfer on channel 1B	Interrupt due to break in transfer on channel 1	Low

Enabling or disabling of each interrupt source is set by means of the DTIE bit for the corresponding channel in DMABCR, and interrupts from each source are sent to the interrupt controller independently.

The relative priority of transfer end interrupts on each channel is decided by the interrupt controller, as shown in table 7.14.

Figure 7.39 shows a block diagram of a transfer end/transfer break interrupt. An interrupt is always generated when the DTIE bit is set to 1 while the DTE bit is cleared to 0.

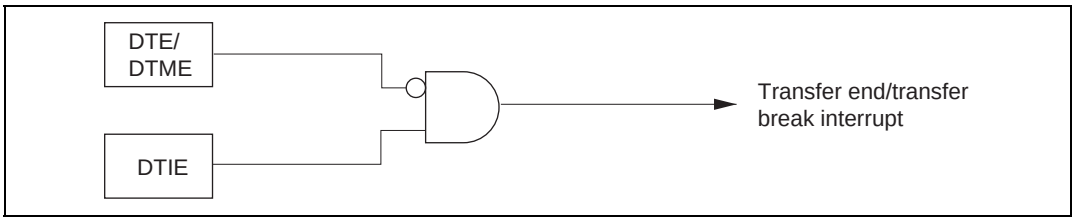


Figure 7.39 Block Diagram of Transfer End/Transfer Break Interrupt

In full address mode, a transfer break interrupt is generated when the DTME bit is cleared to 0 while the DTIEB bit is set to 1.

In both short address mode and full address mode, DMABCR should be set so as to prevent the occurrence of a combination that constitutes a condition for interrupt generation during setting.

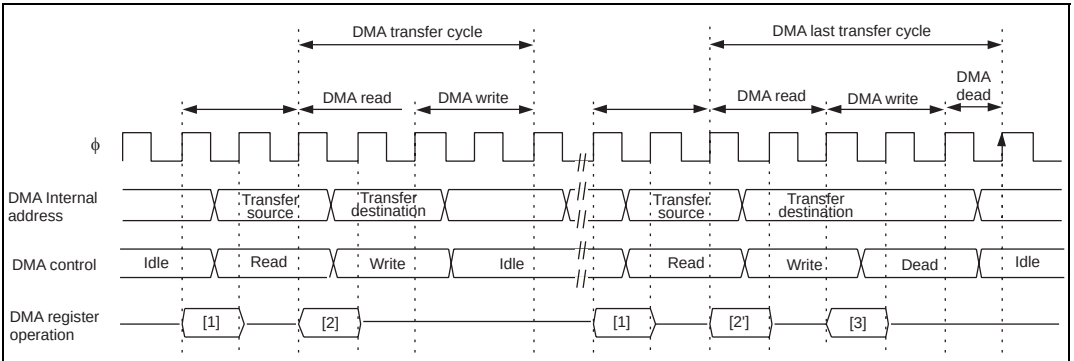
DMAC Register Access during Operation: Except for forced termination, the operating (including transfer waiting state) channel setting should not be changed. The operating channel setting should only be changed when transfer is disabled.

Also, MAC registers should not be written to in a DMA transfer.

DMAC register reads during operation (including the transfer waiting state) are described below.

(a) DMAC control starts one cycle before the bus cycle, with output of the internal address. Consequently, MAR is updated in the bus cycle before DMAC transfer.

Figure 7.40 shows an example of the update timing for DMAC registers in dual address transfer mode.



[1] Transfer source address register MAR operation (incremented/decremented/fixed)
Transfer counter ETCR operation (decremented)
Block size counter ETCR operation (decremented in block transfer mode)

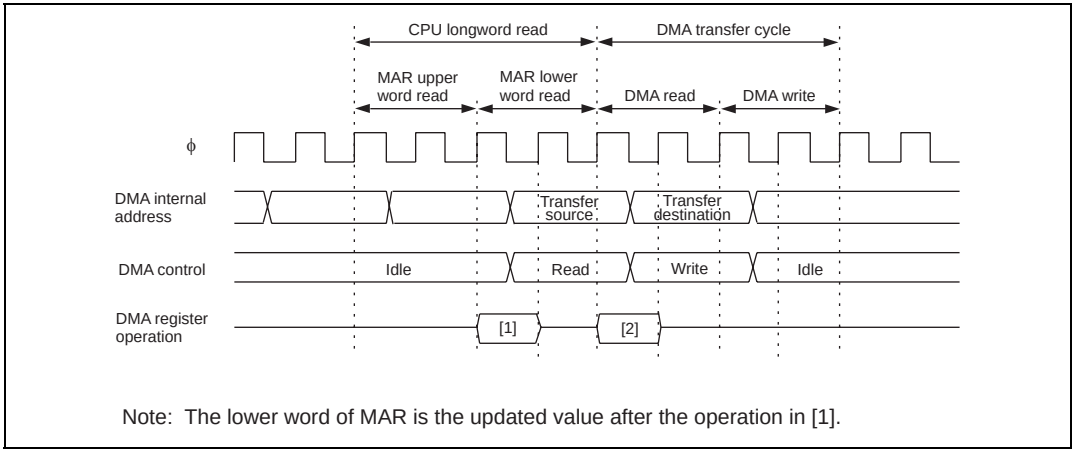
[2] Transfer destination address register MAR operation (incremented/decremented/fixed)

[2'] Transfer destination address register MAR operation (incremented/decremented/fixed)
Block transfer counter ETCR operation (decremented, in last transfer cycle of a block in block transfer mode)

[3] Transfer address register MAR restore operation (in block or repeat transfer mode)
Transfer counter ETCR restore (in repeat transfer mode)
Block size counter ETCR restore (in block transfer mode)

Notes: 1. In single address transfer mode, the update timing is the same as [1].
2. The MAR operation is post-incrementing/decrementing of the DMA internal address value.

Figure 7.40 DMAC Register Update Timing



Note: The lower word of MAR is the updated value after the operation in [1].

Figure 7.41 Contention between DMAC Register Update and CPU Read

Module Stop: When the MSTP15 bit in MSTPCR is set to 1, the DMAC clock stops, and the module stop state is entered. However, 1 cannot be written to the MSTP15 bit if any of the DMAC channels is enabled. This setting should therefore be made when DMAC operation is stopped.

When the DMAC clock stops, DMAC register accesses can no longer be made. Since the following DMAC register settings are valid even in the module stop state, they should be invalidated, if necessary, before a module stop.

- Transfer end/break interrupt (DTE = 0 and DTIE = 1)
- $\overline{\text{TEND}}$ pin enable (TEE = 1)
- $\overline{\text{DACK}}$ pin enable (FAE = 0 and SAE = 1)

Medium-Speed Mode: When the DTA bit is 0, internal interrupt signals specified as DMAC transfer sources are edge-detected.

In medium-speed mode, the DMAC operates on a medium-speed clock, while on-chip supporting modules operate on a high-speed clock. Consequently, if the period in which the relevant interrupt source is cleared by the CPU, DTC, or another DMAC channel, and the next interrupt is generated, is less than one state with respect to the DMAC clock (bus master clock), edge detection may not be possible and the interrupt may be ignored.

Also, in medium-speed mode, $\overline{\text{DREQ}}$ pin sampling is performed on the rising edge of the medium-speed clock.

address transfers and internal accesses (on-chip memory or internal I/O registers) are executed in parallel.

- Write Data Buffer Function and DMAC Register Setting

  If the setting of a register that controls external accesses is changed during execution of an external access by means of the write data buffer function, the external access may not be performed normally. Registers that control external accesses should only be manipulated when external reads, etc., are used with DMAC operation disabled, and the operation is not performed in parallel with external access.

- Write Data Buffer Function and DMAC Operation Timing

  The DMAC can start its next operation during external access using the write data buffer function. Consequently, the $\overline{\text{DREQ}}$ pin sampling timing, $\overline{\text{TEND}}$ output timing, etc., are different from the case in which the write data buffer function is disabled. Also, internal bus cycles maybe hidden, and not visible.

- Write Data Buffer Function and $\overline{\text{TEND}}$ Output

  A low level is not output at the $\overline{\text{TEND}}$ pin if the bus cycle in which a low level is to be output at the $\overline{\text{TEND}}$ pin is an internal bus cycle, and an external write cycle is executed in parallel with this cycle. Note, for example, that a low level may not be output at the $\overline{\text{TEND}}$ pin if the write data buffer function is used when data transfer is performed between an internal I/O register and on-chip memory.

  If at least one of the DMAC transfer addresses is an external address, a low level is output at the $\overline{\text{TEND}}$ pin.

  Figure 7.42 shows an example in which a low level is not output at the $\overline{\text{TEND}}$ pin.

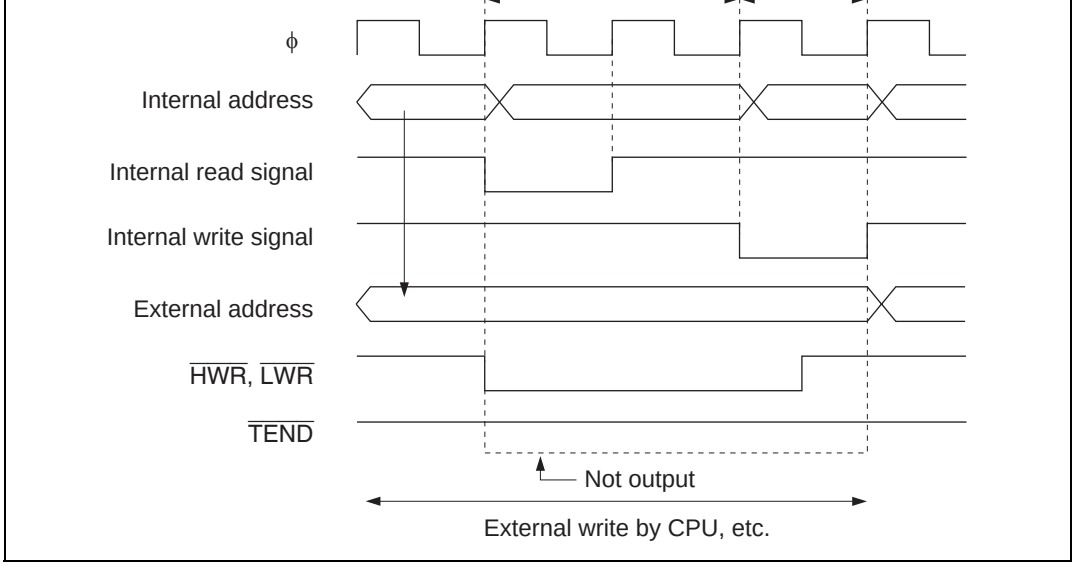


Figure 7.42 Example in Which Low Level is Not Output at $\overline{\text{TEND}}$ Pin

Activation by Falling Edge on $\overline{\text{DREQ}}$ Pin: $\overline{\text{DREQ}}$ pin falling edge detection is performed in synchronization with DMAC internal operations. The operation is as follows:

[1] Activation request wait state: Waits for detection of a low level on the $\overline{\text{DREQ}}$ pin, and switches to [2].

[2] Transfer wait state: Waits for DMAC data transfer to become possible, and switches to [3].

[3] Activation request disabled state: Waits for detection of a high level on the $\overline{\text{DREQ}}$ pin, and switches to [1].

After DMAC transfer is enabled, a transition is made to [1]. Thus, initial activation after transfer is enabled is performed on detection of a low level.

Activation Source Acceptance: At the start of activation source acceptance, a low level is detected in both $\overline{\text{DREQ}}$ pin falling edge sensing and low level sensing. Similarly, in the case of an internal interrupt, the interrupt request is detected. Therefore, a request is accepted from an internal interrupt or $\overline{\text{DREQ}}$ pin low level that occurs before execution of the DMABCRL write to enable transfer.

When the DMAC is activated, take any necessary steps to prevent an internal interrupt or $\overline{\text{DREQ}}$ pin low level remaining from the end of the previous transfer, etc.

or DTC even if DTA is set to 1.

Also, if internal DMAC activation has already been initiated when operation is forcibly terminated, the transfer is executed but flag clearing is not performed for the selected internal interrupt even if DTA is set to 1.

An internal interrupt request following the end of transfer or a forcible termination should be handled by the CPU as necessary.

Channel Re-Setting: To reactivate a number of channels when multiple channels are enabled, use exclusive handling of transfer end interrupts, and perform DMABCR control bit operations exclusively.

Note, in particular, that in cases where multiple interrupts are generated between reading and writing of DMABCR, and a DMABCR operation is performed during new interrupt handling, the DMABCR write data in the original interrupt handling routine will be incorrect, and the write may invalidate the results of the operations by the multiple interrupts. Ensure that overlapping DMABCR operations are not performed by multiple interrupts, and that there is no separation between read and write operations by the use of a bit-manipulation instruction.

Also, when the DTE and DTME bits are cleared by the DMAC or are written with 0, they must first be read while cleared to 0 before the CPU can write a 1 to them.

8.1 Overview

The chip includes a data transfer controller (DTC). The DTC can be activated for data transfer by an interrupt or software.

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)
  - Chain transfer execution can be set after data transfer (when counter = 0)
- Selection of transfer modes
  - Normal, repeat, and block transfer modes available
  - Incrementing, decrementing, and fixing of source and destination addresses can be selected
- 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 all the specified data transfers have ended
- Activation by software is possible
- Module stop mode can be set
  - The initial setting enables DTC registers to be accessed. DTC operation is halted by setting module stop mode

Figure 8.1 shows a block diagram of the DTC.

The DTC's register information is stored in the on-chip RAM*. A 32-bit bus connects the DTC to the on-chip RAM (1 kbyte), enabling 32-bit, 1-state reading and writing of DTC register information.

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

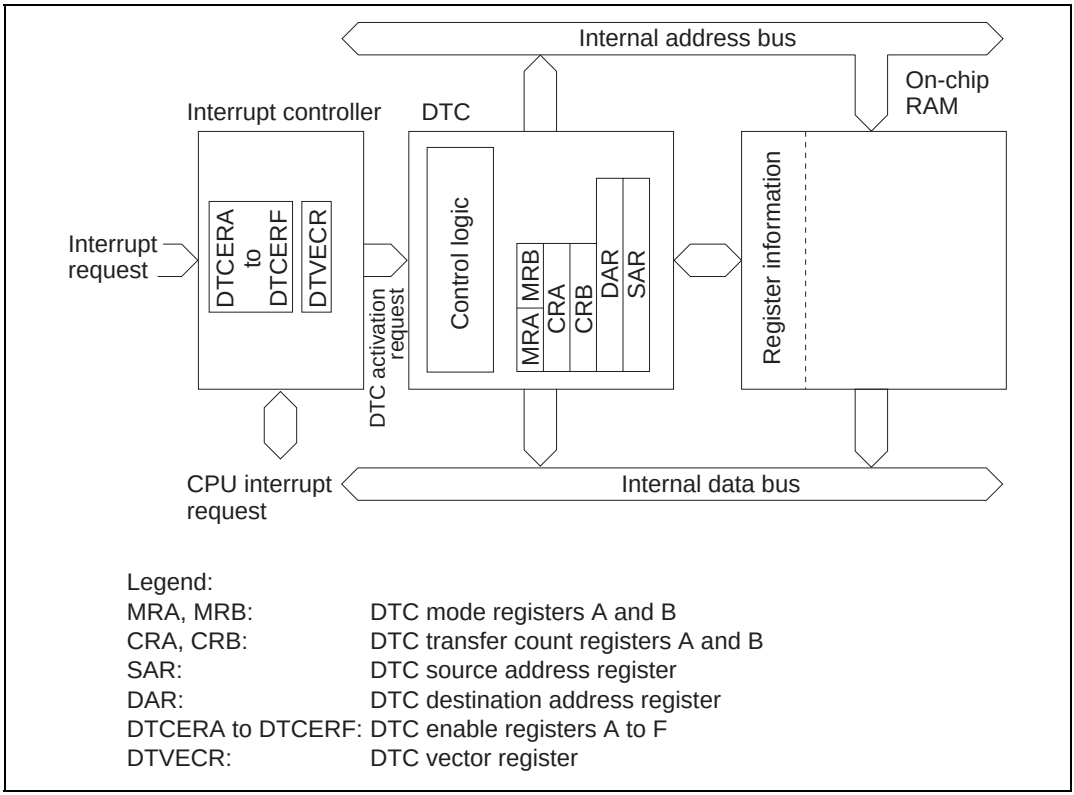


Figure 8.1 Block Diagram of DTC

Table 8.1 summarizes the DTC registers.

Table 8.1 DTC Registers

Name	Abbreviation	R/W	Initial Value	Address*1
DTC mode register A	MRA	—*2	Undefined	—*3
DTC mode register B	MRB	—*2	Undefined	—*3
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'FF30 to H'FF35
DTC vector register	DTVECR	R/W	H'00	H'FF37
Module stop control register	MSTPCR	R/W	H'3FFF	H'FF3C

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'F800 to H'FBFF. It cannot be located in external space. When the DTC is used, do not clear the RAME bit in SYSCR to 0.

8.2.1 DTC Mode Register A (MRA)

Bit	:	7	6	5	4	3	2	1	0
		SM1	SM0	DM1	DM0	MD1	MD0	DTS	Sz
Initial value	:	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined
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 whether SAR is to be incremented, decremented, or left fixed after a data transfer.

Bit 7 SM1	Bit 6 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 whether DAR is to be incremented, decremented, or left fixed after a data transfer.

Bit 5 DM1	Bit 4 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 3 MD1	Bit 2 MD0	Description
0	0	Normal mode
	1	Repeat mode
1	0	Block transfer mode
	1	—

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

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

8.2.2 DTC Mode Register B (MRB)

Bit :	7	6	5	4	3	2	1	0
	CHNE	DISEL	CHNS	—	—	—	—	—
Initial value :	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined
R/W :	—	—	—	—	—	—	—	—

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

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

When CHNE is set to 1, the chain transfer condition can be selected with the CHNS bit.

Bit 7 CHNE	Description
0	End of DTC data transfer (activation waiting state)
1	DTC chain transfer (new register information is read, then data is transferred)

Bit 6—DTC Interrupt Select (DISEL): Specifies whether interrupt requests to the CPU are 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 counter is 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 clear the interrupt source flag of the activating interrupt to 0)

Bit 5—DTC Chain Transfer Select (CHNS): Specifies the chain transfer condition when CHNE is 1.

Bit 7 CHNE	Bit 5 CHNS	Description
0	—	No chain transfer (DTC data transfer end, activation waiting state entered)
1	0	DTC chain transfer
1	1	Chain transfer only when transfer counter = 0

Bits 4 to 0—Reserved: These bits have no effect on DTC operation in the chip and should always be written with 0.

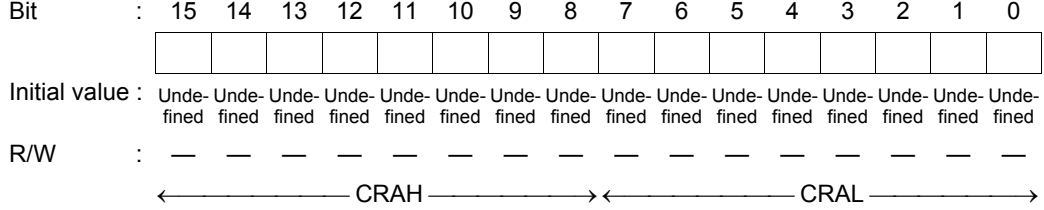
Bit	:	23	22	21	20	19	– – –	4	3	2	1	0
							– – –					
Initial value	:	Undefined	Undefined	Undefined	Undefined	Undefined	– – –	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	:	—	—	—	—	—	– – –	—	—	—	—	—

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

8.2.4 DTC Destination Address Register (DAR)

Bit	:	23	22	21	20	19	– – –	4	3	2	1	0
							– – –					
Initial value	:	Undefined	Undefined	Undefined	Undefined	Undefined	– – –	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	:	—	—	—	—	—	– – –	—	—	—	—	—

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

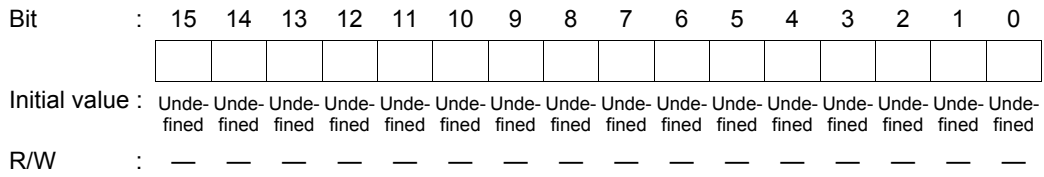


CRA is a 16-bit register that designates the number of times data is to be transferred by the DTC.

In normal mode, the entire CRA register functions as a 16-bit transfer counter (1 to 65,536). It is decremented by 1 every time data is transferred, and transfer ends when the count reaches H'0000.

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

8.2.6 DTC Transfer Count Register B (CRB)



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

Bit	:	7	6	5	4	3	2	1	0
		DTCE7	DTCE6	DTCE5	DTCE4	DTCE3	DTCE2	DTCE1	DTCE0
Initial value	:	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

The DTC enable registers comprise six 8-bit readable/writable registers, DTCERA to DTCERF, with bits corresponding to the interrupt sources that can activate the DTC. These bits enable or disable DTC service for the corresponding interrupt sources.

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

Bit n—DTC Activation Enable (DTCEn)

Bit n DTCEn	Description
0	DTC activation by this interrupt is disabled (Initial value) [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 ended

(n = 7 to 0)

A DTCE bit can be set for each interrupt source that can activate the DTC. The correspondence between interrupt sources and DTCE bits is shown in table 8.5, together with the vector numbers generated by the interrupt controller.

For DTCE bit setting, read/write operations must be performed using bit-manipulation instructions such as BSET and BCLR. For the initial setting only, however, when multiple activation sources are set at one time, it is possible to disable interrupts and write after executing a dummy read on the relevant register.

Bit	:	7	6	5	4	3	2	1	0
		SWDTE	DTVEC6	DTVEC5	DTVEC4	DTVEC3	DTVEC2	DTVEC1	DTVEC0
Initial value	:	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)*

Note: * 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 by 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 activation by software.

Bit 7 SWDTE	Description
0	DTC software activation is disabled (Initial value) [Clearing conditions] • When the DISEL bit is 0 and the specified number of transfers have not ended • When 0 is written after a software activation data-transfer-complete interrupt is issued 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): These bits specify a vector number for DTC software activation.

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

	MSTPCRH								MSTPCRL							
Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value :	0	0	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/W	R/W	R/W

MSTPCR is a 16-bit readable/writable register that performs module stop mode control.

When the MSTP14 bit in MSTPCR is set to 1, DTC operation stops at the end of the bus cycle and a transition is made to module stop mode. However, 1 cannot be written in the MSTP14 bit while the DTC is operating. For details, see section 21.5, Module Stop Mode.

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

Bit 14—Module Stop (MSTP14): Specifies the DTC module stop mode.

Bit 14 MSTP14	Description	
0	DTC module stop mode cleared	(Initial value)
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 and transfers data on the basis of that register information. After the data transfer, it writes updated register information back to memory. Pre-storage of register information in memory makes it possible to transfer data over any required number of channels. Setting the CHNE bit to 1 makes it possible to perform a number of transfers with a single activation. A setting can also be made to have chain transfer performed only when the transfer counter value is 0. This enables DTC re-setting to be performed by the DTC itself.

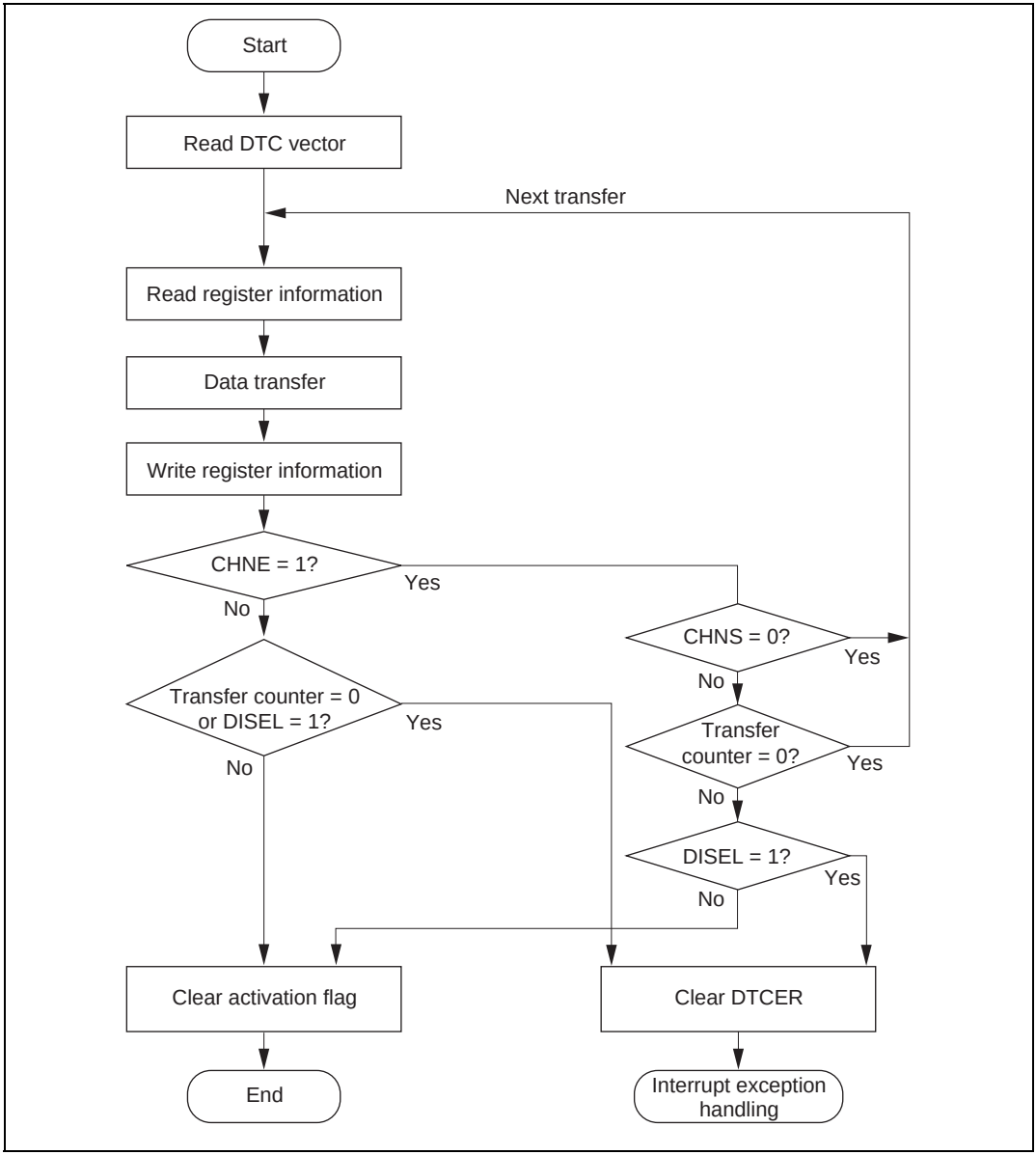


Figure 8.2 Flowchart of DTC Operation

1st Transfer				2nd Transfer				
CHNE	**CHNS**	**DISEL**	**CR**	**CHNE**	**CHNS**	**DISEL**	**CR**	**DTC Transfer**
0	—	0	Not 0	—	—	—	—	Ends at 1st transfer
0	—	0	0	—	—	—	—	Ends at 1st transfer
0	—	1	—	—	—	—	—	Interrupt request to CPU
1	0	—	—	0	—	0	Not 0	Ends at 2nd transfer
				0	—	0	0	Ends at 2nd transfer
				0	—	1	—	Interrupt request to CPU
1	1	0	Not 0	—	—	—	—	Ends at 1st transfer
1	1	—	0	0	—	0	Not 0	Ends at 2nd transfer
				0	—	0	0	Ends at 2nd transfer
				0	—	1	—	Interrupt request to CPU
1	1	1	Not 0	—	—	—	—	Ends at 1st transfer Interrupt request to CPU

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 designates the transfer destination address. After each transfer, SAR and DAR are independently incremented, decremented, or left fixed.

Table 8.3 outlines the functions of the DTC.

Transfer Mode	Activation Source	Address Registers	
		Transfer Source	Transfer Destination
• Normal mode — One transfer request transfers one byte or one word — Memory addresses are incremented 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	• IRQ • TPU TGI • 8-bit timer CMI • SCI TXI or RXI • A/D converter ADI • DMAC DEND • Software	24 bits	24 bits

The DTC operates when activated by an interrupt or by a write to DTVECR by software. An interrupt request can be directed to the CPU or DTC, as designated by the corresponding DTCER bit. An interrupt becomes a DTC activation source when the corresponding bit is set to 1, and a 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 transfer), the activation source or corresponding DTCER bit is cleared. Table 8.4 shows activation source and DTCER clearance. The activation source flag, in the case of RXI0, for example, is the RDRF flag of SCI0.

Table 8.4 Activation Source and DTCER Clearance

Activation Source	When the DISEL Bit Is 0 and the Specified Number of Transfers Have Not Ended	When the DISEL Bit Is 1, or when the Specified Number of Transfers Have Ended
Software activation	The SWDTE bit is cleared to 0	• The SWDTE bit remains set to 1 • An interrupt is issued to the CPU
Interrupt activation	• The corresponding DTCER bit remains set to 1 • The activation source flag is cleared to 0	• The corresponding DTCER bit is cleared to 0 • The activation source flag remains set to 1 • A request is issued to the CPU for the activation source interrupt

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

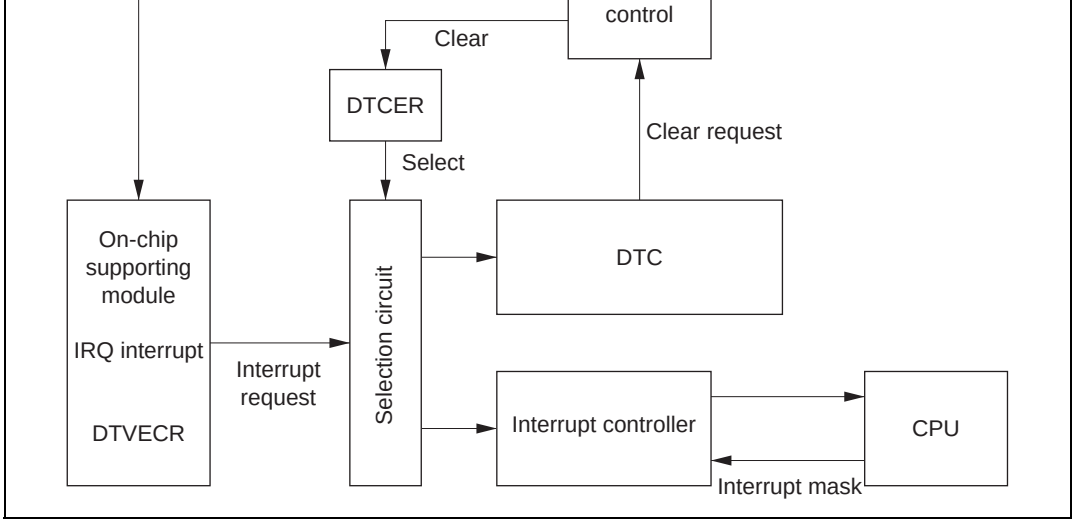


Figure 8.3 Block Diagram of DTC Activation Source Control

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

8.3.3 DTC Vector Table

Figure 8.4 shows the correspondence between DTC vector addresses and register information.

Table 8.5 shows the correspondence between activation, vector addresses, and DTCER bits. When the DTC is activated by software, the vector address is obtained from: H'0400 + (DTVECR[6:0] << 1) (where << 1 indicates a 1-bit left shift). For example, if DTVECR is H'10, the vector address is H'0420.

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

The configuration of the vector address is a 2-byte unit. These two bytes specify the lower bits of the address in the on-chip RAM.

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address	DTCE*	Priority
Write to DTVECR	Software	DTVECR	H'0400+ (DTVECR [6:0]<<1)	—	High
IRQ0	External pin	16	H'0420	DTCEA7	
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	
IRQ6		22	H'042C	DTCEA1	
IRQ7		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	Low

Interrupt Source	Source	Number	Address	DTCE	Priority
TGI3A (GR3A compare match/ input capture)	TPU channel 3	48	H'0460	DTCEC5	High
TGI3B (GR3B compare match/ input capture)		49	H'0462	DTCEC4	
TGI3C (GR3C compare match/ input capture)		50	H'0464	DTCEC3	
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	
CMIA0	8-bit timer channel 0	64	H'0480	DTCED3	
CMIB0		65	H'0482	DTCED2	
CMIA1	8-bit timer channel 1	68	H'0488	DTCED1	
CMIB1		69	H'048A	DTCED0	
DMTEND0A (DMAC transfer complete 0)	DMAC	72	H'0490	DTCEE7	
DMTEND0B (DMAC transfer complete 1)		73	H'0492	DTCEE6	
DMTEND1A (DMAC transfer complete 2)		74	H'0494	DTCEE5	
DMTEND1B (DMAC transfer complete 3)		75	H'0496	DTCEE4	
RXI0 (receive-data-full 0)	SCI channel 0	81	H'04A2	DTCEE3	
TXI0 (transmit-data-empty 0)		82	H'04A4	DTCEE2	
RXI1 (receive-data-full 1)	SCI channel 1	85	H'04AA	DTCEE1	
TXI1 (transmit-data-empty 1)		86	H'04AC	DTCEE0	
RXI2 (receive-data-full 2)	SCI channel 2	89	H'04B2	DTCEF7	
TXI2 (transmit-data-empty 2)		90	H'04B4	DTCEF6	Low

Note: * DTCE bits with no corresponding interrupt are reserved, and should be written with 0.

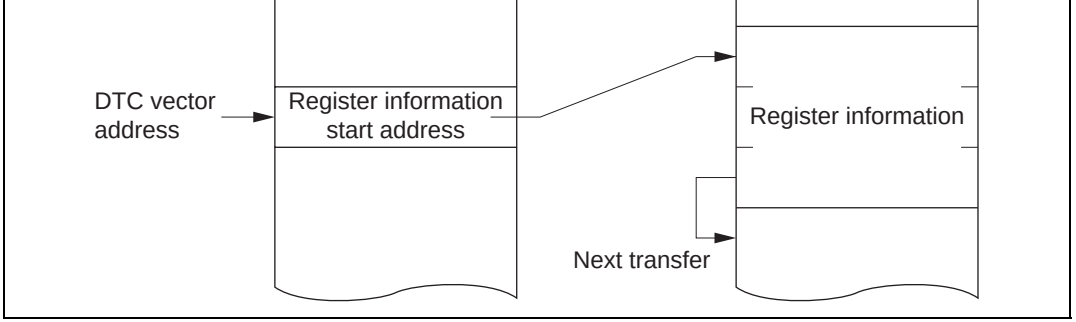


Figure 8.4 Correspondence between DTC Vector Address and Register Information

8.3.4 Location of Register Information in Address Space

Figure 8.5 shows how the register information should be located in the address space.

Locate the MRA, SAR, MRB, DAR, CRA, and CRB registers, in that order, from the start address of the register information (contents of the vector address). In the case of chain transfer, register information should be located in consecutive areas.

Locate the register information in the on-chip RAM (addresses: H'FFF800 to H'FFFBFF).

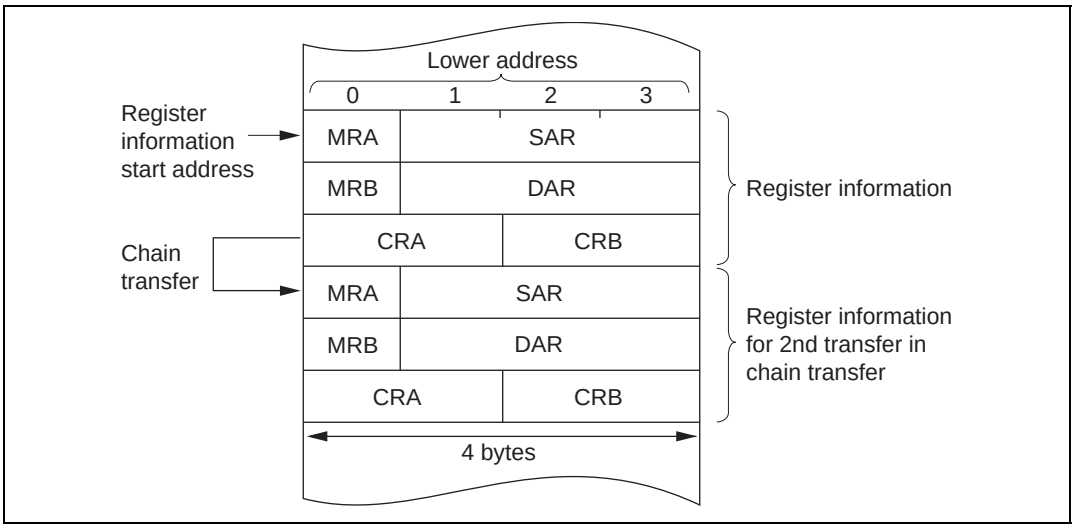


Figure 8.5 Location of DTC Register Information in Address Space

In normal mode, one operation transfers one byte or one word of data.

From 1 to 65,536 transfers can be specified. Once the specified number of transfers have ended, a CPU interrupt can be requested.

Table 8.6 lists the register information in normal mode and figure 8.6 shows the memory map in normal mode.

Table 8.6 Register Information in Normal Mode

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

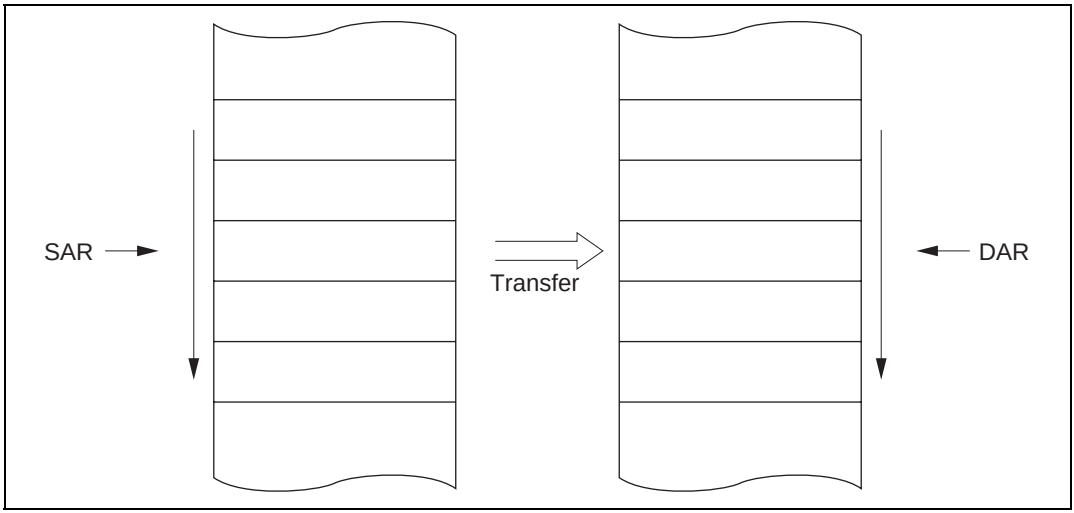


Figure 8.6 Memory Map in Normal Mode

In repeat mode, one operation transfers one byte or one word of data.

From 1 to 256 transfers can be specified. Once the specified number of transfers have ended, the initial state of the transfer counter and the address register specified as the repeat area is restored, and transfer is repeated. In repeat mode the transfer counter value does not reach H'00, and therefore CPU interrupts cannot be requested when DISEL = 0.

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

Table 8.7 Register Information in Repeat Mode

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

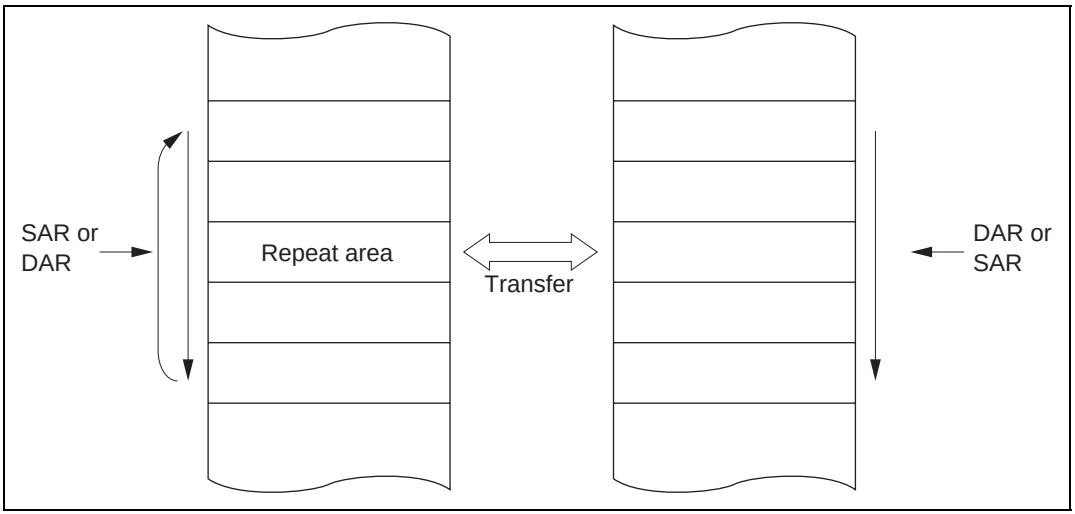


Figure 8.7 Memory Map in Repeat Mode

In block transfer mode, one operation transfers one block of data. Either the transfer source or the transfer destination is designated as a block area.

The block size is 1 to 256. When the transfer of one block ends, the initial state of the block size counter and the address register specified as the block area is restored. The other address register is then incremented, decremented, or left fixed.

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

Table 8.8 lists the register information in block transfer mode and figure 8.8 shows the memory map in block transfer mode.

Table 8.8 Register Information in Block Transfer Mode

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

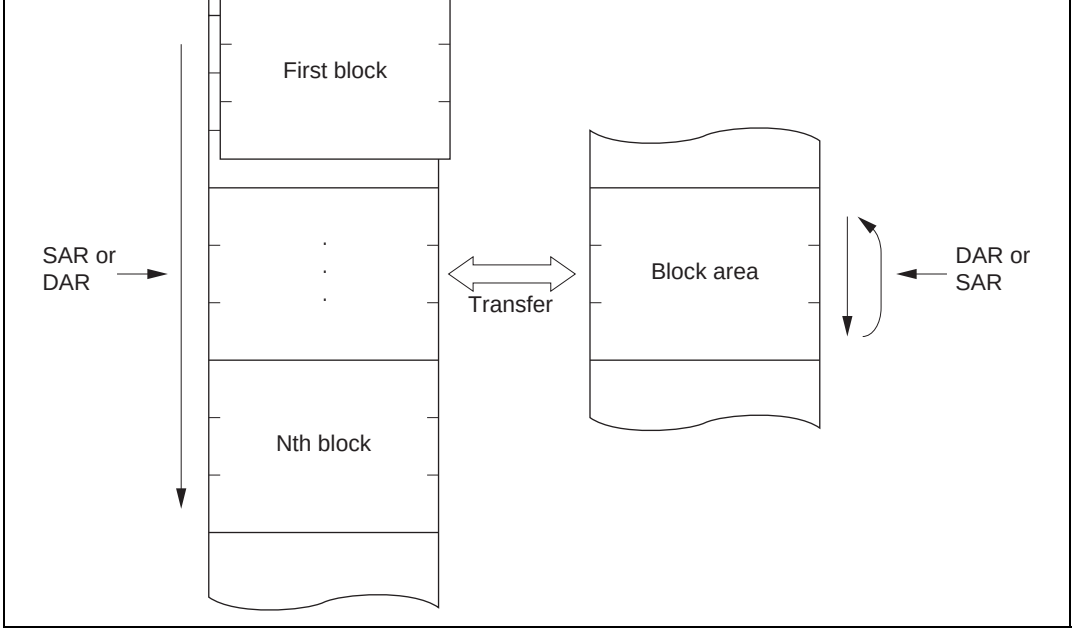


Figure 8.8 Memory Map in Block Transfer Mode

Setting the CHNE bit to 1 enables a number of data transfers to be performed consecutively in response to a single transfer request. It is also possible, by setting both the CHNE bit and CHNS bit to 1, to specify execution of chain transfer only when the transfer counter value is 0. SAR, DAR, CRA, CRB, MRA, and MRB, which define data transfers, can be set independently.

Figure 8.9 shows the memory map for chain transfer.

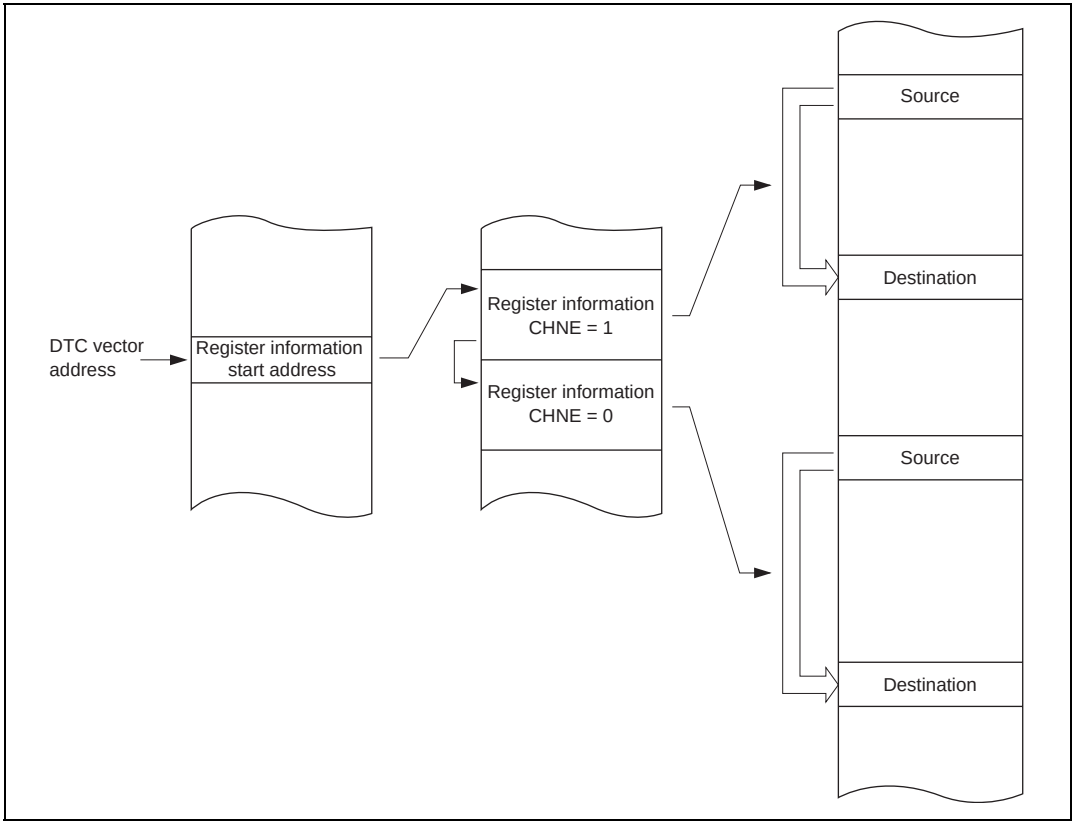


Figure 8.9 Chain Transfer Memory Map

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

Figures 8.10 to 8.12 show examples of DTC operation timing.

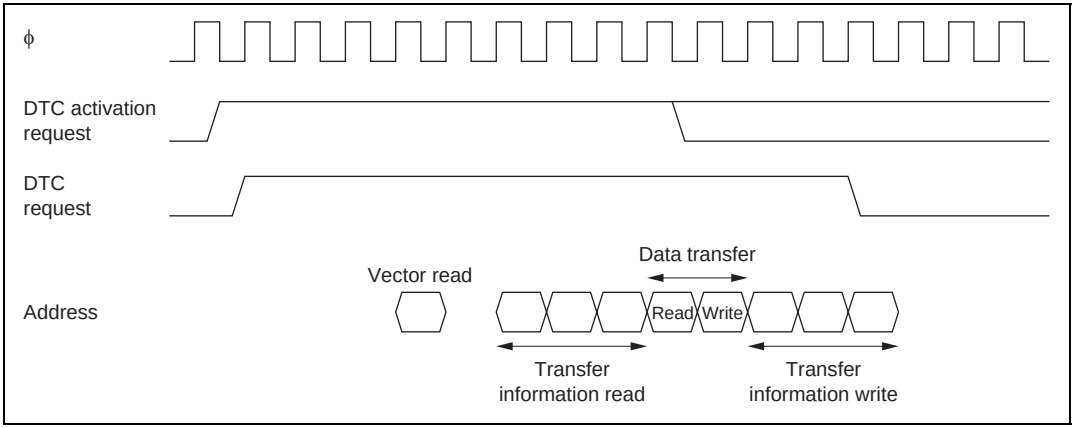


Figure 8.10 DTC Operation Timing (Example in Normal Mode or Repeat Mode)

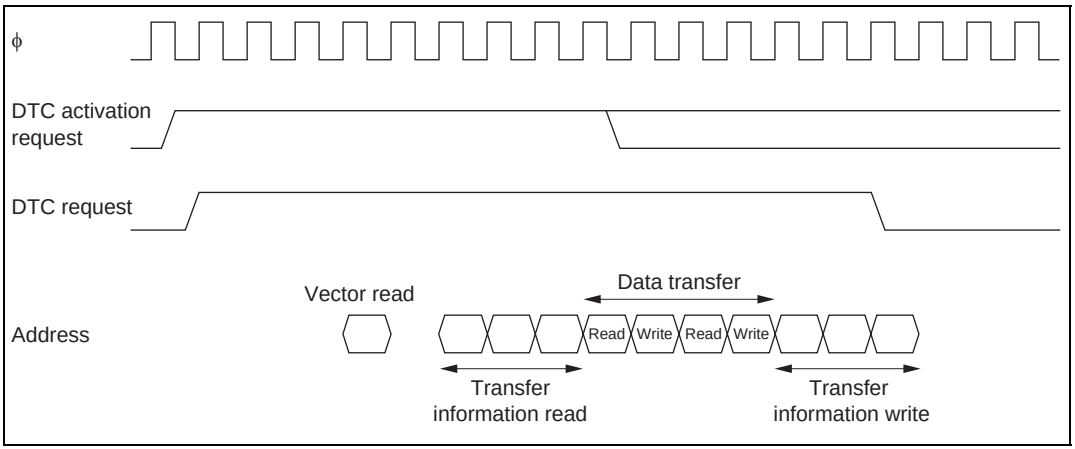


Figure 8.11 DTC Operation Timing (Example of Block Transfer Mode, with Block Size of 2)

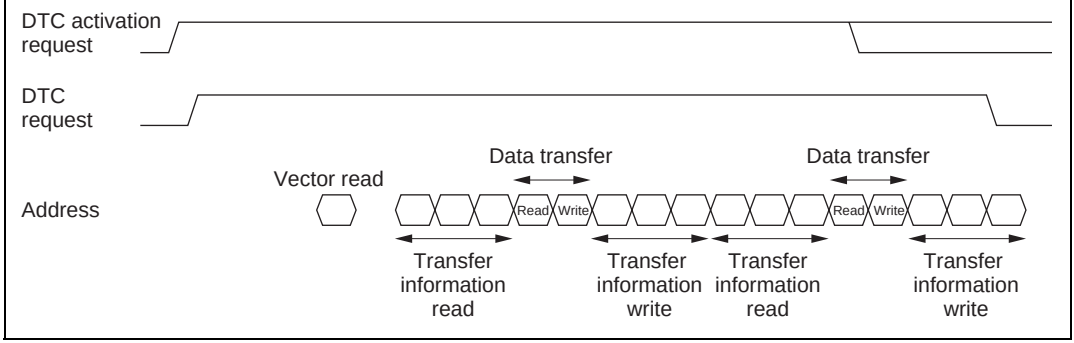


Figure 8.12 DTC Operation Timing (Example of Chain Transfer)

8.3.10 Number of DTC Execution States

Table 8.9 lists execution phases for a single DTC data transfer, and table 8.10 shows the number of states required for each execution phase.

Table 8.9 DTC Execution Phases

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

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

Access To:			On-Chip RAM	On-Chip ROM	Internal I/O Registers		External Devices			
Bus width			32	16	8	16	8	8	16	16
Access states			1	1	2	2	2	3	2	3
Execution phase	Vector read	S_I	—	1	—	—	4	6+2m	2	3+m
	Register information read/write	S_J	1	—	—	—	—	—	—	—
	Byte data read	S_K	1	1	2	2	2	3+m	2	3+m
	Word data read	S_K	1	1	4	2	4	6+2m	2	3+m
	Byte data write	S_L	1	1	2	2	2	3+m	2	3+m
	Word data write	S_L	1	1	4	2	4	6+2m	2	3+m
	Internal operation	S_M	1	1	1	1	1	1	1	1

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

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

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

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

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

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

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

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

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

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

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

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

[3] Check that the SWDTE bit is 0.

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

[5] Check the vector number written to DTVECR.

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

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

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

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

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

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

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

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

trigger cycle updating. Repeat mode transfer to the PPG's NDR is performed in the first half of the chain transfer, and normal mode transfer to the TPU's TGR in the second half. This is because clearing of the activation source and interrupt generation at the end of the specified number of transfers are restricted to the second half of the chain transfer (transfer when CHNE = 0).

[1] Perform settings for transfer to the PPG's NDR. Set MRA to source address incrementing (SM1 = 1, SM0 = 0), fixed destination address (DM1 = DM0 = 0), repeat mode (MD1 = 0, MD0 = 1), and word size (Sz = 1). Set the source side as a repeat area (DTS = 1). Set MRB to chain mode (CHNE = 1, DISEL = 0). Set the data table start address in SAR, the NDRH address in DAR, and the data table size in CRAH and CRAL. CRB can be set to any value.

[2] Perform settings for transfer to the TPU's TGR. Set MRA to source address incrementing (SM1 = 1, SM0 = 0), fixed destination address (DM1 = DM0 = 0), normal mode (MD1 = MD0 = 0), and word size (Sz = 1). Set the data table start address in SAR, the TGRA address 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 transfer register information.

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

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

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

[7] Set the initial output value in PODR, and the next output value in NDR. Set bits in DDR and NDER for which output is to be performed to 1. Using PCR, select the TPU compare match 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 NDR and the set value of the next output trigger period is transferred to TGRA. The activation source TGFA flag is cleared.

[10] When the specified number of transfers are completed (the TPU transfer CRA value is 0), the TGFA flag is held at 1, the DTCE bit is cleared to 0, and a TGIA interrupt request is sent to the CPU. Wrap-up processing should be performed in the interrupt handling routine.

more repeat transfers.

An example is shown in which a 128-kbyte input buffer is configured. The input buffer is assumed to have been set to start at lower address H'0000. Figure 8.13 shows the memory map.

[1] For the first transfer, set the normal mode for input data. Set fixed transfer source address (G/A, etc.), CRA = H'0000 (64k times), and CHNE = 1, CHNS = 1, and DISEL = 0.

[2] Prepare the upper 8-bit addresses of the start addresses for each of the 64k transfer start addresses for the first data transfer in a separate area (in ROM, etc.). For example, if the input buffer comprises H'200000 to H'21FFFF, prepare H'21 and H'20.

[3] For the second transfer, set repeat mode (with the source side as the repeat area) for re-setting the transfer destination address for the first data transfer. Use the upper 8 bits of DAR in the first register information area as the transfer destination. Set CHNE = DISEL = 0. If the above input buffer is specified as H'200000 to H'21FFFF, set the transfer counter to 2.

[4] Execute the first data transfer 64k times by means of interrupts. When the transfer counter for the first data transfer reaches 0, the second data transfer is started. Set the upper 8 bits of the transfer source address for the first data transfer to H'21. The lower 16 bits of the transfer destination address of the first data transfer and the transfer counter are H'0000.

[5] Next, execute the first data transfer the 64k times specified for the first data transfer by means of interrupts. When the transfer counter for the first data transfer reaches 0, the second data transfer is started. Set the upper 8 bits of the transfer source address for the first data transfer to H'20. The lower 16 bits of the transfer destination address of the first data transfer and the transfer counter are H'0000.

[6] Steps [4] and [5] are repeated endlessly. As repeat mode is specified for the second data transfer, an interrupt request is not sent to the CPU.

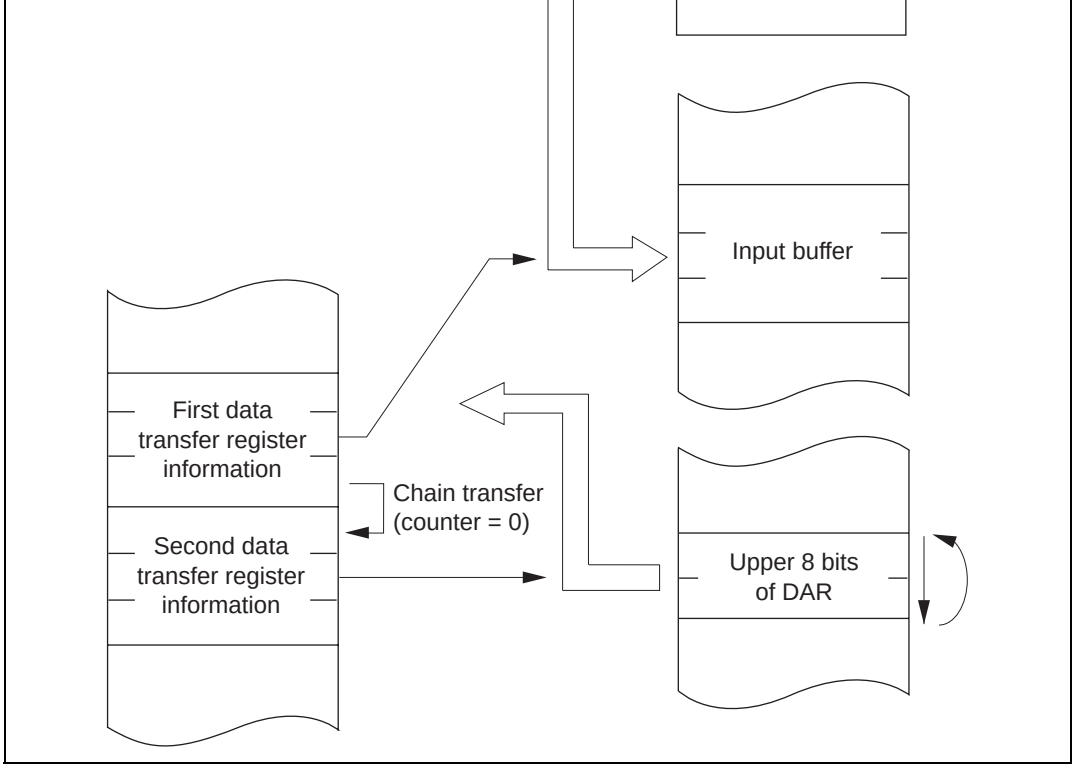


Figure 8.13 Chain Transfer when Counter = 0

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

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

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

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

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

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

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

8.4 Interrupts

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

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

When the DISEL bit is 1 and one data transfer has ended, or the specified number of transfers have ended, after data transfer ends, the SWDTE bit is held at 1 and an SWDTEND interrupt is 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 during a data transfer wait or during data transfer even if the SWDTE bit is set to 1.

Module Stop: When the MSTP14 bit in MSTPCR is set to 1, the DTC clock stops, and the DTC enters the module stop state. However, 1 cannot be written to the MSTP14 bit while the DTC is operating.

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

DMAC Transfer End Interrupt: When DTC transfer is activated by a DMAC transfer end interrupt, regardless of the transfer counter and DISEL bit, the DMAC's DTE bit is not subject to DTC control, and the write data has priority. Consequently, an interrupt request may not be sent to the CPU when the DTC transfer counter reaches 0.

DTCE Bit Setting: For DTCE bit setting, read/write operations must be performed using bit-manipulation instructions such as BSET and BCLR. For the initial setting only, however, when multiple activation sources are set at one time, it is possible to disable interrupts and write after executing a dummy read on the relevant register.

Chain Transfer: When chain transfer is used, clearing of the activation source or DTCER is performed when the last of the chain of data transfers is executed. SCI and A/D converter interrupt/activation sources, on the other hand, are cleared when the DTC reads or writes to the prescribed register.

Therefore, when the DTC is activated by an interrupt or activation source, if a read/write of the relevant register is not included in the last chained data transfer, the interrupt or activation source will be retained.

9.1 Overview

The chip has 15 I/O ports (ports 1 to 3, $P5_0$ to $P5_3$, 6 to 9, and A to G), and two input-only ports (port 4 and $P5_4$ to $P5_7$).

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

Each port includes a data direction register (DDR) that controls input/output (not provided for the input-only ports), a data register (DR) that stores output data, and a port register (PORT) used to read the pin states.

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

Port 3 and port A include an open drain control register (ODR) that controls the on/off state of the output buffer PMOS.

Ports 1 and A to F can drive a single TTL load and 50-pF capacitive load, and ports 2, 3, 5 to 9, and G can drive a single TTL load and 30-pF capacitive load.

Ports 1, 2, 7, and 9, and pins 5_0 to 5_3 (only when used as IRQ inputs), 6_4 to 6_7, and A_4 to A_7, are Schmitt-triggered inputs.

Port	Description	Pins	Mode 4	Mode 5	Mode 6	Mode 7
Port 1	• 8-bit I/O port • Schmitt-triggered input	$P1_7/PO_{15}/TIOCB_2/TCLKD$ $P1_6/PO_{14}/TIOCA_2$ $P1_5/PO_{13}/TIOCB_1/TCLKC$ $P14/PO12/TIOCA_1$ $P1_3/PO_{11}/TIOCD_0/TCLKB$ $P1_2/PO_{10}/TIOCC_0/TCLKA$ $P1_1/PO_9/TIOCB_0$ $P1_0/PO_8/TIOCA_0$	8-bit I/O port also functioning as TPU I/O pins (TCLKA, TCLKB, TCLKC, TCLKD, $TIOCA_0$, $TIOCB_0$, $TIOCC_0$, $TIOCD_0$, $TIOCA_1$, $TIOCB_1$, $TIOCA_2$, $TIOCB_2$) and PPG output pins (PO_{15} to PO_8)			
Port 2	• 8-bit I/O port • Schmitt-triggered input	$P2_7/PO_7/TIOCB_5$ $P2_6/PO_6/TIOCA_5$ $P2_5/PO_5/TIOCB_4$ $P2_4/PO_4/TIOCA_4$ $P2_3/PO_3/TIOCD_3$ $P2_2/PO_2/TIOCC_3$ $P2_1/PO_1/TIOCB_3$ $P2_0/PO_0/TIOCA_3$	8-bit I/O port also functioning as TPU I/O pins ($TIOCA_3$, $TIOCB_3$, $TIOCC_3$, $TIOCD_3$, $TIOCA_4$, $TIOCB_4$, $TIOCA_5$, $TIOCB_5$) and PPG output pins (PO_7 to PO_0)			
Port 3	• 6-bit I/O port • Open-drain output capability	$P3_5/SCK_1$ $P3_4/SCK_0$ $P3_3/RxD_1$ $P3_2/RxD_0$ $P3_1/TxD_1$ $P3_0/TxD_0$	6-bit I/O port also functioning as SCI (channel 0 and 1) I/O pins (TxD_0, RxD_0, SCK_0, TxD_1, RxD_1, SCK_1)			
Port 4	• 8-bit input port	$P4_7/AN_7/DA_1$ $P4_6/AN_6/DA_0$ $P4_5/AN_5$ $P4_4/AN_4$ $P4_3/AN_3$ $P4_2/AN_2$ $P4_1/AN_1$ $P4_0/AN_0$	8-bit input port also functioning as A/D converter analog inputs (AN_7 to AN_0) and D/A converter analog outputs (DA_1 and DA_0)			

	port • 4-bit input port • Schmitt-triggered input ($\overline{IRQ}$ input only)	$P5_6/AN_{14}/DA_2$ $P5_5/AN_{13}$ $P5_4/AN_{12}$	converter analog inputs (AN_{15} to AN_{12}) and D/A converter analog outputs (DA_3 and DA_2)	
		$P5_3/\overline{ADTRG}/\overline{IRQ_7}/\overline{WAIT}/\overline{BREQO}$	I/O port also functioning as A/D converter input pin ($\overline{ADTRG}$), and as interrupt input pin ($\overline{IRQ_7}$) when IRQPAS = 1, $\overline{WAIT}$ input pin when WAITE =1, BREQOE =0, WAITPS = 1, DDR =0, and $\overline{BREQO}$ output pin when WAITE =0, BREQOE = 1, BREQOPS = 1	I/O port also functioning as A/D converter input pin ($\overline{ADTRG}$), and as interrupt input pin ($\overline{IRQ_7}$) when IRQPAS = 1
		$P5_2/SCK_2/\overline{IRQ_6}$ $P5_1/RxD_2/\overline{IRQ_5}$ $P5_0/TxD_2/\overline{IRQ_4}$	I/O port also functioning as SCI (channel 2) I/O pins (TxD_2, RxD_2, SCK_2), and as interrupt input pins ($\overline{IRQ_4}$ to $\overline{IRQ_6}$) when IRQPAS = 1	
Port 6	• 8-bit I/O port • Schmitt-triggered input ($P6_4$ to $P6_7$)	$P6_7/\overline{CS_7}$ $P6_6/\overline{CS_6}$ $P6_5/\overline{IRQ_1}$ $P6_4/\overline{IRQ_0}$ $P6_3$ $P6_2$ $P6_1/\overline{CS_5}$ $P6_0/\overline{CS_4}$	8-bit I/O port also functioning as bus control output pins ($\overline{CS_4}$ to $\overline{CS_7}$), and interrupt input pins ($\overline{IRQ_0}$ and $\overline{IRQ_1}$)	8-bit I/O port also functioning as interrupt input pins ($\overline{IRQ_0}$ and $\overline{IRQ_1}$)
Port 7	• 6-bit I/O port • Schmitt-triggered input	$P7_5/TMO_1$ $P7_4/TMO_0$ $P7_3/TMCI_1$ $P7_2/TMCI_0$ $P7_1/TMRI_1$ $P7_0/TMRI_0$	6-bit I/O port also functioning as 8-bit timer (channels 0 and 1) I/O pins ($TMRI_0$, $TMCI_0$, TMO_0, $TMRI_1$, $TMCI_1$, TMO_1)	
Port 8	• 7-bit I/O port	$P8_6/\overline{WAIT}$ $P8_5/\overline{DACK_1}$ $P8_4/\overline{DACK_0}$ $P8_3/\overline{TEND_1}$ $P8_2/\overline{TEND_0}$ $P8_1/\overline{DREQ_1}$ $P8_0/\overline{DREQ_0}$	7-bit I/O port also functioning as DMA controller I/O pins ($\overline{DREQ_0}$, $\overline{TEND_0}$, $\overline{DACK_0}$, $\overline{DREQ_1}$, $\overline{TEND_1}$, $\overline{DACK_1}$) and $\overline{WAIT}$ input when WAITPS = 0, DDR =0	

	port • Schmitt-triggered input	$P9_6/\overline{IRQ_6}$ $P9_5/\overline{IRQ_5}$ $P9_4/\overline{IRQ_4}$ $P9_3/\overline{IRQ_3}$ $P9_2/\overline{IRQ_2}$	$\overline{IRQ_2}$) when IRQPAS = 0		
Port A	• 8-bit I/O port • Built-in MOS input pull-up • Open-drain output capability • Schmitt-triggered input (PA_4 to PA_7)	PA_7/A_{23} PA_6/A_{22} PA_5/A_{21}	When DDR = 0 (after reset): input port When DDR = 1 and A23E to A21E = 1: address output When DDR = 1 and A23E to A21E = 0: DR value output	When DDR = 0 (after reset): input port When DDR = 1 and A23E to A20E = 1: address output When DDR = 1 and A23E to A20E = 0: DR value output	I/O port
		PA_4/A_{20}	Address output also functioning as output port		
		PA_3/A_{19} to PA_0/A_{16}	Address output	When DDR = 0 (after reset): input ports When DDR = 1: address output	I/O port
Port B	• 8-bit I/O port • Built-in MOS input pull-up	PB_7/A_{15} to PB_0/A_8	Address output	When DDR = 0 (after reset): input port When DDR = 1: address output	I/O port
Port C	• 8-bit I/O port • Built-in MOS input pull-up	PC_7/A_7 to PC_0/A_0	Address output	When DDR = 0 (after reset): input port When DDR = 1: address output	I/O port
Port D	• 8-bit I/O port • Built-in MOS input pull-up	PD_7/D_{15} to PD_0/D_8	Data bus input/output		I/O port

	port • Built-in MOS input pull-up		In 16-bit bus mode: data bus input/output	
Port F	• 8-bit I/O port	PF_7/ϕ	When DDR = 0: input port When DDR = 1 (after reset): ϕ output	When DDR = 0 (after reset): input port When DDR = 1: ϕ output
		$PF_6/\overline{AS}$	When ASOD = 1: I/O port When ASOD = 0: $\overline{AS}$ output	I/O port
		$PF_5/\overline{RD}$ $PF_4/\overline{HWR}$	$\overline{RD}$, $\overline{HWR}$ output	
		$PF_3/\overline{LWR}$	When LWROD = 1: I/O port When LWROD = 0: $\overline{LWR}$ output	
		$PF_2/\overline{LCAS}/\overline{BREQO}$	When BREQOE = 0 (after reset): I/O port When BREQOE = 1 and BREQOPS = 0: $\overline{BREQO}$ output When RMTS2 to RMTS0= B'001 to B'011, and 16-bit access space is set: $\overline{LCAS}$ output	
		$PF_1/\overline{BACK}$ $PF_0/\overline{BREQ}$	When BRLE = 0 (after reset): I/O port When BRLE = 1: $\overline{BREQ}$ input, $\overline{BACK}$ output	
Port G	• 5-bit I/O port	$PG_4/\overline{CS_0}$	When DDR = 0[*2]: input port When DDR = 1[*3]: $\overline{CS_0}$ output	I/O port
		$PG_3/\overline{CS_1}$	When DDR = 0 (after reset): input port When CS167E = 0 and DDR = 1: output port When CS167E = 1 and DDR = 1: $\overline{CS_1}$ output	
		$PG_2/\overline{CS_2}$	When DDR = 0 (after reset): input port When CS25E = 0 and DDR = 1: output port When CS25E = 1 and DDR = 1: $\overline{CS_2}$ output	
		$PG_1/\overline{CS_3}$	When DDR = 0 (after reset): input port When CS25E = 0 and DDR = 1: output port When CS25E = 1 and DDR = 1: $\overline{CS_3}$ output	
		$PG_0/\overline{CAS}$	DRAM space set: $\overline{CAS}$ output Otherwise (after reset): I/O port	

Notes: 1. Only modes 4 and 5 are provided in the ROMless version.
2. After a reset in mode 6
3. After a reset in mode 4 or 5

9.2.1 Overview

Port 1 is an 8-bit I/O port. Port 1 pins also function as PPG output pins (PO_{15} to PO_8) and TPU I/O pins (TCLKA, TCLKB, TCLKC, TCLKD, $TIOCA_0$, $TIOCB_0$, $TIOCC_0$, $TIOCD_0$, $TIOCA_1$, $TIOCB_1$, $TIOCA_2$, and $TIOCB_2$). Port 1 pin functions are the same in all operating modes. Port 1 uses Schmitt-triggered input.

Figure 9.1 shows the port 1 pin configuration.

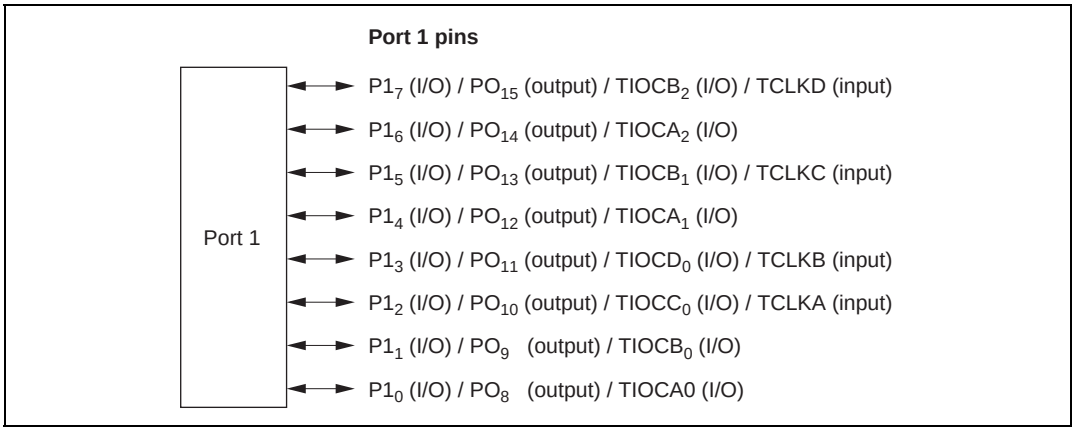


Figure 9.1 Port 1 Pin Functions

Table 9.2 shows the port 1 register configuration.

Table 9.2 Port 1 Registers

Name	Abbreviation	R/W	Initial Value	Address*
Port 1 data direction register	P1DDR	W	H'00	H'FEB0
Port 1 data register	P1DR	R/W	H'00	H'FF60
Port 1 register	PORT1	R	Undefined	H'FF50

Note: * Lower 16 bits of the address.

Port 1 Data Direction Register (P1DDR)

Bit	:	7	6	5	4	3	2	1	0
		P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DDR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W	W

P1DDR is an 8-bit write-only register, the individual bits of which specify input or output for the 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 clearing the bit 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 prior state in software standby mode.

Bit	:	7	6	5	4	3	2	1	0
		P17DR	P16DR	P15DR	P14DR	P13DR	P12DR	P11DR	P10DR
Initial value	:	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

P1DR is an 8-bit readable/writable register that stores output data for the port 1 pins (P17 to P10).

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

Port 1 Register (PORT1)

Bit	:	7	6	5	4	3	2	1	0
		P17	P16	P15	P14	P13	P12	P11	P10
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

Note: * Determined by state of pins $P1_7$ to $P1_0$.

PORT1 is an 8-bit read-only register that shows the pin states. It cannot be written to. Writing of output data for the port 1 pins ($P1_7$ to $P1_0$) 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. If a port 1 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 pin states, as P1DDR and P1DR are initialized. PORT1 retains its prior state in software standby mode.

Port 1 pins also function as PPG output pins (PO_{15} to PO_8) and TPU I/O pins (TCLKA, TCLKB, TCLKC, TCLKD, $TIOCA_0$, $TIOCB_0$, $TIOCC_0$, $TIOCD_0$, $TIOCA_1$, $TIOCB_1$, $TIOCA_2$, and $TIOCB_2$) Port 1 pin functions are shown in table 9.3.

Table 9.3 Port 1 Pin Functions

Pin	Selection Method and Pin Functions
$P1_7/PO_{15}/TIOCB_2/TCLKD$	The pin function is switched as shown below according to the combination of the TPU channel 2 setting (by bits MD3 to MD0 in TMDR2, bits IOB3 to IOB0 in TIOR2, and bits CCLR1 and CCLR0 in TCR2), bits TPSC2 to TPSC0 in TCR0 and TCR5, bit NDER15 in NDERH, and bit P17DDR.

TPU Channel 2 Setting	Table Below (1)	Table Below (2)		
P17DDR	—	0	1	1
NDER15	—	—	0	1
Pin function	$TIOCB_2$ output	$P1_7$ input	$P1_7$ output	PO_{15} output
		$TIOCB_2$ input *1		
	TCLKD input *2			

Notes: 1. $TIOCB_2$ input when MD3 to MD0 = B'0000 or B'01xx, and IOB3 = 1.
2. TCLKD input when the setting for either TCR0 or TCR5 is: TPSC2 to TPSC0 = B'111.
TCLKD input when channels 2 and 4 are set to phase counting mode.

TPU Channel 2 Setting	(2)	(1)	(2)	(2)	(1)	(2)
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 than B'xx00	
CCLR1, CCLR0	—	—	—	—	Other than B'10	B'10
Output function	—	Output compare output	—	—	PWM mode 2 output	—

x: Don't care

$TIOCA_2$ the TPU channel 2 setting (by bits MD3 to MD0 in TMDR2, bits IOA3 to IOA0 in TIOR2, and bits CCLR1 and CCLR0 in TCR2), bit NDER14 in NDERH, and bit P16DDR.

TPU Channel 2 Setting	Table Below (1)	Table Below (2)		
P16DDR	—	0	1	1
NDER14	—	—	0	1
Pin function	$TIOCA_2$ output	$P1_6$ input	$P1_6$ output	PO_{14} output
		$TIOCA_2$ input *1		

TPU Channel 2 Setting	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000, B'01xx		B'001x	B'0010	B'0011	
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		
CCLR1, CCLR0	—	—	—	—	Other than B'01	B'01
Output function	—	Output compare output	—	PWM mode 1 output *2	PWM mode 2 output	—

x: Don't care

Notes: 1. $TIOCA_2$ input when MD3 to MD0 = B'0000 or B'01xx, and IOA3 = 1.
2. $TIOCB_2$ output is disabled.

$TIOCB_1$/TCLKC the TPU channel 1 setting (by bits MD3 to MD0 in TMDR1, bits IOB3 to IOB0 in TIOR1, and bits CCLR1 and CCLR0 in TCR1), bits TPSC2 to TPSC0 in TCR0, TCR2, TCR4, and TCR5, bit NDER13 in NDERH, and bit P15DDR.

TPU Channel 1 Setting	Table Below (1)	Table Below (2)		
P15DDR	—	0	1	1
NDER13	—	—	0	1
Pin function	$TIOCB_1$ output	$P1_5$ input	$P1_5$ output	PO_{13} output
		$TIOCB_1$ input *1		
	TCLKC input *2			

Notes: 1. $TIOCB_1$ input when MD3 to MD0 = B'0000 or B'01xx, and IOB3 to IOB0 = B'10xx.
2. TCLKC input when the setting for either TCR0 or TCR2 is: TPSC2 to TPSC0 = B'110; or when the setting for either TCR4 or TCR5 is TPSC2 to TPSC0 = B'101.
TCLKC input when channels 2 and 4 are set to phase counting mode.

TPU Channel 1 Setting	(2)	(1)	(2)	(2)	(1)	(2)
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 than B'xx00	
CCLR1, CCLR0	—	—	—	—	Other than B'10	B'10
Output function	—	Output compare output	—	—	PWM mode 2 output	—

x: Don't care

$TIOCA_1$ the TPU channel 1 setting (by bits MD3 to MD0 in TMDR1, bits IOA3 to IOA0 in TIOR1, and bits CCLR1 and CCLR0 in TCR1), bit NDER12 in NDERH, and bit P14DDR.

TPU Channel 1 Setting	Table Below (1)	Table Below (2)		
P14DDR	—	0	1	1
NDER12	—	—	0	1
Pin function	$TIOCA_1$ output	$P1_4$ input	$P1_4$ output	PO_{12} output
		$TIOCA_1$ input *1		

TPU Channel 1 Setting	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000, B'01xx		B'001x	B'0010	B'0011	
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		
CCLR1, CCLR0	—	—	—	—	Other than B'01	B'01
Output function	—	Output compare output	—	PWM mode 1 output*2	PWM mode 2 output	—

x: Don't care

Notes: 1. $TIOCA_1$ input when MD3 to MD0 = B'0000 or B'01xx, and IOA3 to IOA0 = B'10xx.

2. $TIOCB_1$ output is disabled.

$TIOCD_0$/TCLKB the TPU channel 0 setting (by bits MD3 to MD0 in TMDR0, bits IOD3 to IOD0 in TIOR0L, and bits CCLR2 to CCLR0 in TCR0), bits TPSC2 to TPSC0 in TCR0 to TCR2, bit NDER11 in NDERH, and bit P13DDR.

TPU Channel 0 Setting	Table Below (1)	Table Below (2)		
P13DDR	—	0	1	1
NDER11	—	—	0	1
Pin function	$TIOCD_0$ output	$P1_3$ input	$P1_3$ output	PO_{11} output
		$TIOCD_0$ input *1		
	TCLKB input *2			

Notes: 1. $TIOCD_0$ input when MD3 to MD0 = B'0000, and IOD3 to IOD0 = B'10xx.
2. TCLKB input when the setting for TCR0 to TCR2 is: TPSC2 to TPSC0 = B'101.
TCLKB input when channels 1 and 5 are set to phase counting mode.

TPU Channel 0 Setting	(2)	(1)	(2)	(2)	(1)	(2)
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	Other than B'xx00	
CCLR2 to CCLR0	—	—	—	—	Other than B'110	B'110
Output function	—	Output compare output	—	—	PWM mode 2 output	—

x: Don't care

TIOCC$_0$/TCLKA

the TPU channel 0 setting (by bits MD3 to MD0 in TMDR0, bits IOC3 to IOC0 in TIOR0L, and bits CCLR2 to CCLR0 in TCR0), bits TPSC2 to TPSC0 in TCR0 to TCR5, bit NDER10 in NDERH, and bit P12DDR.

TPU Channel 0 Setting	Table Below (1)	Table Below (2)		
P12DDR	—	0	1	1
NDER10	—	—	0	1
Pin function	TIOCC$_0$ output	P1$_2$ input	P1$_2$ output	PO$_{10}$ output
		TIOCC$_0$ input *1		
	TCLKA input *2			

TPU Channel 0 Setting	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000		B'001x	B'0010	B'0011	
IOC3 to IOC0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'xx00		
CCLR2 to CCLR0	—	—	—	—	Other than B'101	B'101
Output function	—	Output compare output	—	PWM mode 1 output*3	PWM mode 2 output	—

x: Don't care

Notes: 1. TIOCC$_0$ input when MD3 to MD0 = B'0000, and IOC3 to IOC0 = B'10xx.

2. TCLKA input when the setting for TCR0 to TCR5 is: TPSC2 to TPSC0 = B'100.
   TCLKA input when channels 1 and 5 are set to phase counting mode.

3. TIOCD$_0$ output is disabled.
   When BFA = 1 or BFB = 1 in TMDR0, output is disabled and setting (2) applies.

the TPU channel 0 setting (by bits MD3 to MD0 in TMDR0, bits IOB3 to IOB0 in TIOR0H, and bits CCLR2 to CCLR0 in TCR0), bit NDER9 in NDERH, and bit P11DDR.

TPU Channel 0 Setting	Table Below (1)	Table Below (2)		
P11DDR	—	0	1	1
NDER9	—	—	0	1
Pin function	$TIOCB_0$ output	$P1_1$ input	$P1_1$ output	PO_9 output
		$TIOCB_0$ input *		

Note: * $TIOCB_0$ input when MD3 to MD0 = B'0000, and IOB3 to IOB0 = B'10xx.

TPU Channel 0 Setting	(2)	(1)	(2)	(2)	(1)	(2)
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	Other than B'xx00	
CCLR2 to CCLR0	—	—	—	—	Other than B'010	B'010
Output function	—	Output compare output	—	—	PWM mode 2 output	—

x: Don't care

the TPU channel 0 setting (by bits MD3 to MD0 in TMDR0, bits IOA3 to IOA0 in TIOR0H, and bits CCLR2 to CCLR0 in TCR0), bit NDER8 in NDERH, and bit P10DDR.

TPU Channel 0 Setting	Table Below (1)	Table Below (2)		
P10DDR	—	0	1	1
NDER8	—	—	0	1
Pin function	$TIOCA_0$ output	$P1_0$ input	$P1_0$ output	PO_8 output
		$TIOCA_0$ input *1		

TPU Channel 0 Setting	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000		B'001x	B'0010	B'0011	
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		
CCLR2 to CCLR0	—	—	—	—	Other than B'001	B'001
Output function	—	Output compare output	—	PWM mode 1 output*2	PWM mode 2 output	—

x: Don't care

Notes: 1. $TIOCA_0$ input when MD3 to MD0 = B'0000, and IOA3 to IOA0 = B'10xx.

2. $TIOCB_0$ output is disabled.

9.3.1 Overview

Port 2 is an 8-bit I/O port. Port 2 pins also function as PPG output pins (PO_7 to PO_0) and TPU I/O pins ($TIOCA_3$, $TIOCB_3$, $TIOCC_3$, $TIOCD_3$, $TIOCA_4$, $TIOCB_4$, $TIOCA_5$, and $TIOCB_5$). Port 2 pin functions are the same in all operating modes. Port 2 uses Schmitt-triggered input.

Figure 9.2 shows the port 2 pin configuration.

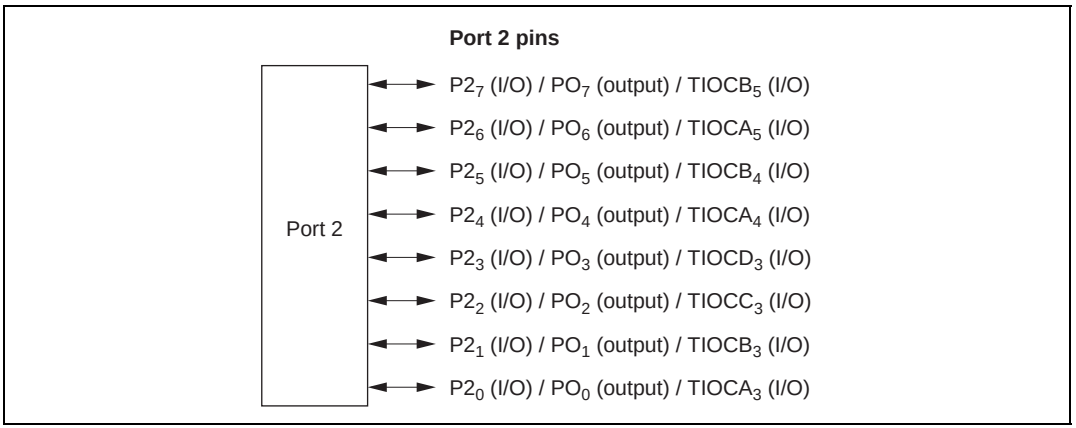


Figure 9.2 Port 2 Pin Functions

Table 9.4 shows the port 2 register configuration.

Table 9.4 Port 2 Registers

Name	Abbreviation	R/W	Initial Value	Address*
Port 2 data direction register	P2DDR	W	H'00	H'FEB1
Port 2 data register	P2DR	R/W	H'00	H'FF61
Port 2 register	PORT2	R	Undefined	H'FF51

Note: * Lower 16 bits of the address.

Port 2 Data Direction Register (P2DDR)

Bit	:	7	6	5	4	3	2	1	0
		P27DDR	P26DDR	P25DDR	P24DDR	P23DDR	P22DDR	P21DDR	P20DDR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W	W

P2DDR is an 8-bit write-only register, the individual bits of which specify input or output for the pins of port 2. P2DDR cannot be read; if it is, an undefined value will be read.

Setting a P2DDR bit to 1 makes the corresponding port 2 pin an output pin, while clearing the bit to 0 makes the pin an input pin.

P2DDR is initialized to H'00 by a reset, and in hardware standby mode. It retains its prior state in software standby mode.

Bit	:	7	6	5	4	3	2	1	0
		P27DR	P26DR	P25DR	P24DR	P23DR	P22DR	P21DR	P20DR
Initial value	:	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

P2DR is an 8-bit readable/writable register that stores output data for the port 2 pins (P27 to P20).

P2DR is initialized to H'00 by a reset, and in hardware standby mode. It retains its prior state in software standby mode.

Port 2 Register (PORT2)

Bit	:	7	6	5	4	3	2	1	0
		P27	P26	P25	P24	P23	P22	P21	P20
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

Note: * Determined by state of pins $P2_7$ to $P2_0$.

PORT2 is an 8-bit read-only register that shows the pin states. It cannot be written to. Writing of output data for the port 2 pins ($P2_7$ to $P2_0$) must always be performed on P2DR.

If a port 2 read is performed while P2DDR bits are set to 1, the P2DR values are read. If a port 2 read is performed while P2DDR bits are cleared to 0, the pin states are read.

After a reset and in hardware standby mode, PORT2 contents are determined by the pin states, as P2DDR and P2DR are initialized. PORT2 retains its prior state in software standby mode.

Port 2 pins also function as PPG output pins (PO_7 to PO_0) and TPU I/O pins ($TIOCA_3$, $TIOCB_3$, $TIOCC_3$, $TIOCD_3$, $TIOCA_4$, $TIOCB_4$, $TIOCA_5$, and $TIOCB_5$). Port 2 pin functions are shown in table 9.5.

Table 9.5 Port 2 Pin Functions

Pin	Selection Method and Pin Functions
$P2_7/PO_7/TIOCB_5$	The pin function is switched as shown below according to the combination of the TPU channel 5 setting (by bits MD3 to MD0 in TMDR5, bits IOB3 to IOB0 in TIOR5, and bits CCLR1 and CCLR0 in TCR5), bit NDER7 in NDERL, and bit P27DDR.

TPU Channel 5 Setting	Table Below (1)	Table Below (2)		
P27DDR	—	0	1	1
NDER7	—	—	0	1
Pin function	$TIOCB_5$ output	$P2_7$ input	$P2_7$ output	PO_7 output
		$TIOCB_5$ input *		

Note: * $TIOCB_5$ input when MD3 to MD0 = B'0000 or B'01xx, and IOB3 = 1.

TPU Channel 5 Setting	(2)	(1)	(2)	(2)	(1)	(2)
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 than B'xx00	
CCLR1, CCLR0	—	—	—	—	Other than B'10	B'10
Output function	—	Output compare output	—	—	PWM mode 2 output	—

x: Don't care

the TPU channel 5 setting (by bits MD3 to MD0 in TMDR5, bits IOA3 to IOA0 in TIOR5, and bits CCLR1 and CCLR0 in TCR5), bit NDER6 in NDERL, and bit P26DDR.

TPU Channel 5 Setting	Table Below (1)	Table Below (2)		
P26DDR	—	0	1	1
NDER6	—	—	0	1
Pin function	$TIOCA_5$ output	$P2_6$ input	$P2_6$ output	PO_6 output
		$TIOCA_5$ input *1		

TPU Channel 5 Setting	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000, B'01xx		B'001x	B'0010	B'0011	
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		
CCLR1, CCLR0	—	—	—	—	Other than B'01	B'01
Output function	—	Output compare output	—	PWM mode 1 output*2	PWM mode 2 output	—

x: Don't care

Notes: 1. $TIOCA_5$ input when MD3 to MD0 = B'0000 or B'01xx, and IOA3 = 1.
2. $TIOCB_5$ output is disabled.

the TPU channel 4 setting (by bits MD3 to MD0 in TMDR4, bits IOB3 to IOB0 in TIOR4, and bits CCLR1 and CCLR0 in TCR4), bit NDER5 in NDERL, and bit P25DDR.

TPU Channel 4 Setting	Table Below (1)	Table Below (2)		
P25DDR	—	0	1	1
NDER5	—	—	0	1
Pin function	$TIOCB_4$ output	$P2_5$ input	$P2_5$ output	PO_5 output
		$TIOCB_4$ input *		

Note: * $TIOCB_4$ input when MD3 to MD0 = B'0000 or B'01xx, and IOB3 to IOB0 = B'10xx.

TPU Channel 4 Setting	(2)	(1)	(2)	(2)	(1)	(2)
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 than B'xx00	
CCLR1, CCLR0	—	—	—	—	Other than B'10	B'10
Output function	—	Output compare output	—	—	PWM mode 2 output	—

x: Don't care

the TPU channel 4 setting (by bits MD3 to MD0 in TMDR4, bits IOA3 to IOA0 in TIOR4, and bits CCLR1 and CCLR0 in TCR4), bit NDER4 in NDERL, and bit P24DDR.

TPU Channel 4 Setting	Table Below (1)	Table Below (2)		
P24DDR	—	0	1	1
NDER4	—	—	0	1
Pin function	$TIOCA_4$ output	$P2_4$ input	$P2_4$ output	PO_4 output
		$TIOCA_4$ input *1		

TPU Channel 4 Setting	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000, B'01xx		B'001x	B'0010	B'0011	
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		
CCLR1, CCLR0	—	—	—	—	Other than B'01	B'01
Output function	—	Output compare output	—	PWM mode 1 output*2	PWM mode 2 output	—

x: Don't care

Notes: 1. $TIOCA_4$ input when MD3 to MD0 = B'0000 or B'01xx, and IOA3 to IOA0 = B'10xx.

2. $TIOCB_4$ output is disabled.

the TPU channel 3 setting (by bits MD3 to MD0 in TMDR3, bits IOD3 to IOD0 in TIOR3L, and bits CCLR2 to CCLR0 in TCR3), bit NDER3 in NDERL, and bit P23DDR.

TPU Channel 3 Setting	Table Below (1)	Table Below (2)		
P23DDR	—	0	1	1
NDER3	—	—	0	1
Pin function	$TIOCD_3$ output	$P2_3$ input	$P2_3$ output	PO_3 output
		$TIOCD_3$ input *		

Note: * $TIOCD_3$ input when MD3 to MD0 = B'0000, and IOD3 to IOD0 = B'10xx.

TPU Channel 3 Setting	(2)	(1)	(2)	(2)	(1)	(2)
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	Other than B'xx00	
CCLR2 to CCLR0	—	—	—	—	Other than B'110	B'110
Output function	—	Output compare output	—	—	PWM mode 2 output	—

x: Don't care

the TPU channel 3 setting (by bits MD3 to MD0 in TMDR3, bits IOC3 to IOC0 in TIOR3L, and bits CCLR2 to CCLR0 in TCR3), bit NDER2 in NDERL, and bit P22DDR.

TPU Channel 3 Setting	Table Below (1)	Table Below (2)		
P22DDR	—	0	1	1
NDER2	—	—	0	1
Pin function	$TIOCC_3$ output	$P2_2$ input	$P2_2$ output	PO_2 output
		$TIOCC_3$ input *1		

TPU Channel 3 Setting	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000		B'001x	B'0010	B'0011	
IOC3 to IOC0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'xx00		
CCLR2 to CCLR0	—	—	—	—	Other than B'101	B'101
Output function	—	Output compare output	—	PWM mode 1 output*2	PWM mode 2 output	—

x: Don't care

Notes: 1. $TIOCC_3$ input when MD3 to MD0 = B'0000, and IOC3 to IOC0 = B'10xx.

2. $TIOCD_3$ output is disabled.
When BFA = 1 or BFB = 1 in TMDR3, output is disabled and setting (2) applies.

the TPU channel 3 setting (by bits MD3 to MD0 in TMDR3, bits IOB3 to IOB0 in TIOR3H, and bits CCLR2 to CCLR0 in TCR3), bit NDER1 in NDERL, and bit P21DDR.

TPU Channel 3 Setting	Table Below (1)	Table Below (2)		
P21DDR	—	0	1	1
NDER1	—	—	0	1
Pin function	$TIOCB_3$ output	$P2_1$ input	$P2_1$ output	PO_1 output
		$TIOCB_3$ input *		

Note: * $TIOCB_3$ input when MD3 to MD0 = B'0000, and IOB3 to IOB0 = B'10xx.

TPU Channel 3 Setting	(2)	(1)	(2)	(2)	(1)	(2)
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	Other than B'xx00	
CCLR2 to CCLR0	—	—	—	—	Other than B'010	B'010
Output function	—	Output compare output	—	—	PWM mode 2 output	—

x: Don't care

the TPU channel 3 setting (by bits MD3 to MD0 in TMDR3, bits IOA3 to IOA0 in TIOR3H, and bits CCLR2 to CCLR0 in TCR3), bit NDER0 in NDERL, and bit P20DDR.

TPU Channel 3 Setting	Table Below (1)	Table Below (2)		
P20DDR	—	0	1	1
NDER0	—	—	0	1
Pin function	$TIOCA_3$ output	$P2_0$ input	$P2_0$ output	PO_0 output
		$TIOCA_3$ input *1		

TPU Channel 3 Setting	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000		B'001x	B'0010	B'0011	
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		
CCLR2 to CCLR0	—	—	—	—	Other than B'001	B'001
Output function	—	Output compare output	—	PWM mode 1 output*2	PWM mode 2 output	—

x: Don't care

Notes: 1. $TIOCA_3$ input when MD3 to MD0 = B'0000, and IOA3 to IOA0 = B'10xx.

2. $TIOCB_3$ output is disabled.

9.4.1 Overview

Port 3 is a 6-bit I/O port. Port 3 pins also function as SCI I/O pins (TxD_0, RxD_0, SCK_0, TxD_1, RxD_1, and SCK_1). Port 3 pin functions are the same in all operating modes.

Figure 9.3 shows the port 3 pin configuration.

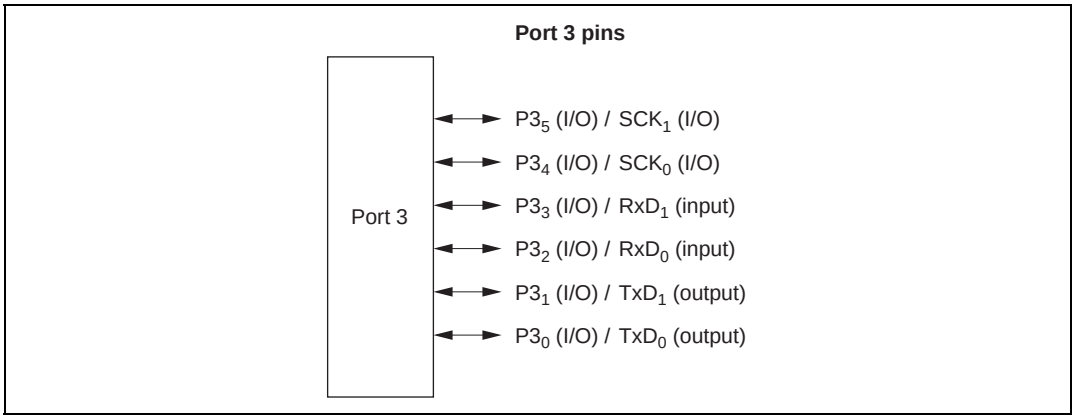


Figure 9.3 Port 3 Pin Functions

9.4.2 Register Configuration

Table 9.6 shows the port 3 register configuration.

Table 9.6 Port 3 Registers

Name	Abbreviation	R/W	Initial Value*2	Address*1
Port 3 data direction register	P3DDR	W	H'00	H'FEB2
Port 3 data register	P3DR	R/W	H'00	H'FF62
Port 3 register	PORT3	R	Undefined	H'FF52
Port 3 open drain control register	P3ODR	R/W	H'00	H'FF76

Notes: 1. Lower 16 bits of the address.
2. Value of bits 5 to 0.

Bit	:	7	6	5	4	3	2	1	0
		—	—	P35DDR	P34DDR	P33DDR	P32DDR	P31DDR	P30DDR
Initial value	:	Undefined	Undefined	0	0	0	0	0	0
R/W	:	—	—	W	W	W	W	W	W

P3DDR is an 8-bit write-only register, the individual bits of which specify input or output for the pins of port 3. Bits 7 and 6 are reserved; they return an undefined value if read, and cannot be modified.

Setting a P3DDR bit to 1 makes the corresponding port 3 pin an output pin, while clearing the bit to 0 makes the pin an input pin.

P3DDR is initialized to H'00 (bits 5 to 0) by a reset, and in hardware standby mode. It retains its prior state in software standby mode. As the SCI is initialized, the pin states are determined by the P3DDR and P3DR specifications.

Port 3 Data Register (P3DR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	P35DR	P34DR	P33DR	P32DR	P31DR	P30DR
Initial value	:	Undefined	Undefined	0	0	0	0	0	0
R/W	:	—	—	R/W	R/W	R/W	R/W	R/W	R/W

P3DR is an 8-bit readable/writable register that stores output data for the port 3 pins ($P3_5$ to $P3_0$).

Bits 7 and 6 are reserved; they return an undefined value if read, and cannot be modified.

P3DR is initialized to H'00 (bits 5 to 0) by a reset, and in hardware standby mode. It retains its prior state in software standby mode.

Bit	:	7	6	5	4	3	2	1	0
		—	—	P35	P34	P33	P32	P31	P30
Initial value	:	Undefined	Undefined	—*	—*	—*	—*	—*	—*
R/W	:	—	—	R	R	R	R	R	R

Note: * Determined by state of pins $P3_5$ to $P3_0$.

PORT3 is an 8-bit read-only register that shows the pin states. Writing of output data for the port 3 pins ($P3_5$ to $P3_0$) must always be performed on P3DR.

Bits 7 and 6 are reserved; they return an undefined value if read, and cannot be modified.

If a port 3 read is performed while P3DDR bits are set to 1, the P3DR values are read. If a port 3 read is performed while P3DDR bits are cleared to 0, the pin states are read.

After a reset and in hardware standby mode, PORT3 contents are determined by the pin states, as P3DDR and P3DR are initialized. PORT3 retains its prior state in software standby mode.

Port 3 Open Drain Control Register (P3ODR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	P35ODR	P34ODR	P33ODR	P32ODR	P31ODR	P30ODR
Initial value	:	Undefined	Undefined	0	0	0	0	0	0
R/W	:	—	—	R/W	R/W	R/W	R/W	R/W	R/W

P3ODR is an 8-bit readable/writable register that controls the PMOS on/off status for each port 3 pin ($P3_5$ to $P3_0$).

Bits 7 and 6 are reserved; they return an undefined value if read, and cannot be modified.

Setting a P3ODR bit to 1 makes the corresponding port 3 pin an NMOS open-drain output pin, while clearing the bit to 0 makes the pin a CMOS output pin.

P3ODR is initialized to H'00 (bits 5 to 0) by a reset, and in hardware standby mode. It retains its prior state in software standby mode.

Port 3 pins also function as SCI I/O pins (TxD_0, RxD_0, SCK_0, TxD_1, RxD_1, and SCK_1). Port 3 pin functions are shown in table 9.7.

Table 9.7 Port 3 Pin Functions

Pin	Selection Method and Pin Functions

$P3_5/SCK_1$

The pin function is switched as shown below according to the combination of bit C/$\overline{A}$ in the SCI1 SMR, bits CKE0 and CKE1 in SCR, and bit P35DDR.

CKE1	0				1
C/$\overline{A}$	0			1	—
CKE0	0		1	—	—
P35DDR	0	1	—	—	—
Pin function	$P3_5$ input pin	$P3_5$ output pin*	SCK_1 output pin*	SCK_1 output pin*	SCK_1 input pin

Note: * When P35ODR = 1, the pin becomes an NMOS open-drain output.

$P3_4/SCK_0$

The pin function is switched as shown below according to the combination of bit C/$\overline{A}$ in the SCI0 SMR, bits CKE0 and CKE1 in SCR, and bit P34DDR.

CKE1	0				1
C/$\overline{A}$	0			1	—
CKE0	0		1	—	—
P34DDR	0	1	—	—	—
Pin function	$P3_4$ input pin	$P3_4$ output pin*	SCK_0 output pin*	SCK_0 output pin*	SCK_0 input pin

Note: * When P34ODR = 1, the pin becomes an NMOS open-drain output.

bit RE in the SCI1 SCR, and bit P33DDR.

RE	0		1
P33DDR	0	1	—
Pin function	$P3_3$ input pin	$P3_3$ output pin*	RxD_1 input pin

Note: * When P33ODR = 1, the pin becomes an NMOS open-drain output.

$P3_2/RxD_0$

The pin function is switched as shown below according to the combination of bit RE in the SCI0 SCR, and bit P32DDR.

RE	0		1
P32DDR	0	1	—
Pin function	$P3_2$ input pin	$P3_2$ output pin*	RxD_0 input pin

Note: * When P32ODR = 1, the pin becomes an NMOS open-drain output.

$P3_1/TxD_1$

The pin function is switched as shown below according to the combination of bit TE in the SCI1 SCR, and bit P31DDR.

TE	0		1
P31DDR	0	1	—
Pin function	$P3_1$ input pin	$P3_1$ output pin*	TxD_1 output pin

Note: * When P31ODR = 1, the pin becomes an NMOS open-drain output.

$P3_0/TxD_0$

The pin function is switched as shown below according to the combination of bit TE in the SCI0 SCR, and bit P30DDR.

TE	0		1
P30DDR	0	1	—
Pin function	$P3_0$ input pin	$P3_0$ output pin*	TxD_0 output pin

Note: * When P30ODR = 1, the pin becomes an NMOS open-drain output.

9.5.1 Overview

Port 4 is an 8-bit input-only port. Port 4 pins also function as A/D converter analog input pins (AN_0 to AN_7) and D/A converter analog output pins (DA_0 and DA_1). Port 4 pin functions are the same in all operating modes. Figure 9.4 shows the port 4 pin configuration.

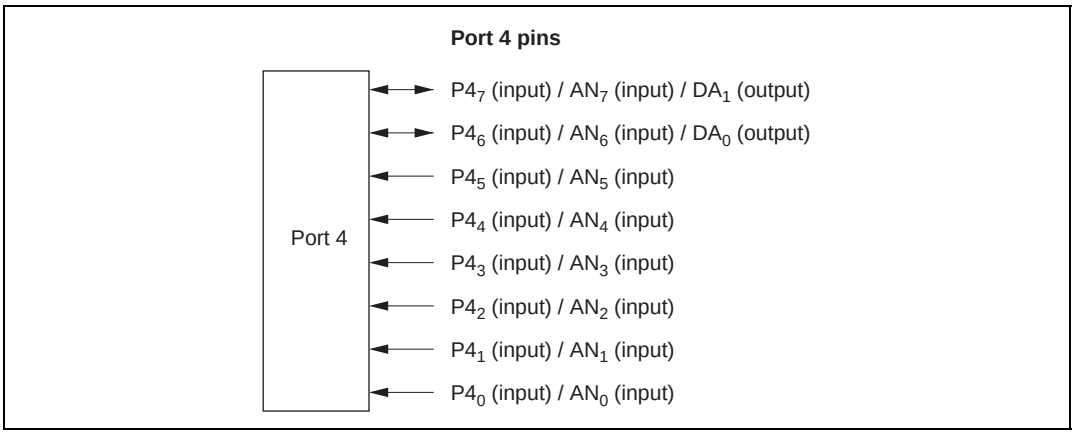


Figure 9.4 Port 4 Pin Functions

9.5.2 Register Configuration

Table 9.8 shows the port 4 register configuration. Port 4 is an input-only port, and does not have a data direction register or data register.

Table 9.8 Port 4 Register

Name	Abbreviation	R/W	Initial Value	Address*
Port 4 register	PORT4	R	Undefined	H'FF53

Note: * Lower 16 bits of the address.

Bit	:	7	6	5	4	3	2	1	0
		P47	P46	P45	P44	P43	P42	P41	P40
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

Note: * Determined by state of pins $P4_7$ to $P4_0$.

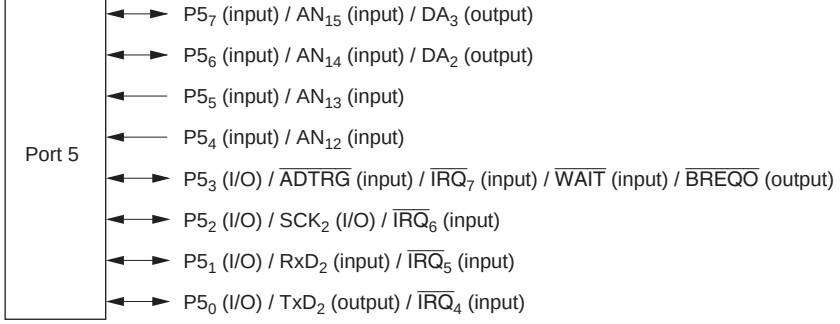
9.5.3 Pin Functions

Port 4 pins also function as A/D converter analog input pins (AN_0 to AN_7) and D/A converter analog output pins (DA_0 and DA_1).

9.6 Port 5

9.6.1 Overview

Port 5 comprises a 4-bit I/O port and a 4-bit input port. Port 5 pins also function as SCI I/O pins (TxD_2, RxD_2, and SCK_2), the A/D converter input pin ($\overline{ADTRG}$), A/D converter analog input pins (AN_{12} to AN_{15}), D/A converter analog output pins (DA_2 and DA_3), interrupt input pins ($\overline{IRQ}_4$ to $\overline{IRQ}_7$), and bus control signal I/O pins ($\overline{WAIT}$ and $\overline{BREQO}$). The pin functions can be switched by means of settings in PFCR2 and SYSCR. $\overline{IRQ}_4$ to $\overline{IRQ}_7$ are Schmitt-triggered inputs. Figure 9.5 shows the port 5 pin configuration.



Pin functions in modes 4 to 6

P5$_7$ (input) / AN$_{15}$ (input) / DA$_3$ (output)

P5$_6$ (input) / AN$_{14}$ (input) / DA$_2$ (output)

P5$_5$ (input) / AN$_{13}$ (input)

P5$_4$ (input) / AN$_{12}$ (input)

P5$_3$ (I/O) / $\overline{\text{ADTRG}}$ (input) / $\overline{\text{IRQ}}_7$ (input) / $\overline{\text{WAIT}}$ (input) / $\overline{\text{BREQO}}$ (output)

P5$_2$ (I/O) / SCK$_2$ (I/O) / $\overline{\text{IRQ}}_6$ (input)

P5$_1$ (I/O) / RxD$_2$ (input) / $\overline{\text{IRQ}}_6$ (input)

P5$_0$ (I/O) / TxD$_2$ (output) / $\overline{\text{IRQ}}_4$ (input)

Pin functions in mode 7

P5$_7$ (input) / AN$_{15}$ (input) / DA$_3$ (output)

P5$_6$ (input) / AN$_{14}$ (input) / DA$_2$ (output)

P5$_5$ (input) / AN$_{13}$ (input)

P5$_4$ (input) / AN$_{12}$ (input)

P5$_3$ (I/O) / $\overline{\text{ADTRG}}$ (intput) / $\overline{\text{IRQ}}_7$ (intput)

P5$_2$ (I/O) / SCK$_2$ (I/O) / $\overline{\text{IRQ}}_6$ (input)

P5$_1$ (I/O) / RxD$_2$ (input) / $\overline{\text{IRQ}}_5$ (input)

P5$_0$ (I/O) / TxD$_2$ (output) / $\overline{\text{IRQ}}_4$ (input)

Figure 9.5 Port 5 Pin Functions

Table 9.9 shows the port 5 register configuration.
Bits 7 to 4 of port 5 are input ports, and have no data direction register or data register.

Table 9.9 Port 5 Registers

Name	Abbreviation	R/W	Initial Value	Address*1
Port 5 data direction register	P5DDR	W	H'0*2	H'FEB4
Port 5 data register	P5DR	R/W	H'0*2	H'FF64
Port 5 register	PORT5	R	Undefined	H'FF54
Port function control register 2	PFCR2	R/W	H'30	H'FFAC
System control register	SYSCR	R/W	H'01	H'FF39

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

Port 5 Data Direction Register (P5DDR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	P53DDR	P52DDR	P51DDR	P50DDR
Initial value	:	Undefined	Undefined	Undefined	Undefined	0	0	0	0
R/W	:	—	—	—	—	W	W	W	W

P5DDR is an 8-bit write-only register, the individual bits of which specify input or output for the pins of port 5. Bits 7 to 4 are reserved. P5DDR cannot be read; if it is, an undefined value will be read.

Setting a P5DDR bit to 1 makes the corresponding port 5 pin an output pin, while clearing the bit to 0 makes the pin an input pin.

P5DDR is initialized to H'0 (bits 3 to 0) by a reset, and in hardware standby mode. It retains its prior state in software standby mode. As the SCI is initialized, the pin states are determined by the P5DDR and P5DR specifications.

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	P53DR	P52DR	P51DR	P50DR
Initial value	:	Undefined	Undefined	Undefined	Undefined	0	0	0	0
R/W	:	—	—	—	—	R/W	R/W	R/W	R/W

P5DR is an 8-bit readable/writable register that stores output data for the port 5 pins ($P5_3$ to $P5_0$).

Bits 7 to 4 are reserved; they return an undefined value if read, and cannot be modified.

P5DR is initialized to H'0 (bits 3 to 0) by a reset, and in hardware standby mode. It retains its prior state in software standby mode.

Port 5 Register (PORT5)

Bit	:	7	6	5	4	3	2	1	0
		P57	P56	P55	P54	P53	P52	P51	P50
Initial value	:	Undefined	Undefined	Undefined	Undefined	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

Note: * Determined by state of pins $P5_7$ to $P5_0$.

PORT5 is an 8-bit read-only register that shows the pin states. It cannot be written to. Writing of output data for the port 5 pins ($P5_3$ to $P5_0$) must always be performed on P5DR.

Bits 7 to 4 always return the pin states when a port 5 read is performed, without regard to P5DDR.

If a port 5 read is performed while P5DDR bits are set to 1, the P5DR values are read. If a port 5 read is performed while P5DDR bits are cleared to 0, the pin states are read.

After a reset and in hardware standby mode, PORT5 contents are determined by the pin states, as P5DDR and P5DR are initialized. PORT5 retains its prior state in software standby mode.

Bit	:	7	6	5	4	3	2	1	0
		WAITPS	BREQOPS	CS167E	CS25E	ASOD	—	—	—
Initial value	:	0	0	1	1	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R	R	R

PFCR2 is an 8-bit readable/writable register that performs I/O port control. PFCR2 is initialized to H'30 by a reset, and in hardware standby mode.

Bit 7—WAIT Pin Select (WAITPS): Selects the $\overline{\text{WAIT}}$ input pin. Set the WAITPS bit before setting the DDR bit clear to 0 and the WAITE bit in BCRL to 1.

Bit 7 WAITPS	Description	
0	$\overline{\text{WAIT}}$ input is $P8_6$ pin	(Initial value)
1	$\overline{\text{WAIT}}$ input is $P5_3$ pin	

Bit 6—BREQO Pin Select (BREQOPS): Selects the $\overline{\text{BREQO}}$ output pin. Set the BREQOPS bit before setting the BREQOE bit in BCRL to 1.

Bit 6 BREQOPS	Description	
0	$\overline{\text{BREQO}}$ output is PF_2 pin	(Initial value)
1	$\overline{\text{BREQO}}$ output is $P5_3$ pin	

Bit 5—CS167 Enable (CS167E): Enables or disables $\overline{\text{CS}}_1$, $\overline{\text{CS}}_6$, and $\overline{\text{CS}}_7$ output. For details, see section 9.7, Port 6, and section 9.17, Port G.

Bit 4—CS25 Enable (CS25E): Enables or disables $\overline{\text{CS}}_2$, $\overline{\text{CS}}_3$, $\overline{\text{CS}}_4$, and $\overline{\text{CS}}_5$ output. For details, see section 9.7, Port 6, and section 9.17, Port G.

Bit 3—AS Output Disable (ASOD): Enables or disables $\overline{\text{AS}}$ output. For details, see section 9.16, Port F.

Bit	:	7	6	5	4	3	2	1	0
		—	—	INTM1	INTM0	NMIEG	LWROD	IRQPAS	RAME
Initial value	:	0	0	0	0	0	0	0	1
R/W	:	R/W	—	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, controls the $\overline{\text{LWR}}$ pin, switches the $\overline{\text{IRQ}}_4$ to $\overline{\text{IRQ}}_7$ input pins, and selects the detected edge for NMI. SYSCR is initialized to H'01 by a reset, and in hardware standby mode. It is not initialized in software standby mode.

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

Bit 3—NMI Edge Select (NMIEG): Selects the input edge for the NMI pin. For details, see section 5, Interrupt Controller.

Bit 2—LWR Output Disable (LWROD): Enables or disables $\overline{\text{LWR}}$ output. For details, see section 9.16, Port F.

Bit 1—IRQ Port Switching Select (IRQPAS): Selects switching of input pins for $\overline{\text{IRQ}}_4$ to $\overline{\text{IRQ}}_7$. $\overline{\text{IRQ}}_4$ to $\overline{\text{IRQ}}_7$ input is always performed from one of the ports.

Bit 1 IRQPAS	Description	
0	$P9_4$ to $P9_7$ used for $\overline{\text{RQ}}_4$ to $\overline{\text{IRQ}}_7$ input	(Initial value)
1	$P5_3$ to $P5_0$ used for $\overline{\text{IRQ}}_4$ to $\overline{\text{IRQ}}_7$ input	

Bit 0—RAM Enable (RAME): Enables or disables on-chip RAM. For details, see section 18, RAM.

Port 5 pins also function as SCI I/O pins (TxD_2, RxD_2, and SCK_2), the A/D converter input pin ($\overline{ADTRG}$), interrupt input pins ($\overline{IRQ_4}$ to $\overline{IRQ_7}$), and bus control signal I/O pins ($\overline{WAIT}$ and $\overline{BREQO}$). Port 5 pins $P5_7$ to $P5_4$ also function as A/D converter analog input pins (AN_{12} to AN_{15}) and D/A converter analog output pins (DA_2 and DA_3). Port 5 pin functions are shown in table 9.10.

Table 9.10 Port 5 Pin Functions

<table>
<tr><th>Pin</th><th>Selection Method and Pin Functions</th></tr>
<tr><td>$P5_7/AN_{15}/DA_3$
$P5_6/AN_{14}/DA_2$
$P5_5/AN_{13}$
$P5_4/AN_{12}$</td><td>These pins also function as A/D converter analog input pins (AN_{12} to AN_{15}) and D/A converter analog output pins (DA_2 and DA_3). $P5_7$ to $P5_4$ have no data direction register.</td></tr>
<tr><td>$P5_3/\overline{ADTRG}/\overline{IRQ_7}/\overline{WAIT}/\overline{BREQO}$</td><td>The pin function is switched as shown below according to the combination of the operating mode, bits TRGS1 and TRGS0 in the A/D control register (ADCR), and bits IRQPAS, WAITE, WAITPS, BREQOE, BREQOPS, and P53DDR.
<table>
<tr><td>Operating mode</td><td colspan="6">Modes 4 to 6</td><td colspan="2">Mode 7</td></tr>
<tr><td>[BREQOE · BREQOPS]</td><td colspan="4">0</td><td colspan="2">1</td><td colspan="2">—</td></tr>
<tr><td>[WAITE · WAITPS]</td><td colspan="2">0</td><td colspan="2">1</td><td>0</td><td>1</td><td colspan="2">—</td></tr>
<tr><td>P53DDR</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>Pin function</td><td>$P5_3$ input pin</td><td>$P5_3$ output pin</td><td>$\overline{WAIT}$ input pin</td><td>Setting pro-hibited</td><td>$\overline{BREQ_0}$ output pin</td><td>Setting pro-hibited</td><td>$P5_3$ input pin</td><td>$P5_3$ output pin</td></tr>
<tr><td></td><td colspan="8">$\overline{ADTRG}$ input pin*1</td></tr>
<tr><td></td><td colspan="8">$\overline{IRQ_7}$ interrupt input pin*2</td></tr>
</table>
Notes: 1. $\overline{ADTRG}$ input when TRGS0 = TRGS1 = 0.
2. $\overline{IRQ_7}$ input when IRQPAS = 1.</td></tr>
</table>

bit C/$\overline{A}$ in the SCI2 SMR, bits CKE0 and CKE1 in SCR, and bits IRQPAS and P52DDR.

CKE1	0				1
C/$\overline{A}$	0			1	—
CKE0	0		1	—	—
P52DDR	0	1	—	—	—
Pin function	$P5_2$ input pin	$P5_2$ output pin	SCK_2 output pin	SCK_2 output pin	SCK_2 input pin
	$\overline{IRQ}_6$ interrupt input pin*				

Note: * $\overline{IRQ}_6$ input when IRQPAS = 1.

$P5_1/RxD_2/\overline{IRQ}_5$

The pin function is switched as shown below according to the combination of bit RE in the SCI2 SCR, and bits IRQPAS and P51DDR.

RE	0		1
P51DDR	0	1	—
Pin function	$P5_1$ input pin	$P5_1$ output pin	RxD_2 input pin
	$\overline{IRQ}_5$ interrupt input pin*		

Note: * $\overline{IRQ}_5$ input when IRQPAS = 1.

$P5_0/TxD_2/\overline{IRQ}_4$

The pin function is switched as shown below according to the combination of bit TE in the SCI2 SCR, and bits IRQPAS and P50DDR.

TE	0		1
P50DDR	0	1	—
Pin function	$P5_0$ input pin	$P5_0$ output pin	TxD_2 output pin
	$\overline{IRQ}_4$ interrupt input pin*		

Note: * $\overline{IRQ}_4$ input when IRQPAS = 1.

9.7.1 Overview

Port 6 is an 8-bit I/O port. Port 6 pins also function as interrupt input pins ($\overline{IRQ_0}$ and $\overline{IRQ_1}$) and bus control output pins ($\overline{CS_4}$ to $\overline{CS_7}$). The functions of pins $P6_5$ to $P6_2$ are the same in all operating modes, while the functions of pins $P6_7$, $P6_6$, $P6_1$, and $P6_0$ change according to the operating mode. Switching of $\overline{CS_4}$ to $\overline{CS_7}$ output can be performed by setting PFCR2. Pins $P6_7$ to $P6_4$ are Schmitt-triggered inputs. Figure 9.6 shows the port 6 pin configuration.

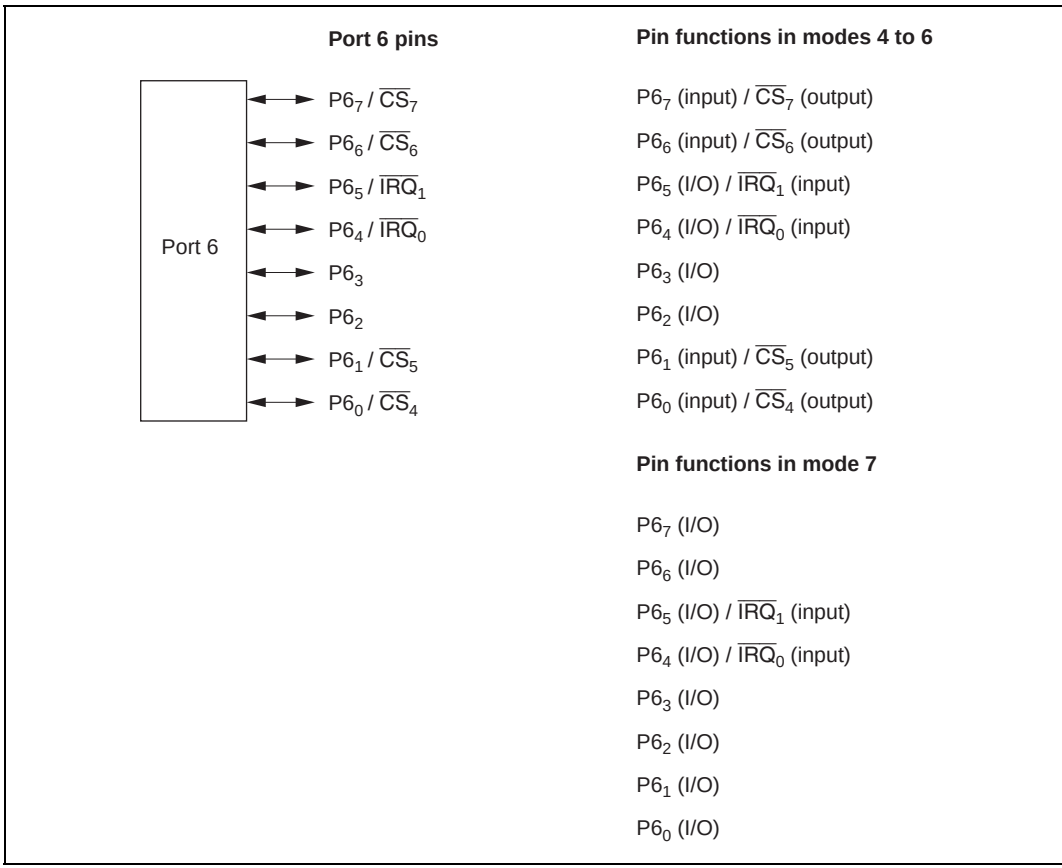


Figure 9.6 Port 6 Pin Functions

Table 9.11 shows the port 6 register configuration.

Table 9.11 Port 6 Registers

Name	Abbreviation	R/W	Initial Value	Address*
Port 6 data direction register	P6DDR	W	H'00	H'FEB5
Port 6 data register	P6DR	R/W	H'00	H'FF65
Port 6 register	PORT6	R	Undefined	H'FF55
Port function control register 2	PFCR2	R/W	H'30	H'FFAC

Note: * Lower 16 bits of the address.

Port 6 Data Direction Register (P6DDR)

Bit	:	7	6	5	4	3	2	1	0
		P67DDR	P66DDR	P65DDR	P64DDR	P63DDR	P62DDR	P61DDR	P60DDR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W	W

P6DDR is an 8-bit write-only register, the individual bits of which specify input or output for the pins of port 6. P6DDR cannot be read; if it is, an undefined value will be read.

Setting a P6DDR bit to 1 makes the corresponding port 6 pin an output pin, while clearing the bit to 0 makes the pin an input pin.

P6DDR is initialized to H'00 by a reset, and in hardware standby mode. It retains its prior state in software standby mode.

Bit	:	7	6	5	4	3	2	1	0
		P67DR	P66DR	P65DR	P64DR	P63DR	P62DR	P61DR	P60DR
Initial value	:	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

P6DR is an 8-bit readable/writable register that stores output data for the port 6 pins ($P6_7$ to $P6_0$).

P6DR is initialized to H'00 by a reset, and in hardware standby mode. It retains its prior state in software standby mode.

Port 6 Register (PORT6)

Bit	:	7	6	5	4	3	2	1	0
		P67	P66	P65	P64	P63	P62	P61	P60
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

Note: * Determined by state of pins $P6_7$ to $P6_0$.

PORT6 is an 8-bit read-only register that shows the pin states. It cannot be written to. Writing of output data for the port 6 pins ($P6_7$ to $P6_0$) must always be performed on P6DR.

If a port 6 read is performed while P6DDR bits are set to 1, the P6DR values are read. If a port 6 read is performed while P6DDR bits are cleared to 0, the pin states are read.

After a reset and in hardware standby mode, PORT6 contents are determined by the pin states, as P6DDR and P6DR are initialized. PORT6 retains its prior state in software standby mode.

Bit	:	7	6	5	4	3	2	1	0
		WAITPS	BREQOPS	CS167E	CS25E	ASOD	—	—	—
Initial value	:	0	0	1	1	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R	R	R

PFCR2 is an 8-bit readable/writable register that performs I/O port control. PFCR2 is initialized to H'30 by a reset, and in hardware standby mode.

Bit 7—WAIT Pin Select (WAITPS): Selects the $\overline{\text{WAIT}}$ input pin. For details, see section 9.6, Port 5.

Bit 6—BREQO Pin Select (BREQOPS): Selects the $\overline{\text{BREQO}}$ output pin. For details, see section 9.6, Port 5.

Bit 5—CS167 Enable (CS167E): Enables or disables $\overline{\text{CS}}_1$, $\overline{\text{CS}}_6$, and $\overline{\text{CS}}_7$ output. Only change the CS167E bit setting when the DDR bits are cleared to 0.

Bit 5 CS167E	Description	
0	$\overline{\text{CS}}_1$, $\overline{\text{CS}}_6$, and $\overline{\text{CS}}_7$ output disabled (can be used as I/O ports)	
1	$\overline{\text{CS}}_1$, $\overline{\text{CS}}_6$, and $\overline{\text{CS}}_7$ output enabled	(Initial value)

Bit 4—CS25 Enable (CS25E): Enables or disables $\overline{\text{CS}}_2$, $\overline{\text{CS}}_3$, $\overline{\text{CS}}_4$, and $\overline{\text{CS}}_5$ output. Only change the CS25E bit setting when the DDR bits are cleared to 0.

Bit 4 CS25E	Description	
0	$\overline{\text{CS}}_2$, $\overline{\text{CS}}_3$, $\overline{\text{CS}}_4$, and $\overline{\text{CS}}_5$ output disabled (can be used as I/O ports)	
1	$\overline{\text{CS}}_2$, $\overline{\text{CS}}_3$, $\overline{\text{CS}}_4$, and $\overline{\text{CS}}_5$ output enabled	(Initial value)

Bit 3—AS Output Disable (ASOD): Enables or disables $\overline{\text{AS}}$ output. For details, see section 9.16, Port F.

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

Port 6 pins also function as interrupt input pins ($\overline{IRQ_0}$ and $\overline{IRQ_1}$) and bus control output pins (CS_4 to $\overline{CS_7}$). Port 6 pin functions are shown in table 9.12.

Table 9.12 Port 6 Pin Functions

Pin	Selection Method and Pin Functions
$P6_7/\overline{CS_7}$	The pin function is switched as shown below according to the combination of bits P67DDR and CS167E.

Mode	Modes 4 to 6				Mode 7	
CS167E	0		1		0	1
P67DDR	0	1	0	1	—	—
Pin function	$P6_7$ input pin	$P6_7$ output pin	$P6_7$ input pin	$\overline{CS_7}$ output pin	$P6_7$ input pin	$P6_7$ output pin

$P6_6/\overline{CS_6}$ The pin function is switched as shown below according to the combination of bits P66DDR and CS167E.

Mode	Modes 4 to 6				Mode 7	
CS167E	0		1		0	1
P66DDR	0	1	0	1	—	—
Pin function	$P6_6$ input pin	$P6_6$ output pin	$P6_6$ input pin	$\overline{CS_6}$ output pin	$P6_6$ input pin	$P6_6$ output pin

$P6_5/\overline{IRQ_1}$ The pin function is switched as shown below according to bit P65DDR.

P65DDR	0	1
Pin function	$P6_5$ input pin	$P6_5$ output pin
	$\overline{IRQ_1}$ interrupt input pin	

$P6_4/\overline{IRQ_0}$ The pin function is switched as shown below according to bit P64DDR.

P64DDR	0	1
Pin function	$P6_4$ input pin	$P6_4$ output pin
	$\overline{IRQ_0}$ interrupt input pin	

P63DDR	0	1
Pin function	$P6_3$ input pin	$P6_3$ output pin

$P6_2$

The pin function is switched as shown below according to bit P62DDR.

P62DDR	0	1
Pin function	$P6_2$ input pin	$P6_2$ output pin

$P6_1/\overline{CS}_5$

The pin function is switched as shown below according to the combination of bits P61DDR and CS25E.

Mode	Modes 4 to 6				Mode 7	
CS25E	0		1		0	1
P61DDR	0	1	0	1	—	—
Pin function	$P6_1$ input pin	$P6_1$ output pin	$P6_1$ input pin	$\overline{CS}_5$ output pin	$P6_1$ input pin	$P6_1$ output pin

$P6_0/\overline{CS}_4$

The pin function is switched as shown below according to the combination of bits P60DDR and CS25E.

Mode	Modes 4 to 6				Mode 7	
CS25E	0		1		0	1
P60DDR	0	1	0	1	—	—
Pin function	$P6_0$ input pin	$P6_0$ output pin	$P6_0$ input pin	$\overline{CS}_4$ output pin	$P6_0$ input pin	$P6_0$ output pin

9.8.1 Overview

Port 7 is a 6-bit I/O port. Port 7 pins also function as 8-bit timer I/O pins ($TMRI_0$, $TMCI_0$, TMO_0, $TMRI_1$, $TMCI_1$, TMO_1). Port 7 pin functions are the same in all operating modes. Port 7 uses Schmitt-triggered input.

Figure 9.7 shows the port 7 pin configuration.

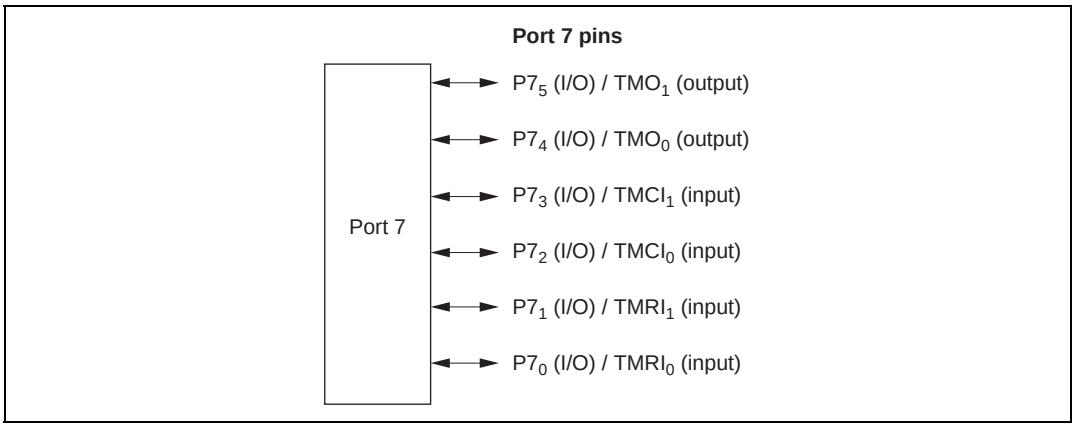


Figure 9.7 Port 7 Pin Functions

Table 9.13 shows the port 7 register configuration.

Table 9.13 Port 7 Registers

Name	Abbreviation	R/W	Initial Value*2	Address*1
Port 7 data direction register	P7DDR	W	H'00	H'FEB6
Port 7 data register	P7DR	R/W	H'00	H'FF66
Port 7 register	PORT7	R	Undefined	H'FF56

Notes: 1. Lower 16 bits of the address.
2. Value of bits 5 to 0.

Port 7 Data Direction Register (P7DDR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	P75DDR	P74DDR	P73DDR	P72DDR	P71DDR	P70DDR
Initial value	:	Undefined	Undefined	0	0	0	0	0	0
R/W	:	—	—	W	W	W	W	W	W

P7DDR is a 6-bit write-only register, the individual bits of which specify input or output for the pins of port 7. P2DDR cannot be read; if it is, an undefined value will be read. Bits 7 and 6 are reserved.

Setting a P7DDR bit to 1 makes the corresponding port 7 pin an output pin, while clearing the bit to 0 makes the pin an input port.

P7DDR is initialized to H'00 (bits 5 to 0) by a reset, and in hardware standby mode. It retains its prior state in software standby mode.

Bit	:	7	6	5	4	3	2	1	0
		—	—	P75DR	P74DR	P73DR	P72DR	P71DR	P70DR
Initial value	:	Undefined	Undefined	0	0	0	0	0	0
R/W	:	—	—	R/W	R/W	R/W	R/W	R/W	R/W

P7DR is a 6-bit readable/writable register that stores output data for the port 7 pins ($P7_5$ to $P7_0$). Bits 7 and 6 are reserved; they return an undefined value if read, and cannot be modified.

P7DR is initialized to H'00 (bits 5 to 0) by a reset, and in hardware standby mode. It retains its prior state in software standby mode.

Port 7 Register (PORT7)

Bit	:	7	6	5	4	3	2	1	0
		—	—	P75	P74	P73	P72	P71	P70
Initial value	:	Undefined	Undefined	—*	—*	—*	—*	—*	—*
R/W	:	—	—	R	R	R	R	R	R

Note: * Determined by state of pins $P7_5$ to $P7_0$.

PORT7 is a 6-bit read-only register that shows the pin states. It cannot be written to. Writing of output data for the port 7 pins ($P7_5$ to $P7_0$) must always be performed on P7DR.

Bits 7 and 6 are reserved, they return an undefined value if read, and cannot be modified.

If a port 7 read is performed while P7DDR bits are set to 1, the P7DR values are read. If a port 7 read is performed while P7DDR bits are cleared to 0, the pin states are read.

After a reset and in hardware standby mode, PORT7 contents are determined by the pin states, as P7DDR and P7DR are initialized. PORT7 retains its prior state in software standby mode.

Port 7 pins also function as 8-bit timer I/O pins ($TMRI_0$, $TMCI_0$, TMO_0, $TMRI_1$, $TMCI_1$, and TMO_1). Port 7 pin functions are shown in table 9.14.

Table 9.14 Port 7 Pin Functions

Pin	Selection Method and Pin Functions
$P7_5/TMO_1$	The pin function is switched as shown below according to the combination of bits OS3 to OS0 in 8-bit timer TCSR1 and bit P75DDR.

OS3 to OS0	All 0		Not all 0
P75DDR	0	1	—
Pin function	$P7_5$ input pin	$P7_5$ output pin	TMO_1 output pin

$P7_4/TMO_0$	The pin function is switched as shown below according to the combination of bits OS3 to OS0 in 8-bit timer TCSR0 and bit P74DDR.

OS3 to OS0	All 0		Not all 0
P74DDR	0	1	—
Pin function	$P7_4$ input pin	$P7_4$ output pin	TMO_0 output pin

$P7_3/TMCI_1$	The pin function is switched as shown below according to bit P73DDR. When this pin is used as an 8-bit timer external clock input pin, the external clock is selected with bits CKS2 to CKS0 in TCR1.

P73DDR	0	1
Pin function	$P7_3$ input pin	$P7_3$ output pin
	$TMCI_1$ input pin	

this pin is used as an 8-bit timer external clock input pin, the external clock is selected with bits CKS2 to CKS0 in TCR0.

P72DDR	0	1
Pin function	$P7_2$ input pin	$P7_2$ output pin
	$TMCI_0$ input pin	

$P7_1/TMRI_1$

The pin function is switched as shown below according to bit P71DDR. When this pin is used as an 8-bit timer counter reset pin, bits CCLR1 and CCLR0 in TCR1 are both set to 1.

P71DDR	0	1
Pin function	$P7_1$ input pin	$P7_1$ output pin
	$TMRI_1$ input pin	

$P7_0/TMRI_0$

The pin function is switched as shown below according to bit P70DDR. When this pin is used as an 8-bit timer counter reset pin, bits CCLR1 and CCLR0 in TCR0 are both set to 1.

P70DDR	0	1
Pin function	$P7_0$ input pin	$P7_0$ output pin
	$TMRI_0$ input pin	

9.9.1 Overview

Port 8 is a 7-bit I/O port. Port 8 pins also function as DMAC I/O pins ($\overline{DREQ_0}$, $\overline{TEND_0}$, $\overline{DACK_0}$, $\overline{DREQ_1}$, $\overline{TEND_1}$, and $\overline{DACK_1}$) and a bus control signal input pin ($\overline{WAIT}$). Figure 9.8 shows the port 8 pin configuration.

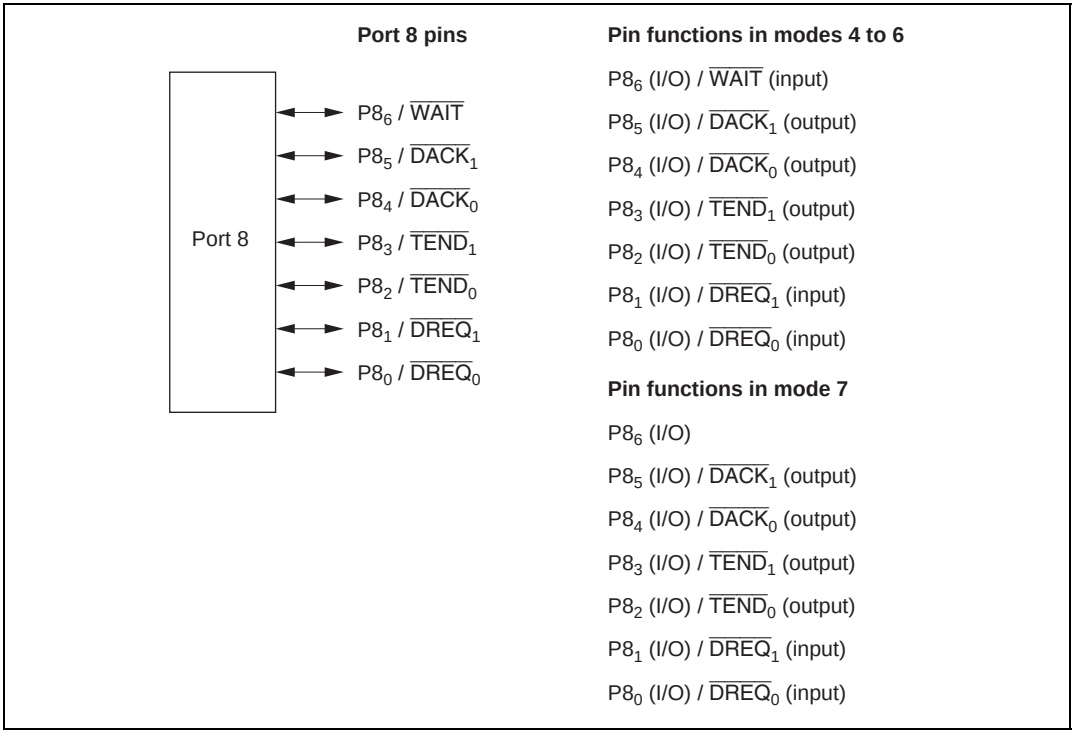


Figure 9.8 Port 8 Pin Functions

Table 9.15 shows the port 8 register configuration.

Table 9.15 Port 8 Registers

Name	Abbreviation	R/W	Initial Value*2	Address*1
Port 8 data direction register	P8DDR	W	H'00	H'FEB7
Port 8 data register	P8DR	R/W	H'00	H'FF67
Port 8 register	PORT8	R	Undefined	H'FF57
Port function control register2	PFCR2	R/W	H'30	H'FFAC

Notes: 1. Lower 16 bits of the address.
2. Value of bits 6 to 0.

Port 8 Data Direction Register (P8DDR)

Bit	:	7	6	5	4	3	2	1	0
		—	P86DDR	P85DDR	P84DDR	P83DDR	P82DDR	P81DDR	P80DDR
Initial value	:	Undefined	0	0	0	0	0	0	0
R/W	:	—	W	W	W	W	W	W	W

P8DDR is a 7-bit write-only register, the individual bits of which specify input or output for the pins of port 8. P2DDR cannot be read; if it is, an undefined value will be read. Bit 7 is reserved.

Setting a P8DDR bit to 1 makes the corresponding port 8 pin an output pin, while clearing the bit to 0 makes the pin an input pin.

P8DDR is initialized to H'00 (bits 6 to 0) by a reset, and in hardware standby mode. It retains its prior state in software standby mode.

Bit	:	7	6	5	4	3	2	1	0
		—	P86DR	P85DR	P84DR	P83DR	P82DR	P81DR	P80DR
Initial value	:	Undefined	0	0	0	0	0	0	0
R/W	:	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P8DR is a 7-bit readable/writable register that stores output data for the port 8 pins ($P8_6$ to $P8_0$).

Bit 7 is reserved; it returns an undefined value if read, and cannot be modified.

P8DR is initialized to H'00 (bits 6 to 0) by a reset, and in hardware standby mode. It retains its prior state in software standby mode.

Port 8 Register (PORT8)

Bit	:	7	6	5	4	3	2	1	0
		—	P86	P85	P84	P83	P82	P81	P80
Initial value	:	Undefined	—*	—*	—*	—*	—*	—*	—*
R/W	:	—	R	R	R	R	R	R	R

Note: * Determined by state of pins $P8_6$ to $P8_0$.

PORT8 is a 7-bit read-only register that shows the pin states. It cannot be written to. Writing of output data for the port 8 pins ($P8_6$ to $P8_0$) must always be performed on P8DR.

Bit 7 is reserved; it returns an undefined value if read, and cannot be modified.

If a port 8 read is performed while P8DDR bits are set to 1, the P8DR values are read. If a port 8 read is performed while P8DDR bits are cleared to 0, the pin states are read.

After a reset and in hardware standby mode, PORT8 contents are determined by the pin states, as P8DDR and P8DR are initialized. PORT8 retains its prior state in software standby mode.

Bit	:	7	6	5	4	3	2	1	0
		WAITPS	BREQOPS	CS167E	CS25E	ASOD	—	—	—
Initial value	:	0	0	1	1	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R	R	R

PFCR2 is an 8-bit readable/writable register that performs I/O port control. PFCR2 is initialized to H'30 by a reset, and in hardware standby mode.

Bit 7—WAIT Pin Select (WAITPS): Selects the $\overline{\text{WAIT}}$ output pin. Set the WAITPS bit before setting the DDR bit clear to 0 and the WAITE bit in BCRL to 1.

Bit 7 WAITPS	Description	
0	$\overline{\text{WAIT}}$ output is pin $P8_6$	(Initial value)
1	$\overline{\text{WAIT}}$ output is pin $P5_3$	

Bit 6—BREQO Pin Select (BREQOPS): Selects the $\overline{\text{BREQO}}$ output pin. For details, see section 9.6, Port 5, and section 9.16, Port F.

Bit 5—CS167 Enable (CS167E): Enables or disables $\overline{\text{CS}}_1$, $\overline{\text{CS}}_6$, and $\overline{\text{CS}}_7$ output. For details, see section 9.7, Port 6, and section 9.17, Port G.

Bit 4—CS25 Enable (CS25E): Enables or disables $\overline{\text{CS}}_2$, $\overline{\text{CS}}_3$, $\overline{\text{CS}}_4$, and $\overline{\text{CS}}_5$ output. For details, see section 9.7, Port 6, and section 9.17, Port G.

Bit 3—AS Output Disable (ASOD): Enables or disables $\overline{\text{AS}}$ output. For details, see section 9.16, Port F.

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

Port 8 pins also function as DMAC I/O pins ($\overline{DREQ_0}$, $\overline{TEND_0}$, $\overline{DACK_0}$, $\overline{DREQ_1}$, $\overline{TEND_1}$, and $\overline{DACK_1}$) and a bus control signal input pin ($\overline{WAIT}$). Port 8 pin functions are shown in table 9.16.

Table 9.16 Port 8 Pin Functions

Pin	Selection Method and Pin Functions
$P8_6/\overline{WAIT}$	The pin function is switched as shown below according to the combination of the operating mode and bits WAITE, WAITPS, and P86DDR.

Operating mode	Modes 4 to 6				Mode 7	
[WAITE · $\overline{WAITPS}$]	0		1		—	
P86DDR	0	1	0	1	0	1
Pin function	$P8_6$ input pin	$P8_6$ output pin	$\overline{WAIT}$ input pin	Setting pro-hibited	$P8_6$ input pin	$P8_6$ output pin

$P8_5/\overline{DACK_1}$	The pin function is switched as shown below according to the combination of bit SAE1 in DMABCRH and bit P85DDR.

SAE1	0		1
P85DDR	0	1	—
Pin function	$P8_5$ input pin	$P8_5$ output pin	$\overline{DACK_1}$ output pin

$P8_4/\overline{DACK_0}$	The pin function is switched as shown below according to the combination of bit SAE0 in DMABCRH and bit P84DDR.

SAE0	0		1
P84DDR	0	1	—
Pin function	$P8_4$ input pin	$P8_4$ output pin	$\overline{DACK_0}$ output pin

bit TEE1 in the DMAC's DMATCR and bit P83DDR.

TEE1	0		1
P83DDR	0	1	—
Pin function	$P8_3$ input pin	$P8_3$ output pin	$\overline{TEND}_1$ output pin

$P8_2/\overline{TEND}_0$ The pin function is switched as shown below according to the combination of bit TEE0 in the DMAC's DMATCR and bit P82DDR.

TEE0	0		1
P82DDR	0	1	—
Pin function	$P8_2$ input pin	$P8_2$ output pin	$\overline{TEND}_0$ output pin

$P8_1/\overline{DREQ}_1$ The pin function is switched as shown below according to bit P81DDR.

P81DDR	0	1
Pin function	$P8_1$ input pin	$P8_1$ output pin
	$\overline{DREQ}_1$ input pin	

$P8_0/\overline{DREQ}_0$ The pin function is switched as shown below according to bit P80DDR.

P80DDR	0	1
Pin function	$P8_0$ input pin	$P8_0$ output pin
	$\overline{DREQ}_0$ input pin	

9.10.1 Overview

Port 9 is a 6-bit I/O port. Port 9 pins also function as interrupt input pins ($\overline{IRQ}_2$, $\overline{IRQ}_3$, $\overline{IRQ}_4$, $\overline{IRQ}_5$, $\overline{IRQ}_6$, and $\overline{IRQ}_7$). When the IRQPAS bit is set to 1, inputs $\overline{IRQ}_4$ to $\overline{IRQ}_7$ are switched to $P5_0$ to $P5_3$. Port 9 pin functions are the same in all operating modes. Port 9 uses Schmitt-triggered input. Figure 9.9 shows the port 9 pin configuration.

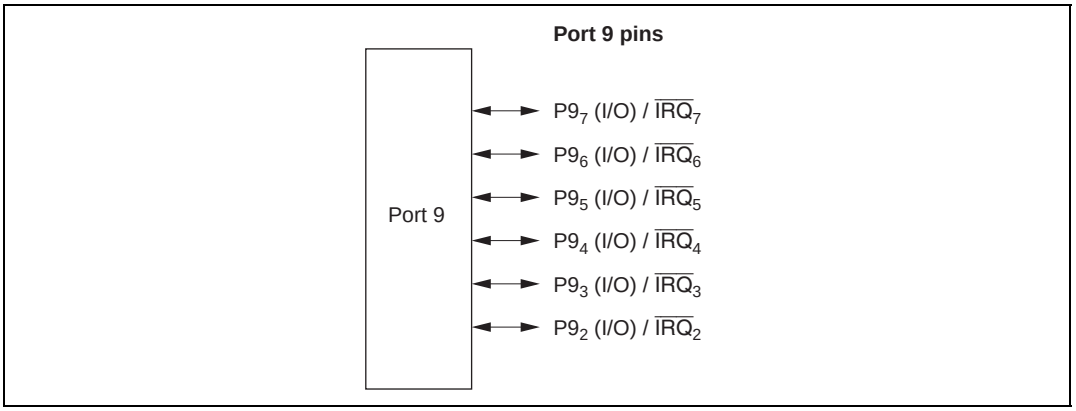


Figure 9.9 Port 9 Pin Functions

Table 9.17 shows the port 9 register configuration.

Table 9.17 Port 9 Registers

Name	Abbreviation	R/W	Initial Value	Address*1
Port 9 data direction register	P9DDR	W	H'00*2	H'FEB8
Port 9 data register	P9DR	R/W	H'00*2	H'FF68
Port 9 register	PORT9	R	Undefined	H'FF58
System control register	SYSCR	R/W	H'01	H'FF39

Notes: 1. Lower 16 bits of the address.
2. Value of bits 7 to 2.

Port 9 Data Direction Register (P9DDR)

Bit	:	7	6	5	4	3	2	1	0
		P97DDR	P96DDR	P95DDR	P94DDR	P93DDR	P92DDR	—	—
Initial value	:	0	0	0	0	0	0	Undefined	Undefined
R/W	:	W	W	W	W	W	W	—	—

P9DDR is a 6-bit write-only register, the individual bits of which specify input or output for the pins of port 9. P2DDR cannot be read; if it is, an undefined value will be read. Bits 1 and 0 are reserved.

Setting a P9DDR bit to 1 makes the corresponding port 9 pin an output pin, while clearing the bit to 0 makes the pin an input pin.

P9DDR is initialized to H'00 (bits 7 to 2) by a reset, and in hardware standby mode. It retains its prior state in software standby mode.

Bit	:	7	6	5	4	3	2	1	0
		P97DR	P96DR	P95DR	P94DR	P93DR	P92DR	—	—
Initial value	:	0	0	0	0	0	0	Undefined	Undefined
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	—	—

P9DR is a 6-bit readable/writable register that stores output data for the port 9 pins ($P9_7$ to $P9_2$). Bits 1 and 0 are reserved; they return an undefined value if read, and cannot be modified. P9DR is initialized to H'00 (bits 7 to 2) by a reset, and in hardware standby mode. It retains its prior state in software standby mode.

Port 9 Register (PORT9)

Bit	:	7	6	5	4	3	2	1	0
		P97	P96	P95	P94	P93	P92	—	—
Initial value	:	—*	—*	—*	—*	—*	—*	Undefined	Undefined
R/W	:	R	R	R	R	R	R	—	—

Note: * Determined by state of pins $P9_7$ to $P9_2$.

PORT9 is a 6-bit read-only register that shows the pin states. It cannot be written to. Writing of output data for the port 9 pins ($P9_7$ to $P9_2$) must always be performed on P9DR.

Bits 1 and 0 are reserved; they return an undefined value if read, and cannot be modified.

If a port 9 read is performed while P9DDR bits are set to 1, the P9DR values are read. If a port 9 read is performed while P9DDR bits are cleared to 0, the pin states are read.

After a reset and in hardware standby mode, PORT9 contents are determined by the pin states, as P9DDR and P9DR are initialized. PORT9 retains its prior state in software standby mode.

Bit	:	7	6	5	4	3	2	1	0
		—	—	INTM1	INTM0	NMIEG	LWROD	IRQPAS	RAME
Initial value	:	0	0	0	0	0	0	0	1
R/W	:	R/W	—	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, controls the $\overline{\text{LWR}}$ pin, switches the $\overline{\text{IRQ}}_4$ to $\overline{\text{IRQ}}_7$ input pins, and selects the detected edge for NMI. SYSCR is initialized to H'01 by a reset, and in hardware standby mode. It is not initialized in software standby mode.

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

Bit 3—NMI Edge Select (NMIEG): Selects the input edge for the NMI pin. For details, see section 5, Interrupt Controller.

Bit 2—LWR Output Disable (LWROD): Enables or disables $\overline{\text{LWR}}$ output. For details, see section 9.16, Port F.

Bit 1—IRQ Port Switching Select (IRQPAS): Selects switching of input pins $\overline{\text{IRQ}}_4$ to $\overline{\text{IRQ}}_7$. $\overline{\text{IRQ}}_4$ to $\overline{\text{IRQ}}_7$ input is always performed from one of the ports.

Bit 1 IRQPAS	Description	
0	$P9_4$ to $P9_7$ used for $\overline{\text{IRQ}}_4$ to $\overline{\text{IRQ}}_7$ input	(Initial value)
1	$P5_0$ to $P5_3$ used for $\overline{\text{IRQ}}_4$ to $\overline{\text{IRQ}}_7$ input	

Bit 0—RAM Enable (RAME): Enables or disables on-chip RAM. For details, see section 18, RAM.

Port 9 pins also function as interrupt input pins ($\overline{IRQ_2}$, $\overline{IRQ_3}$, $\overline{IRQ_4}$, $\overline{IRQ_5}$, $\overline{IRQ_6}$, and $\overline{IRQ_7}$). Port 9 pin functions are shown in table 9.18.

Table 9.18 Port 9 Pin Functions

Pin	Selection Method and Pin Functions
$P9_7/\overline{IRQ_7}$	The pin function is switched as shown below according to the combination of bits P97DDR and IRQPAS.

P97DDR	0	1
Pin function	$P9_7$ input pin	$P9_7$ output pin
	$\overline{IRQ_7}$ interrupt input pin*	

Note: * $\overline{IRQ_7}$ input when IRQPAS = 0.

$P9_6/\overline{IRQ_6}$ The pin function is switched as shown below according to the combination of bits P96DDR and IRQPAS.

P96DDR	0	1
Pin function	$P9_6$ input pin	$P9_6$ output pin
	$\overline{IRQ_6}$ interrupt input pin*	

Note: * $\overline{IRQ_6}$ input when IRQPAS = 0.

$P9_5/\overline{IRQ_5}$ The pin function is switched as shown below according to the combination of bits P95DDR and IRQPAS.

P95DDR	0	1
Pin function	$P9_5$ input pin	$P9_5$ output pin
	$\overline{IRQ_5}$ interrupt input pin*	

Note: * $\overline{IRQ_5}$ input when IRQPAS = 0.

$P9_4/\overline{IRQ_4}$ The pin function is switched as shown below according to the combination of bits P94DDR and IRQPAS.

P94DDR	0	1
Pin function	$P9_4$ input pin	$P9_4$ output pin
	$\overline{IRQ_4}$ interrupt input pin*	

Note: * $\overline{IRQ_4}$ input when IRQPAS = 0.

bits P93DDR and IRQPAS.

P93DDR	0	1
Pin function	$P9_3$ input pin	$P9_3$ output pin
	$\overline{IRQ}_3$ interrupt input pin*	

Note: * $\overline{IRQ}_3$ input when IRQPAS = 0.

$P9_2/\overline{IRQ}_2$

The pin function is switched as shown below according to the combination of bits P92DDR and IRQPAS.

P92DDR	0	1
Pin function	$P9_2$ input pin	$P9_2$ output pin
	$\overline{IRQ}_2$ interrupt input pin*	

Note: * $\overline{IRQ}_2$ input when IRQPAS = 0.

9.11.1 Overview

Port A is an 8-bit I/O port. Port A pins also function as address bus outputs. The pin functions change according to the operating mode. The address output or port output function can be selected by means of bits A23E to A20E in PFCR1.

Port A has a built-in MOS input pull-up function that can be controlled by software. Pins PA_7 to PA_4 are Schmitt-triggered inputs.

Figure 9.10 shows the port A pin configuration.

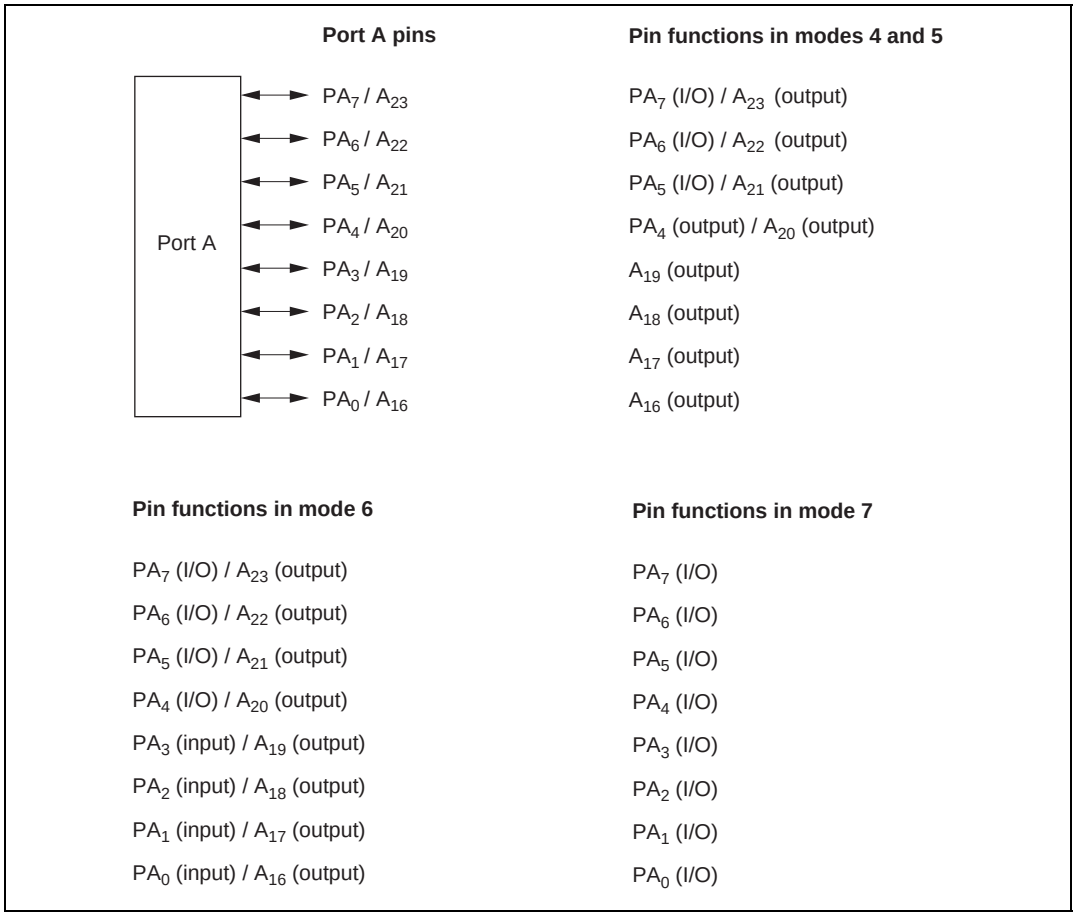


Figure 9.10 Port A Pin Functions

Table 9.19 shows the port A register configuration.

Table 9.19 Port A Registers

Name	Abbreviation	R/W	Initial Value	Address*
Port A data direction register	PADDR	W	H'00	H'FEB9
Port A data register	PADR	R/W	H'00	H'FF69
Port A register	PORTA	R	Undefined	H'FF59
Port A MOS pull-up control register	PAPCR	R/W	H'00	H'FF70
Port A open drain control register	PAODR	R/W	H'00	H'FF77
Port function control register 1	PFCR1	R/W	H'0F	H'FF45

Note: * Lower 16 bits of the address.

Port A Data Direction Register (PADDR)

Bit	:	7	6	5	4	3	2	1	0
		PA7DDR	PA6DDR	PA5DDR	PA4DDR	PA3DDR	PA2DDR	PA1DDR	PA0DDR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W	W

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

PADDR is initialized to H'00 by a reset, and in hardware standby mode. It retains its prior state in software standby mode. The OPE bit in SBYCR is used to select whether the address output pins retain their output state or become high-impedance when a transition is made to software standby mode.

Port A Data Register (PADR)

Bit	:	7	6	5	4	3	2	1	0
		PA7DR	PA6DR	PA5DR	PA4DR	PA3DR	PA2DR	PA1DR	PA0DR
Initial value	:	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

PADR is an 8-bit readable/writable register that stores output data for the port A pins (PA_7 to PA_0).

Port A Register (PORTA)

Bit	:	7	6	5	4	3	2	1	0
		PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

Note: * Determined by state of pins PA_7 to PA_0.

PORTA is an 8-bit read-only register that shows the pin states. It cannot be written to. Writing of output data for the port A pins (PA_7 to PA_0) must always be performed on PADR.

If a port A read is performed while PADDR bits are set to 1, the PADR values are read. If a port A 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 pin states, as PADDR and PADR are initialized. PORTA retains its prior state in software standby mode.

Port A MOS Pull-Up Control Register (PAPCR)

Bit	:	7	6	5	4	3	2	1	0
		PA7PCR	PA6PCR	PA5PCR	PA4PCR	PA3PCR	PA2PCR	PA1PCR	PA0PCR
Initial value	:	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

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

All the bits are valid in modes 6 and 7, and bits 7 to 5 are valid in modes 4 and 5. When a PADDR bit is cleared to 0 (input port setting), setting the corresponding PAPCR bit to 1 turns on the MOS input pull-up for the corresponding pin.

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

Bit	:	7	6	5	4	3	2	1	0
		PA7ODR	PA6ODR	PA5ODR	PA4ODR	PA3ODR	PA2ODR	PA1ODR	PA0ODR
Initial value	:	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

PAODR is an 8-bit readable/writable register that controls whether PMOS is on or off for each port A pin (PA_7 to PA_0).

PAODR is valid only in mode 7. Do not PAODR bits to 1 in modes 4 to 6.

Setting a PAODR bit to 1 makes the corresponding port A pin an NMOS open-drain output, while clearing the bit to 0 makes the pin a CMOS output.

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

Port Function Control Register 1 (PFCR1)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	A23E	A22E	A21E	A20E
Initial value	:	0	0	0	0	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PFCR1 is an 8-bit readable/writable register that performs I/O port control. PFCR1 is initialized to H'0F by a reset, and in hardware standby mode.

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

Bit 3—Address 23 Enable (A23E): Enables or disables address output 23 (A_{23}). This bit is valid in modes 4 to 6.

Bit 3 A23E	Description	
0	DR is output when PA7DDR = 1	
1	A_{23} is output when PA7DDR = 1	(Initial value)

Bit 2 A22E	Description	
0	DR is output when PA6DDR = 1	
1	A_{22} is output when PA6DDR = 1	(Initial value)

Bit 1—Address 21 Enable (A21E): Enables or disables address output 21 (A_{21}). This bit is valid in modes 4 to 6.

Bit 1 A21E	Description	
0	DR is output when PA5DDR = 1	
1	A_{21} is output when PA5DDR = 1	(Initial value)

Bit 0—Address 20 Enable (A20E): Enables or disables address output 20 (A_{20}). This bit is valid in modes 4 to 6.

Bit 0 A20E	Description	
0	DR is output when PA4DDR = 1	
1	A_{20} is output when PA4DDR = 1	(Initial value)

Port A pins function as address outputs and I/O ports. Port A pin functions are shown in table 9.20.

Table 9.20 Port A Pin Functions

Pin	Selection Method and Pin Functions
PA_7/A_{23}	The pin function is switched as shown below according to the combination of the operating mode and bits A23E and PA7DDR.

Operating mode	Modes 4 to 6				Mode 7	
A23E	0		1		—	
PA7DDR	0	1	0	1	0	1
Pin function	PA_7 input pin	PA_7 output pin	PA_7 input pin	A_{23} output pin	PA_7 input pin	PA_7 output pin*

Note: * NMOS open-drain output when PA7ODR = 1.

PA_6/A_{22} The pin function is switched as shown below according to the combination of the operating mode and bits A22E and PA6DDR.

Operating mode	Modes 4 to 6				Mode 7	
A22E	0		1		—	
PA6DDR	0	1	0	1	0	1
Pin function	PA_6 input pin	PA_6 output pin	PA_6 input pin	A_{22} output pin	PA_6 input pin	PA_6 output pin*

Note: * NMOS open-drain output when PA6ODR = 1.

PA_5/A_{21} The pin function is switched as shown below according to the combination of the operating mode and bits A21E and PA5DDR.

Operating mode	Modes 4 to 6				Mode 7	
A21E	0		1		—	
PA5DDR	0	1	0	1	0	1
Pin function	PA_5 input pin	PA_5 output pin	PA_5 input pin	A_{21} output pin	PA_5 input pin	PA_5 output pin*

Note: * NMOS open-drain output when PA5ODR = 1.

the operating mode and bits A20E and PA4DDR.

Operating mode	Modes 4 and 5			Mode 6				Mode 7	
A20E	0		1	0		1		—	
PA4DDR	0	1	—	0	1	0	1	0	1
Pin function	Setting pro-hibited	PA_4 output pin	A_{20} output pin	PA_4 input pin	PA_4 output pin	PA_4 input pin	A_{20} output pin	PA_4 input pin	PA_4 output pin*

Note: * NMOS open-drain output when PA4ODR = 1.

PA_3/A_{19}
PA_2/A_{18}
PA_1/A_{17}
PA_0/A_{16}

The pin function is switched as shown below according to the combination of the operating mode and bit PAnDDR.

Operating mode	Modes 4 and 5	Mode 6		Mode 7	
PAnDDR	—	0	1	0	1
Pin function	Address output pin	PA_n input pin*1	A_m output pin*1	PA_n input pin*1	PA_n output pin*1*2

Notes: 1. n = 0 to 3, m = 16 to 19
2. PAn output is NMOS open-drain output when PAnODR = 1.

Port A has a built-in MOS input pull-up function that can be controlled by software. This MOS input pull-up function can be used by pins PA_7 to PA_5 in modes 4 and 5, and by all pins in modes 6 and 7. MOS input pull-up can be specified as on or off on an individual bit basis.

When a PADDR bit is cleared to 0, setting the corresponding PAPCR bit to 1 turns on the MOS input pull-up for that pin.

The MOS input pull-up function is in the off state after a reset, and in hardware standby mode. The prior state is retained in software standby mode.

Table 9.21 summarizes the MOS input pull-up states.

Table 9.21 MOS Input Pull-Up States (Port A)

Modes		Reset	Hardware Standby Mode	Software Standby Mode	In Other Operations
6, 7	PA_7 to PA_0	Off	Off	On/off	On/off
4, 5	PA_7 to PA_5			On/off	On/off
	PA_4 to PA_0			Off	Off

Legend

Off: MOS input pull-up is always off.

On/off: On when PADDR = 0 and PAPCR = 1; otherwise off.

9.12.1 Overview

Port B is an 8-bit I/O port. Port B has an address bus output function, and the pin functions change according to the operating mode.

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

Figure 9.11 shows the port B pin configuration.

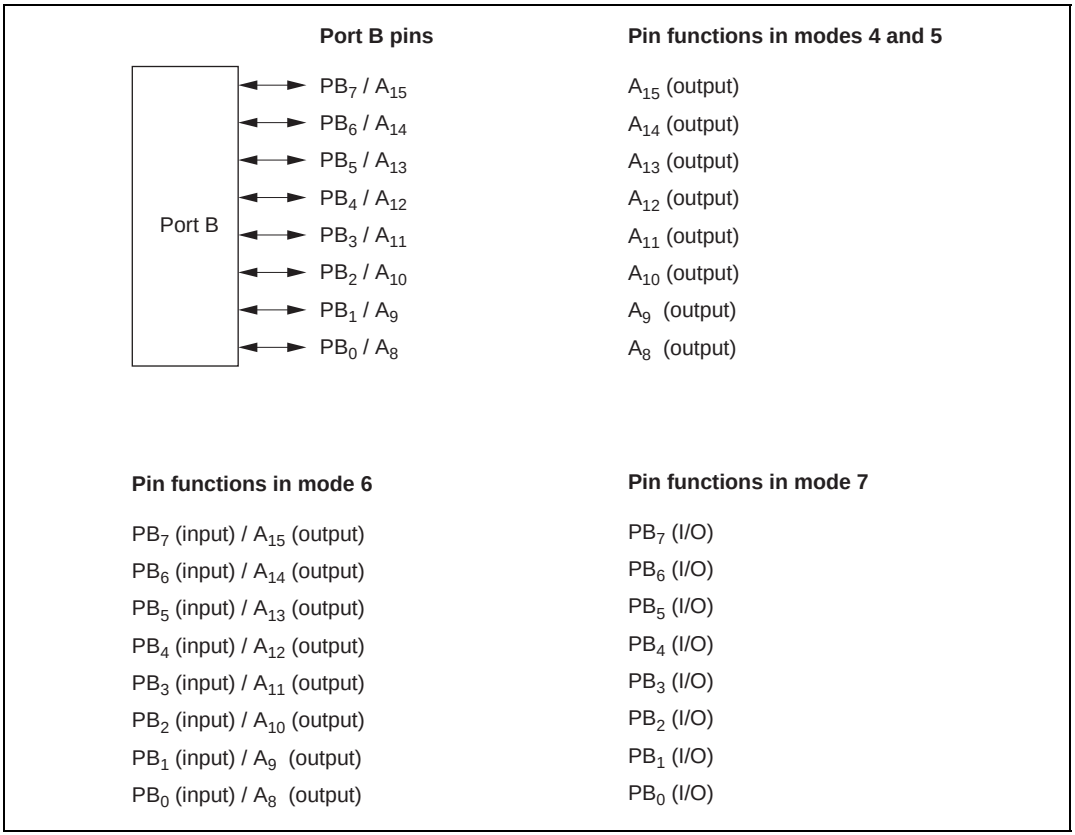


Figure 9.11 Port B Pin Functions

Table 9.22 shows the port B register configuration.

Table 9.22 Port B Registers

Name	Abbreviation	R/W	Initial Value	Address*
Port B data direction register	PBDDR	W	H'00	H'FEBA
Port B data register	PBDR	R/W	H'00	H'FF6A
Port B register	PORTB	R	Undefined	H'FF5A
Port B MOS pull-up control register	PBPCR	R/W	H'00	H'FF71

Note: * Lower 16 bits of the address.

Port B Data Direction Register (PBDDR)

Bit	:	7	6	5	4	3	2	1	0
		PB7DDR	PB6DDR	PB5DDR	PB4DDR	PB3DDR	PB2DDR	PB1DDR	PB0DDR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W	W

PBDDR is an 8-bit write-only register, the individual bits of which specify input or output for the 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 prior state in software standby mode. The OPE bit in SBYCR is used to select whether the address output pins retain their output state or become high-impedance when a transition is made to software standby mode.

- Modes 4 and 5
  The corresponding port B pins are address outputs irrespective of the value of the PBDDR bits.
- Mode 6
  Setting a PBDDR bit to 1 makes the corresponding port B pin an address output, 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, while clearing the bit to 0 makes the pin an input port.

Bit	:	7	6	5	4	3	2	1	0
		PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR
Initial value	:	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

PBDR is an 8-bit readable/writable register that stores output data for the port B pins (PB_7 to PB_0). PBDR is initialized to H'00 by a reset, and in hardware standby mode. It retains its prior state in software standby mode.

Port B Register (PORTB)

Bit	:	7	6	5	4	3	2	1	0
		PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

Note: * Determined by state of pins PB_7 to PB_0.

PORTB is an 8-bit read-only register that shows the pin states. It cannot be written to. Writing of output data for the port B pins (PB_7 to PB_0) 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. If a port B 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 pin states, as PBDDR and PBDR are initialized. PORTB retains its prior state in software standby mode.

Bit	:	7	6	5	4	3	2	1	0
		PB7PCR	PB6PCR	PB5PCR	PB4PCR	PB3PCR	PB2PCR	PB1PCR	PB0PCR
Initial value	:	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

PBPCR is an 8-bit readable/writable register that controls the MOS input pull-up function incorporated into port B on an individual bit basis.

When a PBDDR bit is cleared to 0 (input port setting) in mode 6 or 7, setting the corresponding PBPCR bit to 1 turns on the MOS input pull-up for the corresponding pin.

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

9.12.3 Pin Functions

Modes 4 and 5: In modes 4 and 5, port B pins are automatically designated as address outputs.

Port B pin functions in modes 4 and 5 are shown in figure 9.12.

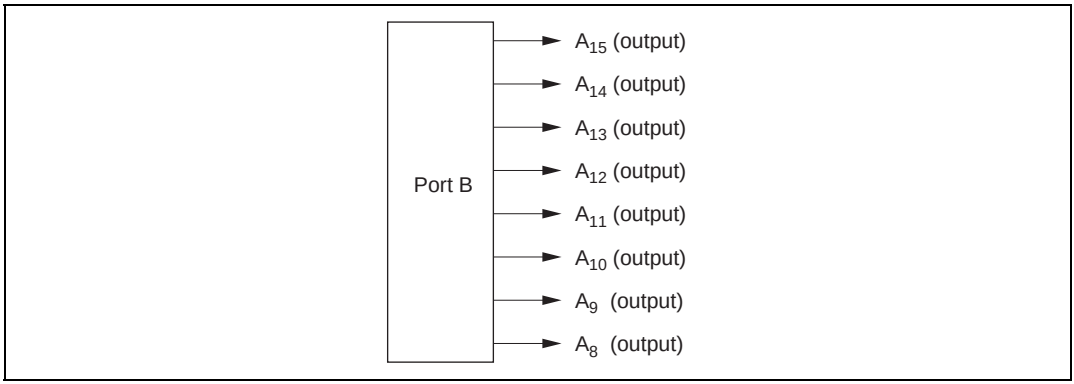


Figure 9.12 Port B Pin Functions (Modes 4 and 5)

an address output, while clearing the bit to 0 makes the pin an input port.

Port B pin functions in mode 6 are shown in figure 9.13

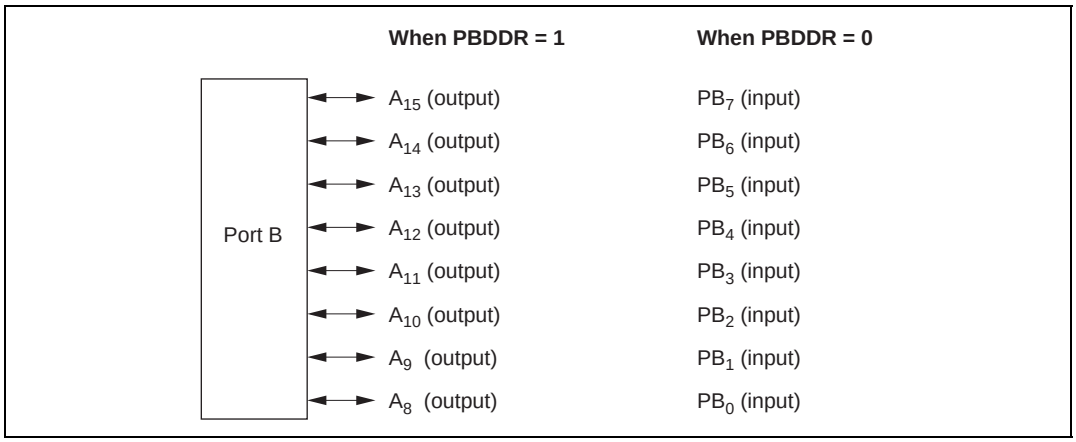


Figure 9.13 Port B Pin Functions (Mode 6)

Mode 7: In mode 7, port B pins function as I/O ports. Input or output can be specified for each pin on an individual bit basis. Setting a PBDDR bit to 1 makes the corresponding port B pin an output port, while clearing the bit to 0 makes the pin an input port.

Port B pin functions in mode 7 are shown in figure 9.14.

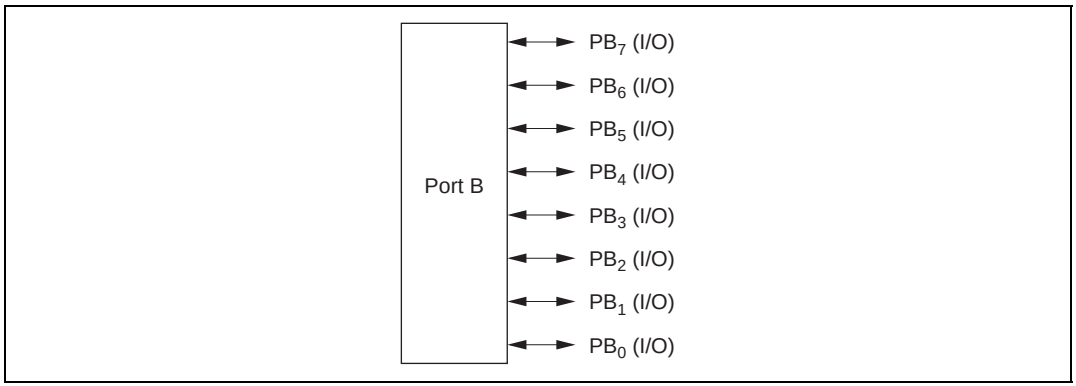


Figure 9.14 Port B Pin Functions (Mode 7)

Port B has a built-in MOS input pull-up function that can be controlled by software. This MOS input pull-up function can be used in modes 6 and 7, and can be specified as on or off on an individual bit basis.

When a PBDDR bit is cleared to 0 in mode 6 or 7, setting the corresponding PBPCR bit to 1 turns on the MOS input pull-up for that pin.

The MOS input pull-up function is in the off state after a reset, and in hardware standby mode. The prior state is retained in software standby mode.

Table 9.23 summarizes the MOS input pull-up states.

Table 9.23 MOS Input Pull-Up States (Port B)

Modes	Reset	Hardware Standby Mode	Software Standby Mode	In Other Operations
4, 5	Off	Off	Off	Off
6, 7			On/off	On/off

Legend

Off: MOS input pull-up is always off.

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

9.13.1 Overview

Port C is an 8-bit I/O port. Port C has an address bus output function, and the pin functions change according to the operating mode.

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

Figure 9.15 shows the port C pin configuration.

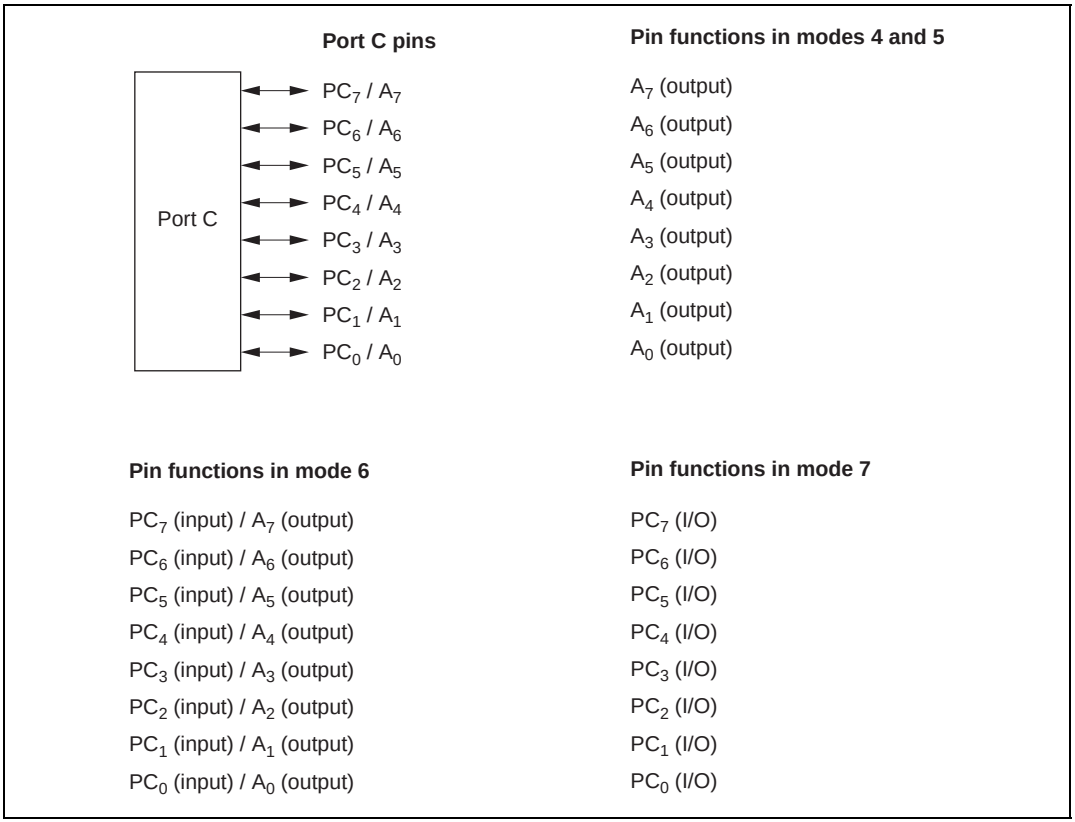


Figure 9.15 Port C Pin Functions

Table 9.24 shows the port C register configuration.

Table 9.24 Port C Registers

Name	Abbreviation	R/W	Initial Value	Address*
Port C data direction register	PCDDR	W	H'00	H'FEBB
Port C data register	PCDR	R/W	H'00	H'FF6B
Port C register	PORTC	R	Undefined	H'FF5B
Port C MOS pull-up control register	PCPCR	R/W	H'00	H'FF72

Note: * Lower 16 bits of the address.

Port C Data Direction Register (PCDDR)

Bit	:	7	6	5	4	3	2	1	0
		PC7DDR	PC6DDR	PC5DDR	PC4DDR	PC3DDR	PC2DDR	PC1DDR	PC0DDR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W	W

PCDDR is an 8-bit write-only register, the individual bits of which specify input or output for the 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 prior state in software standby mode. The OPE bit in SBYCR is used to select whether the address output pins retain their output state or become high-impedance when a transition is made to software standby mode.

- Modes 4 and 5
  The corresponding port C pins are address outputs irrespective of the value of the PCDDR bits.
- Mode 6
  Setting a PCDDR bit to 1 makes the corresponding port C pin an address output, while 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 clearing the bit to 0 makes the pin an input port.

Bit	:	7	6	5	4	3	2	1	0
		PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR
Initial value	:	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

PCDR is an 8-bit readable/writable register that stores output data for the port C pins (PC_7 to PC_0).

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

Port C Register (PORTC)

Bit	:	7	6	5	4	3	2	1	0
		PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

Note: * Determined by state of pins PC_7 to PC_0.

PORTC is an 8-bit read-only register that shows the pin states. It cannot be written to. Writing of output data for the port C pins (PC_7 to PC_0) 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. If a port C 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 pin states, as PCDDR and PCDR are initialized. PORTC retains its prior state in software standby mode.

Bit	:	7	6	5	4	3	2	1	0
		PC7PCR	PC6PCR	PC5PCR	PC4PCR	PC3PCR	PC2PCR	PC1PCR	PC0PCR
Initial value	:	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

PCPCR is an 8-bit readable/writable register that controls the MOS input pull-up function incorporated into port C on an individual bit basis.

When a PCDDR bit is cleared to 0 (input port setting) in mode 6 or 7, setting the corresponding PCPCR bit to 1 turns on the MOS input pull-up for the corresponding pin.

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

9.13.3 Pin Functions

Modes 4 and 5: In modes 4 and 5, port C pins are automatically designated as address outputs.

Port C pin functions in modes 4 and 5 are shown in figure 9.16.

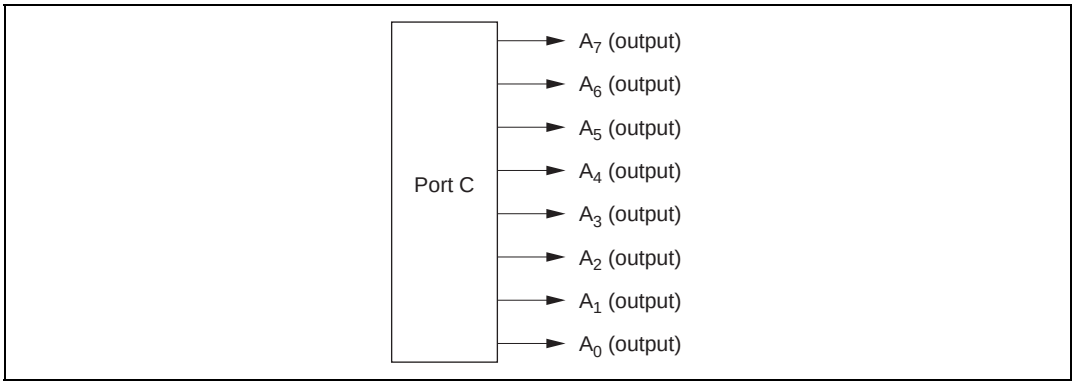


Figure 9.16 Port C Pin Functions (Modes 4 and 5)

an address output, while clearing the bit to 0 makes the pin an input port.

Port C pin functions in mode 6 are shown in figure 9.17.

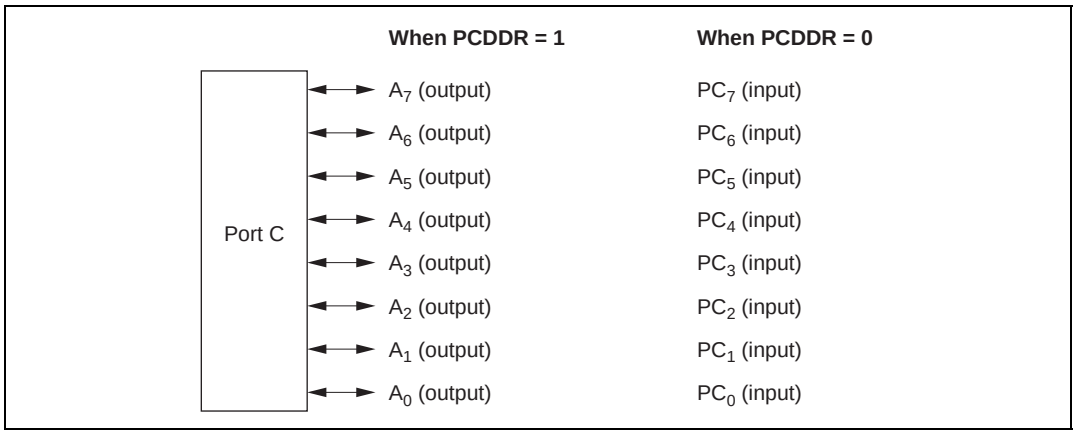


Figure 9.17 Port C Pin Functions (Mode 6)

Mode 7: In mode 7, port C pins function as I/O ports. Input or output can be specified for each pin on an individual bit basis. Setting a PCDDR bit to 1 makes the corresponding port C pin an output port, while clearing the bit to 0 makes the pin an input port.

Port C pin functions in mode 7 are shown in figure 9.18.

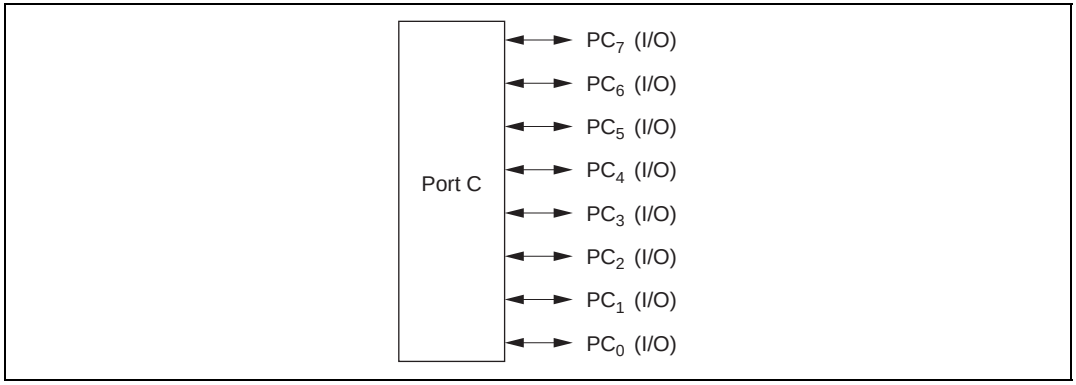


Figure 9.18 Port C Pin Functions (Mode 7)

Port C has a built-in MOS input pull-up function that can be controlled by software. This MOS input pull-up function can be used in modes 6 and 7, and can be specified as on or off on an individual bit basis.

When a PCDDR bit is cleared to 0 in mode 6 or 7, setting the corresponding PCPCR bit to 1 turns on the MOS input pull-up for that pin.

The MOS input pull-up function is in the off state after a reset, and in hardware standby mode. The prior state is retained in software standby mode.

Table 9.25 summarizes the MOS input pull-up states.

Table 9.25 MOS Input Pull-Up States (Port C)

Modes	Reset	Hardware Standby Mode	Software Standby Mode	In Other Operations
4, 5	Off	Off	Off	Off
6, 7			On/off	On/off

Legend

Off: MOS input pull-up is always off.

On/off: On when PCDDR = 0 and PCPCR = 1; otherwise off.

9.14.1 Overview

Port D is an 8-bit I/O port. Port D has a data bus I/O function, and the pin functions change according to the operating mode.

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

Figure 9.19 shows the port D pin configuration.

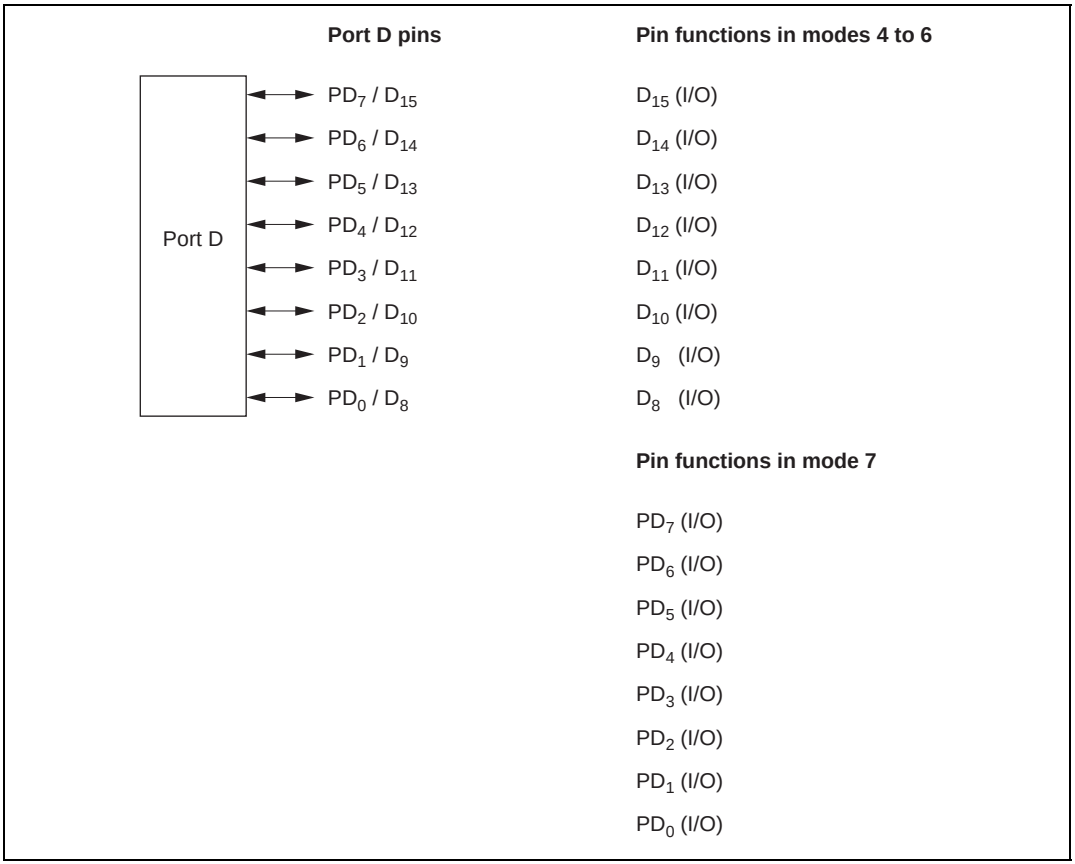


Figure 9.19 Port D Pin Functions

Table 9.26 shows the port D register configuration.

Table 9.26 Port D Registers

Name	Abbreviation	R/W	Initial Value	Address*
Port D data direction register	PDDDR	W	H'00	H'FEBC
Port D data register	PDDR	R/W	H'00	H'FF6C
Port D register	PORTD	R	Undefined	H'FF5C
Port D MOS pull-up control register	PDPCR	R/W	H'00	H'FF73

Note: * Lower 16 bits of the address.

Port D Data Direction Register (PDDDR)

Bit	:	7	6	5	4	3	2	1	0
		PD7DDR	PD6DDR	PD5DDR	PD4DDR	PD3DDR	PD2DDR	PD1DDR	PD0DDR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W	W

PDDDR is an 8-bit write-only register, the individual bits of which specify input or output for the 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 prior state in software standby mode.

- Modes 4 to 6

  The input/output direction specification by PDDDR is ignored, and port D is automatically designated for data I/O.

- Mode 7

  Setting a PDDDR bit to 1 makes the corresponding port D pin an output port, while clearing the bit to 0 makes the pin an input port.

Bit	:	7	6	5	4	3	2	1	0
		PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR
Initial value	:	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

PDDR is an 8-bit readable/writable register that stores output data for the port D pins (PD_7 to PD_0).

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

Port D Register (PORTD)

Bit	:	7	6	5	4	3	2	1	0
		PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

Note: * Determined by state of pins PD_7 to PD_0.

PORTD is an 8-bit read-only register that shows the pin states. It cannot be written to. Writing of output data for the port D pins (PD_7 to PD_0) must always be performed on PDDR.

If a port D read is performed while PDDDR bits are set to 1, the PDDR values are read. If a port D 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 pin states, as PDDDR and PDDR are initialized. PORTD retains its prior state in software standby mode.

Bit	:	7	6	5	4	3	2	1	0
		PD7PCR	PD6PCR	PD5PCR	PD4PCR	PD3PCR	PD2PCR	PD1PCR	PD0PCR
Initial value	:	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

PDPCR is an 8-bit readable/writable register that controls the MOS input pull-up function 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 corresponding 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 prior state in software standby mode.

9.14.3 Pin Functions

Modes 4 to 6: In modes 4 to 6, port D pins are automatically designated as data I/O pins.

Port D pin functions in modes 4 to 6 are shown in figure 9.20.

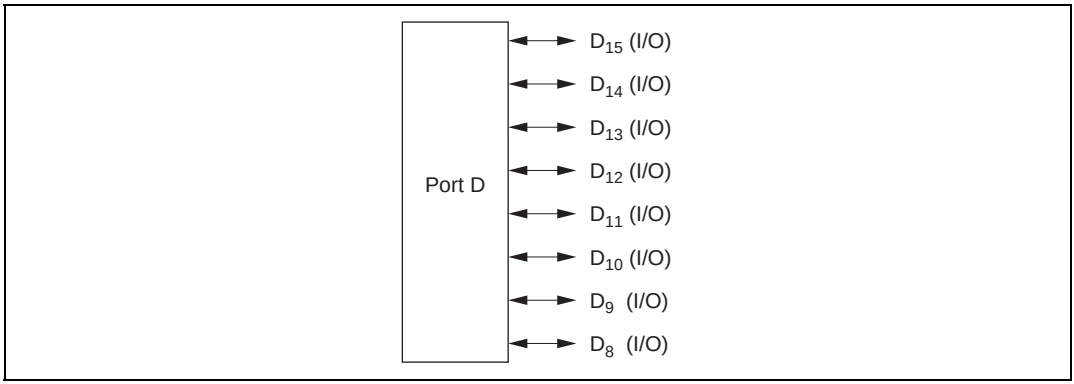


Figure 9.20 Port D Pin Functions (Modes 4 to 6)

port, while clearing the bit to 0 makes the pin an input port.

Port D pin functions in mode 7 are shown in figure 9.21.

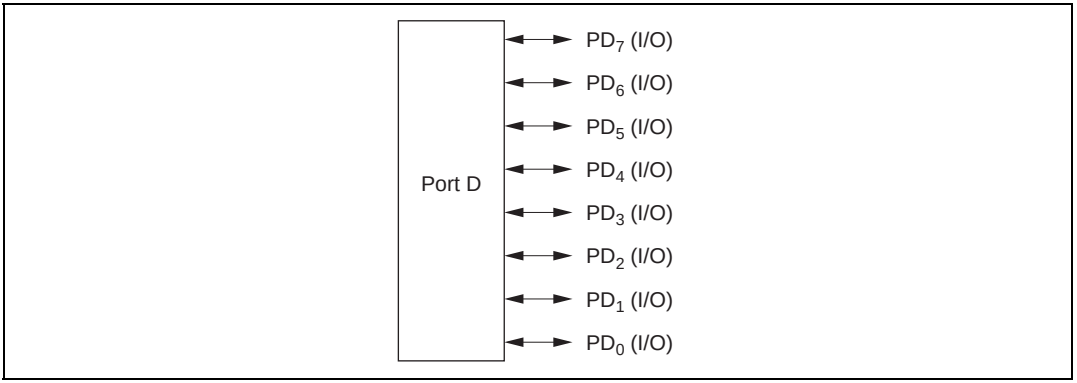


Figure 9.21 Port D Pin Functions (Mode 7)

Port D has a built-in MOS input pull-up function that can be controlled by software. This MOS input pull-up function can be used in mode 7, and can be specified as on or off on an individual bit basis.

When a PDDDR bit is cleared to 0 in mode 7, setting the corresponding PDPCR bit to 1 turns on the MOS input pull-up for that pin.

The MOS input pull-up function is in the off state after a reset, and in hardware standby mode. The prior state is retained in software standby mode.

Table 9.27 summarizes the MOS input pull-up states.

Table 9.27 MOS Input Pull-Up States (Port D)

Modes	Reset	Hardware Standby Mode	Software Standby Mode	In Other Operations
4 to 6	Off	Off	Off	Off
7			On/off	On/off

Legend

Off: MOS input pull-up is always off.

On/off: On when PDDDR = 0 and PDPCR = 1; otherwise off.

9.15.1 Overview

Port E is an 8-bit I/O port. Port E has a data bus I/O function, and the pin functions change according to the operating mode and whether 8-bit or 16-bit bus mode is selected.

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

Figure 9.22 shows the port E pin configuration.

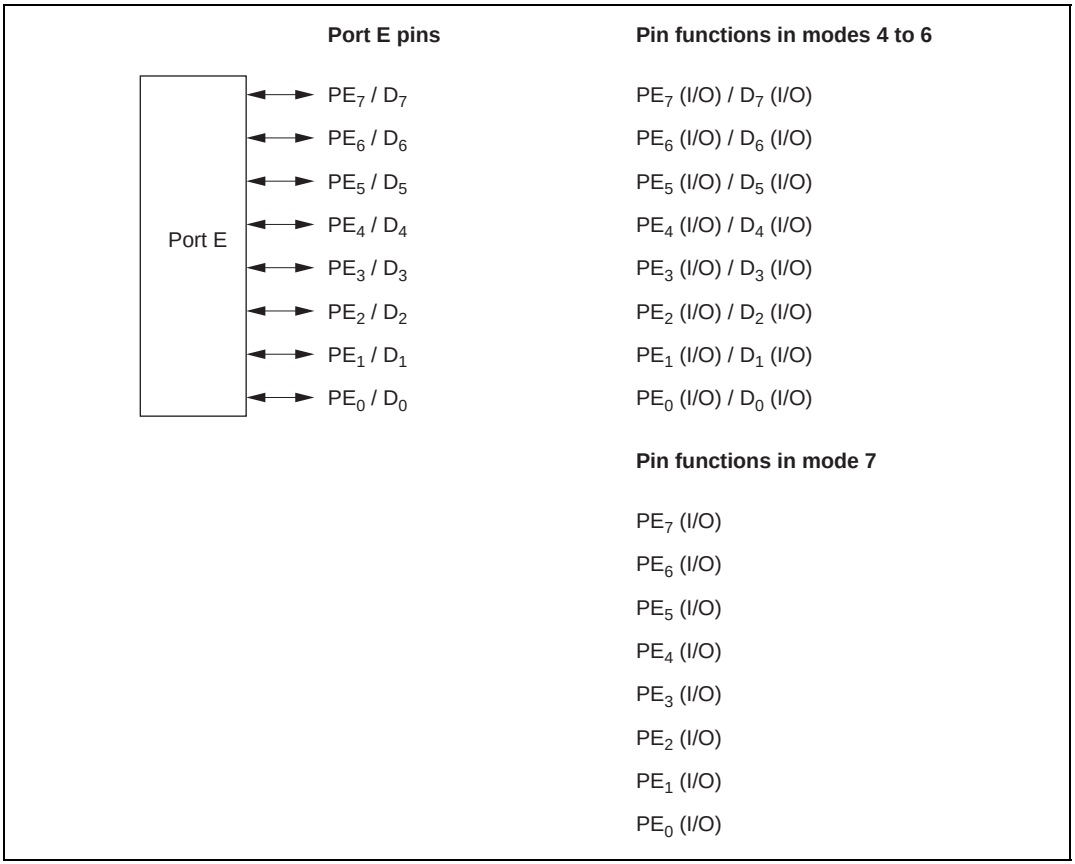


Figure 9.22 Port E Pin Functions

Table 9.28 shows the port E register configuration.

Table 9.28 Port E Registers

Name	Abbreviation	R/W	Initial Value	Address*
Port E data direction register	PEDDR	W	H'00	H'FEBD
Port E data register	PEDR	R/W	H'00	H'FF6D
Port E register	PORTE	R	Undefined	H'FF5D
Port E MOS pull-up control register	PEPCR	R/W	H'00	H'FF74

Note: * Lower 16 bits of the address.

Port E Data Direction Register (PEDDR)

Bit	:	7	6	5	4	3	2	1	0
		PE7DDR	PE6DDR	PE5DDR	PE4DDR	PE3DDR	PE2DDR	PE1DDR	PE0DDR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W	W

PEDDR is an 8-bit write-only register, the individual bits of which specify input or output for the 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 prior state in 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 PEDDR bit to 1 makes the corresponding port E pin an output port, while clearing the bit to 0 makes the pin an input port.

  When 16-bit bus mode has been selected, the input/output direction specification by PEDDR is ignored, and port E is designated for data I/O.

  For details of 8-bit and 16-bit bus modes, see section 6, Bus Controller.

- Mode 7

  Setting a PEDDR bit to 1 makes the corresponding port E pin an output port, while clearing the bit to 0 makes the pin an input port.

Bit	:	7	6	5	4	3	2	1	0
		PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR
Initial value	:	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

PEDR is an 8-bit readable/writable register that stores output data for the port E pins (PE_7 to PE_0).

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

Port E Register (PORTE)

Bit	:	7	6	5	4	3	2	1	0
		PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

Note: * Determined by state of pins PE_7 to PE_0.

PORTE is an 8-bit read-only register that shows the pin states. It cannot be written to. Writing of output data for the port E pins (PE_7 to PE_0) 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. If a port E 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 pin states, as PEDDR and PEDR are initialized. PORTE retains its prior state in software standby mode.

Bit	:	7	6	5	4	3	2	1	0
		PE7PCR	PE6PCR	PE5PCR	PE4PCR	PE3PCR	PE2PCR	PE1PCR	PE0PCR
Initial value	:	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

PEPCR is an 8-bit readable/writable register that controls the MOS input pull-up function incorporated into port E on an individual bit basis.

When a PEDDR bit is cleared to 0 (input port setting) in mode 4, 5, or 6 with 8-bit bus mode selected, or in mode 7, setting the corresponding PEPCR bit to 1 turns on the MOS input pull-up for the corresponding pin.

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

Modes 4 to 6: In modes 4 to 6, when 8-bit access is designated and 8-bit bus mode is selected, port E pins are automatically designated as I/O ports. Setting a PEDDR bit to 1 makes the corresponding port E pin an output port, while clearing the bit to 0 makes the pin an input port.

When 16-bit bus mode is selected, the input/output direction specification by PEDDR is ignored, and port E is designated for data I/O.

Port E pin functions in modes 4 to 6 are shown in figure 9.23.

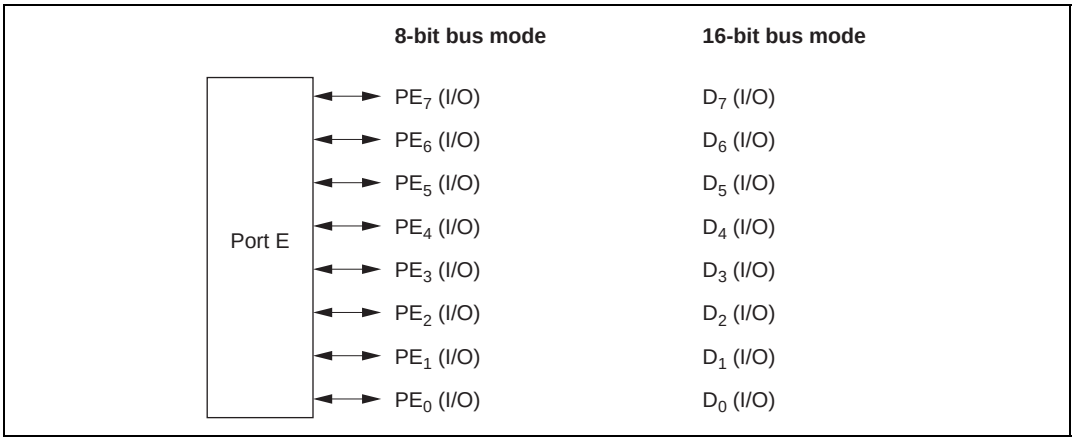


Figure 9.23 Port E Pin Functions (Modes 4 to 6)

while clearing the bit to 0 makes the pin an input port.

Port E pin functions in mode 7 are shown in figure 9.24.

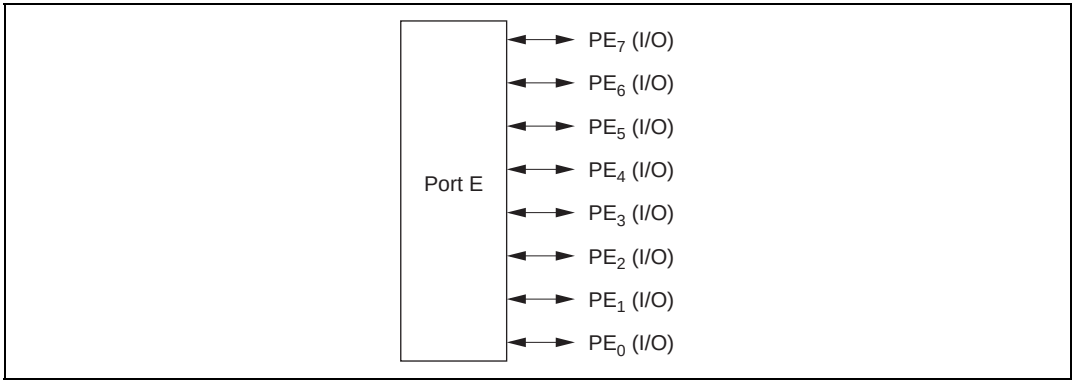


Figure 9.24 Port E Pin Functions (Mode 7)

Port E has a built-in MOS input pull-up function that can be controlled by software. This MOS input pull-up function can be used in modes 4, 5, and 6 when 8-bit bus mode is selected, or in mode 7, and can be specified as on or off on an individual bit basis.

When a PEDDR bit is cleared to 0 in mode 4, 5, or 6 when 8-bit bus mode is selected, or in mode 7, setting the corresponding PEPCR bit to 1 turns on the MOS input pull-up for that pin.

The MOS input pull-up function is in the off state after a reset, and in hardware standby mode. The prior state is retained in software standby mode.

Table 9.29 summarizes the MOS input pull-up states.

Table 9.29 MOS Input Pull-Up States (Port E)

Modes		Reset	Hardware Standby Mode	Software Standby Mode	In Other Operations
7		Off	Off	On/off	On/off
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.

9.16.1 Overview

Port F is an 8-bit I/O port. Port F pins also function as bus control signal input/output pins ($\overline{AS}$, $\overline{RD}$, $\overline{HWR}$, $\overline{LWR}$, $\overline{LCAS}$, $\overline{BREQO}$, $\overline{BREQ}$, and $\overline{BACK}$) and the system clock (ϕ) output pin. The $\overline{AS}$, $\overline{LWR}$, and $\overline{BREQO}$ output pins can be switched by means of settings in PFCR2 and SYSCR.

Figure 9.25 shows the port F pin configuration.

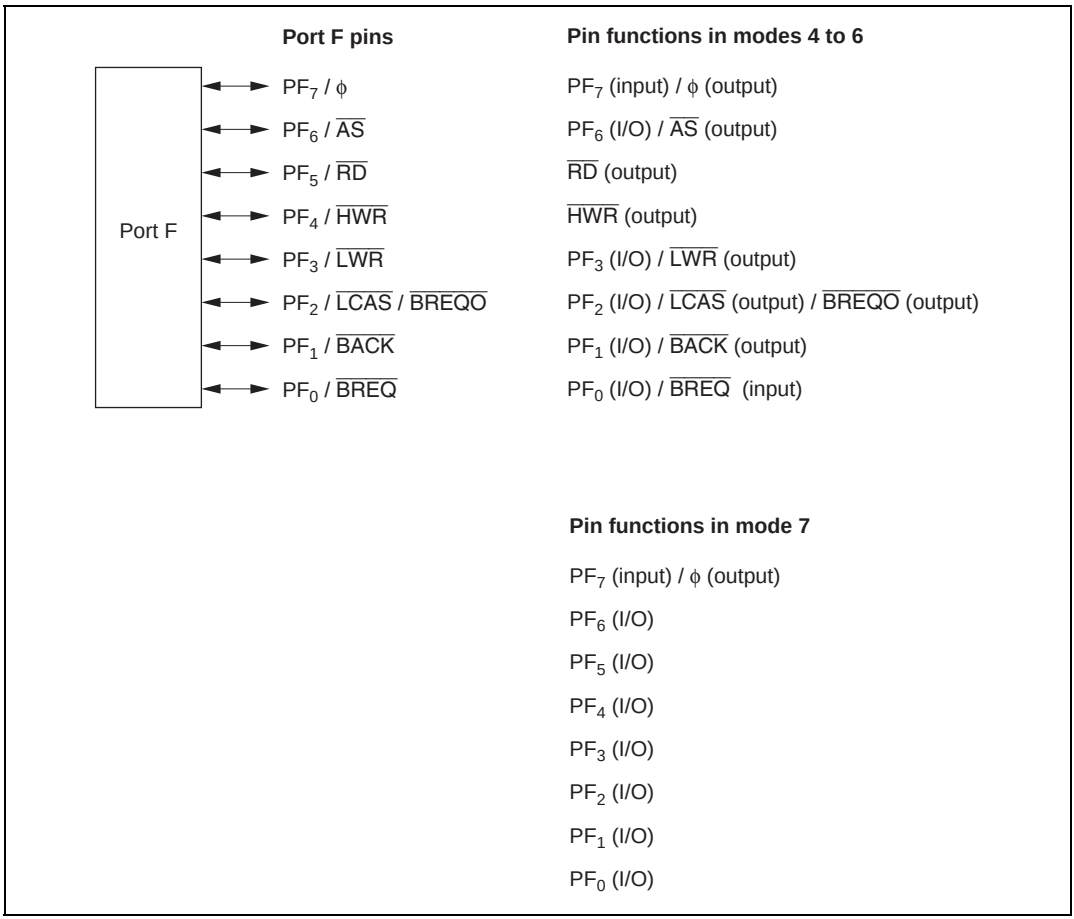


Figure 9.25 Port F Pin Functions

Table 9.30 shows the port F register configuration.

Table 9.30 Port F Registers

Name	Abbreviation	R/W	Initial Value	Address*1
Port F data direction register	PFDDR	W	H'80/H'00*2	H'FEBE
Port F data register	PFDR	R/W	H'00	H'FF6E
Port F register	PORTF	R	Undefined	H'FF5E
Port function control register 2	PFCR2	R/W	H'30	H'FFAC
System control register	SYSCR	R/W	H'01	H'FF39

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	0
	PF7DDR	PF6DDR	PF5DDR	PF4DDR	PF3DDR	PF2DDR	PF1DDR	PF0DDR
Modes 4 to 6								
Initial value :	1	0	0	0	0	0	0	0
R/W :	W	W	W	W	W	W	W	W
Mode 7								
Initial value :	0	0	0	0	0	0	0	0
R/W :	W	W	W	W	W	W	W	W

PFDDR is an 8-bit write-only register, the individual bits of which specify input or output for the 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 6, and to H'00 in mode 7. It retains its prior state in software standby mode. The OPE bit in SBYCR is used to select whether the bus control output pins retain their output state or become high-impedance when a transition is made to software standby mode.

Bit	:	7	6	5	4	3	2	1	0
		PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1DR	PF0DR
Initial value	:	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

PFDR is an 8-bit readable/writable register that stores output data for the port F pins (PF_7 to PF_0).

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

Port F Register (PORTF)

Bit	:	7	6	5	4	3	2	1	0
		PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

Note: * Determined by state of pins PF_7 to PF_0.

PORTF is an 8-bit read-only register that shows the pin states. It cannot be written to. Writing of output data for the port F pins (PF_7 to PF_0) 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. If a port F 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 pin states, as PFDDR and PFDR are initialized. PORTF retains its prior state in software standby mode.

Bit	:	7	6	5	4	3	2	1	0
		WAITPS	BREQOPS	CS167E	CS25E	ASOD	—	—	—
Initial value	:	0	0	1	1	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R	R	R

Bit 7—WAIT Pin Select (WAITPS): Selects the $\overline{\text{WAIT}}$ input pin. For details, see section 9.6, Port 5.

Bit 6—BREQO Pin Select (BREQOPS): Seelects the $\overline{\text{BREQO}}$ output pin. Set the BREQOPS bit before setting the BREQOE bit in BCRL to 1.

Bit 6 BREQOPS	Description	
0	$\overline{\text{BREQO}}$ output pin PF_2	(Initial value)
1	$\overline{\text{BREQO}}$ output pin $P5_3$	

Bit 5—CS167 Enable (CS167E): Enables or disables $\overline{CS}_1$, $\overline{CS}_6$, and $\overline{CS}_7$ output. For details, see section 9.7, Port 6, and section 9.17, Port G.

Bit 4—CS25 Enable (CS25E): Enables or disables $\overline{CS}_2$, $\overline{CS}_3$, $\overline{CS}_4$, and $\overline{CS}_5$ output. For details, see section 9.7, Port 6, and section 9.17, Port G.

Bit 3—AS Output Disable (ASOD): Enables or disables $\overline{\text{AS}}$ output. This bit is valid in modes 4 to 6.

Bit 3 ASOD	Description	
0	PF_6 is used as $\overline{\text{AS}}$ output pin	(Initial value)
1	PF_6 is designated as I/O port, and does not function as $\overline{\text{AS}}$ output pin	

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

Bit	:	7	6	5	4	3	2	1	0
		—	—	INTM1	INTM0	NMIEG	LWROD	IRQPAS	RAME
Initial value	:	0	0	0	0	0	0	0	1
R/W	:	R/W	—	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, controls the $\overline{\text{LWR}}$ pin, switches the $\overline{\text{IRQ}}_4$ to $\overline{\text{IRQ}}_7$ input pins, and selects the detected edge for NMI. SYSCR is initialized to H'01 by a reset, and in hardware standby mode. It is not initialized in software standby mode.

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

Bit 3—NMI Edge Select (NMIEG): Selects the input edge for the NMI pin. For details, see section 5, Interrupt Controller.

Bit 2—LWR Output Disable (LWROD): Enables or disables $\overline{\text{LWR}}$ output. This bit is valid in modes 4 to 6.

Bit 2 LWROD	Description	
0	PF_3 is designated as $\overline{\text{LWR}}$ output pin	(Initial value)
1	PF_3 is designated as I/O port, and does not function as $\overline{\text{LWR}}$ output pin	

Bit 1—IRQ Port Switching Select (IRQPAS): Selects switching of input pins for $\overline{\text{IRQ}}_4$ to $\overline{\text{IRQ}}_7$. For details, see section 9.6, Port 5.

Bit 0—RAM Enable (RAME): Enables or disables on-chip RAM. For details, see section 18, RAM.

Port F pins also function as bus control signal input/output pins (AS, RD, HWR, LWR, LCAS, $\overline{\text{BREQO}}$, $\overline{\text{BREQ}}$, and $\overline{\text{BACK}}$) and the system clock (ϕ) output pin. The pin functions differ between modes 4 to 6, and mode 7. Port F pin functions are shown in table 9.31.

Table 9.31 Port F Pin Functions

Pin	Selection Method and Pin Functions

PF_7/ϕ

The pin function is switched as shown below according to bit PF7DDR.

PF7DDR	0	1
Pin function	PF_7 input pin	ϕ output pin

$PF_6/\overline{\text{AS}}$

The pin function is switched as shown below according to the operating mode, bit PF6DDR, and bit ASOD in PFCR2.

Operating Mode	Modes 4 to 6			Mode 7	
ASOD	0	1		—	
PF6DDR	—	0	1	0	1
Pin function	$\overline{\text{AS}}$ output pin	PF_6 input pin	PF_6 output pin	PF_6 input pin	PF_6 output pin

$PF_5/\overline{\text{RD}}$

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	1
Pin function	$\overline{\text{RD}}$ output pin	PF_5 input pin	PF_5 output pin

$PF_4/\overline{\text{HWR}}$

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	1
Pin function	$\overline{\text{HWR}}$ output pin	PF_4 input pin	PF_4 output pin

bit PF3DDR, and bit LWROD in SYSCR.

Operating Mode	Modes 4 to 6			Mode 7	
LWROD	0	1		—	
PF3DDR	—	0	1	0	1
Pin function	$\overline{\text{LWR}}$ output pin	PF_3 input pin	PF_3 output pin	PF_3 input pin	PF_3 output pin

$PF_2/\overline{\text{LCAS}}/\overline{\text{BREQO}}$

The pin function is switched as shown below according to the combination of the operating mode, and bits RMTS2 to RMTS0, BREQOE, ABW5 to ABW2, BREQOPS, and PF2DDR.

Operating Mode	Modes 4 to 6				Mode 7	
[DRAM space setting] · [16-bit access setting]	0			1	—	
[BREQOE · $\overline{\text{BREQOPS}}$]	0		1	—	—	
PF2DDR	0	1	—	—	0	1
Pin function	PF_2 input pin	PF_2 output pin	$\overline{\text{BREQO}}$ output pin	$\overline{\text{LCAS}}$ output pin	PF_2 input pin	PF_2 output pin

$PF_1/\overline{\text{BACK}}$

The pin function is switched as shown below according to the combination of the operating mode, and bits BRLE and PF1DDR.

Operating Mode	Modes 4 to 6			Mode 7	
BRLE	0		1	—	
PF1DDR	0	1	—	0	1
Pin function	PF_1 input pin	PF_1 output pin	$\overline{\text{BACK}}$ output pin	PF_1 input pin	PF_1 output pin

the operating mode, and bits BRLE and PF0DDR.

Operating Mode	Modes 4 to 6			Mode 7	
BRLE	0		1	—	
PF0DDR	0	1	—	0	1
Pin function	PF_0 input pin	PF_0 output pin	$\overline{BREQ}$ input pin	PF_0 input pin	PF_0 output pin

9.17 Port G

9.17.1 Overview

Port G is a 5-bit I/O port. Port G pins also function as bus control signal output pins ($\overline{CS0}$ to $\overline{CS3}$, and $\overline{CAS}$). Enabling or disabling of $\overline{CS}_1$ to $\overline{CS}_2$ output can be changed by a setting in PFCR2.

Figure 9.26 shows the port G pin configuration.

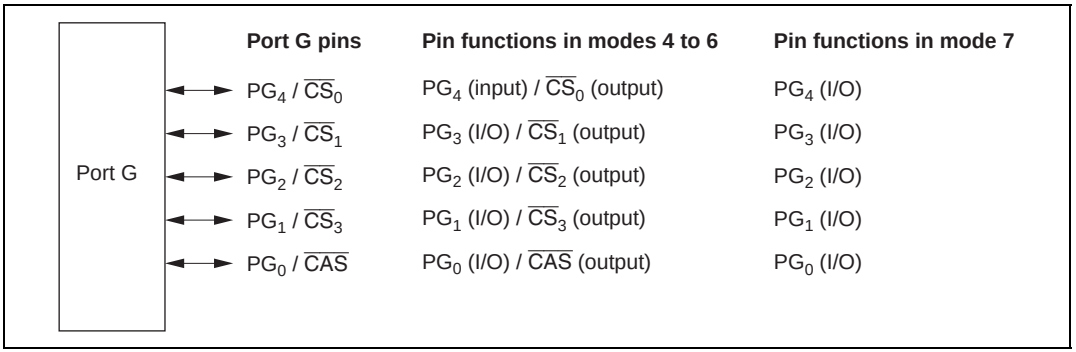


Figure 9.26 Port G Pin Functions

Table 9.32 shows the port G register configuration.

Table 9.32 Port G Registers

Name	Abbreviation	R/W	Initial Value*2	Address*1
Port G data direction register	PGDDR	W	H'10/H'00*3	H'FEBF
Port G data register	PGDR	R/W	H'00	H'FF6F
Port G register	PORTG	R	Undefined	H'FF5F
Port function control register 2	PFCR2	R/W	H'30	H'FFAC

Notes: 1. Lower 16 bits of the address.
2. Value of bits 4 to 0.
3. Initial value depends on the mode.

Port G Data Direction Register (PGDDR)

Bit :	7	6	5	4	3	2	1	0
	—	—	—	PG4DDR	PG3DDR	PG2DDR	PG1DDR	PG0DDR
Modes 4 and 5								
Initial value :	Undefined	Undefined	Undefined	1	0	0	0	0
R/W :	—	—	—	W	W	W	W	W
Modes 6 and 7								
Initial value :	Undefined	Undefined	Undefined	0	0	0	0	0
R/W :	—	—	—	W	W	W	W	W

PGDDR is an 8-bit write-only register, the individual bits of which specify input or output for the pins of port G. PGDDR cannot be read, and bits 7 to 5 are reserved. If PGDDR is read, an undefined value will be read.

The PG4DDR bit is initialized by a reset, and in hardware standby mode, to 1 in modes 4 and 5, and to 0 in modes 6 and 7. PGDDR retains its prior state in software standby mode. The OPE bit in SBYCR is used to select whether the bus control output pins retain their output state or become high-impedance when a transition is made to software standby mode.

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	PG4DR	PG3DR	PG2DR	PG1DR	PG0DR
Initial value	:	Undefined	Undefined	Undefined	0	0	0	0	0
R/W	:	—	—	—	R/W	R/W	R/W	R/W	R/W

PGDR is an 8-bit readable/writable register that stores output data for the port G pins (PG_4 to PG_0).

Bits 7 to 5 are reserved; they return an undefined value if read, and cannot be modified.

PGDR is initialized to H'00 (bits 4 to 0) by a reset, and in hardware standby mode. It retains its prior state in software standby mode.

Port G Register (PORTG)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	PG4	PG3	PG2	PG1	PG0
Initial value	:	Undefined	Undefined	Undefined	—*	—*	—*	—*	—*
R/W	:	—	—	—	R	R	R	R	R

Note: * Determined by state of pins PG_4 to PG_0.

PORTG is an 8-bit read-only register that shows the pin states. It cannot be written to. Writing of output data for the port G pins (PG_4 to PG_0) must always be performed on PGDR.

Bits 7 to 5 are reserved; they return an undefined value if read, and cannot be modified.

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

After a reset and in hardware standby mode, PORTG contents are determined by the pin states, as PGDDR and PGDR are initialized. PORTG retains its prior state in software standby mode.

Bit	:	7	6	5	4	3	2	1	0
		WAITPS	BREQOPS	CS167E	CS25E	ASOD	—	—	—
Initial value	:	0	0	1	1	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R	R	R

PFCR2 is an 8-bit readable/writable register that performs I/O port control. PFCR2 is initialized to H'30 by a reset, and in hardware standby mode.

Bit 7—WAIT Pin Select (WAITPS): Selects the $\overline{\text{WAIT}}$ input pin. For details, see section 9.6, Port 5.

Bit 6—BREQO Pin Select (BREQOPS): Selects the $\overline{\text{BREQO}}$ output pin. For details, see section 9.6, Port 5.

Bit 5—CS167 Enable (CS167E): Enables or disables $\overline{\text{CS}}_1$, $\overline{\text{CS}}_6$, and $\overline{\text{CS}}_7$ output. Change the CS167E setting only when the DDR bits are cleared to 0.

Bit 5 CS167E	Description	
0	$\overline{\text{CS}}_1$, $\overline{\text{CS}}_6$, and $\overline{\text{CS}}_7$ output disabled (can be used as I/O ports)	
1	$\overline{\text{CS}}_1$, $\overline{\text{CS}}_6$, and $\overline{\text{CS}}_7$ output enabled	(Initial value)

Bit 4—CS25 Enable (CS25E): Enables or disables $\overline{\text{CS}}_2$, $\overline{\text{CS}}_3$, $\overline{\text{CS}}_4$, and $\overline{\text{CS}}_5$ output. Change the CS25E setting only when the DDR bits are cleared to 0.

Bit 4 CS25E	Description	
0	$\overline{\text{CS}}_2$, $\overline{\text{CS}}_3$, $\overline{\text{CS}}_4$, and $\overline{\text{CS}}_5$ output disabled (can be used as I/O ports)	
1	$\overline{\text{CS}}_2$, $\overline{\text{CS}}_3$, $\overline{\text{CS}}_4$, and $\overline{\text{CS}}_5$ output enabled	(Initial value)

Bit 3—AS Output Disable (ASOD): Enables or disables $\overline{\text{AS}}$ output. For details, see section 9.16, Port F.

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

Port G pins also function as bus control signal output pins (CS0 to CS3, and CAS). The pin functions are different between in mode 7, and modes 4 to 6. Port G pin functions are shown in table 9.33.

Table 9.33 Port G Pin Functions

Pin — **Selection Method and Pin Functions**

$PG_4/\overline{CS_0}$

The pin function is switched as shown below according to the operating mode and bit PG4DDR.

Operating Mode	Modes 4 to 6		Mode 7	
PG4DDR	0	1	0	1
Pin function	PG_4 input pin	$\overline{CS_0}$ output pin	PG_4 input pin	PG_4 output pin

$PG_3/\overline{CS_1}$

The pin function is switched as shown below according to the operating mode and bits PG3DDR and CS167E.

Operating Mode	Modes 4 to 6				Mode 7	
CS167E	0		1		0	1
PG3DDR	0	1	0	1	—	—
Pin function	PG_3 input pin	PG_3 output pin	PG_3 input pin	$\overline{CS_1}$ output pin	PG_3 input pin	PG_3 output pin

$PG_2/\overline{CS_2}$

The pin function is switched as shown below according to the operating mode and bits PG2DDR and CS25E.

Operating Mode	Modes 4 to 6				Mode 7	
CS25E	0		1		0	1
PG2DDR	0	1	0	1	—	—
Pin function	PG_2 input pin	PG_2 output pin	PG_2 input pin	$\overline{CS_2}$ output pin	PG_2 input pin	PG_2 output pin

and bits PG1DDR and CS25E.

Operating Mode	Modes 4 to 6				Mode 7	
CS25E	0		1		0	1
PG1DDR	0	1	0	1	—	—
Pin function	PG_1 input pin	PG_1 output pin	PG_1 input pin	$\overline{CS}_3$ output pin	PG_1 input pin	PG_1 output pin

$PG_0/\overline{CAS}$

The pin function is switched as shown below according to the combination of the operating mode and bits RMTS2 to RMTS0 and PG0DDR.

Operating Mode	Modes 4 to 6			Mode 7	
RMTS2 to RMTS0	B'000, B'100 to B'111		B'001 to B'011	—	
PG0DDR	0	1	—	0	1
Pin function	PG_0 input pin	PG_0 output pin	$\overline{CAS}$ output pin	PG_0 input pin	PG_0 output pin

10.1 Overview

The chip has an on-chip 16-bit timer pulse unit (TPU) that comprises six 16-bit timer channels.

10.1.1 Features

- Maximum 16-pulse input/output
  - A total of 16 timer general registers (TGRs) are provided (four each for channels 0 and 3, and two each for channels 1, 2, 4, and 5), each of which can be set independently as an 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 detection
  - Counter clear operation: Counter clearing possible by compare match or input capture
  - 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 synchronous operation
- 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 1 (channel 4) overflow/underflow
- Fast access via internal 16-bit bus
  - Fast access is possible via a 16-bit bus interface

overflow interrupt can be requested independently

— For channels 1, 2, 4, and 5, two compare match/input capture dual-function interrupts, one overflow interrupt, and one underflow interrupt can be requested independently

- Automatic transfer of register data
  — Block transfer, 1-word data transfer, and 1-byte data transfer possible by data transfer controller (DTC) or DMA controller (DMAC) activation
- Programmable pulse generator (PPG) output trigger can be generated
  — Channel 0 to 3 compare match/input capture signals can be used as PPG output trigger
- 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 converter conversion start trigger
- Module stop mode can be set
  — As the initial setting, TPU operation is halted. Register access is enabled by exiting module stop mode

Table 10.1 TPU Functions

Item		Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5
Count clock		φ/1 φ/4 φ/16 φ/64 TCLKA TCLKB TCLKC TCLKD	φ/1 φ/4 φ/16 φ/64 φ/256 TCLKA TCLKB	φ/1 φ/4 φ/16 φ/64 φ/1024 TCLKA TCLKB TCLKC	φ/1 φ/4 φ/16 φ/64 φ/256 φ/1024 φ/4096 TCLKA	φ/1 φ/4 φ/16 φ/64 φ/1024 TCLKA TCLKC	φ/1 φ/4 φ/16 φ/64 φ/256 TCLKA TCLKC TCLKD
General registers		TGR0A TGR0B	TGR1A TGR1B	TGR2A TGR2B	TGR3A TGR3B	TGR4A TGR4B	TGR5A TGR5B
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	TIOCA5 TIOCB5
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	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		○	—	—	○	—	—

activation	compare match or input capture	compare match or input capture	compare match or input capture	compare match or input capture	compare match or input capture	compare match or input capture
DTC activation	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	TGR compare match or input capture
A/D conversion start trigger	TGR0A compare match or input capture	TGR1A compare match or input capture	TGR2A compare match or input capture	TGR3A compare match or input capture	TGR4A compare match or input capture	TGR5A compare match or input capture
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 sources • Compare match or input capture 5A • Compare match or input capture 5B • Overflow • Underflow

Legend:

○: Possible

—: Not possible

Figure 10.1 shows a block diagram of the TPU.

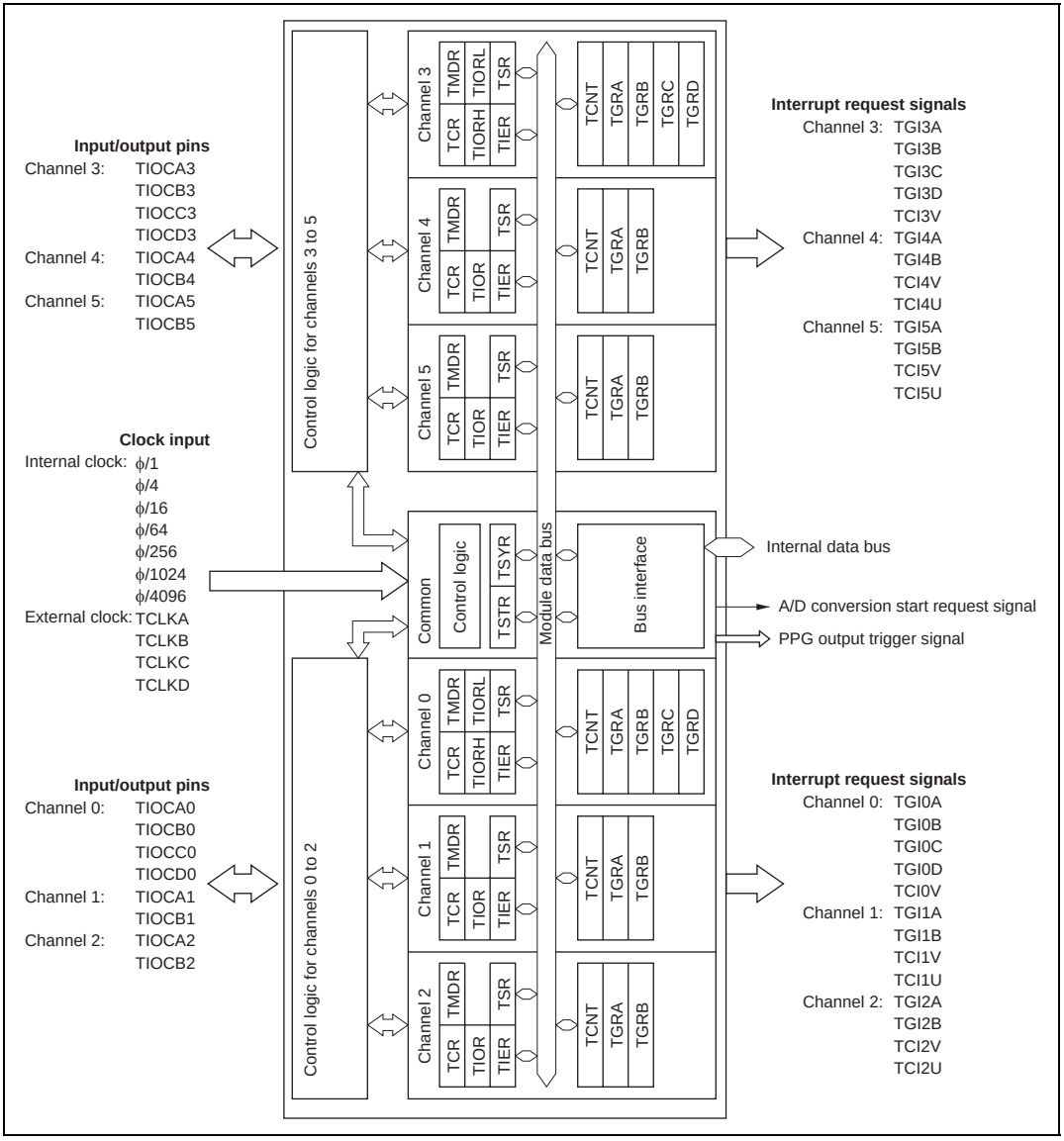


Figure 10.1 Block Diagram of TPU

Table 10.2 summarizes the TPU pins.

Table 10.2 TPU Pins

Channel	Name	Symbol	I/O	Function
All	Clock input A	TCLKA	Input	External clock A input pin (Channel 1 and 5 phase counting mode A phase input)
	Clock input B	TCLKB	Input	External clock B input pin (Channel 1 and 5 phase counting mode B phase input)
	Clock input C	TCLKC	Input	External clock C input pin (Channel 2 and 4 phase counting mode A phase input)
	Clock input D	TCLKD	Input	External clock D input pin (Channel 2 and 4 phase counting mode B phase input)
0	Input capture/out compare match A0	TIOCA0	I/O	TGR0A input capture input/output compare output/PWM output pin
	Input capture/out compare match B0	TIOCB0	I/O	TGR0B input capture input/output compare output/PWM output pin
	Input capture/out compare match C0	TIOCC0	I/O	TGR0C input capture input/output compare output/PWM output pin
	Input capture/out compare match D0	TIOCD0	I/O	TGR0D input capture input/output compare output/PWM output pin
1	Input capture/out compare match A1	TIOCA1	I/O	TGR1A input capture input/output compare output/PWM output pin
	Input capture/out compare match B1	TIOCB1	I/O	TGR1B input capture input/output compare output/PWM output pin
2	Input capture/out compare match A2	TIOCA2	I/O	TGR2A input capture input/output compare output/PWM output pin
	Input capture/out compare match B2	TIOCB2	I/O	TGR2B input capture input/output compare output/PWM output pin

	compare match A3			output/PWM output pin
	Input capture/out compare match B3	TIOCB3	I/O	TGR3B input capture input/output compare output/PWM output pin
	Input capture/out compare match C3	TIOCC3	I/O	TGR3C input capture input/output compare output/PWM output pin
	Input capture/out compare match D3	TIOCD3	I/O	TGR3D input capture input/output compare output/PWM output pin
4	Input capture/out compare match A4	TIOCA4	I/O	TGR4A input capture input/output compare output/PWM output pin
	Input capture/out compare match B4	TIOCB4	I/O	TGR4B input capture input/output compare output/PWM output pin
5	Input capture/out compare match A5	TIOCA5	I/O	TGR5A input capture input/output compare output/PWM output pin
	Input capture/out compare match B5	TIOCB5	I/O	TGR5B input capture input/output compare output/PWM output pin

Table 10.3 summarizes the TPU registers.

Table 10.3 TPU Registers

Channel	Name	Abbreviation	R/W	Initial Value	Address*1
0	Timer control register 0	TCR0	R/W	H'00	H'FFD0
	Timer mode register 0	TMDR0	R/W	H'C0	H'FFD1
	Timer I/O control register 0H	TIOR0H	R/W	H'00	H'FFD2
	Timer I/O control register 0L	TIOR0L	R/W	H'00	H'FFD3
	Timer interrupt enable register 0	TIER0	R/W	H'40	H'FFD4
	Timer status register 0	TSR0	R/(W)*2	H'C0	H'FFD5
	Timer counter 0	TCNT0	R/W	H'0000	H'FFD6
	Timer general register 0A	TGR0A	R/W	H'FFFF	H'FFD8
	Timer general register 0B	TGR0B	R/W	H'FFFF	H'FFDA
	Timer general register 0C	TGR0C	R/W	H'FFFF	H'FFDC
	Timer general register 0D	TGR0D	R/W	H'FFFF	H'FFDE
1	Timer control register 1	TCR1	R/W	H'00	H'FFE0
	Timer mode register 1	TMDR1	R/W	H'C0	H'FFE1
	Timer I/O control register 1	TIOR1	R/W	H'00	H'FFE2
	Timer interrupt enable register 1	TIER1	R/W	H'40	H'FFE4
	Timer status register 1	TSR1	R/(W)*2	H'C0	H'FFE5
	Timer counter 1	TCNT1	R/W	H'0000	H'FFE6
	Timer general register 1A	TGR1A	R/W	H'FFFF	H'FFE8
	Timer general register 1B	TGR1B	R/W	H'FFFF	H'FFEA
2	Timer control register 2	TCR2	R/W	H'00	H'FFF0
	Timer mode register 2	TMDR2	R/W	H'C0	H'FFF1
	Timer I/O control register 2	TIOR2	R/W	H'00	H'FFF2
	Timer interrupt enable register 2	TIER2	R/W	H'40	H'FFF4
	Timer status register 2	TSR2	R/(W)*2	H'C0	H'FFF5
	Timer counter 2	TCNT2	R/W	H'0000	H'FFF6
	Timer general register 2A	TGR2A	R/W	H'FFFF	H'FFF8
	Timer general register 2B	TGR2B	R/W	H'FFFF	H'FFFA

	Timer mode register 3	TMDR3	R/W	H'C0	H'FE81
	Timer I/O control register 3H	TIOR3H	R/W	H'00	H'FE82
	Timer I/O control register 3L	TIOR3L	R/W	H'00	H'FE83
	Timer interrupt enable register 3	TIER3	R/W	H'40	H'FE84
	Timer status register 3	TSR3	R/(W)*2	H'C0	H'FE85
	Timer counter 3	TCNT3	R/W	H'0000	H'FE86
	Timer general register 3A	TGR3A	R/W	H'FFFF	H'FE88
	Timer general register 3B	TGR3B	R/W	H'FFFF	H'FE8A
	Timer general register 3C	TGR3C	R/W	H'FFFF	H'FE8C
	Timer general register 3D	TGR3D	R/W	H'FFFF	H'FE8E
4	Timer control register 4	TCR4	R/W	H'00	H'FE90
	Timer mode register 4	TMDR4	R/W	H'C0	H'FE91
	Timer I/O control register 4	TIOR4	R/W	H'00	H'FE92
	Timer interrupt enable register 4	TIER4	R/W	H'40	H'FE94
	Timer status register 4	TSR4	R/(W)*2	H'C0	H'FE95
	Timer counter 4	TCNT4	R/W	H'0000	H'FE96
	Timer general register 4A	TGR4A	R/W	H'FFFF	H'FE98
	Timer general register 4B	TGR4B	R/W	H'FFFF	H'FE9A
5	Timer control register 5	TCR5	R/W	H'00	H'FEA0
	Timer mode register 5	TMDR5	R/W	H'C0	H'FEA1
	Timer I/O control register 5	TIOR5	R/W	H'00	H'FEA2
	Timer interrupt enable register 5	TIER5	R/W	H'40	H'FEA4
	Timer status register 5	TSR5	R/(W)*2	H'C0	H'FEA5
	Timer counter 5	TCNT5	R/W	H'0000	H'FEA6
	Timer general register 5A	TGR5A	R/W	H'FFFF	H'FEA8
	Timer general register 5B	TGR5B	R/W	H'FFFF	H'FEAA
All	Timer start register	TSTR	R/W	H'00	H'FFC0
	Timer synchro register	TSYR	R/W	H'00	H'FFC1
	Module stop control register	MSTPCR	R/W	H'3FFF	H'FF3C

Notes: 1. Lower 16 bits of the address.
2. Can only be written with 0 for flag clearing.

10.2.1 Timer Control Registers (TCR)

Channel 0: TCR0
Channel 3: TCR3

Bit	:	7	6	5	4	3	2	1	0
		CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
Initial value	:	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

Channel 1: TCR1
Channel 2: TCR2
Channel 4: TCR4
Channel 5: TCR5

Bit	:	7	6	5	4	3	2	1	0
		—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	—	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 TCR registers, one for each of channels 0 to 5. The TCR registers are initialized to H'00 by a reset and in hardware standby mode.

TCR register settings should be made only when TCNT operation is stopped.

Channel	Bit 7 CCLR2	Bit 6 CCLR1	Bit 5 CCLR0	Description
0, 3	0	0	0	TCNT clearing disabled (Initial value)
			1	TCNT cleared by TGRA compare match/input capture
		1	0	TCNT cleared by TGRB compare match/input capture
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation*1
	1	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRC compare match/input capture*2
		1	0	TCNT cleared by TGRD compare match/input capture*2
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation*1

Channel	Bit 7 Reserved*3	Bit 6 CCLR1	Bit 5 CCLR0	Description
1, 2, 4, 5	0	0	0	TCNT clearing disabled (Initial value)
			1	TCNT cleared by TGRA compare match/input capture
		1	0	TCNT cleared by TGRB compare match/input capture
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation*1

Notes: 1. Synchronous operation setting is performed by setting the SYNC bit in TSYR to 1.
2. When TGRC or TGRD is used as a buffer register, TCNT is not cleared because the buffer register setting has priority, and compare match/input capture does not occur.
3. Bit 7 is reserved in channels 1, 2, 4, and 5. It is always read as 0 and cannot be modified.

edges = ϕ/2 rising edge). If phase counting mode is used on channels 1, 2, 4, and 5, this setting is ignored and the phase counting mode setting has priority.

Bit 4 CKEG1	Bit 3 CKEG0	Description	
0	0	Count at rising edge	(Initial value)
	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 setting is ignored if the input clock is ϕ/1, or when overflow/underflow of another channel is selected.

Bits 2 to 0—Time Prescaler 2 to 0 (TPSC2 to TPSC0): These bits select the TCNT counter clock. The clock source can be selected independently for each channel. Table 10.4 shows the clock sources that can be set for each channel.

Table 10.4 TPU Clock Sources

	Internal Clock							External Clock				Overflow/ Underflow on Another Channel
Channel	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

0	0	0	0	Internal clock: counts on ϕ/1	(Initial value)
			1	Internal clock: counts on ϕ/4	
		1	0	Internal clock: counts on ϕ/16	
			1	Internal clock: counts on ϕ/64	
	1	0	0	External clock: counts on TCLKA pin input	
			1	External clock: counts on TCLKB pin input	
		1	0	External clock: counts on TCLKC pin input	
			1	External clock: counts on TCLKD pin input	

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description	
1	0	0	0	Internal clock: counts on ϕ/1	(Initial value)
			1	Internal clock: counts on ϕ/4	
		1	0	Internal clock: counts on ϕ/16	
			1	Internal clock: counts on ϕ/64	
	1	0	0	External clock: counts on TCLKA pin input	
			1	External clock: counts on TCLKB pin input	
		1	0	Internal clock: counts on ϕ/256	
			1	Counts on TCNT2 overflow/underflow	

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	(Initial value)
			1	Internal clock: counts on ϕ/4	
		1	0	Internal clock: counts on ϕ/16	
			1	Internal clock: counts on ϕ/64	
	1	0	0	External clock: counts on TCLKA pin input	
			1	External clock: counts on TCLKB pin input	
		1	0	External clock: counts on TCLKC pin input	
			1	Internal clock: counts on ϕ/1024	

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

3	0	0	0	Internal clock: counts on ϕ/1	(Initial value)
			1	Internal clock: counts on ϕ/4	
		1	0	Internal clock: counts on ϕ/16	
			1	Internal clock: counts on ϕ/64	
	1	0	0	External clock: counts on TCLKA pin input	
			1	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	(Initial value)
			1	Internal clock: counts on ϕ/4	
		1	0	Internal clock: counts on ϕ/16	
			1	Internal clock: counts on ϕ/64	
	1	0	0	External clock: counts on TCLKA pin input	
			1	External clock: counts on TCLKC pin input	
		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	(Initial value)
			1	Internal clock: counts on ϕ/4	
		1	0	Internal clock: counts on ϕ/16	
			1	Internal clock: counts on ϕ/64	
	1	0	0	External clock: counts on TCLKA pin input	
			1	External clock: counts on TCLKC pin input	
		1	0	Internal clock: counts on ϕ/256	
			1	External clock: counts on TCLKD pin input	

Note: This setting is ignored when channel 5 is in phase counting mode.

Channel 0: TMDR0
Channel 3: TMDR3

Bit	:	7	6	5	4	3	2	1	0
		—	—	BFB	BFA	MD3	MD2	MD1	MD0
Initial value	:	1	1	0	0	0	0	0	0
R/W	:	—	—	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	0
		—	—	—	—	MD3	MD2	MD1	MD0
Initial value	:	1	1	0	0	0	0	0	0
R/W	:	—	—	—	—	R/W	R/W	R/W	R/W

The TMDR registers are 8-bit readable/writable registers that are used to set the operating mode for each channel. The TPU has six TMDR registers, one for each channel. The TMDR registers 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 cannot be modified and are always read as 1.

Bit 5—Buffer Operation B (BFB): Specifies whether TGRB is to operate in the normal way, or TGRB and TGRD are to be used together for buffer operation. When TGRD is used as a buffer register, TGRD input capture/output compare is not generated.

In channels 1, 2, 4, and 5, which have no TGRD, bit 5 is reserved. It is always read as 0 and cannot be modified.

Bit 5 BFB	Description	
0	TGRB operates normally	(Initial value)
1	TGRB and TGRD used together for buffer operation	

register, TGRC input capture/output compare is not generated.

In channels 1, 2, 4, and 5, which have no TGRC, bit 4 is reserved. It is always read as 0 and cannot be modified.

Bit 4 BFA	Description	
0	TGRA operates normally	(Initial value)
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 operating mode.

Bit 3 MD3*1	Bit 2 MD2*2	Bit 1 MD1	Bit 0 MD0	Description	
0	0	0	0	Normal operation	(Initial value)
			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	*	*	*	—	

*: Don't care

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. For these channels, 0 should always be written to MD2.

Channel 0: TIOR0H
Channel 1: TIOR1
Channel 2: TIOR2
Channel 3: TIOR3H
Channel 4: TIOR4
Channel 5: TIOR5

Bit	:	7	6	5	4	3	2	1	0
		IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
Initial value	:	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

Channel 0: TIOR0L
Channel 3: TIOR3L

Bit	:	7	6	5	4	3	2	1	0
		IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0
Initial value	:	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

Note: When TGRC or TGRD is designated for buffer operation, this setting is invalid and the register operates as a buffer register.

The TIOR registers are 8-bit registers that control the TGR registers. The TPU has eight TIOR registers, two each for channels 0 and 3, and one each for channels 1, 2, 4, and 5. The TIOR 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 specified by TIOR is valid when the counter is stopped (the CST bit in TSTR is cleared to 0). Note also that, in PWM mode 2, the output at the point at which the counter is cleared to 0 is specified.

Bits IOB3 to IOB0 specify the function of TGRB.
Bits IOD3 to IOD0 specify the function of TGRD.

Channel	Bit 7 IOB3	Bit 6 IOB2	Bit 5 IOB1	Bit 4 IOB0	Description		
0	0	0	0	0	TGR0B is output compare register	Output disabled	(Initial value)
				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 ϕ/1 is used as the TCNT1 count clock, this setting is invalid and input capture is not generated.

0	0	0	0	0	TGR0D is output compare register*2	Output disabled	(Initial value)
				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*2	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

Notes: 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.

2. When the BFB bit in TMDR0 is set to 1 and TGR0D is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

1	0	0	0	0	TGR1B is output compare register	Output disabled	(Initial value)
				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

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	(Initial value)
				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

3	0	0	0	0	TGR3B is output compare register	Output disabled	(Initial value)
				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 ϕ/1 is used as the TCNT4 count clock, this setting is invalid and input capture is not generated.

3	0	0	0	0	TGR3D is output compare register*2	Output disabled	(Initial value)
				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.

4	0	0	0	0	TGR4B is output compare register	Output disabled	(Initial value)
				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

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	(Initial value)
				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

IOA3 to IOA0 specify the function of TGRA.
IOC3 to IOC0 specify the function of TGRC.

Channel	Bit 3 IOA3	Bit 2 IOA2	Bit 1 IOA1	Bit 0 IOA0	Description		
0	0	0	0	0	TGR0A is output compare register	Output disabled	(Initial value)
				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	TGR0A is input capture register	Capture input source is TIOCA0 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

*: Don’t care

0	0	0	0	0	TGR0C is output compare register*1	Output disabled	(Initial value)
				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	TGR0C is input capture register*1	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 count-up/count-down

*: Don't care

Note: 1. When the BFA bit in TMDR0 is set to 1 and TGR0C is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

1	0	0	0	0	TGR1A is output compare register	Output disabled	(Initial value)
				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	TGR1A is input capture register	Capture input source is TIOCA1 pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges
		1	*	*		Capture input source is TGR0A compare match/ input capture	Input capture at generation of channel 0/TGR0A compare match/input capture

*: Don't care

2	0	0	0	0	TGR2A is output compare register	Output disabled	(Initial value)
				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	TGR2A is input capture register	Capture input source is TIOCA2 pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges

*: Don't care

Channel	Bit 3 IOA3	Bit 2 IOA2	Bit 1 IOA1	Bit 0 IOA0	Description		
3	0	0	0	0	TGR3A is output compare register	Output disabled	(Initial value)
				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	TGR3A is input capture register	Capture input source is TIOCA3 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

*: Don't care

3	0	0	0	0	TGR3C is output compare register*1	Output disabled	(Initial value)
				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	TGR3C is input capture register*1	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 count-up/count-down

*: Don't care

Note: 1. When the BFA bit in TMDR3 is set to 1 and TGR3C is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

4	0	0	0	0	TGR4A is output compare register	Output disabled	(Initial value)
				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	TGR4A is input capture register	Capture input source is TIOCA4 pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges
		1	*	*		Capture input source is TGR3A compare match/ input capture	Input capture at generation of TGR3A compare match/input capture

*: Don't care

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	(Initial value)
				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	TGR5A is input capture register	Capture input source is TIOCA5 pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges

*: Don't care

Channel 0: TIER0

Channel 3: TIER3

Bit	:	7	6	5	4	3	2	1	0
		TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
Initial value	:	0	1	0	0	0	0	0	0
R/W	:	R/W	—	—	R/W	R/W	R/W	R/W	R/W

Channel 1: TIER1

Channel 2: TIER2

Channel 4: TIER4

Channel 5: TIER5

Bit	:	7	6	5	4	3	2	1	0
		TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
Initial value	:	0	1	0	0	0	0	0	0
R/W	:	R/W	—	R/W	R/W	—	—	R/W	R/W

The TIER registers are 8-bit registers that control enabling or disabling of interrupt requests for each channel. The TPU has six TIER registers, one for each channel. The TIER registers are initialized to H'40 by a reset and in hardware standby mode.

Bit 7—A/D Conversion Start Request Enable (TTGE): Enables or disables generation of A/D conversion start requests by TGRA input capture/compare match.

Bit 7 TTGE	Description	
0	A/D conversion start request generation disabled	(Initial value)
1	A/D conversion start request generation enabled	

Bit 6—Reserved: This bit cannot be modified and is always read as 1.

In channels 0 and 3, bit 5 is reserved. It is always read as 0 and cannot be modified.

Bit 5 TCIEU	Description	
0	Interrupt requests (TCIU) by TCFU disabled	(Initial value)
1	Interrupt requests (TCIU) by TCFU enabled	

Bit 4—Overflow Interrupt Enable (TCIEV): Enables or disables interrupt requests (TCIV) by the TCFV bit when the TCFV bit in TSR is set to 1.

Bit 4 TCIEV	Description	
0	Interrupt requests (TCIV) by TCFV disabled	(Initial value)
1	Interrupt requests (TCIV) by TCFV enabled	

Bit 3—TGR Interrupt Enable D (TGIED): Enables or disables interrupt requests (TGID) by the 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 modified.

Bit 3 TGIED	Description	
0	Interrupt requests (TGID) by TGFD disabled	(Initial value)
1	Interrupt requests (TGID) by TGFD enabled	

Bit 2—TGR Interrupt Enable C (TGIEC): Enables or disables interrupt requests (TGIC) by the 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 modified.

Bit 2 TGIEC	Description	
0	Interrupt requests (TGIC) by TGFC disabled	(Initial value)
1	Interrupt requests (TGIC) by TGFC enabled	

Bit 1 TGIEB	Description	
0	Interrupt requests (TGIB) by TGFB disabled	(Initial value)
1	Interrupt requests (TGIB) by TGFB enabled	

Bit 0—TGR Interrupt Enable A (TGIEA): Enables or disables interrupt requests (TGIA) by the TGFA bit when the TGFA bit in TSR is set to 1.

Bit 0 TGIEA	Description	
0	Interrupt requests (TGIA) by TGFA disabled	(Initial value)
1	Interrupt requests (TGIA) by TGFA enabled	

Channel 0: TSR0
Channel 3: TSR3

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
Initial value	:	1	1	0	0	0	0	0	0
R/W	:	—	—	—	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

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

Channel 1: TSR1
Channel 2: TSR2
Channel 4: TSR4
Channel 5: TSR5

Bit	:	7	6	5	4	3	2	1	0
		TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA
Initial value	:	1	1	0	0	0	0	0	0
R/W	:	R	—	R/(W)*	R/(W)*	—	—	R/(W)*	R/(W)*

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

The TSR registers are 8-bit registers that indicate the status of each channel. The TPU has six TSR registers, one for each channel. The TSR registers are initialized to H'C0 by a reset and in hardware standby mode.

Bit 7—Count Direction Flag (TCFD): Status flag that shows the direction in which TCNT counts in channels 1, 2, 4, and 5.

In channels 0 and 3, bit 7 is reserved. It is always read as 1 and cannot be modified.

Bit 7 TCFD	Description	
0	TCNT counts down	
1	TCNT counts up	(Initial value)

Bit 6—Reserved: This bit cannot be modified and is always read as 1.

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	(Initial value)
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 occurred.

Bit 4 TCFV	Description	
0	[Clearing condition] When 0 is written to TCFV after reading TCFV = 1	(Initial value)
1	[Setting condition] When the TCNT value overflows (changes from H'FFFF to H'0000)	

Bit 3—Input Capture/Output Compare Flag D (TGFD): Status flag that indicates the occurrence of TGRD input capture or compare match 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 modified.

Bit 3 TGFD	Description	
0	[Clearing conditions] • When DTC is activated by TGID interrupt while DISEL bit of MRB in DTC is 0 • When 0 is written to TGFD after reading TGFD = 1	(Initial value)
1	[Setting conditions] • When TCNT = TGRD while TGRD is functioning as output compare register • When TCNT value is transferred to TGRD by input capture signal while TGRD is functioning as input capture register	

In channels 1, 2, 4, and 5, bit 2 is reserved. It is always read as 0 and cannot be modified.

Bit 2 TGFC	Description
0	[Clearing conditions] (Initial value) • When DTC is activated by TGIC interrupt while DISEL bit of MRB in DTC is 0 • When 0 is written to TGFC after reading TGFC = 1
1	[Setting conditions] • When TCNT = TGRC while TGRC is functioning as output compare register • When TCNT value is transferred to TGRC by input capture signal while TGRC is functioning as input capture register

Bit 1—Input Capture/Output Compare Flag B (TGFB): Status flag that indicates the occurrence of TGRB input capture or compare match.

Bit 1 TGFB	Description
0	[Clearing conditions] (Initial value) • When DTC is activated by TGIB interrupt while DISEL bit of MRB in DTC 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

Bit 0 TGFA	Description
0	[Clearing conditions] (Initial value) • When DTC is activated by TGIA interrupt while DISEL bit of MRB in DTC is 0 • When DMAC is activated by TGIA interrupt while DTA bit of DMABCR in DMAC is 1 • When 0 is written to TGFA after reading TGFA = 1
1	[Setting conditions] • When TCNT = TGRA while TGRA is functioning as output compare register • When TCNT value is transferred to TGRA by input capture signal while TGRA is functioning as input capture register

10.2.6 Timer Counters (TCNT)

Channel 0: TCNT0 (up-counter)
Channel 1: TCNT1 (up/down-counter*)
Channel 2: TCNT2 (up/down-counter*)
Channel 3: TCNT3 (up-counter)
Channel 4: TCNT4 (up/down-counter*)
Channel 5: TCNT5 (up/down-counter*)

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	:	0	0	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/W	R/W

Note: * These counters can be used as up/down-counters only in phase counting mode or when counting overflow/underflow on another channel. In other cases they function as up-counters.

The TCNT registers are 16-bit counters. The TPU has six TCNT counters, one for each channel.

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

The TCNT counters cannot be accessed in 8-bit units; they must always be accessed as a 16-bit unit.

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	:	1	1	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/W	R/W	R/W

The TGR registers are 16-bit registers with a dual function as output compare and input capture registers. The TPU has 16 TGR registers, four each for channels 0 and 3 and two each for channels 1, 2, 4, and 5. TGRC and TGRD for channels 0 and 3 can also be designated for operation as buffer registers*. The TGR registers are initialized to H'FFFF by a reset and in hardware standby mode.

The TGR registers cannot be accessed in 8-bit units; they must always be accessed as a 16-bit unit.

Note: * TGR buffer register combinations are TGRA-TGRC and TGRB-TGRD.

Bit	:	7	6	5	4	3	2	1	0
		—	—	CST5	CST4	CST3	CST2	CST1	CST0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	—	—	R/W	R/W	R/W	R/W	R/W	R/W

TSTR is an 8-bit readable/writable register that selects operation/stoppage for channels 0 to 5. TSTR is initialized to H'00 by a reset, and in hardware standby mode. When setting the operating mode in TMDR or setting the count clock in TCR, first stop the TCNT counter.

Bits 7 and 6—Reserved: Must always be written with 0.

Bits 5 to 0—Counter Start 5 to 0 (CST5 to CST0): These bits select operation or stoppage for TCNT.

Bit n CSTn	Description	
0	TCNTn count operation is stopped	(Initial value)
1	TCNTn performs count operation	

n = 5 to 0

Note: If 0 is written to the CST bit during operation with the TIOC pin designated for output, the counter stops but the TIOC pin output compare output level is retained. If TIOR is written to when the CST bit is cleared to 0, the pin output level will be changed to the set initial output value.

Bit	:	7	6	5	4	3	2	1	0
		—	—	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	—	—	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 synchronous operation for the channel 0 to 4 TCNT counters. A channel performs synchronous operation when 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: Must always be written with 0.

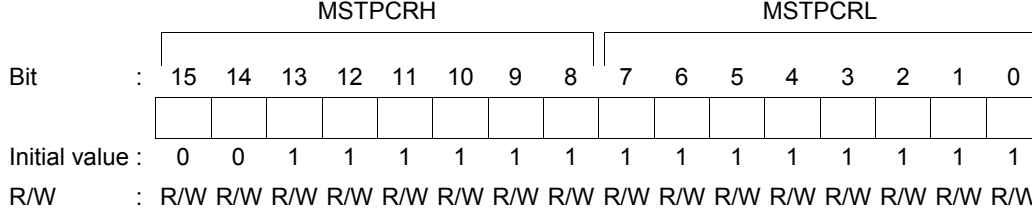
Bits 5 to 0—Timer Synchro 5 to 0 (SYNC5 to SYNC0): These bits select whether operation is independent of or synchronized with other channels.

When synchronous operation is selected, synchronous presetting of multiple channels*1, and synchronous clearing through counter clearing on another channel*2 are possible.

Bit n SYNCn	Description
0	TCNTn operates independently (TCNT presetting/clearing is unrelated to other channels) (Initial value)
1	TCNTn performs synchronous operation TCNT synchronous presetting*1/synchronous clearing*2 is possible

n = 5 to 0

Notes: 1. To set synchronous operation, the SYNC bits for at least two channels must be set to 1.
2. To set synchronous clearing, in addition to the SYNC bit, the TCNT clearing source must also be set by means of bits CCLR2 to CCLR0 in TCR.



MSTPCR is a 16-bit readable/writable register that performs module stop mode control.

When the MSTP13 bit in MSTPCR is set to 1, TPU operation stops at the end of the bus cycle and a transition is made to module stop mode. Registers cannot be read or written to in module stop mode. For details, see section 21.5, Module Stop Mode.

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

Bit 13—Module Stop (MSTP13): Specifies the TPU module stop mode.

Bit 13 MSTP13	Description	
0	TPU module stop mode cleared	
1	TPU module stop mode set	(Initial value)

10.3.1 16-Bit Registers

TCNT and TGR are 16-bit registers. As the data bus to the bus master is 16 bits wide, these registers can be read and written to in 16-bit units.

These registers cannot be read or written to in 8-bit units; 16-bit access must always be used.

An example of 16-bit register access operation is shown in figure 10.2.

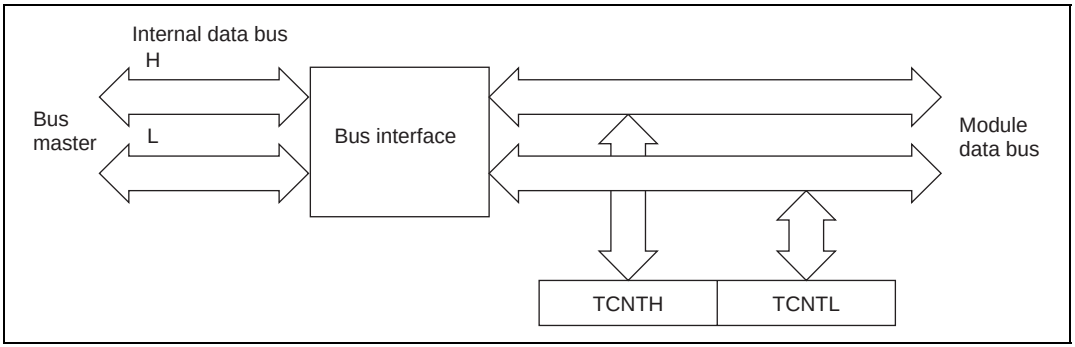


Figure 10.2 16-Bit Register Access Operation [Bus Master ↔ TCNT (16 Bits)]

10.3.2 8-Bit Registers

Registers other than TCNT and TGR are 8-bit. As the data bus to the CPU is 16 bits wide, these registers can be read and written to in 16-bit units. They can also be read and written to in 8-bit units.

Examples of 8-bit register access operation are shown in figures 10.3 to 10.5.

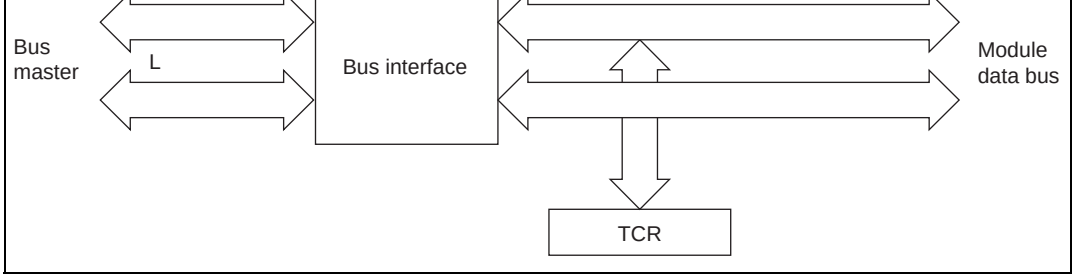


Figure 10.3 8-Bit Register Access Operation [Bus Master ↔ TCR (Upper 8 Bits)]

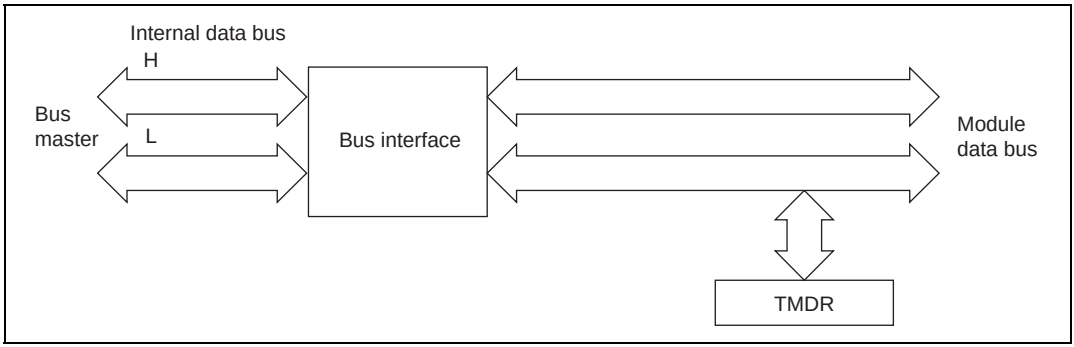


Figure 10.4 8-Bit Register Access Operation [Bus Master ↔ TMDR (Lower 8 Bits)]

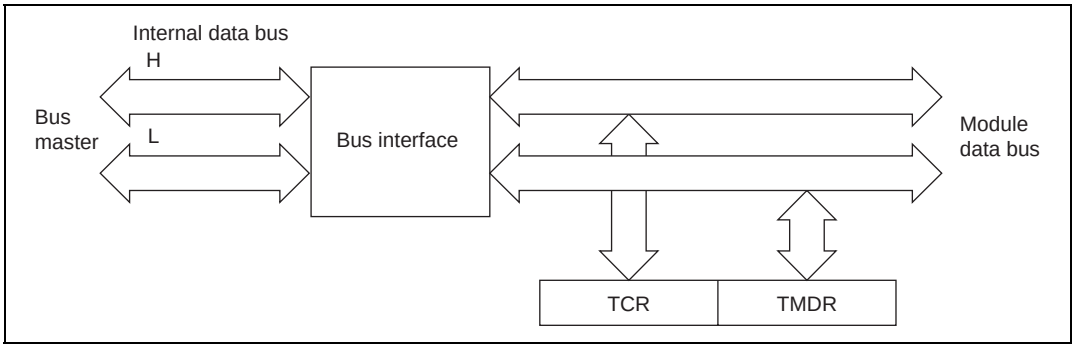


Figure 10.5 8-Bit Register Access Operation [Bus Master ↔ TCR and TMDR (16 Bits)]

10.4.1 Overview

Operation in each mode is outlined below.

Normal Operation: Each channel has a TCNT and TGR register. TCNT performs up-counting, and is also capable of free-running operation, synchronous counting, and external event counting.

Each TGR can be used as an input capture register or output compare register.

Synchronous Operation: When synchronous operation is designated for a channel, TCNT for that channel performs synchronous presetting. That is, when TCNT for a channel designated for synchronous operation is rewritten, the TCNT counters for the other channels are also rewritten at the same time. Synchronous clearing of the TCNT counters is also possible by setting the timer 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 channel is transferred to TGR.
- When TGR is an input capture register
  When input capture occurs, the value in TCNT is transferred to TGR and the value previously held in TGR is transferred to the buffer register.

Cascaded Operation: The channel 1 counter (TCNT1) and channel 2 counter (TCNT2), or the channel 4 counter (TCNT4) and channel 5 counter (TCNT5), can be connected together to operate as a 32-bit counter.

PWM Mode: In this mode, a PWM waveform is output. The output level can be set by means of TIOR. A PWM waveform with a duty of between 0% and 100% can be output, according to the setting of each TGR register.

Phase Counting Mode: In this mode, TCNT is incremented or decremented by detecting the phases of two clocks input from the external clock input pins in channels 1, 2, 4, and 5. When phase counting mode is set, the corresponding TCLK pin functions as the clock pin, and TCNT performs up/down-counting.

This can be used for two-phase encoder pulse input.

Counter Operation: When one of bits CST0 to CST5 is set to 1 in TSTR, the TCNT counter for the corresponding channel starts counting. TCNT can operate as a free-running counter, periodic counter, and so on.

- Example of count operation setting procedure

  Figure 10.6 shows an example of the count operation setting procedure.

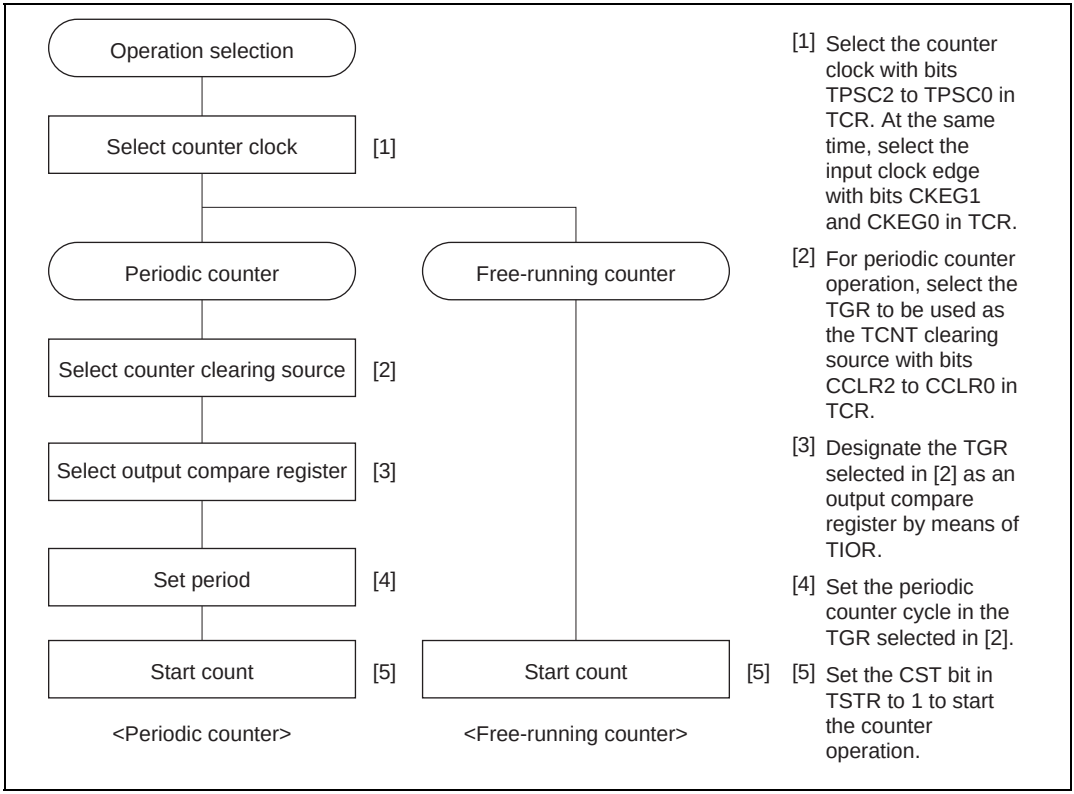


Figure 10.6 Example of Counter Operation Setting Procedure

counters. When the relevant bit in TSTR is set to 1 the corresponding TCNT counter starts up-count operation as a free-running counter. When TCNT overflows (from H'FFFF to H'0000), the TCFV bit in TSR is set to 1. If the value of the corresponding TCIEV bit in TIER is 1 at this point, the TPU requests an interrupt. After overflow, TCNT starts counting up again from H'0000.

Figure 10.7 illustrates free-running counter operation.

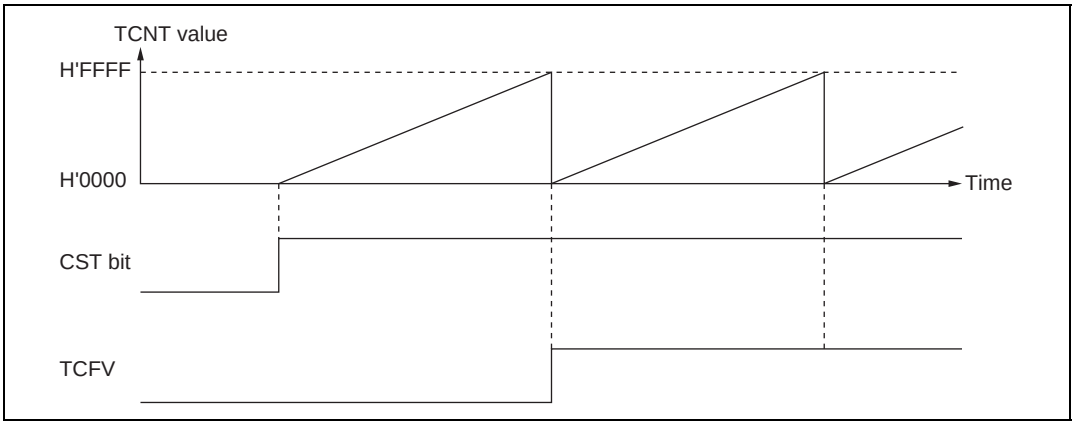


Figure 10.7 Free-Running Counter Operation

When compare match is selected as the TCNT clearing source, the TCNT counter for the relevant channel performs periodic count operation. The TGR register for setting the period is designated as an output compare register, and counter clearing by compare match is selected by means of bits CCLR2 to CCLR0 in TCR. After the settings have been made, TCNT starts up-count operation as a periodic counter when the corresponding bit in TSTR is set to 1. When the count value matches the value in TGR, the TGF bit in TSR is set to 1 and TCNT is cleared to H'0000.

If the value of the corresponding TGIE bit in TIER is 1 at this point, the TPU requests an interrupt. After a compare match, TCNT starts counting up again from H'0000.

Figure 10.8 illustrates periodic counter operation.

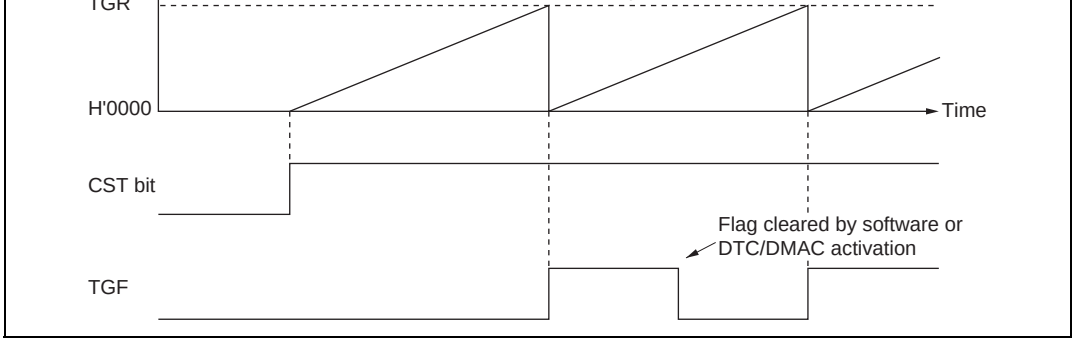


Figure 10.8 Periodic Counter Operation

Waveform Output by Compare Match: The TPU can perform 0, 1, or toggle output from the 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 compare match

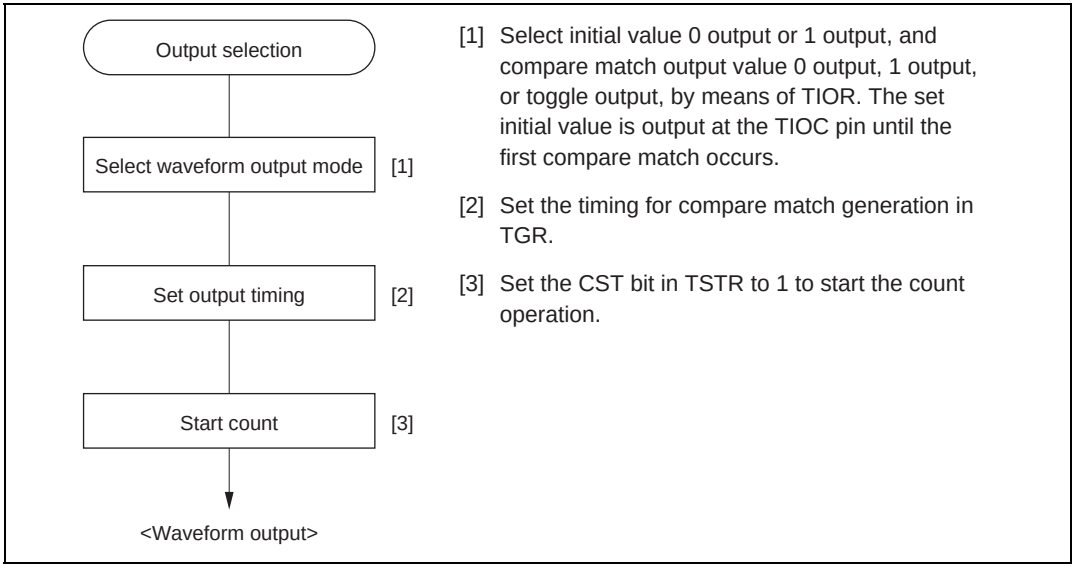


Figure 10.9 Example of Setting Procedure for Waveform Output by Compare Match

In this example TCNT has been designated as a free-running counter, and settings have been made so that 1 is output by compare match A, and 0 is output by compare match B. When the set level and the pin level coincide, the pin level does not change.

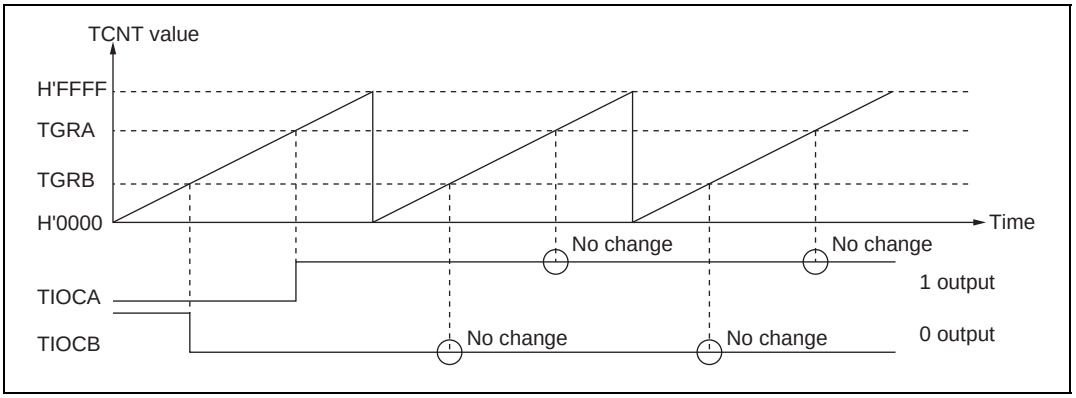


Figure 10.10 Example of 0 Output/1 Output Operation

Figure 10.11 shows an example of toggle output.

In this example TCNT has been designated as a periodic counter (with counter clearing performed by compare match B), and settings have been made so that output is toggled by both compare match A and compare match B.

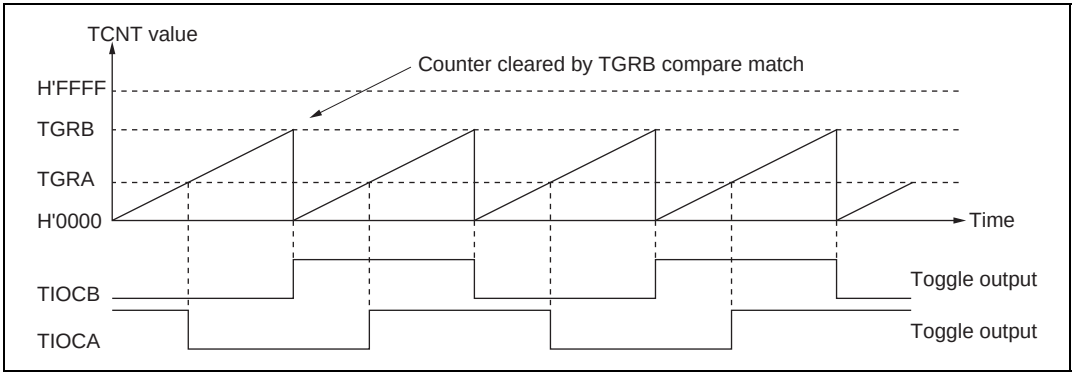


Figure 10.11 Example of Toggle Output Operation

Rising edge, falling edge, or both edges can be selected as the detected edge. For channels 0, 1, 3, and 4, it is also possible to specify another channel's counter input clock or compare match signal as the input capture source.

Note: When another channel's counter input clock is used as the input capture input for channels 0 and 3, $\phi/1$ should not be selected as the counter input clock used for input capture input. Input capture will not be generated if $\phi/1$ is selected.

- Example of input capture operation setting procedure

  Figure 10.12 shows an example of the input capture operation setting procedure.

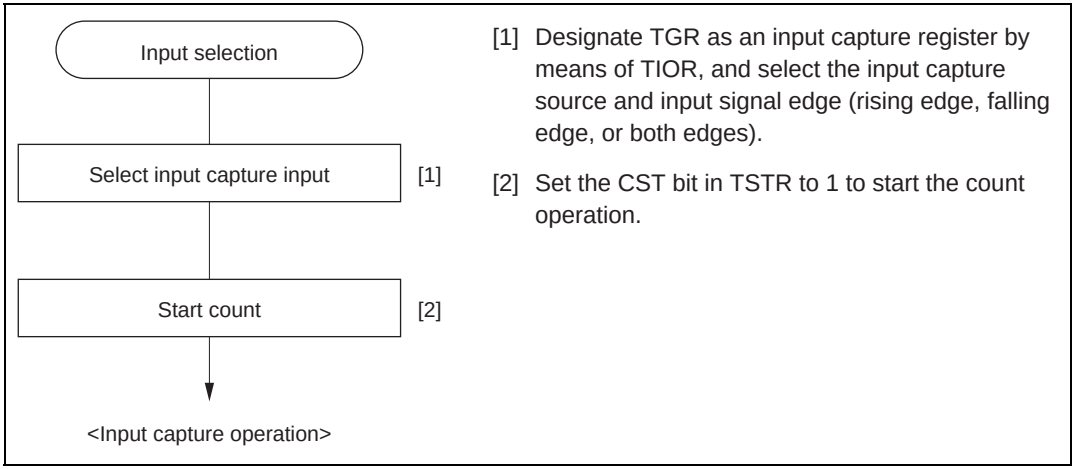


Figure 10.12 Example of Input Capture Operation Setting Procedure

In this example both rising and falling edges have been selected as the TIOCA pin input capture input edge, falling edge has been selected as the TIOCB pin input capture input edge, and counter clearing by TGRB input capture has been designated for TCNT.

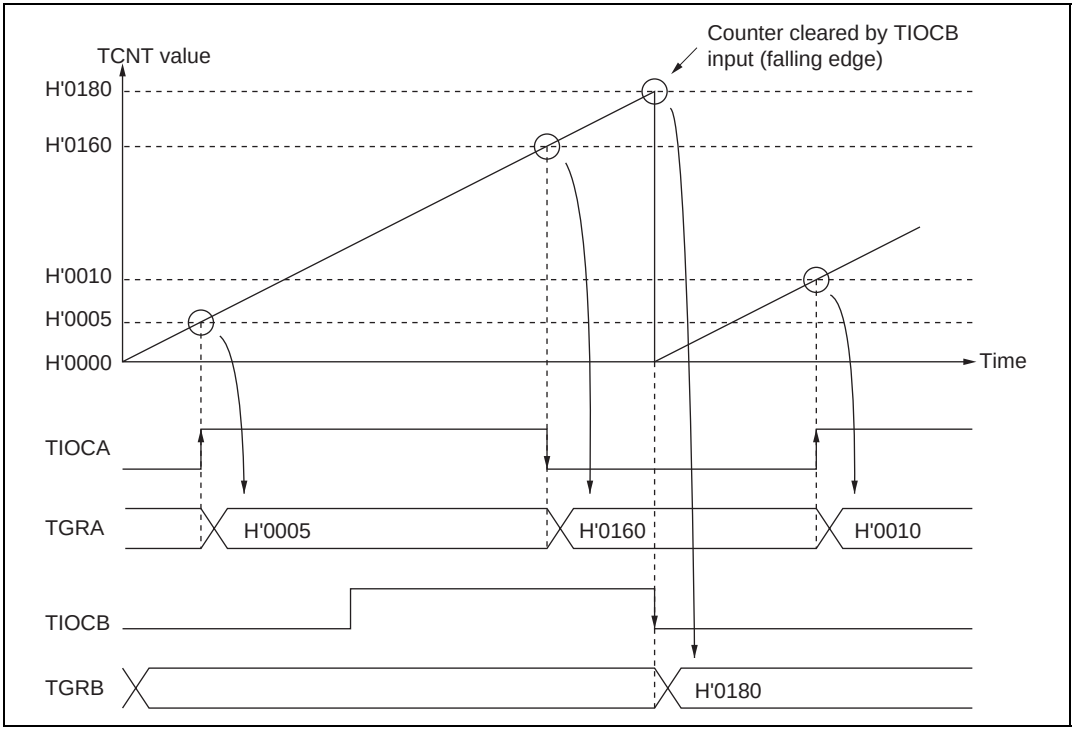


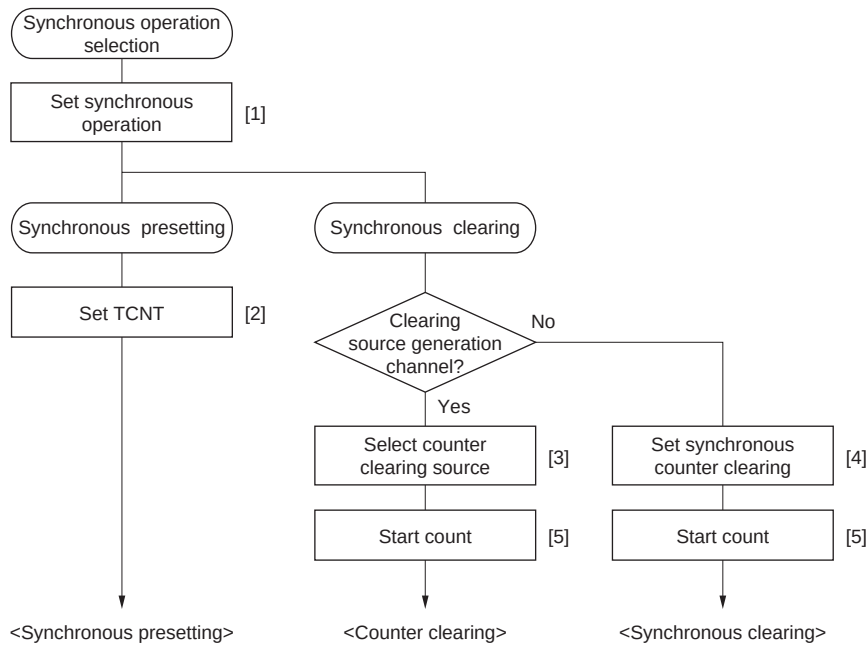
Figure 10.13 Example of Input Capture Operation

In synchronous operation, the values in a number of TCNT counters can be rewritten simultaneously (synchronous presetting). Also, a number of TCNT counters can be cleared simultaneously by making the appropriate setting in TCR (synchronous clearing).

Synchronous operation enables TGR to be incremented with respect to a single time base.

Channels 0 to 5 can all be designated for synchronous operation.

Example of Synchronous Operation Setting Procedure: Figure 10.14 shows an example of the synchronous operation setting procedure.



[1] Set to 1 the SYNC bits in TSYR corresponding to the channels to be designated for synchronous operation.

[2] When the TCNT counter of any of the channels designated for synchronous operation is written to, the same value is simultaneously written to the other TCNT counters.

[3] Use bits CCLR2 to CCLR0 in TCR to specify TCNT clearing by input capture/output compare, etc.

[4] Use bits CCLR2 to CCLR0 in TCR to designate synchronous clearing for the counter clearing source.

[5] Set to 1 the CST bits in TSTR for the relevant channels, to start the count operation.

Figure 10.14 Example of Synchronous Operation Setting Procedure

In this example, synchronous operation and PWM mode 1 have been designated for channels 0 to 2, TGR0B compare match has been set as the channel 0 counter clearing source, and synchronous clearing has been set for the channel 1 and 2 counter clearing source.

Three-phase PWM waveforms are output from pins TIOC0A, TIOC1A, and TIOC2A. At this time, synchronous presetting, and synchronous clearing by TGR0B compare match, is performed for channel 0 to 2 TCNT counters, and the data set in TGR0B is used as the PWM cycle.

For details of PWM modes, see section 10.4.6, PWM Modes.

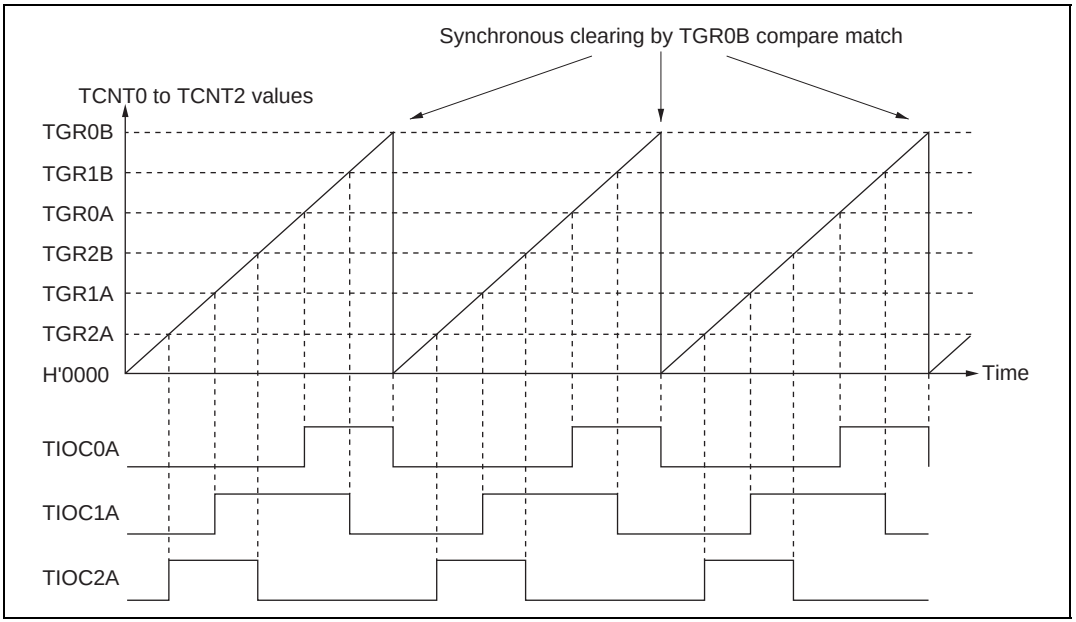


Figure 10.15 Example of Synchronous Operation

Buffer operation, provided for channels 0 and 3, enables TGRC and TGRD to be used as buffer registers.

Buffer operation differs depending on whether TGR has been designated as an input capture register or as a compare match register.

Table 10.5 shows the register combinations used in buffer 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 channel is transferred to the timer general register.

  This operation is illustrated in figure 10.16.

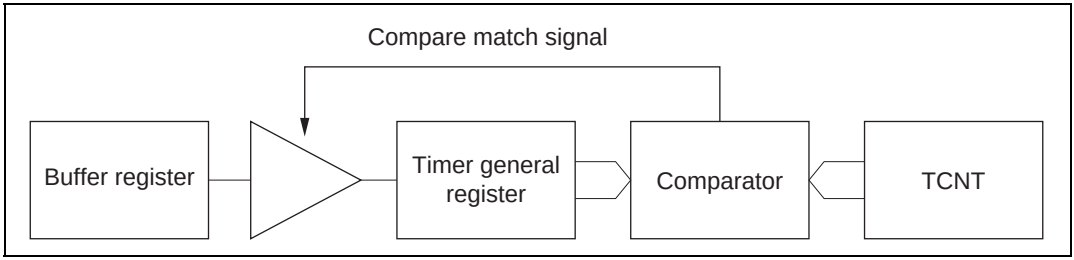


Figure 10.16 Compare Match Buffer Operation

held in the timer general register is transferred to the buffer register.

This operation is illustrated in figure 10.17.

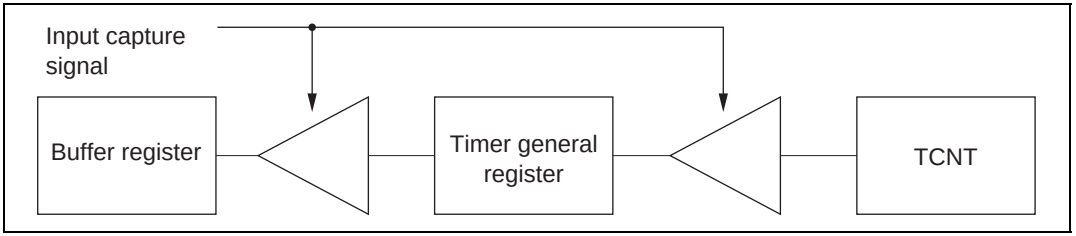


Figure 10.17 Input Capture Buffer Operation

Example of Buffer Operation Setting Procedure: Figure 10.18 shows an example of the buffer operation setting procedure.

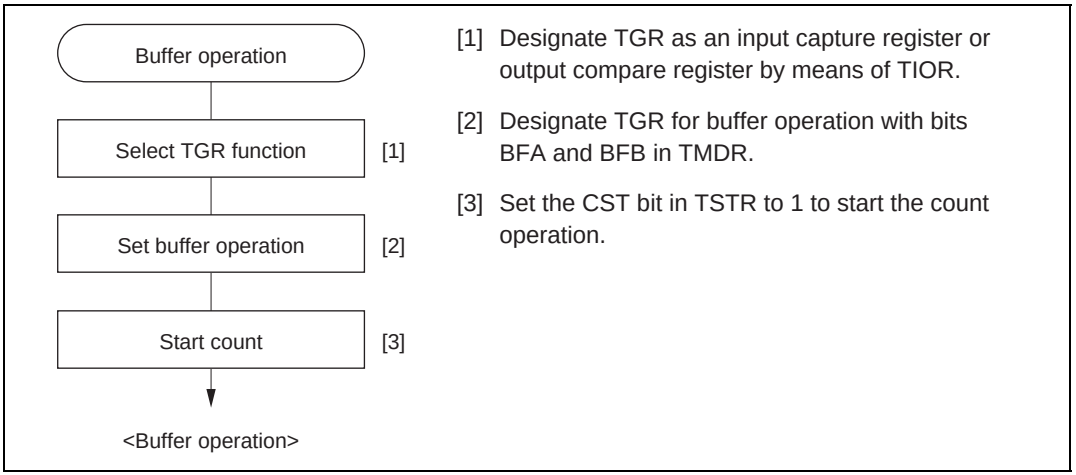


Figure 10.18 Example of Buffer Operation Setting Procedure

- When TGR is an output compare register

  Figure 10.19 shows an operation example in which PWM mode 1 has been designated for channel 0, and buffer operation has been designated for TGRA and TGRC. The settings used in this example are TCNT clearing by compare match B, 1 output at compare match A, and 0 output at compare match B.

  As buffer operation has been set, when compare match A occurs the output changes and the value in buffer register TGRC is simultaneously transferred to timer general register TGRA. This operation is repeated each time compare match A occurs.

  For details of PWM modes, see section 10.4.6, PWM Modes.

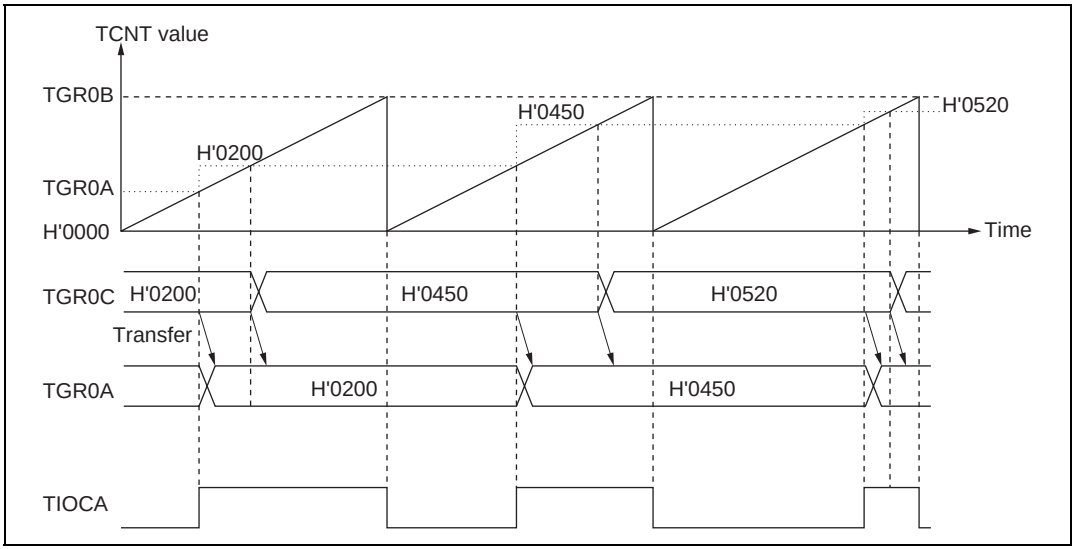


Figure 10.19 Example of Buffer Operation (1)

capture register, and buffer operation has been designated for TGRA and TGRC.

Counter clearing by TGRA input capture has been set for TCNT, and both rising and falling edges have been selected as the TIOCA pin input capture input edge.

As buffer operation has been set, when the TCNT value is stored in TGRA upon occurrence of input capture A, the value previously stored in TGRA is simultaneously transferred to TGRC.

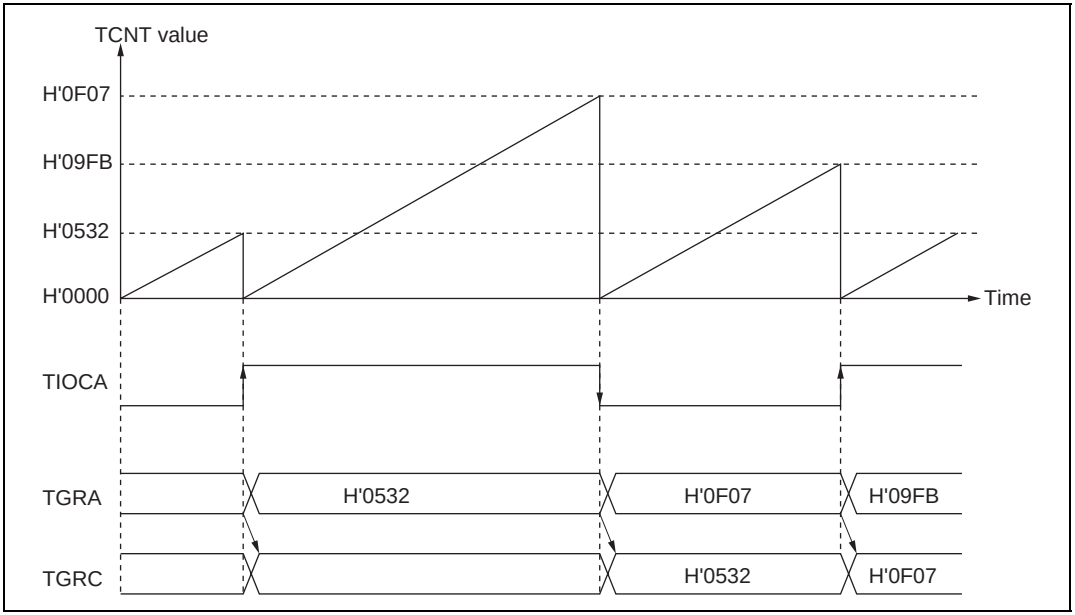


Figure 10.20 Example of Buffer Operation (2)

In cascaded operation, two 16-bit counters for different channels are used together as a 32-bit counter.

This function works by counting the channel 1 (channel 4) counter clock upon overflow/underflow of TCNT2 (TCNT5) as set in bits TPSC2 to TPSC0 in TCR.

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 is invalid 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 of the setting procedure for cascaded operation.

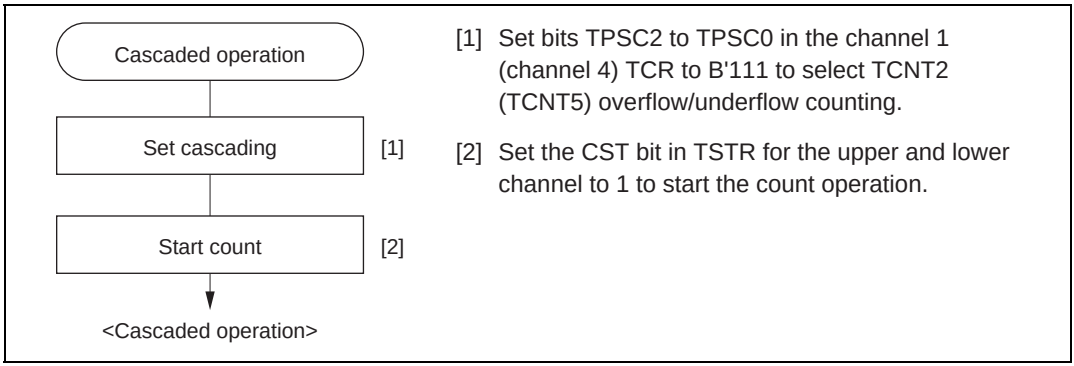


Figure 10.21 Cascaded Operation Setting Procedure

as input capture registers, and TIOC pin rising edge has been selected.

When a rising edge is input to the TIOCA1 and TIOCA2 pins simultaneously, the upper 16 bits of the 32-bit data are transferred to TGR1A, and the lower 16 bits to TGR2A.

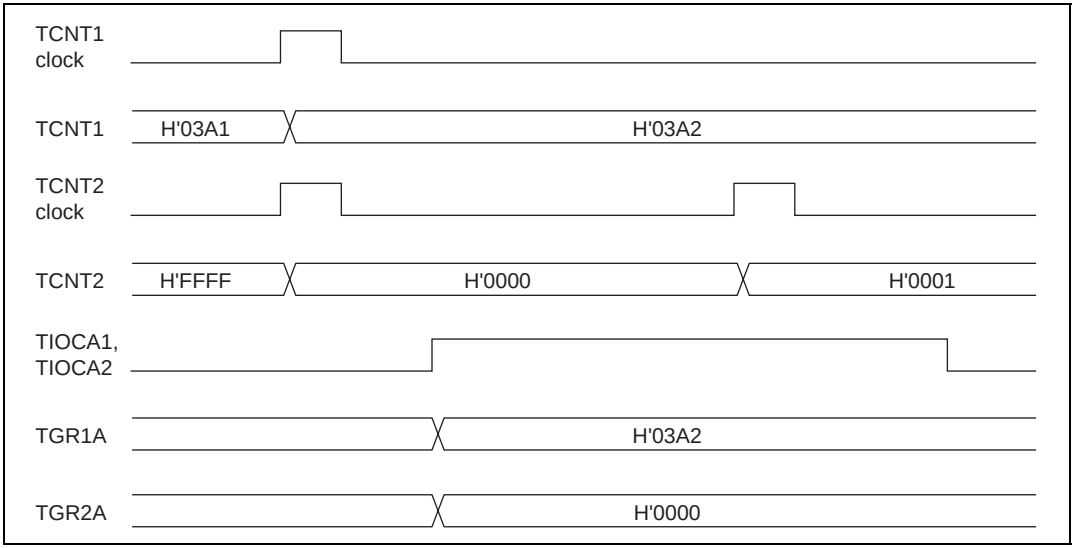


Figure 10.22 Example of Cascaded Operation (1)

Figure 10.23 illustrates the operation when counting upon TCNT2 overflow/underflow has been set for TCNT1, and phase counting mode has been designated for channel 2.

TCNT1 is incremented by TCNT2 overflow and decremented by TCNT2 underflow.

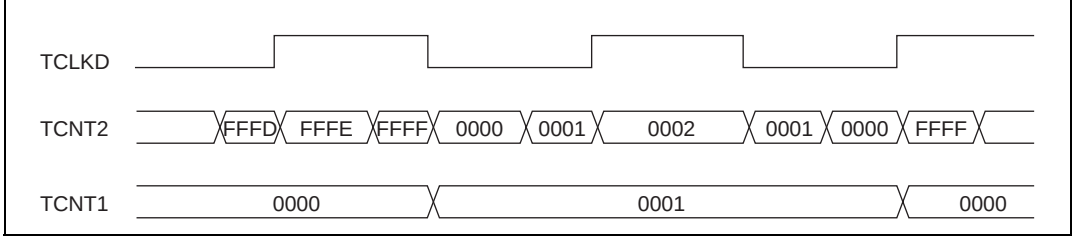


Figure 10.23 Example of Cascaded Operation (2)

10.4.6 PWM Modes

In PWM mode, PWM waveforms are output from the output pins. 0, 1, or toggle output can be selected as the output level in response to compare match of each TGR.

Designating TGR compare match as the counter clearing source enables the period to be set in that register. All channels can be designated for PWM mode independently. Synchronous operation is also possible.

There are two PWM modes, as described below.

- PWM mode 1

  PWM output is generated from the TIOCA and TIOCC pins by pairing TGRA with TGRB and TGRC with TGRD. The output specified by bits IOA3 to IOA0 and IOC3 to IOC0 in TIOR is output from the TIOCA and TIOCC pins at compare matches A and C, and the output specified by bits IOB3 to IOB0 and IOD3 to IOD0 in TIOR is output at compare matches B and D. The initial output value is the value set in TGRA or TGRC. If the set values of paired TGRs are identical, the output value does not change when a compare match occurs.

  In PWM mode 1, a maximum 8-phase PWM output is possible.

- PWM mode 2

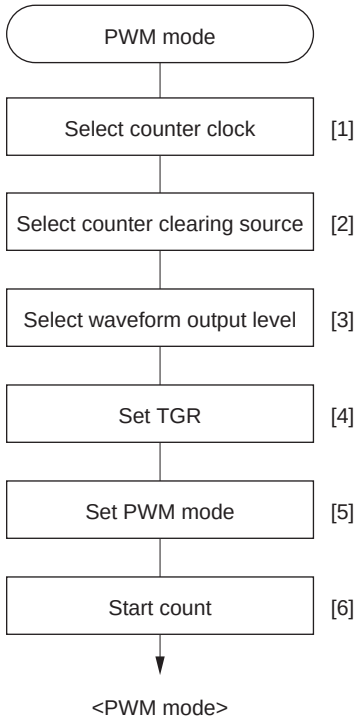
  PWM output is generated using one TGR as the period register and the others as duty registers. The output specified in TIOR is performed by means of compare matches. Upon counter clearing by a synchronization register compare match, the output value of each pin is the initial value set in TIOR. If the set values of the period and duty registers are identical, the output value does not change when a compare match occurs.

  In PWM mode 2, a maximum 15-phase PWM output is possible by combined use with synchronous operation.

Table 10.7 PWM Output Registers and Output Pins

		Output Pins	
Channel	**Registers**	**PWM Mode 1**	**PWM Mode 2**
0	TGR0A	TIOCA0	TIOCA0
	TGR0B		TIOCB0
	TGR0C	TIOCC0	TIOCC0
	TGR0D		TIOCD0
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 period is set.



[1] Select the counter clock with bits TPSC2 to TPSC0 in TCR. At the same time, select the input clock edge with bits CKEG1 and CKEG0 in TCR.

[2] Use bits CCLR2 to CCLR0 in TCR to select the TGR to be used as the TCNT clearing source.

[3] Use TIOR to designate the TGR as an output compare register, and select the initial value and output value.

[4] Set the period in the TGR selected in [2], and set the duty in the other TGR.

[5] Select the PWM mode with bits MD3 to MD0 in TMDR.

[6] Set the CST bit in TSTR to 1 to start the count operation.

Figure 10.24 Example of PWM Mode Setting Procedure

In this example, TGRA compare match is set as the TCNT clearing source, 0 is set for the TGRA initial output value and output value, and 1 is set as the TGRB output value.

In this case, the value set in TGRA is used as the period, and the value set in TGRB as the duty.

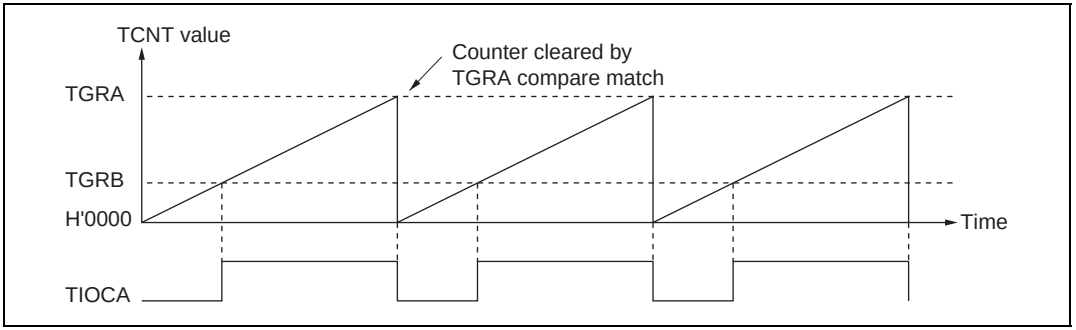


Figure 10.25 Example of PWM Mode Operation (1)

In this example, synchronous operation is designated for channels 0 and 1, TGR1B compare match is set as the TCNT clearing source, and 0 is set for the initial output value and 1 for the output value of the other TGR registers (TGR0A to TGR0D, TGR1A), to output a 5-phase PWM waveform.

In this case, the value set in TGR1B is used as the period, and the values set in the other TGR registers as the duty.

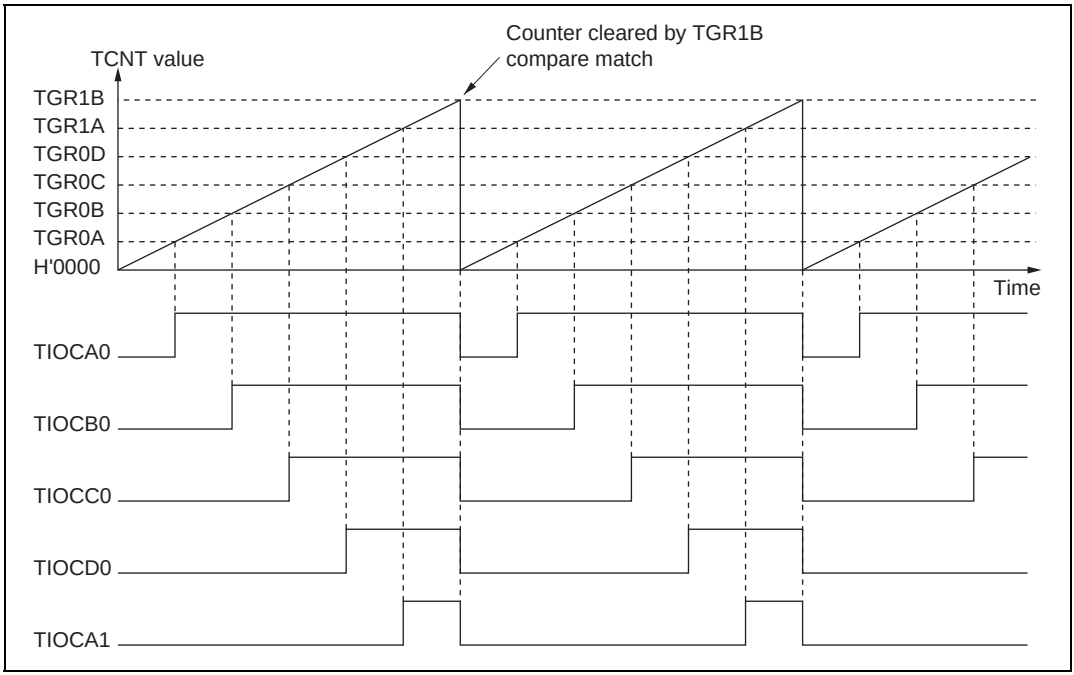


Figure 10.26 Example of PWM Mode Operation (2)

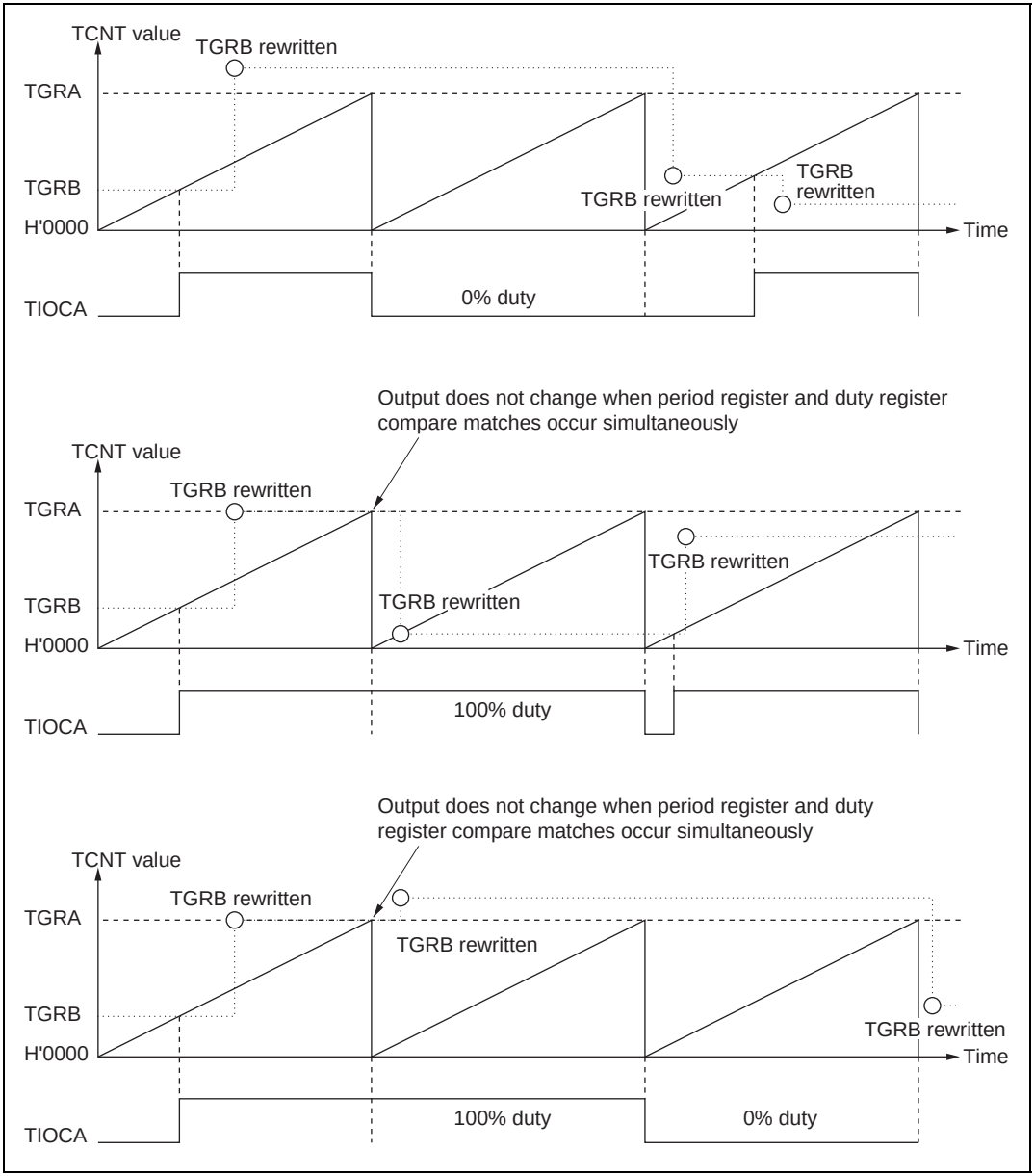


Figure 10.27 Examples of PWM Mode Operation (3)

In phase counting mode, the phase difference between two external clock inputs is detected and TCNT is incremented/decremented accordingly. This mode can be set for channels 1, 2, 4, and 5.

When phase counting mode is set, an external clock is selected as the counter input clock and TCNT operates as an up/down-counter regardless of the setting of bits TPSC2 to TPSC0 and bits CKEG1 and CKEG0 in TCR. However, the functions of bits CCLR1 and CCLR0 in TCR, and of TIOR, TIER, and TGR are valid, and input capture/compare match and interrupt functions can be used.

When overflow occurs while TCNT is counting up, the TCFV flag in TSR is set; when underflow occurs while TCNT is counting down, the TCFU flag is set.

The TCFD bit in TSR is the count direction flag. Reading the TCFD flag provides an indication of whether TCNT is counting up or down.

Table 10.8 shows the correspondence between external clock pins and channels.

Table 10.8 Phase Counting Mode Clock Input Pins

	External Clock Pins	
Channels	**A-Phase**	**B-Phase**
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 example of the phase counting mode setting procedure.

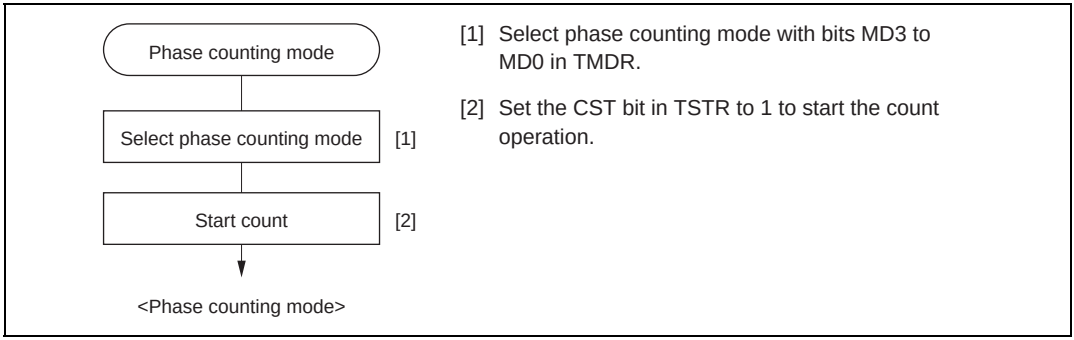


Figure 10.28 Example of Phase Counting Mode Setting Procedure

according to the count conditions.

- Phase counting mode 1

  Figure 10.29 shows an example of phase counting mode 1 operation, and table 10.9 summarizes the TCNT up/down-count conditions.

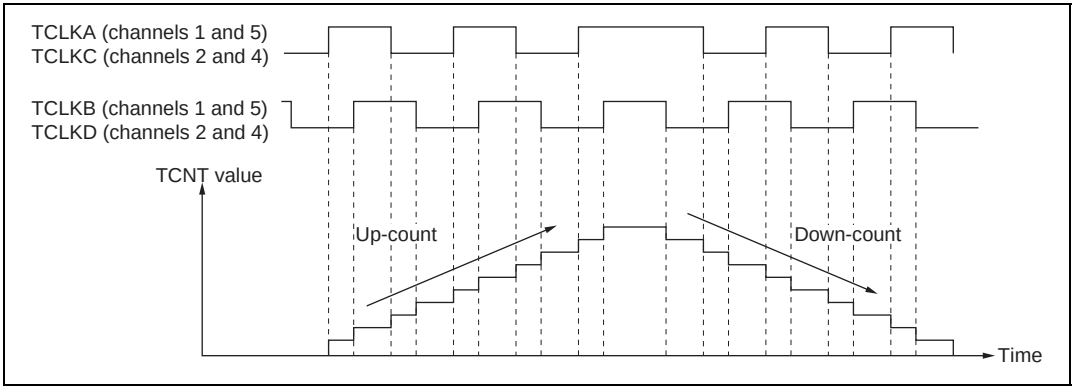


Figure 10.29 Example of Phase Counting Mode 1 Operation

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	↓	
↑	Low level	
↓	High level	
High level	↓	Down-count
Low level	↑	
↑	High level	
↓	Low level	

Legend:

↑: Rising edge

↓: Falling edge

summarizes the TCNT up/down-count conditions.

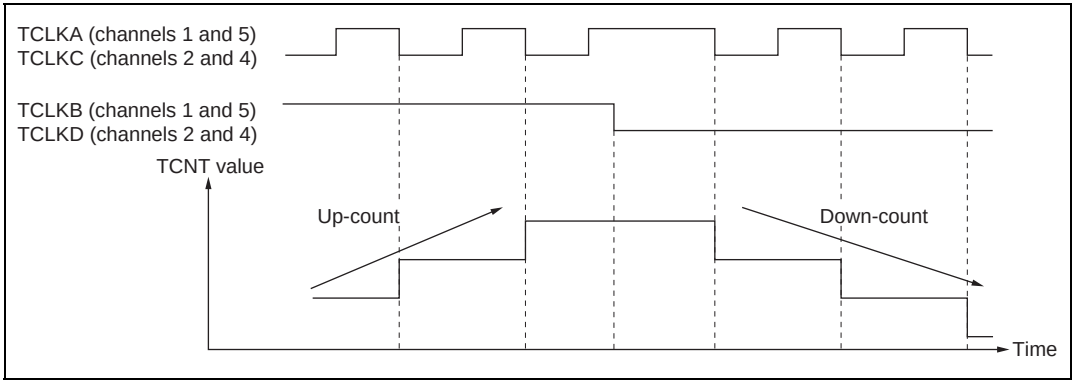


Figure 10.30 Example of Phase Counting Mode 2 Operation

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	↓	
↑	Low level	
↓	High level	Up-count
High level	↓	Don't care
Low level	↑	
↑	High level	
↓	Low level	Down-count

Legend:
↑ : Rising edge
↓ : Falling edge

summarizes the TCNT up/down-count conditions.

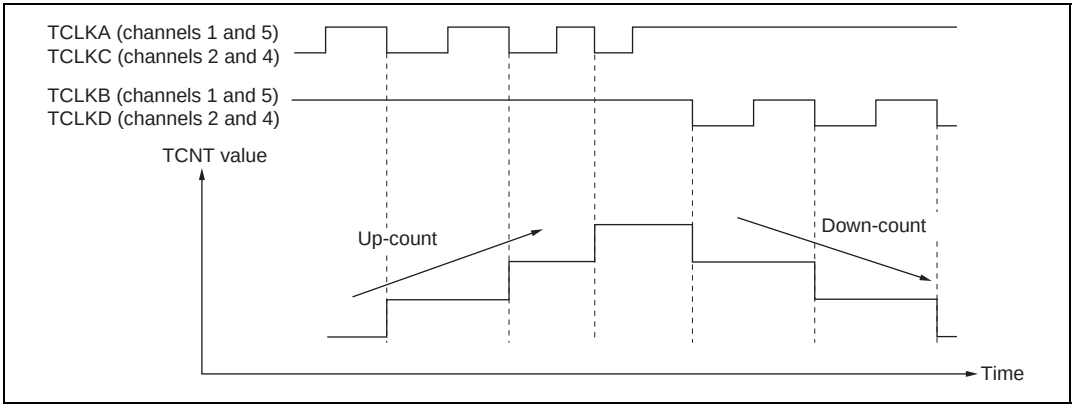


Figure 10.31 Example of Phase Counting Mode 3 Operation

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	↓	
↑	Low level	
↓	High level	Up-count
High level	↓	Down-count
Low level	↑	Don't care
↑	High level	
↓	Low level	

Legend:

↑: Rising edge

↓: Falling edge

summarizes the TCNT up/down-count conditions.

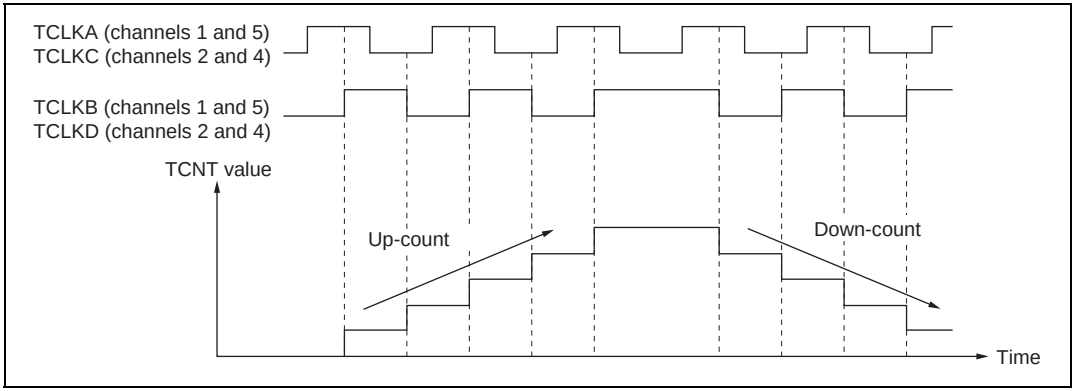


Figure 10.32 Example of Phase Counting Mode 4 Operation

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	↓	
↑	Low level	Don't care
↓	High level	
High level	↓	Down-count
Low level	↑	
↑	High level	Don't care
↓	Low level	

Legend:
↑ : Rising edge
↓ : Falling edge

motor 2-phase encoder pulses in order to detect the position or speed.

Channel 1 is set to phase counting mode 1, and the encoder pulse A-phase and B-phase are input to TCLKA and TCLKB.

Channel 0 operates with TCNT counter clearing by TGR0C compare match; TGR0A and TGR0C are used for the compare match function, and are set with the speed control period and position control period. TGR0B is used for input capture, with TGR0B and TGR0D operating in buffer mode. The channel 1 counter input clock is designated as the TGR0B input capture source, and 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 and TGR0C compare matches are selected as the input capture source, and store the up/down-counter values for the control periods.

This procedure enables accurate position/speed detection to be achieved.

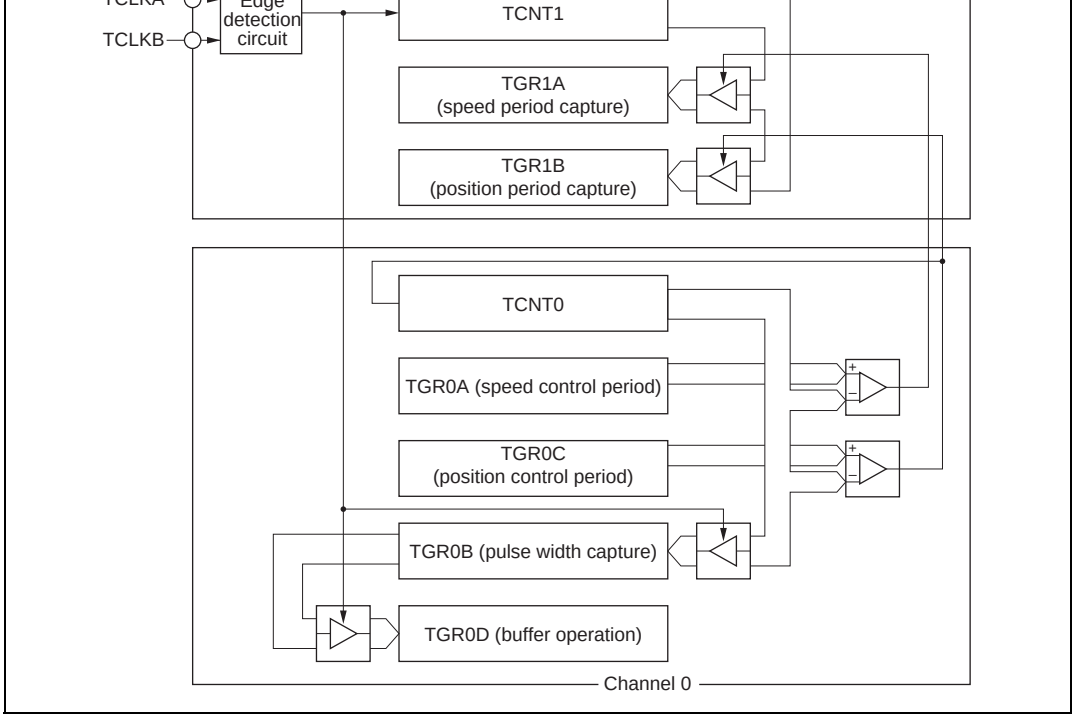


Figure 10.33 Phase Counting Mode Application Example

10.5 Interrupts

10.5.1 Interrupt Sources and Priorities

There are three kinds of TPU interrupt source: TGR input capture/compare match, TCNT overflow, and TCNT underflow. Each interrupt source has its own status flag and enable/disable bit, allowing generation of interrupt request signals to be enabled or disabled individually.

When an interrupt request is generated, the corresponding status flag in TSR is set to 1. If the corresponding enable/disable bit in TIER is set to 1 at this time, an interrupt is requested. The interrupt request is cleared by clearing the status flag to 0.

Relative channel priorities can be changed by the interrupt controller, but the priority order within a channel is fixed. For details, see section 5, Interrupt Controller.

Table 10.13 lists the TPU interrupt sources.

Channel	Interrupt Source	Description	DMAC Activation	DTC Activation	Priority
0	TGI0A	TGR0A input capture/compare match	Possible	Possible	High
	TGI0B	TGR0B input capture/compare match	Not possible	Possible	
	TGI0C	TGR0C input capture/compare match	Not possible	Possible	
	TGI0D	TGR0D input capture/compare match	Not possible	Possible	
	TCI0V	TCNT0 overflow	Not possible	Not possible	
1	TGI1A	TGR1A input capture/compare match	Possible	Possible	
	TGI1B	TGR1B input capture/compare match	Not possible	Possible	
	TCI1V	TCNT1 overflow	Not possible	Not possible	
	TCI1U	TCNT1 underflow	Not possible	Not possible	
2	TGI2A	TGR2A input capture/compare match	Possible	Possible	
	TGI2B	TGR2B input capture/compare match	Not possible	Possible	
	TCI2V	TCNT2 overflow	Not possible	Not possible	
	TCI2U	TCNT2 underflow	Not possible	Not possible	
3	TGI3A	TGR3A input capture/compare match	Possible	Possible	
	TGI3B	TGR3B input capture/compare match	Not possible	Possible	
	TGI3C	TGR3C input capture/compare match	Not possible	Possible	
	TGI3D	TGR3D input capture/compare match	Not possible	Possible	
	TCI3V	TCNT3 overflow	Not possible	Not possible	
4	TGI4A	TGR4A input capture/compare match	Possible	Possible	
	TGI4B	TGR4B input capture/compare match	Not possible	Possible	
	TCI4V	TCNT4 overflow	Not possible	Not possible	
	TCI4U	TCNT4 underflow	Not possible	Not possible	
5	TGI5A	TGR5A input capture/compare match	Possible	Possible	
	TGI5B	TGR5B input capture/compare match	Not possible	Possible	
	TCI5V	TCNT5 overflow	Not possible	Not possible	
	TCI5U	TCNT5 underflow	Not possible	Not possible	Low

Note: This table shows the initial state immediately after a reset. The relative channel priorities can be changed by the interrupt controller.

match on a particular channel. The interrupt request is cleared by clearing the TGF flag to 0. The TPU has 16 input capture/compare match interrupts, four each for channels 0 and 3, and two each for channels 1, 2, 4, and 5.

Overflow Interrupt: An interrupt is requested if the TCIEV bit in TIER is set to 1 when the TCFV flag in TSR is set to 1 by the occurrence of TCNT overflow on a channel. The interrupt request is cleared by clearing the TCFV flag to 0. The TPU has six overflow interrupts, one for each channel.

Underflow Interrupt: An interrupt is requested if the TCIEU bit in TIER is set to 1 when the TCFU flag in TSR is set to 1 by the occurrence of TCNT underflow on a channel. The interrupt request is cleared by clearing the TCFU flag to 0. The TPU has four underflow interrupts, one each for channels 1, 2, 4, and 5.

10.5.2 DTC/DMAC Activation

DTC Activation: The DTC can be activated by the TGR input capture/compare match interrupt for a channel. For details, see section 8, Data Transfer Controller.

A total of 16 TPU input capture/compare match interrupts can be used as DTC activation sources, four each for channels 0 and 3, and two each for channels 1, 2, 4, and 5.

DMAC Activation: The DMAC can be activated by the TGRA input capture/compare match interrupt for a channel. For details, see section 7, DMA Controller.

In the TPU, a total of six TGRA input capture/compare match interrupts can be used as DMAC activation sources, one for each channel.

10.5.3 A/D Converter Activation

The A/D converter can be activated by the TGRA input capture/compare match for a channel.

If the TTGE bit in TIER is set to 1 when the TGFA flag in TSR is set to 1 by the occurrence of a TGRA input capture/compare match on a particular channel, a request to start A/D conversion is sent to the A/D converter. If the TPU conversion start trigger has been selected on the A/D converter side at this time, A/D conversion is started.

In the TPU, a total of six TGRA input capture/compare match interrupts can be used as A/D converter conversion start sources, one for each channel.

10.6.1 Input/Output Timing

TCNT Count Timing: Figure 10.34 shows TCNT count timing in internal clock operation, and figure 10.35 shows TCNT count timing in external clock operation.

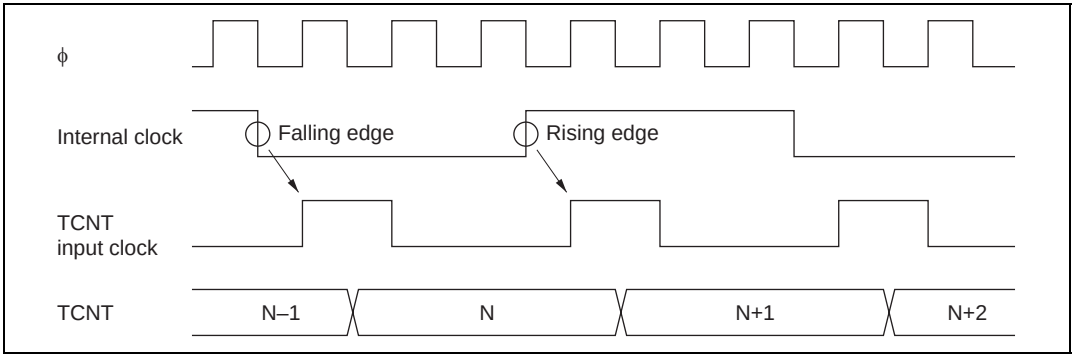


Figure 10.34 Count Timing in Internal Clock Operation

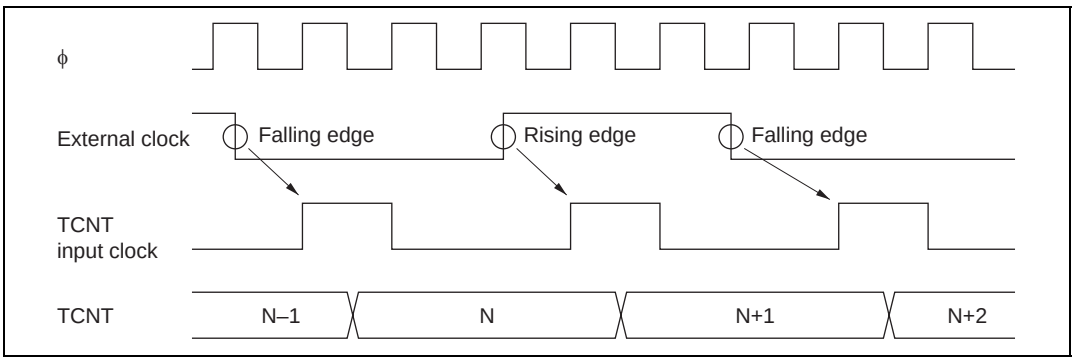


Figure 10.35 Count Timing in External Clock Operation

When a compare match signal is generated, the output value set in TIOR is output at the output compare output pin (TIOC pin). After a match between TCNT and TGR, the compare match signal is not generated until the TCNT input clock is generated.

Figure 10.36 shows output compare output timing.

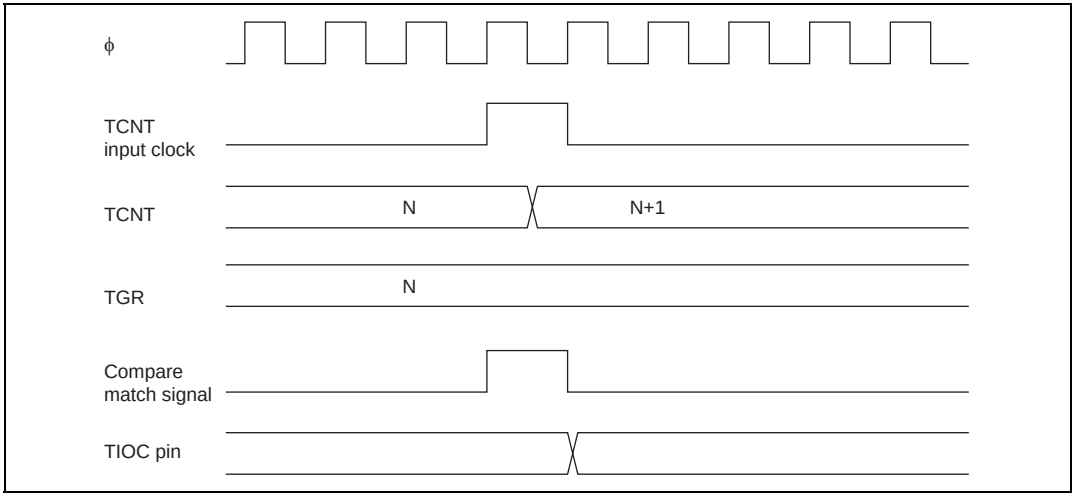


Figure 10.36 Output Compare Output Timing

Input Capture Signal Timing: Figure 10.37 shows input capture signal timing.

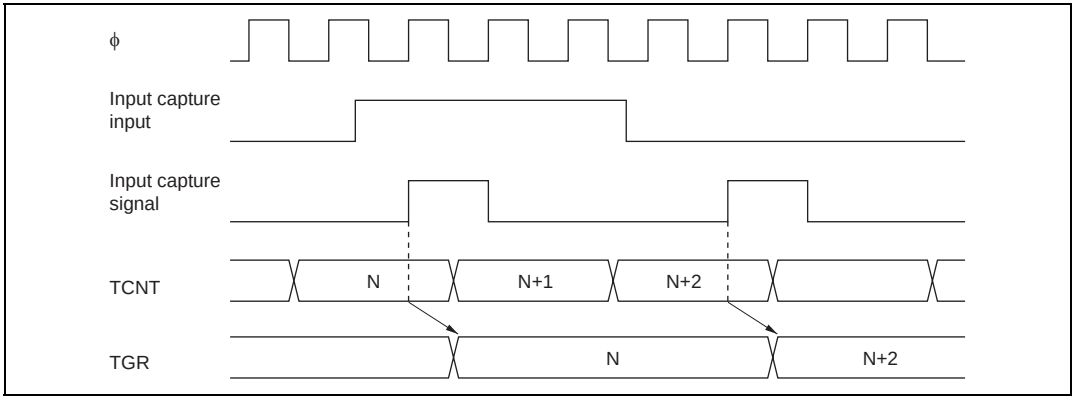


Figure 10.37 Input Capture Input Signal Timing

the timing when counter clearing by input capture occurrence is specified.

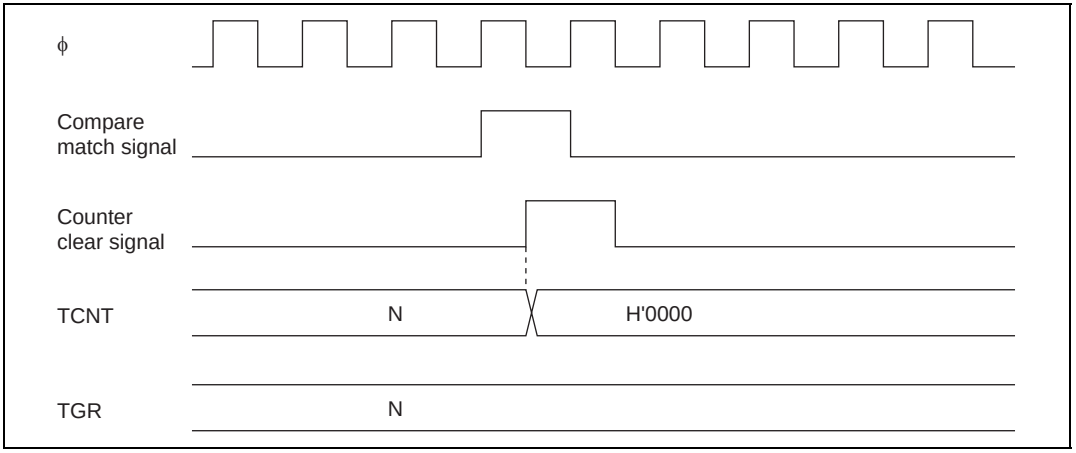


Figure 10.38 Counter Clear Timing (Compare Match)

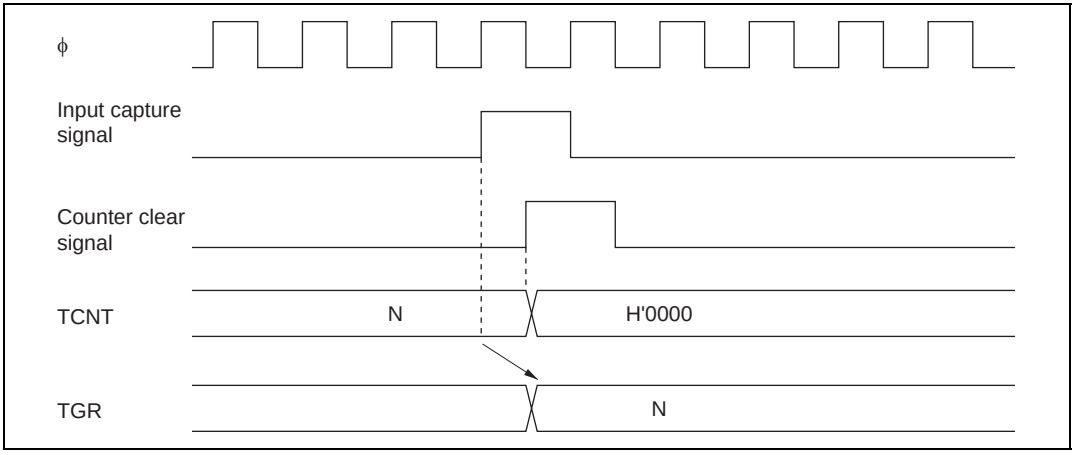


Figure 10.39 Counter Clear Timing (Input Capture)

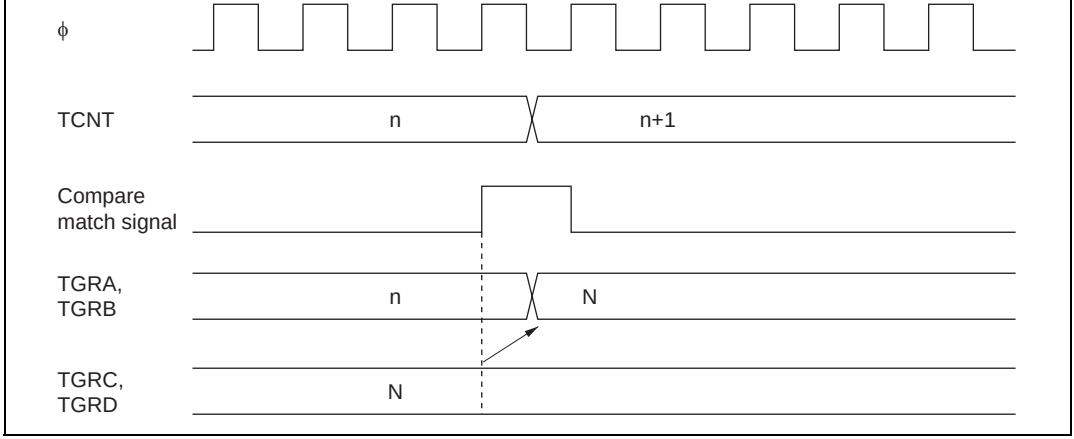


Figure 10.40 Buffer Operation Timing (Compare Match)

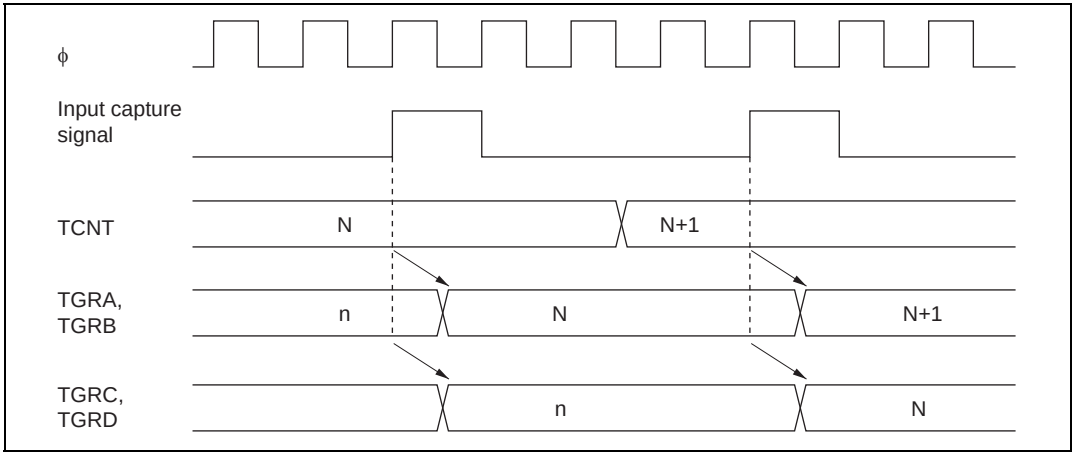


Figure 10.41 Buffer Operation Timing (Input Capture)

TGF Flag Setting Timing in Case of Compare Match: Figure 10.42 shows the timing for setting of the TGF flag in TSR by compare match occurrence, and TGI interrupt request signal timing.

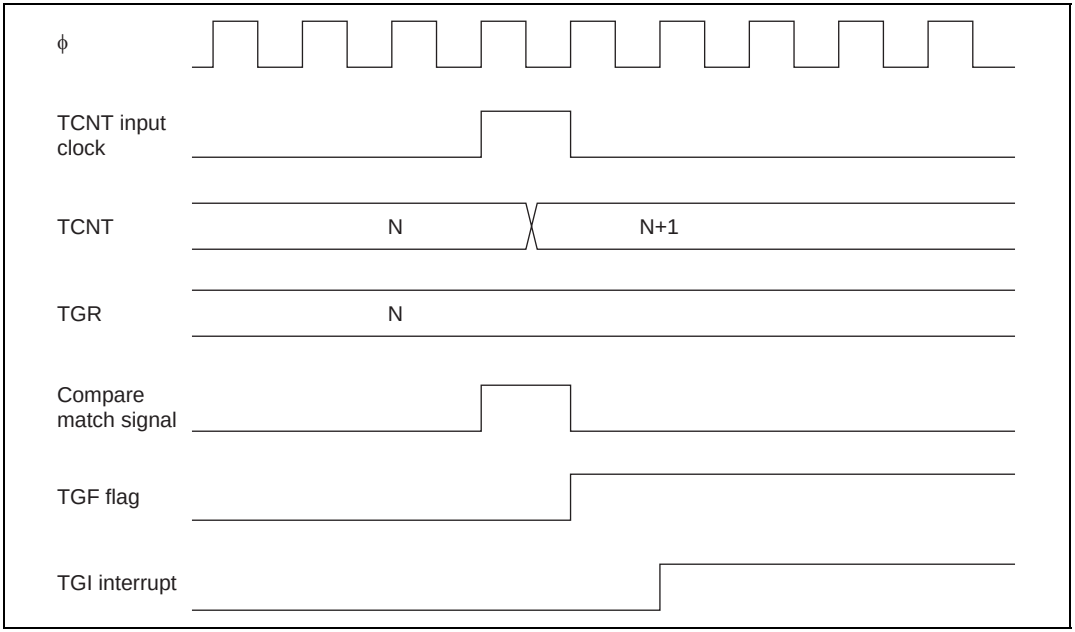


Figure 10.42 TGI Interrupt Timing (Compare Match)

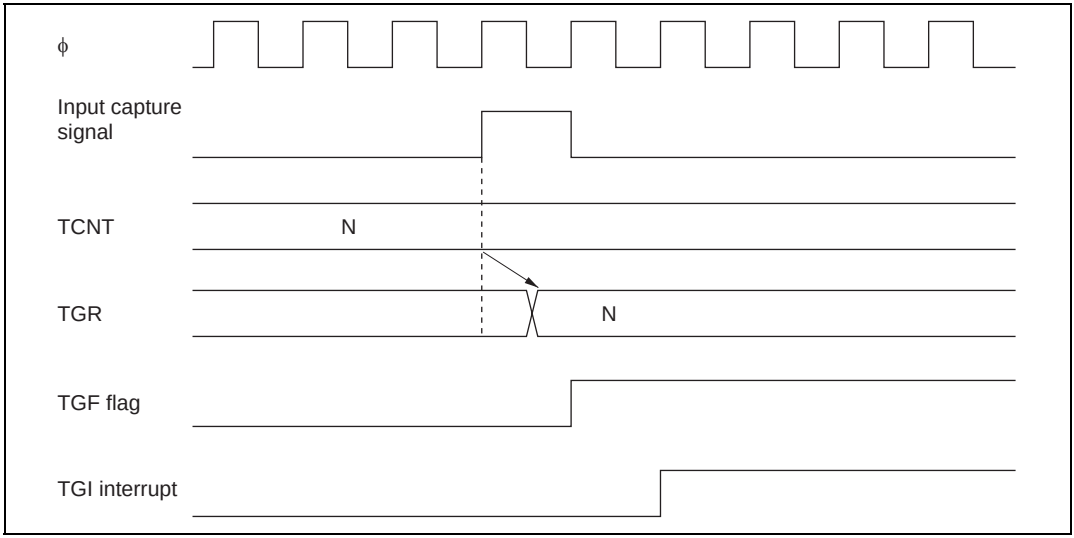


Figure 10.43 TGI Interrupt Timing (Input Capture)

Figure 10.45 shows the timing for setting of the TCFU flag in TSR by underflow occurrence, and TCIU interrupt request signal timing.

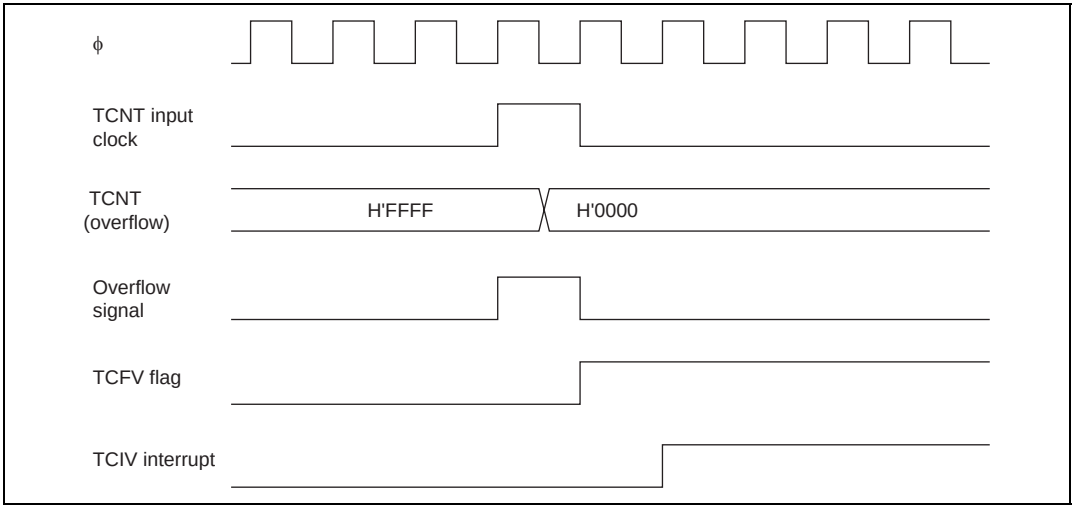


Figure 10.44 TCIV Interrupt Setting Timing

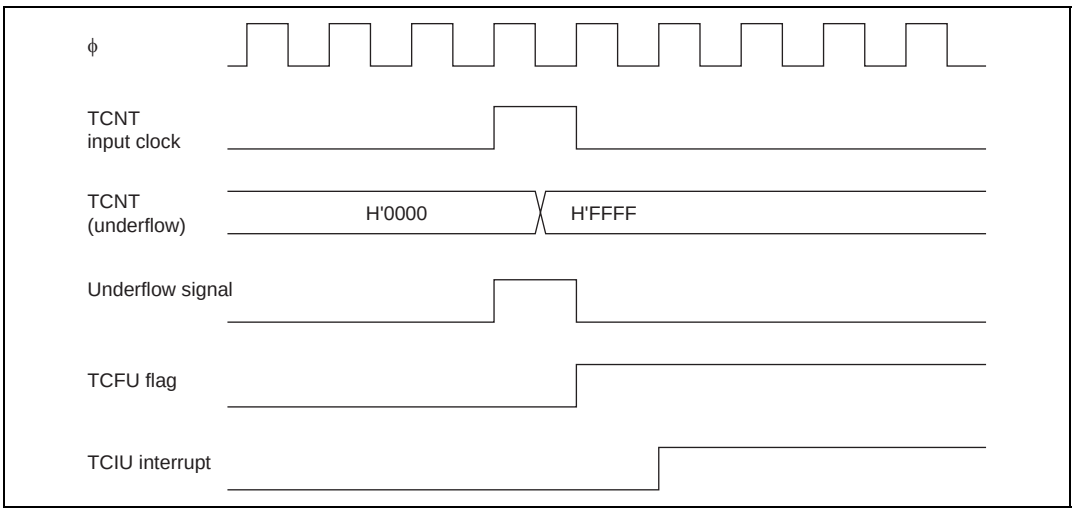


Figure 10.45 TCIU Interrupt Setting Timing

shows the timing for status flag clearing by the CPU, and figure 10.47 shows the timing for status flag clearing by the DTC or DMAC.

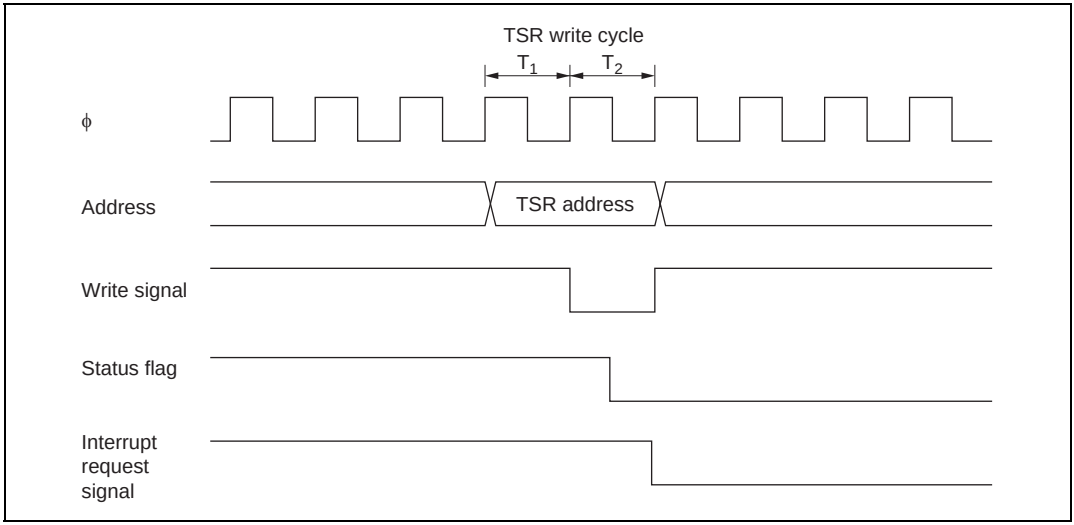


Figure 10.46 Timing for Status Flag Clearing by CPU

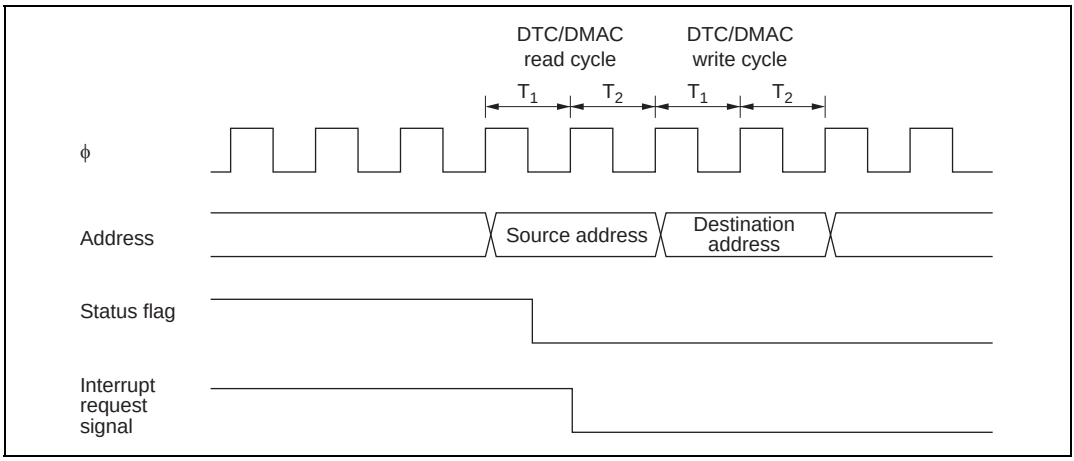


Figure 10.47 Timing for Status Flag Clearing by DTC/DMAC Activation

Note that the kinds of operation and contention described below can occur during TPU operation.

Input Clock Restrictions: The input clock pulse width must be at least 1.5 states in the case of single-edge detection, and at least 2.5 states in the case of both-edge detection. The TPU will not operate properly with a narrower pulse width.

In phase counting mode, the phase difference and overlap between the two input clocks must be at least 1.5 states, and the pulse width must be at least 2.5 states. Figure 10.48 shows the input clock conditions in phase counting mode.

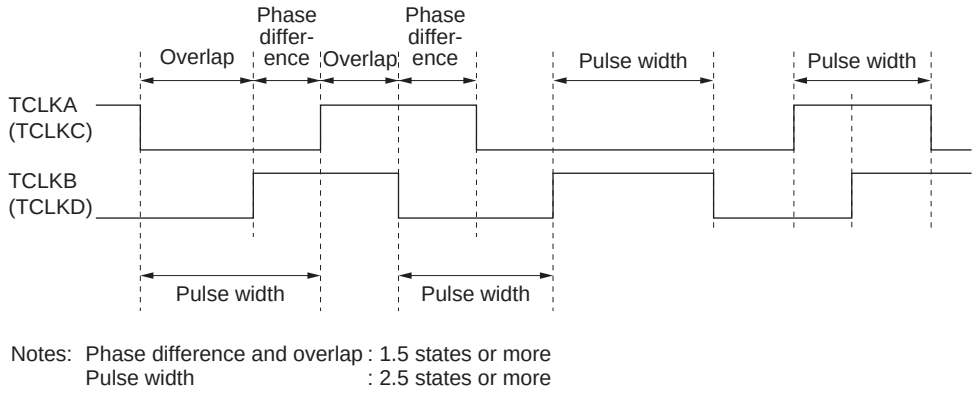


Figure 10.48 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

Caution on Period Setting: When counter clearing by compare match is set, TCNT is cleared in the final state in which it matches the TGR value (the point at which the count value matched by TCNT is updated). Consequently, the actual counter frequency is given by the following formula:

$$f = \frac{\phi}{(N + 1)}$$

Where f : Counter frequency
ϕ : Operating frequency
N : TGR set value

write is not performed.

Figure 10.49 shows the timing in this case.

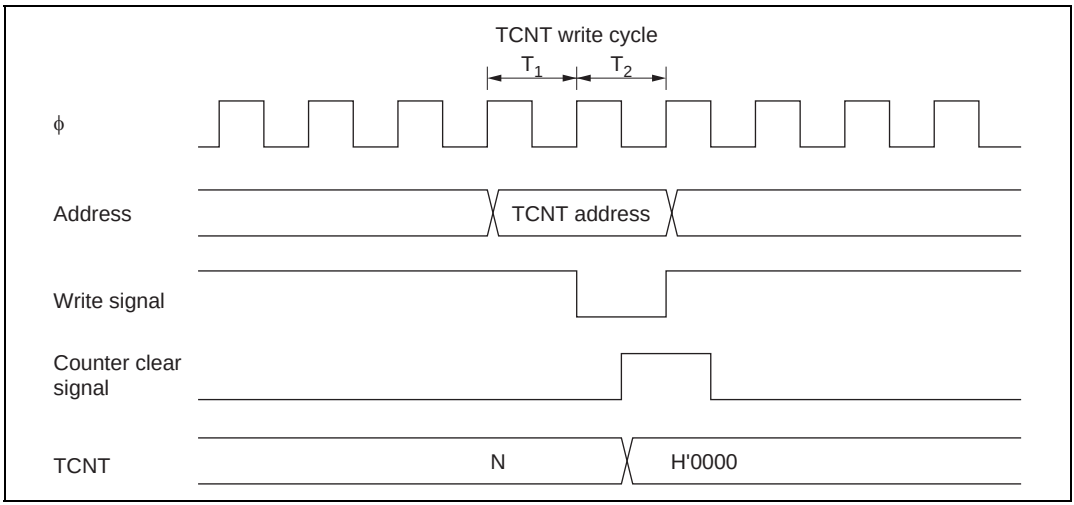


Figure 10.49 Contention between TCNT Write and Clear Operations

Figure 10.50 shows the timing in this case.

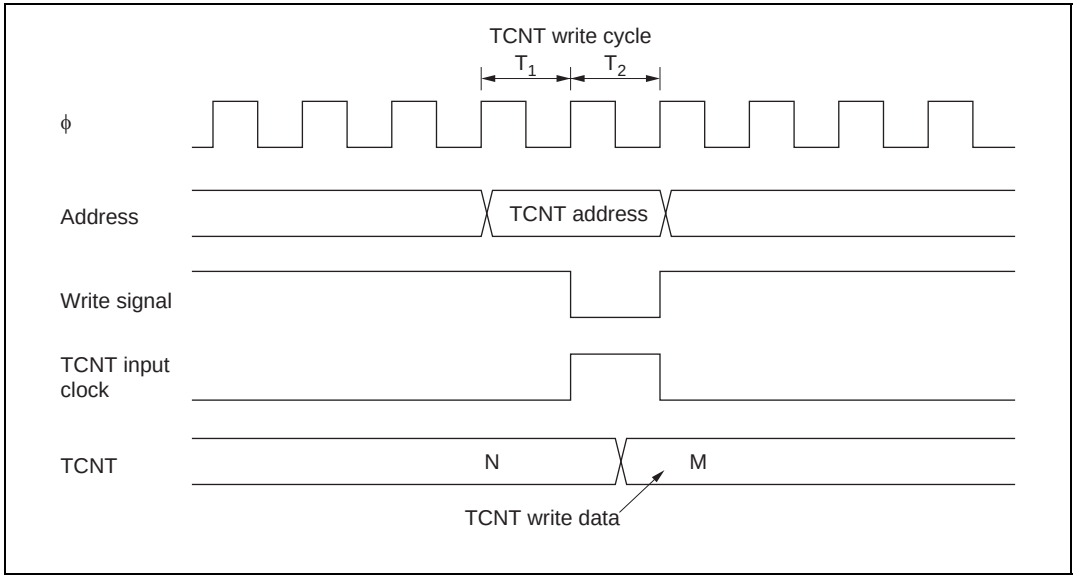


Figure 10.50 Contention between TCNT Write and Increment Operations

inhibited. A compare match does not occur even if the same value as before is written.

Figure 10.51 shows the timing in this case.

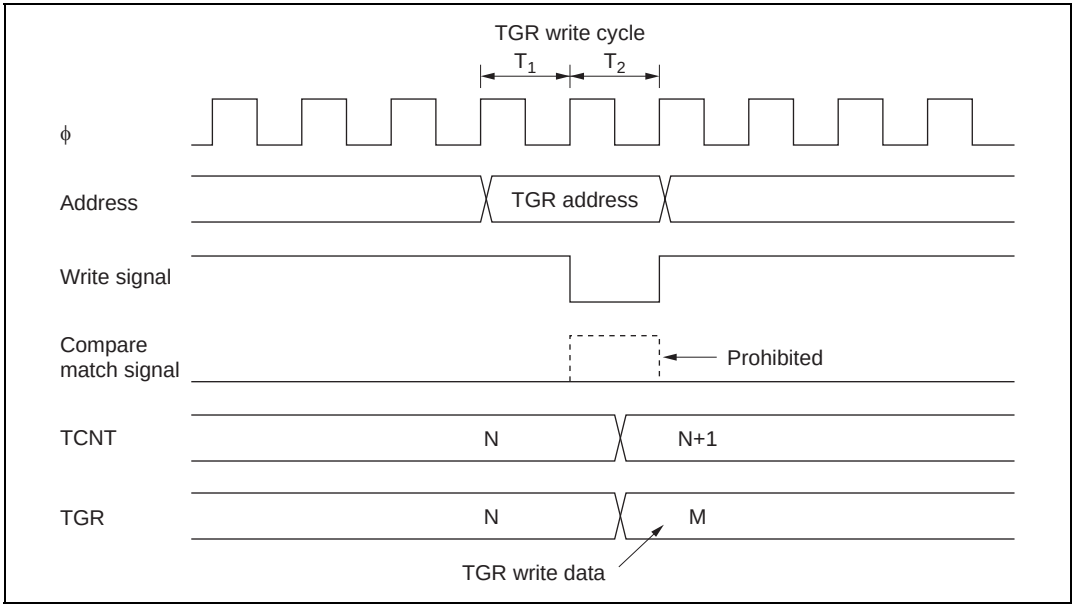


Figure 10.51 Contention between TGR Write and Compare Match

data prior to the write.

Figure 10.52 shows the timing in this case.

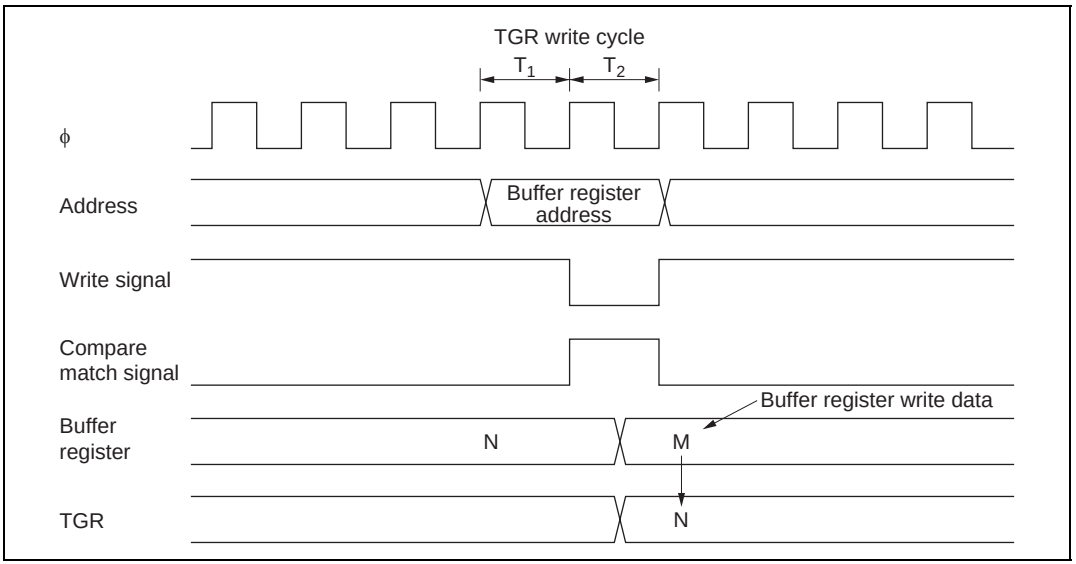


Figure 10.52 Contention between Buffer Register Write and Compare Match

Figure 10.53 shows the timing in this case.

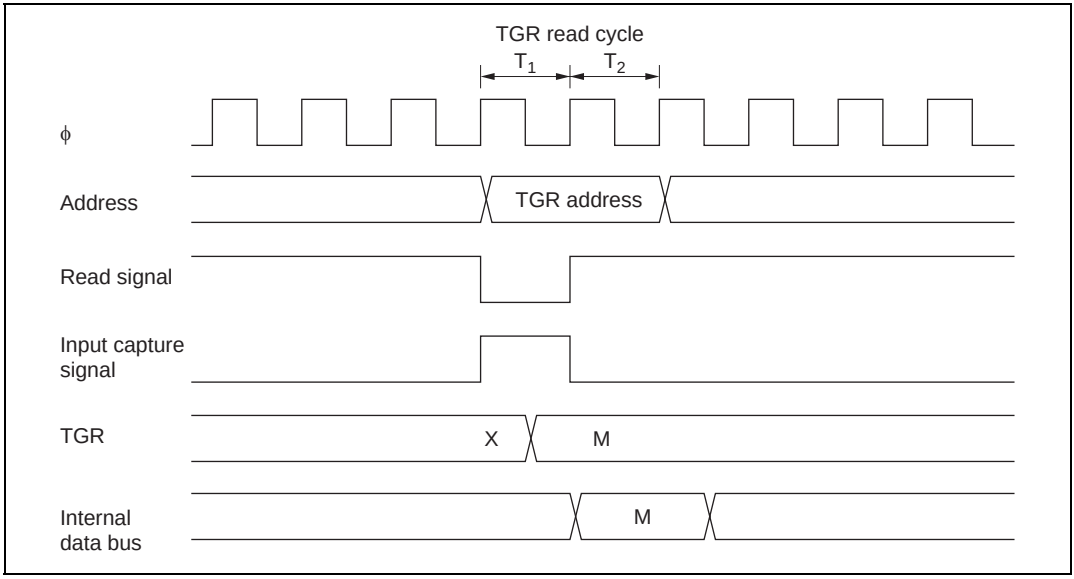


Figure 10.53 Contention between TGR Read and Input Capture

TGR is not performed.

Figure 10.54 shows the timing in this case.

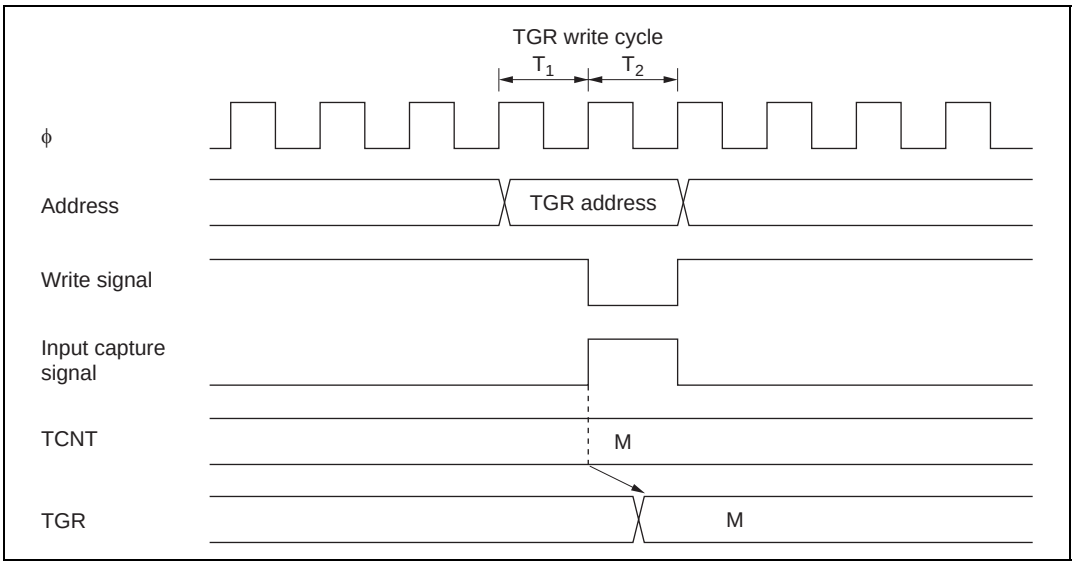


Figure 10.54 Contention between TGR Write and Input Capture

write to the buffer register is not performed.

Figure 10.55 shows the timing in this case.

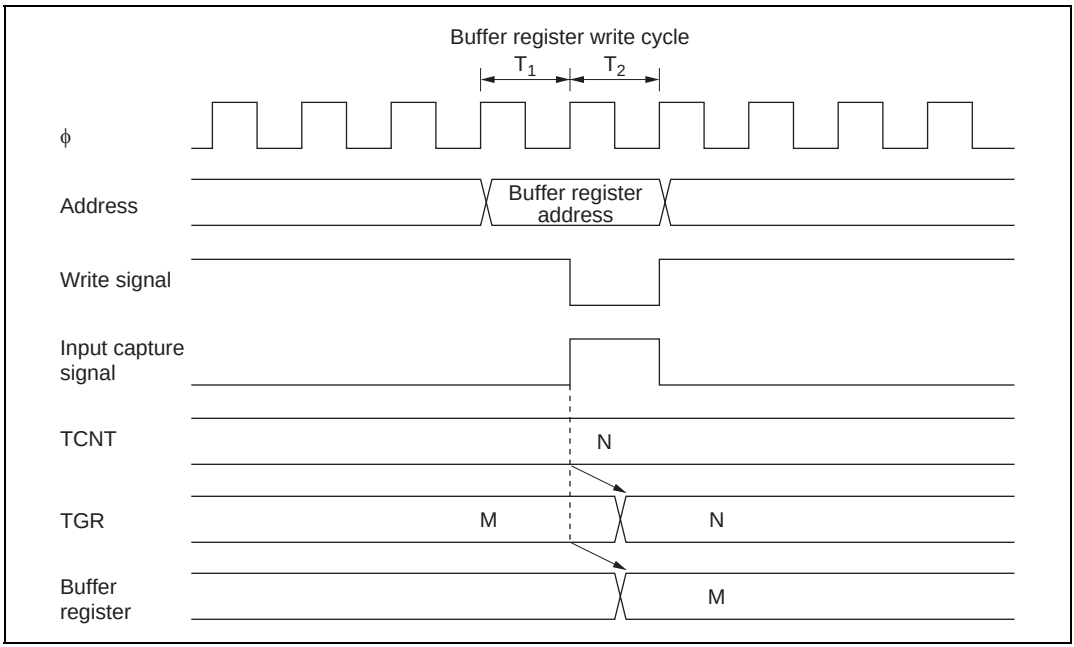


Figure 10.55 Contention between Buffer Register Write and Input Capture

takes precedence.

Figure 10.56 shows the operation timing when a TGR compare match is specified as the clearing source, and H'FFFF is set in TGR.

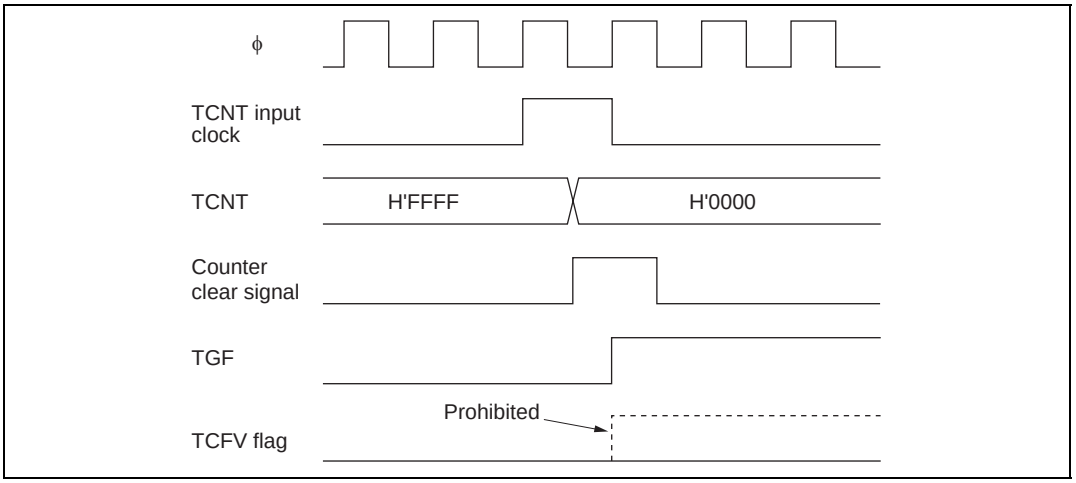


Figure 10.56 Contention between Overflow and Counter Clearing

precedence and the TCFV/TCFU flag in TSR is not set.

Figure 10.57 shows the operation timing when there is contention between TCNT write and overflow.

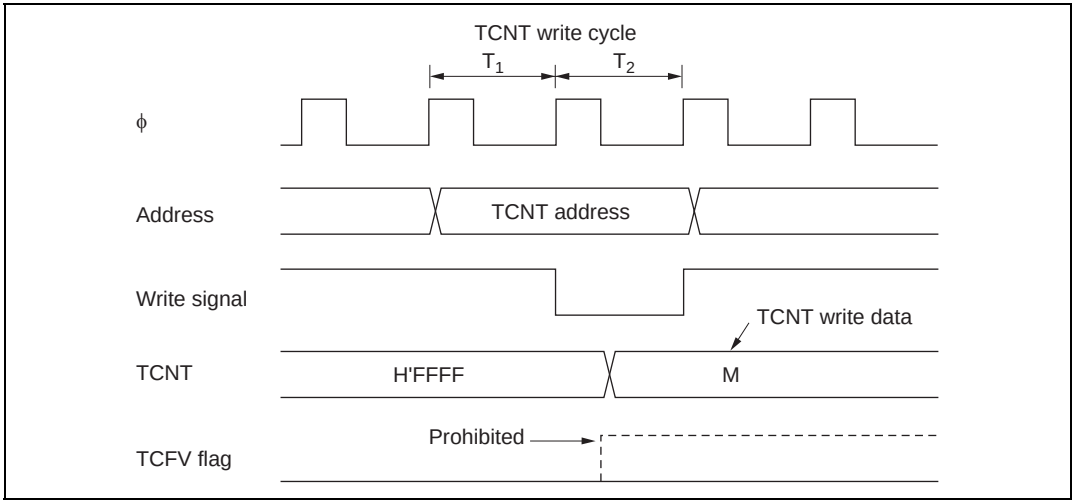


Figure 10.57 Contention between TCNT Write and Overflow

Multiplexing of I/O Pins: In the chip, the TCLKA input pin is multiplexed with the TIOCC0 I/O pin, the TCLKB input pin with the TIOCD0 I/O pin, the TCLKC input pin with the TIOCB1 I/O pin, and the TCLKD input pin with the TIOCB2 I/O pin. When an external clock is input, compare match output should not be performed from a multiplexed pin.

Interrupts and Module Stop Mode: If module stop mode is entered when an interrupt has been requested, it will not be possible to clear the CPU interrupt source or the DMAC or DTC activation source. Interrupts should therefore be disabled before entering module stop mode.

11.1 Overview

The chip has a built-in programmable pulse generator (PPG) that provides pulse outputs by using the 16-bit timer-pulse unit (TPU) as a time base. The PPG pulse outputs are divided into 4-bit groups (group 3 to group 0) that can operate both simultaneously and independently.

11.1.1 Features

PPG features are listed below.

- 16-bit output data
  - Maximum 16-bit data can be output, and output can be enabled on a bit-by-bit basis
- Four output groups
  - Output trigger signals can be selected in 4-bit groups to provide up to four different 4-bit outputs
- Selectable output trigger signals
  - Output trigger signals can be selected for each group from the compare match signals of 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) and DMA controller (DMAC)
  - The compare match signals selected as output trigger signals can activate the DTC or DMAC for sequential output of data without CPU intervention
- Inverted output can be set
  - 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 exiting module stop mode

Figure 11.1 shows a block diagram of the PPG.

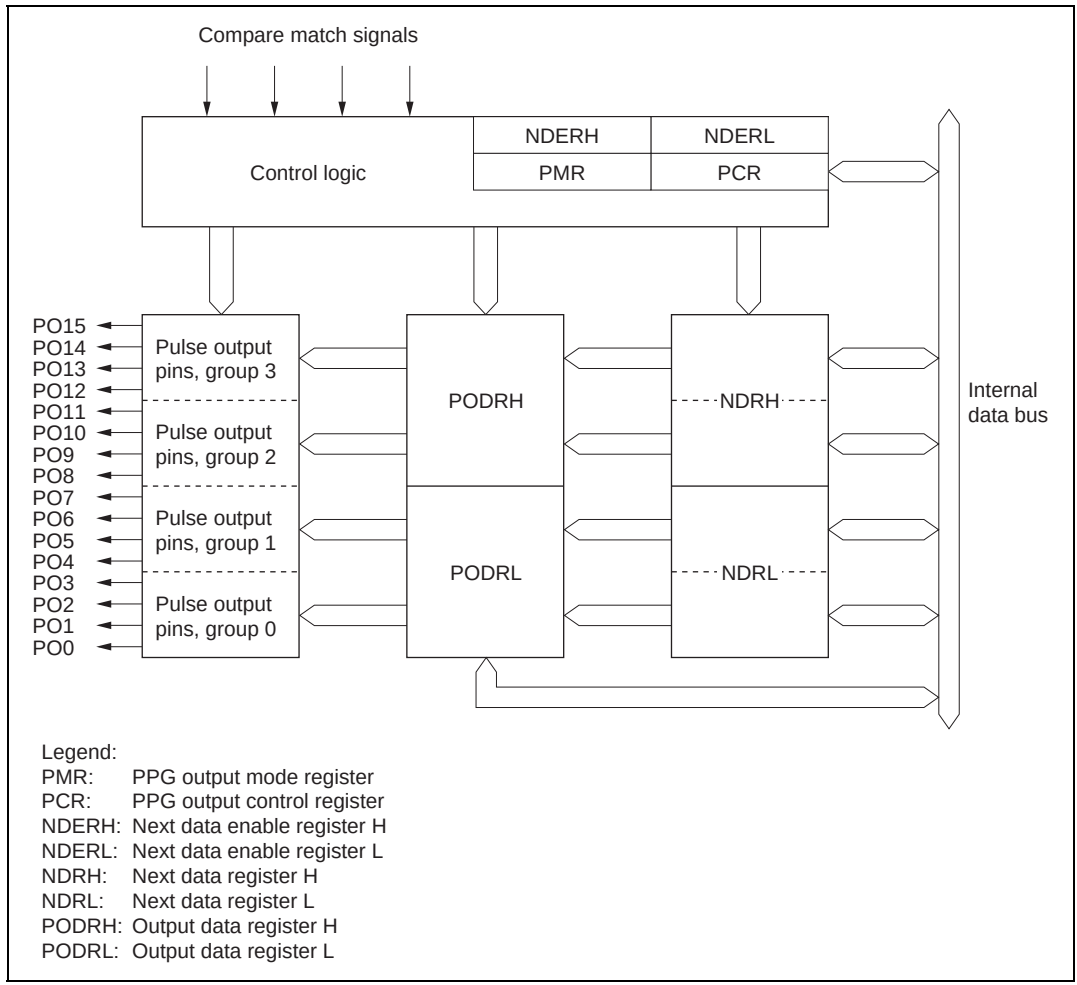


Figure 11.1 Block Diagram of PPG

Table 11.1 summarizes the PPG pins.

Table 11.1 PPG Pins

Name	Symbol	I/O	Function
Pulse output 0	PO0	Output	Group 0 pulse output
Pulse output 1	PO1	Output	
Pulse output 2	PO2	Output	
Pulse output 3	PO3	Output	
Pulse output 4	PO4	Output	Group 1 pulse output
Pulse output 5	PO5	Output	
Pulse output 6	PO6	Output	
Pulse output 7	PO7	Output	
Pulse output 8	PO8	Output	Group 2 pulse output
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	

Table 11.2 summarizes the PPG registers.

Table 11.2 PPG Registers

Name	Abbreviation	R/W	Initial Value	Address*1
PPG output control register	PCR	R/W	H'FF	H'FF46
PPG output mode register	PMR	R/W	H'F0	H'FF47
Next data enable register H	NDERH	R/W	H'00	H'FF48
Next data enable register L	NDERL	R/W	H'00	H'FF49
Output data register H	PODRH	R/(W)*2	H'00	H'FF4A
Output data register L	PODRL	R/(W) *2	H'00	H'FF4B
Next data register H	NDRH	R/W	H'00	H'FF4C/ H'FF4E*3
Next data register L	NDRL	R/W	H'00	H'FF4D/ H'FF4F*3
Port 1 data direction register	P1DDR	W	H'00	H'FEB0
Port 2 data direction register	P2DDR	W	H'00	H'FEB1
Module stop control register	MSTPCR	R/W	H'3FFF	H'FF3C

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 the PCR setting, the NDRH address is H'FF4C. When the output triggers are different, the NDRH address is H'FF4E for group 2 and H'FF4C for group 3.

   Similarly, when the same output trigger is selected for pulse output groups 0 and 1 by the PCR setting, the NDRL address is H'FF4D. When the output triggers are different, the NDRL address is H'FF4F for group 0 and H'FF4D for group 1.

11.2.1 Next Data Enable Registers H and L (NDERH, NDERL)

NDERH

Bit	:	7	6	5	4	3	2	1	0
		NDER15	NDER14	NDER13	NDER12	NDER11	NDER10	NDER9	NDER8
Initial value	:	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

NDERL

Bit	:	7	6	5	4	3	2	1	0
		NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1	NDER0
Initial value	:	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

NDERH and NDERL are 8-bit readable/writable registers that enable or disable pulse output on a bit-by-bit basis.

If a bit is enabled for pulse output by NDERH or NDERL, the NDR value is automatically transferred to the corresponding PODR bit when the TPU compare match event specified by PCR occurs, updating the output value. If pulse output is disabled, the bit value is not transferred from 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 mode. They are not initialized in software standby mode.

NDERH Bits 7 to 0—Next Data Enable 15 to 8 (NDER15 to NDER8): These bits enable or 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 are not transferred to POD15 to POD8) (Initial value)
1	Pulse outputs PO15 to PO8 are enabled (NDR15 to NDR8 are transferred to POD15 to POD8)

Bits 7 to 0 NDER7 to NDER0	Description
0	Pulse outputs PO7 to PO0 are disabled (NDR7 to NDR0 are not transferred to POD7 to POD0) (Initial value)
1	Pulse outputs PO7 to PO0 are enabled (NDR7 to NDR0 are transferred to POD7 to POD0)

11.2.2 Output Data Registers H and L (PODRH, PODRL)

PODRH

Bit	:	7	6	5	4	3	2	1	0
		POD15	POD14	POD13	POD12	POD11	POD10	POD9	POD8
Initial value	:	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)*

PODRL

Bit	:	7	6	5	4	3	2	1	0
		POD7	POD6	POD5	POD4	POD3	POD2	POD1	POD0
Initial value	:	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)*

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 in pulse output.

NDRH and NDRL are 8-bit readable/writable registers that store the next data for pulse output. During pulse output, the contents of NDRH and NDRL are transferred to the corresponding bits in PODRH and PODRL when the TPU compare match event specified by PCR occurs. The NDRH and NDRL addresses differ depending on whether pulse output groups have the same output trigger or different output triggers. For details see section 11.2.4, Notes on NDR Access.

NDRH and NDRL are each initialized to H'00 by a reset and in hardware standby mode. They are 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 have the same output trigger or different output triggers.

Same Trigger for Pulse Output Groups: If pulse output groups 2 and 3 are triggered by the same compare match event, the NDRH address is H'FF4C. The upper 4 bits belong to group 3 and the lower 4 bits to group 2. Address H'FF4E consists entirely of reserved bits that cannot be modified and are always read as 1.

Address H'FF4C

Bit	:	7	6	5	4	3	2	1	0
		NDR15	NDR14	NDR13	NDR12	NDR11	NDR10	NDR9	NDR8
Initial value	:	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

Address H'FF4E

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	—	—	—	—
Initial value	:	1	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 NDRL address is H'FF4D. The upper 4 bits belong to group 1 and the lower 4 bits to group 0. Address H'FF4F consists entirely of reserved bits that cannot be modified and are always read as 1.

Bit	:	7	6	5	4	3	2	1	0
		NDR7	NDR6	NDR5	NDR4	NDR3	NDR2	NDR1	NDR0
Initial value	:	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

Address H'FF4F

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

Different Triggers for Pulse Output Groups: If pulse output groups 2 and 3 are triggered by different compare match events, the address of the upper 4 bits in NDRH (group 3) is H'FF4C and the address of the lower 4 bits (group 2) is H'FF4E. Bits 3 to 0 of address H'FF4C and bits 7 to 4 of address H'FF4E are reserved bits that cannot be modified and are always read as 1.

Address H'FF4C

Bit	:	7	6	5	4	3	2	1	0
		NDR15	NDR14	NDR13	NDR12	—	—	—	—
Initial value	:	0	0	0	0	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	—	—	—	—

Address H'FF4E

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	NDR11	NDR10	NDR9	NDR8
Initial value	:	1	1	1	1	0	0	0	0
R/W	:	—	—	—	—	R/W	R/W	R/W	R/W

If pulse output groups 0 and 1 are triggered by different compare match event, the address of the upper 4 bits in NDRL (group 1) is H'FF4D and the address of the lower 4 bits (group 0) is H'FF4F. Bits 3 to 0 of address H'FF4D and bits 7 to 4 of address H'FF4F are reserved bits that cannot be modified and are always read as 1.

Bit	:	7	6	5	4	3	2	1	0
		NDR7	NDR6	NDR5	NDR4	—	—	—	—
Initial value	:	0	0	0	0	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	—	—	—	—

Address H'FF4F

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	NDR3	NDR2	NDR1	NDR0
Initial value	:	1	1	1	1	0	0	0	0
R/W	:	—	—	—	—	R/W	R/W	R/W	R/W

11.2.5 PPG Output Control Register (PCR)

Bit	:	7	6	5	4	3	2	1	0
		G3CMS1	G3CMS0	G2CMS1	G2CMS0	G1CMS1	G1CMS0	G0CMS1	G0CMS0
Initial value	:	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

PCR is an 8-bit readable/writable register that selects output trigger signals for PPG outputs on a group-by-group basis.

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

Bits 7 and 6—Group 3 Compare Match Select 1 and 0 (G3CMS1, G3CMS0): These bits 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	(Initial value)

Bit 5 G2CMS1	Bit 4 G2CMS0	Description	
		Output Trigger for Pulse Output Group 2	
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	(Initial value)

Bits 3 and 2—Group 1 Compare Match Select 1 and 0 (G1CMS1, G1CMS0): These bits select the compare match that triggers pulse output group 1 (pins PO7 to PO4).

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	(Initial value)

Bits 1 and 0—Group 0 Compare Match Select 1 and 0 (G0CMS1, G0CMS0): These bits select the compare match that triggers pulse output group 0 (pins PO3 to PO0).

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	(Initial value)

Bit	:	7	6	5	4	3	2	1	0
		G3INV	G2INV	G1INV	G0INV	G3NOV	G2NOV	G1NOV	G0NOV
Initial value	:	1	1	1	1	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PMR is an 8-bit readable/writable register that selects pulse output inversion and non-overlapping operation for each group.

The output trigger period of a non-overlapping operation PPG output waveform is set in TGRB and the non-overlap margin is set in TGRA. The output values change at compare match A and B.

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 initialized in software standby mode.

Bit 7—Group 3 Inversion (G3INV): Selects direct output or inverted output for pulse output 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 PODRH)
1	Direct output for pulse output group 3 (high-level output at pin for a 1 in PODRH) (Initial value)

Bit 6—Group 2 Inversion (G2INV): Selects direct output or inverted output for pulse output 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 PODRH)
1	Direct output for pulse output group 2 (high-level output at pin for a 1 in PODRH) (Initial value)

Bit 5 G1INV	Description
0	Inverted output for pulse output group 1 (low-level output at pin for a 1 in PODRL)
1	Direct output for pulse output group 1 (high-level output at pin for a 1 in PODRL) (Initial value)

Bit 4—Group 0 Inversion (G0INV): Selects direct output or inverted output for pulse output group 0 (pins PO3 to PO0).

Bit 4 G0INV	Description
0	Inverted output for pulse output group 0 (low-level output at pin for a 1 in PODRL)
1	Direct output for pulse output group 0 (high-level output at pin for a 1 in PODRL) (Initial value)

Bit 3—Group 3 Non-Overlap (G3NOV): Selects normal or non-overlapping operation for pulse output group 3 (pins PO15 to PO12).

Bit 3 G3NOV	Description
0	Normal operation in pulse output group 3 (output values updated at compare match A in the selected TPU channel) (Initial value)
1	Non-overlapping operation in pulse output group 3 (independent 1 and 0 output at compare match A or B in the selected TPU channel)

Bit 2—Group 2 Non-Overlap (G2NOV): Selects normal or non-overlapping operation for pulse output group 2 (pins PO11 to PO8).

Bit 2 G2NOV	Description
0	Normal operation in pulse output group 2 (output values updated at compare match A in the selected TPU channel) (Initial value)
1	Non-overlapping operation in pulse output group 2 (independent 1 and 0 output at compare match A or B in the selected TPU channel)

Bit 1 G1NOV	Description
0	Normal operation in pulse output group 1 (output values updated at compare match A in the selected TPU channel) (Initial value)
1	Non-overlapping operation in pulse output group 1 (independent 1 and 0 output at compare match A or B in the selected TPU channel)

Bit 0—Group 0 Non-Overlap (G0NOV): Selects normal or non-overlapping operation for pulse output group 0 (pins PO3 to PO0).

Bit 0 G0NOV	Description
0	Normal operation in pulse output group 0 (output values updated at compare match A in the selected TPU channel) (Initial value)
1	Non-overlapping operation in pulse output group 0 (independent 1 and 0 output at 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	0
		P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DDR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W	W

P1DDR is an 8-bit write-only register, the individual bits of which specify input or output for the pins of port 1.

Port 1 is multiplexed with pins PO15 to PO8. Bits corresponding to pins used for PPG output must be set to 1. For further information about P1DDR, see section 9, I/O Port.

Bit	:	7	6	5	4	3	2	1	0
		P27DDR	P26DDR	P25DDR	P24DDR	P23DDR	P22DDR	P21DDR	P20DDR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W	W

P2DDR is an 8-bit write-only register, the individual bits of which specify input or output for the pins of port 2.

Port 2 is multiplexed with pins PO7 to PO0. Bits corresponding to pins used for PPG output must be set to 1. For further information about P2DDR, see section 9, I/O Port.

11.2.9 Module Stop Control Register (MSTPCR)

		MSTPCRH								MSTPCRL							
Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	:	0	0	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/W	R/W	R/W

MSTPCR is a 16-bit readable/writable register that performs module stop mode control.

When the MSTP11 bit in MSTPCR is set to 1, PPG operation stops at the end of the bus cycle and a transition is made to module stop mode. Registers cannot be read or written to in module stop mode. For details, see section 21.5, Module Stop Mode.

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

Bit 11—Module Stop (MSTP11): Specifies the PPG module stop mode.

Bit 11 MSTP11	Description	
0	PPG module stop mode cleared	
1	PPG module stop mode set	(Initial value)

11.3.1 Overview

PPG pulse output is enabled when the corresponding bits in P1DDR, P2DDR, and NDER are set to 1. In this state the corresponding PODR contents are output.

When the compare match event specified by PCR occurs, the corresponding NDR bit contents are transferred to PODR to update the output values.

Figure 11.2 illustrates the PPG output operation and table 11.3 summarizes the PPG operating conditions.

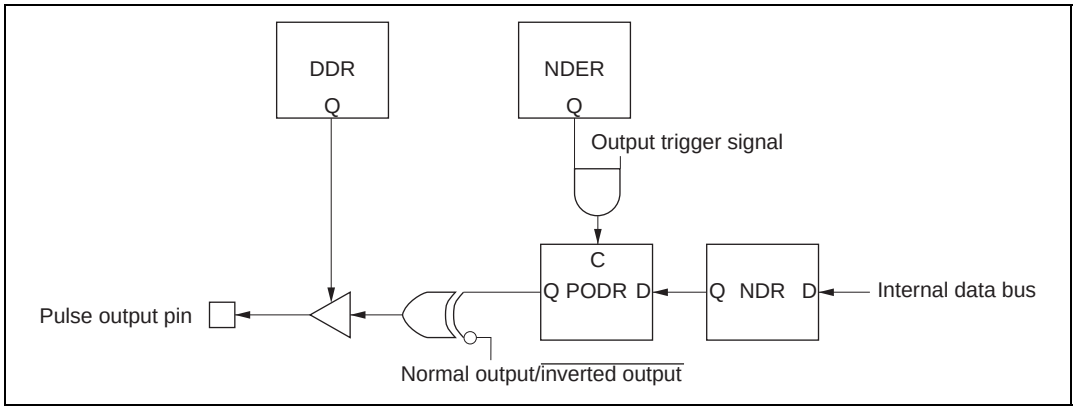


Figure 11.2 PPG Output Operation

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 when compare match occurs, the NDR bit value is transferred to the PODR bit)
	1	PPG pulse output

Sequential output of data of up to 16 bits is possible by writing new output data to NDR before the next compare match. For details of non-overlapping operation, see section 11.3.4, Non-Overlapping Pulse Output.

If pulse output is enabled, NDR contents are transferred to PODR and output when the specified compare match event occurs. Figure 11.3 shows the timing of these operations for the case of normal output in groups 2 and 3, triggered by compare match A.

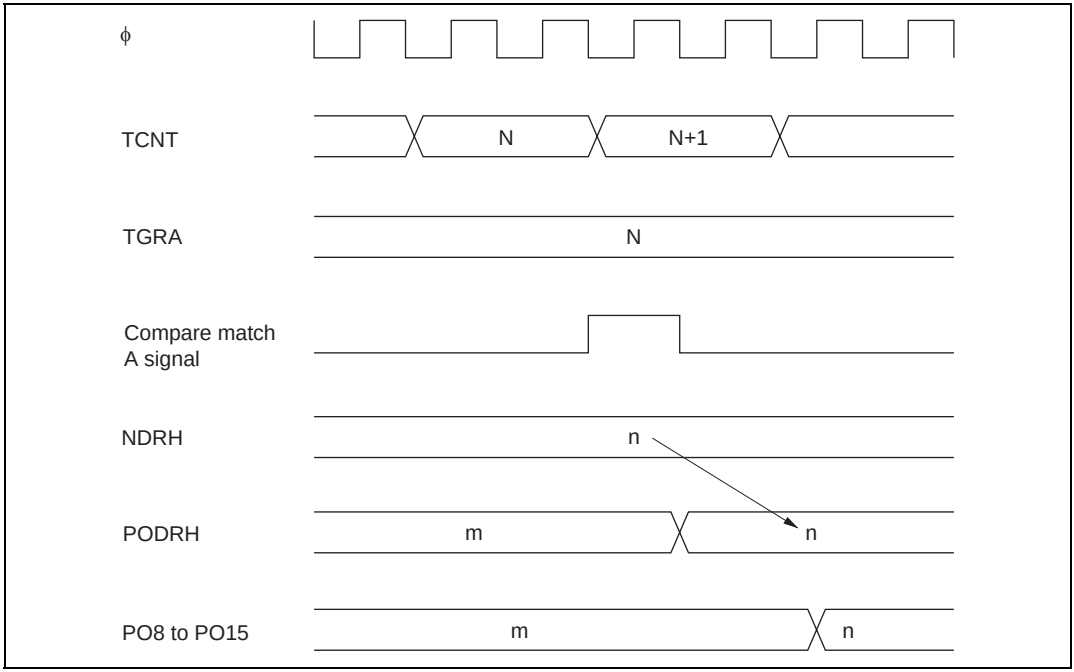
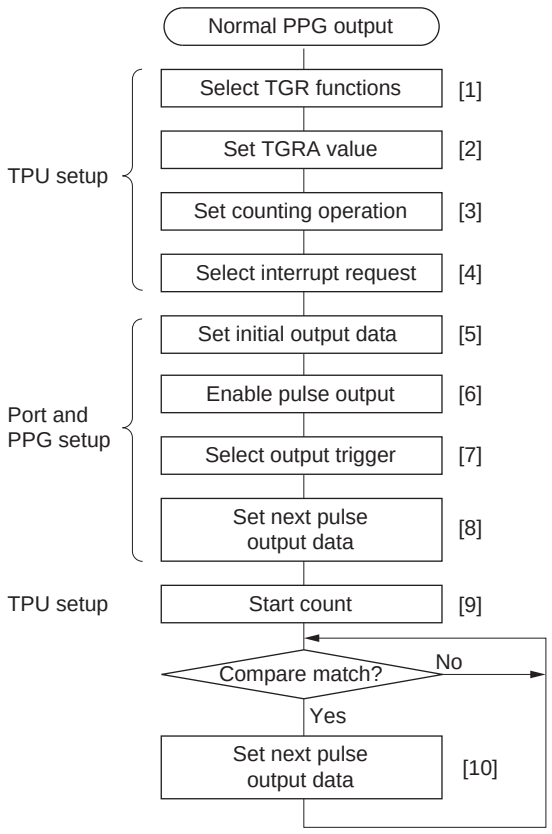


Figure 11.3 Timing of Transfer and Output of NDR Contents (Example)

Sample Setup Procedure for Normal Pulse Output: Figure 11.4 shows a sample procedure for setting up normal pulse output.



[1] Set TIOR to make TGRA an output compare register (with output disabled).

[2] Set the PPG output trigger period.

[3] Select the counter clock source with bits TPSC2 to TPSC0 in TCR. Select the counter clear source with bits CCLR1 and CCLR0.

[4] Enable the TGIA interrupt in TIER. The DTC or DMAC can also be set up to transfer data to NDR.

[5] Set the initial output values in PODR.

[6] Set the DDR and NDER bits for the pins to be used for pulse output to 1.

[7] Select the TPU compare match event to be used as the output trigger in PCR.

[8] Set the next pulse output values in NDR.

[9] Set the CST bit in TSTR to 1 to start the TCNT counter.

[10] At each TGIA interrupt, set the next output values in NDR.

Figure 11.4 Setup Procedure for Normal Pulse Output (Example)

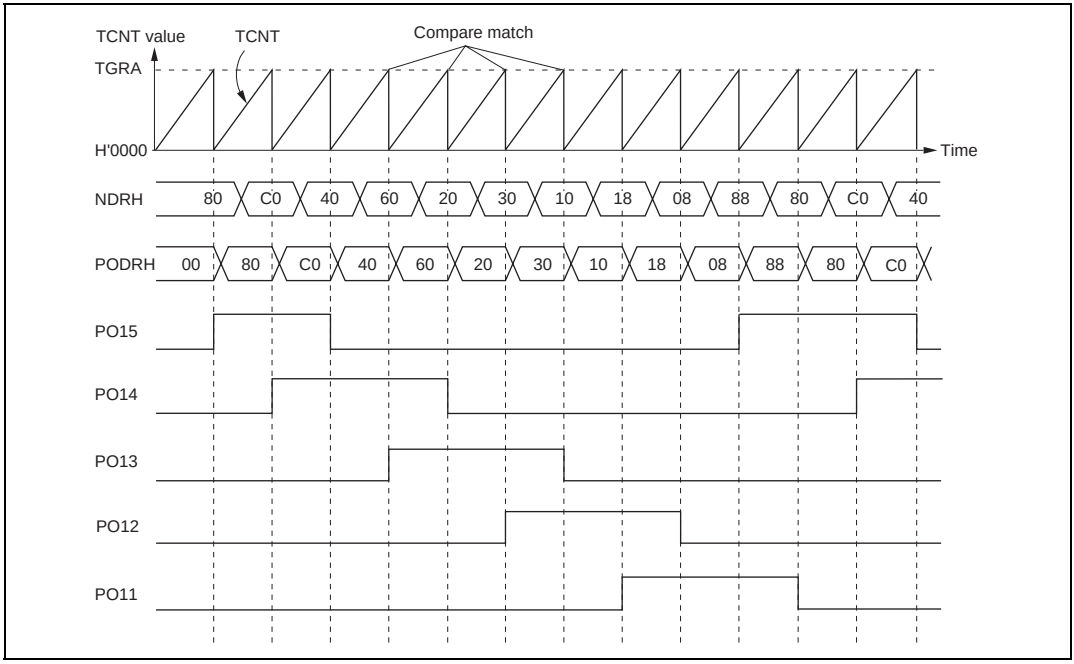


Figure 11.5 Normal Pulse Output Example (Five-Phase Pulse Output)

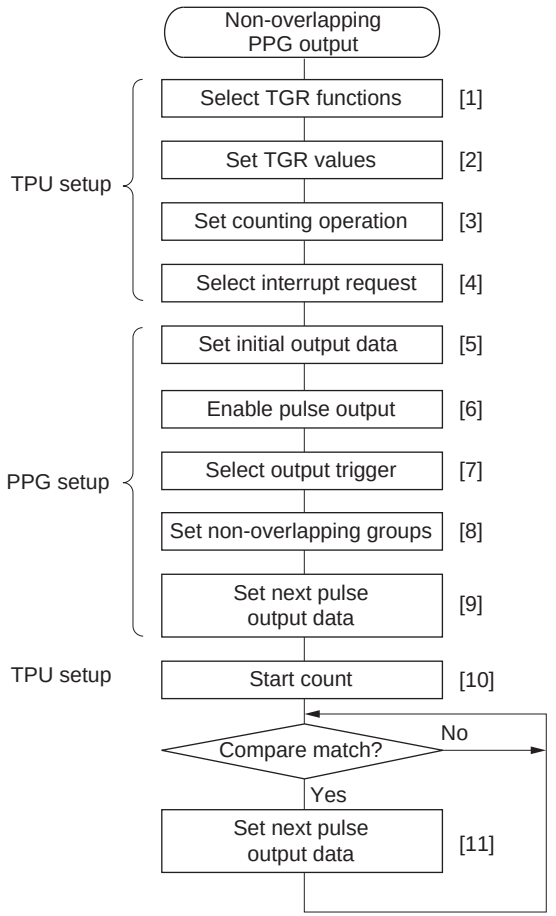
[1] Set up the TPU channel to be used as the output trigger channel so that TGRA is an output compare register and the counter will be cleared by compare match A. Set the trigger period in TGRA and set the TGIEA bit in TIER to 1 to enable the compare match A (TGIA) interrupt.

[2] Write H'F8 in P1DDR and NDERH, and set the G3CMS1, G3CMS0, G2CMS1, and G2CMS0 bits in PCR to select compare match in the TPU channel set up in the previous step to be the output trigger. Write output data H'80 in NDRH.

[3] The timer counter in the TPU channel starts. When compare match A occurs, the NDRH contents are transferred to PODRH and output. The TGIA interrupt handling routine writes the next output data (H'C0) in NDRH.

[4] Five-phase overlapping pulse output (one or two phases active at a time) can be obtained subsequently by writing H'40, H'60, H'20, H'30. H'10, H'18, H'08, H'88, ... at successive TGIA interrupts. If the DTC or DMAC is set for activation by this interrupt, pulse output can be obtained without imposing a load on the CPU.

Sample Setup Procedure for Non-Overlapping Pulse Output: Figure 11.6 shows a sample procedure for setting up non-overlapping pulse output.



[1] Set TIOR to make TGRA and TGRB an output compare registers (with output disabled).

[2] Set the pulse output trigger period in TGRB and the non-overlap margin in TGRA.

[3] Select the counter clock source with bits TPSC2 to TPSC0 in TCR. Select the counter clear source with bits CCLR1 and CCLR0.

[4] Enable the TGIA interrupt in TIER. The DTC or DMAC can also be set up to transfer data to NDR.

[5] Set the initial output values in PODR.

[6] Set the DDR and NDER bits for the pins to be used for pulse output to 1.

[7] Select the TPU compare match event to be used as the pulse output trigger in PCR.

[8] In PMR, select the groups that will operate in non-overlap mode.

[9] Set the next pulse output values in NDR.

[10] Set the CST bit in TSTR to 1 to start the TCNT counter.

[11] At each TGIA interrupt, set the next output values in NDR.

Figure 11.6 Setup Procedure for Non-Overlapping Pulse Output (Example)

phase complementary non-overlapping pulse output.

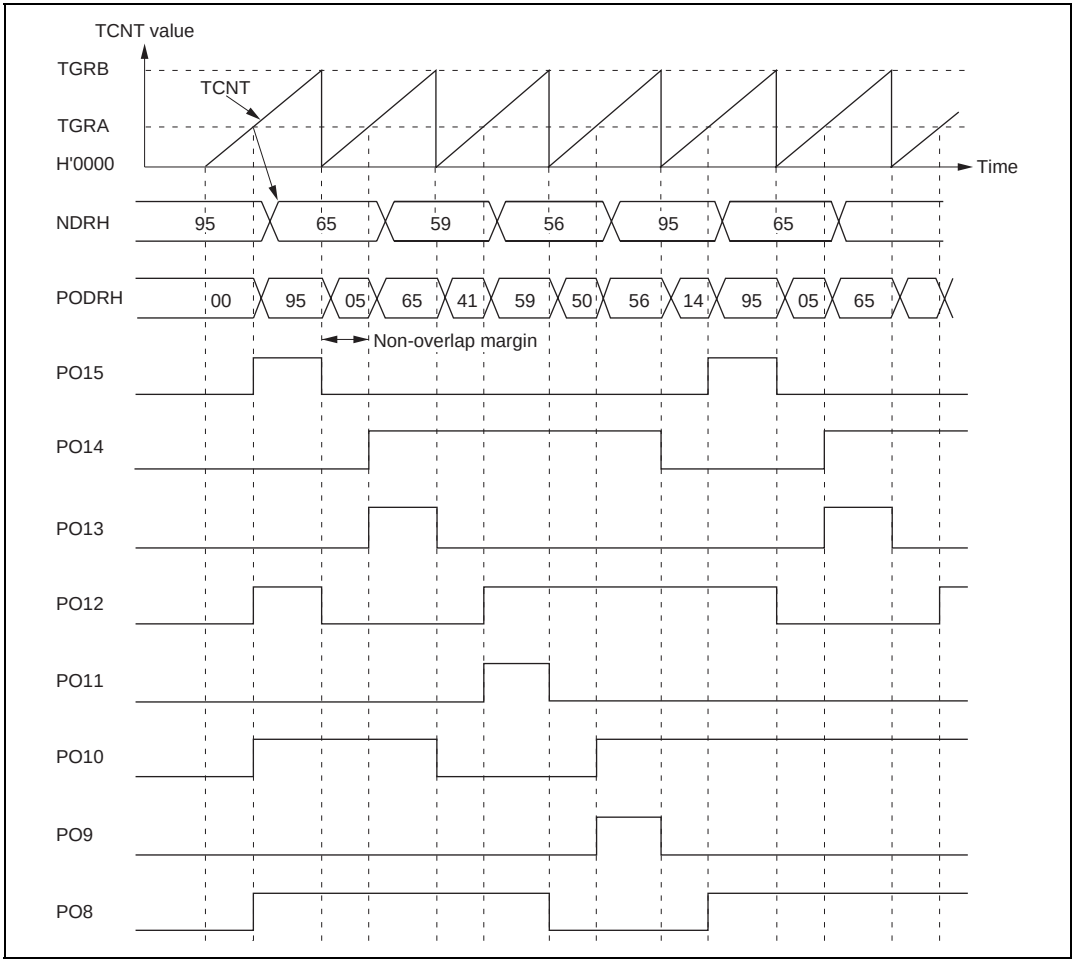


Figure 11.7 Non-Overlapping Pulse Output Example (Four-Phase Complementary)

TGRA, and set the counter to be cleared by compare match B. Set the TGIEA bit in TIER to 1 to enable the TGIA interrupt.

[2] Write H'FF in P1DDR and NDERH, and set the G3CMS1, G3CMS0, G2CMS1, and G2CMS0 bits in PCR to select compare match in the TPU channel set up in the previous step to be the output trigger. Set the G3NOV and G2NOV bits in PMR to 1 to select non-overlapping output. Write output data H'95 in NDRH.

[3] The timer counter in the TPU channel starts. When a compare match with TGRB occurs, outputs change from 1 to 0. When a compare match with TGRA occurs, outputs change from 0 to 1 (the change from 0 to 1 is delayed by the value set in TGRA). The TGIA interrupt handling routine writes the next output data (H'65) in NDRH.

[4] Four-phase complementary non-overlapping pulse output can be obtained subsequently by writing H'59, H'56, H'95, ... at successive TGIA interrupts. If the DTC or DMAC is set for activation by this interrupt, pulse output can be obtained without imposing a load on the CPU.

If the G3INV, G2INV, G1INV, and G0INV bits in PMR are cleared to 0, values that are the inverse of the PODR contents can be output.

Figure 11.8 shows the outputs when G3INV and G2INV are cleared to 0, in addition to the settings of figure 11.7.

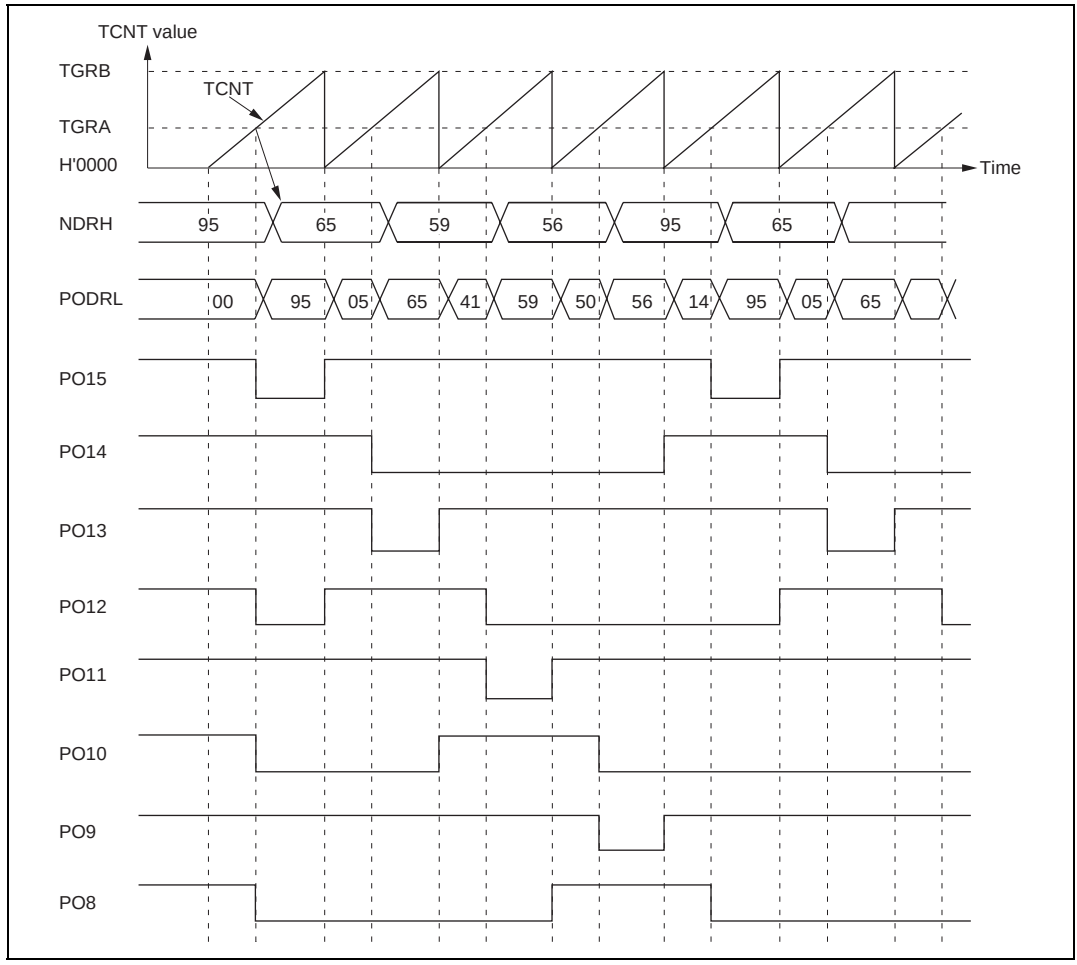


Figure 11.8 Inverted Pulse Output (Example)

Pulse output can be triggered by TPU input capture as well as by compare match. If TGRA functions as an input capture register in the TPU channel selected by PCR, pulse output will be triggered by the input capture signal.

Figure 11.9 shows the timing of this output.

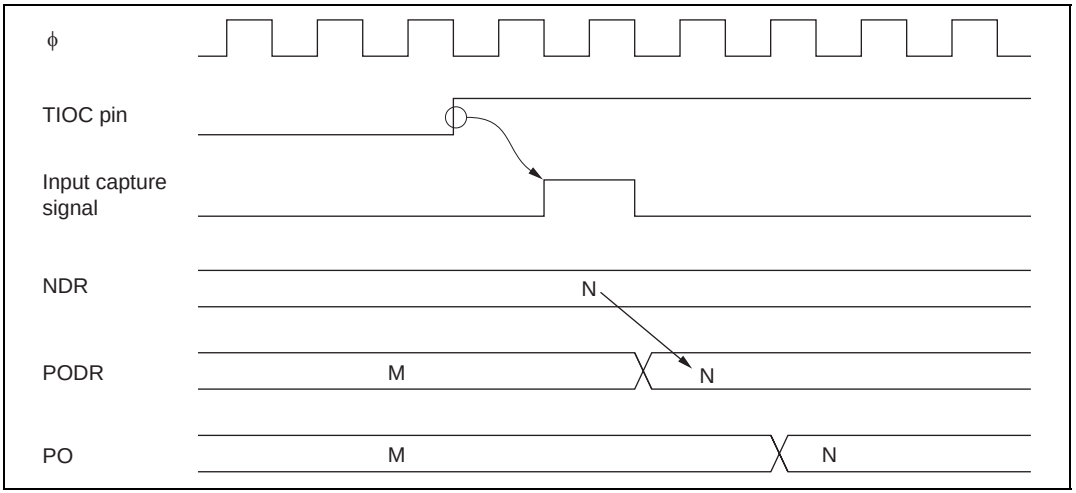


Figure 11.9 Pulse Output Triggered by Input Capture (Example)

11.4.1 Operation of Pulse Output Pins

Pins PO0 to PO15 are also used for other supporting functions such as the TPU. When output by another supporting function is enabled, the corresponding pins cannot be used for pulse output. Note, however, that data transfer from NDR bits to PODR bits takes place, regardless of the usage of the pins.

Pin functions should be changed only under conditions in which the output trigger event will not occur.

11.4.2 Note on Non-Overlapping Output

During non-overlapping operation, the transfer of NDR bit 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 transferred if their value is 1.

Figure 11.10 illustrates the non-overlapping pulse output operation.

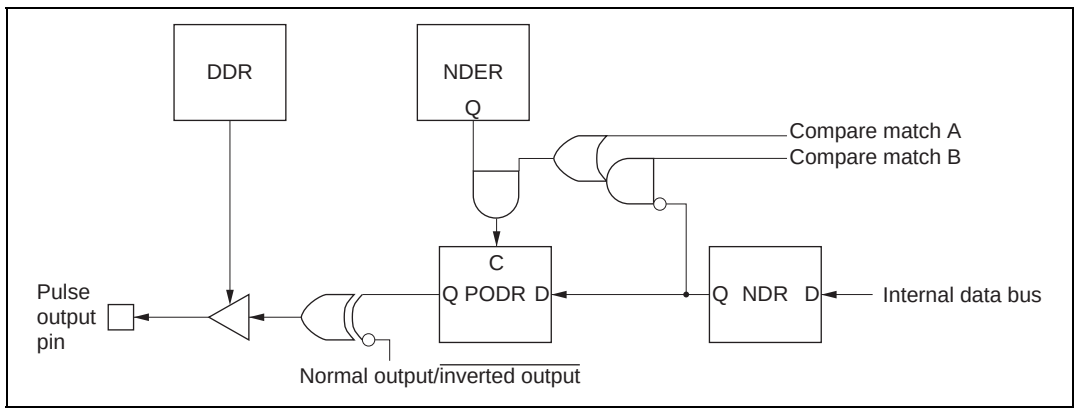


Figure 11.10 Non-Overlapping Pulse Output

match B to compare match A (the non-overlap margin).

This can be accomplished by having the TGIA interrupt handling routine write the next data in NDR, or by having the TGIA interrupt activate the DTC or DMAC. Note, however, that the next data must be written before the next compare match B occurs.

Figure 11.11 shows the timing of this operation.

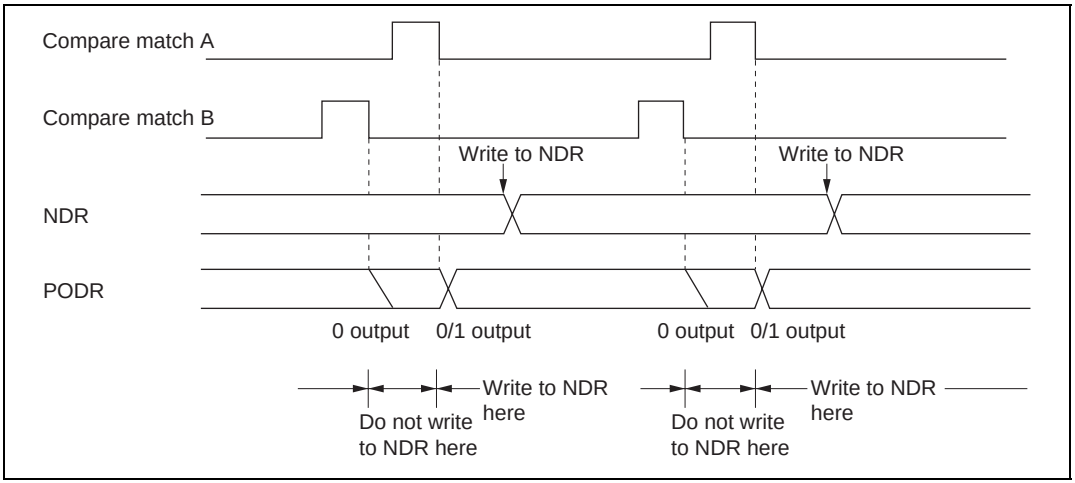


Figure 11.11 Non-Overlapping Operation and NDR Write Timing

12.1 Overview

The chip includes an 8-bit timer module with two channels (TMR0 and TMR1). Each channel has an 8-bit counter (TCNT) and two time constant registers (TCORA and TCORB) that are constantly compared with the TCNT value to detect compare match events. The 8-bit timer module can thus be used for a variety of functions, including pulse output with an arbitrary duty cycle.

12.1.1 Features

The features of the 8-bit timer module are listed below.

- Selection of four clock sources
  The counters can be driven by one of three internal clock signals (φ/8, φ/64, or φ/8192) or an external clock input (enabling use as an external event counter)
- Selection of three ways to clear the counters
  The counters can be cleared on compare match A or B, or by an external reset signal
- Timer output control by a combination of two compare match signals
  The timer output signal in each channel is controlled by a combination of two independent compare match signals, enabling the timer to generate output waveforms with an arbitrary duty cycle or PWM output
- Provision for cascading of two channels
  - Operation as a 16-bit timer is possible, using channel 0 for the upper 8 bits and channel 1 for the lower 8 bits (16-bit count mode)
  - Channel 1 can be used to count channel 0 compare matches (compare match count mode)
- Three independent interrupts
  Compare match A and B and overflow interrupts can be requested independently
- A/D converter conversion start trigger can be generated
  Channel 0 compare match A signal can be used as an A/D converter conversion start trigger
- Module stop mode can be set
  As the initial setting, 8-bit timer operation is halted. Register access is enabled by exiting module stop mode

Figure 12.1 shows a block diagram of the 8-bit timer module.

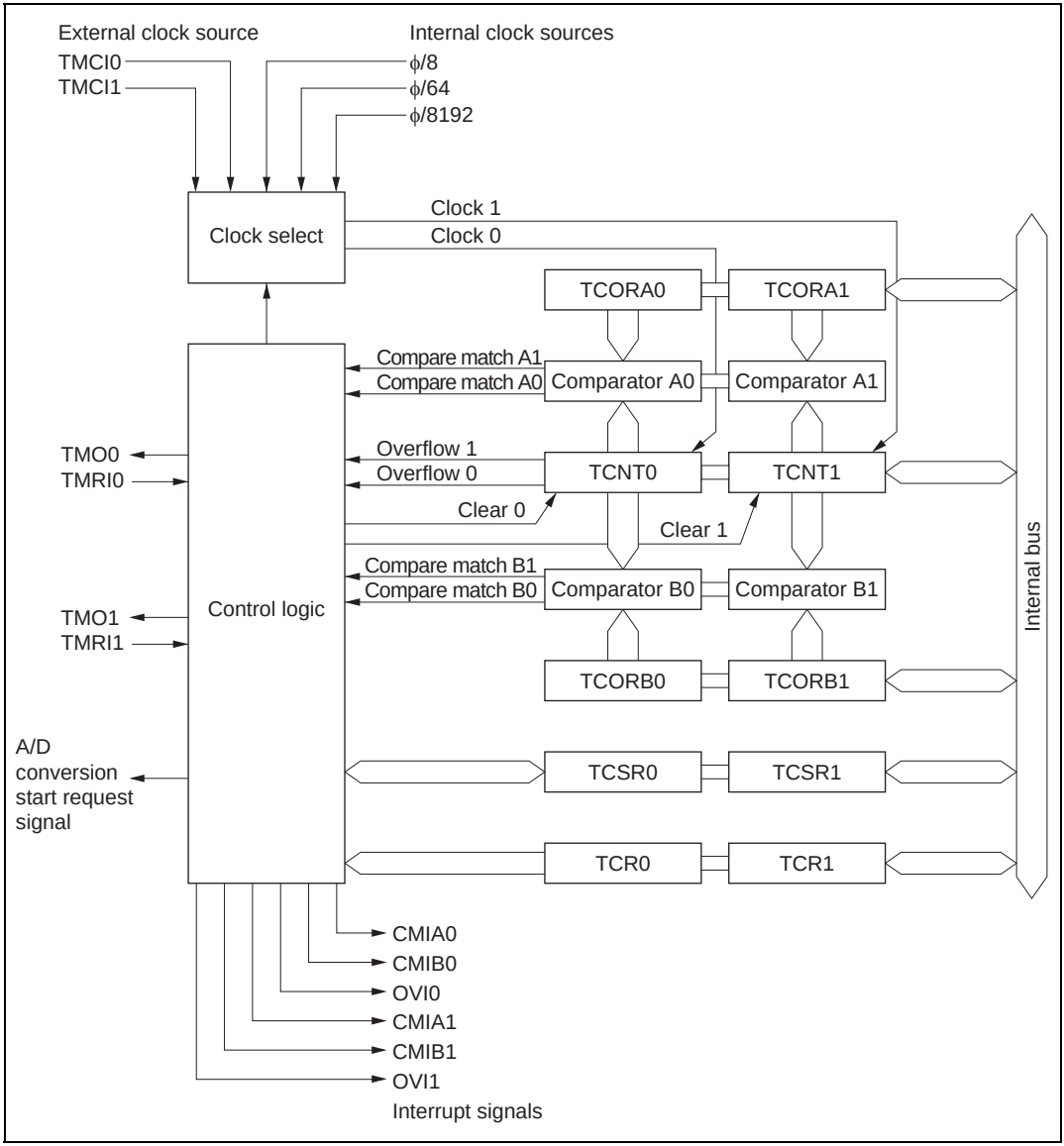


Figure 12.1 Block Diagram of 8-Bit Timer Module

Table 12.1 summarizes the input and output pins of the 8-bit timer module.

Table 12.1 Input and Output Pins of 8-Bit Timer

Channel	Name	Symbol	I/O	Function
0	Timer output pin 0	TMO0	Output	Outputs at compare match
	Timer clock input pin 0	TMCI0	Input	Inputs external clock for counter
	Timer reset input pin 0	TMRI0	Input	Inputs external reset to counter
1	Timer output pin 1	TMO1	Output	Outputs at compare match
	Timer clock input pin 1	TMCI1	Input	Inputs external clock for counter
	Timer reset input pin 1	TMRI1	Input	Inputs external reset to counter

12.1.4 Register Configuration

Table 12.2 summarizes the registers of the 8-bit timer module.

Table 12.2 8-Bit Timer Registers

Channel	Name	Abbreviation	R/W	Initial value	Address*1
0	Timer control register 0	TCR0	R/W	H'00	H'FFB0
	Timer control/status register 0	TCSR0	R/(W)*2	H'00	H'FFB2
	Time constant register A0	TCORA0	R/W	H'FF	H'FFB4
	Time constant register B0	TCORB0	R/W	H'FF	H'FFB6
	Timer counter 0	TCNT0	R/W	H'00	H'FFB8
1	Timer control register 1	TCR1	R/W	H'00	H'FFB1
	Timer control/status register 1	TCSR1	R/(W)*2	H'10	H'FFB3
	Time constant register A1	TCORA1	R/W	H'FF	H'FFB5
	Time constant register B1	TCORB1	R/W	H'FF	H'FFB7
	Timer counter 1	TCNT1	R/W	H'00	H'FFB9
All	Module stop control register	MSTPCR	R/W	H'3FFF	H'FF3C

Notes: 1. Lower 16 bits of the address
2. Only 0 can be written to bits 7 to 5, to clear these flags.

Each pair of registers for channel 0 and channel 1 is a 16-bit register with the upper 8 bits for channel 0 and the lower 8 bits for channel 1, so they can be accessed together by a word transfer instruction.

12.2.1 Timer Counters 0 and 1 (TCNT0, TCNT1)

	TCNT0								TCNT1							
Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value :	0	0	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/W	R/W

TCNT0 and TCNT1 are 8-bit readable/writable up-counters that increment on pulses generated from an internal or external clock source. This clock source is selected by clock select bits CKS2 to CKS0 in TCR. The CPU can read or write to TCNT0 and TCNT1 at all times.

TCNT0 and TCNT1 comprise a single 16-bit register, so they can be accessed together by a word transfer instruction.

TCNT0 and TCNT1 can be cleared by an external reset input or by a compare match signal. Which signal is to be used for clearing is selected by clock clear bits CCLR1 and CCLR0 in TCR.

When a timer counter overflows from H'FF to H'00, OVF in TCSR is set to 1.

TCNT0 and TCNT1 are each initialized to H'00 by a reset and in hardware standby mode.

12.2.2 Time Constant Registers A0 and A1 (TCORA0, TCORA1)

	TCORA0								TCORA1							
Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value :	1	1	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/W	R/W	R/W

TCORA0 and TCORA1 are 8-bit readable/writable registers. TCORA0 and TCORA1 comprise a single 16-bit register so they can be accessed together by a word transfer instruction.

TCORA is continually compared with the value in TCNT. When a match is detected, the corresponding CMFA flag in TCSR is set. Note, however, that comparison is disabled during the T_2 state of a TCOR write cycle.

TCORA0 and TCORA1 are each initialized to H'FF by a reset and in hardware standby mode.

12.2.3 Time Constant Registers B0 and B1 (TCORB0, TCORB1)

	TCORB0								TCORB1							
Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value :	1	1	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/W	R/W	R/W

TCORB0 and TCORB1 are 8-bit readable/writable registers. TCORB0 and TCORB1 comprise a single 16-bit register so they can be accessed together by a word transfer instruction.

TCORB is continually compared with the value in TCNT. When a match is detected, the corresponding CMFB flag in TCSR is set. Note, however, that comparison is disabled during the T_2 state of a TCOR write cycle.

The timer output can be freely controlled by these compare match signals and the settings of output select bits OS3 and OS2 in TCSR.

TCORB0 and TCORB1 are each initialized to H'FF by a reset and in hardware standby mode.

12.2.4 Time Control Registers 0 and 1 (TCR0, TCR1)

Bit :	7	6	5	4	3	2	1	0
	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0
Initial value :	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

TCR0 and TCR1 are 8-bit readable/writable registers that select the clock source and the time at which TCNT is cleared, and enable interrupts.

TCR0 and TCR1 are each initialized to H'00 by a reset and in hardware standby mode.

For details of this timing, see section 12.3, Operation.

Bit 7 CMIEB	Description	
0	CMFB interrupt requests (CMIB) are disabled	(Initial value)
1	CMFB interrupt requests (CMIB) are enabled	

Bit 6—Compare Match Interrupt Enable A (CMIEA): Selects whether CMFA interrupt requests (CMIA) are enabled or disabled when the CMFA flag in TCSR is set to 1.

Bit 6 CMIEA	Description	
0	CMFA interrupt requests (CMIA) are disabled	(Initial value)
1	CMFA interrupt requests (CMIA) are enabled	

Bit 5—Timer Overflow Interrupt Enable (OVIE): Selects whether OVF interrupt requests (OVI) are enabled or disabled when the OVF flag in TCSR is set to 1.

Bit 5 OVIE	Description	
0	OVF interrupt requests (OVI) are disabled	(Initial value)
1	OVF interrupt requests (OVI) are enabled	

Bits 4 and 3—Counter Clear 1 and 0 (CCLR1 and CCLR0): These bits select the method by which TCNT is cleared: by compare match A or B, or by an external reset input.

Bit 4 CCLR1	Bit 3 CCLR0	Description	
0	0	Clearing is disabled	(Initial value)
	1	Clear by compare match A	
1	0	Clear by compare match B	
	1	Clear by rising edge of external reset input	

Bits 2 to 0—Clock Select 2 to 0 (CKS2 to CKS0): These bits select whether the clock input to TCNT is an internal or external clock.

Three internal clocks can be selected, all divided from the system clock (ϕ): ϕ/8, ϕ/64, and ϕ/8192. The falling edge of the selected internal clock triggers the count.

Some functions differ between channel 0 and channel 1.

Bit 2 CKS2	Bit 1 CKS1	Bit 0 CKS0	Description	
0	0	0	Clock input disabled	(Initial value)
		1	Internal clock, counted at falling edge of ϕ/8	
	1	0	Internal clock, counted at falling edge of ϕ/64	
		1	Internal clock, counted at falling edge of ϕ/8192	
1	0	0	For channel 0: count at TCNT1 overflow signal*	
			For channel 1: count at TCNT0 compare match A*	
		1	External clock, counted at rising edge	
	1	0	External clock, counted at falling edge	
		1	External clock, counted at both rising and falling edges	

Note: * If the count input of channel 0 is the TCNT1 overflow signal and that of channel 1 is the TCNT0 compare match signal, no incrementing clock is generated. Do not use this setting.

12.2.5 Timer Control/Status Registers 0 and 1 (TCSR0, TCSR1)

TCSR0

Bit	:	7	6	5	4	3	2	1	0
		CMFB	CMFA	OVF	ADTE	OS3	OS2	OS1	OS0
Initial value	:	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

TCSR1

Bit	:	7	6	5	4	3	2	1	0
		CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0
Initial value	:	0	0	0	1	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 to bits 7 to 5, to clear these flags.

TCSR0 is initialized to H'00, and TCSR1 to H'10, by a reset and in hardware standby mode.

Bit 7—Compare Match Flag B (CMFB): Status flag indicating whether the values of TCNT and TCORB match.

Bit 7 CMFB	Description	
0	[Clearing conditions]	(Initial value)
	• Cleared by reading CMFB when CMFB = 1, then writing 0 to CMFB	
	• When DTC is activated by CMIB interrupt while DISEL bit of MRB in DTC is 0	
1	[Setting condition]	
	Set when TCNT matches TCORB	

Bit 6—Compare Match Flag A (CMFA): Status flag indicating whether the values of TCNT and TCORA match.

Bit 6 CMFA	Description	
0	[Clearing conditions]	(Initial value)
	• Cleared by reading CMFA when CMFA = 1, then writing 0 to CMFA	
	• When DTC is activated by CMIA interrupt while DISEL bit of MRB in DTC is 0	
1	[Setting condition]	
	Set when TCNT matches TCORA	

Bit 5—Timer Overflow Flag (OVF): Status flag indicating that TCNT has overflowed (changed from H'FF to H'00).

Bit 5 OVF	Description	
0	[Clearing condition]	(Initial value)
	Cleared by reading OVF when OVF = 1, then writing 0 to OVF	
1	[Setting condition]	
	Set when TCNT overflows from H'FF to H'00	

In TCSR1, this bit is reserved: it is always read as 1 and cannot be modified.

Bit 4 ADTE	Description	
0	A/D converter start requests by compare match A are disabled	(Initial value)
1	A/D converter start requests by compare match A are enabled	

Bits 3 to 0—Output Select 3 to 0 (OS3 to OS0): These bits specify how the timer output level is to be changed by a compare match of TCOR and TCNT.

Bits OS3 and OS2 select the effect of compare match B on the output level, bits OS1 and OS0 select the effect of compare match A on the output level, and both of them can be controlled independently.

Note, however, that priorities are set such that: toggle output > 1 output > 0 output. If compare matches occur simultaneously, the output changes according to the compare match with the higher priority.

Timer output is disabled when bits OS3 to OS0 are all 0.

After a reset, the timer output is 0 until the first compare match event occurs.

Bit 3 OS3	Bit 2 OS2	Description	
0	0	No change when compare match B occurs	(Initial value)
	1	0 is output when compare match B occurs	
1	0	1 is output when compare match B occurs	
	1	Output is inverted when compare match B occurs (toggle output)	

Bit 1 OS1	Bit 0 OS0	Description	
0	0	No change when compare match A occurs	(Initial value)
	1	0 is output when compare match A occurs	
1	0	1 is output when compare match A occurs	
	1	Output is inverted when compare match A occurs (toggle output)	

	MSTPCRH								MSTPCRL							
Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value :	0	0	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/W	R/W	R/W

MSTPCR is a 16-bit readable/writable register that performs module stop mode control.

When the MSTP12 bit in MSTPCR is set to 1, the 8-bit timer operation stops at the end of the bus cycle and a transition is made to module stop mode. Registers cannot be read or written to in module stop mode. For details, see section 21.5, Module Stop Mode.

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

Bit 12—Module Stop (MSTP12): Specifies the 8-bit timer module stop mode.

Bit 12 MSTP12	Description	
0	8-bit timer module stop mode cleared	
1	8-bit timer module stop mode set	(Initial value)

12.3.1 TCNT Incrementation Timing

TCNT is incremented by input clock pulses (either internal or external).

Internal Clock: Three different internal clock signals (ϕ/8, ϕ/64, or ϕ/8192) divided from the system clock (ϕ) can be selected, by setting bits CKS2 to CKS0 in TCR. Figure 12.2 shows the count timing.

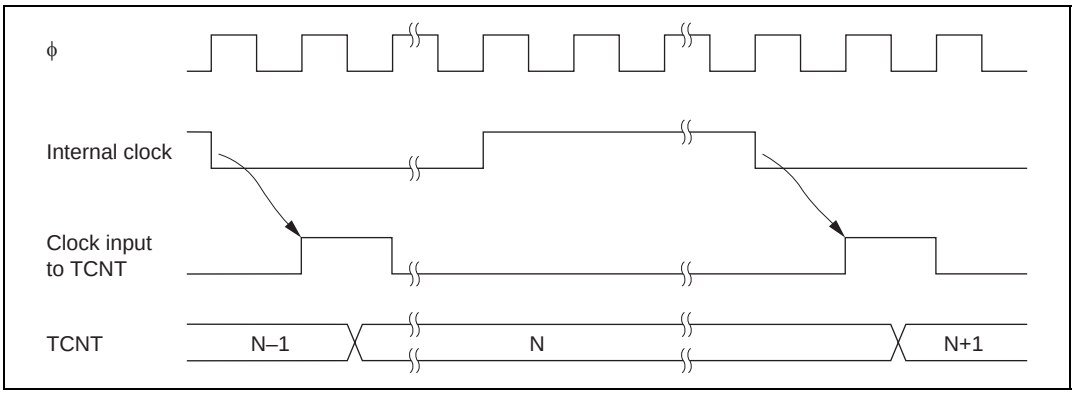


Figure 12.2 Count Timing for Internal Clock Input

External Clock: Three incrementation methods can be selected by setting bits CKS2 to CKS0 in TCR: at the rising edge, the falling edge, and both rising and falling edges.

Note that the external clock pulse width must be at least 1.5 states for incrementation at a single edge, and at least 2.5 states for incrementation at both edges. The counter will not increment correctly if the pulse width is less than these values.

Figure 12.3 shows the timing of incrementation at both edges of an external clock signal.

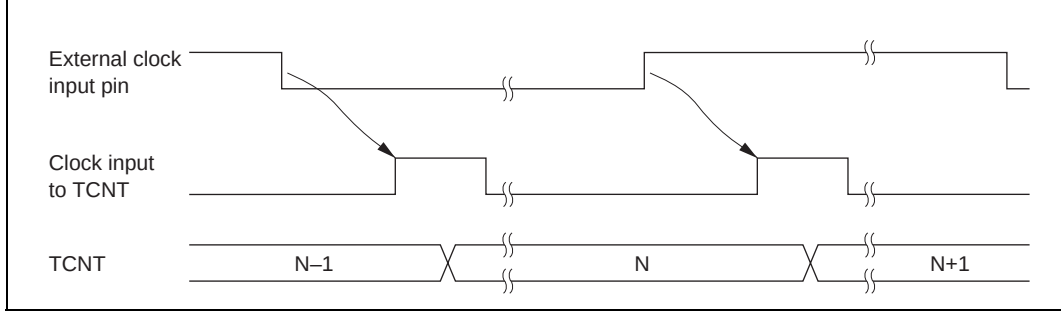


Figure 12.3 Count Timing for External Clock Input

12.3.2 Compare Match Timing

Setting of Compare Match Flags A and B (CMFA, CMFB): The CMFA and CMFB flags in TCSR are set to 1 by a compare match signal generated when the TCOR and TCNT values match. The compare match signal is generated at the last state in which the match is true, just before the timer counter is updated.

Therefore, when TCOR and TCNT match, the compare match signal is not generated until the next incrementation clock input. Figure 12.4 shows this timing.

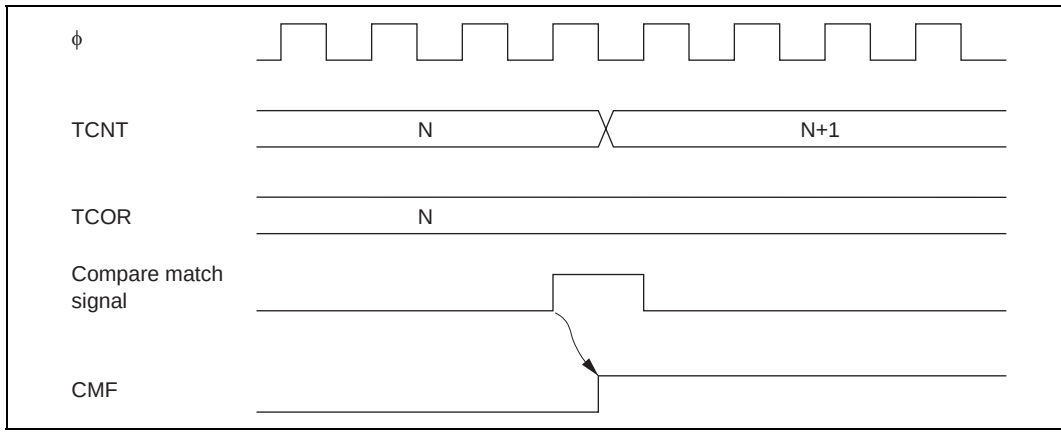


Figure 12.4 Timing of CMF Setting

change to 0, change to 1, or toggle.

Figure 12.5 shows the timing when the output is set to toggle at compare match A.

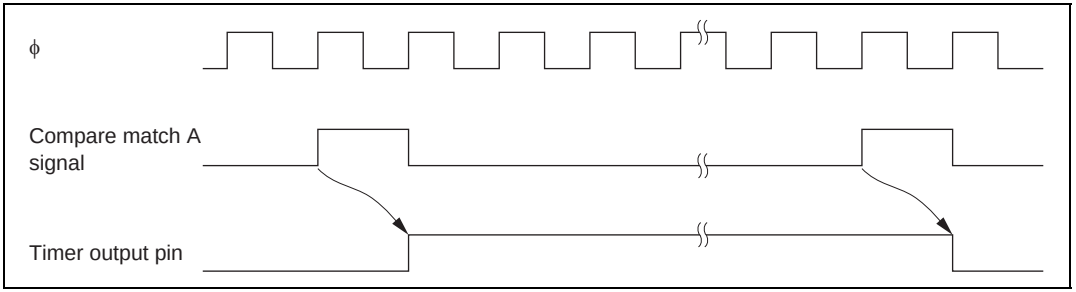


Figure 12.5 Timing of Timer Output

Timing of Compare Match Clear: The timer counter is cleared when compare match A or B occurs, depending on the setting of the CCLR1 and CCLR0 bits in TCR. Figure 12.6 shows the timing of this operation.

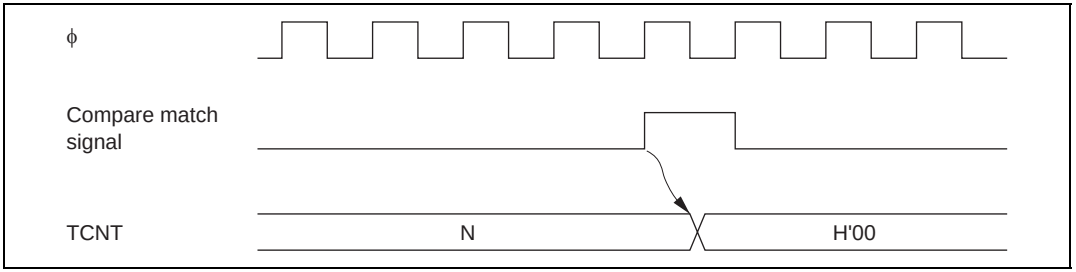


Figure 12.6 Timing of Compare Match Clear

TCNT is cleared at the rising edge of an external reset input, depending on the settings of the CCLR1 and CCLR0 bits in TCR. The clear pulse width must be at least 1.5 states. Figure 12.7 shows the timing of this operation.

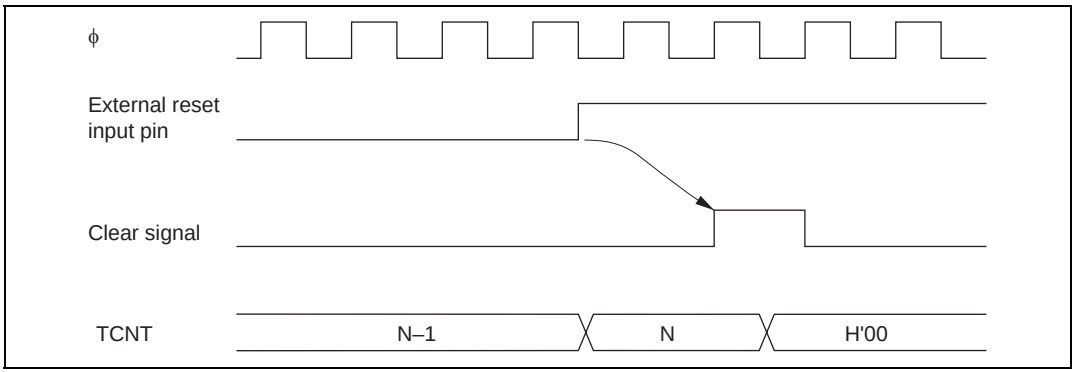


Figure 12.7 Timing of Clearance by External Reset

12.3.4 Timing of Overflow Flag (OVF) Setting

The OVF in TCSR is set to 1 when TCNT overflows (changes from H'FF to H'00). Figure 12.8 shows the timing of this operation.

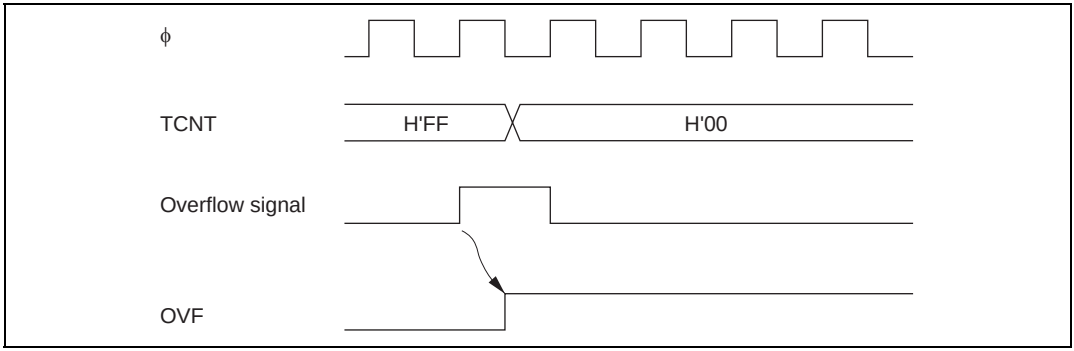


Figure 12.8 Timing of OVF Setting

If bits CKS2 to CKS0 in either TCR0 or TCR1 are set to B'100, the 8-bit timers of the two channels are cascaded. With this configuration, a single 16-bit timer could be used (16-bit counter mode) or compare matches of the 8-bit channel 0 could be counted by the timer of channel 1 (compare match counter mode). In this case, the timer operates as below.

16-Bit Counter Mode: When bits CKS2 to CKS0 in TCR0 are set to B'100, the timer functions as a single 16-bit timer with channel 0 occupying the upper 8 bits and channel 1 occupying the lower 8 bits.

- Setting of compare match flags
  - The CMF flag in TCSR0 is set to 1 when a 16-bit compare match event occurs.
  - The CMF flag in TCSR1 is set to 1 when a lower 8-bit compare match event occurs.
- Counter clear specification
  - If the CCLR1 and CCLR0 bits in TCR0 have been set for counter clear at compare match, the 16-bit counter (TCNT0 and TCNT1 together) is cleared when a 16-bit compare match event occurs. The 16-bit counter (TCNT0 and TCNT1 together) is cleared even if counter clear by the TMRI0 pin has also been set.
  - The settings of the CCLR1 and CCLR0 bits in TCR1 are ignored. The lower 8 bits cannot be cleared independently.
- Pin output
  - Control of output from the TMO0 pin by bits OS3 to OS0 in TCSR0 is in accordance with the 16-bit compare match conditions.
  - Control of output from the TMO1 pin by bits OS3 to OS0 in TCSR1 is in accordance with the lower 8-bit compare match conditions.

Compare Match Counter Mode: When bits CKS2 to CKS0 in TCR1 are B'100, TCNT1 counts compare match A's for channel 0.

Channels 0 and 1 are controlled independently. Conditions such as setting of the CMF flag, generation of interrupts, output from the TMO pin, and counter clear are in accordance with the settings for each channel.

Usage Note: If the 16-bit counter mode and compare match counter mode are set simultaneously, the input clock pulses for TCNT0 and TCNT1 are not generated and thus the counters will stop operating. Software should therefore avoid using both these modes.

12.4.1 Interrupt Sources and DTC Activation

There are three 8-bit timer interrupt sources: CMIA, CMIB, and OVI. Their relative priorities are shown in table 12.3. Each interrupt source is set as enabled or disabled by the corresponding interrupt enable bit in TCR, and independent interrupt requests are sent for each to the interrupt controller. It is also possible to activate the DTC by means of CMIA and CMIB interrupts.

Table 12.3 8-Bit Timer Interrupt Sources

Channel	Interrupt Source	Description	DTC Activation	Priority
0	CMIA0	Interrupt by CMFA	Possible	High
	CMIB0	Interrupt by CMFB	Possible	
	OVI0	Interrupt by OVF	Not possible	
1	CMIA1	Interrupt by CMFA	Possible	
	CMIB1	Interrupt by CMFB	Possible	
	OVI1	Interrupt by OVF	Not possible	Low

Note: This table shows the initial state immediately after a reset. The relative channel priorities can be changed by the interrupt controller.

12.4.2 A/D Converter Activation

The A/D converter can be activated only by channel 0 compare match A.

If the ADTE bit in TCSR0 is set to 1 when the CMFA flag is set to 1 by the occurrence of channel 0 compare match A, a request to start A/D conversion is sent to the A/D converter. If the 8-bit timer conversion start trigger has been selected on the A/D converter side at this time, A/D conversion is started.

In the example below, the 8-bit timer is used to generate a pulse output with a selected duty cycle, as shown in figure 12.9. The control bits are set as follows:

[1] In TCR, bit CCLR1 is cleared to 0 and bit CCLR0 is set to 1 so that the timer counter is cleared when its value matches the constant in TCORA.

[2] In TCSR, bits OS3 to OS0 are set to B'0110, causing the output to change to 1 at a TCORA compare match and to 0 at a TCORB compare match.

With these settings, the 8-bit timer provides output of pulses at a rate determined by TCORA with a pulse width determined by TCORB. No software intervention is required.

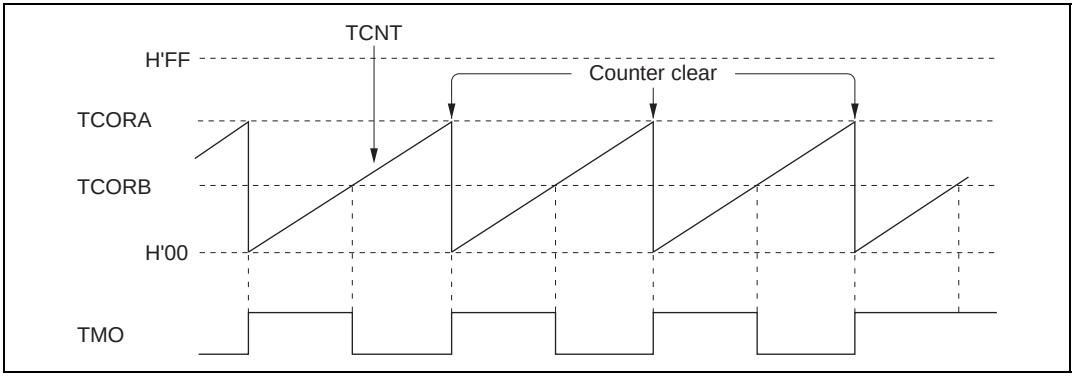


Figure 12.9 Example of Pulse Output

Note that the following kinds of contention can occur in the 8-bit timer module.

12.6.1 Contention between TCNT Write and Clear

If a timer counter clock pulse is generated during the T_2 state of a TCNT write cycle, the clear takes priority, so that the counter is cleared and the write is not performed.

Figure 12.10 shows this operation.

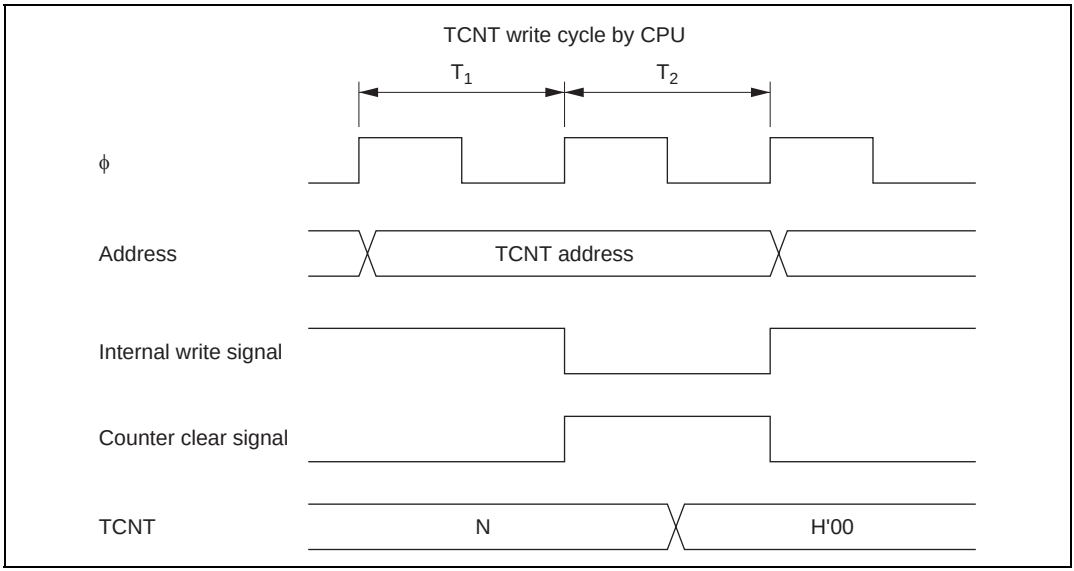


Figure 12.10 Contention between TCNT Write and Clear

If a timer counter clock pulse is generated during the T_2 state of a TCNT write cycle, the write takes priority and the counter is not incremented.

Figure 12.11 shows this operation.

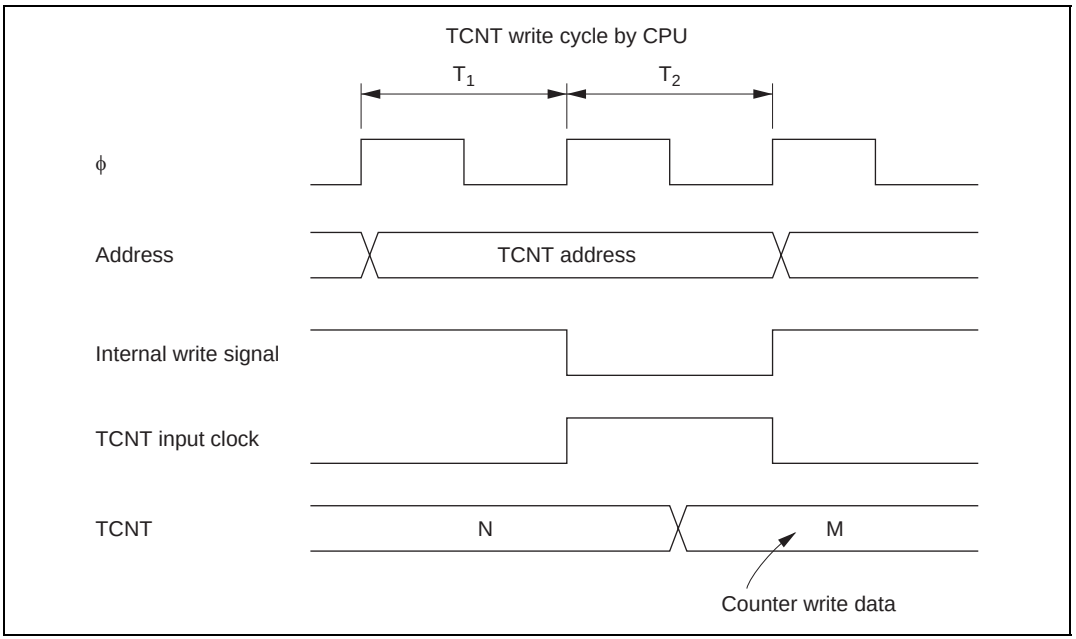


Figure 12.11 Contention between TCNT Write and Increment

During the T_2 state of a TCOR write cycle, the TCOR write has priority and the compare match signal is inhibited even if a compare match event occurs.

Figure 12.12 shows this operation.

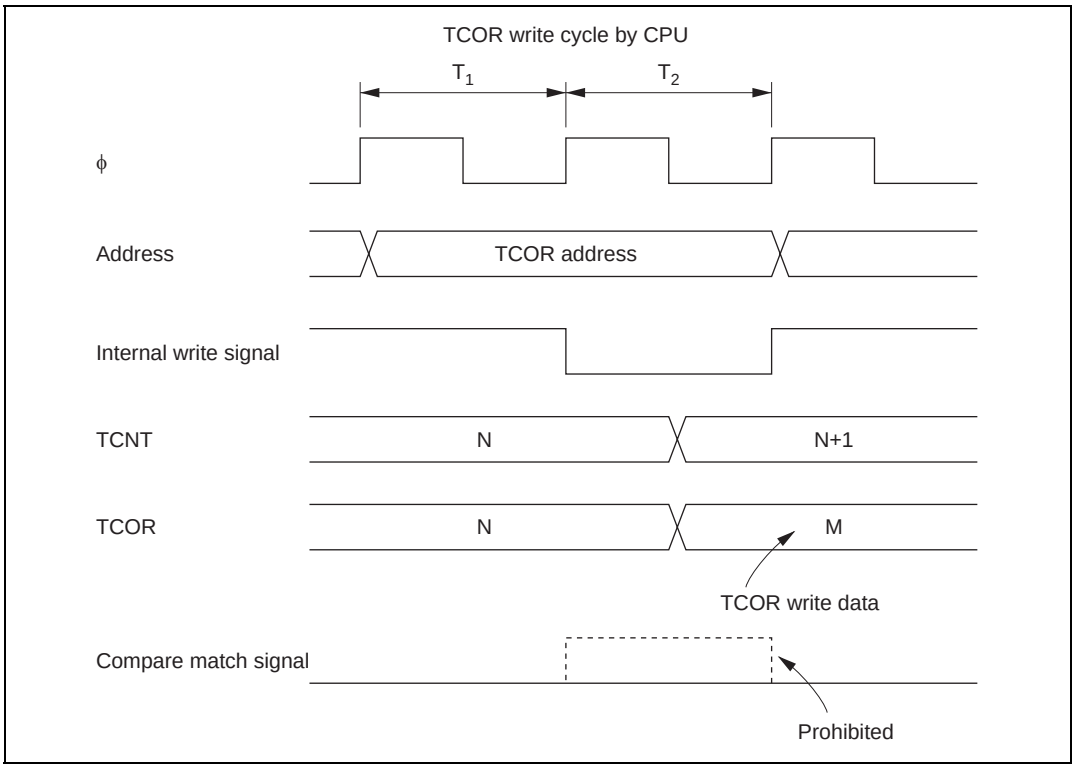


Figure 12.12 Contention between TCOR Write and Compare Match

If compare match events A and B occur at the same time, the 8-bit timer operates in accordance with the priorities for the output statuses set for compare match A and compare match B, as shown in table 12.4.

Table 12.4 Timer Output Priorities

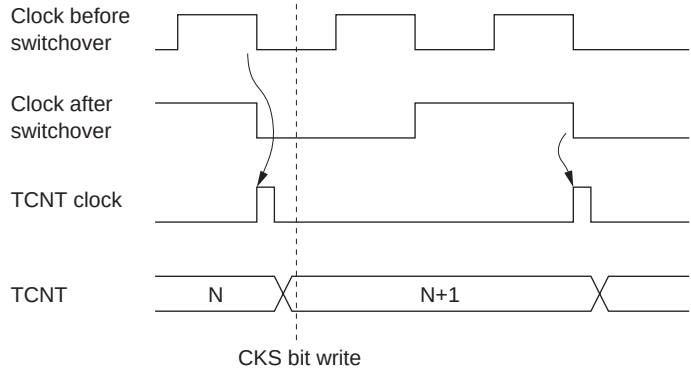
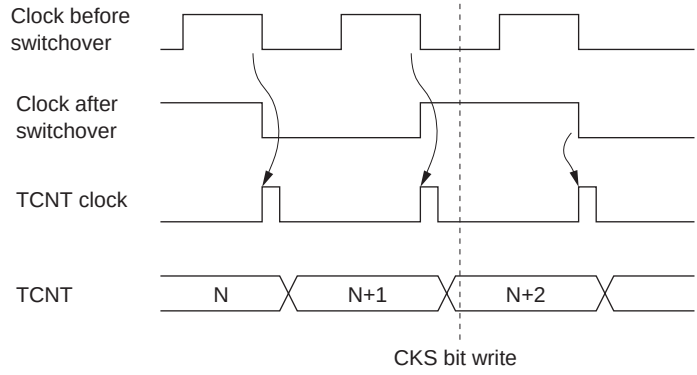
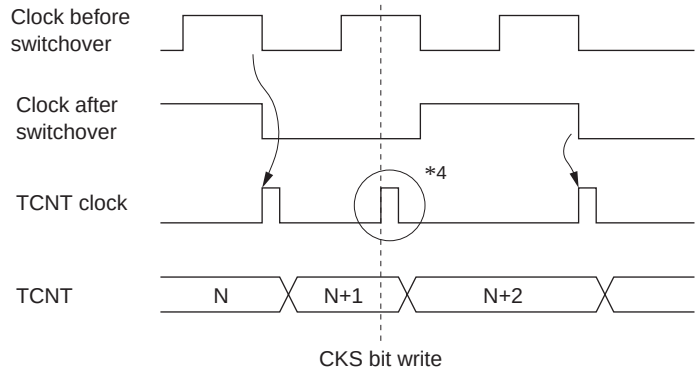
Output Setting	Priority
Toggle output	High
1 output	↑
0 output	↑
No change	Low

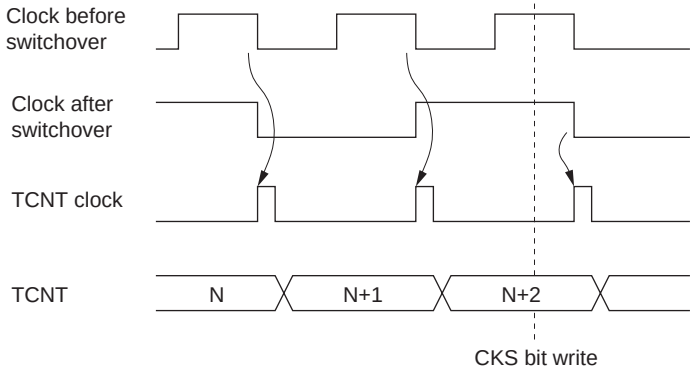
12.6.5 Switching of Internal Clocks and TCNT Operation

TCNT may increment erroneously when the internal clock is switched over. Table 12.5 shows the relationship between the timing at which the internal clock is switched (by writing to the CKS1 and CKS0 bits) and the TCNT operation.

When the TCNT clock is generated from an internal clock, the falling edge of the internal clock pulse is detected. If clock switching causes a change from high to low level, as shown in case 3 in table 12.5, a TCNT clock pulse is generated on the assumption that the switchover is a falling edge. This increments TCNT.

The erroneous incrementation can also happen when switching between internal and external clocks.

No.	Timing of Switchover by Means of CKS1 and CKS0 Bits	TCNT Clock Operation
1	Switching from low to low*1	
2	Switching from low to high*2	
3	Switching from high to low*3	

No.	and CKS0 Bits	TCNT Clock Operation
4	Switching from high to high	

Notes: 1. Includes switching from low to stop, and from stop to low.
2. Includes switching from stop to high.
3. Includes switching from high to stop.
4. Generated on the assumption that the switchover is a falling edge; TCNT is incremented.

12.6.6 Interrupts and Module Stop Mode

If module stop mode is entered when an interrupt has been requested, it will not be possible to clear the CPU interrupt source or the DMAC or DTC activation source. Interrupts should therefore be disabled before entering module stop mode.

13.1 Overview

The chip has a single-channel on-chip watchdog timer (WDT) for monitoring system operation. The WDT outputs an overflow signal ($\overline{\text{WDTOVF}}$) if a system crash prevents the CPU from writing to the timer counter, allowing it to overflow. At the same time, the WDT can also generate an internal reset signal for the chip.

When this watchdog function is not needed, the WDT can be used as an interval timer. In interval timer operation, an interval timer interrupt is generated each time the counter overflows.

13.1.1 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 whether or not the entire chip is reset at the same time
- Interrupt generation when in interval timer mode
  If the counter overflows, the WDT generates an interval timer interrupt
- Choice of eight counter clock sources

Figure 13.1 shows a block diagram of the WDT.

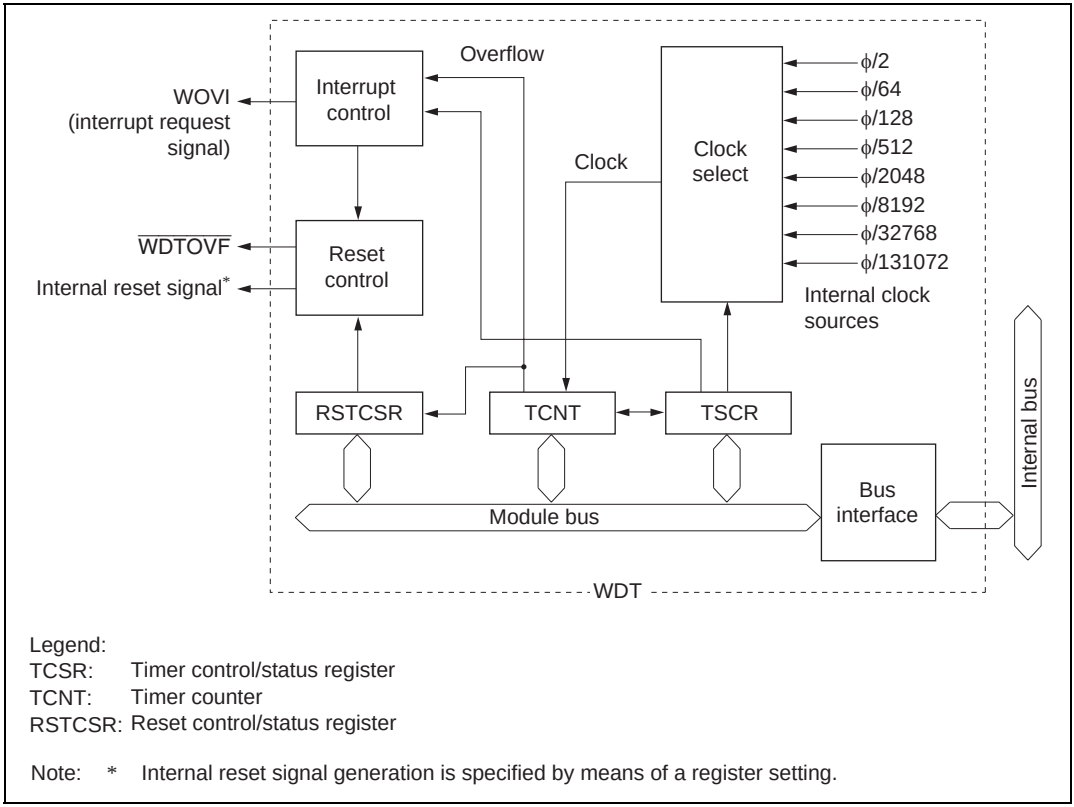


Figure 13.1 Block Diagram of WDT

Table 13.1 describes the WDT output pin.

Table 13.1 WDT Pin

Name	Symbol	I/O	Function
Watchdog timer overflow	$\overline{\text{WDTOVF}}$	Output	Outputs counter overflow signal in watchdog timer mode

13.1.4 Register Configuration

The WDT has three registers, as summarized in table 13.2. These registers control clock selection, WDT mode switching, and the reset signal.

Table 13.2 WDT Registers

				Address*1	
Name	**Abbreviation**	**R/W**	**Initial Value**	**Write*2**	**Read**
Timer control/status register	TCSR	R/(W)*3	H'18	H'FFBC	H'FFBC
Timer counter	TCNT	R/W	H'00	H'FFBC	H'FFBD
Reset control/status register	RSTCSR	R/(W)*3	H'1F	H'FFBE	H'FFBF

Notes: 1. Lower 16 bits of the address.
2. For details of write operations, see section 13.2.4, Notes on Register Access.
3. Only a write of 0 is permitted to bit 7, to clear the flag.

13.2.1 Timer Counter (TCNT)

Bit	:	7	6	5	4	3	2	1	0
Initial value	:	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

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 the internal clock source selected by bits CKS2 to CKS0 in TCSR. When the count overflows (changes from H'FF to H'00), either the watchdog timer overflow signal ($\overline{\text{WDTOVF}}$) or an interval timer interrupt (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 bit is cleared to 0. It is not initialized in software standby mode.

Note: * The $\overline{\text{WDTOVF}}$ pin function cannot be used in the F-ZTAT version.

Bit	:	7	6	5	4	3	2	1	0
		OVF	WT/$\overline{IT}$	TME	—	—	CKS2	CKS1	CKS0
Initial value	:	0	0	0	1	1	0	0	0
R/W	:	R/(W)*	R/W	R/W	—	—	R/W	R/W	R/W

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

TCSR is an 8-bit readable/writable* register. Its functions include selecting the clock source to be input to TCNT, and the timer mode.

TCR is initialized to H'18 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Note: * TCSR is write-protected by a password to prevent accidental overwriting. For details see section 13.2.4, Notes on Register Access.

Bit 7—Overflow Flag (OVF): Indicates that TCNT has overflowed from H'FF to H'00, when in interval timer mode. This flag cannot be set during watchdog timer operation.

Bit 7 OVF	Description	
0	[Clearing condition] Cleared by reading TCSR when OVF = 1, then writing 0 to OVF	(Initial value)
1	[Setting condition] Set when TCNT overflows (changes from H'FF to H'00) in interval timer mode	

Bit 6—Timer Mode Select (WT/$\overline{IT}$): Selects whether the WDT is used as a watchdog timer or interval timer. If used as an interval timer, the WDT generates an interval timer interrupt request (WOVI) when TCNT overflows. If used as a watchdog timer, the WDT generates the $\overline{WDTOVF}$ signal when TCNT overflows.

Bit 6 WT/$\overline{IT}$	Description	
0	Interval timer: Sends the CPU an interval timer interrupt request (WOVI) when TCNT overflows	(Initial value)
1	Watchdog timer: Generates the $\overline{WDTOVF}$ signal when TCNT overflows*	

Note: * For details of the case where TCNT overflows in watchdog timer mode, see section 13.2.3, Reset Control/Status Register (RSTCSR).

Bit 5 TME	Description	
0	TCNT is initialized to H'00 and halted	(Initial value)
1	TCNT counts	

Bits 4 and 3—Reserved: These bits cannot be modified and are always read as 1.

Bits 2 to 0—Clock Select 2 to 0 (CKS2 to CKS0): These bits select one of eight internal clock sources, obtained by dividing the system clock (ϕ), for input to TCNT.

			Description	
Bit 2 CKS2	**Bit 1 CKS1**	**Bit 0 CKS0**	**Clock**	**Overflow Period (when ϕ = 20 MHz)***
0	0	0	ϕ/2 (Initial value)	25.6 µs
		1	ϕ/64	819.2 µs
	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: * The overflow period is the time from when TCNT starts counting up from H'00 until overflow occurs.

Bit	:	7	6	5	4	3	2	1	0
		WOVF	RSTE	—	—	—	—	—	—
Initial value	:	0	0	0	1	1	1	1	1
R/W	:	R/(W)*	R/W	R/W	—	—	—	—	—

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

RSTCSR is an 8-bit readable/writable* register that controls the generation of the internal reset signal when TCNT overflows, and selects the type of internal reset signal.

RSTCSR is initialized to H'1F by a reset signal from the $\overline{\text{RES}}$ pin, but not by the WDT internal reset signal caused by overflows.

Note: * RSTCSR is write-protected by a password to prevent accidental overwriting. For details see section 13.2.4, Notes on Register Access.

Bit 7—Watchdog Timer Overflow Flag (WOVF): Indicates that TCNT has overflowed (changed from H'FF to H'00) during watchdog timer operation. This bit is not set in interval timer mode.

Bit 7 WOVF	Description	
0	[Clearing condition] Cleared by reading RSTCSR when WOVF = 1, then writing 0 to WOVF	(Initial value)
1	[Setting condition] Set when TCNT overflows (changes from H'FF to H'00) during watchdog timer operation	

Bit 6—Reset Enable (RSTE): Specifies whether or not a reset signal is generated in the chip if TCNT overflows during watchdog timer operation.

Bit 6 RSTE	Description	
0	Reset signal is not generated if TCNT overflows*	(Initial value)
1	Reset signal is generated if TCNT overflows	

Note: * The modules within the chip are not reset, but TCNT and TCSR within the WDT are reset.

Bits 4 to 0—Reserved: These bits cannot be modified and are always read as 1.

13.2.4 Notes on Register Access

The watchdog timer's TCNT, TCSR, and RSTCSR registers differ from other registers in being more difficult to write to. The procedures for writing to and reading these registers are given below.

Writing to TCNT and TCSR: These registers must be written to by a word transfer instruction. They cannot be written to with byte instructions.

Figure 13.2 shows the format of data written to TCNT and TCSR. TCNT and TCSR both have the same write address. For a write to TCNT, the upper byte of the written word must contain H'5A and the lower byte must contain the write data. For a write to TCSR, the upper byte of the written word must contain H'A5 and the lower byte must contain the write data. This transfers the write data from the lower byte to TCNT or TCSR.

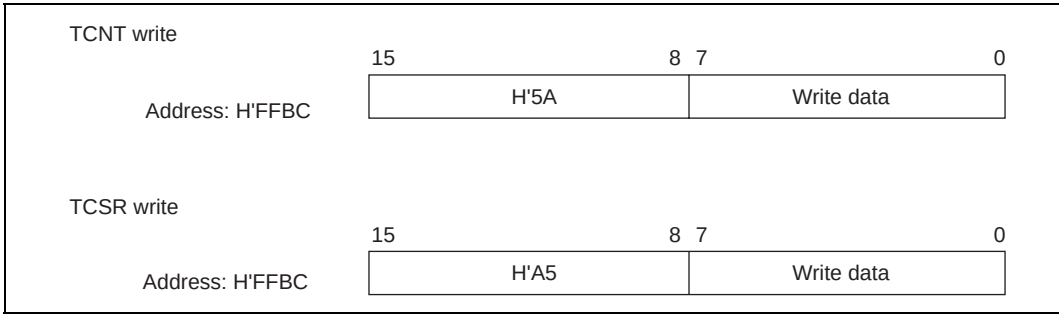


Figure 13.2 Writing to TCNT and TCSR

Writing to RSTCSR: RSTCSR must be written to by a word transfer instruction to address H'FFBE. It cannot be written to with byte instructions.

Figure 13.3 shows the format of data written to RSTCSR. The method of writing 0 to the WOVF bit differs from that for writing to the RSTE bit.

To write 0 to the WOVF bit, the write data must have H'A5 in the upper byte and H'00 in the lower byte. This clears the WOVF bit to 0, but has no effect on the RSTE bit. To write to the RSTE bit, the upper byte must contain H'5A and the lower byte must contain the write data. This writes the value in bit 6 of the lower byte into the RSTE bit, but has no effect on the WOVF bit.

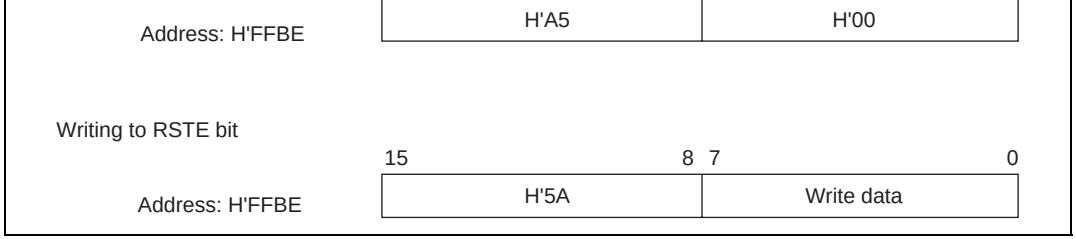


Figure 13.3 Writing to RSTCSR

Reading TCNT, TCSR, and RSTCSR: These registers are read in the same way as other registers. The read addresses are H'FFBC for TCSR, H'FFBD for TCNT, and H'FFBF for RSTCSR.

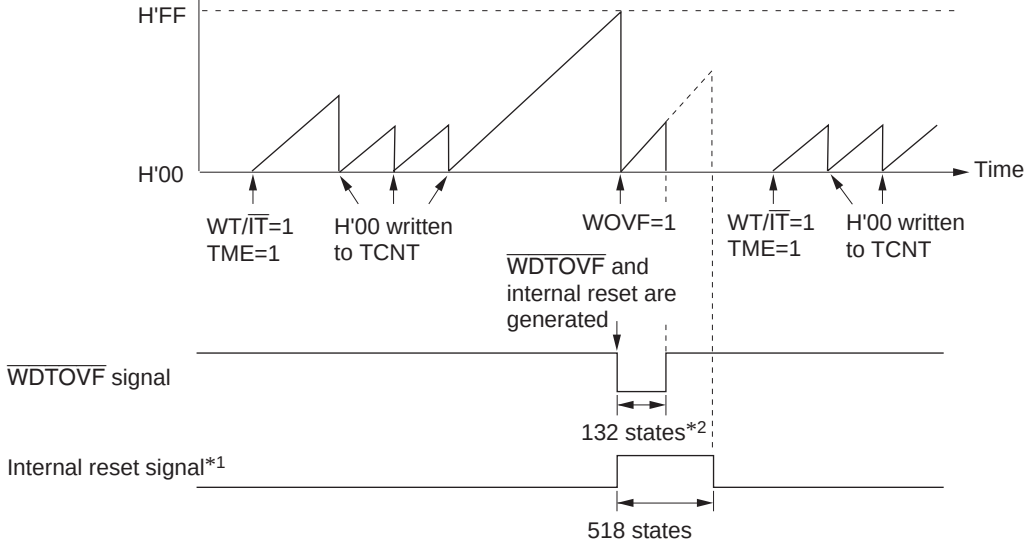
13.3 Operation

13.3.1 Operation in Watchdog Timer Mode

To use the WDT as a watchdog timer, set the WT/$\overline{IT}$ and TME bits to 1. Software must prevent TCNT overflows by rewriting the TCNT value (normally be writing H'00) before overflow occurs. This ensures that TCNT does not overflow while the system is operating normally. If TCNT overflows without being rewritten because of a system crash or other error, the $\overline{WDTOVF}$ signal is output. This is shown in figure 13.4. This $\overline{WDTOVF}$ signal can be used to reset the system. The $\overline{WDTOVF}$ signal is output for 132 states when RSTE = 1, and for 130 states when RSTE = 0.

If TCNT overflows when 1 is set in the RSTE bit in RSTCSR, a signal that resets the chip internally is generated at the same time as the $\overline{WDTOVF}$ signal. The internal reset signal is output for 518 states.

If a reset caused by a signal input to the $\overline{RES}$ pin occurs at the same time as a reset caused by a WDT overflow, the $\overline{RES}$ pin reset has priority and the WOVF bit in RSTCSR is cleared to 0.



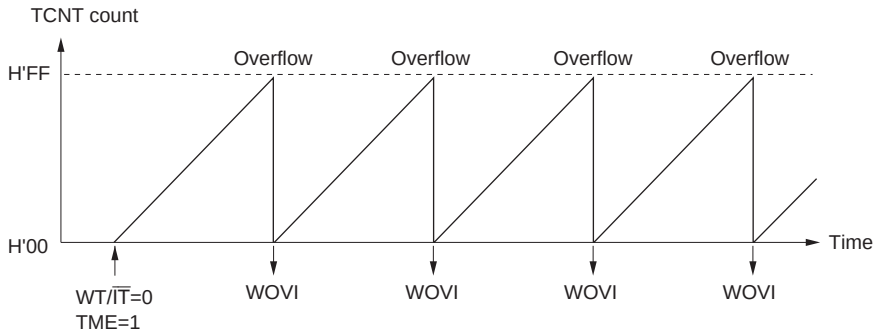
Legend:

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 13.4 Operation in Watchdog Timer Mode

To use the WDT as an interval timer, clear the WT/IT bit in TCSR to 0 and set the TME bit to 1. An interval timer interrupt (WOVI) is generated each time TCNT overflows, provided that the WDT is operating as an interval timer, as shown in figure 13.5. This function can be used to generate interrupt requests at regular intervals.



Legend:
WOVI: Interval timer interrupt request generation

Figure 13.5 Operation in Interval Timer Mode

The OVF flag is set to 1 if TCNT overflows during interval timer operation. At the same time, an interval timer interrupt (WOVI) is requested. This timing is shown in figure 13.6.

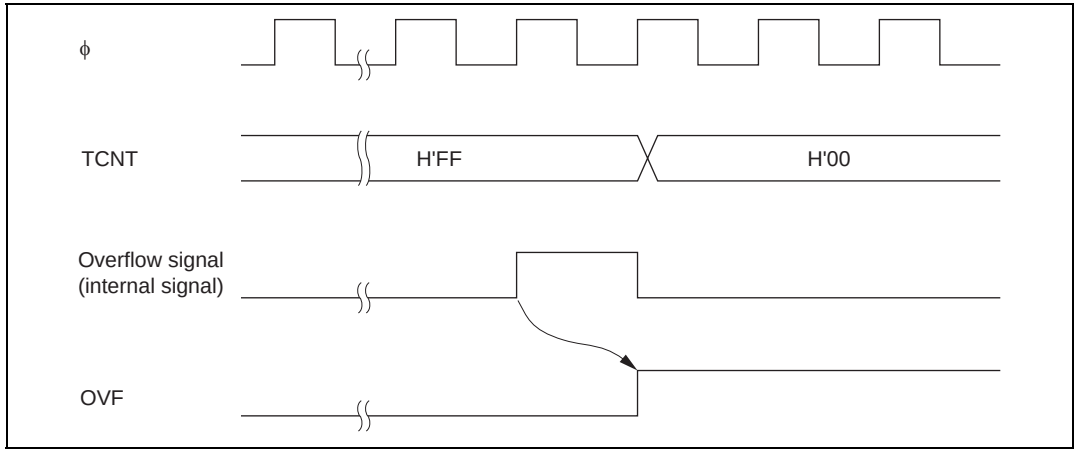


Figure 13.6 Timing of OVF Setting

The WOVF flag is set to 1 if TCNT overflows during watchdog timer operation. At the same time, the $\overline{\text{WDTOVF}}$ signal goes low. If TCNT overflows while the RSTE bit in RSTCSR is set to 1, an internal reset signal is generated for the entire chip. Figure 13.7 shows the timing in this case.

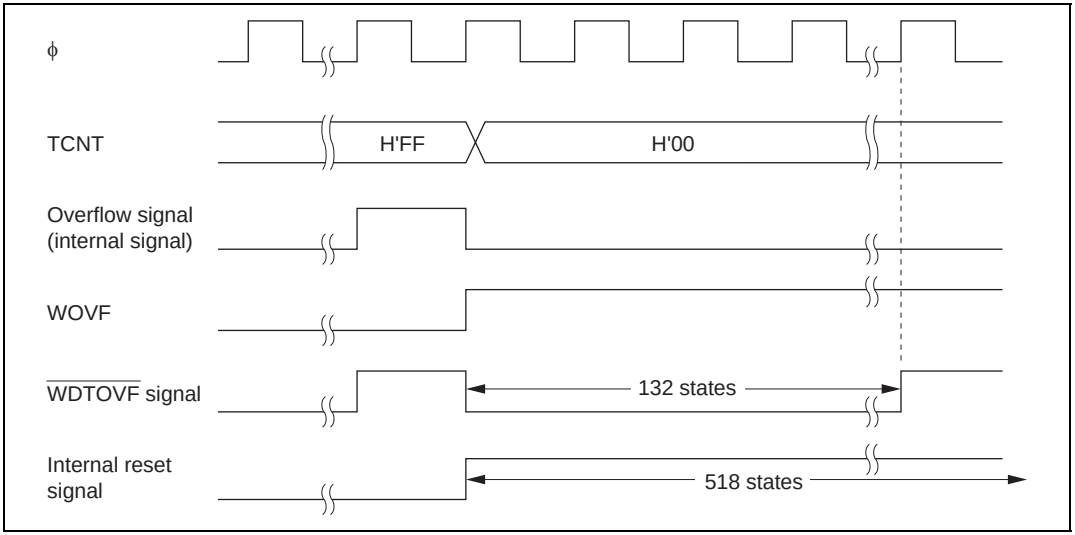


Figure 13.7 Timing of WOVF Setting

During interval timer mode operation, an overflow generates an interval timer interrupt (WOVI). The interval timer interrupt is requested whenever the OVF flag is set to 1 in TCSR.

13.5 Usage Notes

13.5.1 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, the write takes priority and the timer counter is not incremented. Figure 13.8 shows this operation.

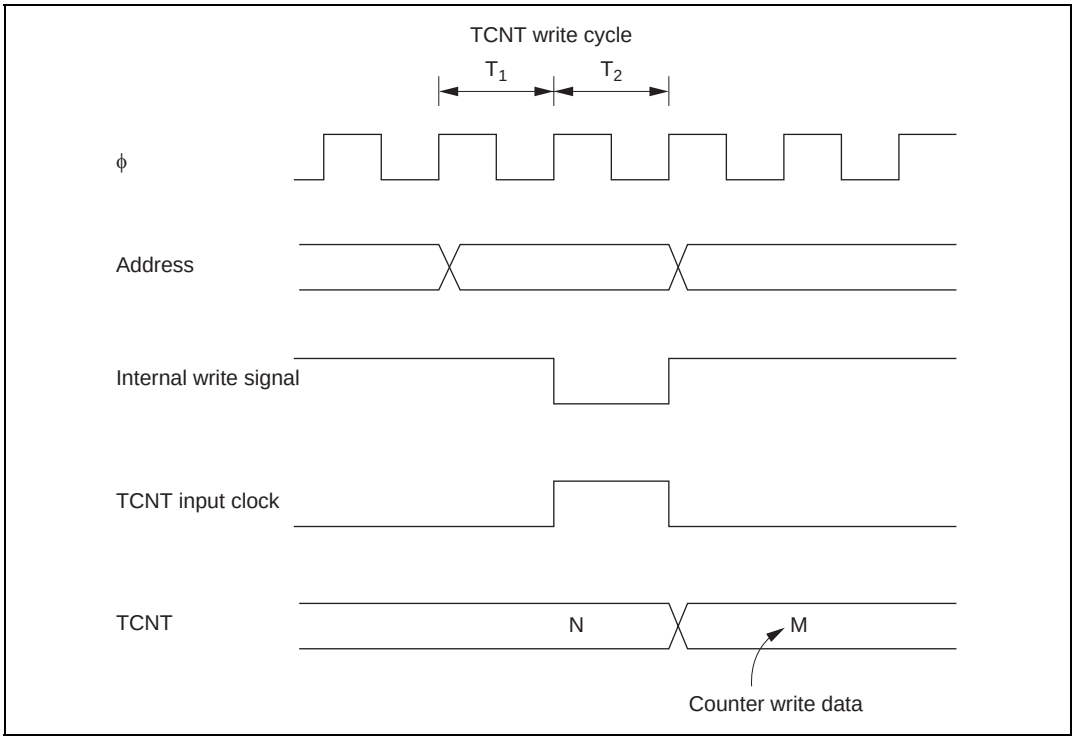


Figure 13.8 Contention between TCNT Write and Increment

If bits CKS2 to CKS0 in TCSR are written to while the WDT is operating, errors may occur in the incrementation. Software must stop the watchdog timer (by clearing the TME bit to 0) before changing the value of bits CKS2 to CKS0.

13.5.3 Switching between Watchdog Timer Mode and Interval Timer Mode

If the mode is switched from watchdog timer to interval timer, or vice versa, while the WDT is operating, errors may occur in the incrementation. Software must stop the watchdog timer (by clearing the TME bit to 0) before switching the mode.

13.5.4 System Reset by $\overline{\text{WDTOVF}}$ Signal

If the $\overline{\text{WDTOVF}}$ output signal is input to the $\overline{\text{RES}}$ pin of the chip, the chip will not be initialized correctly. Make sure that the $\overline{\text{WDTOVF}}$ signal is not input logically to the $\overline{\text{RES}}$ pin. To reset the entire system by means of the $\overline{\text{WDTOVF}}$ signal, use the circuit shown in figure 13.9.

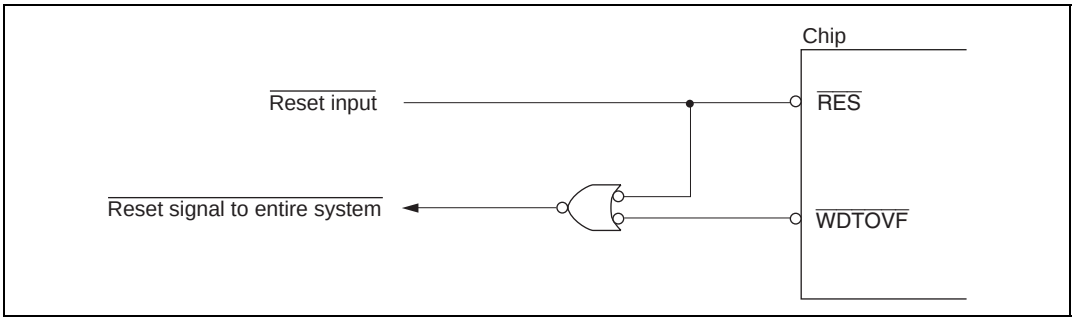


Figure 13.9 Circuit for System Reset by $\overline{\text{WDTOVF}}$ Signal (Example)

The chip is not reset internally if TCNT overflows while the RSTE bit is cleared to 0 during watchdog timer operation, but TCNT and TCSR of the WDT are reset.

TCNT, TCSR, and RSTCSR cannot be written to while the $\overline{\text{WDTOVF}}$ signal is low. Also note that a read of the WOVF flag is not recognized during this period. To clear the WOVF flag, therefore, read RSTCSR after the $\overline{\text{WDTOVF}}$ signal goes high, then write 0 to the WOVF flag.

14.1 Overview

The chip is equipped with a serial communication interface (SCI) that can handle both asynchronous and synchronous serial communication. A function is also provided for serial communication between processors (multiprocessor communication function).

14.1.1 Features

SCI features are listed below.

- Choice of asynchronous or synchronous serial communication mode

  Asynchronous mode

  — Serial data communication executed using an asynchronous system in which synchronization is achieved character by character

  Serial data communication can be carried out with standard asynchronous communication chips such as a Universal Asynchronous Receiver/Transmitter (UART) or Asynchronous Communication Interface Adapter (ACIA)

  — A 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 directly in case of a framing error

  Synchronous mode

  — Serial data communication synchronized with a clock

  — Serial data communication can be carried out with other chips that have a synchronous communication function

  — One serial data transfer format

  Data length: 8 bits

  — Receive error detection: Overrun errors detected

to be executed simultaneously

— Double-buffering is used in both the transmitter and the receiver, enabling continuous transmission and continuous reception of serial data

- Choice of LSB-first or MSB-first transfer
  — Can be selected regardless of the communication mode* (except in the case of asynchronous mode 7-bit data)
- Built-in baud rate generator allows any bit rate to be selected
- Choice of serial clock source: internal clock from baud rate generator or external clock from SCK pin
- Four interrupt sources
  — Four interrupt sources—transmit-data-empty, transmit-end, receive-data-full, and receive-error—that can issue requests independently
  — The transmit-data-empty and receive-data-full interrupts can activate the DMA controller (DMAC) or data 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 exiting module stop mode

Note: * Descriptions in this section refer to LSB-first transfer.

Figure 14.1 shows a block diagram of the SCI.

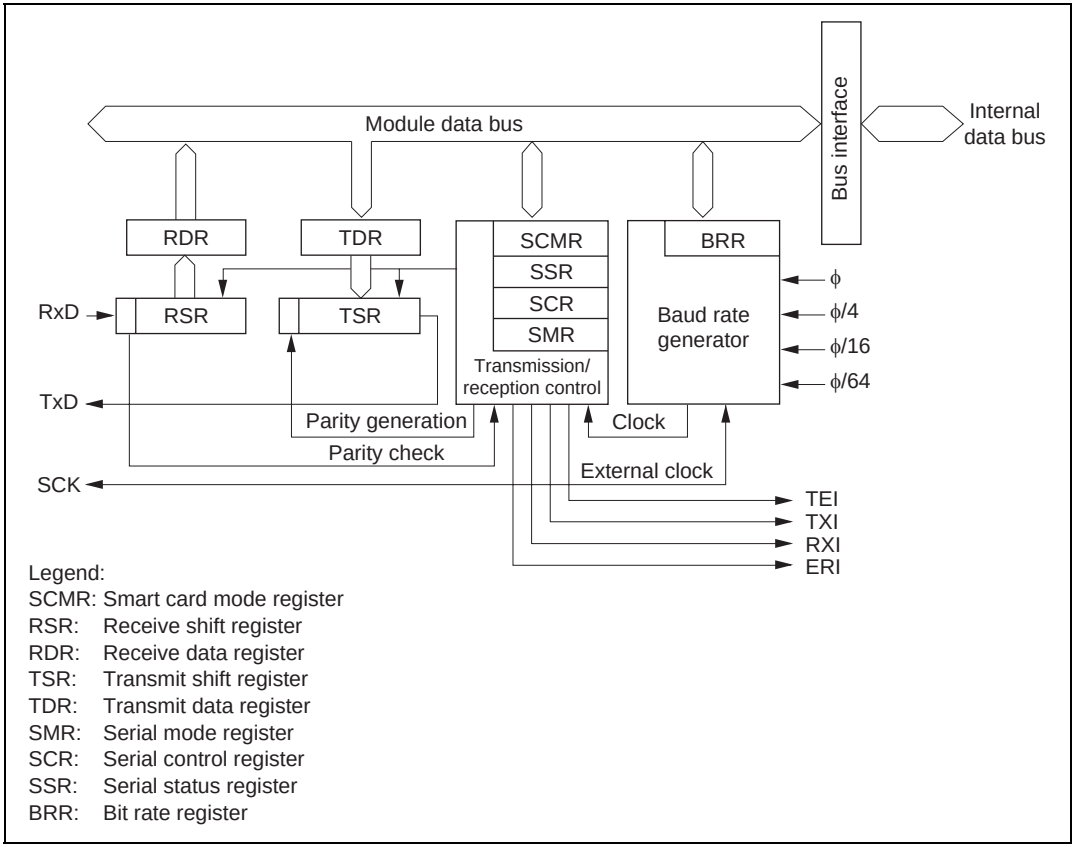


Figure 14.1 Block Diagram of SCI

Table 14.1 shows the serial pins for each SCI channel.

Table 14.1 SCI Pins

Channel	Pin Name	Symbol	I/O	Function
0	Serial clock pin 0	SCK0	I/O	SCI0 clock input/output
	Receive data pin 0	RxD0	Input	SCI0 receive data input
	Transmit data pin 0	TxD0	Output	SCI0 transmit data output
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 output
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 output

The SCI has the internal registers shown in table 14.2. These registers are used to specify asynchronous mode or synchronous mode, the data format, and the bit rate, and to control the transmitter/receiver.

Table 14.2 SCI Registers

Channel	Name	Abbreviation	R/W	Initial Value	Address*1
0	Serial mode register 0	SMR0	R/W	H'00	H'FF78
	Bit rate register 0	BRR0	R/W	H'FF	H'FF79
	Serial control register 0	SCR0	R/W	H'00	H'FF7A
	Transmit data register 0	TDR0	R/W	H'FF	H'FF7B
	Serial status register 0	SSR0	R/(W)*2	H'84	H'FF7C
	Receive data register 0	RDR0	R	H'00	H'FF7D
	Smart card mode register 0	SCMR0	R/W	H'F2	H'FF7E
1	Serial mode register 1	SMR1	R/W	H'00	H'FF80
	Bit rate register 1	BRR1	R/W	H'FF	H'FF81
	Serial control register 1	SCR1	R/W	H'00	H'FF82
	Transmit data register 1	TDR1	R/W	H'FF	H'FF83
	Serial status register 1	SSR1	R/(W)*2	H'84	H'FF84
	Receive data register 1	RDR1	R	H'00	H'FF85
	Smart card mode register 1	SCMR1	R/W	H'F2	H'FF86
2	Serial mode register 2	SMR2	R/W	H'00	H'FF88
	Bit rate register 2	BRR2	R/W	H'FF	H'FF89
	Serial control register 2	SCR2	R/W	H'00	H'FF8A
	Transmit data register 2	TDR2	R/W	H'FF	H'FF8B
	Serial status register 2	SSR2	R/(W)*2	H'84	H'FF8C
	Receive data register 2	RDR2	R	H'00	H'FF8D
	Smart card mode register 2	SCMR2	R/W	H'F2	H'FF8E
All	Module stop control register	MSTPCR	R/W	H'3FFF	H'FF3C

Notes: 1. Lower 16 bits of the address.
2. Can only be written with 0 for flag clearing.

14.2.1 Receive Shift Register (RSR)

Bit	:	7	6	5	4	3	2	1	0
R/W	:	—	—	—	—	—	—	—	—

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 with the LSB (bit 0), and converts it to parallel data. When one byte of data has been received, it is transferred to RDR automatically.

RSR cannot be directly read or written to by the CPU.

14.2.2 Receive Data Register (RDR)

Bit	:	7	6	5	4	3	2	1	0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R	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 from RSR to RDR where it is stored, and completes the receive operation. After this, RSR is receive-enabled.

Since RSR and RDR function as a double buffer in this way, continuous receive operations can 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, and in standby mode or module stop mode.

Bit	:	7	6	5	4	3	2	1	0
R/W	:	—	—	—	—	—	—	—	—

TSR is a register used to transmit serial data.

To perform serial data transmission, the SCI first transfers transmit data from TDR to TSR, then 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 TDR to TSR, and transmission started, automatically. However, data transfer from TDR to TSR is not performed if the TDRE bit in SSR is set to 1.

TSR cannot be directly read or written to by the CPU.

14.2.4 Transmit Data Register (TDR)

Bit	:	7	6	5	4	3	2	1	0
Initial value	:	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

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 to TSR and starts serial transmission. Continuous serial transmission can be carried out by writing the next 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, and in standby mode or module stop mode.

Bit	:	7	6	5	4	3	2	1	0
		C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0
Initial value	:	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

SMR is an 8-bit register used to set the SCI's serial transfer format and select the baud rate 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. In software standby mode and module stop mode it retains its previous state.

Bit 7—Communication Mode (C/$\overline{A}$): Selects asynchronous mode or synchronous mode as the SCI operating mode.

Bit 7 C/$\overline{A}$	**Description**	
0	Asynchronous mode	(Initial value)
1	Synchronous mode	

Bit 6—Character Length (CHR): Selects 7 or 8 bits as the data length in asynchronous mode. In synchronous mode, a fixed data length of 8 bits is used regardless of the CHR setting.

Bit 6 CHR	**Description**	
0	8-bit data	(Initial value)
1	7-bit data*	

Note: * When 7-bit data is selected, the MSB (bit 7) of TDR is not transmitted, and it is not possible to choose between LSB-first or MSB-first transfer.

multiprocessor format, parity bit addition and checking is not performed, regardless of the PE bit setting.

Bit 5 PE	Description	
0	Parity bit addition and checking disabled	(Initial value)
1	Parity bit addition and checking enabled*	

Note:* When the PE bit is set to 1, the parity (even or odd) specified by the O/$\overline{E}$ bit is added to transmit data before transmission. In reception, the parity bit is checked for the parity (even or odd) specified by the O/$\overline{E}$ bit.

Bit 4—Parity Mode (O/$\overline{E}$): Selects either even or odd parity for use in parity addition and checking.

The O/$\overline{E}$ bit setting is only valid when the PE bit is set to 1, enabling parity bit addition and checking, in asynchronous mode. The O/$\overline{E}$ bit setting is invalid in synchronous mode, and when parity addition and checking is disabled in asynchronous mode.

Bit 4 O/$\overline{E}$	Description	
0	Even parity*1	(Initial value)
1	Odd parity*2	

Notes: 1. When even parity is set, parity bit addition is performed in transmission so that the total 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 receive character plus the parity bit is even.
2. When odd parity is set, parity bit addition is performed in transmission so that the total 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 receive character plus the parity bit is odd.

Bit 3—Stop Bit Length (STOP): Selects 1 or 2 bits as the stop bit length in asynchronous mode. The STOP bits setting is only valid in asynchronous mode. If synchronous mode is set the STOP bit setting is invalid since stop bits are not added.

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.	(Initial value)
1	2 stop bits: In transmission, two 1-bits (stop bits) are added to the end of a transmit character before it is sent.	

In reception, only the first stop bit is checked, regardless of the STOP bit setting. If the second stop bit is 1, it is treated as a stop bit; if it is 0, it is treated as the start bit of the next transmit character.

Bit 2—Multiprocessor Mode (MP): Selects multiprocessor format. When multiprocessor format is selected, the PE bit and O/$\overline{E}$ bit parity settings are invalid. The MP bit setting is only valid in asynchronous mode; it is invalid in synchronous mode.

For details of the multiprocessor communication function, see section 14.3.3, Multiprocessor Communication Function.

Bit 2 MP	Description	
0	Multiprocessor function disabled	(Initial value)
1	Multiprocessor format selected	

Bits 1 and 0—Clock Select 1 and 0 (CKS1, CKS0): These bits select the clock source for the baud rate generator. The clock source can be selected from ϕ, $\phi/4$, $\phi/16$, and $\phi/64$, according to the setting of bits CKS1 and CKS0.

For the relation between the clock source, the bit rate register setting, and the baud rate, see section 14.2.8, Bit Rate Register (BRR).

Bit 1 CKS1	Bit 0 CKS0	Description	
0	0	ϕ clock	(Initial value)
	1	$\phi/4$ clock	
1	0	$\phi/16$ clock	
	1	$\phi/64$ clock	

Bit	:	7	6	5	4	3	2	1	0
		TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
Initial value	:	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

SCR is a register that performs enabling or disabling of SCI transfer operations, serial clock output 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. In software standby mode and module stop mode it retains its previous state.

Bit 7—Transmit Interrupt Enable (TIE): Enables or disables transmit-data-empty interrupt (TXI) request generation when serial transmit data is transferred from TDR to TSR and the TDRE flag in SSR is set to 1.

Bit 7 TIE	**Description**	
0	Transmit-data-empty interrupt (TXI) requests disabled*	(Initial value)
1	Transmit-data-empty interrupt (TXI) requests enabled	

Note:* TXI interrupt request cancellation can be performed by reading 1 from the TDRE flag, then clearing it to 0, or by clearing the TIE bit to 0.

from RSR to RDR and the RDRF flag in SSR is set to 1.

Bit 6 RIE	Description
0	Receive-data-full interrupt (RXI) request and receive-error interrupt (ERI) request disabled* (Initial value)
1	Receive-data-full interrupt (RXI) request and receive-error interrupt (ERI) request enabled

Note:* RXI and ERI interrupt request cancellation can be performed by reading 1 from the RDRF flag, or the FER, PER, or ORER flag, then clearing the flag to 0, or by clearing the RIE bit to 0.

Bit 5—Transmit Enable (TE): Enables or disables the start of serial transmission by the SCI.

Bit 5 TE	Description
0	Transmission disabled*1 (Initial value)
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 TDR and the TDRE flag in SSR is cleared to 0.

   SMR setting must be performed to decide the transfer format before setting the TE bit to 1.

Bit 4—Receive Enable (RE): Enables or disables the start of serial reception by the SCI.

Bit 4 RE	Description
0	Reception disabled*1 (Initial value)
1	Reception enabled*2

Notes: 1. Clearing the RE bit to 0 does not affect the RDRF, FER, PER, and ORER flags, which retain their states.

2. Serial reception is started in this state when a start bit is detected in asynchronous mode or serial clock input is detected in synchronous mode.

   SMR setting must be performed to decide the transfer format before setting the RE bit to 1.

The MPIE bit setting is invalid in synchronous mode or when the MP bit is cleared to 0.

Bit 3 MPIE	Description	
0	Multiprocessor interrupts disabled (normal reception performed) [Clearing conditions] • When the MPIE bit is cleared to 0 • When data with MPB = 1 is received	(Initial value)
1	Multiprocessor interrupts enabled* Receive-data-full interrupt (RXI) requests, receive-error interrupt (ERI) requests, and setting of the RDRF, FER, and ORER flags in SSR are disabled until data with the multiprocessor bit set to 1 is received.	

Note: * When receive data including MPB = 0 is received, receive data transfer from RSR to RDR, receive error detection, and setting of the RDRF, FER, and ORER flags in SSR, is not performed. When receive data including MPB = 1 is received, the MPB bit in SSR is set to 1, the MPIE bit is cleared to 0 automatically, and generation of RXI and ERI interrupts (when the TIE and RIE bits in SCR are set to 1) and FER and ORER flag setting is enabled.

Bit 2—Transmit End Interrupt Enable (TEIE): Enables or disables transmit-end interrupt (TEI) request generation when there is no valid transmit data in TDR in MSB data transmission.

Bit 2 TEIE	Description	
0	Transmit end interrupt (TEI) request disabled*	(Initial value)
1	Transmit end interrupt (TEI) request enabled*	

Note: * TEI cancellation can be performed by reading 1 from the TDRE flag in SSR, then clearing it to 0 and clearing the TEND flag to 0, or by clearing the TEIE bit to 0.

Bits 1 and 0—Clock Enable 1 and 0 (CKE1, CKE0): These bits are used to select the SCI clock source and enable or disable clock output from the SCK pin. The combination of the CKE1 and CKE0 bits determines whether the SCK pin functions as an I/O port, the serial clock output pin, or the serial clock input pin.

The setting of the CKE0 bit, however, is only valid for internal clock operation (CKE1 = 0) in asynchronous mode. The CKE0 bit setting is invalid in synchronous mode, and in the case of external clock operation (CKE1 = 1). Set CKE1 and CKE0 before determining the SCI operating mode with SMR.

Bit 1 CKE1	Bit 0 CKE0	Description	
0	0	Asynchronous mode	Internal clock/SCK pin functions as I/O port*1
		Synchronous mode	Internal clock/SCK pin functions as serial clock output
	1	Asynchronous mode	Internal clock/SCK pin functions as clock output*2
		Synchronous mode	Internal clock/SCK pin functions as serial clock output
1	0	Asynchronous mode	External clock/SCK pin functions as clock input*3
		Synchronous mode	External clock/SCK pin functions as serial clock input
	1	Asynchronous mode	External clock/SCK pin functions as clock input*3
		Synchronous mode	External clock/SCK pin functions as serial clock 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.

Bit	:	7	6	5	4	3	2	1	0
		TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
Initial value	:	1	0	0	0	0	1	0	0
R/W	:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/W

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

SSR is an 8-bit register containing status flags that indicate the operating status of the SCI, and multiprocessor bits.

SSR can be read or written to by the CPU at all times. However, 1 cannot be written to flags TDRE, RDRF, ORER, PER, and FER. Also note that in order to clear these flags they must be read as 1 beforehand. The TEND flag and MPB flag are read-only flags and cannot be modified.

SSR is initialized to H'84 by a reset, and in standby mode or module stop mode.

Bit 7—Transmit Data Register Empty (TDRE): Indicates that data has been transferred from 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 DMAC or DTC is activated by a TXI interrupt and writes data to TDR
1	[Setting conditions] (Initial value) • When the TE bit in SCR is 0 • When data is transferred from TDR to TSR and data can be written to TDR

Bit 6 RDRF	Description
0	[Clearing conditions] (Initial value) • When 0 is written to RDRF after reading RDRF = 1 • When the DMAC or 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

Note: RDR and the RDRF flag are not affected and retain their previous values when an error is 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 overrun error will occur and the receive data will be lost.

Bit 5—Overrun Error (ORER): Indicates that an overrun error occurred during reception, causing abnormal termination.

Bit 5 ORER	Description
0	[Clearing condition] (Initial value)*1 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 in SCR is cleared to 0.

2. The receive data prior to the overrun error is retained in RDR, and the data received subsequently is lost. Also, subsequent serial reception cannot be continued while the ORER flag is set to 1. In synchronous mode, serial transmission cannot be continued, either.

Bit 4 FER	Description
0	[Clearing condition] (Initial value)*1 When 0 is written to FER after reading FER = 1
1	[Setting condition] When the SCI checks the stop bit at the end of the receive data when 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 in SCR is cleared to 0.

2. In 2-stop-bit mode, only the first stop bit is checked for a value of 0; the second stop bit is not checked. If a framing error occurs, the receive data is transferred to RDR but the RDRF flag is not set. Also, subsequent serial reception cannot be continued while the FER flag is set to 1. In synchronous mode, serial transmission cannot be continued, either.

Bit 3—Parity Error (PER): Indicates that a parity error occurred during reception using parity addition in asynchronous mode, causing abnormal termination.

Bit 3 PER	Description
0	[Clearing condition] (Initial value)*1 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 bit does not 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 in SCR is cleared to 0.

2. If a parity error occurs, the receive data is transferred to RDR but the RDRF flag is not set. Also, subsequent serial reception cannot be continued while the PER flag is set to 1. In synchronous mode, serial transmission cannot be continued, either.

The TEND flag is read-only and cannot be modified.

Bit 2 TEND	Description	
0	[Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DMAC or DTC is activated by a TXI interrupt and writes data to TDR	
1	[Setting conditions] • When the TE bit in SCR is 0 • When TDRE = 1 at transmission of the last bit of a 1-byte serial transmit character	(Initial value)

Bit 1—Multiprocessor Bit (MPB): When reception is performed using multiprocessor format in 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] When data with a 0 multiprocessor bit is received	(Initial value)*
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 multiprocessor format.

Bit 0—Multiprocessor Bit Transfer (MPBT): When transmission is performed using multiprocessor format in asynchronous mode, MPBT stores the multiprocessor bit to be added to the transmit data.

The MPBT bit setting is invalid when multiprocessor format is not used, when not transmitting, and in synchronous mode.

Bit 0 MPBT	Description	
0	Data with a 0 multiprocessor bit is transmitted	(Initial value)
1	Data with a 1 multiprocessor bit is transmitted	

Bit	:	7	6	5	4	3	2	1	0
Initial value	:	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

BRR is an 8-bit register that sets the serial transfer bit rate in accordance with the baud rate 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. In software standby mode and module stop mode it retains its previous state.

As baud rate generator control is performed independently for each channel, different values can be set for each channel.

Table 14.3 shows sample BRR settings in asynchronous mode, and table 14.4 shows sample BRR settings in synchronous mode.

Table 14.3 BRR Settings for Various Bit Rates (Asynchronous Mode)

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

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

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

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

	φ = 14 MHz			φ = 14.7456 MHz			φ = 16 MHz			φ = 17.2032 MHz		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	248	–0.17	3	64	0.70	3	70	0.03	3	75	0.48
150	2	181	0.16	2	191	0.00	2	207	0.16	2	223	0.00
300	2	90	0.16	2	95	0.00	2	103	0.16	2	111	0.00
600	1	181	0.16	1	191	0.00	1	207	0.16	1	223	0.00
1200	1	90	0.16	1	95	0.00	1	103	0.16	1	111	0.00
2400	0	181	0.16	0	191	0.00	0	207	0.16	0	223	0.00
4800	0	90	0.16	0	95	0.00	0	103	0.16	0	111	0.00
9600	0	45	–0.93	0	47	0.00	0	51	0.16	0	55	0.00
19200	0	22	–0.93	0	23	0.00	0	25	0.16	0	27	0.00
31250	0	13	0.00	0	14	–1.70	0	15	0.00	0	16	1.20
38400	—	—	—	0	11	0.00	0	12	0.16	0	13	0.00

(bits/s)	n	N	(%)	n	N	(%)	n	N	(%)	n	N	(%)
110	3	79	–0.12	3	86	0.31	3	88	–0.25	3	110	–0.02
150	2	233	0.16	2	255	0.00	3	64	0.16	3	80	0.47
300	2	116	0.16	2	127	0.00	2	129	0.16	2	162	–0.15
600	1	233	0.16	1	255	0.00	2	64	0.16	2	80	0.47
1200	1	116	0.16	1	127	0.00	1	129	0.16	1	162	–0.15
2400	0	233	0.16	0	255	0.00	1	64	0.16	1	80	0.47
4800	0	116	0.16	0	127	0.00	0	129	0.16	0	162	–0.15
9600	0	58	–0.69	0	63	0.00	0	64	0.16	0	80	0.47
19200	0	28	1.02	0	31	0.00	0	32	–1.36	0	40	–0.76
31250	0	17	0.00	0	19	–1.70	0	19	0.00	0	24	1.00
38400	0	14	–2.34	0	15	0.00	0	15	1.73	0	19	1.73

Bit Rate	φ = 2 MHz		φ = 4 MHz		φ = 8 MHz		φ = 10 MHz		φ = 16 MHz		φ = 20 MHz		φ = 25 MHz	
(bits/s)	n	N	n	N	n	N	n	N	n	N	n	N	n	N
110	3	70												
250	2	124	2	249	3	124	—	—	3	249				
500	1	249	2	124	2	249	—	—	3	124	—	—		
1 k	1	124	1	249	2	124	—	—	2	249	—	—	3	97
2.5 k	0	199	1	99	1	199	2	124	2	99	2	124	2	155
5 k	0	99	0	199	1	99	1	249	1	199	1	249	2	77
10 k	0	49	0	99	0	199	1	124	1	99	1	124	1	155
25 k	0	19	0	39	0	79	0	199	0	159	0	199	0	249
50 k	0	9	0	19	0	39	0	99	0	79	0	99	0	124
100 k	0	4	0	9	0	19	0	49	0	39	0	49	0	62
250 k	0	1	0	3	0	7	0	19	0	15	0	19	0	24
500 k	0	0*	0	1	0	3	0	9	0	7	0	9	—	—
1 M			0	0*	0	1			0	3	0	4	—	—
2.5 M							0	0*			0	1	—	—
5 M											0	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 1%.

Asynchronous mode:

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

Synchronous mode:

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

Where B: Bit rate (bits/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:

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

Table 14.5 Maximum Bit Rate for Each Frequency (Asynchronous Mode)

ϕ (MHz)	Maximum Bit Rate (bits/s)	n	N
2	62500	0	0
2.097152	65536	0	0
2.4576	76800	0	0
3	93750	0	0
3.6864	115200	0	0
4	125000	0	0
4.9152	153600	0	0
5	156250	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
25	781250	0	0

ϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bits/s)
2	0.5000	31250
2.097152	0.5243	32768
2.4576	0.6144	38400
3	0.7500	46875
3.6864	0.9216	57600
4	1.0000	62500
4.9152	1.2288	76800
5	1.2500	78125
6	1.5000	93750
6.144	1.5360	96000
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
25	6.2500	390625

φ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bits/s)
2	0.3333	333333.3
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
25	4.1667	4166666.7

14.2.9 Smart Card Mode Register (SCMR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	SDIR	SINV	—	SMIF
Initial value	:	1	1	1	1	0	0	1	0
R/W	:	—	—	—	—	R/W	R/W	—	R/W

SCMR selects LSB-first or MSB-first transfer by means of bit SDIR. Except in the case of asynchronous mode 7-bit data, LSB-first or MSB-first transfer can be selected regardless of the serial communication mode. The descriptions in this chapter refer to LSB-first transfer.

For details of the other bits in SCMR, see section 15.2.1, Smart Card Mode Register (SCMR).

SCMR is initialized to H'F2 by a reset and in hardware standby mode. In software standby mode and module stop mode it retains its previous state.

Bits 7 to 4—Reserved: These bits cannot be modified and are always read as 1.

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	(Initial value)
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. The SINV bit does not affect the logic level of the parity bit(s): parity bit inversion requires inversion of the O/$\overline{E}$ bit in SMR.

Bit 2 SINV	Description	
0	TDR contents are transmitted without modification Receive data is stored in RDR without modification	(Initial value)
1	TDR contents are inverted before being transmitted Receive data is stored in RDR in inverted form	

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

Bit 0—Smart Card Interface Mode Select (SMIF): When the smart card interface operates as a normal SCI, 0 should be written to this bit.

Bit 0 SMIF	Description	
0	Operates as normal SCI (smart card interface function disabled)	(Initial value)
1	Smart card interface function enabled	

	MSTPCRH								MSTPCRL							
Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value :	0	0	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/W	R/W	R/W

MSTPCR is a 16-bit readable/writable register that performs module stop mode control.

When the corresponding bit of bits MSTP7 to MSTP5 is set to 1, SCI operation stops at the end of the bus cycle and a transition is made to module stop mode. Registers cannot be read or written to in module stop mode. For details, see section 21.5, Module Stop Mode.

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

Bit 7—Module Stop (MSTP7): Specifies the SCI channel 2 module stop mode.

Bit 7 MSTP7	Description	
0	SCI channel 2 module stop mode cleared	
1	SCI channel 2 module stop mode set	(Initial value)

Bit 6—Module Stop (MSTP6): Specifies the SCI channel 1 module stop mode.

Bit 6 MSTP6	Description	
0	SCI channel 1 module stop mode cleared	
1	SCI channel 1 module stop mode set	(Initial value)

Bit 5—Module Stop (MSTP5): Specifies the SCI channel 0 module stop mode.

Bit 5 MSTP5	Description	
0	SCI channel 0 module stop mode cleared	
1	SCI channel 0 module stop mode set	(Initial value)

14.3.1 Overview

The SCI can carry out serial communication in two modes: asynchronous mode in which synchronization is achieved character by character, and synchronous mode in which synchronization is achieved with clock pulses.

Selection of asynchronous or synchronous mode and the transmission format is made using SMR as shown in table 14.8. The SCI clock is determined by a combination of the C/$\overline{A}$ bit in SMR and the CKE1 and CKE0 bits in SCR, as shown in table 14.9.

Asynchronous Mode

- Data length: Choice of 7 or 8 bits
- Choice of parity addition, multiprocessor bit addition, and addition of 1 or 2 stop bits (the combination of these parameters determines the transfer format and character length)
- 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 frequency as 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 built-in baud rate generator is not used)

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 off-chip
  - When external clock is selected:
    The built-in baud rate generator is not used, and the SCI operates on the input serial clock

SMR Settings						SCI Transfer Format			
Bit 7	Bit 6	Bit 2	Bit 5	Bit 3					
C/$\overline{A}$	CHR	MP	PE	STOP	Mode	Data Length	Multi-processor Bit	Parity Bit	Stop Bit Length
0	0	0	0	0	Asynchronous mode	8-bit data	No	No	1 bit
				1					2 bits
			1	0				Yes	1 bit
				1					2 bits
	1		0	0		7-bit data		No	1 bit
				1					2 bits
			1	0				Yes	1 bit
				1					2 bits
	0	1	—	0	Asynchronous mode (multi-processor format)	8-bit data	Yes	No	1 bit
			—	1					2 bits
	1		—	0		7-bit data			1 bit
			—	1					2 bits
1	—	—	—	—	Synchronous mode	8-bit data	No		None

Table 14.9 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 frequency as bit rate
	1	0		External	Inputs clock with frequency of 16 times the bit rate
		1			
1	0	0	Synchronous mode	Internal	Outputs serial clock
		1			
	1	0		External	Inputs serial clock
		1			

In asynchronous mode, characters are sent or received, each preceded by a start bit indicating the start of communication and one or two stop bits indicating the end of communication. Serial communication is thus carried out with synchronization established on a character-by-character basis.

Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communication. Both the transmitter and the receiver also have a double-buffered structure, so that data can be read or written during transmission or reception, enabling continuous data transfer.

Figure 14.2 shows the general format for asynchronous serial communication.

In asynchronous serial communication, the communication line is usually held in the mark state (high level). The SCI monitors the communication line, and when it goes to the space state (low level), recognizes a start bit and starts serial communication.

One serial communication character consists of a start bit (low level), followed by data (in LSB-first order), a parity bit (high or low level), and finally one or two stop bits (high level).

In asynchronous mode, the SCI performs synchronization at the falling edge of the start bit in reception. The SCI samples the data on the 8th pulse of a clock with a frequency of 16 times the length of one bit, so that the transfer data is latched at the center of each bit.

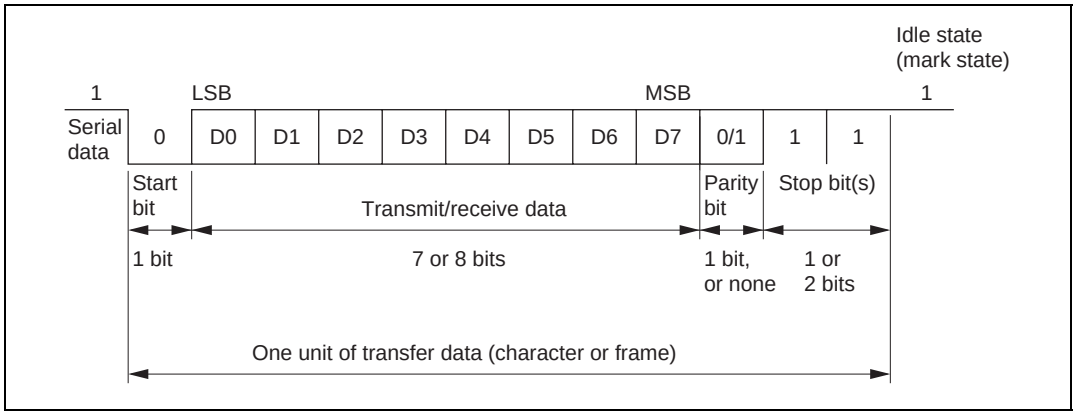


Figure 14.2 Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, Two Stop Bits)

Table 14.10 shows the data transfer formats that can be used in asynchronous mode. Any of 12 transfer formats can be selected according to the SMR setting.

Table 14.10 Serial Transfer Formats (Asynchronous Mode)

SMR Settings				Serial Transfer Format and Frame Length
CHR	PE	MP	STOP	1 2 3 4 5 6 7 8 9 10 11 12
0	0	0	0	S 8-bit data STOP
0	0	0	1	S 8-bit data STOP STOP
0	1	0	0	S 8-bit data P STOP
0	1	0	1	S 8-bit data P STOP STOP
1	0	0	0	S 7-bit data STOP
1	0	0	1	S 7-bit data STOP STOP
1	1	0	0	S 7-bit data P STOP
1	1	0	1	S 7-bit data P STOP STOP
0	—	1	0	S 8-bit data MPB STOP
0	—	1	1	S 8-bit data MPB STOP STOP
1	—	1	0	S 7-bit data MPB STOP
1	—	1	1	S 7-bit data MPB STOP STOP

Legend:
S: Start bit
STOP: Stop bit
P: Parity bit
MPB: Multiprocessor bit

Either an internal clock generated by the built-in baud rate generator or an external clock input at the SCK pin can be selected as the SCI's serial clock, according to the setting of the C/$\overline{A}$ bit in SMR and the CKE1 and CKE0 bits in SCR. For details of SCI clock source selection, see table 14.9.

When an external clock is input at the SCK pin, the clock frequency should be 16 times the bit rate used.

When the SCI is operated on an internal clock, the clock can be output from the SCK pin. The frequency of the clock output in this case is equal to the bit rate, and the phase is such that the rising edge of the clock is at the center of each transmit data bit, as shown in figure 14.3.

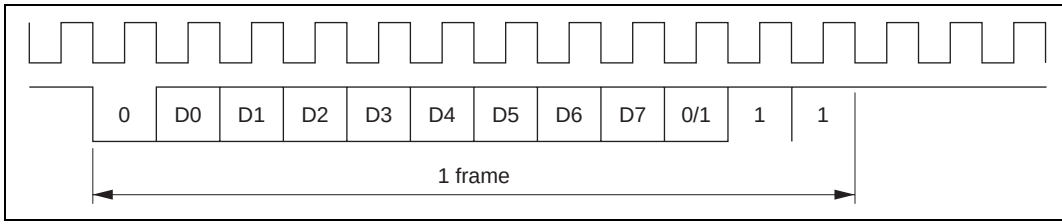


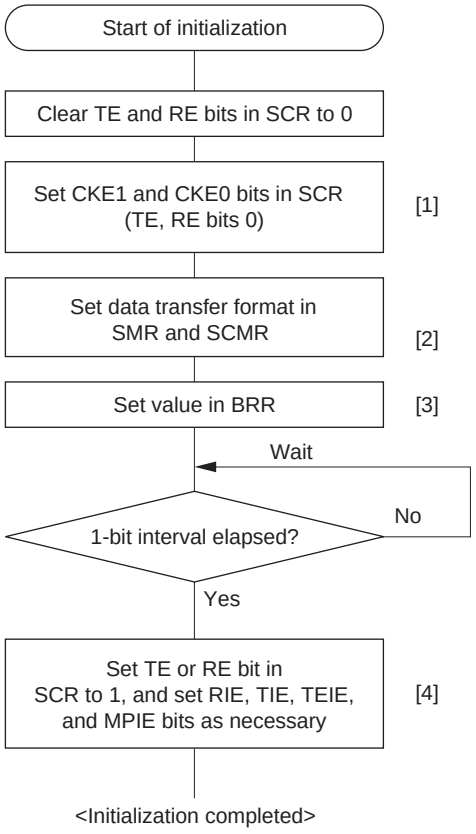
Figure 14.3 Relation between Output Clock and Transfer Data Phase (Asynchronous Mode)

Data Transfer Operations

SCI initialization (asynchronous mode): Before transmitting or receiving data, first 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 cleared to 0 before making the change using the following procedure. When the TE bit is cleared to 0, the TDRE flag is set to 1 and TSR is initialized. Note that clearing the RE bit to 0 does not change the 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, including initialization, since operation will be unreliable in this case.



[1] Set the clock selection in SCR. Be sure to clear bits RIE, TIE, TEIE, MPIE, TE, and RE, to 0.

When the clock is selected in asynchronous mode, it is output immediately after SCR settings are made.

[2] Set the data transfer format in SMR and SCMR.

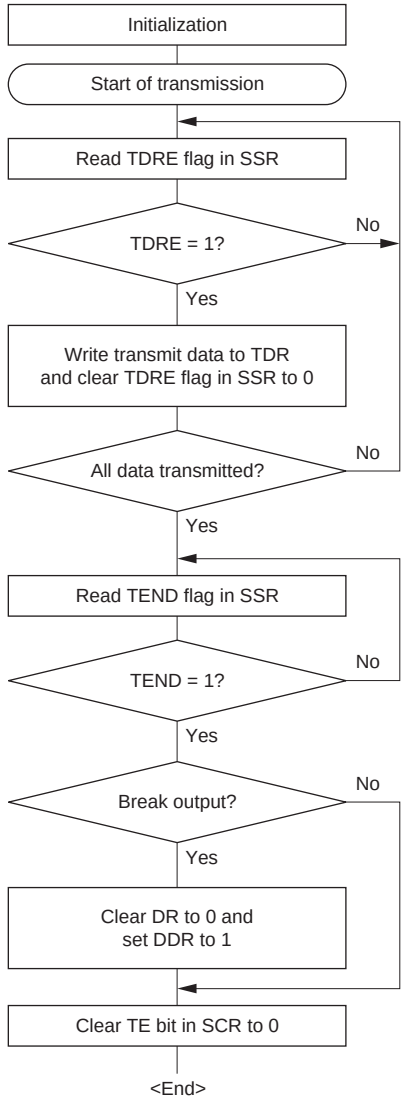
[3] Write a value corresponding to the bit rate to BRR. (Not necessary if an external clock is used.)

[4] Wait at least one bit interval, then set the TE bit or RE bit in SCR to 1. Also set the RIE, TIE, TEIE, and MPIE bits as necessary.

Setting the TE or RE bit enables the TxD or RxD pin to be used.

Figure 14.4 Sample SCI Initialization Flowchart

The following procedure should be used for serial data transmission.



[1] SCI initialization:
The TxD pin is automatically designated as the transmit data output pin.
After the TE bit is set to 1, a frame of 1s is output, and transmission is enabled.

[2] SCI status check and transmit data write:
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR and clear the TDRE flag to 0.

[3] Serial transmission continuation procedure:
To continue serial transmission, read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and then clear the TDRE flag to 0. Checking and clearing of the TDRE flag is automatic when the DMAC or DTC is activated by a transmit-data-empty interrupt (TXI) request, and data is written to TDR.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, set DDR for the port corresponding to the TxD pin to 1, clear DR to 0, then clear the TE bit in SCR to 0.

Figure 14.5 Sample Serial Transmission Flowchart

[1] The SCI monitors the TDRE flag in SSR, and if is 0, recognizes that data has been written to TDR, and transfers the data from TDR to TSR.

[2] After transferring data from TDR to TSR, the SCI sets the TDRE flag to 1 and starts transmission.

If the TIE bit is set to 1 at this time, a transmit-data-empty interrupt (TXI) is generated.

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 also be 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 sent.

[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 bit is sent, 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, and then the mark state is entered in which 1 is output continuously. If the TEIE bit in SCR is set to 1 at this time, a TEI interrupt request is generated.

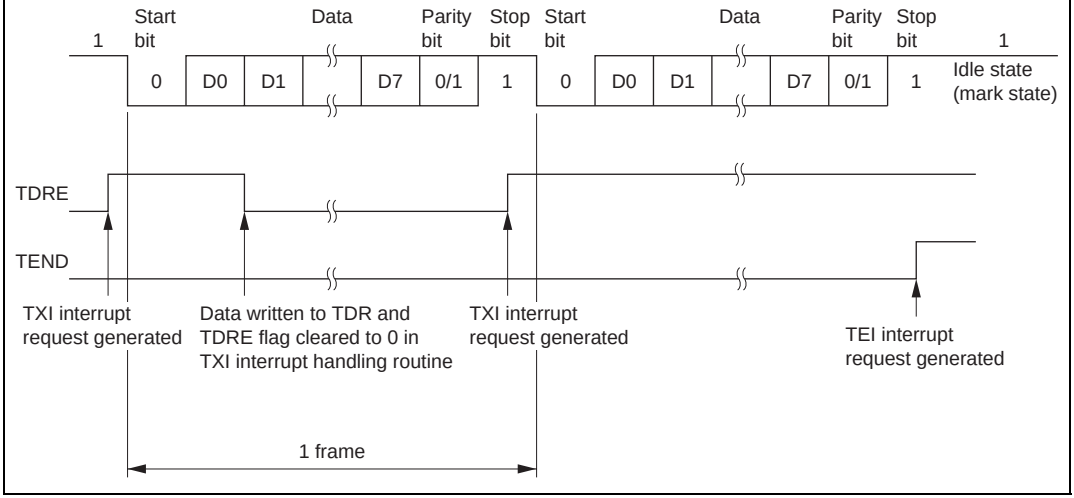
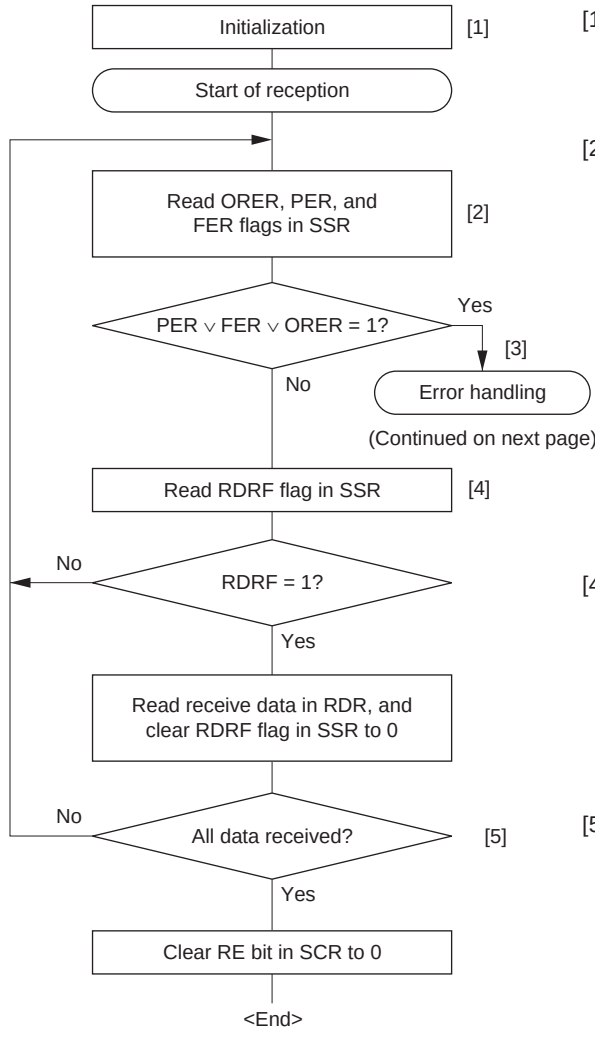


Figure 14.6 Example of Transmit Operation in Asynchronous Mode (Example with 8-Bit Data, Parity, One Stop Bit)

The following procedure should be used for serial data reception.



[1] SCI initialization:
The RxD pin is automatically designated as the receive data input pin.

[2] [3] Receive error handling and break detection:
If a receive error occurs, read the ORER, PER, and FER flags in SSR to identify the error. After performing the appropriate error processing, ensure that the ORER, PER, and FER flags are all cleared to 0. Reception cannot be resumed if any of these flags are set to 1. In the case of a framing error, a break can be detected by reading the value of the input port corresponding to the RxD pin.

[4] SCI status check and receive data read :
Read SSR and check that RDRF = 1, then read the receive data in RDR and clear the RDRF flag to 0. Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial reception continuation procedure:
To continue serial reception, before the stop bit for the current frame is received, read the RDRF flag, read RDR, and clear the RDRF flag to 0. The RDRF flag is cleared automatically when the DMAC or DTC is activated by an RXI interrupt and the RDR value is read.

Figure 14.7 Sample Serial Reception Flowchart

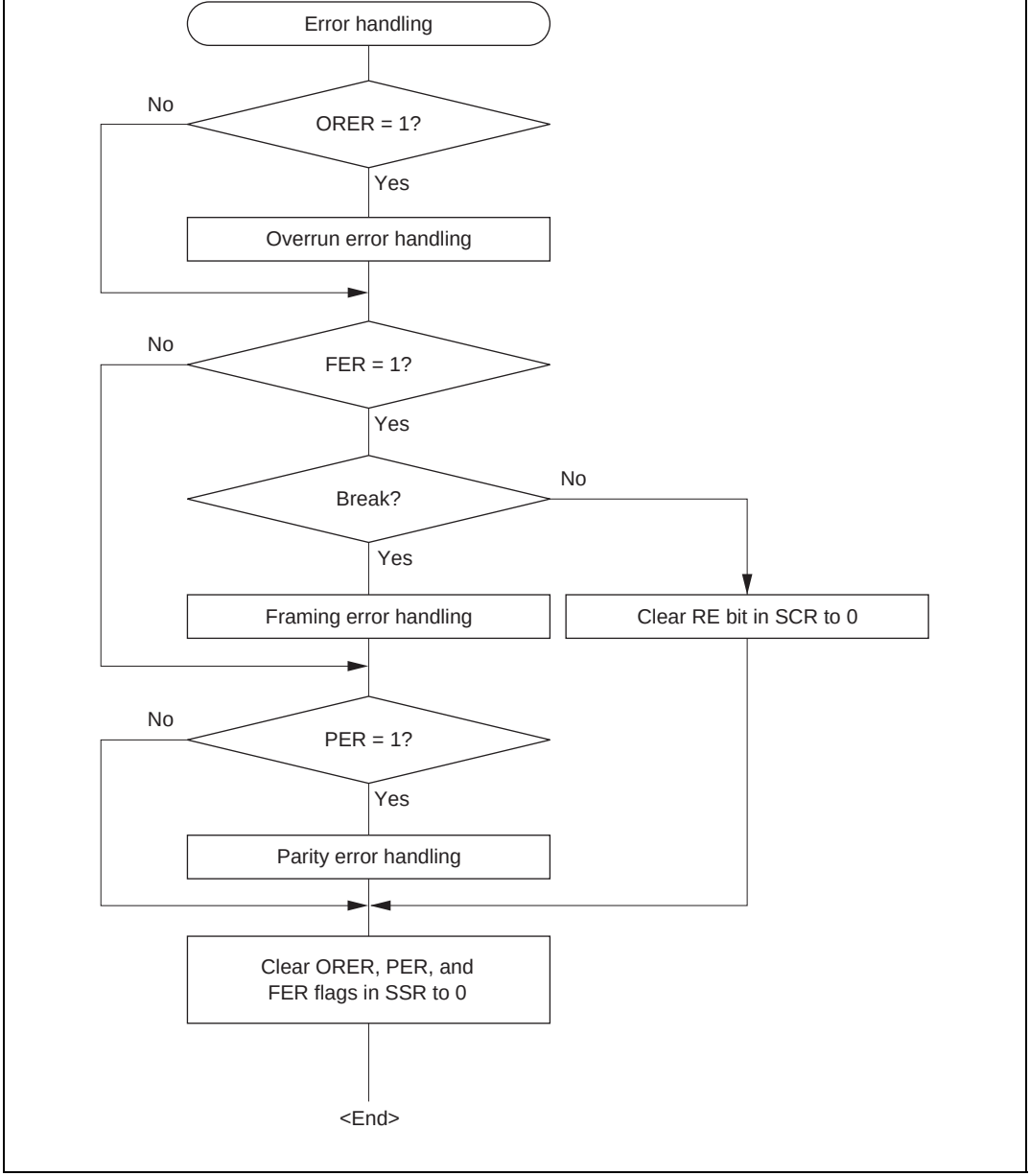


Figure 14.7 Sample Serial Reception Flowchart (cont)

[1] The SCI monitors the communication line, and if a 0 stop bit is detected, performs internal synchronization and starts reception.

[2] The received data is stored in RSR in LSB-to-MSB order.

[3] The parity bit and stop bit are received.

After receiving these bits, the SCI carries out the following checks.

[a] Parity check:

The SCI checks whether the number of 1 bits in the receive data agrees with the parity (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 can be transferred from RSR to RDR.

If all the above checks are passed, the RDRF flag is set to 1, and the receive data is stored in RDR.

If a receive error* is detected in the error check, the operation is as shown in table 14.11.

Note: * Subsequent receive operations cannot be performed when a receive error has occurred. Also note that the RDRF flag is not set to 1 in reception, and so the error flags must be cleared to 0.

[4] If the RIE bit in SCR is set to 1 when the RDRF flag changes to 1, a receive-data-full interrupt (RXI) request is generated.

Also, if the RIE bit in SCR is set to 1 when the ORER, PER, or FER flag changes to 1, a receive-error interrupt (ERI) request is generated.

Receive Error	Abbreviation	Condition	Data Transfer
Overrun error	ORER	When the next data reception is completed while the RDRF flag in SSR is set to 1	Receive data is not transferred from RSR to RDR
Framing error	FER	When the stop bit is 0	Receive data is transferred from RSR to RDR
Parity error	PER	When the received data differs from the parity (even or odd) set in SMR	Receive data is transferred from RSR to RDR

Figure 14.8 shows an example of the operation for reception in asynchronous mode.

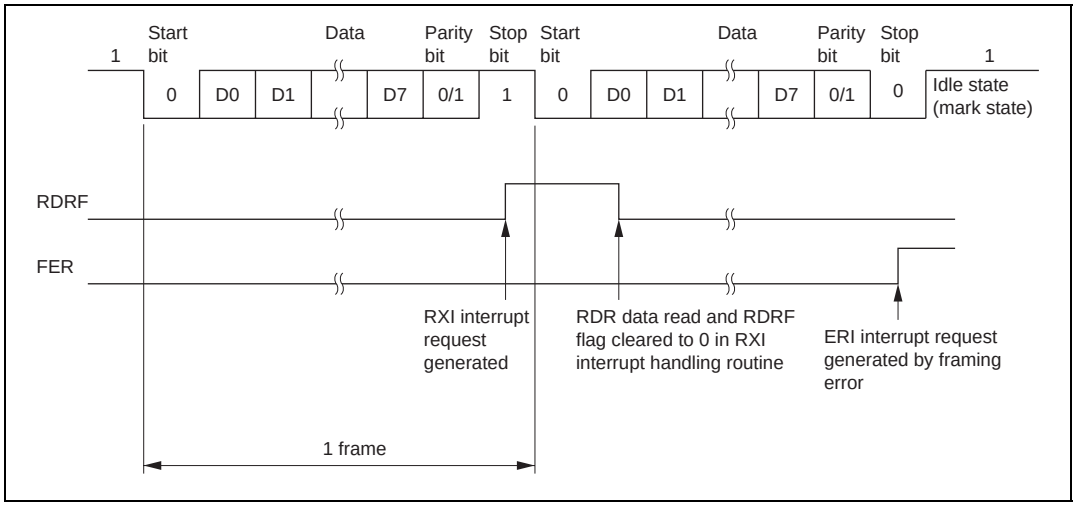


Figure 14.8 Example of SCI Receive Operation (Example with 8-Bit Data, Parity, One Stop Bit)

The multiprocessor communication function performs serial communication using the multiprocessor format, in which a multiprocessor bit is added to the transfer data, in asynchronous mode. Use of this function enables data transfer to be performed among a number of processors sharing a single serial communication line.

When multiprocessor communication is carried out, each receiving station is addressed by a unique ID code.

The serial communication cycle consists of two component cycles: an ID transmission cycle which specifies the receiving station, and a data transmission cycle. The multiprocessor bit is used 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 to perform serial communication as data with a 1 multiprocessor bit added. It then sends transmit data as data with a 0 multiprocessor bit added.

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 data with its own ID. The station whose ID matches then receives the data sent next. Stations whose ID does not match continue to skip the data until data with a 1 multiprocessor bit is again received. In this way, data communication is carried out among a number of processors.

Figure 14.9 shows an example of inter-processor communication using the multiprocessor format.

Data Transfer Formats

There are four data transfer formats.

When the multiprocessor format is specified, the parity bit specification is invalid.

For details, see table 14.10.

See section 14.3.2, Operation in Asynchronous Mode.

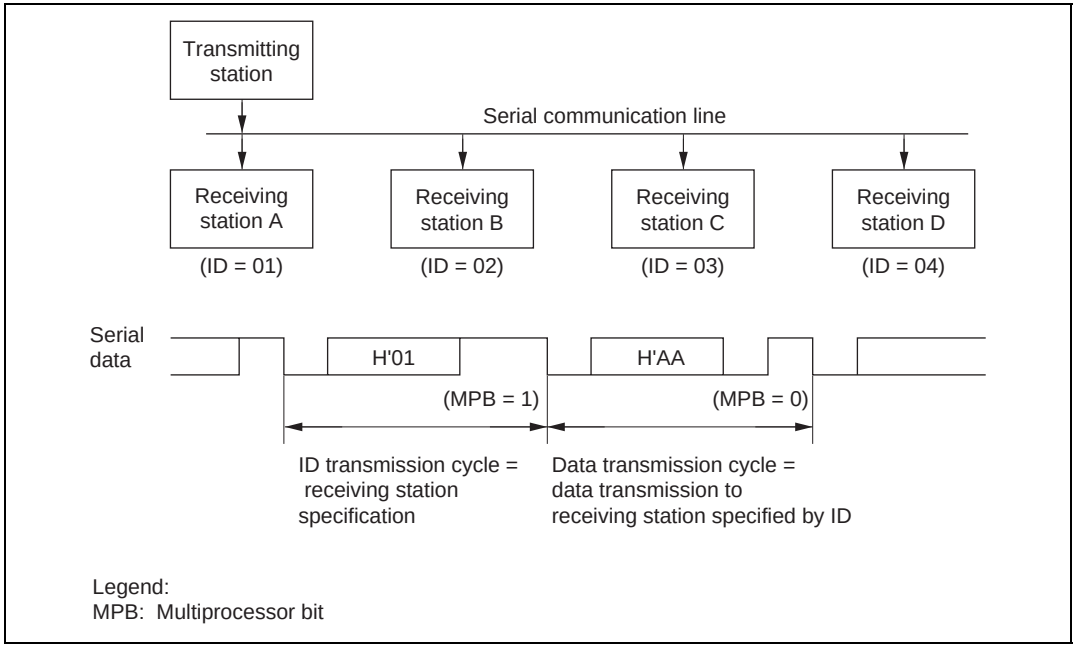
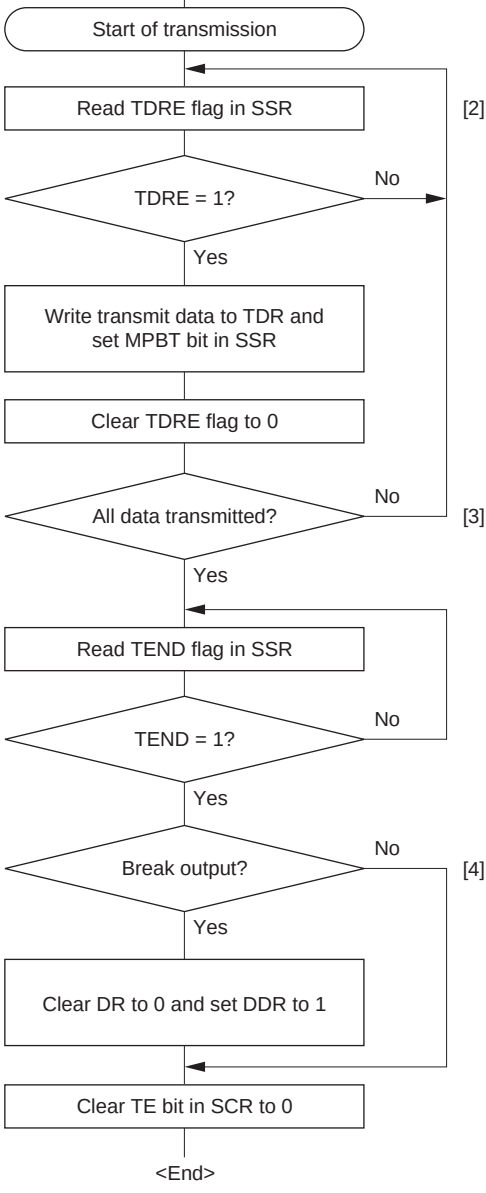


Figure 14.9 Example of Inter-Processor Communication Using Multiprocessor Format (Transmission of Data H'AA to Receiving Station A)

Data Transfer Operations

Multiprocessor serial data transmission: Figure 14.10 shows a sample flowchart for multiprocessor serial data transmission.

The following procedure should be used for multiprocessor serial data transmission.



designated as the transmit data output pin.
After the TE bit is set to 1, a frame of 1s is output, and transmission is enabled.

[2] SCI status check and transmit data write:
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR. Set the MPBT bit in SSR to 0 or 1. Finally, clear the TDRE flag to 0.

[3] Serial transmission continuation procedure:
To continue serial transmission, be sure to read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and then clear the TDRE flag to 0. Checking and clearing of the TDRE flag is automatic when the DMAC or DTC is activated by a transmit-data-empty interrupt (TXI) request, and data is written to TDR.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, set the port DDR to 1, clear DR to 0, then clear the TE bit in SCR to 0.

Figure 14.10 Sample Multiprocessor Serial Transmission Flowchart

[1] The SCI monitors the TDRE flag in SSR, and if is 0, recognizes that data has been written to TDR, and transfers the data from TDR to TSR.

[2] After transferring data from TDR to TSR, the SCI sets the TDRE flag to 1 and starts transmission.

If the TIE bit is set to 1 at this time, a transmit-data-empty interrupt (TXI) is generated.

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 is sent, 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, and then the mark state is entered in which 1 is output continuously. If the TEIE bit in SCR is set to 1 at this time, a transmit-end interrupt (TEI) request is generated.

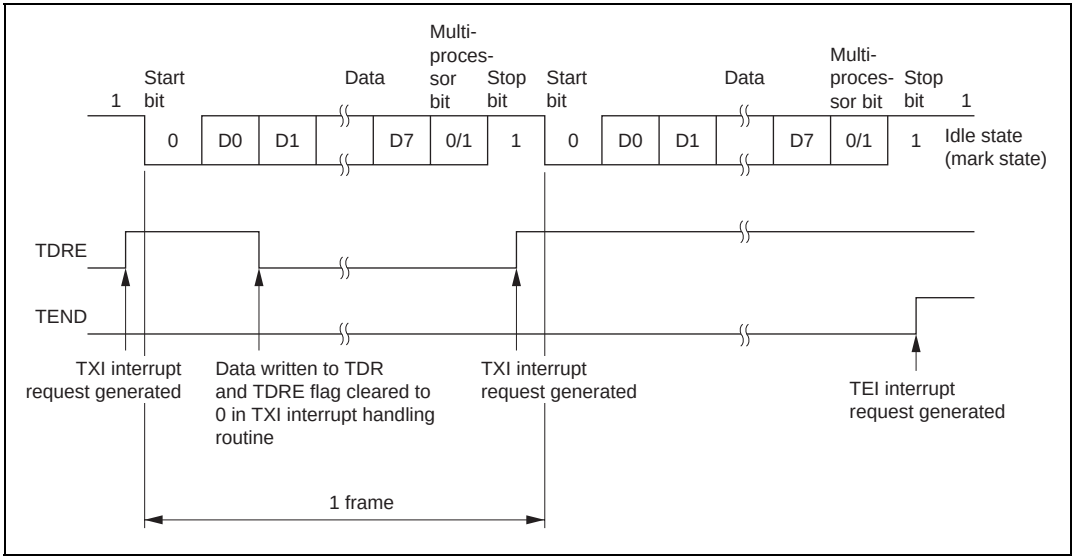
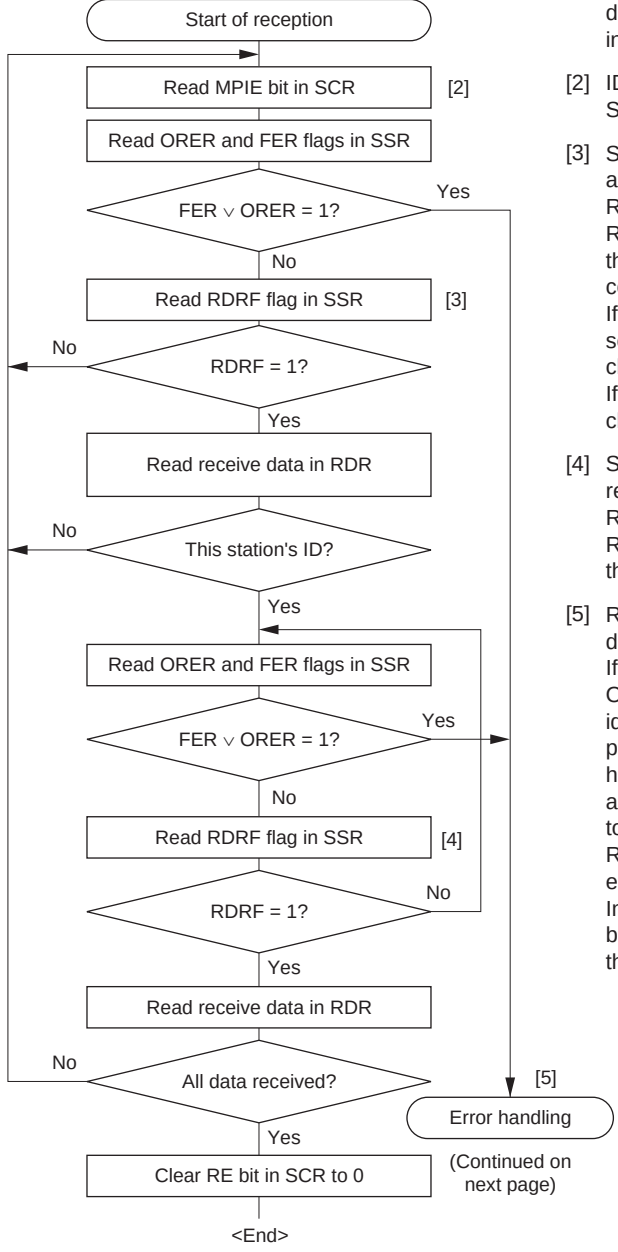


Figure 14.11 Example of SCI Transmit Operation
(Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)

Multiprocessor serial data reception: Figure 14.12 shows a sample flowchart for multiprocessor serial reception.

The following procedure should be used for multiprocessor serial data reception.



designated as the receive data input pin.

[2] ID reception cycle:
Set the MPIE bit in SCR to 1.

[3] SCI status check, ID reception and comparison:
Read SSR and check that the RDRF flag is set to 1, then read the receive data in RDR and compare it with this station's ID.
If the data is not this station's ID, set the MPIE bit to 1 again, and 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 the RDRF flag is set to 1, then read the data in RDR.

[5] Receive error handling and break detection:
If a receive error occurs, read the ORER and FER flags in SSR to identify the error. After performing the appropriate error handling, ensure that the ORER and FER flags are both cleared to 0.
Reception cannot be resumed if either of these flags is set to 1.
In the case of a framing error, a break can be detected by reading the RxD pin value.

Figure 14.12 Sample Multiprocessor Serial Reception Flowchart

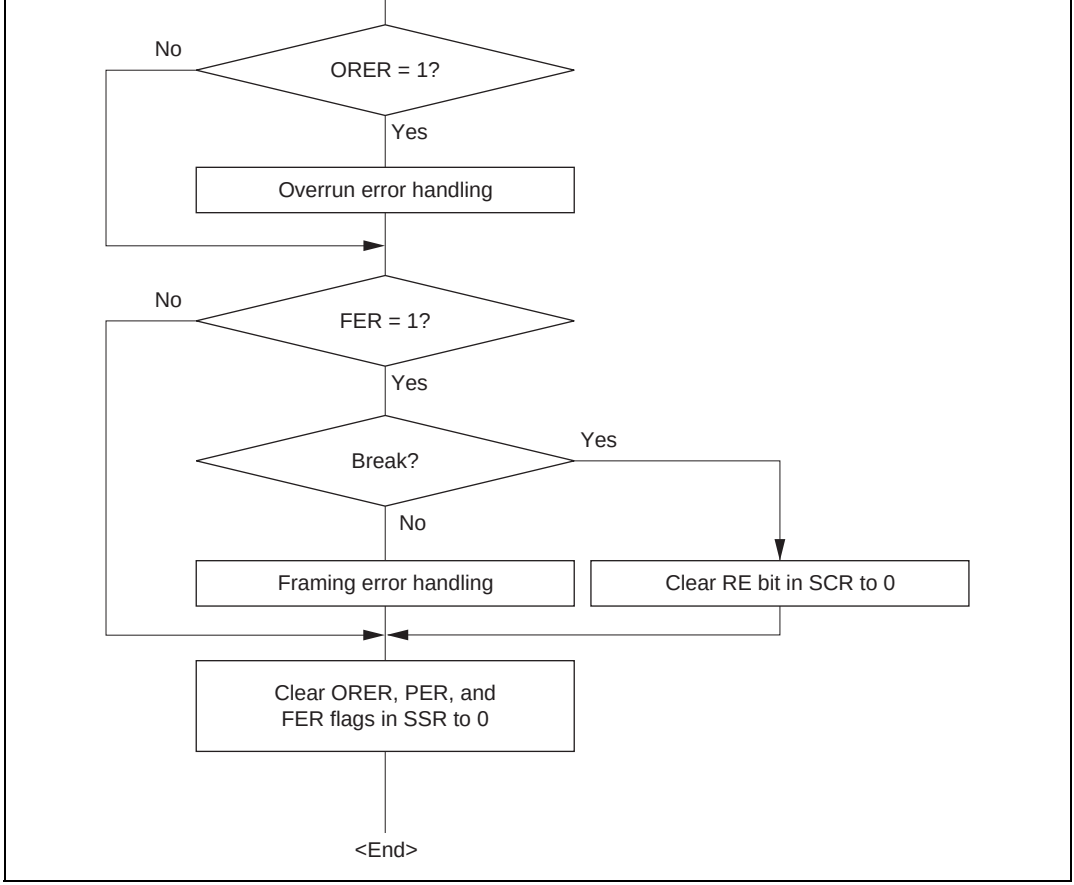


Figure 14.12 Sample Multiprocessor Serial Reception Flowchart (cont)

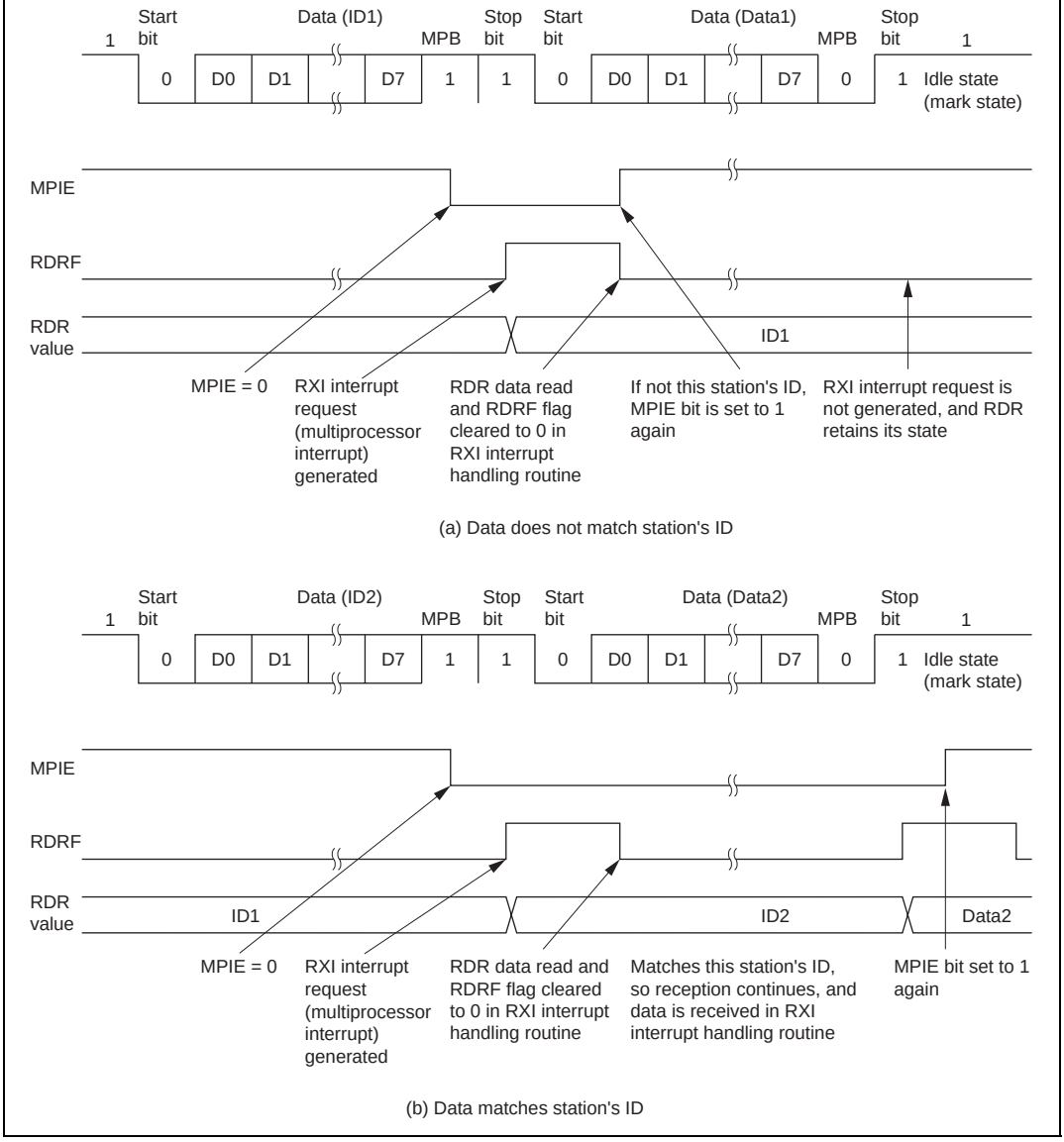


Figure 14.13 Example of SCI Receive Operation (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)

In synchronous mode, data is transmitted or received in synchronization with clock pulses, making it suitable for high-speed serial communication.

Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communication by use of a common clock. Both the transmitter and the receiver also have a double-buffered structure, so that data can be read or written during transmission or reception, enabling continuous data transfer.

Figure 14.14 shows the general format for synchronous serial communication.

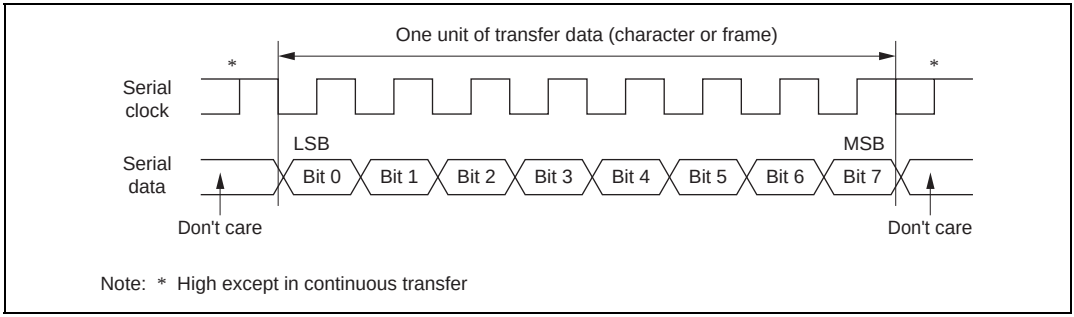


Figure 14.14 Data Format in Synchronous Communication

In synchronous serial communication, data on the communication line is output from one falling edge of the serial clock to the next. Data confirmation is guaranteed at the rising edge of the serial clock.

In synchronous serial communication, one character consists of data output starting with the LSB and ending with the MSB. After the MSB is output, the communication line holds the MSB state.

In synchronous mode, the SCI receives data in synchronization with the rising edge of the serial clock.

Data Transfer Format

A fixed 8-bit data format is used.

No parity or multiprocessor bits are added.

Either an internal clock generated by the built-in baud rate generator or an external serial clock input at the SCK pin can be selected, according to the setting of the C/$\overline{A}$ bit in SMR and the CKE1 and CKE0 bits in SCR. For details of SCI clock source selection, see table 14.9.

When the SCI is operated on an internal clock, the serial clock is output from the SCK pin.

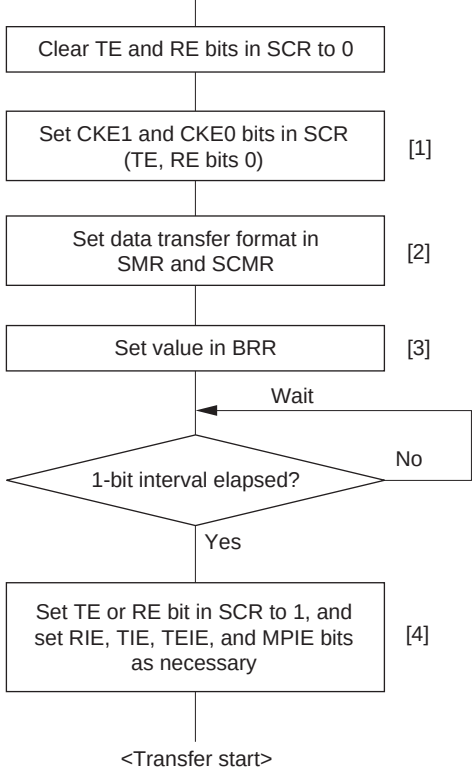
Eight serial clock pulses are output in the transfer of one character, and when no transfer is performed the clock is fixed high. When only receive operations are performed, however, the serial clock is output until an overrun error occurs or the RE bit is cleared to 0. To perform receive operations in units of one character, an external clock should be selected as the clock source.

Data Transfer Operations

SCI initialization (synchronous mode): Before transmitting or receiving data, first 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 cleared to 0 before making the change using the following procedure. When the TE bit is cleared to 0, the TDRE flag is set to 1 and TSR is initialized. Note that clearing the RE bit to 0 does not change the contents of the RDRF, PER, FER, and ORER flags, or the contents of RDR.

Figure 14.15 shows a sample SCI initialization flowchart.



and RE, to 0.

[2] Set the data transfer format in SMR and SCMR.

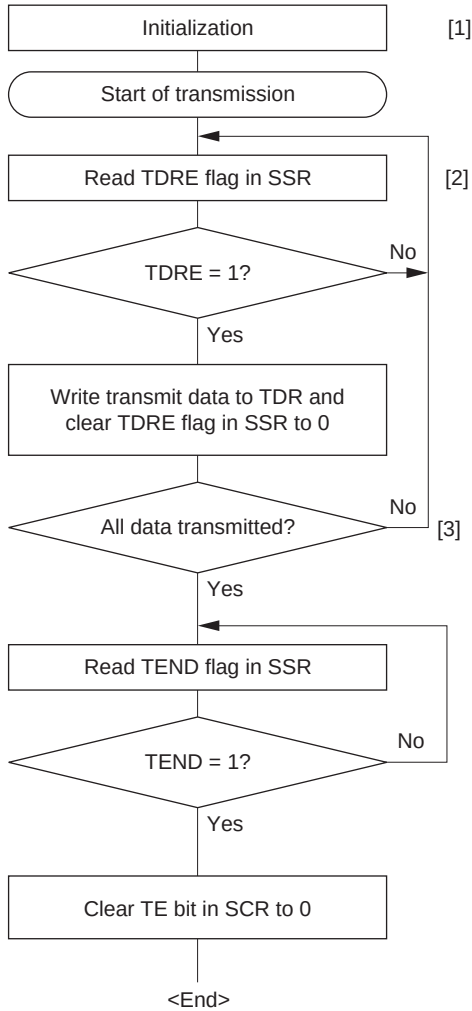
[3] Write a value corresponding to the bit rate to BRR. (Not necessary if an external clock is used.)

[4] Wait at least one bit interval, then set the TE bit or RE bit in SCR to 1. Also set the RIE, TIE, TEIE, and MPIE bits as necessary. Setting the TE or RE bit enables the TxD or RxD pin to be used.

Note: In simultaneous transmit and receive operations, the TE and RE bits should both be cleared to 0 or set to 1 simultaneously.

Figure 14.15 Sample SCI Initialization Flowchart

The following procedure should be used for serial data transmission.



[1] SCI initialization:
The TxD pin is automatically designated as the transmit data output pin.

[2] SCI status check and transmit data write:
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR and clear the TDRE flag to 0.

[3] Serial transmission continuation procedure:
To continue serial transmission, be sure to read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and then clear the TDRE flag to 0.
Checking and clearing of the TDRE flag is automatic when the DMAC or DTC is activated by a transmit-data-empty interrupt (TXI) request and data is written to TDR.

Figure 14.16 Sample Serial Transmission Flowchart

[1] The SCI monitors the TDRE flag in SSR, and if is 0, recognizes that data has been written to TDR, and transfers the data from TDR to TSR.

[2] After transferring data from TDR to TSR, the SCI sets the TDRE flag to 1 and starts transmission. If the TIE bit is set to 1 at this time, a transmit-data-empty interrupt (TXI) is generated.

When clock output mode has been set, the SCI outputs 8 serial clock pulses. When use of an 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 ending with 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 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 MSB (bit 7) is sent, and the 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.

Figure 14.17 shows an example of SCI operation in transmission.

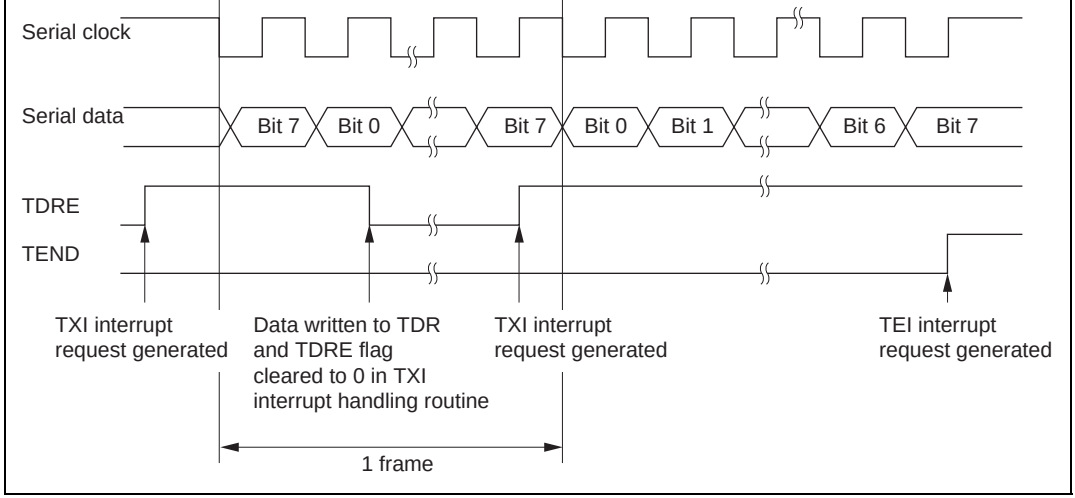


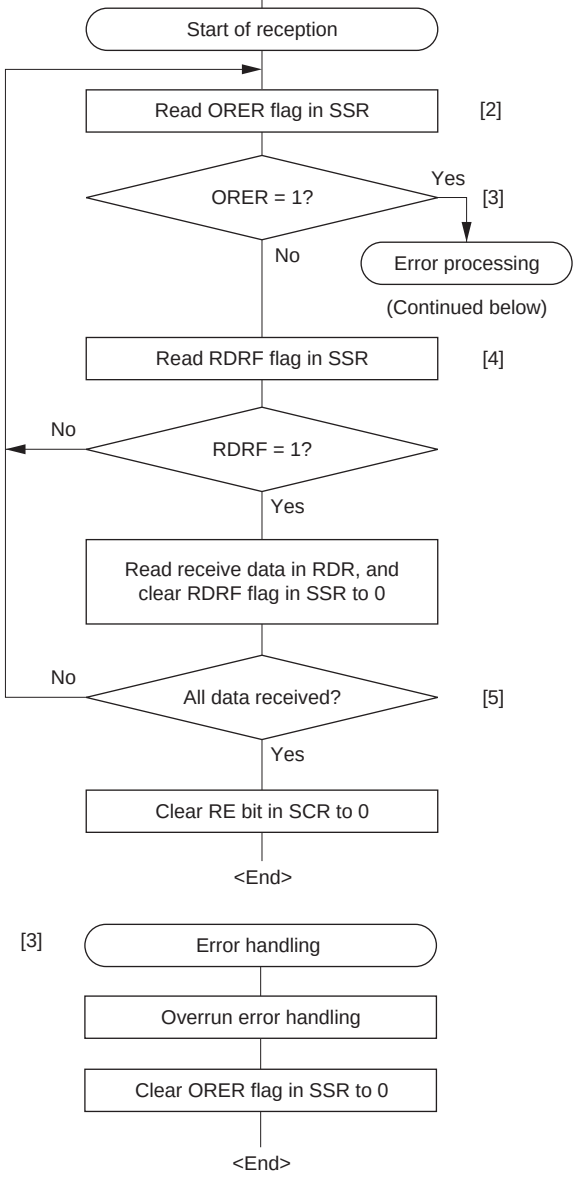
Figure 14.17 Example of SCI Transmit Operation

Serial data reception (synchronous mode): Figure 14.18 shows a sample flowchart for serial reception.

The following procedure should be used for serial data reception.

When changing the operating mode from asynchronous to synchronous, be sure to check 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 nor receive operations will be possible.



designated as the receive data input pin.

[2] [3] Receive error handling:
If a receive error occurs, read the ORER flag in SSR, and after performing the appropriate error handling, clear the ORER flag to 0. Transfer cannot be resumed if the ORER flag is set to 1.

[4] SCI status check and receive data read:
Read SSR and check that the RDRF flag is set to 1, then read the receive data in RDR and clear the RDRF flag to 0. Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial reception continuation procedure:
To continue serial reception, before the MSB (bit 7) of the current frame is received, finish reading the RDRF flag, reading RDR, and clearing the RDRF flag to 0. The RDRF flag is cleared automatically when the DMAC or DTC is activated by a receive-data-full interrupt (RXI) request and the RDR value is read.

Figure 14.18 Sample Serial Reception Flowchart

[1] The SCI performs internal initialization in synchronization with serial clock input or output.

[2] The received data is stored in RSR in LSB-to-MSB order.

After reception, the SCI checks whether the RDRF flag is 0 and the receive data can be transferred from RSR to RDR.

If this check is passed, the RDRF flag is set to 1, and the receive data is stored in RDR. If a receive error is detected in the error check, the operation is as shown in table 14.11.

Neither transmit nor receive operations can be performed subsequently when a receive error 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-full interrupt (RXI) request is generated.

Also, if the RIE bit in SCR is set to 1 when the ORER flag changes to 1, a receive-error interrupt (ERI) request is generated.

Figure 14.19 shows an example of SCI operation in reception.

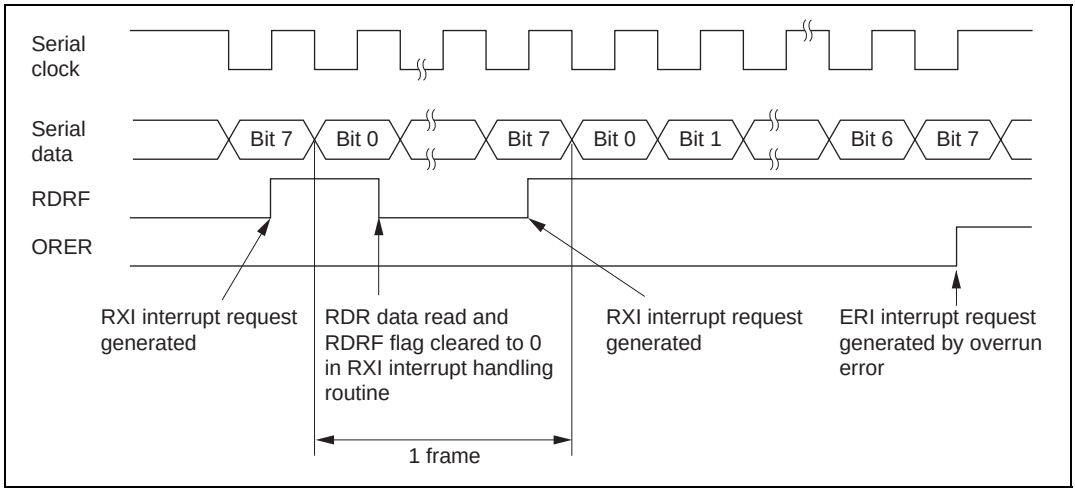
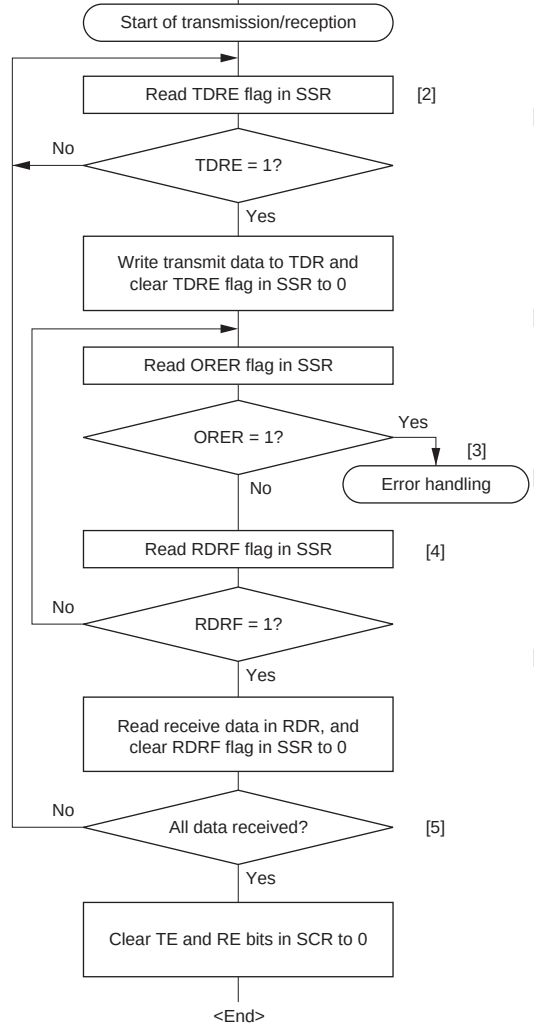


Figure 14.19 Example of SCI Receive Operation

Simultaneous serial data transmission and reception (synchronous mode): Figure 14.20 shows a sample flowchart for simultaneous serial transmit and receive operations.

The following procedure should be used for simultaneous serial data transmit and receive operations.



Note: When switching from transmit or receive operation to simultaneous transmit and receive operations, first clear the TE and RE bits to 0, then set both these bits to 1 simultaneously.

RxD pin is designated as the receive data input pin, enabling simultaneous transmit and receive operations.

[2] SCI status check and transmit data write:
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR and clear the TDRE flag to 0.
Transition of the TDRE flag from 0 to 1 can also be identified by a TXI interrupt.

[3] Receive error handling:
If a receive error occurs, read the ORER flag in SSR, and after performing the appropriate error handling, clear the ORER flag to 0. Transmission/reception cannot be resumed if the ORER flag is set to 1.

[4] SCI status check and receive data read:
Read SSR and check that the RDRF flag is set to 1, then read the receive data in RDR and clear the RDRF flag to 0. Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial transmission/reception continuation procedure:
To continue serial transmission/reception, before the MSB (bit 7) of the current frame is received, finish reading the RDRF flag, reading RDR, and clearing the RDRF flag to 0. Also, before the MSB (bit 7) of the current frame is transmitted, read 1 from the TDRE flag to confirm that writing is possible. Then write data to TDR and clear the TDRE flag to 0.
Checking and clearing of the TDRE flag is automatic when the DMAC or DTC is activated by a transmit-data-empty interrupt (TXI) request and data is written to TDR. Also, the RDRF flag is cleared automatically when the DMAC or DTC is activated by a receive-data-full interrupt (RXI) request and the RDR value is read.

Figure 14.20 Sample Flowchart of Simultaneous Serial Transmit and Receive Operations

The SCI has four interrupt sources: the transmit-end interrupt (TEI) request, receive-error interrupt (ERI) request, receive-data-full interrupt (RXI) request, and transmit-data-empty interrupt (TXI) request. Table 14.12 shows the interrupt sources and their relative priorities. Individual interrupt sources can be enabled or disabled with the TIE, RIE, and TEIE bits in the SCR. Each kind of interrupt request is sent to the interrupt controller independently.

When the TDRE flag in SSR is set to 1, a TXI interrupt request is generated. When the TEND flag in SSR is set to 1, a TEI interrupt request is generated. A TXI interrupt can activate the DMAC or DTC to perform data transfer. The TDRE flag is cleared to 0 automatically when data transfer is performed by the DMAC or DTC. The DMAC and 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 the ORER, PER, or FER flag in SSR is set to 1, an ERI interrupt request is generated. An RXI interrupt can activate the DMAC or DTC to perform data transfer. The RDRF flag is cleared to 0 automatically when data transfer is performed by the DMAC or DTC. The DMAC and DTC cannot be activated by an ERI interrupt request.

Also note that the DMAC cannot be activated by an SCI channel 2 interrupt.

Channel	Interrupt Source	Description	DTC Activation	DMAC Activation	Priority*
0	ERI	Interrupt due to receive error (ORER, FER, or PER)	Not possible	Not possible	High
	RXI	Interrupt due to receive data full state (RDRF)	Possible	Possible	
	TXI	Interrupt due to transmit data empty state (TDRE)	Possible	Possible	
	TEI	Interrupt due to transmission end (TEND)	Not possible	Not possible	
1	ERI	Interrupt due to receive error (ORER, FER, or PER)	Not possible	Not possible	
	RXI	Interrupt due to receive data full state (RDRF)	Possible	Possible	
	TXI	Interrupt due to transmit data empty state (TDRE)	Possible	Possible	
	TEI	Interrupt due to transmission end (TEND)	Not possible	Not possible	
2	ERI	Interrupt due to receive error (ORER, FER, or PER)	Not possible	Not possible	
	RXI	Interrupt due to receive data full state (RDRF)	Possible	Not possible	
	TXI	Interrupt due to transmit data empty state (TDRE)	Possible	Not possible	
	TEI	Interrupt due to transmission end (TEND)	Not possible	Not possible	Low

Note: * This table shows the initial state immediate after a reset. Relative priorities among channels can be changed by the interrupt controller.

A TEI interrupt is requested when the TEND flag is set to 1 while the TEIE bit is set to 1. The TEND flag is cleared at the same time as the TDRE flag. Consequently, if a TEI interrupt and a TXI interrupt are requested simultaneously, the TXI interrupt may be accepted first, with the result that the TDRE and TEND flags are cleared. Note that the TEI interrupt will not be accepted in this case.

The following points should be noted when using the SCI.

Relation between Writes to TDR and the TDRE Flag: The TDRE flag in SSR is a status flag that indicates that transmit data has been transferred from TDR to TSR. When the SCI transfers data from TDR to TSR, the TDRE flag is set to 1.

Data can be written to TDR regardless of the state of the TDRE flag. However, if new data is written to TDR when the TDRE flag is cleared to 0, the data stored in TDR will be lost since it has not yet been transferred to TSR. It is therefore essential to check that the TDRE flag is set to 1 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 is as shown in table 14.13. If there is an overrun error, data is not transferred from RSR to RDR, and the receive data is lost.

Table 14.13 State of SSR Status Flags and Transfer of Receive Data

SSR Status Flags				Receive Data Transfer	
RDRF	**ORER**	**FER**	**PER**	**from RSR to RDR**	**Receive Error Status**
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 + framing error
1	1	0	1	X	Overrun error + parity error
0	0	1	1	○	Framing error + parity error
1	1	1	1	X	Overrun error + framing error + parity error

Notes: ○: Receive data is transferred from RSR to RDR.
X: Receive data is not transferred from RSR to RDR.

the input from the RxD pin becomes all 0s, and so the FER flag is set, and the parity error flag (PER) may also be set.

Note that, since the SCI continues the receive operation after receiving a break, even if the FER flag is cleared to 0, it will be set to 1 again.

Sending a Break (Asynchronous Mode Only): The TxD pin has a dual function as an I/O port whose direction (input or output) is determined by DR and DDR. This can be used to send a break.

Between serial transmission initialization and setting of the TE bit to 1, the mark state is replaced by the value of DR (the pin does not function as the TxD pin until the TE bit is set to 1). Therefore, DDR and DR for the port corresponding to the TxD pin should first be set to 1.

To send a break during serial transmission, first clear DR to 0, then clear the TE bit to 0.

When the TE bit is cleared to 0, the transmitter is initialized regardless of the current transmission state, the TxD pin becomes an I/O port, and 0 is output from the TxD pin.

Receive Error Flags and Transmit Operations (Synchronous Mode Only): Transmission cannot be started when a receive error flag (ORER, PER, or FER) is set to 1, even if the TDRE flag is cleared to 0. Be sure to clear the receive error flags to 0 before starting transmission.

Note also that receive error flags cannot be cleared to 0 even if the RE bit is cleared to 0.

Receive Data Sampling Timing and Receive Margin in Asynchronous Mode: In asynchronous mode, the SCI operates on a base clock with a frequency of 16 times the transfer rate.

In reception, the SCI samples the falling edge of the start bit using the base clock, and performs internal synchronization. Receive data is latched internally at the rising edge of the 8th pulse of the base clock. This is illustrated in figure 14.21.

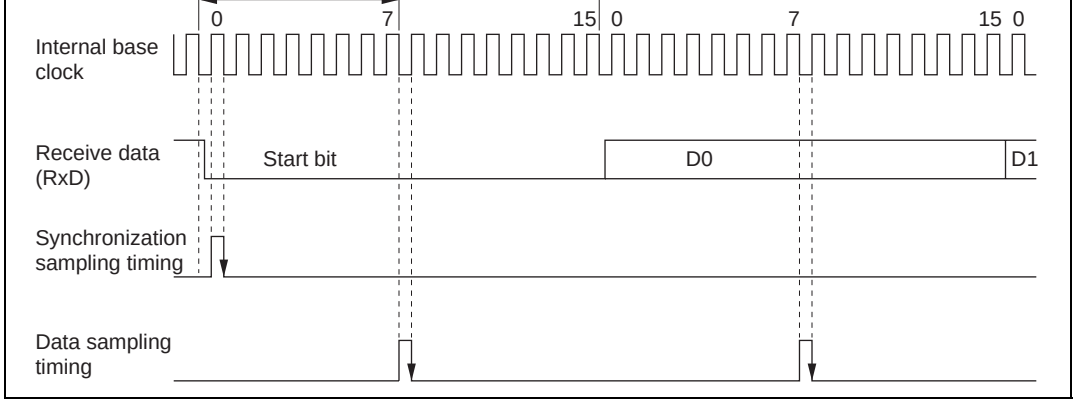


Figure 14.21 Receive Data Sampling Timing in Asynchronous Mode

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

$$M = \left| \left(0.5 - \frac{1}{2N}\right) - (L - 0.5)\,F - \frac{|D - 0.5|}{N}\,(1 + F) \right| \times 100\% \quad \text{... Formula (1)}$$

Where M : Receive 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 receive margin of 46.875% is given by formula (2) below.

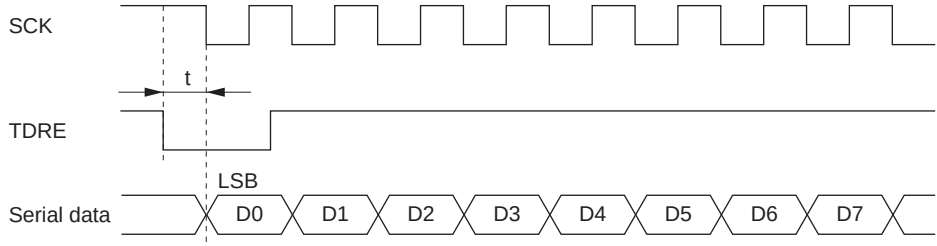
When D = 0.5 and F = 0,

$$M = \left(0.5 - \frac{1}{2 \times 16}\right) \times 100\%$$

$$= 46.875\% \quad \text{... Formula (2)}$$

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

- When an external clock source is used as the serial clock, the transmit clock should not be input until at least 5 ϕ clock cycles after TDR is updated by the DMAC or DTC. Misoperation may occur if the transmit clock is input within 4 ϕ clocks after TDR is updated. (Figure 14.22)
- When RDR is read by the DMAC or DTC, be sure to set the activation source to the relevant SCI receive-data-full interrupt (RXI).



Note: When operating on an external clock, set t > 4 clocks.

Figure 14.22 Example of Synchronous Transmission Using DTC

- Transmission

  Operation should be stopped (by clearing TE, TIE, and TEIE to 0) before making a module stop mode or software standby mode transition. TSR, TDR, and SSR are reset. The output pin states in module stop mode or software standby mode depend on the port settings, and becomes high-level output after the relevant mode is cleared. If a transition is made during transmission, the data being transmitted will be undefined. When transmitting without changing the transmit mode after the relevant mode is cleared, transmission can be started by setting TE to 1 again, and performing the following sequence: SSR read → TDR write → TDRE clearance. To transmit with a different transmit mode after clearing the relevant mode, the procedure must be started again from initialization. Figure 14.23 shows a sample flowchart for mode transition during transmission. Port pin states are shown in figures 14.24 and 14.25.

  Operation should also be stopped (by clearing TE, TIE, and TEIE to 0) before making a transition from transmission by DTC transfer to module stop mode or software standby mode transition. To perform transmission with the DTC after the relevant mode is cleared, setting TE and TIE to 1 will set the TXI flag and start DTC transmission.

- Reception

  Receive operation should be stopped (by clearing RE to 0) before making a module stop mode or software standby mode transition. RSR, RDR, and SSR are reset. If a transition is made without stopping operation, the data being received will be invalid.

  To continue receiving without changing the reception mode after the relevant mode is cleared, set RE to 1 before starting reception. To receive with a different receive mode, the procedure must be started again from initialization.

  Figure 14.26 shows a sample flowchart for mode transition during reception.

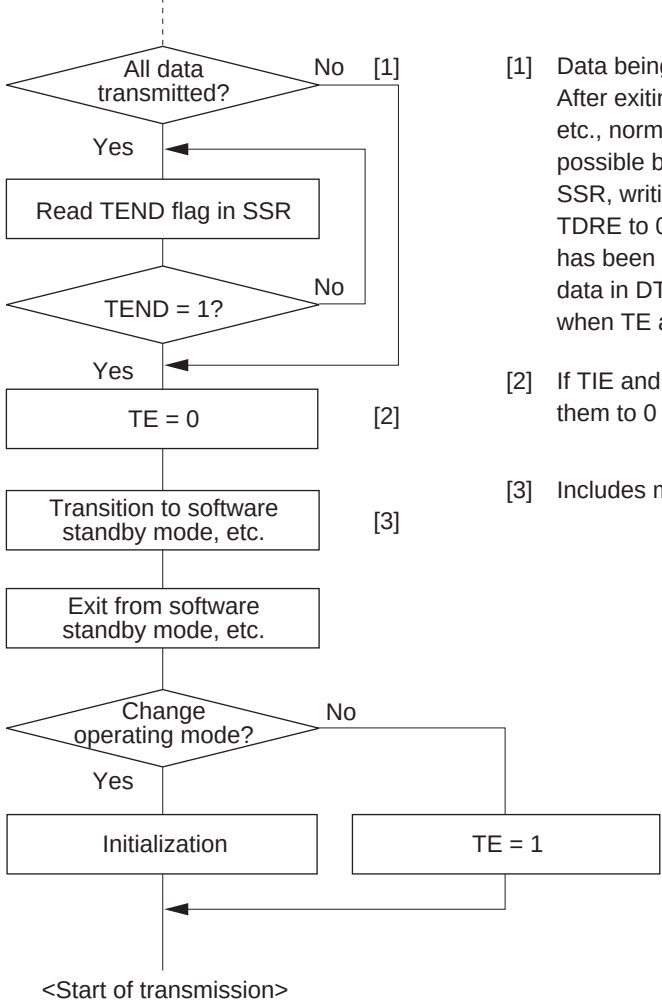


Figure 14.23 Sample Flowchart for Mode Transition during Transmission

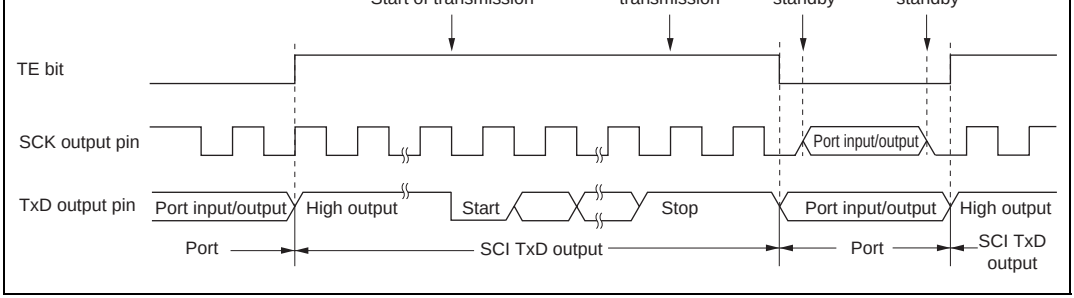


Figure 14.24 Asynchronous Transmission Using Internal Clock

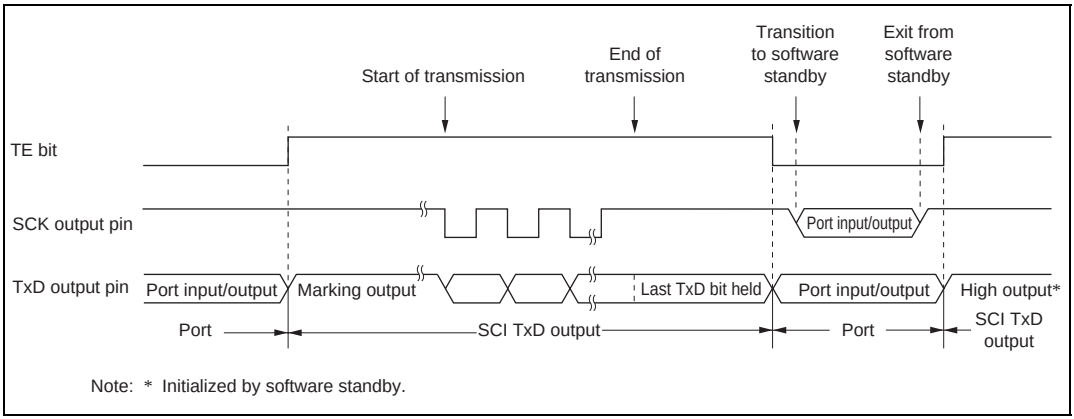


Figure 14.25 Synchronous Transmission Using Internal Clock

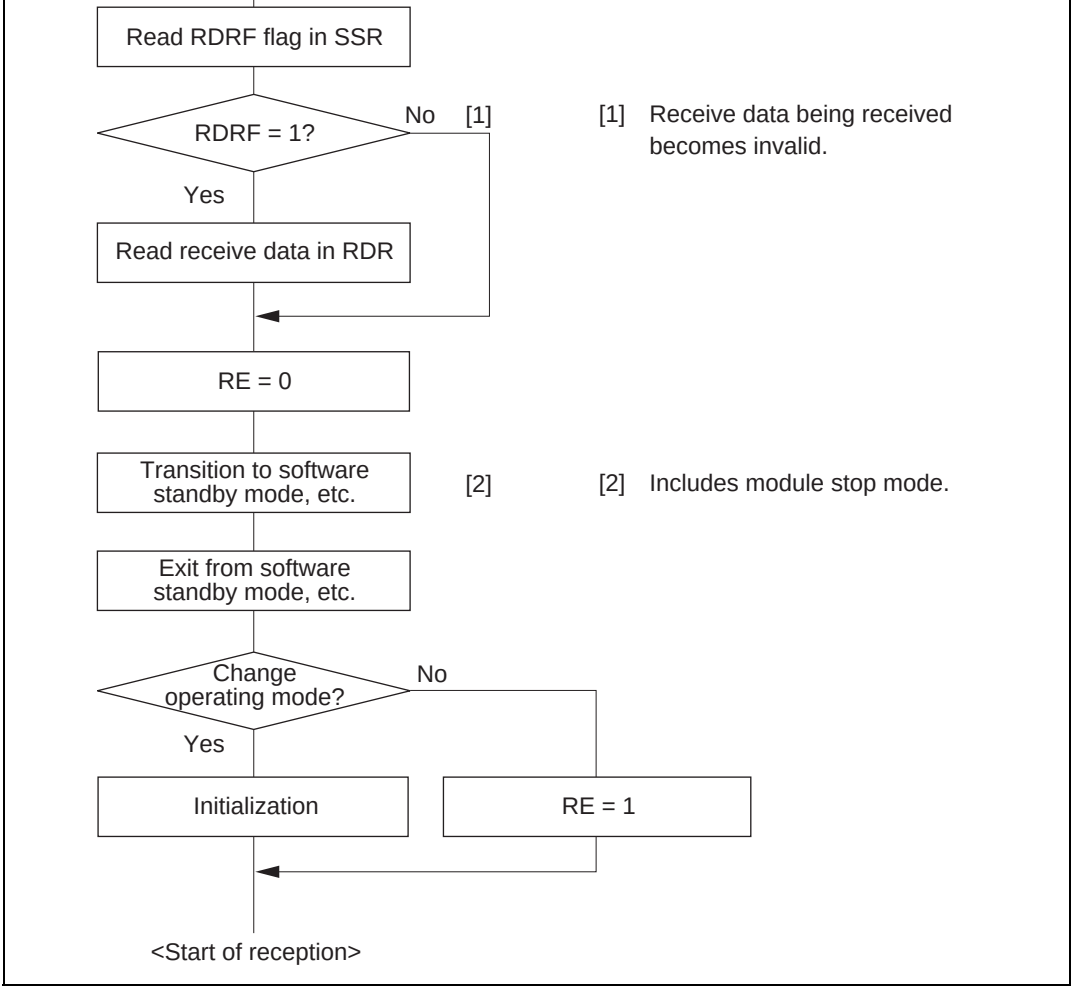


Figure 14.26 Sample Flowchart for Mode Transition during Reception

15.1 Overview

The SCI supports an IC card (smart card) interface conforming to ISO/IEC 7816-3 (identification card) as a serial communication interface extension function.

Switching between the normal serial communication interface and the smart card interface is carried out by means of a register setting.

15.1.1 Features

Features of the smart card interface supported by the chip is 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
- Built-in baud rate generator allows any bit rate to be selected
- Three interrupt sources
  - Three interrupt sources (transmit-data-empty, receive-data-full, and transmit/receive-error) that can issue requests independently
  - The transmit-data-empty and receive-data-full interrupts can activate the DMA controller (DMAC) or data transfer controller (DTC) to execute data transfer

Figure 15.1 shows a block diagram of the smart card interface.

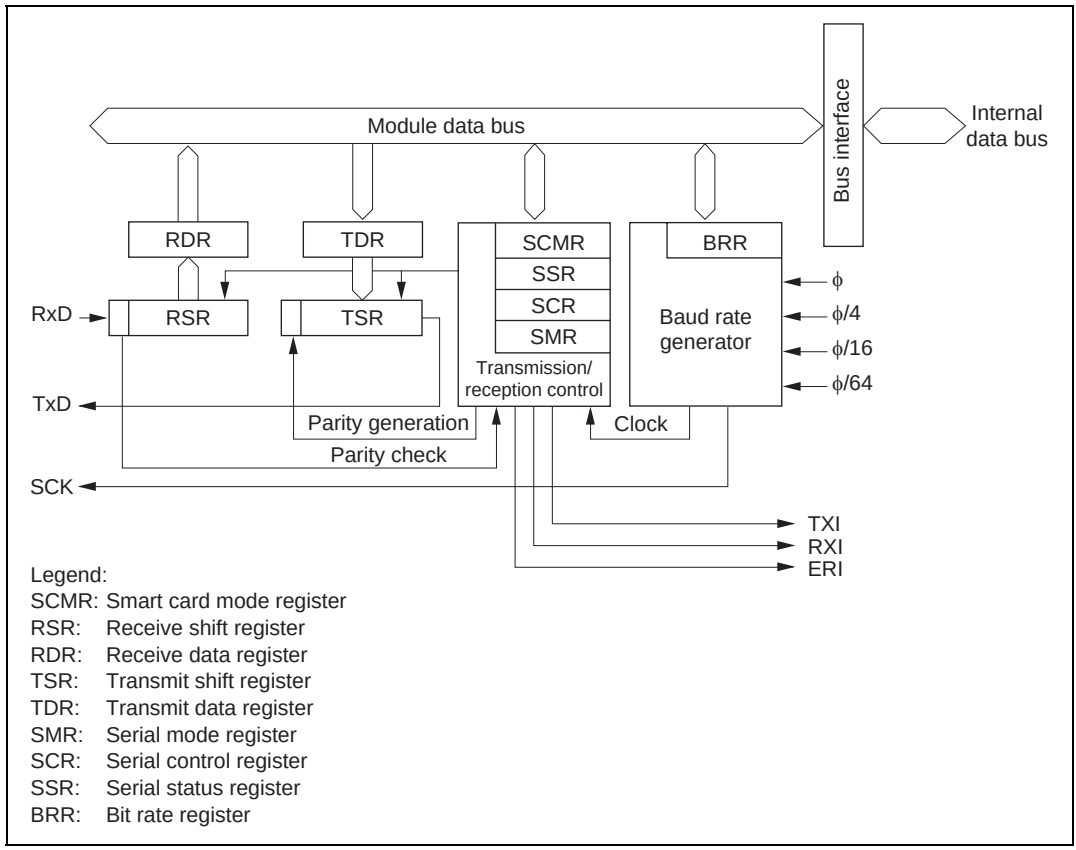


Figure 15.1 Block Diagram of Smart Card Interface

Table 15.1 shows the smart card interface pin configuration.

Table 15.1 Smart Card Interface Pins

Channel	Pin Name	Symbol	I/O	Function
0	Serial clock pin 0	SCK0	I/O	SCI0 clock input/output
	Receive data pin 0	RxD0	Input	SCI0 receive data input
	Transmit data pin 0	TxD0	Output	SCI0 transmit data output
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 output
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 output

Table 15.2 shows the registers used by the smart card interface. Details of SMR, BRR, SCR, TDR, RDR, and MSTPCR are the same as for the normal SCI function: see the register descriptions in section 14, Serial Communication Interface (SCI).

Table 15.2 Smart Card Interface Registers

Channel	Name	Abbreviation	R/W	Initial Value	Address*1
0	Serial mode register 0	SMR0	R/W	H'00	H'FF78
	Bit rate register 0	BRR0	R/W	H'FF	H'FF79
	Serial control register 0	SCR0	R/W	H'00	H'FF7A
	Transmit data register 0	TDR0	R/W	H'FF	H'FF7B
	Serial status register 0	SSR0	R/(W)*2	H'84	H'FF7C
	Receive data register 0	RDR0	R	H'00	H'FF7D
	Smart card mode register 0	SCMR0	R/W	H'F2	H'FF7E
1	Serial mode register 1	SMR1	R/W	H'00	H'FF80
	Bit rate register 1	BRR1	R/W	H'FF	H'FF81
	Serial control register 1	SCR1	R/W	H'00	H'FF82
	Transmit data register 1	TDR1	R/W	H'FF	H'FF83
	Serial status register 1	SSR1	R/(W) *2	H'84	H'FF84
	Receive data register 1	RDR1	R	H'00	H'FF85
	Smart card mode register 1	SCMR1	R/W	H'F2	H'FF86
2	Serial mode register 2	SMR2	R/W	H'00	H'FF88
	Bit rate register 2	BRR2	R/W	H'FF	H'FF89
	Serial control register 2	SCR2	R/W	H'00	H'FF8A
	Transmit data register 2	TDR2	R/W	H'FF	H'FF8B
	Serial status register 2	SSR2	R/(W) *2	H'84	H'FF8C
	Receive data register 2	RDR2	R	H'00	H'FF8D
	Smart card mode register 2	SCMR2	R/W	H'F2	H'FF8E
All	Module stop control register	MSTPCR	R/W	H'3FFF	H'FF3C

Notes: 1. Lower 16 bits of the address.
2. Can only be written with 0 for flag clearing.

Registers added with the smart card interface and bits for which the function changes are described here.

15.2.1 Smart Card Mode Register (SCMR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	SDIR	SINV	—	SMIF
Initial value	:	1	1	1	1	0	0	1	0
R/W	:	—	—	—	—	R/W	R/W	—	R/W

SCMR is an 8-bit readable/writable register that selects the smart card interface function.

SCMR is initialized to H'F2 by a reset and in hardware standby mode. In software standby mode and module stop mode it retains its previous state.

Bits 7 to 4—Reserved: These bits cannot be modified and are always read as 1.

Bit 3—Smart Card Data Transfer Direction (SDIR): Selects the serial/parallel conversion format.

Bit 3 SDIR	Description	
0	TDR contents are transmitted LSB-first Receive data is stored in RDR LSB-first	(Initial value)
1	TDR contents are transmitted MSB-first Receive data is stored in RDR MSB-first	

The SINV bit does not affect the logic level of the parity bit. For parity-related setting procedures, see section 15.3.4, Register Settings.

Bit 2 SINV	Description	
0	TDR contents are transmitted as they are Receive data is stored as it is in RDR	(Initial value)
1	TDR contents are inverted before being transmitted Receive data is stored in inverted form in RDR	

Bit 1—Reserved: Read-only bit, always read as 1.

Bit 0—Smart Card Interface Mode Select (SMIF): Enables or disables the smart card interface function.

Bit 0 SMIF	Description	
0	Smart card interface function is disabled	(Initial value)
1	Smart card interface function is enabled	

Bit	:	7	6	5	4	3	2	1	0
		TDRE	RDRF	ORER	ERS	PER	TEND	MPB	MPBT
Initial value	:	1	0	0	0	0	1	0	0
R/W	:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/W

Note: * Only 0 can be written to bits 7 to 3, to clear these flags.

Bit 4 of SSR has a different function in smart card interface mode. Coupled with this, the setting 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 14.2.7, Serial Status Register (SSR).

Bit 4—Error Signal Status (ERS): In smart card interface mode, bit 4 indicates the status of the error signal sent back from the receiving end in transmission. Framing errors are not detected in smart card interface mode.

Bit 4 ERS	Description
0	Indicates data received normally with no error signal (Initial value) [Clearing conditions] • Upon reset, and in standby mode or module stop mode • When 0 is written to ERS after reading ERS = 1
1	Indicates an error signal was sent showing detection of a parity error at the receiving side [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.

However, the setting conditions for the TEND bit, are as shown below.

Bit 2 TEND	Description
0	Indicates transfer in progress [Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DMAC or DTC is activated by a TXI interrupt and writes data to TDR
1	Indicates transfer complete (Initial value) [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 transmission of a 1-byte serial character when GM = 0 and BLK = 0 • When TDRE = 1 and ERS = 0 (normal transmission) 1.5 etu after transmission of a 1-byte serial character when GM = 0 and BLK = 1 • When TDRE = 1 and ERS = 0 (normal transmission) 1.0 etu after transmission of a 1-byte serial character when GM = 1 and BLK = 0 • When TDRE = 1 and ERS = 0 (normal transmission) 1.0 etu after transmission of a 1-byte serial character when GM = 1 and BLK = 1

Note: etu: Elementary time unit (time for transfer of 1 bit)

Bit	:	7	6	5	4	3	2	1	0
		GM	BLK	PE	O/$\overline{E}$	BCP1	BCP0	CKS1	CKS0
Initial value	:	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

Note: * When using the smart card interface, set bit 5 to 1.

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, this bit is set to 1, the timing of setting of the TEND flag that indicates transmission completion is advanced, and clock output control mode addition is performed. The contents of the clock output control 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 (Initial value) • 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)

Bit 6 BLK	Description
0	Normal smart card interface mode operation (Initial value) • Error signal transmission/detection and automatic data retransmission performed • 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 performed • TXI interrupt generated by TDRE flag • TEND flag set 11.5 etu after start of transmission (11.0 etu in GSM mode)

Note: etu: Elementary time unit (time for transfer of 1 bit)

Bits 3 and 2—Base Clock Pulse 1 and 2 (BCP1, BCP0): These bits specify the number of base clock periods in a 1-bit transfer interval on the smart card interface.

Bit 3 BCP1	Bit 2 BCP0	Description	
0	0	32 clock periods	(Initial value)
	1	64 clock periods	
1	0	372 clock periods	
	1	256 clock periods	

Bits 5, 4, 1, and 0—Operate in the same way as for the normal SCI. For details, see section 14.2.5, Serial Mode Register (SMR).

Bit	:	7	6	5	4	3	2	1	0
		TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
Initial value	:	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

In smart card interface mode, the function of bits 1 and 0 of SCR changes when bit 7 of the serial 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 14.2.6, Serial Control Register (SCR).

Bits 1 and 0—Clock Enable 1 and 0 (CKE1, CKE0): These bits are used to select the SCI clock 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 enabling and disabling, the clock output can be specified as being fixed high or low.

SCMR	SMR	SCR Setting		
SMIF	**GM**	**CKE1**	**CKE0**	**SCK Pin Function**
0	See the SCI specification			
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 output fixed low
1	1	0	1	Outputs clock as SCK output pin
1	1	1	0	Operates as SCK output pin, with output fixed high
1	1	1	1	Outputs clock as SCK output pin

15.3.1 Overview

The main functions of the smart card interface are as follows.

- One frame consists of 8-bit data plus a parity bit.
- In transmission, a guard time of at least 2 etu (1 etu in block transfer mode) (elementary time unit: the time for transfer of 1 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 one etu period, 10.5 etu after the start bit. (This does not apply to block transfer mode.)
- If the error signal is sampled during transmission, the same data is transmitted automatically after the elapse of 2 etu or longer. (This does not apply to block transfer mode.)
- Only asynchronous communication is supported; there is no synchronous communication function.

15.3.2 Pin Connections

Figure 15.2 shows a schematic diagram of smart card interface related pin connections.

In communication with an IC card, since both transmission and reception are carried out on a single data communication line, the chip's TxD pin and RxD pin should both be connected to the line, as shown in the figure. The data communication 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 pin output is input to the CLK pin of the IC card. No connection is needed if the IC card uses an internal clock.

Chip port output is used as the reset signal.

Other pins must normally be connected to the power supply or ground.

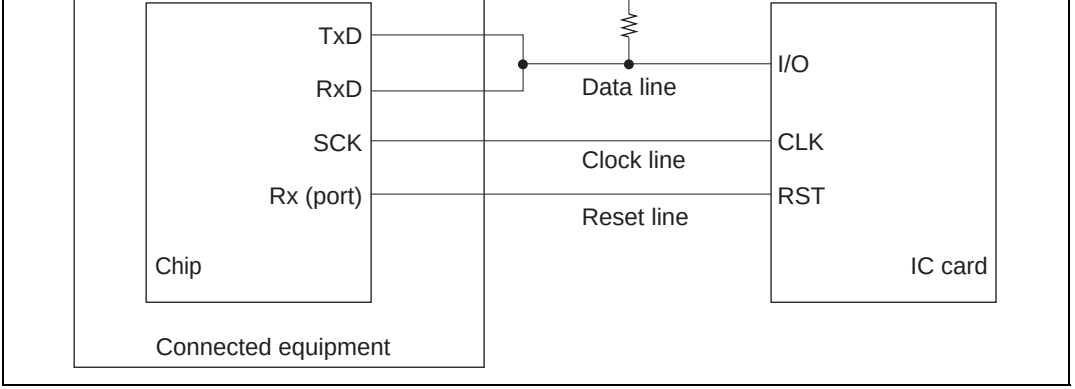


Figure 15.2 Schematic Diagram of Smart Card Interface Pin Connections

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.

Normal Transfer Mode: Figure 15.3 shows the smart card interface data format in the normal transfer mode. In reception in this mode, a parity check is carried out on each frame. If an error is detected an error signal is sent back to the transmitting end, and retransmission of the data is requested. If an error signal is sampled during transmission, the same data is retransmitted.

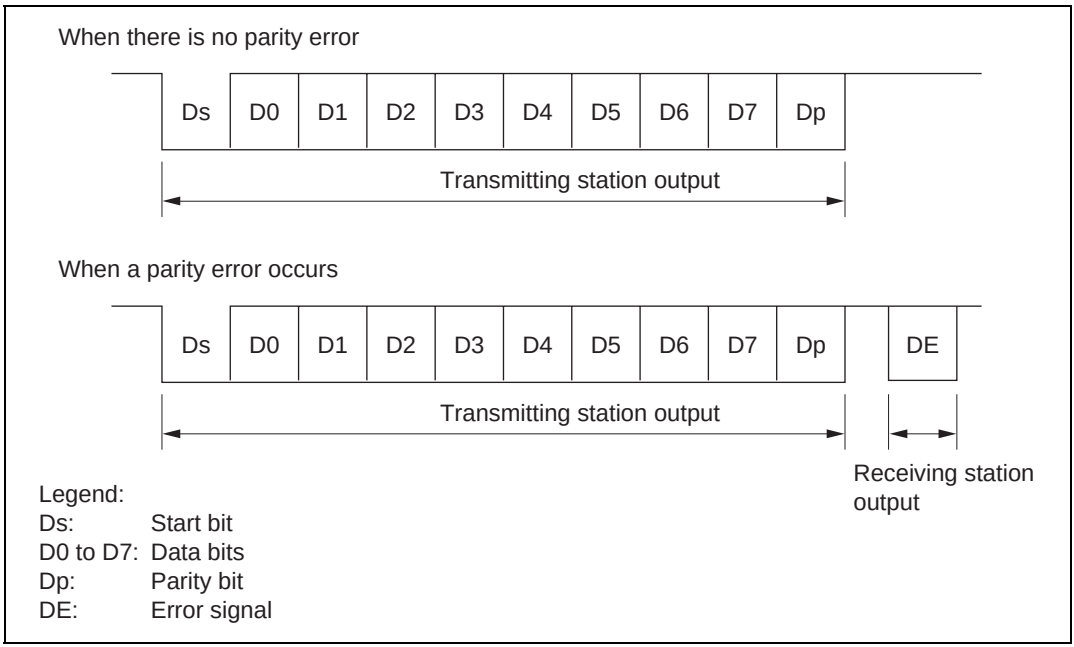


Figure 15.3 Smart Card Interface Data Format

[1] When the data line is not in use it is in the high-impedance state, and is fixed high with a pull-up resistor.

[2] The transmitting station starts transfer of one frame of data. The data frame starts with a start 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 state. The data 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 waits for reception of the next data.

If a parity error occurs, however, the receiving station outputs an error signal (DE, low-level) to request retransmission of the data. After outputting the error signal for the prescribed length of time, the receiving station places the signal line in the high-impedance state again. The 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 the next data frame.

If it does receive an error signal, however, it returns to step [2] and retransmits the data in which the error occurred.

Block Transfer Mode: The operation sequence in block transfer mode is as follows.

[1] When the data line is not in use it is in the high-impedance state, and is fixed high with a pull-up resistor.

[2] The transmitting station starts transfer of one frame of data. The data frame starts with a start 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 state. The data line is pulled high with a pull-up resistor.

[4] The receiving station carries out a parity check, but does not output an error signal even if an error has occurred. Since subsequent receive operations cannot be carried out if an error occurs, the error flag must be cleared to 0 before the parity bit for the next frame is received.

[5] The transmitting station proceeds to transmit the next data frame.

Table 15.3 shows a bit map of the registers used by the smart card interface.

Bits indicated as 0 or 1 must be set to the value shown. The setting of other bits is described below.

Table 15.3 Smart Card Interface Register Settings

	Bit							
Register	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
SMR	GM	BLK	1	O/$\overline{E}$	BCP1	BCP0	CKS1	CKS0
BRR	BRR7	BRR6	BRR5	BRR4	BRR3	BRR2	BRR1	BRR0
SCR	TIE	RIE	TE	RE	0	0	CKE1*	CKE0
TDR	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1	TDR0
SSR	TDRE	RDRF	ORER	ERS	PER	TEND	0	0
RDR	RDR7	RDR6	RDR5	RDR4	RDR3	RDR2	RDR1	RDR0
SCMR	—	—	—	—	SDIR	SINV	—	SMIF

Notes: — : Unused bit.
* The CKE1 bit must be cleared to 0 when the GM bit in SMR is cleared to 0.

SMR Settings: The GM bit is cleared to 0 in normal smart card interface mode, and set to 1 in GSM mode. The O/$\overline{E}$ bit is cleared to 0 if the IC card is of the direct convention type, and set to 1 if of the inverse convention type.

Bits CKS1 and CKS0 select the clock source of the built-in baud rate generator, and bits BCP1 and BCP0 select the number of base clock cycles during transfer of one bit. For details, see section 15.3.5, Clock.

The BLK bit is cleared to 0 when using the normal smart card interface mode, and set to 1 when using block transfer mode.

BRR Setting: BRR is used to set the bit rate. See section 15.3.5, Clock, for the method of calculating the value to be set.

SCR Settings: The function of the TIE, RIE, TE, and RE bits is the same as for the normal SCI. For details, see section 14, Serial Communication Interface (SCI).

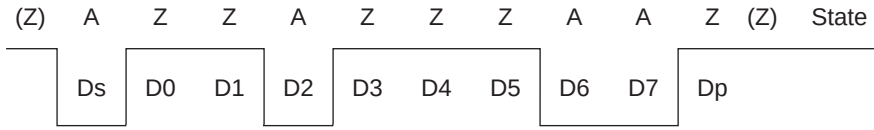
Bits CKE1 and CKE0 specify the clock output. When the GM bit in SMR is cleared to 0, set these bits to B'00 if a clock is not to be output, or to B'01 if a clock is to be output. When the GM bit in SMR is set to 1, clock output is performed. The clock output can also be fixed high or low.

The SINV bit is cleared to 0 if the IC card is of the direct convention type, and set to 1 if of the inverse convention type.

The SMIF bit is set to 1 when the smart card interface is used.

Examples of register settings and the waveform of the start character are shown below for the two types of IC card (direct convention and inverse convention).

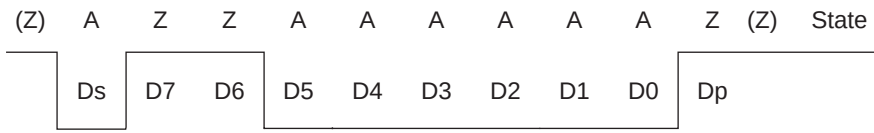
- Direct convention (SDIR = SINV = O/$\overline{E}$ = 0)



  With the direct convention type, the logic 1 level corresponds to state Z and the logic 0 level to state A, and transfer is performed in LSB-first order. The start character data above is H'3B.

  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 logic 0 level to state Z, and transfer is performed in MSB-first order. The start character data above is H'3F.

  The parity bit is 0, corresponding to state Z, since even parity is stipulated for the smart card.

  With the chip, inversion specified by the SINV bit applies only to the data bits, D7 to D0. For parity bit inversion, the O/$\overline{E}$ bit in SMR should be set to odd parity mode (the same applies to both transmission and reception).

Only an internal clock generated by the built-in baud rate generator can be used as the transmit/receive clock for the smart card interface. The bit rate is set with BRR and the CKS1, CKS0, BCP1, and BCP0 bits in SMR. The formula for calculating the bit rate is as shown below. Table 15.5 shows some sample bit rates.

If clock output is selected by setting CKE0 to 1, the clock is output from the SCK pin. The clock 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 (bits/s)
ϕ = Operating frequency (MHz)
n = See table 15.4
S = Number of internal clock cycles in 1-bit period set by bits BCP1 and BCP0

Table 15.4 Correspondence between n and CKS1, CKS0

n	CKS1	CKS0
0	0	0
1		1
2	1	0
3		1

Table 15.5 Examples of Bit Rate B (bits/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**	**20.00**	**25.00**
0	13441	14400	17473	19200	21505	24194	26882	33602
1	6720	7200	8737	9600	10753	12097	13441	16801
2	4480	4800	5824	6400	7168	8065	8961	11201

Note: Bit rates are rounded to the nearest whole number.

smaller error is specified.

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

Table 15.6 Examples of BRR Settings for Bit Rate B (bits/s) (When n = 0 and S = 372)

	ϕ (MHz)															
	7.1424		10.00		10.7136		13.00		14.2848		16.00		18.00		20.00		25.00	
Bits/s	N	Error	N	Error	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	2	6.60	3	12.49

Table 15.7 Maximum Bit Rate at Various Frequencies (Smart Card Interface Mode) (When S = 372)

ϕ (MHz)	Maximum Bit Rate (bits/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
25.00	33602	0	0

The bit rate error is given by the following formula:

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

Initialization: Before transmitting or receiving data, initialize the SCI as described below. Initialization is also necessary when switching from transmit mode to receive mode, or vice versa.

[1] Clear the TE and RE bits in SCR to 0.

[2] Clear the error flags ERS, PER, and ORER in SSR to 0.

[3] Set the GM, BLK, O/$\overline{E}$, BCP1, BCP0, CKS1, and CKS0 bits in SMR, and set the PE bit to 1.

[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 to SCI pins, and are placed in the high-impedance state.

[5] Set the value corresponding to the bit rate in BRR.

[6] Set the CKE1 and CKE0 bits in SCR. Clear the TIE, RIE, TE, RE, MPIE, and TEIE bits to 0.
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 not set the TE bit and RE bit at the same time, except for self-diagnosis.

different from that for the normal SCI. Figure 15.4 shows a flowchart for transmitting, and figure 15.5 shows the relation between a transmit operation and the internal registers.

[1] Perform smart card interface mode initialization as described above in Initialization.

[2] Check that the ERS error flag in SSR is cleared to 0.

[3] Repeat steps [2] and [3] until it can be confirmed that the TEND flag in SSR is set to 1.

[4] Write the transmit data to TDR, clear the TDRE flag to 0, and perform the transmit operation. 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 handling or data transfer by the DMAC or DTC is possible.

If transmission ends and the TEND flag is set to 1 while the TIE bit is set to 1 and interrupt requests are enabled, a transmit-data-empty interrupt (TXI) request will be generated. If an error occurs in transmission and the ERS flag is set to 1 while the RIE bit is set to 1 and interrupt requests are enabled, a transmit/receive-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 TEND flag setting timing is shown in figure 15.6.

If the DMAC or DTC is activated by a TXI request, the number of bytes set in the DMAC or DTC can be transmitted automatically, including automatic retransmission.

For details, see Interrupt Operation (Except Block Transfer Mode) and Data Transfer Operation by DMAC or DTC below.

Note: For details of operation in block transfer mode, see section 14.3.2, Operation in Asynchronous Mode.

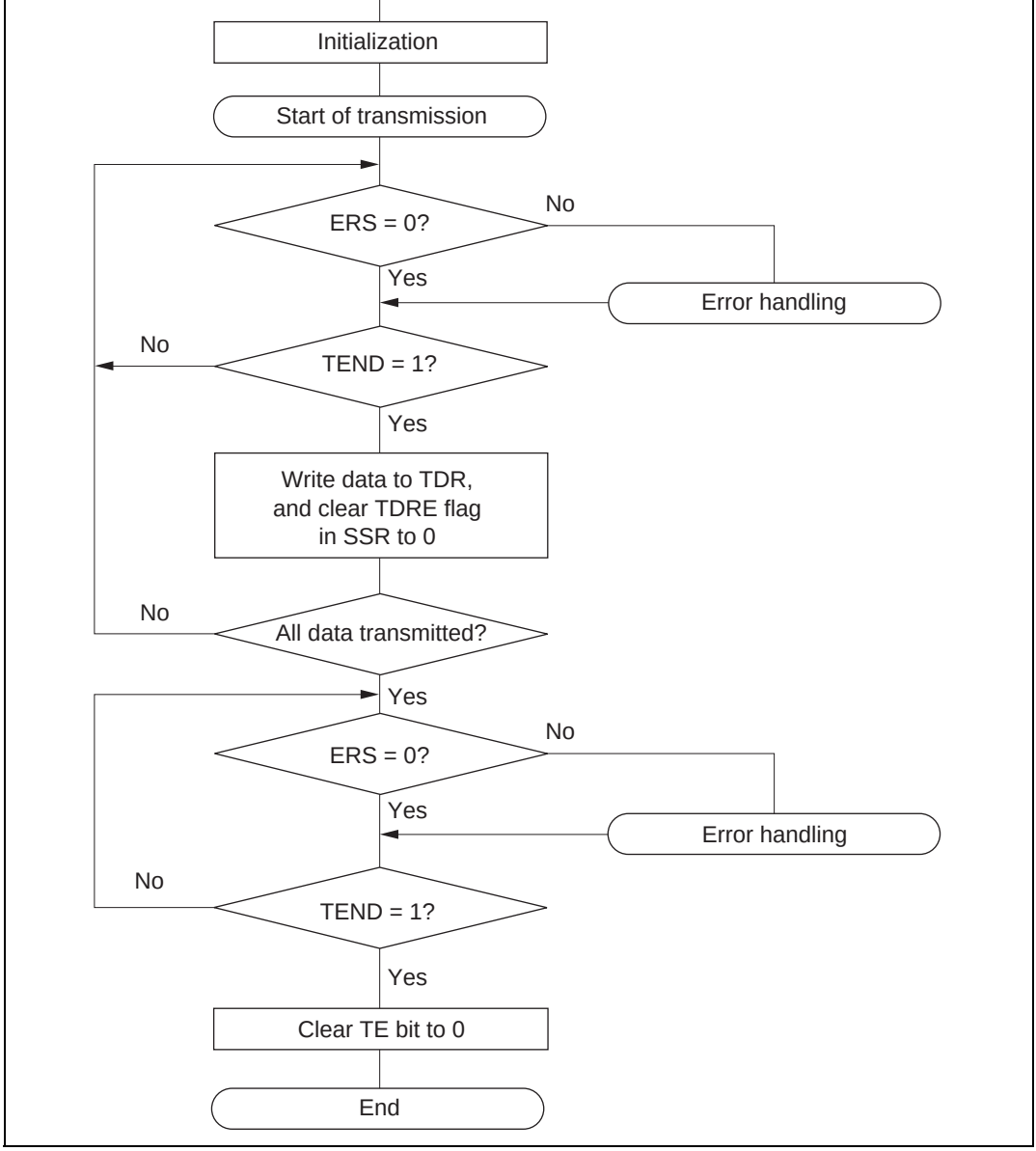
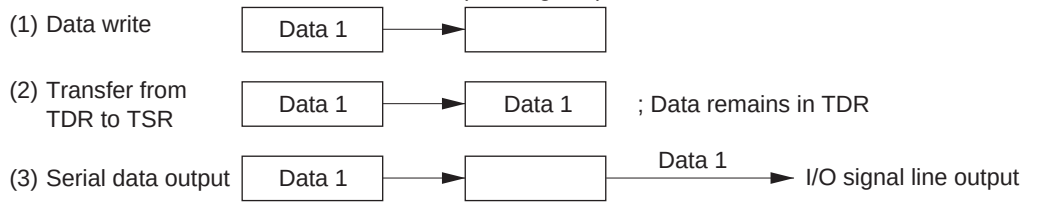


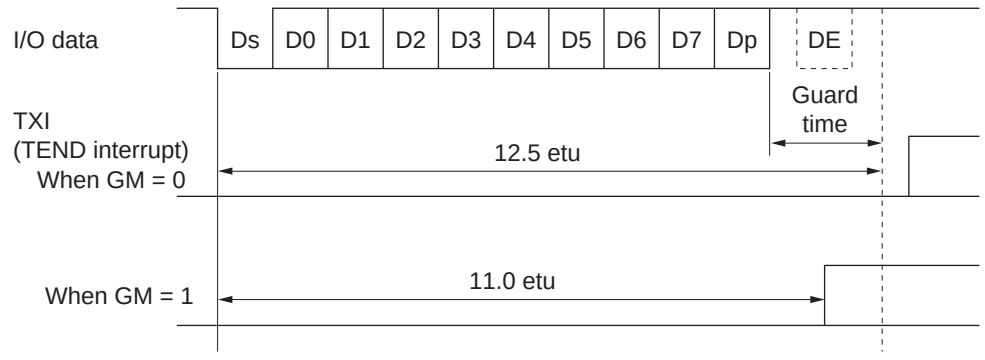
Figure 15.4 Sample Transmission Flowchart



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 flag is set

Note: When the ERS flag is set, it should be cleared until transfer of the last bit (D7 in LSB-first transmission, D0 in MSB-first transmission) of the next transfer data to be transmitted has been completed.

Figure 15.5 Relation between Transmit Operation and Internal Registers



Legend:
Ds: Start bit
D0 to D7: Data bits
Dp: Parity bit
DE: Error signal

Note: etu: Elementary time unit (time for transfer of 1 bit)

Figure 15.6 TEND Flag Generation Timing in Transmission

transmission processing flow.

[1] Perform smart card interface mode initialization as described above in Initialization.

[2] Check that the ORER flag and PER flag in SSR are cleared to 0. If either is set, perform the appropriate receive error handling, then clear both the ORER and the PER flag to 0.

[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.

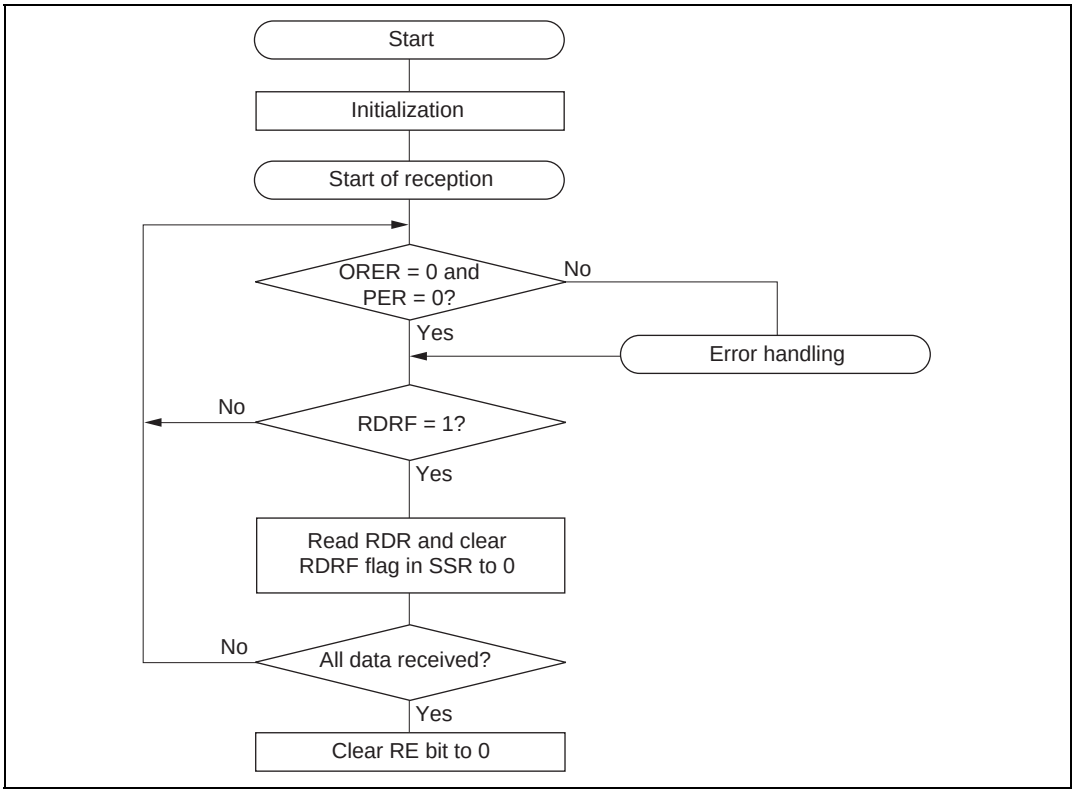


Figure 15.7 Sample Reception Flowchart

If reception ends and the RDRF flag is set to 1 while the RIE bit is set to 1 and interrupt requests are enabled, a receive data full interrupt (RXI) request will be generated. If an error occurs in reception and either the ORER flag or the PER flag is set to 1, a transmit/receive-error interrupt (ERI) request will be generated.

If the DMAC or DTC is activated by an RXI request, the receive data in which the error occurred is skipped, and only the number of bytes of receive data set in the DMAC or DTC are transferred.

For details, see Interrupt Operation (Except Block Transfer Mode) and Data Transfer Operation by DMAC or DTC below.

If a parity error occurs during reception and the PER is set to 1, the received data is still transferred to RDR, and therefore this data can be read.

Note: For details of operation in block transfer mode, see section 14.3.2, Operation in Asynchronous Mode.

Mode Switching Operation: When switching from receive mode to transmit mode, first confirm that the receive operation has been completed, then start from initialization, clearing RE bit to 0 and setting TE bit to 1. The RDRF flag or the PER and ORER flags can be used to check that the receive operation has been completed.

When switching from transmit mode to receive mode, first confirm that the transmit operation has been completed, then start from initialization, clearing TE bit to 0 and setting RE bit to 1. The TEND flag can be used to check that the transmit operation has been completed.

Fixing Clock Output: When the GM bit in SMR is set to 1, the clock output can be fixed with bits CKE1 and CKE0 in SCR. At this time, the minimum clock pulse width can be made the specified width.

Figure 15.8 shows the timing for fixing the clock output. In this example, GM is set to 1, CKE1 is cleared to 0, and the CKE0 bit is controlled.

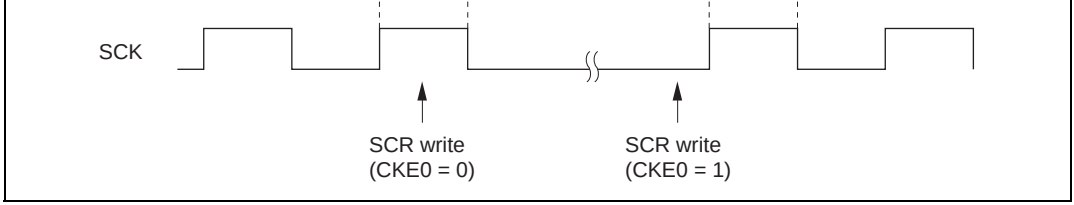


Figure 15.8 Timing for Fixing Clock Output

Interrupt Operation (Except Block Transfer Mode): There are three interrupt sources in smart card interface mode: transmit-data-empty interrupt (TXI) requests, transmit/receive-error interrupt (ERI) requests, and receive-data-full interrupt (RXI) requests. The transmit-end interrupt (TEI) request is not used in this mode.

When the TEND flag in SSR is set to 1, a TXI interrupt request is generated.

When the RDRF flag in SSR is set to 1, an RXI interrupt request is generated.

When any of flags ORER, PER, and ERS in SSR is set to 1, an ERI interrupt request is generated. The relationship between the operating states and interrupt sources is shown in table 15.8.

Note: For details of operation in block transfer mode, see section 14.4, SCI Interrupts.

Table 15.8 Smart Card Mode Operating States and Interrupt Sources

Operating State		Flag	Enable Bit	Interrupt Source	DTC Activation	DMAC Activation
Transmit Mode	Normal operation	TEND	TIE	TXI	Possible	Possible
	Error	ERS	RIE	ERI	Not possible	Not possible
Receive Mode	Normal operation	RDRF	RIE	RXI	Possible	Possible
	Error	PER, ORER	RIE	ERI	Not possible	Not possible

also set to 1 at the same time as the TEND flag in SSR, and a TXI interrupt is generated. If the TXI request is designated beforehand as a DMAC or DTC activation source, the DMAC or DTC will be activated by the TXI request, and transfer of the transmit data will be carried out. The TDRE and TEND flags are automatically cleared to 0 when data transfer is performed by the DMAC or DTC. In the event of an error, the SCI retransmits the same data automatically. The TEND flag remains cleared to 0 during this time, and the DMAC is not activated. Thus, the number of bytes specified by the SCI and DMAC are transmitted automatically even in retransmission following an error. However, the ERS flag is not cleared automatically when an error occurs, and so the RIE bit should be set to 1 beforehand so that an ERI request will be generated in the event of an error, and the ERS flag will be cleared.

When performing transfer using the DMAC or DTC, it is essential to set and enable the DMAC or DTC before carrying out SCI setting. For details of the DMAC and DTC setting procedures, see section 7, DMA Controller, and section 8, Data Transfer Controller.

In a receive operation, an RXI interrupt request is generated when the RDRF flag in SSR is set to 1. If the RXI request is designated beforehand as a DMAC or DTC activation source, the DMAC or DTC will be activated by the RXI request, and transfer of the receive data will be carried out. The RDRF flag is cleared to 0 automatically when data transfer is performed by the DMAC or DTC. If an error occurs, an error flag is set but the RDRF flag is not. Consequently, the DMAC or DTC is not activated, but instead, an ERI interrupt request is sent to the CPU. Therefore, the error flag should be cleared.

Note: For details of operation in block transfer mode, see section 14.4, SCI Interrupts.

Switching the Mode: When switching between smart card interface mode and software standby mode, the following switching procedure should be followed in order to maintain the clock duty.

- 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 SCK pin to 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 the transmit/receive operation. At the same time, set the CKE1 bit to the value for the fixed output state in 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 preserved.

[5] Write H'00 to SMR and SCMR.

[6] Make the transition to the software standby state.

- When returning to smart card interface mode from software standby mode

[7] Exit the software standby state.

[8] Set the CKE1 bit in SCR to the value for the fixed output state (current SCK pin state) when software standby mode is initiated.

[9] Set smart card interface mode and output the clock. Signal generation is started with the normal duty.

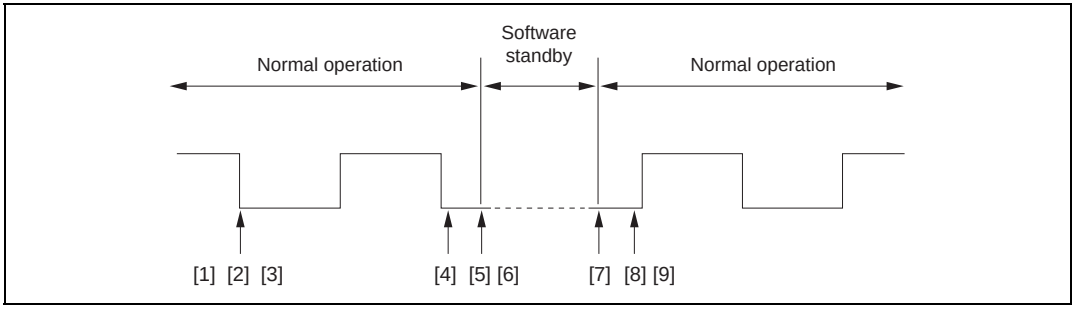


Figure 15.9 Clock Halt and Restart Procedure

[1] The initial state is port input and high impedance. Use a pull-up resistor or pull-down resistor to fix the potential.

[2] Fix the SCK pin to the specified output level with the CKE1 bit in SCR.

[3] Set SMR and SCMR, and switch to smart card mode operation.

[4] Set the CKE0 bit in SCR to 1 to start clock output.

15.3.8 Operation in Block Transfer Mode

Operation in block transfer mode is the same as in SCI asynchronous mode, except for the following points. For details, see section 14.3.2, Operation in Asynchronous Mode.

Data Format: The data format is 8 bits with parity. There is no stop bit, but there is a guard time of 2 or more bits (1 or more bits in reception).

Also, except during transmission (with start bit, data bits, and parity bit), the transmission pins go to the high-impedance state, so the signal lines must be fixed high with a pull-up resistor.

Transmit/Receive Clock: Only an internal clock generated by the built-in baud rate generator can be used as the transmit/receive clock. The number of basic clock periods in a 1-bit transfer interval can be set to 32, 64, 372, or 256 with bits BCP1 and BCP0. For details, see section 15.3.5, Clock.

ERS (FER) Flag: As with the normal smart card interface, the ERS flag indicates the error signal status, but since error signal transmission and reception is not performed, this flag is always cleared to 0.

The following points should be noted when using the SCI as a smart card interface.

Receive Data Sampling Timing and Receive Margin in Smart Card Interface Mode: In smart card interface mode, the SCI operates on a base clock with a frequency of 32, 64, 372, or 256 times the transfer rate (determined by bits BCP1 and BCP0).

In reception, the SCI samples the falling edge of the start bit using the base clock, and performs internal synchronization. Receive data is latched internally at the rising edge of the 16th, 32nd, 186th, or 128th pulse of the base clock. Use of a 372-times clock is illustrated in figure 15.10.

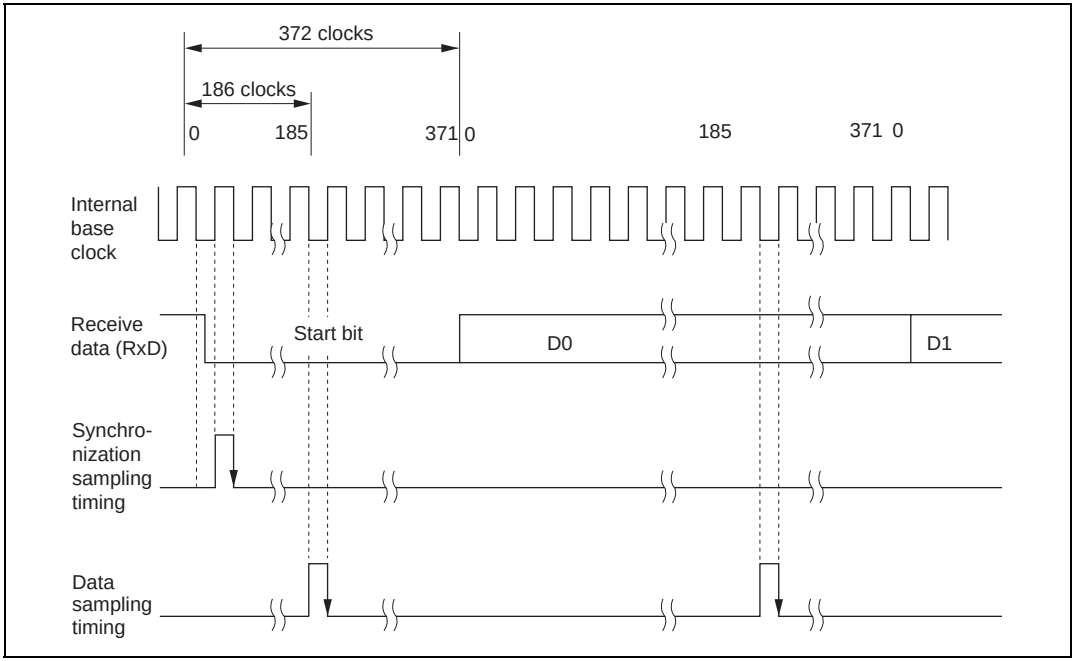


Figure 15.10 Receive Data Sampling Timing in Smart Card Mode (When Using 372-Times Clock)

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

Where M: Receive margin (%)
N: Ratio of bit rate to clock (N = 32, 64, 372, 256)
D: Clock duty (D = 0 to 1.0)
L: Frame length (L = 10)
F: Absolute value of clock frequency deviation

Assuming values of F = 0, D = 0.5, and N = 372 in the above formula, the receive margin formula is as follows.

When D = 0.5 and F = 0,

$$M = (0.5 - 1/2 \times 372) \times 100\%$$
$$= 49.866\%$$

- Retransfer operation when SCI is in receive mode

  Figure 15.11 illustrates the retransfer operation when the SCI is in receive mode.

[1] If an error is found when the received parity bit is checked, the PER bit in SSR is automatically set to 1. If the RIE bit in SCR is enabled at this time, an ERI interrupt request is generated. The PER bit in SSR should be kept cleared to 0 until the next parity bit is sampled.

[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 not set.

[4] If no error is found when the received parity bit is checked, the receive operation is judged to have been completed normally, and the RDRF flag in SSR is automatically set to 1. If the RIE bit in SCR is enabled at this time, an RXI interrupt request is generated.

If DMAC or 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 DMAC or DTC, the RDRF flag is automatically cleared to 0.

[5] When a normal frame is received, the pin retains the high-impedance state at the timing for error signal transmission.

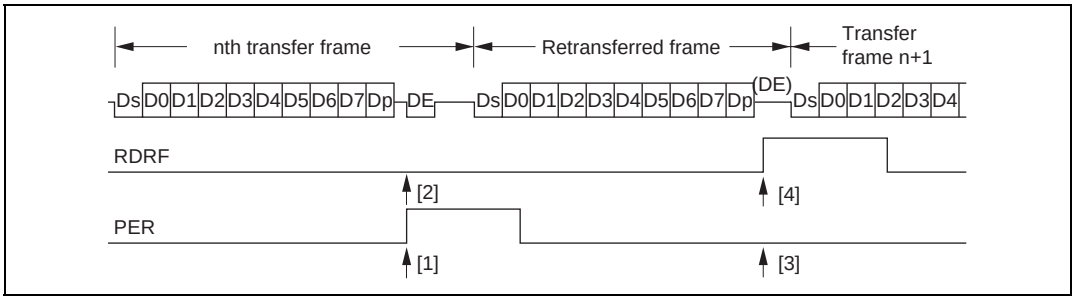


Figure 15.11 Retransfer Operation in SCI Receive Mode

[6] If an error signal is sent back from the receiving end after transmission of one frame is completed, the ERS bit in SSR is set to 1. If the RIE bit in SCR is enabled at this time, an ERI interrupt request is generated. The ERS bit in SSR should be kept cleared to 0 until the next parity bit is sampled.

[7] The TEND bit in SSR is not set for a frame for which an error signal indicating an abnormality is received.

[8] If an error signal is not sent back from the receiving end, the ERS bit in SSR is not set.

[9] If an error signal is not sent back from the receiving end, transmission of one frame, including a retransfer, is judged to have been completed, and the TEND bit in SSR is set to 1. If the TIE bit in SCR is enabled at this time, a TXI interrupt request is generated.

If data transfer by the DMAC or DTC by means of the TXI source is enabled, the next data can be written to TDR automatically. When data is written to TDR by the DMAC or DTC, the TDRE bit is automatically cleared to 0.

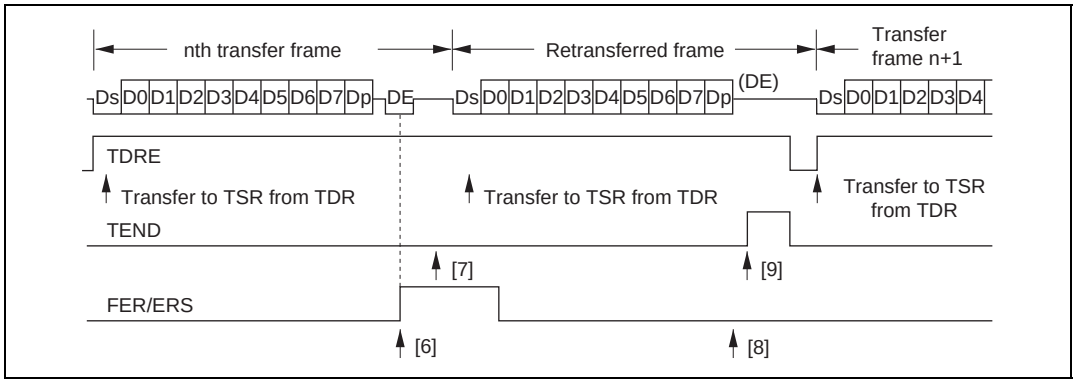


Figure 15.12 Retransfer Operation in SCI Transmit Mode

16.1 Overview

The chip incorporates a successive-approximations type 10-bit A/D converter that allows up to twelve analog input channels to be selected.

16.1.1 Features

A/D converter features are listed below.

- 10-bit resolution
- Twelve input channels
- Settable analog conversion voltage range
  - — Conversion of analog voltages with the reference voltage pin (V_{ref}) as the analog reference voltage
- High-speed conversion
  - — Minimum conversion time: 6.7 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 8-bit timer), 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 conversion
  - — The DMA controller (DMAC) or data transfer controller (DTC) can be activated for data transfer by an interrupt
- Module stop mode can be set
  - — As the initial setting, A/D converter operation is halted. Register access is enabled by exiting module stop mode.

Figure 16.1 shows a block diagram of the A/D converter.

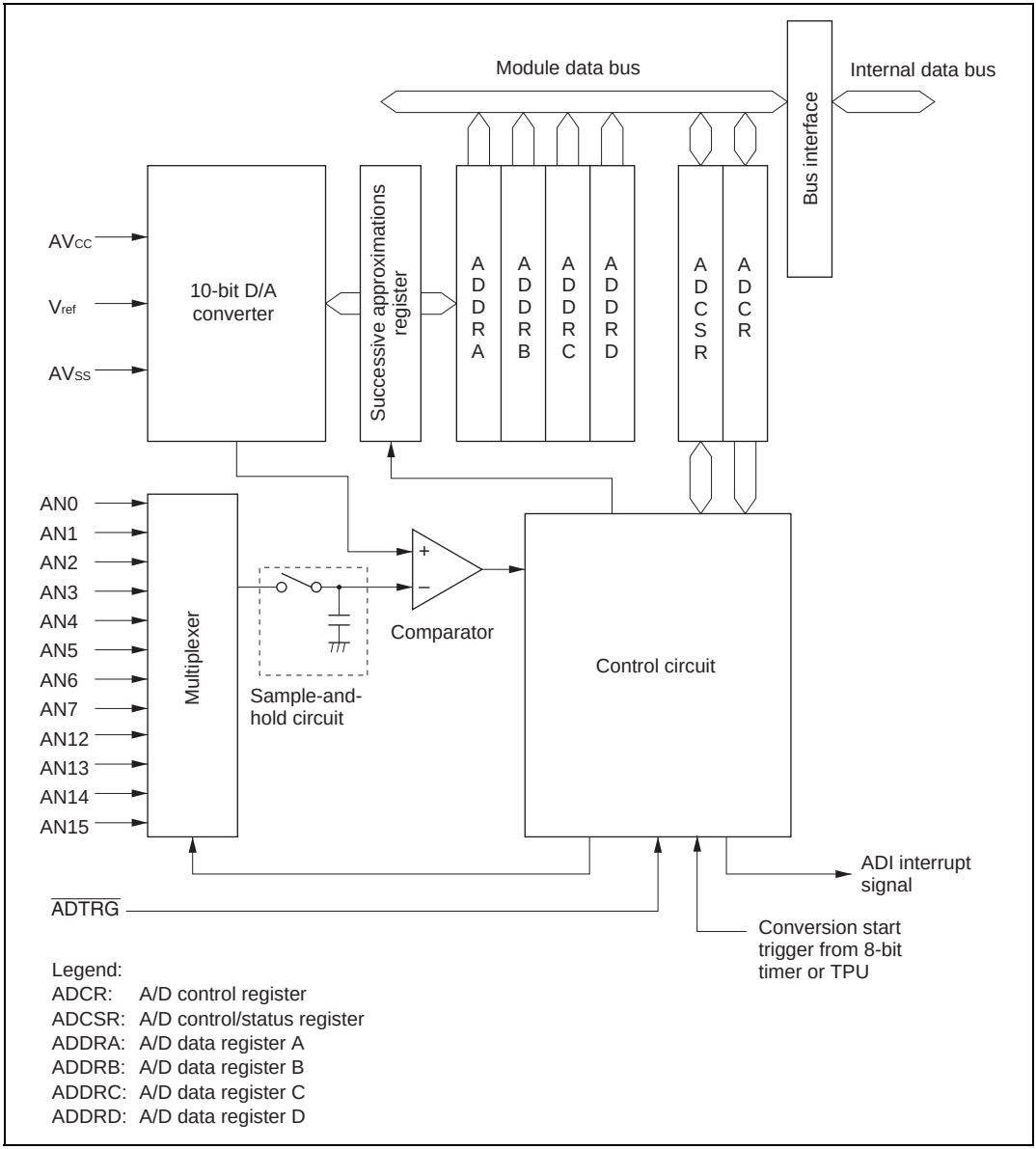


Figure 16.1 Block Diagram of A/D Converter

Table 16.1 summarizes the input pins used by the A/D converter.

The AV_{CC} and AV_{SS} pins are the power supply pins for the analog block in the A/D converter. The V_{ref} pin is the A/D conversion reference voltage pin.

The twelve analog input pins are divided into two channel sets and two groups: channel set 0 (AN0 to AN7), channel set 1 (AN12 to AN15), group 0 (AN0 to AN3), and group 1 (AN4 to AN7, AN12 to AN15).

Table 16.1 A/D Converter Pins

Pin Name	Symbol	I/O	Function
Analog power supply pin	AV_{CC}	Input	Analog block power supply
Analog ground pin	AV_{SS}	Input	Analog block ground and A/D conversion reference voltage
Reference voltage pin	V_{ref}	Input	A/D conversion reference voltage
Analog input pin 0	AN0	Input	Channel set 0 (CH3 = 1), group 0 analog inputs
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 = 1), group 1 analog inputs
Analog input pin 5	AN5	Input	
Analog input pin 6	AN6	Input	
Analog input pin 7	AN7	Input	
Analog input pin 12	AN12	Input	Channel set 1 (CH3 = 0), group 1 analog inputs
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 conversion

Table 16.2 summarizes the registers of the A/D converter.

Table 16.2 A/D Converter Registers

Name	Abbreviation	R/W	Initial Value	Address*1
A/D data register AH	ADDRAH	R	H'00	H'FF90
A/D data register AL	ADDRAL	R	H'00	H'FF91
A/D data register BH	ADDRBH	R	H'00	H'FF92
A/D data register BL	ADDRBL	R	H'00	H'FF93
A/D data register CH	ADDRCH	R	H'00	H'FF94
A/D data register CL	ADDRCL	R	H'00	H'FF95
A/D data register DH	ADDRDH	R	H'00	H'FF96
A/D data register DL	ADDRDL	R	H'00	H'FF97
A/D control/status register	ADCSR	R/(W)*2	H'00	H'FF98
A/D control register	ADCR	R/W	H'3F	H'FF99
Module stop control register	MSTPCR	R/W	H'3FFF	H'FF3C

Notes: 1. Lower 16 bits of the address.
2. Bit 7 can only be written with 0 for flag clearing.

16.2.1 A/D Data Registers A to D (ADDRA to ADDRD)

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	—	—	—	—	—	—
Initial value	:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
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 the results of A/D conversion.

The 10-bit data resulting from A/D conversion is transferred to the ADDR register for the selected channel and stored there. The upper 8 bits of the converted data are transferred to the upper byte (bits 15 to 8) of ADDR, and the lower 2 bits are transferred to the lower byte (bits 7 and 6) and stored. Bits 5 to 0 are always read as 0.

The correspondence between the analog input channels and ADDR registers is shown in table 16.3.

The ADDR registers can always be read by the CPU. 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.

The ADDR registers are initialized to H'0000 by a reset, and in standby mode or module stop mode.

Table 16.3 Analog Input Channels and Corresponding ADDR Registers

Analog Input Channel				
Channel Set 0 (CH3 = 1)		Channel Set 1 (CH3 = 0)		
Group 0 (CH2 = 0)	Group 1 (CH2 = 1)	Group 0 (CH2 = 0)	Group 1 (CH2 = 1)	A/D Data Register
AN0	AN4	Setting prohibited	AN12	ADDRA
AN1	AN5	Setting prohibited	AN13	ADDRB
AN2	AN6	Setting prohibited	AN14	ADDRC
AN3	AN7	Setting prohibited	AN15	ADDRD

Bit	:	7	6	5	4	3	2	1	0
		ADF	ADIE	ADST	SCAN	CKS	CH2	CH1	CH0
Initial value	:	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

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 and shows the status of the operation.

ADCSR is initialized to H'00 by a reset, and in standby mode or module stop mode.

Bit 7—A/D End Flag (ADF): Status flag that indicates the end of A/D conversion.

Bit 7 ADF	Description	
0	[Clearing conditions] • When 0 is written to the ADF flag after reading ADF = 1 • When the DMAC or DTC is activated by an ADI interrupt and ADDR is read	(Initial value)
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) requests at the end of A/D conversion.

Bit 6 ADIE	Description	
0	A/D conversion end interrupt (ADI) request disabled	(Initial value)
1	A/D conversion end interrupt (ADI) request enabled	

The ADST bit can be set to 1 by software, a timer conversion start trigger, or the A/D external trigger input pin ($\overline{\text{ADTRG}}$).

Bit 5 ADST	Description	
0	• A/D conversion stopped	(Initial value)
1	• Single mode: A/D conversion is started. Cleared to 0 automatically when conversion on the specified channel ends • Scan mode: A/D conversion is started. Conversion continues sequentially on the selected channels until ADST is cleared to 0 by software, a reset, or a transition to standby mode or module stop mode	

Bit 4—Scan Mode (SCAN): Selects single mode or scan mode as the A/D conversion operating mode. See section 16.4, Operation, for details of single mode and scan mode operation. Only set the SCAN bit while conversion is stopped (ADST = 0).

Bit 4 SCAN	Description	
0	Single mode	(Initial value)
1	Scan mode	

Bit 3—Clock Select (CKS): Used together with the CKS1 bit in ADCR to set the A/D conversion time. Only change the conversion time while conversion is stopped (ADST = 0).

ADCR Bit 3 CKS1	Bit 3 CKS	Description	
0	0	Conversion time = 530 states (max.)	
	1	Conversion time = 68 states (max.)	
1	0	Conversion time = 266 states (max.)	(Initial value)
	1	Conversion time = 134 states (max.)	

Bits 2 to 0—Channel Select 2 to 0 (CH2 to CH0): These bits are used together with the CH3 bit in ADCR and SCAN bit to select the analog input channels.

Only set the input channel(s) while conversion is stopped (ADST = 0).

CH3	CH2	CH1	CH0	(SCAN = 0)	(SCAN = 1)
0	0	0	0	Setting prohibited	Setting prohibited
			1		
		1	0		
			1		
	1	0	0	AN12	AN12
			1	AN13	AN12, AN13
		1	0	AN14	AN12 to AN14
			1	AN15	AN12 to AN15
1	0	0	0	AN0 (Initial value)	AN0
			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

16.2.3 A/D Control Register (ADCR)

Bit	:	7	6	5	4	3	2	1	0
		TRGS1	TRGS0	—	—	CKS1	CH3	—	—
Initial value	:	0	0	1	1	1	1	1	1
R/W	:	R/W	R/W	—	—	R/W	R/W	—	—

ADCR is an 8-bit readable/writable register that enables or disables external triggering of A/D conversion operations.

ADCR is initialized to H'3F by a reset, and in standby mode or module stop mode.

Bits 7 and 6—Timer Trigger Select 1 and 0 (TRGS1, TRGS0): These bits select enabling or disabling of the start of A/D conversion by a trigger signal. Only set bits TRGS1 and TRGS0 while conversion is stopped (ADST = 0).

0	0	A/D conversion start by external trigger is disabled	(Initial value)
	1	A/D conversion start by external trigger (TPU) is enabled	
1	0	A/D conversion start by external trigger (8-bit timer) is enabled	
	1	A/D conversion start by external trigger pin ($\overline{\text{ADTRG}}$) is enabled	

Bits 5, 4, 1, and 0—Reserved: These bits cannot be modified and are always read as 1.

Bit 3—Clock Select 1 (CKS1): Used together with the CKS bit in ADCSR to set the A/D conversion time. See the description of the CKS bit for details.

Bit 2—Channel Select 3 (CH3): Used together with bits CH2, CH1, and CH0 in ADCSR to select the analog input channel(s). See the description of bits CH2, CH1, and CH0 for details.

16.2.4 Module Stop Control Register (MSTPCR)

	MSTPCRH								MSTPCRL							
Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value :	0	0	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/W	R/W	R/W

MSTPCR is a 16-bit readable/writable register that performs module stop mode control.

When the MSTP9 bit in MSTPCR is set to 1, A/D converter operation stops at the end of the bus cycle and a transition is made to module stop mode. Registers cannot be read or written to in module stop mode. For details, see section 21.5, Module Stop Mode.

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

Bit 9—Module Stop (MSTP9): Specifies the A/D converter module stop mode.

Bit 9 MSTP9	Description	
0	A/D converter module stop mode cleared	
1	A/D converter module stop mode set	(Initial value)

ADDRA to ADDRD are 16-bit registers, and the data bus to the bus master is 8 bits wide. Therefore, in accesses by the bus master, the upper byte is accessed directly, but the lower byte is accessed via a temporary register (TEMP).

A data read from ADDR is performed as follows. When the upper byte is read, the upper byte value is transferred to the CPU and the lower byte value is transferred to TEMP. Next, when the lower byte is read, the TEMP contents are transferred to the CPU.

When reading ADDR, always read the upper byte before the lower byte. It is possible to read only 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.

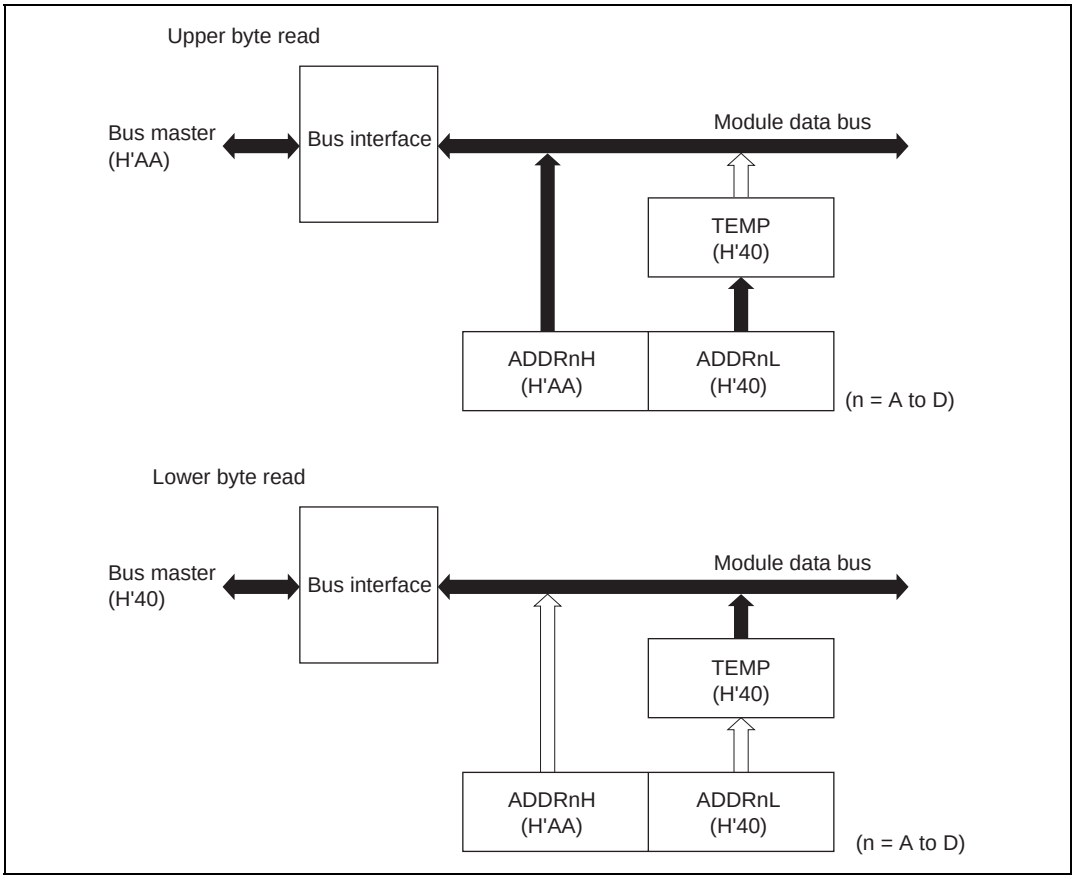


Figure 16.2 ADDR Access Operation (Reading H'AA40)

The A/D converter operates by successive approximations with 10-bit resolution. It has two operating modes: single mode and scan mode.

16.4.1 Single Mode (SCAN = 0)

Single mode is selected when A/D conversion is to be performed on a single channel only. A/D conversion is started when the ADST bit is set to 1 by software or by external trigger input. The ADST bit remains set to 1 during A/D conversion, and is automatically cleared to 0 when conversion ends.

On completion of conversion, the ADF flag is set to 1. If the ADIE bit is set to 1 at this time, an ADI interrupt request is generated. The ADF flag is cleared by writing 0 to it after reading ADCSR.

When the operating mode or analog input channel must be changed during analog conversion, to prevent incorrect operation, first clear the ADST bit to 0 in ADCSR to halt A/D conversion. After making the necessary changes, set the ADST bit to 1 to start A/D conversion again. The ADST 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. Figure 16.3 shows a timing diagram for this example.

[1] Single mode is selected (SCAN = 0), input channel AN1 is selected (CH3 = 1, CH2 = 0, CH1 = 0, CH0 = 1), the A/D interrupt is enabled (ADIE = 1), and A/D conversion is started (ADST = 1).

[2] When A/D conversion is completed, the result is transferred to ADDRB. At the same time the ADF flag is set to 1, the ADST bit is cleared to 0, and the A/D converter becomes idle.

[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 conversion result (ADDRB).

[7] Execution of the A/D interrupt handling routine ends. After that, if the ADST bit is set to 1, A/D conversion starts again and steps [2] to [7] are repeated.

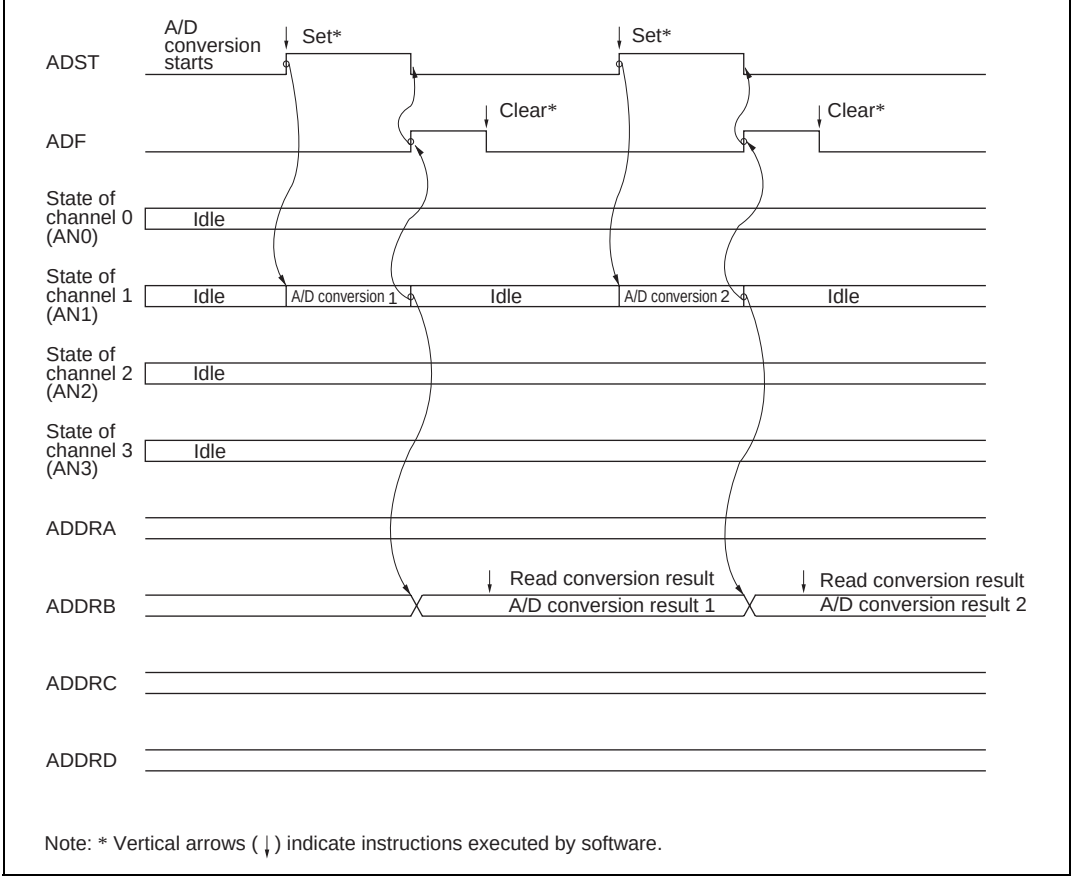


Figure 16.3 Example of A/D Converter Operation (Single Mode, Channel 1 Selected)

Scan mode is useful for monitoring analog inputs in a group of one or more channels. When the ADST bit is set to 1 by software, timer, or external trigger input, A/D conversion starts on the first channel in the group (AN0). When two or more channels are selected, after conversion of the first channel ends, conversion of the second channel (AN1) starts immediately. A/D conversion continues cyclically on the selected channels until the ADST bit is cleared to 0. The conversion results are transferred for storage into the ADDR registers corresponding to the channels.

When the operating mode or analog input channel must be changed during analog conversion, to prevent incorrect operation, first clear the ADST bit to 0 in ADCSR to halt A/D conversion. After making the necessary changes, set the ADST bit to 1 to start A/D conversion again. The ADST bit can be set at the same time as the operating mode or input channel is changed.

Typical operations when three channels (AN0 to AN2) are selected in scan mode are described next. Figure 16.4 shows a timing diagram for this example.

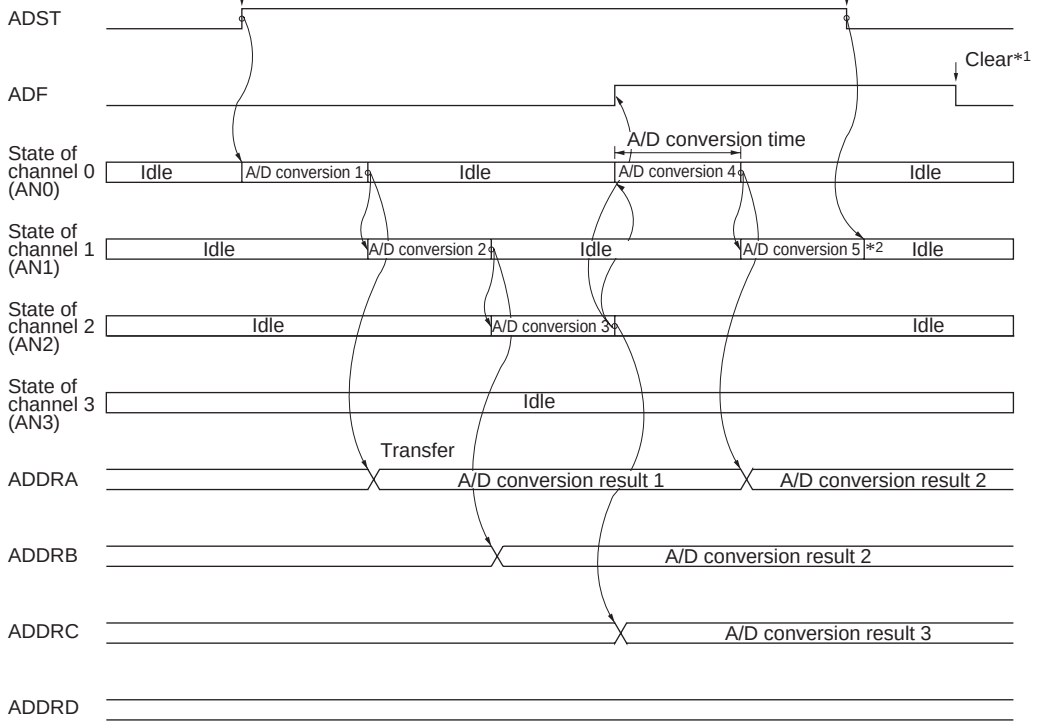
[1] Scan mode is selected (SCAN = 1), channel set 0 is selected (CH3 = 0), scan group 0 is selected (CH2 = 0), analog input channels AN0 to AN2 are selected (CH1 = 1, CH0 = 0), and A/D conversion is started (ADST = 1)

[2] When A/D conversion of the first channel (AN0) is completed, the result is transferred to 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 flag is set to 1 and conversion of the first channel (AN0) starts again. If the ADIE bit is set to 1 at this 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 ADST bit is cleared to 0, A/D conversion stops. After that, if the ADST bit is set to 1, A/D conversion starts again from the first channel (AN0).



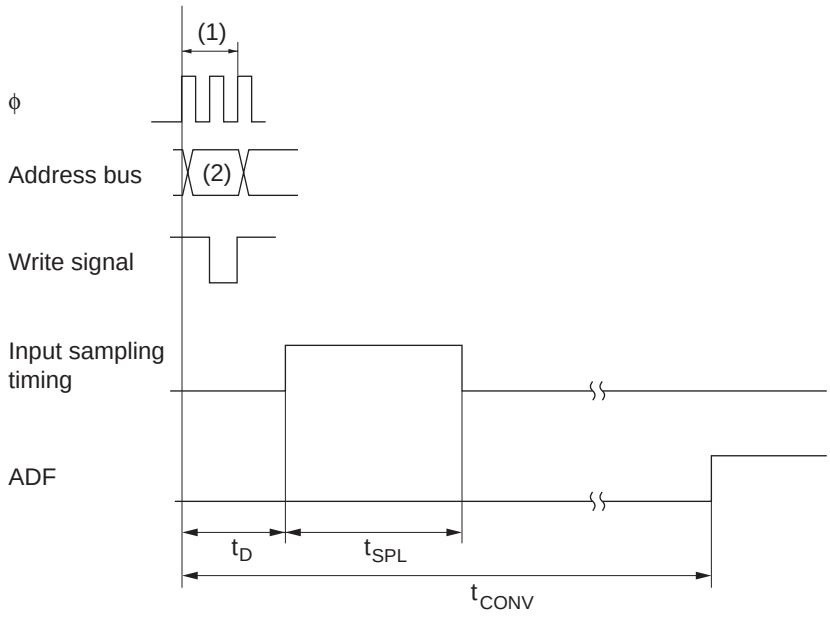
Notes: 1. Vertical arrows (↓) indicate instructions executed by software.
2. Data currently being converted is ignored.

Figure 16.4 Example of A/D Converter Operation (Scan Mode, Channels AN0 to AN2 Selected)

The A/D converter has a built-in sample-and-hold circuit. The A/D converter samples the analog input at a time t_D after the ADST bit is set to 1, then starts conversion. Figure 16.5 shows the A/D conversion timing. Table 16.4 indicates the A/D conversion time.

As indicated in figure 16.5, the A/D conversion time includes t_D and the input sampling time. The length of t_D varies depending on the timing of the write access to ADCSR. The total conversion time therefore varies within the ranges indicated in table 16.4.

In scan mode, the values given in table 16.4 apply to the first conversion time. In the second and subsequent conversions the conversion time is as shown in table 16.5.



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

		CKS1 = 0						CKS1 = 1					
		CKS = 0			CKS = 1			CKS = 0			CKS = 1		
Item	**Symbol**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**
A/D conversion start delay	t_D	18	—	33	4	—	5	10	—	17	6	—	9
Input sampling time	t_{SPL}	—	127	—	—	15	—	—	63	—	—	31	—
A/D conversion time	t_{CONV}	515	—	530	67	—	68	259	—	266	131	—	134

Note: Values in the table are the number of states.

Table 16.5 A/D Conversion Time (Scan Mode)

CKS1	CKS	Conversion Time (States)
0	0	512 (Fixed)
	1	64 (Fixed)
1	0	256 (Fixed)
	1	128 (Fixed)

16.4.4 External Trigger Input Timing

A/D conversion can be externally triggered. When the TRGS1 and TRGS0 bits are set to B'11 in ADCR, external trigger input is enabled at the $\overline{\text{ADTRG}}$ pin. A falling edge at the $\overline{\text{ADTRG}}$ pin sets the ADST bit to 1 in ADCSR, starting A/D conversion. Other operations, in both single and scan modes, are the same as when the ADST bit has been set to 1 by software. Figure 16.6 shows the timing.

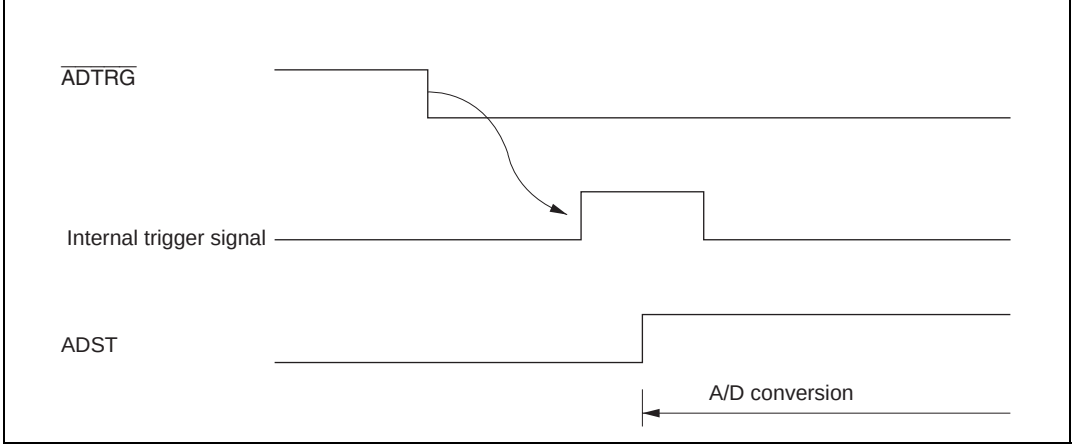


Figure 16.6 External Trigger Input Timing

16.5 Interrupts

The A/D converter generates an A/D conversion end interrupt (ADI) at the end of A/D conversion. ADI interrupt requests can be enabled or disabled by means of the ADIE bit in ADCSR.

The DTC or DMAC can be activated by an ADI interrupt. Having the converted data read by the DTC or DMAC in response to an ADI interrupt enables continuous conversion to be achieved without imposing a 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	DMAC Activation
ADI	Interrupt due to end of conversion	Possible	Possible

The following points should be noted when using the A/D converter.

Setting Range of Analog Power Supply and Other Pins

1. Analog input voltage range
   The voltage applied to analog input pins ANn during A/D conversion should be in the range $AV_{SS} \leq ANn \leq V_{ref}$.

2. Relation between AV_{CC}, AV_{SS} and V_{CC}, V_{SS}
   As the relationship between AV_{CC}, AV_{SS} and V_{CC}, V_{SS}, set $AV_{SS} = V_{SS}$. If the A/D converter is not used, the AV_{CC} and AV_{SS} pins must not be left open.

3. V_{ref} input range
   The analog reference voltage input at the V_{ref} pin should be set in the range $V_{ref} \leq AV_{CC}$. The V_{ref} pin should be set as $V_{ref} = V_{CC}$ when the A/D converter is not used. Do not leave the V_{ref} pin open.

If conditions 1, 2, and 3 above are not met, the reliability of the device may be adversely affected.

Notes on Board Design: In board design, digital circuitry and analog circuitry should be as mutually isolated as possible, and layout in which digital circuit signal lines and analog circuit signal lines cross or are in close proximity should be avoided as far as possible. Failure to do so may result in incorrect operation of the analog circuitry due to inductance, adversely affecting A/D conversion values.

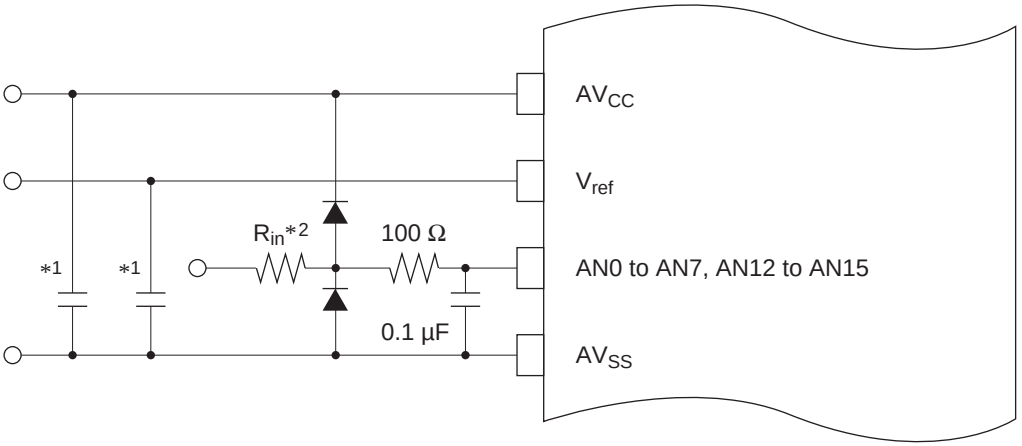
Also, digital circuitry must be isolated from the analog input signals (AN0 to AN7 and AN12 to AN15), analog reference power supply (V_{ref}), and analog power supply (AV_{CC}) by the analog ground (AV_{SS}). Also, the analog ground (AV_{SS}) should be connected at one point to a stable digital ground (V_{SS}) on the board.

Notes on Noise Countermeasures: A protection circuit connected to prevent damage due to an abnormal voltage such as an excessive surge at the analog input pins (AN0 to AN7 and AN12 to AN15) and analog reference power supply (V_{ref}) should be connected between AV_{CC} and AV_{SS} as shown in figure 16.7.

Also, the bypass capacitors connected to AV_{CC} and V_{ref} and the filter capacitor connected to AN0 to AN7 must be connected to AV_{SS}.

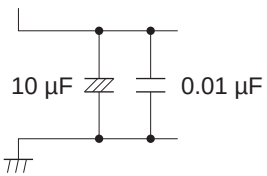
If a filter capacitor is connected as shown in figure 16.7, the input currents at the analog input pins (AN0 to AN7 and AN12 to AN15) are averaged, and so an error may arise. Also, when A/D

input impedance (R_{in}), an error will arise in the analog input pin voltage. Careful consideration is therefore required when deciding the circuit constants.



Notes: Values are reference values.

1.



2. R_{in}: Input impedance

Figure 16.7 Example of Analog Input Protection Circuit

- 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 characteristic when the digital output changes from the minimum voltage value B'0000000000 to B'0000000001. (See figure 16.9.)

- Full-scale error

  The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from B'1111111110 to B'1111111111. (See figure 16.9.)

- Quantization error

  The deviation inherent in the A/D converter, given by 1/2 LSB. (See figure 16.8.)

- Nonlinearity error

  The error with respect to the ideal A/D conversion characteristic between the zero voltage and the full-scale voltage. Does not include the offset error, full-scale error, or quantization error.

- Absolute precision

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

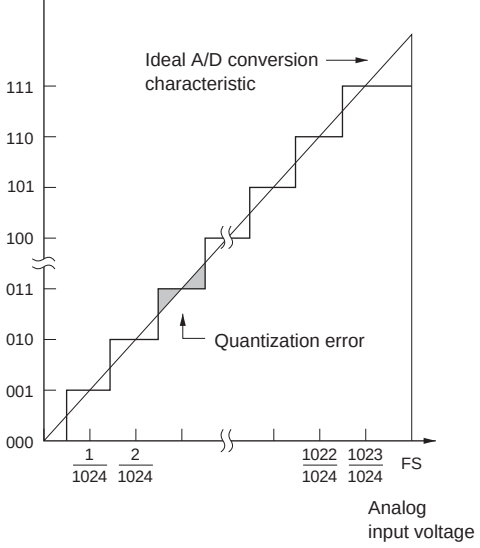


Figure 16.8 A/D Conversion Precision Definitions (1)

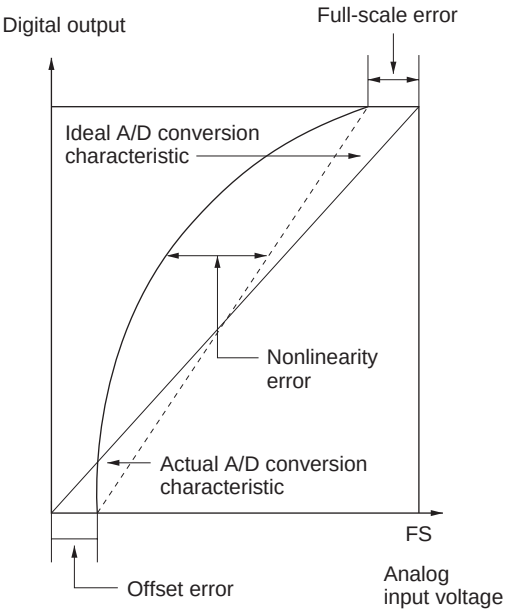


Figure 16.9 A/D Conversion Precision Definitions (2)

This specification is provided to enable the A/D converter's sample-and-hold circuit input capacitance to be charged within the sampling time; if the sensor output impedance exceeds 5 kΩ, charging may be insufficient and it may not be possible to guarantee the A/D conversion precision.

If a large capacitance is provided externally, the input load will essentially comprise only the internal input resistance of 10 kΩ, and the signal source impedance is ignored. However, since a low-pass filter effect is obtained in this case, it may not be possible to follow an analog signal with a large differential coefficient (e.g., 5 mV/µs or greater).

When converting a high-speed analog signal, a low-impedance buffer should be inserted.

Influences on Absolute Precision: Adding capacitance results in coupling with GND, and therefore noise in GND may adversely affect absolute precision. Be sure to make the connection to an electrically stable GND such as AV_{SS}.

Care is also required to insure that filter circuits do not communicate with digital signals on the mounting board, so acting as antennas.

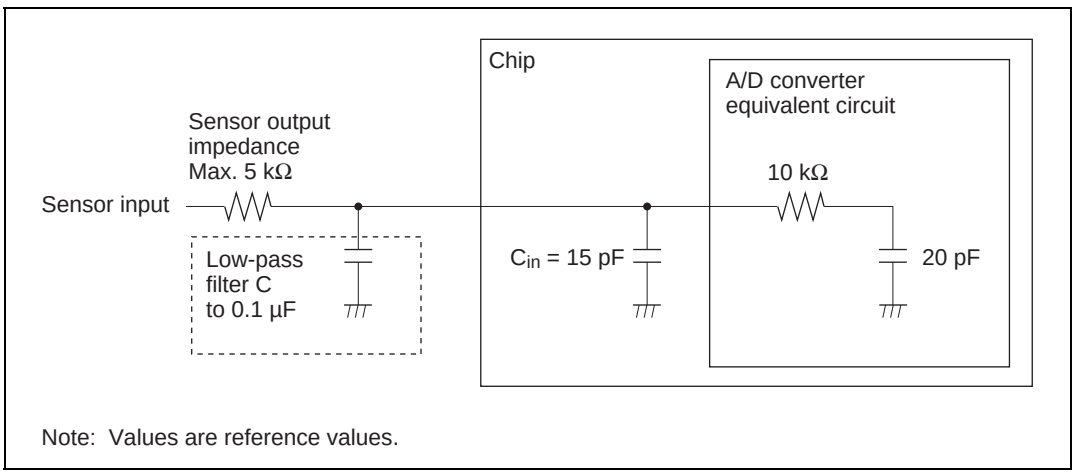


Figure 16.10 Example of Analog Input Circuit

17.1 Overview

The chip includes an 8-bit resolution D/A converter with maximum four analog signal output channels.

17.1.1 Features

D/A converter features are listed below.

- 8-bit resolution
- Four output channels
- Maximum conversion time of 10 µs (with 20 pF-load)
- Output voltage of 0 V to V_{ref}
- D/A output hold function in software standby mode
- Module stop mode can be set
  - As the initial setting, D/A converter operation is halted. Register access is enabled by exiting module stop mode.

Figure 17.1 shows a block diagram of the D/A converter.

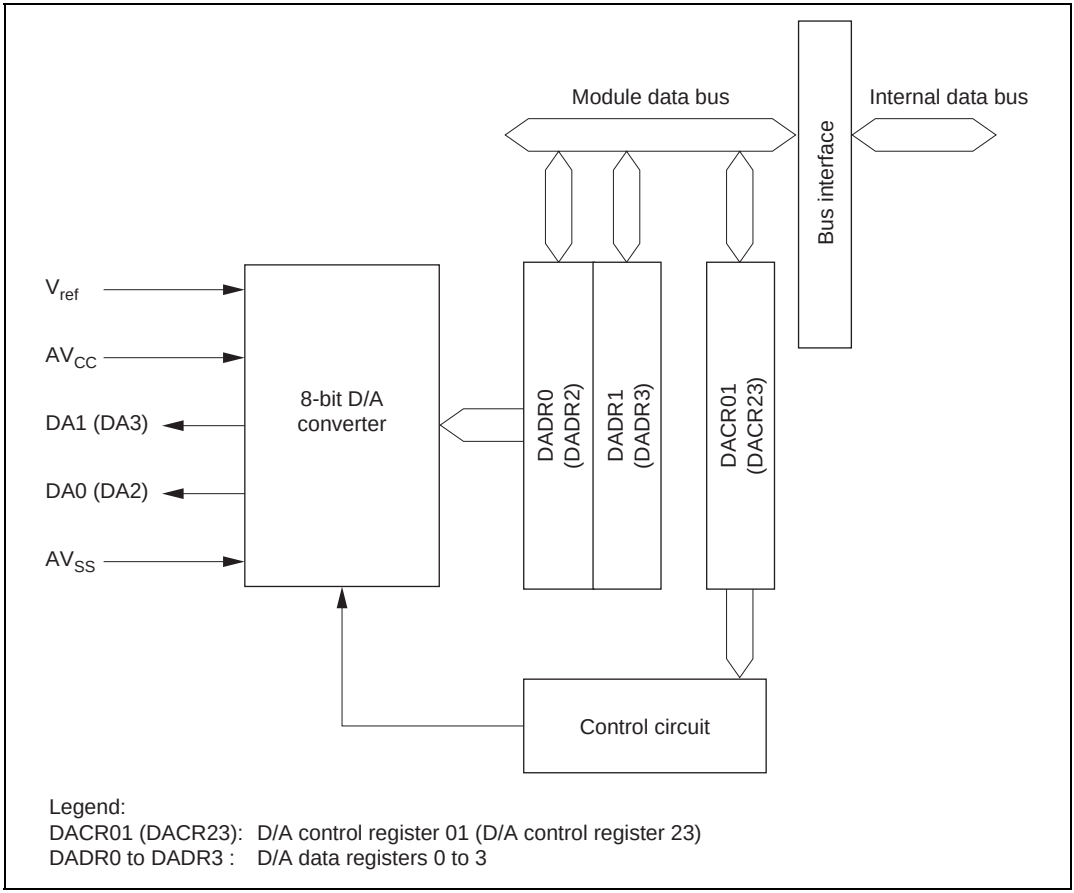


Figure 17.1 Block Diagram of D/A Converter

Table 17.1 summarizes the input and output pins of the D/A converter.

Table 17.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Analog power pin	AV_{CC}	Input	Analog power source
Analog ground pin	AV_{SS}	Input	Analog ground and reference voltage
Analog output pin 0	DA0	Output	Channel 0 analog output
Analog output pin 1	DA1	Output	Channel 1 analog output
Analog output pin 2	DA2	Output	Channel 2 analog output
Analog output pin 3	DA3	Output	Channel 3 analog output
Reference voltage pin	V_{ref}	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

Channels	Name	Abbreviation	R/W	Initial Value	Address*
0, 1	D/A data register 0	DADR0	R/W	H'00	H'FFA4
	D/A data register 1	DADR1	R/W	H'00	H'FFA5
	D/A control register 01	DACR01	R/W	H'1F	H'FFA6
2, 3	D/A data register 2	DADR2	R/W	H'00	H'FFA8
	D/A data register 3	DADR3	R/W	H'00	H'FFA9
	D/A control register 23	DACR12	R/W	H'1F	H'FFAA
Common	Module stop control register	MSTPCR	R/W	H'3FFF	H'FF3C

Note: * Lower 16 bits of the address.

17.2.1 D/A Data Registers 0 to 3 (DADR0 to DADR3)

Bit	:	7	6	5	4	3	2	1	0
Initial value	:	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

DADR0 to DADR3 are 8-bit readable/writable registers that store data for conversion.

Whenever output is enabled, the values in DADR0 to DADR3 are converted and output from the analog output pins.

DADR0 to DADR3 are each initialized to H'00 by a reset and in hardware standby mode.

17.2.2 D/A Control Registers 01 and 23 (DACR01, DACR23)

Bit	:	7	6	5	4	3	2	1	0
		DAOE1	DAOE0	DAE	—	—	—	—	—
Initial value	:	0	0	0	1	1	1	1	1
R/W	:	R/W	R/W	R/W	—	—	—	—	—

DACR01 and DACR23 are 8-bit readable/writable registers that control the operation of the D/A converter.

DACR01 and DACR23 are each 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	Analog output DA1 (DA3) is disabled	(Initial value)
1	Channel 1 (channel 3) D/A conversion is enabled; analog output DA1 (DA3) is enabled	

Bit 6 DAOE0	Description	
0	Analog output DA0 (DA2) is disabled	(Initial value)
1	Channel 0 (channel 2) D/A conversion is enabled; analog output DA0 (DA2) is enabled	

Bit 5—D/A Enable (DAE): Used together with the DAOE0 and DAOE1 bits to control D/A conversion. When the DAE bit is cleared to 0, channel 0 and 1 D/A conversions are controlled independently. When the DAE bit is set to 1, channel 0 and 1 D/A conversions are controlled together.

Output of conversion results is always controlled independently by the DAOE0 and DAOE1 bits.

Bit 7 DAOE1	Bit 6 DAOE0	Bit 5 DAE	Description
0	0	*	Channel 0 and 1 (channel 2 and 3) D/A conversions disabled
	1	0	Channel 0 (channel 2) D/A conversion enabled Channel 1 (channel 3) D/A conversion disabled
		1	Channel 0 and 1 (channel 2 and 3) D/A conversions enabled
1	0	0	Channel 0 (channel 2) D/A conversion disabled Channel 1 (channel 3) D/A conversion enabled
		1	Channel 0 and 1 (channel 2 and 3) D/A conversions enabled
	1	*	Channel 0 and 1 (channel 2 and 3) D/A conversions enabled

*: Don't care

If the chip enters software standby mode when D/A conversion is enabled, the D/A output is held and the analog power current is the same as during D/A conversion. When it is necessary to reduce the analog power current in software standby mode, clear both the DAOE0 and DAOE1 bits to 0 to disable D/A output.

Bits 4 to 0—Reserved: These bits cannot be modified and are always read as 1.

	MSTPCRH								MSTPCRL							
Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value :	0	0	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/W	R/W	R/W

MSTPCR is a 16-bit readable/writable register that performs module stop mode control.

When the MSTP10 bit or MSTP4 bit in MSTPCR is set to 1, D/A converter operation stops at the end of the bus cycle and a transition is made to module stop mode. Registers cannot be read or written to in module stop mode. For details, see section 21.5, Module Stop Mode.

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

Bit 10—Module Stop (MSTP10): Specifies the D/A converter channel 0 and 1module stop mode.

Bit 10 MSTP10	**Description**	
0	D/A converter (channel 0 and 1) module stop mode cleared	
1	D/A converter (channel 0 and 1) module stop mode set	(Initial value)

Bit 4—Module Stop (MSTP4): Specifies the D/A converter channel 2 and 3 module stop mode.

Bit 4 MSTP4	**Description**	
0	D/A converter (channel 2 and 3) module stop mode cleared	
1	D/A converter (channel 2 and 3) module stop mode set	(Initial value)

The D/A converter includes D/A conversion circuits for two channels, each of which can operate independently.

D/A conversion is performed continuously while enabled by DACR. If either DADR0 or DADR1 is written to, the new data is immediately converted. The conversion result is output by setting the corresponding DAOE0 or DAOE1 bit to 1.

The operation example described in this section concerns D/A conversion on channel 0. Figure 17.2 shows the timing of this operation.

[1] Write the conversion data to DADR0.

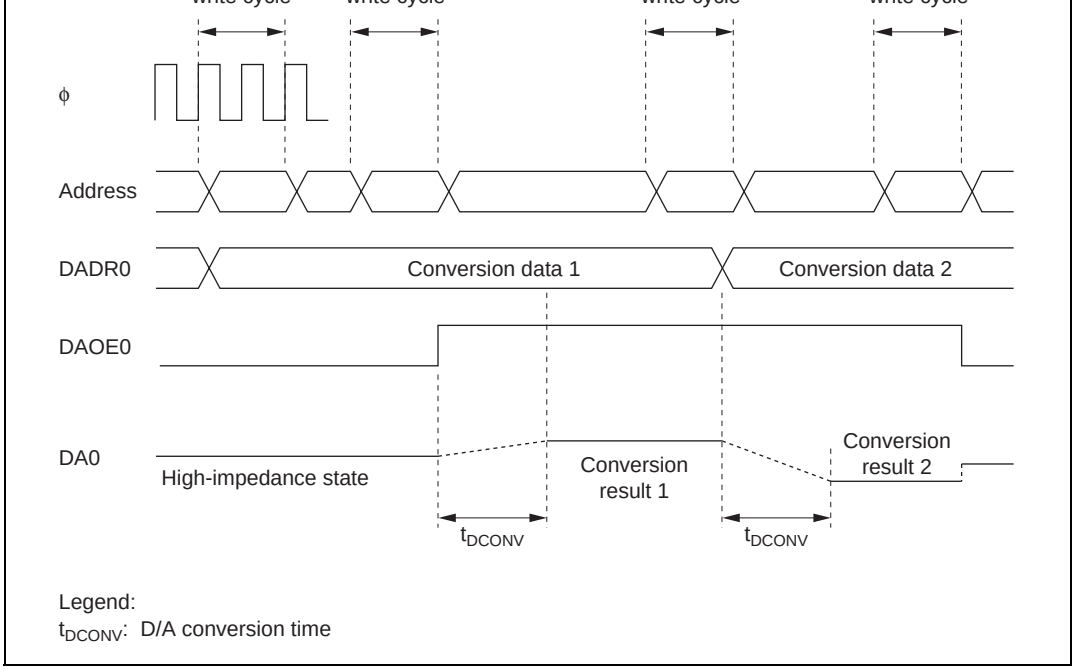
[2] Set the DAOE0 bit in DACR01 to 1. D/A conversion is started and the DA0 pin becomes an output pin. The conversion result is output after the conversion time has elapsed. The output value is expressed by the following formula:

$$\frac{\text{DADR contents}}{256} \times V_{ref}$$

The conversion results are output continuously until DADR0 is written to again or the DAOE0 bit is cleared to 0.

[3] If DADR0 is written to again, the new data is immediately converted. The new conversion result is output after the conversion time has elapsed.

[4] If the DAOE0 bit is cleared to 0, the DA0 pin becomes an input pin.



Legend:
t_{DCONV}: D/A conversion time

Figure 17.2 Example of D/A Converter Operation

18.1 Overview

The H8S/2339 has 32 kbytes of on-chip high-speed static RAM, the H8S/2338 and H8S/2332 have 8 kbytes. The RAM is connected to the CPU by a 16-bit data bus, enabling one-state access by the CPU to both byte data 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 (RAME) in the system control register (SYSCR).

18.1.1 Block Diagram

Figure 18.1 shows a block diagram of 32 kbytes of on-chip RAM.

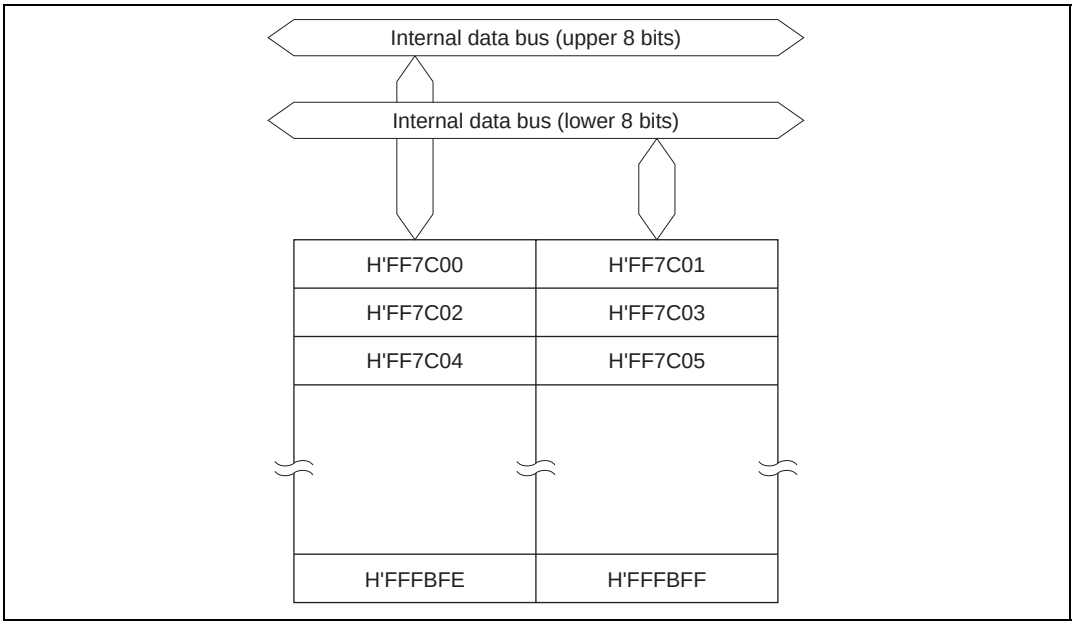


Figure 18.1 Block Diagram of RAM (32 kbytes)

The on-chip RAM is controlled by SYSCR. Table 18.1 shows the address and initial value of SYSCR.

Table 18.1 RAM Register

Name	Abbreviation	R/W	Initial Value	Address*
System control register	SYSCR	R/W	H'01	H'FF39

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	0
		—	—	INTM1	INTM0	NMIEG	LWROD	IRQPAS	RAME
Initial value	:	0	0	0	0	0	0	0	1
R/W	:	R/W	—	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 other bits in SYSCR, see section 5.2.1, System Control Register (SYSCR).

Bit 0—RAM Enable (RAME): Enables or disables the on-chip RAM. The RAME bit is initialized when the reset state is released. It is not initialized in software standby mode.

Bit 0 RAME	Description	
0	On-chip RAM is disabled	
1	On-chip RAM is enabled	(Initial value)

When the RAME bit is set to 1, accesses to addresses H'FFDC00 to H'FFFBFF are directed to the on-chip RAM. When the RAME bit is cleared to 0, the off-chip address space is accessed.

Since the on-chip RAM is connected to the CPU by an internal 16-bit data bus, it can be written to 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 must start at an even address.

Note: The amount of on-chip RAM differs depending on the product. Refer to section 3.5, Memory Map in Each Operating Mode, for details.

18.4 Usage Note

DTC register information can be located in addresses H'FFF800 to H'FFFBFF. When the DTC is used, the RAME bit must not be cleared to 0.

19.1 Overview

This chip has 384 or 256 kbytes of on-chip flash memory, or 256 or 128 kbytes of on-chip mask ROM. The ROM is connected to the bus master via a 16-bit data bus, enabling both byte and word data to be accessed in one state. Instruction fetching is thus speeded up, and processing speed increased.

The on-chip ROM is enabled and disabled by means of the mode pins (MD2 to MD0) and the EAE bit in BCRL.

The flash memory version of the chip can be erased and programmed with a PROM programmer, as well as on-board.

19.1.1 Block Diagram

Figure 19.1 shows a block diagram of 256 kbytes of on-chip ROM.

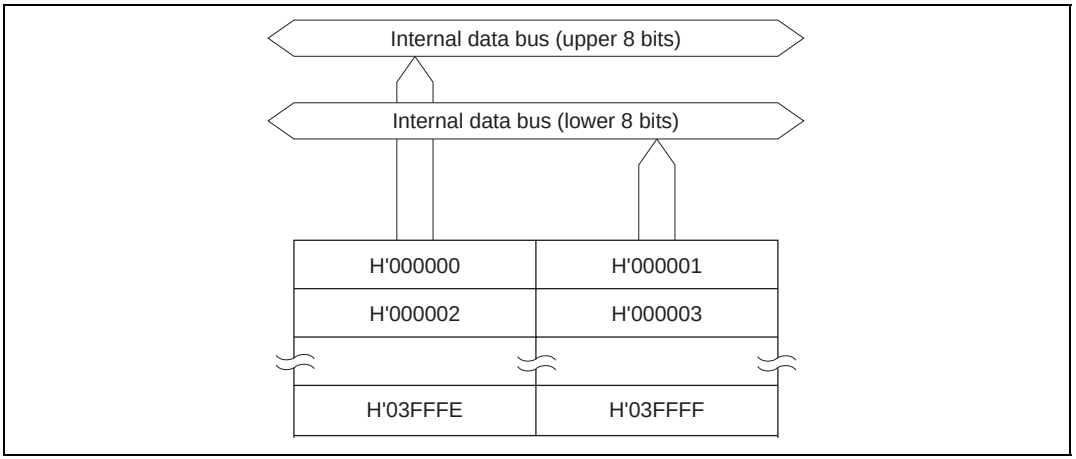


Figure 19.1 Block Diagram of ROM (256 kbytes)

The operating mode of the chip is controlled by the mode pins and the BCRL register. The ROM-related registers are shown in table 19.1.

Table 19.1 ROM Registers

Register Name	Abbreviation	R/W	Initial Value	Address*
Mode control register	MDCR	R/W	Undefined	H'FF3B
Bus controller register	BCRL	R/W	Undefined	H'FED5

Note: * Lower 16 bits of the address.

19.2 Register Descriptions

19.2.1 Mode Control Register (MDCR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	—	MDS2	MDS1	MDS0
Initial value	:	1	0	0	0	0	—*	—*	—*
R/W	:	—	—	—	—	—	R	R	R

Note: * Determined by pins MD2 to MD0.

MDCR is an 8-bit read-only register used to monitor the current operating mode of the chip.

Bit 7—Reserved: This bit cannot be modified and is always read as 1.

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

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

Bit	:	7	6	5	4	3	2	1	0
		BRLE	BREQOE	EAE	—	DDS	—	WDBE	WAITE
Initial value	:	0	0	1	1	1	1	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Enabling or disabling of part of the on-chip ROM area in the chip can be selected by means of the EAE bit in BCRL. For details of the other bits in BCRL, see section 6.2.5, Bus Control Register L (BCRL).

Bit 5—External Address Enable (EAE): Selects whether addresses H'010000 to H'03FFFF*2 are to be internal addresses or external addresses.

Bit 5	Description	
	H8S/2339, H8S/2338	**H8S/2337**
0	On-chip ROM	Addresses H'010000 to H'01FFFF are on-chip ROM or address H'020000 to H'03FFFF are reserved area*1
1	Addresses H'010000 to H'03FFFF*2 are external addresses in external expanded mode or reserved area*1 in single-chip mode (Initial value)	

Notes: 1. Do not access a reserved area.
2. Addresses H'010000 to H'05FFFF in the H8S/2339.

19.3 Operation

The on-chip ROM is connected to the CPU by a 16-bit data bus, and both byte and word data can be accessed in one state. Even addresses are connected to the upper 8 bits, and odd addresses to the lower 8 bits. Word data must start at an even address.

The on-chip ROM is enabled and disabled by setting the mode pins (MD2 to MD0) and the EAE bit in BCRL. These settings are shown in tables 19.2 and 19.3.

		Mode Pins				BCRL	
Mode	**Operating Mode**	**FWE**	**MD2**	**MD1**	**MD0**	**EAE**	**On-Chip ROM**
1	—	0	0	0	1	—	—
2				1	0		
3					1		
4	Advanced expanded mode with on-chip ROM disabled		1	0	0	—	Disabled
5	Advanced expanded mode with on-chip ROM disabled				1		
6	Advanced expanded mode with on-chip ROM enabled			1	0	0	Enabled (256 kbytes)*1*5
						1	Enabled (64 kbytes)
7	Advanced single-chip mode				1	0	Enabled (256 kbytes) *1*5
						1	Enabled (64 kbytes)
8	—	1	0	0	0	—	—
9					1		
10	Boot mode (advanced expanded mode with on-chip ROM enabled)*3			1	0	0	Enabled (256 kbytes) *2*5
						1	Enabled (64 kbytes)
11	Boot mode (advanced single-chip mode) *4				1	0	Enabled (256 kbytes) *2*5
						1	Enabled (64 kbytes)
12	—		1	0	0	—	—
13					1		
14	User program mode (advanced expanded mode with on-chip ROM enabled)*3			1	0	0	Enabled (256 kbytes) *1*5
						1	Enabled (64 kbytes)
15	User program mode (advanced single-chip mode)*4				1	0	Enabled (256 kbytes) *1*5
						1	Enabled (64 kbytes)

Notes: 1. Note that in modes 6, 7, 14, and 15, the on-chip ROM that can be used after a reset is the 64-kbyte area from H'000000 to H'00FFFF.

from H'000000 to H'00FFFF.

3. Apart from the fact that flash memory can be erased and programmed, operation is the same as in advanced expanded mode with on-chip ROM enabled.
4. Apart from the fact that flash memory can be erased and programmed, operation is the same as in advanced single-chip mode.
5. The capacity of on-chip ROM in the H8S/2338 F-ZTAT is 256 kbytes.

Table 19.3 Operating Modes and ROM (H8S/2339 F-ZTAT and Mask ROM Version)

		Mode Pins			BCRL	
Mode	**Operating Mode**	**MD2**	**MD1**	**MD0**	**EAE**	**On-Chip ROM**
1	—	0	0	1	—	—
2*3			1	0		
3*3				1		
4	Advanced expanded mode with on-chip ROM disabled	1	0	0	—	Disabled
5	Advanced expanded mode with on-chip ROM disabled			1		
6	Advanced expanded mode with on-chip ROM enabled		1	0	0	Enabled (256 kbytes)*1*2
					1	Enabled (64 kbytes)
7	Advanced single-chip mode			1	0	Enabled (256 kbytes)*1*2
					1	Enabled (64 kbytes)

Notes: 1. Note that in modes 6 and 7, the on-chip ROM that can be used after a reset is the 64-kbyte area from H'000000 to H'00FFFF.

2. The amount of on-chip RAM differs depending on the product. Refer to section 3.5, Memory Map in Each Operating Mode, for details.
3. Boot mode in the H8S/2339 F-ZTAT.
   See table 19.9, for information on H8S/2339 F-ZTAT user boot modes. See table 19.9, for information on H8S/2339 F-ZTAT user program modes.

19.4.1 Features

The H8S/2339 F-ZTAT has 384 kbytes of on-chip flash memory. The features of the flash memory are summarized below.

- 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. Erasing is performed by block erase (in single-block units). To erase the entire flash memory, the individual blocks must be erased sequentially. Block erasing can be performed as required on 4-kbyte, 32-kbyte, and 64-kbyte blocks.
- Programming/erase times

  The flash memory programming time is 10.0 ms (typ.) for simultaneous 128-byte programming, equivalent to 78 µs (typ.) per byte, and the erase time is 50 ms (typ.).
- Reprogramming capability

  The flash memory can be reprogrammed min. 100 times.
- On-board programming modes

  There are two modes in which flash memory can be programmed/erased/verified on-board:
  - Boot mode
  - User program mode
- Automatic bit rate adjustment

  With data transfer in boot mode, the bit rate of the chip can be automatically adjusted to match the transfer bit rate of the host.
- Flash memory emulation by RAM

  Part of the RAM area can be overlapped onto flash memory, to emulate flash memory updates in real time.
- Protect modes

  There are three protect modes, hardware, software, and error protect, which allow protected status to be designated for flash memory program/erase/verify operations.

well as in on-board programming mode.

19.4.2 Overview

Block Diagram

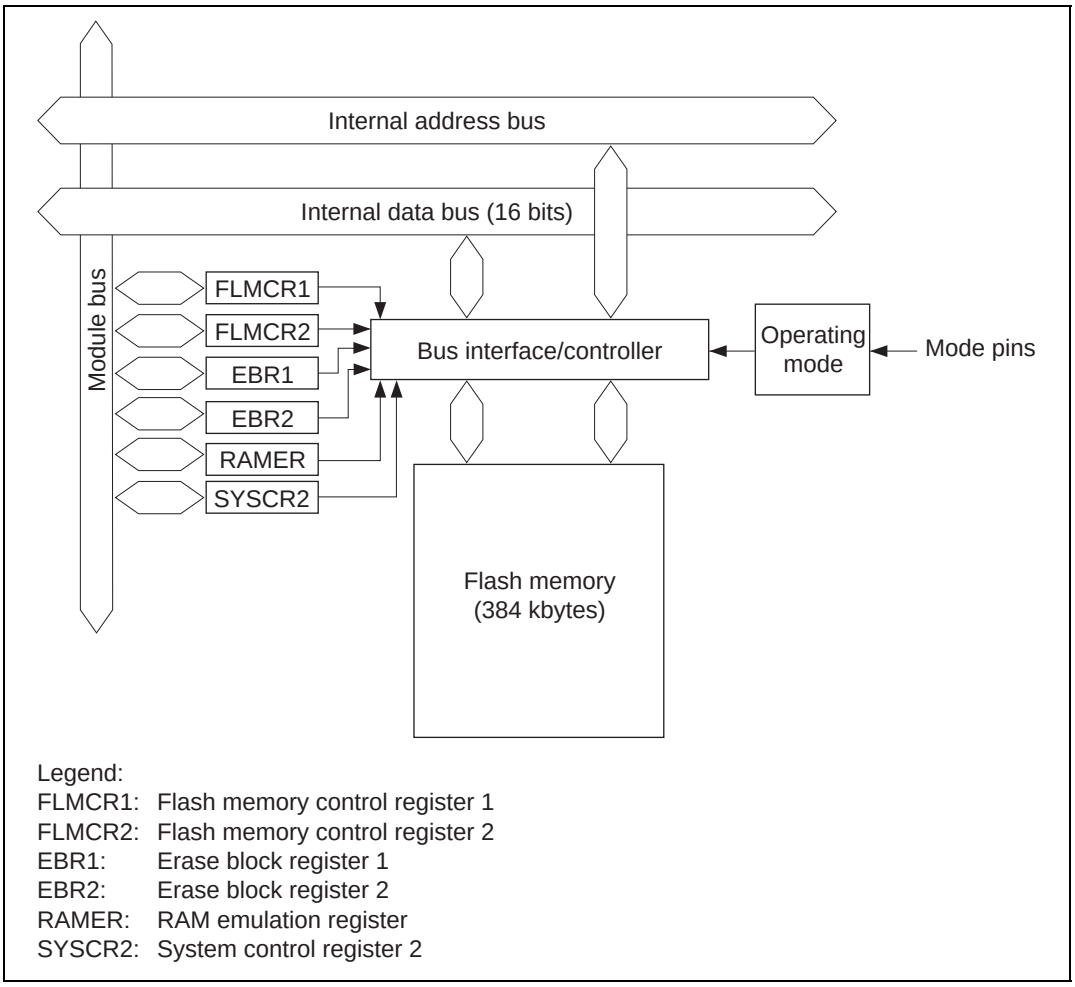
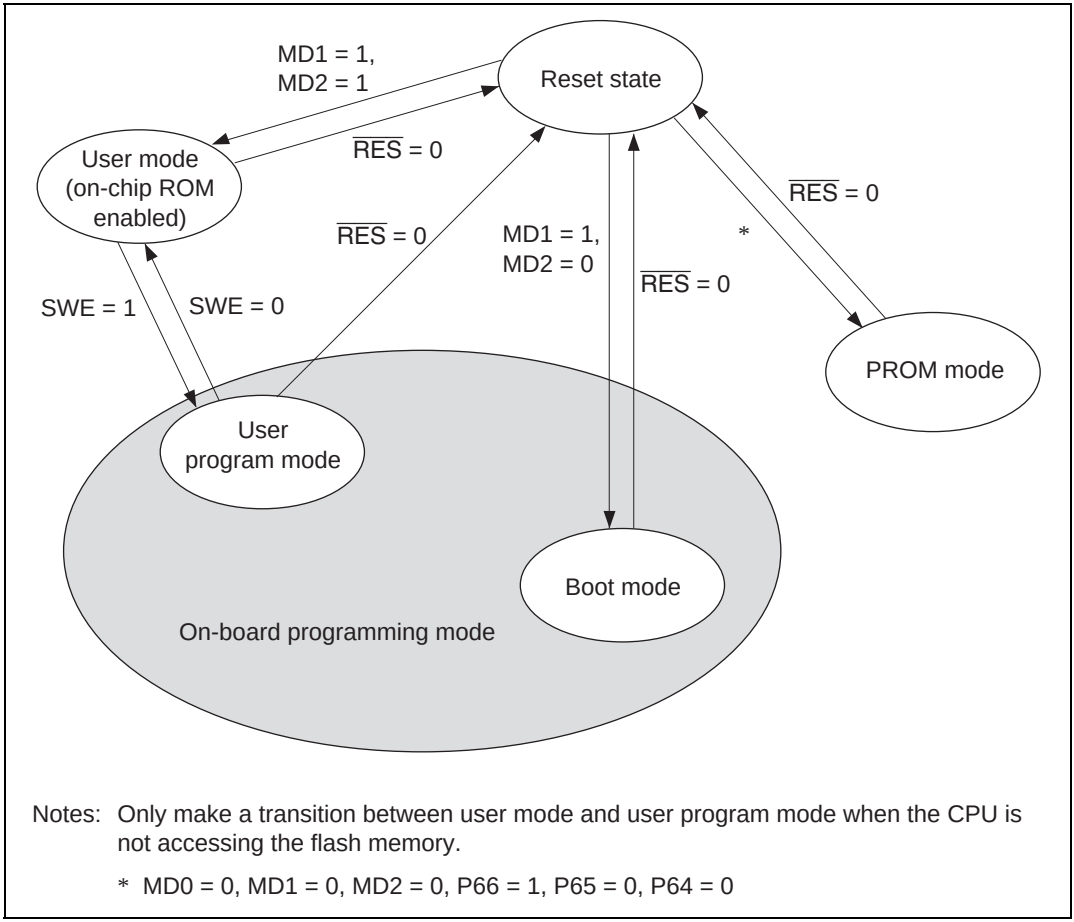


Figure 19.2 Block Diagram of Flash Memory

Mode Transitions: When the mode pins are set in the reset state and a reset-start is executed, the chip enters one of the operating modes shown in figure 19.3. In user mode, flash memory can be read but not programmed or erased.

Flash memory can be programmed and erased in boot mode, user program mode, and PROM mode.



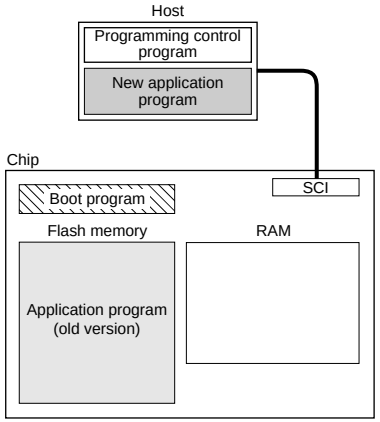
Notes: Only make a transition between user mode and user program mode when the CPU is not accessing the flash memory.

* MD0 = 0, MD1 = 0, MD2 = 0, P66 = 1, P65 = 0, P64 = 0

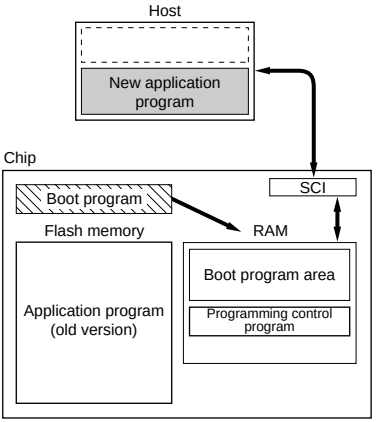
Figure 19.3 Flash Memory Mode Transitions

- Boot mode

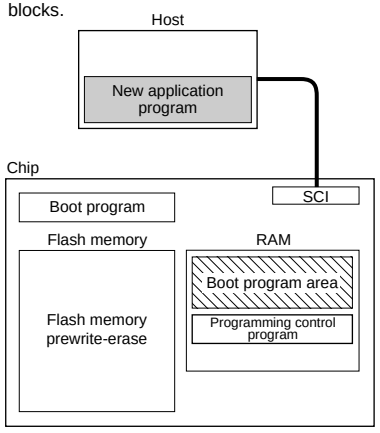
1. Initial state
   The old program version or data remains written in the flash memory. The user should prepare the programming control program and new application program beforehand in the host.



2. Programming control program transfer
   When boot mode is entered, the boot program in the chip (originally incorporated in the chip) is started and the programming control program in the host is transferred to RAM via SCI communication. The boot program required for flash memory erasing is automatically transferred to the RAM boot 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, entire flash memory erasure is performed, without regard to blocks.



4. Writing new application program
   The programming control program transferred from the host to RAM is executed, and the new application program in the host is written into the flash memory.

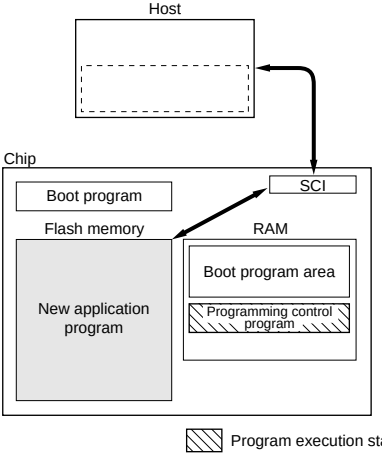
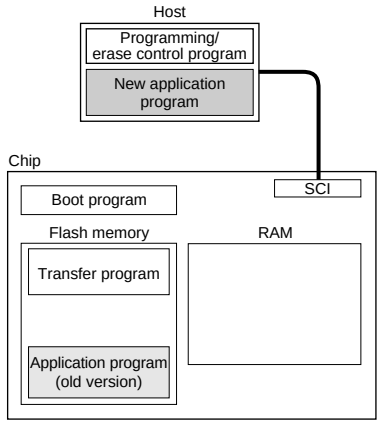
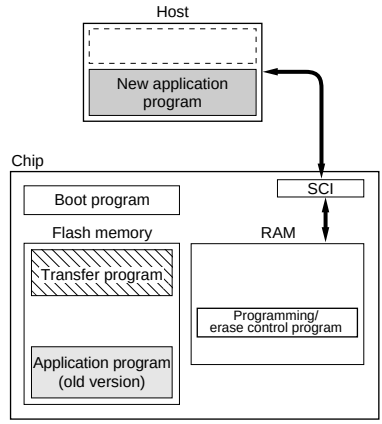


Figure 19.4 Boot Mode

1. Initial state
   (1) The program that will transfer the programming/erase control program to on-chip RAM should be written into the flash memory by the user beforehand. (2) The programming/erase control program should be prepared in the host or in the flash memory.

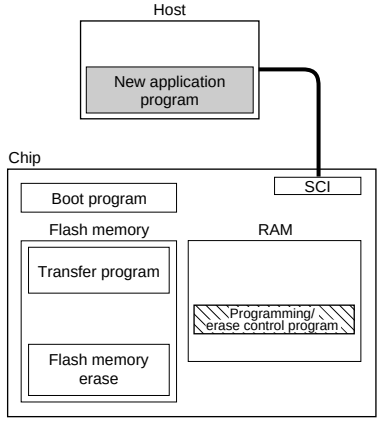
2. Programming/erase control program transfer
   Executes the transfer program in the flash memory, and transfers the programming/erase control program to RAM.





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 host is written into the erased flash memory blocks. Do not write to unerased blocks.



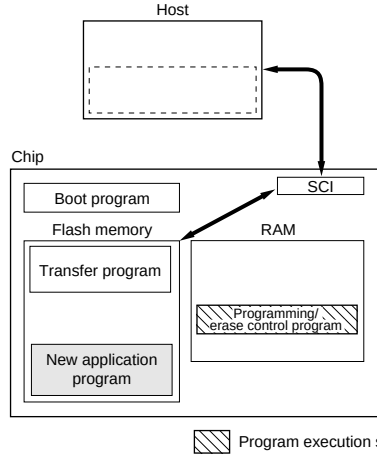


Figure 19.5 User Program Mode (Example)

Reading Overlap RAM Data in User Mode and User Program Mode: Emulation should be performed in user mode or user program mode. When the emulation block set in RAMER is accessed while the emulation function is being executed, data written in the overlap RAM is read.

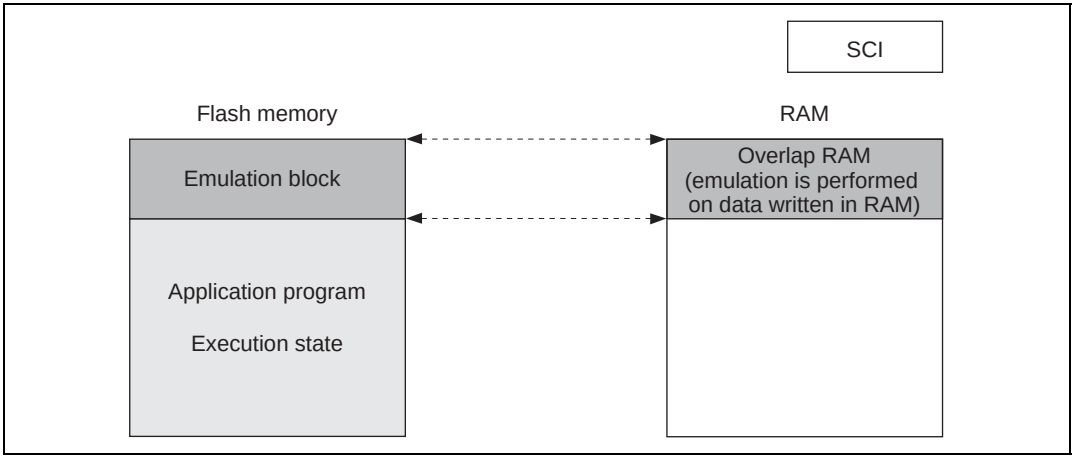


Figure 19.6 Reading Overlap RAM Data in User Mode and User Program Mode

flash memory.

When the programming control program is transferred to RAM, ensure that the transfer destination and the overlap RAM do not overlap, as this will cause data in the overlap RAM to be rewritten.

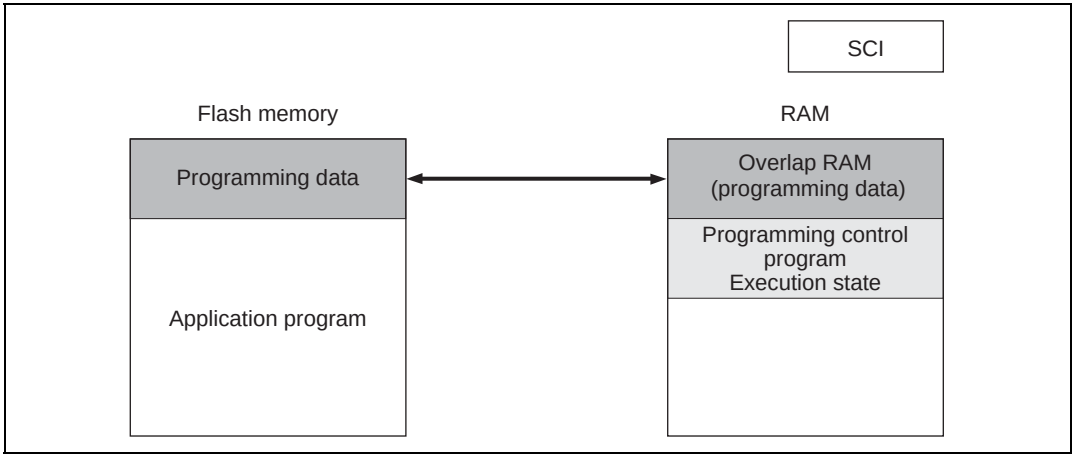


Figure 19.7 Writing Overlap RAM Data in User Program Mode

19.4.6 Differences between Boot Mode and User Program Mode

Table 19.4 Differences between Boot Mode and User Program Mode

	Boot Mode	User Program Mode
Entire memory erase	Yes	Yes
Block erase	No	Yes
Programming control program*	Program/program-verify	Erase/erase-verify/program/ program-verify/emulation

Note: * To be provided by the user, in accordance with the recommended algorithm.

The flash memory is divided into five 64-kbyte blocks, one 32-kbyte block, and eight 4-kbyte blocks.

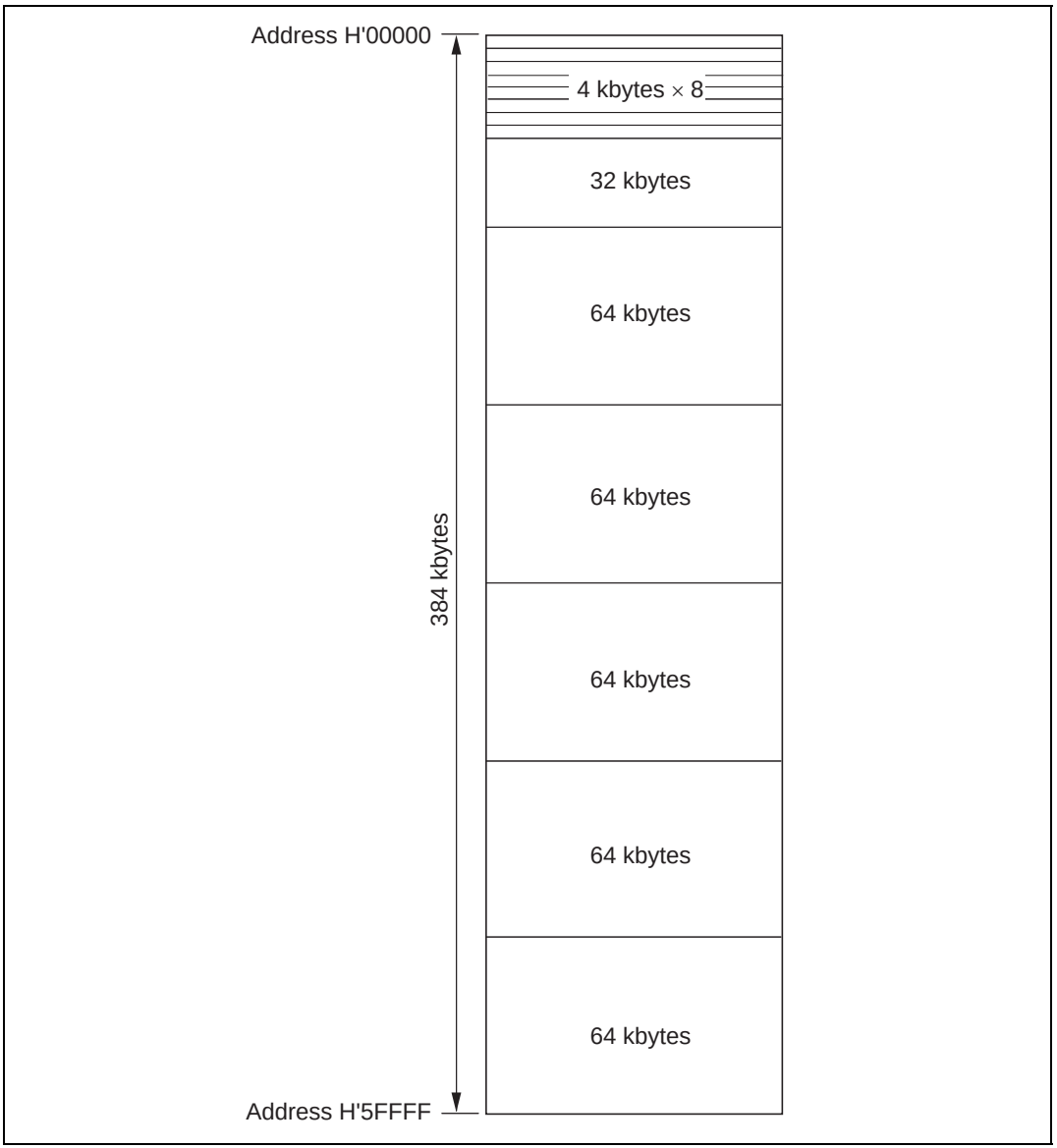


Figure 19.8 Flash Memory Block Configuration

The flash memory is controlled by means of the pins shown in table 19.5.

Table 19.5 Flash Memory Pins

Pin Name	Abbreviation	I/O	Function
Reset	$\overline{\text{RES}}$	Input	Reset
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 64	P64	Input	Sets MCU operating mode in PROM mode
Port 65	P65	Input	Sets MCU operating mode in PROM mode
Port 66	P66	Input	Sets MCU operating mode in PROM mode
Transmit data	TxD1	Output	Serial transmit data output
Receive data	RxD1	Input	Serial receive data input

The registers used to control the on-chip flash memory when enabled are shown in table 19.6. In order to access the FLMCR1, FLMCR2, EBR1, and EBR2 registers, the FLSHE bit must be set to 1 in SYSCR2 (except RAMER).

Table 19.6 Flash Memory Registers

Register Name	Abbreviation	R/W	Initial Value	Address*1
Flash memory control register 1	FLMCR1*5	R/W*3	H'80	H'FFC8*2
Flash memory control register 2	FLMCR2*5	R/W*3	H'00	H'FFC9*2
Erase block register 1	EBR1*5	R/W*3	H'00*4	H'FFCA*2
Erase block register 2	EBR2*5	R/W*3	H'00*4	H'FFCB*2
System control register 2	SYSCR2*6	R/W	H'00	H'FF42
RAM emulation register	RAMER	R/W	H'00	H'FEDB

Notes: 1. Lower 16 bits of the address.
2. Flash memory. Registers selection is performed by the FLSHE bit in system control register 2 (SYSCR2).
3. In modes in which the on-chip flash memory is disabled, a read will return H'00, and writes are invalid.
4. If the SWE bit in FLMCR1 is not set, these registers are initialized to H'00.
5. FLMCR1, FLMCR2, EBR1, and EBR2 are 8-bit registers. Only byte accesses are valid for these registers, the access requiring 2 states.
6. The SYSCR2 register can only be used in the F-ZTAT version. In the mask ROM version this register will return an undefined value if read, and cannot be modified.

19.5.1 Flash Memory Control Register 1 (FLMCR1)

Bit	:	7	6	5	4	3	2	1	0
		FWE	SWE	ESU	PSU	EV	PV	E	P
Initial value	:	1	0	0	0	0	0	0	0
R/W	:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

FLMCR1 is an 8-bit register used for flash memory operating mode control. Program-verify mode or erase-verify mode is entered by setting SWE to 1, then setting the EV or PV bit. Program mode is entered by setting SWE to 1, then setting the PSU bit, and finally setting the P bit. Erase mode is entered by setting SWE to 1, then setting the ESU bit, and finally setting the E bit. FLMCR1 is initialized to H'80 by a reset, and in hardware standby mode and software standby mode. When on-chip flash memory is disabled, a read will return H'00, and writes are invalid.

Writing to bits ESU, PSU, EV, and PV in FLMCR1 is enabled only when SWE = 1; writing to the E bit is enabled only when SWE = 1, and ESU = 1; and writing to the P bit is enabled only when SWE = 1, and PSU = 1.

Bit 7—Flash Write Enable Bit (FWE): Sets hardware protection against flash memory programming/erasing. These bits cannot be modified and are always read as 1 in this model.

Bit 6—Software Write Enable Bit (SWE): Enables or disables flash memory programming and erasing. This bit should be set when setting FLMCR1 bits 5 to 0, EBR1 bits 7 to 0, and EBR2 bits 5 to 0.

When SWE = 1, the flash memory can only be read in program-verify or erase-verify mode.

Bit 6 SWE	Description	
0	Writes disabled	(Initial value)
1	Writes enabled	

Bit 5 ESU	Description	
0	Erase setup cleared	(Initial value)
1	Erase setup [Setting condition] When SWE = 1	

Bit 4—Program Setup Bit (PSU): Prepares for a transition to program mode. Do not set the SWE, ESU, EV, PV, E, or P bit at the same time.

Bit 4 PSU	Description	
0	Program setup cleared	(Initial value)
1	Program setup [Setting condition] When SWE = 1	

Bit 3—Erase-Verify (EV): Selects erase-verify mode transition or clearing. Do not set the SWE, ESU, PSU, PV, E, or P bit at the same time.

Bit 3 EV	Description	
0	Erase-verify mode cleared	(Initial value)
1	Transition to erase-verify mode [Setting condition] When SWE = 1	

Bit 2 PV	Description	
0	Program-verify mode cleared	(Initial value)
1	Transition to program-verify mode [Setting condition] When SWE = 1	

Bit 1—Erase (E): Selects erase mode transition or clearing. Do not set the SWE, ESU, PSU, EV, PV, or P bit at the same time.

Bit 1 E	Description	
0	Erase mode cleared	(Initial value)
1	Transition to erase mode [Setting condition] When SWE = 1, and ESU = 1	

Bit 0—Program (P): Selects program mode transition or clearing. Do not set the SWE, PSU, ESU, EV, PV, or E bit at the same time.

Bit 0 P	Description	
0	Program mode cleared	(Initial value)
1	Transition to program mode [Setting condition] When SWE = 1, and PSU = 1	

Bit	:	7	6	5	4	3	2	1	0
		FLER	—	—	—	—	—	—	—
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R	—	—	—	—	—	—	—

FLMCR2 is an 8-bit register that controls the flash memory operating modes. FLMCR2 is initialized to H'00 by a reset, and in hardware standby mode and software standby mode.

When on-chip flash memory is disabled, a read will return H'00 and writes are invalid.

Bit 7—Flash Memory Error (FLER): Indicates that an error has occurred during an operation on flash memory (programming or erasing). When FLER is set to 1, flash memory goes to the error-protection state.

Bit 7 FLER	Description	
0	Flash memory is operating normally	(Initial value)
	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 cannot be modified and are always read as 0.

Bit	:	7	6	5	4	3	2	1	0
EBR1		EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0
Initial value	:	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

EBR1 is an 8-bit register that specifies the flash memory erase area block by block. EBR1 is initialized to H'00 by a reset, in hardware standby mode and software standby mode, and the SWE bit in FLMCR1 is not set. When a bit in EBR1 is set, the corresponding block can be erased. Other blocks are erase-protected. Set only one bit in EBR1 and EBR2 together (setting more than one bit will automatically clear all EBR1 and EBR2 bits to 0). 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 19.7.

19.5.4 Erase Block Registers 2 (EBR2)

Bit	:	7	6	5	4	3	2	1	0
EBR2		—	—	EB13	EB12	EB11	EB10	EB9	EB8
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	—	—	R/W	R/W	R/W	R/W	R/W	R/W

EBR2 is an 8-bit register that specifies the flash memory erase area block by block. EBR2 is initialized to H'00 by a reset, in hardware standby mode and software standby mode, and when the SWE bit in FLMCR1 is not set. When a bit in EBR2 is set, the corresponding block can be erased. Other blocks are erase-protected. Set only one bit in EBR2 and EBR1 together (setting more than one bit will automatically clear all EBR1 and EBR2 bits to 0). Bits 7 and 6 are reserved; they are always read as 0 and cannot be modified. 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 19.7.

Block (Size)	Address
EB0 (4 kbytes)	H'000000 to H'000FFF
EB1 (4 kbytes)	H'001000 to H'001FFF
EB2 (4 kbytes)	H'002000 to H'002FFF
EB3 (4 kbytes)	H'003000 to H'003FFF
EB4 (4 kbytes)	H'004000 to H'004FFF
EB5 (4 kbytes)	H'005000 to H'005FFF
EB6 (4 kbytes)	H'006000 to H'006FFF
EB7 (4 kbytes)	H'007000 to H'007FFF
EB8 (32 kbytes)	H'008000 to H'00FFFF
EB9 (64 kbytes)	H'010000 to H'01FFFF
EB10 (64 kbytes)	H'020000 to H'02FFFF
EB11 (64 kbytes)	H'030000 to H'03FFFF
EB12 (64 kbytes)	H'040000 to H'04FFFF
EB13 (64 kbytes)	H'050000 to H'05FFFF

19.5.5 System Control Register 2 (SYSCR2)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	FLSHE	—	—	—
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	—	—	—	—	R/W	—	—	R/W

SYSCR2 is an 8-bit readable/writable register that performs on-chip flash memory control.

SYSCR2 is initialized to H'00 by a reset and in hardware standby mode.

SYSCR2 can only be used in the F-ZTAT version. In the mask ROM version this register will return an undefined value if read, and cannot be modified.

Bits 7 to 4—Reserved: These bits cannot be modified and are always read as 0.

enables the flash memory control registers to be read and written to. Clearing FLSHE to 0 designates these registers as unselected (the register contents are retained).

Bit 3 FLSHE	Description
0	Flash control registers are not selected for addresses H'FFFFC8 to H'FFFFCB (Initial value)
1	Flash control registers are selected for addresses H'FFFFC8 to H'FFFFCB

Bits 2 and 1—Reserved: These bits cannot be modified and are always read as 0.

Bit 0—Reserved: This bit should be written with 0.

19.5.6 RAM Emulation Register (RAMER)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	RAMS	RAM2	RAM1	RAM0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	—	—	—	—	R/W	R/W	R/W	R/W

RAMER specifies the area of flash memory to be overlapped with part of RAM when emulating real-time flash memory programming. RAMER is initialized to H'00 by a reset and in hardware standby mode. It is not initialized in software standby mode. RAMER settings should be made in user mode or user program mode.

Flash memory area divisions are shown in table 19.8. To ensure correct operation of the emulation function, the ROM for which RAM emulation is performed should not be accessed immediately after this register has been modified. Normal execution of an access immediately after register modification is not guaranteed.

Bits 7 to 4—Reserved: These bits cannot be modified and are always read as 0.

Bit 3 RAMS	Description	
0	Emulation not selected	(Initial value)
	Program/erase-protection of all flash memory blocks is disabled	
1	Emulation selected	
	Program/erase-protection of all flash memory blocks is enabled	

Bits 2 to 0—Flash Memory Area Selection (RAM2 to RAM0): These bits are used together with bit 3 to select the flash memory area to be overlapped with RAM. (See table 19.8.)

Table 19.8 Flash Memory Area Divisions

RAM Area	Block Name	RAMS	RAM2	RAM1	RAM0
H'FFDC00 to H'FFEBFF	RAM area, 4 kbytes	0	*	*	*
H'000000 to H'000FFF	EB0 (4 kbytes)	1	0	0	0
H'001000 to H'001FFF	EB1 (4 kbytes)	1	0	0	1
H'002000 to H'002FFF	EB2 (4 kbytes)	1	0	1	0
H'003000 to H'003FFF	EB3 (4 kbytes)	1	0	1	1
H'004000 to H'004FFF	EB4 (4 kbytes)	1	1	0	0
H'005000 to H'005FFF	EB5 (4 kbytes)	1	1	0	1
H'006000 to H'006FFF	EB6 (4 kbytes)	1	1	1	0
H'007000 to H'007FFF	EB7 (4 kbytes)	1	1	1	1

*: Don't care

When pins are set to on-board programming mode, program/erase/verify operations can be performed on the on-chip flash memory. There are two on-board programming modes: boot mode and user program mode. The pin settings for transition to each of these modes are shown in table 19.9. For a diagram of the transitions to the various flash memory modes, see figure 19.3.

Table 19.9 Setting On-Board Programming Modes

Mode		Pins		
MCU Mode	**CPU Operating Mode**	**MD2**	**MD1**	**MD0**
Boot mode	Advanced expanded mode with on-chip ROM enabled	0	1	0
	Advanced single-chip mode			1
User program mode*	Advanced expanded mode with on-chip ROM enabled	1	1	0
	Advanced single-chip mode			1

Note: * Normally, user mode should be used. Set the SWE bit to 1 to make a transition to user program mode before performing a program/erase/verify operation.

When boot mode is used, the flash memory programming control program must be prepared in the host beforehand. The channel 1 SCI to be used is set to asynchronous mode.

When a reset-start is executed after the H8S/2339 F-ZTAT chip's pins have been set to boot mode, the boot program built into the chip is started and the programming control program prepared in the host is serially transmitted to the chip via the SCI. In the chip, the programming control program received via the SCI is written into the programming control program area in on-chip RAM. After the transfer is completed, control branches to the start address of the programming control program area and the programming control program execution state is entered (flash memory programming is performed).

The transferred programming control program must therefore include coding that follows the programming algorithm given later.

The system configuration in boot mode is shown in figure 19.9, and the boot program mode execution procedure in figure 19.10.

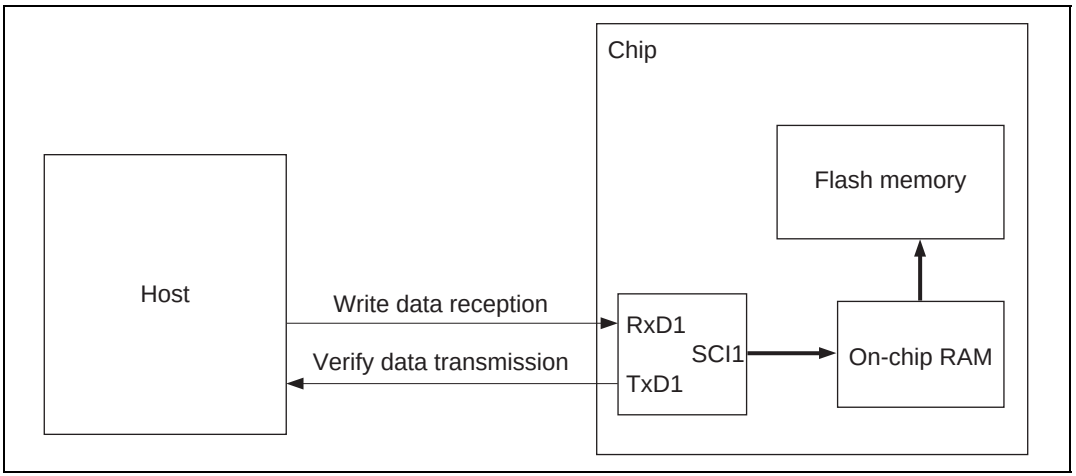
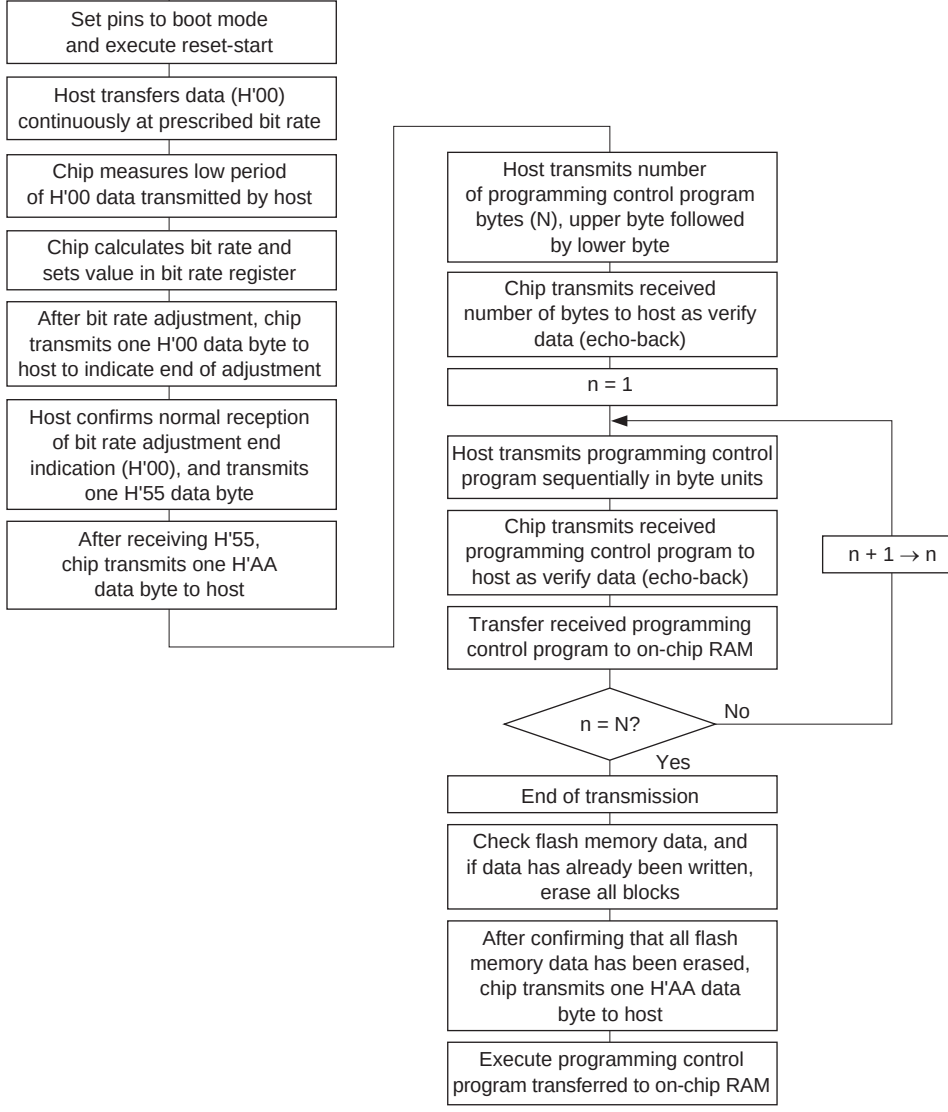


Figure 19.9 System Configuration in Boot Mode



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 operations are halted.

Figure 19.10 Boot Mode Execution Procedure

continuously from the host. The SCI transmit/receive format should be set as follows: 8-bit data, 1 stop bit, no parity. The chip calculates the bit rate of the transmission from the host from the measured low period, and transmits one H'00 byte to the host to indicate the end of bit rate adjustment. The host should confirm that this adjustment end indication (H'00) has been received normally, and transmit one H'55 byte to the chip. If reception cannot be performed normally, initiate boot mode again (reset), and repeat the above operations. Depending on the host's transmission bit rate and the chip's system clock frequency, there will be a discrepancy between the bit rates of the host and the chip. To ensure correct SCI operation, the host's transfer bit rate should be set to 9,600 or 19,200 bps.

Table 19.10 shows typical host transfer bit rates and system clock frequencies for which automatic adjustment of the MCU's bit rate is possible. The boot program should be executed within this system clock range.

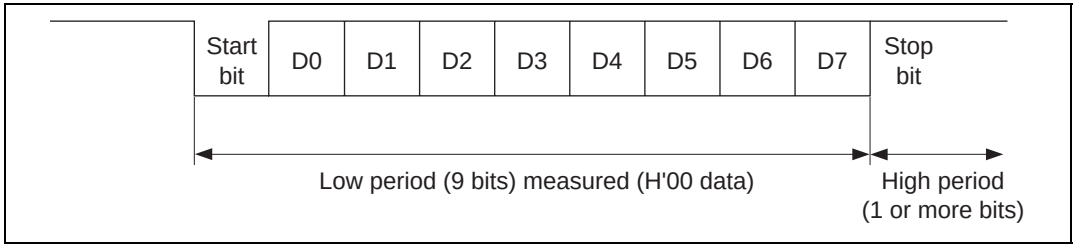
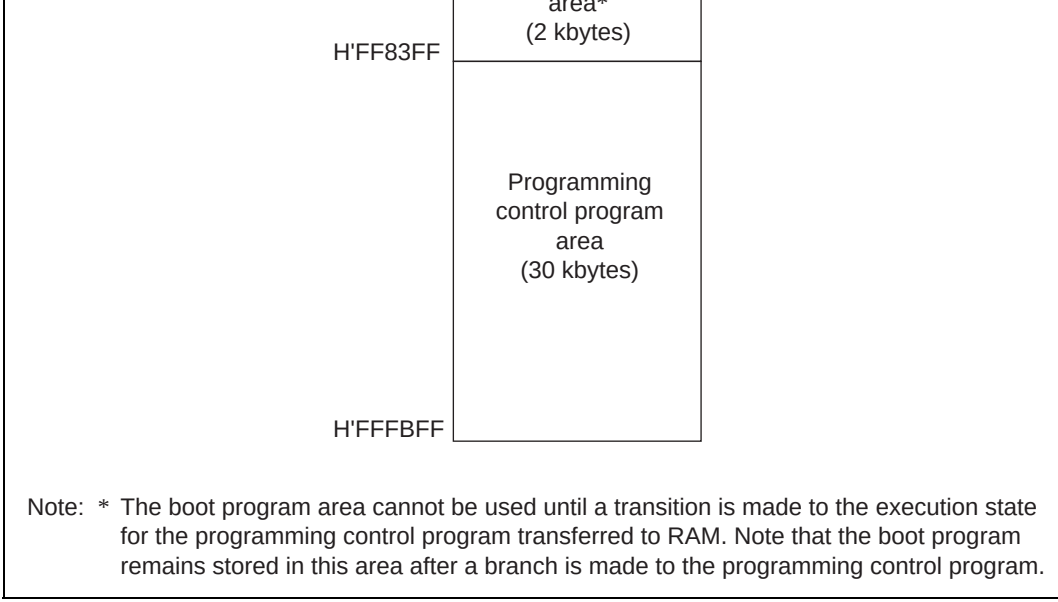


Figure 19.11 Automatic SCI Bit Rate Adjustment

Table 19.10 System Clock Frequencies for which Automatic Adjustment of H8S/2339 F-ZTAT Bit Rate is Possible

Host Bit Rate	System Clock Frequency for which Automatic Adjustment of H8S/2339 F-ZTAT Bit Rate is Possible
19,200 bps	16 MHz to 25 MHz
9,600 bps	8 MHz to 25 MHz

On-Chip RAM Area Divisions in Boot Mode: In boot mode, the 2-kbyte area from H'FF7C00 to H'FF83FF is reserved for use by the boot program, as shown in figure 19.12. The area to which the programming control program is transferred is H'FF8400 to H'FFFBFF. The boot program area can be used when the programming control program transferred into RAM enters the execution state. A stack area should be set up as required.



Note: * The boot program area cannot be used until a transition is made to the execution state for the programming control program transferred to RAM. Note that the boot program remains stored in this area after a branch is made to the programming control program.

Figure 19.12 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 input at the SCI's RxD1 pin. The reset should end with RxD1 high. After the reset ends, it takes approximately 100 states before the chip is ready to measure the low-level period of the RxD1 pin.
- In boot mode, if any data has been programmed into the flash memory (if all data is not 1), all flash memory blocks are erased. Boot mode is for use when user program mode is unavailable, such as the first time on-board programming is performed, or if the program activated in user program mode is accidentally erased.
- Interrupts cannot be used while the flash memory is being programmed or erased.
- The RxD1 and TxD1 pins should be pulled up on the board.

the RE and TE bits in SCR to 0), but the adjusted bit rate value remains set in BRR. The transmit data output pin, TxD1, goes to the high-level output state (P31DDR = 1, P31DR = 1).

The contents of the CPU's internal general registers are undefined at this time, so these registers must be initialized immediately after branching to the programming control program. In particular, since the stack pointer (SP) is used implicitly in subroutine calls, etc., a stack area must be specified for use by the programming control program.

Initial settings must also be made for the other on-chip registers.

- Boot mode can be entered by making the pin settings shown in table 19.9 and executing a reset-start.

  Boot mode can be cleared by driving the reset pin low, waiting at least 20 states, then setting the mode pins, and executing reset release*1. Boot mode can also be cleared by a WDT overflow reset.

  Do not change the mode pin input levels in boot mode.

- If the mode pin input levels are changed (for example, from low to high) during a reset, the state of ports with multiplexed address functions and bus control output pins ($\overline{AS}$, $\overline{RD}$, $\overline{HWR}$) will change according to the change in the microcomputer's operating mode*2.

  Therefore, care must be taken to make pin settings to prevent these pins from becoming output signal pins during a reset, or to prevent collision with signals outside the microcomputer.

Notes: 1. Mode pins input must satisfy the mode programming setup time (t_{MDS} = 200 ns) with respect to the reset release timing.
2. See section 9, I/O Ports.

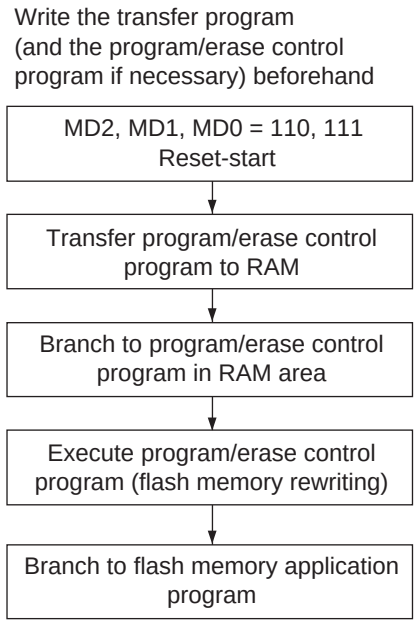
19.6.2 User Program Mode

When set to user program mode, the chip can program and erase its flash memory by executing a user program/erase control program. Therefore, on-board reprogramming of the on-chip flash memory can be carried out by providing on-board means supply of programming data, and storing a program/erase control program in part of the program area if necessary.

To select user program mode, select a mode that enables the on-chip flash memory (mode 6 or 7). In this mode, on-chip supporting modules other than flash memory operate as they normally would in modes 6 and 7.

The flash memory itself cannot be read while the SWE bit is set to 1 to perform programming or erasing, so the control program that performs programming and erasing should be run in on-chip

Figure 19.13 shows the procedure for executing the program/erase control program when transferred to on-chip RAM.



Note: The watchdog timer should be activated to prevent overprogramming or overerasing due to program runaway, etc.

Figure 19.13 User Program Mode Execution Procedure

In the on-board programming modes, flash memory programming and erasing is performed by software, using the CPU. There are four flash memory operating modes: program mode, erase mode, program-verify mode, and erase-verify mode. Transitions to these modes can be made by setting the PSU, ESU, P, E, PV, and EV bits in FLMCR1.

The flash memory cannot be read while being programmed or erased. Therefore, the program that controls flash memory programming/erasing (the programming control program) should be located and executed in on-chip RAM or external memory. When the program is located in external memory, an instruction for programming the flash memory and the following instruction should be located in on-chip RAM. The DMAC or DTC should not be activated before or after the instruction for programming the flash memory is executed.

Notes: 1. Operation is not guaranteed if setting/resetting of the SWE, ESU, PSU, EV, PV, E, and P bits in FLMCR1 is executed by a program in flash memory.
2. Perform programming in the erased state. Do not perform additional programming on previously programmed addresses.

19.7.1 Program Mode

Follow the procedure shown in the program/program-verify flowchart in figure 19.14 to write data or programs to flash memory. Performing program operations according to this flowchart will enable data or programs to be written to flash memory without subjecting the device to voltage stress or sacrificing program data reliability. Programming should be carried out 128 bytes at a time.

For the wait times (x, y, z1, z2, z3 α, ß, γ, ε, η, and θ) after bits are set or cleared in flash memory control register 1 (FLMCR1) and the maximum number of programming operations (N), see section 22.2.6, Flash Memory Characteristics.

Following the elapse of (x) µs or more after the SWE bit is set to 1 in flash memory control register 1 (FLMCR1), 128-byte program data is stored in the program data area and reprogram data area, and the 128-byte data in the reprogram data area is written consecutively to the write addresses. The lower 8 bits of the first address written to must be H'00 or H'80. 128 consecutive byte data transfers are performed. The program address and program data are latched in the flash memory. 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.

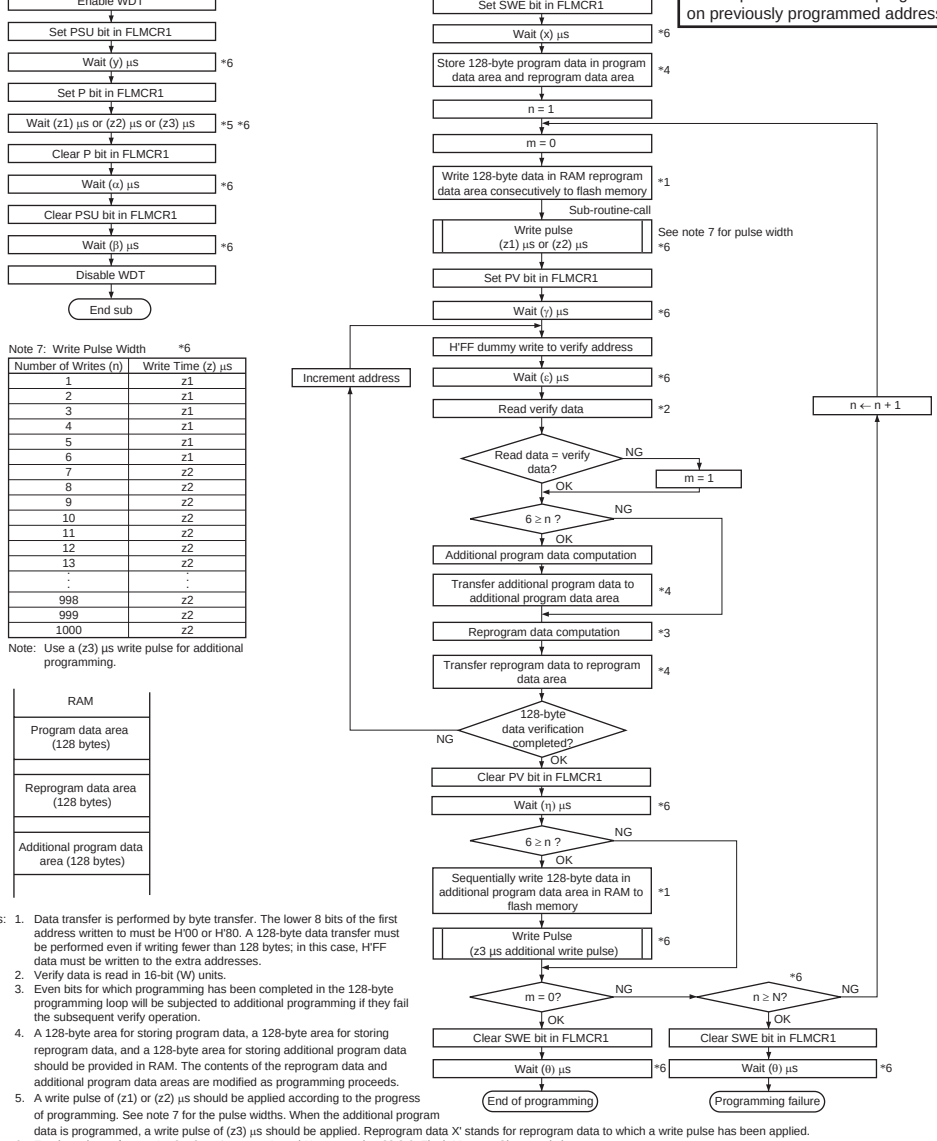
Next, the watchdog timer is set to prevent overprogramming in the event of program runaway, etc. Set a value greater than (y + z2 + α + β) µs as the WDT overflow period. After this, preparation for program mode (program setup) is carried out by setting the PSU bit in FLMCR1, and after the

programming time according to the table in the programming flowchart.

19.7.2 Program-Verify Mode

In program-verify mode, the data written in program mode is read to check whether it has been correctly written in the flash memory.

After the elapse of a given programming time, the programming mode is exited (the P bit in FLMCR1 is cleared to 0, then the PSU bit is cleared to 0 at least (α) µs later). Next, the watchdog timer is cleared after the elapse of (β) µs or more, and the operating mode is switched to program-verify mode by setting the PV bit in FLMCR1. Before reading in program-verify mode, a dummy write of H'FF data should be made to the addresses to be read. The dummy write should be executed after the elapse of (γ) µs or more. When the flash memory is read in this state (verify data is read in 16-bit units), the data at the latched address is read. Wait at least (ε) µs after the dummy write before performing this read operation. Next, the originally written data is compared with the verify data, and reprogram data is computed (see figure 19.14) and transferred to the reprogram data area. After 128 bytes of data have been verified, exit program-verify mode, wait for at least (η) µs, then clear the SWE bit in FLMCR1 to 0, and wait again for at least (θ) µs. If reprogramming is necessary, set program mode again, and repeat the program/program-verify sequence as before. However, ensure that the program/program-verify sequence is not repeated more than (N) times on the same bits.



Note 7: Write Pulse Width *6

Number of Writes (n)	Write Time (z) µs
1	z1
2	z1
3	z1
4	z1
5	z1
6	z1
7	z2
8	z2
9	z2
10	z2
11	z2
12	z2
13	z2
.	.
.	.
998	z2
999	z2
1000	z2

Note: Use a (z3) µs write pulse for additional programming.

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 (W) units.
3. Even bits for which programming has been completed in the 128-byte programming loop will be subjected to additional programming 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 program data should be provided in RAM. The contents of the reprogram data and additional program data areas are modified as programming proceeds.
5. A write pulse of (z1) or (z2) µs should be applied according to the progress of programming. See note 7 for the pulse widths. When the additional program data is programmed, a write pulse of (z3) µs should be applied. Reprogram data X' stands for reprogram data to which a write pulse has been applied.
6. For the values of x, y, z1, z2, z3, α, β, γ, ε, η, θ, and N, see section 22.2.6, Flash Memory Characteristics.

Program Data Operation Chart

Original Data (D)	Verify Data (V)	Reprogram Data (X)	Comments
0	0	1	Programming completed
	1	0	Programming incomplete; reprogram
1	0	1	
	1		Still in erased state; no action

Additional Program Data Operation Chart

Reprogram Data (X')	Verify Data (V)	Additional Program Data (Y')	Comments
0	0	0	Additional programming executed
	1	1	Additional programming not executed
1	0		Additional programming not executed
	1		Additional programming not executed

Figure 19.14 Program/Program-Verify Flowchart

Flash memory erasing should be performed block by block following the procedure shown in the erase/erase-verify flowchart (single-block erase) shown in figure 19.15.

For the wait times (x, y, z, α, ß, γ, ε, η, θ) after bits are set or cleared in flash memory control register 1 (FLMCR1) and the maximum number of programming operations (N), see section 22.2.6, Flash Memory Characteristics.

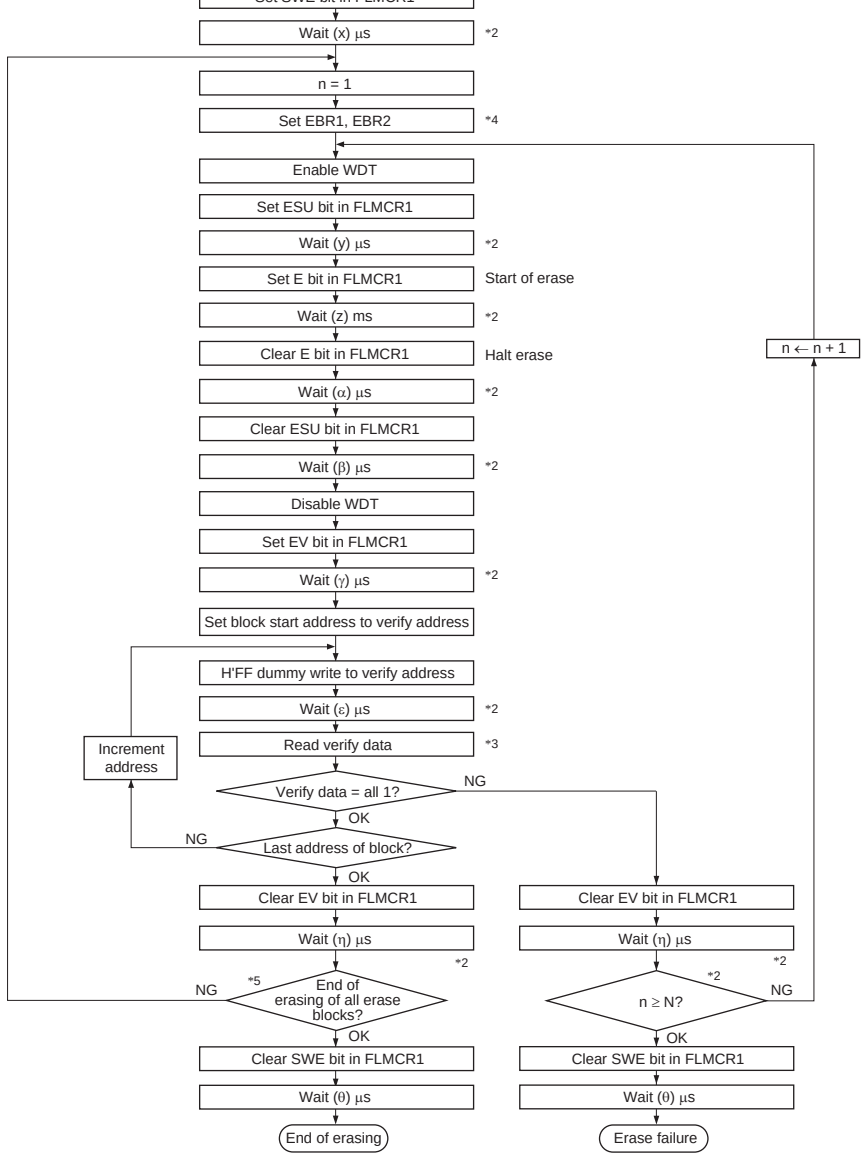
To perform data or program erasure, make a 1 bit setting for the flash memory area to be erased in erase block register 1 or 2 (EBR1 or EBR2) at least (x) µs after setting the SWE bit to 1 in flash memory control register 1 (FLMCR1). Next, the watchdog timer is set to prevent overerasing in the event of program runaway, etc. Set a value greater than (y + z + α + ß) ms as the WDT overflow period. After this, preparation for erase mode (erase setup) is carried out by setting the ESU bit in FLMCR1, and after the elapse of (y) µs or more, the operating mode is switched to erase mode by setting the E bit in FLMCR1. The time during which the E bit is set is the flash memory erase time. Ensure that the erase time does not exceed (z) ms.

Note: With flash memory erasing, prewriting (setting all data in the memory to be erased to 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 been correctly erased.

After the elapse of the erase time, erase mode is exited (the E bit in FLMCR1 is cleared to 0, then the ESU bit in FLMCR1 is cleared to 0 at least (α) µs later), the watchdog timer is cleared after the elapse of (β) µs or more, and the operating mode is switched to erase-verify mode by setting the EV bit in FLMCR1. Before reading in erase-verify mode, a dummy write of H'FF data should be made to the addresses to be read. The dummy write should be executed after the elapse of (γ) µs or more. When the flash memory is read in this state (verify data is read in 16-bit units), the data at the latched address is read. Wait at least (ε) µs after the dummy write before performing this read operation. If the read data has been erased (all 1), a dummy write is performed to the next address, and erase-verify is performed. If the read data has not been erased, set erase mode again, and repeat the erase/erase-verify sequence in the same way. However, ensure that the erase/erase-verify sequence is not repeated more than (N) times. When verification is completed, exit erase-verify mode, and wait for at least (η) µs. If erasure has been completed on all the erase blocks, clear the SWE bit in FLMCR1 to 0 and wait for at least (θ) µs. If there are any unerased blocks, make a 1 bit setting for the flash memory area to be erased, and repeat the erase/erase-verify sequence in the same way.



Notes: 1. Prewriting (setting erase block data to all 0) is not necessary.
2. The values of x, y, z, α, β, γ, ε, η, θ, and N are shown in the section 22.2.6, Flash Memory Characteristics.
3. Verify data is read in 16-bit (W) units.
4. Set only one bit in EBR1or EBR2. More than one bit cannot be set.
5. Erasing is performed in block units. To erase a number of blocks, the individual blocks must be erased sequentially.

Figure 19.15 Erase/Erase-Verify Flowchart

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

19.8.1 Hardware Protection

Hardware protection refers to a state in which programming/erasing of flash memory is forcibly disabled or aborted. Settings in flash memory control registers 1 and 2 (FLMCR1, FLMCR2) and erase block registers 1 and 2 (EBR1, EBR2) are reset. (See table 19.11.)

Table 19.11 Hardware Protection

		Functions	
Item	**Description**	**Program**	**Erase**
Reset/standby protection	• In a reset (including a WDT overflow 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 section 22.2.3, AC Characteristics.	Yes	Yes

19.8.2 Software Protection

Software protection can be implemented by setting the SWE bit in flash memory control register 1 (FLMCR1), erase block registers 1 and 2 (EBR1, EBR2), and the RAMS bit in the RAM emulation register (RAMER). When software protection is in effect, setting the P or E bit in FLMCR1 does not cause a transition to program mode or erase mode. (See table 19.12.)

Item	Description	Functions: Program	Functions: Erase
SWE bit protection	• Clearing the SWE bit to 0 in FLMCR1 sets the program/erase-protected state for all blocks. (Execute in on-chip RAM or external memory.)	Yes	Yes
Block specification protection	• Erase protection can be set for individual blocks by settings in erase block registers 1 and 2 (EBR1, EBR2). • Setting EBR1 and EBR2 to H'00 places all blocks in the erase-protected state.	—	Yes
Emulation protection	• Setting the RAMS bit to 1 in the RAM emulation register (RAMER) places all blocks in the program/erase-protected state.	Yes	Yes

19.8.3 Error Protection

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

If the MCU malfunctions during flash memory programming/erasing, the FLER bit is set to 1 in FLMCR2 and the error protection state is entered. The FLMCR1, FLMCR2, EBR1, and EBR2 settings are retained, but program mode or erase mode is aborted at the point at which the error occurred. Program mode or erase mode cannot be re-entered by re-setting the P or E bit. However, PV and EV bit setting is enabled, and a transition can be made to verify mode.

FLER bit setting conditions are as follows:

- When flash memory is read during programming/erasing (including a vector read or instruction fetch)
- Immediately after exception handling (excluding a reset) during programming/erasing
- When a SLEEP instruction (including software standby) is executed during programming/erasing
- When a bus master other than the CPU (the DMAC or DTC) has control of the bus during programming/erasing

Figure 19.16 shows the flash memory state transition diagram.

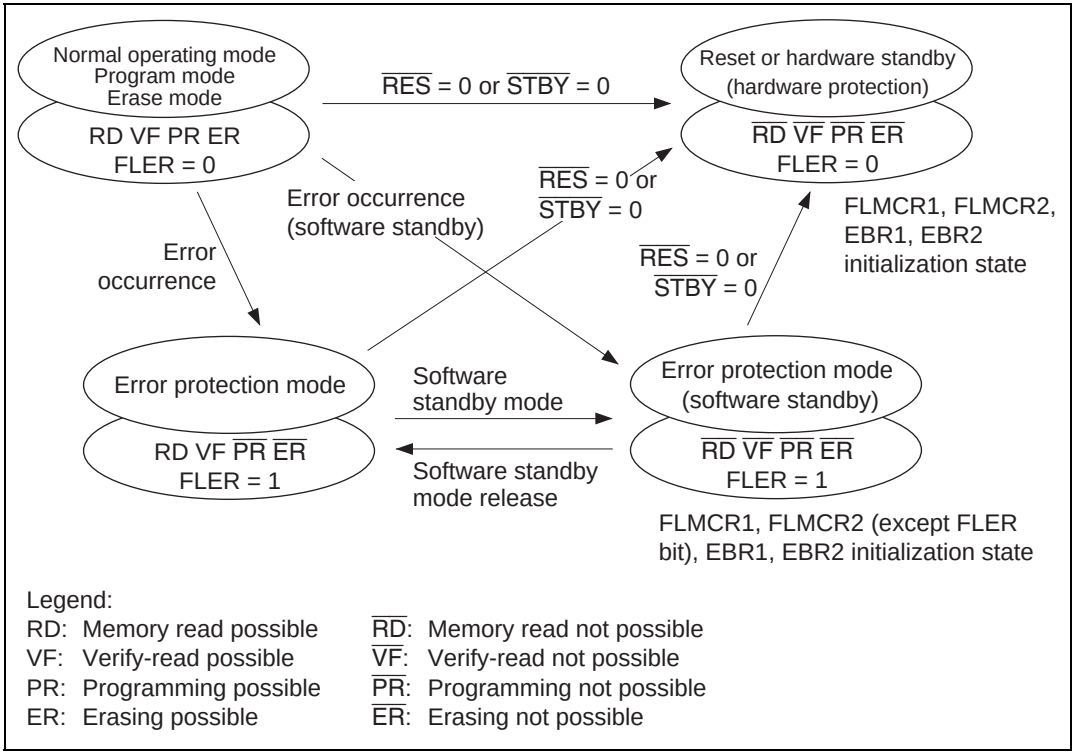


Figure 19.16 Flash Memory State Transitions

19.9.1 Emulation in RAM

Making a setting in the RAM emulation register (RAMER) enables part of RAM to be overlapped onto the flash memory area so that data to be written to flash memory can be emulated in RAM in real time. After the RAMER setting has been made, accesses can be made from the flash memory area or the RAM area overlapping flash memory. Emulation can be performed in user mode and user program mode. Figure 19.17 shows an example of emulation of real-time flash memory programming.

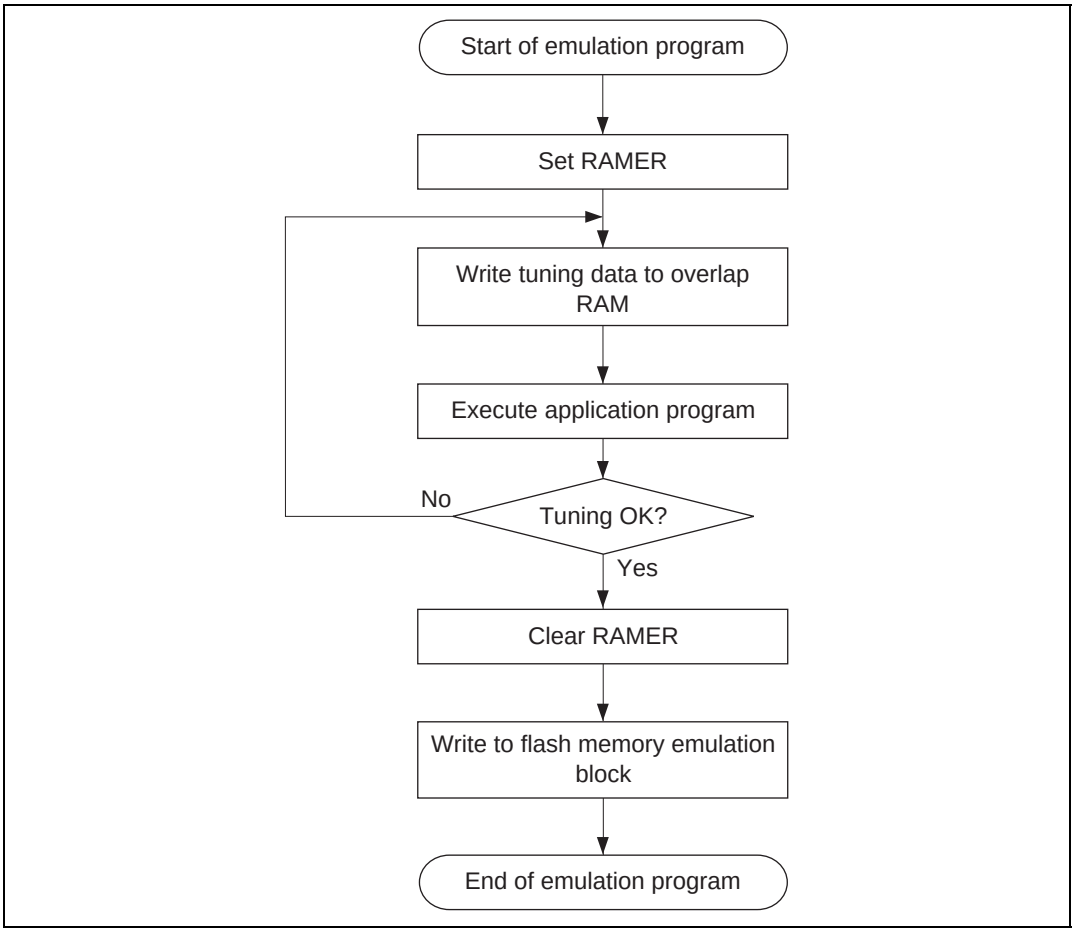


Figure 19.17 Flowchart for Flash Memory Emulation in RAM

An example in which flash memory block area EB1 is overlapped is shown below.

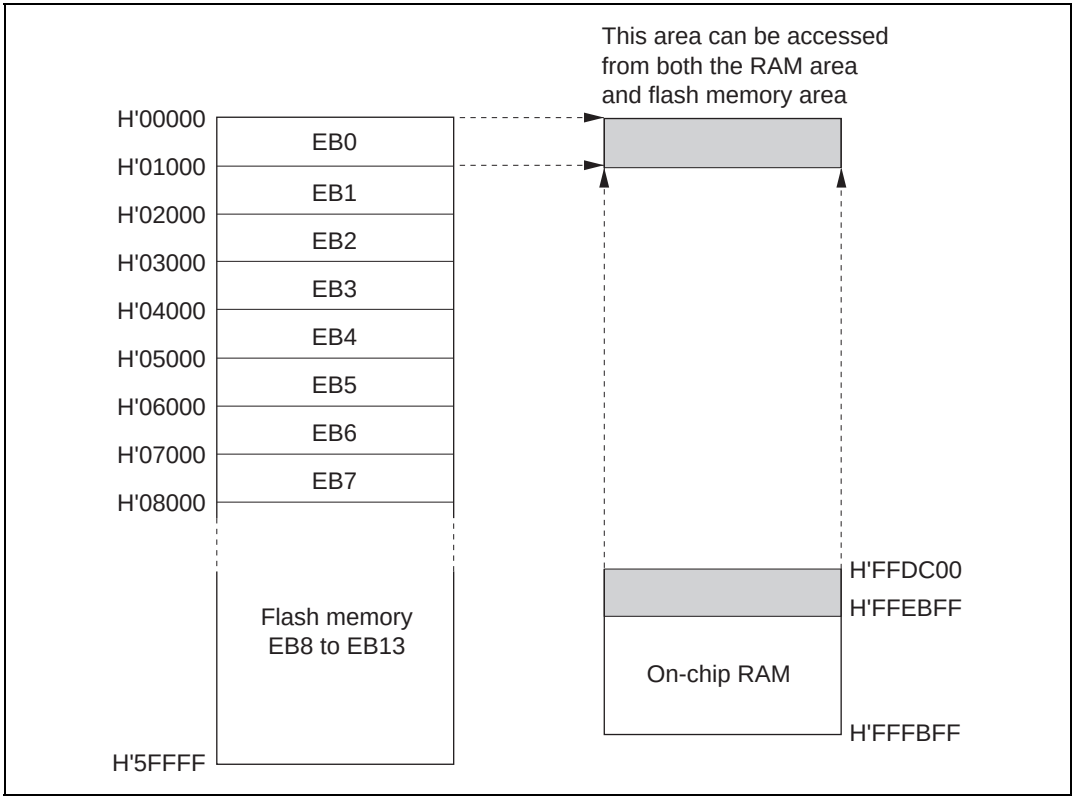


Figure 19.18 Example of RAM Overlap Operation

Example in which Flash Memory Block Area EB1 is Overlapped

1. Set bits RAMS, RAM2, RAM1, and RAM0 in RAMER to 1, 0, 0, 1, to overlap part of RAM onto the area (EB1) 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 RAM overlap.
4. The data written in the overlapping RAM is written into the flash memory space (EB1).

Notes: 1. When the RAMS bit is set to 1, program/erase protection is enabled for all blocks regardless of the value of RAM2, RAM1, and RAM0 (emulation protection). In this state, setting the P or E bit in flash memory control register 1 (FLMCR1) will not cause

2. A RAM area cannot be erased by execution of software in accordance with the erase algorithm while flash memory emulation in RAM is being used.
3. Block area EB0 includes the vector table. When performing RAM emulation, the vector table is needed by the overlap RAM.

19.10 Interrupt Handling when Programming/Erasing Flash Memory

All interrupts, including NMI input, are disabled when flash memory is being programmed or erased (when the P or E bit is set in FLMCR1), and while the boot program is executing in boot mode*1, to give priority to the program or erase operation. There are three reasons for this:

1. Interrupt during programming or erasing might cause a violation of the programming or erasing algorithm, with the result that normal operation could not be assured.
2. In the interrupt exception handling sequence during programming or erasing, the vector would not be read correctly*2, possibly resulting in MCU runaway.
3. If an interrupt occurred during boot program execution, it would not be possible to execute the normal boot mode sequence.

For these reasons, in on-board programming mode alone there are conditions for disabling interrupts, as an exception to the general rule. However, this provision does not guarantee normal erasing and programming or MCU operation. All interrupt requests, including NMI, must therefore be restricted inside and outside the MCU when programming or erasing flash memory. The NMI interrupt is also disabled in the error-protection state while the P or E bit remains set in FLMCR1.

Notes: 1. Interrupt requests must be disabled inside and outside the MCU until the programming control program has completed programming.
2. The vector may not be read correctly in this case for the following two reasons:
   - If flash memory is read while being programmed or erased (while the P or E bit is set in FLMCR1), correct read data will not be obtained (undetermined values will be returned).
   - If the interrupt entry in the vector table has not been programmed yet, interrupt exception handling will not be executed correctly.

19.11.1 PROM Mode Setting

Programs and data can be written and erased in PROM mode as well as in the on-board programming modes. In PROM mode, the on-chip ROM can be freely programmed using a PROM programmer that supports the Renesas microcomputer device type with 512-kbyte on-chip flash memory (FZTAT512V3A). Flash memory read mode, auto-program mode, auto-erase mode, and status read mode are supported with this device type. In auto-program mode, auto-erase mode, and status read mode, a status polling procedure is used, and in status read mode, detailed internal signals are output after execution of an auto-program or auto-erase operation.

Table 19.13 shows PROM mode pin settings.

Table 19.13 PROM Mode Pin Settings

Pin Names	Settings/External Circuit Connection
Mode pins: MD2, MD1, MD0	Low-level input
Mode setting pins: P66, P65, P64	High-level input to P66, low-level input to P65 and P64
$\overline{\text{STBY}}$ pin	High-level input (do not select hardware standby mode)
$\overline{\text{RES}}$ pin	Reset circuit
XTAL, EXTAL pins	Oscillator circuit
Other pins requiring setting: P32, P25	High-level input to P32, low-level input to P25

19.11.2 Socket Adapters and Memory Map

In PROM mode, a socket adapter is connected to the chip as shown in figure 19.20. Figure 19.19 shows the on-chip ROM memory map and figure 19.20 shows the socket adapter pin assignments.

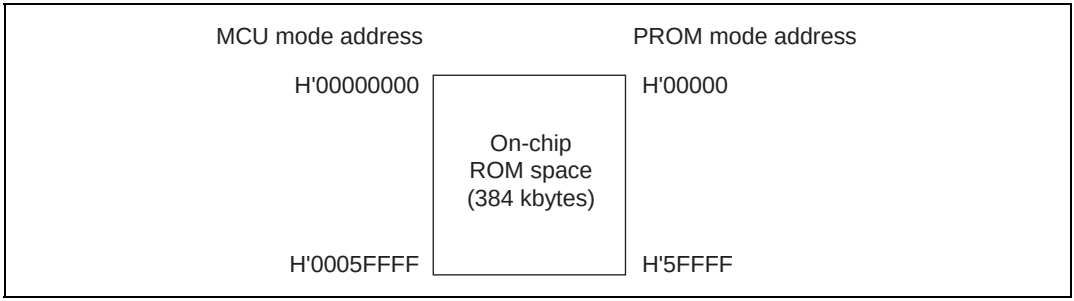
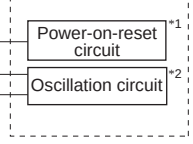


Figure 19.19 Memory Map in PROM Mode

5	A_0	21	A_0
6	A_1	22	A_1
7	A_2	23	A_2
8	A_3	24	A_3
10	A_4	25	A_4
11	A_5	26	A_5
12	A_6	27	A_6
13	A_7	28	A_7
14	A_8	29	A_8
15	A_9	31	A_9
16	A_{10}	32	A_{10}
17	A_{11}	33	A_{11}
19	A_{12}	34	A_{12}
20	A_{13}	35	A_{13}
21	A_{14}	36	A_{14}
22	A_{15}	37	A_{15}
23	A_{16}	38	A_{16}
24	A_{17}	39	A_{17}
25	A_{18}	10	A_{18}
52	D_8	19	I/O_0
53	D_9	18	I/O_1
54	D_{10}	17	I/O_2
55	D_{11}	16	I/O_3
57	D_{12}	15	I/O_4
58	D_{13}	14	I/O_5
59	D_{14}	13	I/O_6
60	D_{15}	12	I/O_7
83	$\overline{CE}$	2	$\overline{CE}$
84	$\overline{OE}$	20	$\overline{OE}$
82	$\overline{WE}$	3	$\overline{WE}$
97	EMLE*3	4	FWE
3, 36, 39, 61, 64, 89, 90, 91, 96, 113, 114	V_{CC}	1, 40	V_{CC}
9, 18, 27, 37, 38, 47, 56, 71, 81, 94, 123, 124, 135, 136, 137	V_{SS}	11, 30	V_{SS}
		5, 6, 7	NC
		8	A_{20}
		9	A_{19}
88	$\overline{RES}$		
92	XTAL		
93	EXTAL		
Other pins	NC (OPEN)		



Legend:
EMLE: Emulation enable
I/O_7 to I/O_0: Data input/output
A_{18} to A_0: Address input
$\overline{CE}$: Chip enable
$\overline{OE}$: Output enable
$\overline{WE}$: Write enable

Notes: This figure shows pin assignments, and does not show the entire socket adapter circuit.
1. A reset oscillation stabilization time (t_{osc1}) of at least 10 ms is required.
2. A 12-MHz crystal resonator should be used.
3. As the FWE pin becomes VCC in the H8S/2339 F-ZTAT, the EMLE pin is ignored in PROM mode.

Figure 19.20 H8S/2339 F-ZTAT Socket Adapter Pin Assignments

Table 19.14 shows how the different operating modes are set when using PROM mode, and table 19.15 lists the commands used in PROM mode. Details of each mode are given below.

Memory Read Mode: Memory read mode supports byte reads.

Auto-Program Mode: Auto-program mode supports programming of 128 bytes at a time. Status polling is used to confirm the end of auto-programming.

Auto-Erase Mode: Auto-erase mode supports automatic erasing of the entire flash memory. Status polling is used to confirm the end of auto-erasing.

Status Read Mode: Status polling is used for auto-programming and auto-erasing, and normal termination can be confirmed by reading the I/O_6 signal. In status read mode, error information is output if an error occurs.

Table 19.14 Settings for Each Operating Mode in PROM Mode

	Pin Names				
Mode	$\overline{CE}$	$\overline{OE}$	$\overline{WE}$	**I/O_7 to I/O_0**	**A_{18} to A_0**
Read	L	L	H	Data output	Ain
Output disable	L	H	H	Hi-Z	X
Command write	L	H	L	Data input	Ain*2
Chip disable*1	H	X	X	Hi-Z	X

Legend:

H: High level

L: Low level

Hi-Z: High impedance

X: Don't care

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.

Command Name	Number of Cycles	1st Cycle			2nd Cycle		
		Mode	Address	Data	Mode	Address	Data
Memory read mode	1 + n	Write	X	H'00	Read	RA	Dout
Auto-program mode	129	Write	X	H'40	Write	PA	Din
Auto-erase mode	2	Write	X	H'20	Write	X	H'20
Status read mode	2	Write	X	H'71	Write	X	H'71

Legend:

RA: Read address

PA: Program address

Notes: 1. In auto-program mode, 129 cycles are required for command writing by a simultaneous 128-byte write.

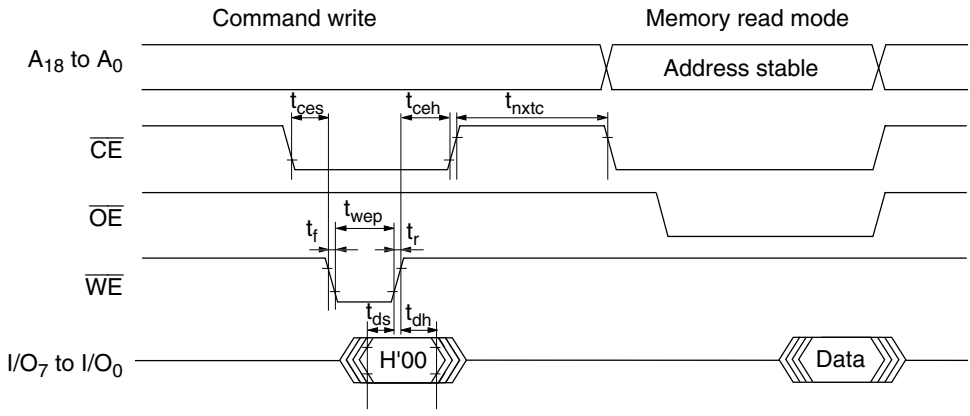
2. In memory read mode, the number of cycles depends on the number of address write cycles (n).

19.11.4 Memory Read Mode

- After the end of an auto-program, auto-erase, or status read operation, the command wait state is entered. To read memory contents, a transition must be made to memory read mode by means of a command write before the read is executed.
- Command writes can be performed in memory read mode, just as in the command wait state.
- Once memory read mode has been entered, consecutive reads can be performed.
- After power-on, memory read mode is entered.

Conditions: V_{CC} = 3.3 V ±0.3 V, V_{SS} = 0 V, T_a = 25°C ±3°C

Item	Symbol	Min	Max	Unit
Command write cycle	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
$\overline{WE}$ rise time	t_r	—	30	ns
$\overline{WE}$ fall time	t_f	—	30	ns



Note: Data is latched at the rising edge of $\overline{WE}$.

Figure 19.21 Memory Read Mode Timing Waveforms after Command Write

Conditions: V_{CC} = 3.3 V ±0.3 V, V_{SS} = 0 V, T_a = 25°C ±3°C

Item	Symbol	Min	Max	Unit
Command write cycle	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
$\overline{WE}$ rise time	t_r	—	30	ns
$\overline{WE}$ fall time	t_f	—	30	ns

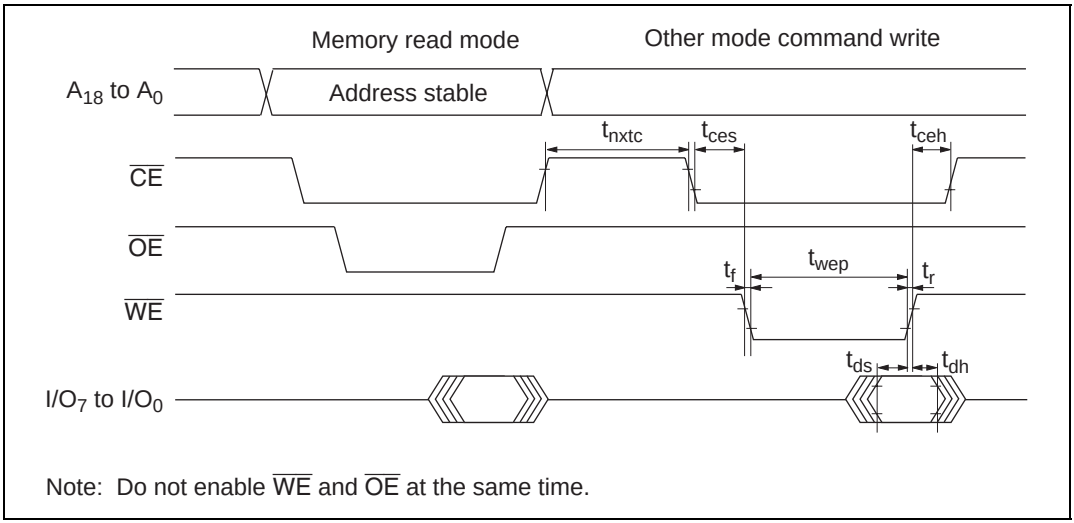


Figure 19.22 Timing Waveforms when Entering Another Mode from Memory Read Mode

Conditions: V_{CC} = 3.3 V ±0.3 V, V_{SS} = 0 V, T_a = 25°C ±3°C

Item	Symbol	Min	Max	Unit
Access time	t_{acc}	—	20	µs
$\overline{CE}$ output delay time	t_{ce}	—	150	ns
$\overline{OE}$ output delay time	t_{oe}	—	150	ns
Output disable delay time	t_{df}	—	100	ns
Data output hold time	t_{oh}	5	—	ns

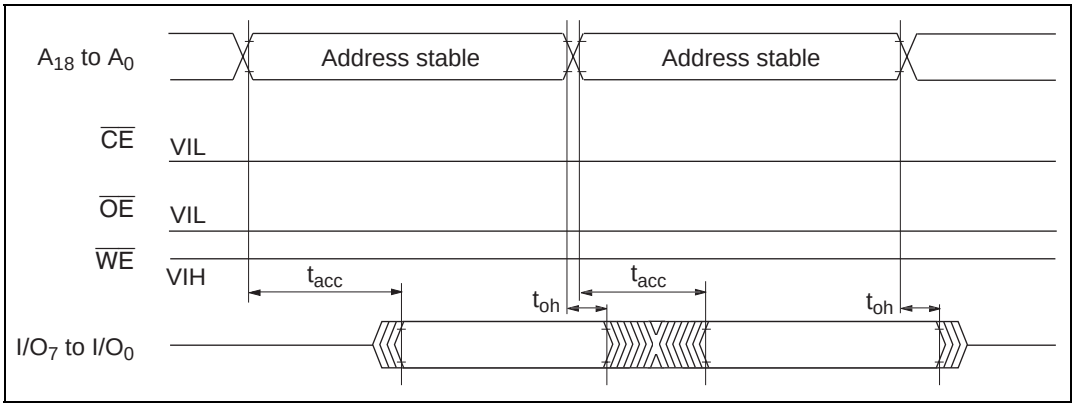


Figure 19.23 Timing Waveforms for $\overline{CE}$/$\overline{OE}$ Enable State Read

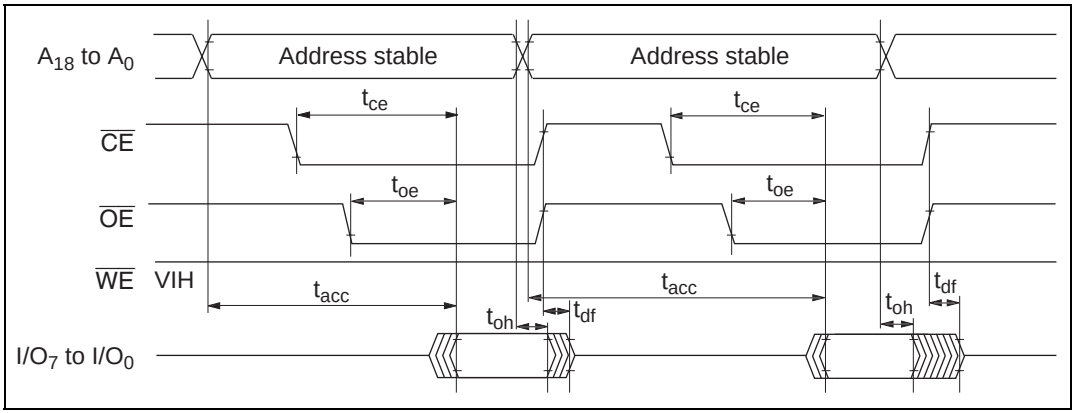


Figure 19.24 Timing Waveforms for $\overline{CE}$/$\overline{OE}$ Clocked Read

- In auto-program mode, 128 bytes are programmed simultaneously. For this purpose, 128 consecutive byte data transfers should be performed.
- 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.
- The lower 7 bits of the transfer address must be held low. If an invalid address is input, memory programming will be started but a programming error will occur.
- Memory address transfer is executed in the second cycle (figure 19.25). Do not perform transfer later than the second cycle.
- Do not perform a command write during a programming operation.
- Perform one auto-programming operation for a 128-byte block for each address. One or more additional programming operations cannot be carried out on address blocks that have already been programmed.
- Confirm normal end of auto-programming by checking I/O_6. Alternatively, status read mode can also be used for this purpose (the I/O_7 status polling pin is used to identify the end of an auto-program operation).
- Status polling I/O_6 and I/O_7 information is retained until the next command write. As long as the next command write has not been performed, reading is possible by enabling $\overline{CE}$ and $\overline{OE}$.

Table 19.19 AC Characteristics in Auto-Program Mode

Conditions: V_{CC} = 3.3 V ±0.3 V, V_{SS} = 0 V, T_a = 25°C ±5°C

Item	Symbol	Min	Max	Unit
Command write cycle	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
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
$\overline{WE}$ rise time	t_r	—	30	ns
$\overline{WE}$ fall time	t_f	—	30	ns

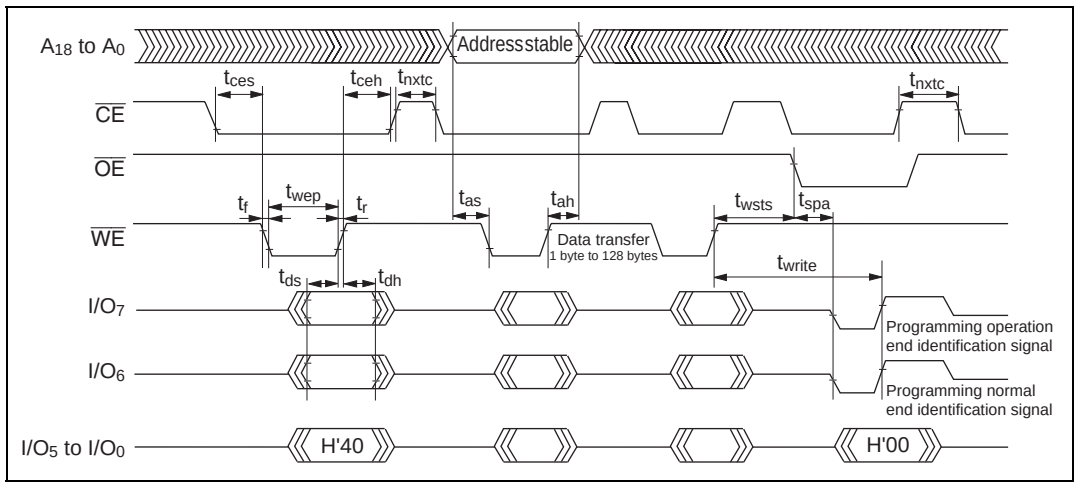


Figure 19.25 Auto-Program Mode Timing Waveforms

- Auto-erase mode supports only total memory erasing.
- Do not perform a command write during auto-erasing.
- Confirm normal end of auto-erasing by checking I/O_6. Alternatively, status read mode can also be used for this purpose (the I/O_7 status polling pin is used to identify the end of an auto-erase operation).
- Status polling I/O_6 and I/O_7 pin information is retained until the next command write. As long as the next command write has not been performed, reading is possible by enabling $\overline{CE}$ and $\overline{OE}$.

AC Characteristics

Table 19.20 AC Characteristics in Auto-Erase Mode

Conditions: V_{CC} = 3.3 V ±0.3 V, V_{SS} = 0 V, T_a = 25°C ±5°C

Item	Symbol	Min	Max	Unit
Command write cycle	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
Status polling start time	t_{ests}	1	—	ms
Status polling access time	t_{spa}	—	150	ns
Memory erase time	t_{erase}	100	40000	ms
$\overline{WE}$ rise time	t_r	—	30	ns
$\overline{WE}$ fall time	t_f	—	30	ns

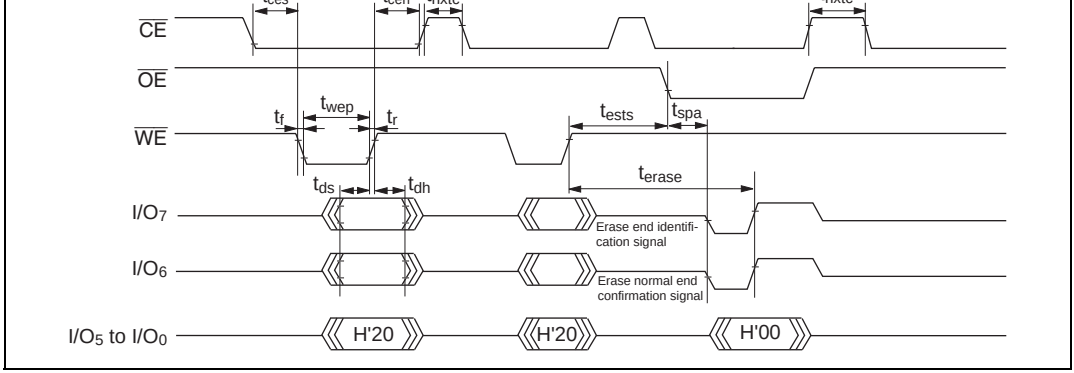


Figure 19.26 Auto-Erase Mode Timing Waveforms

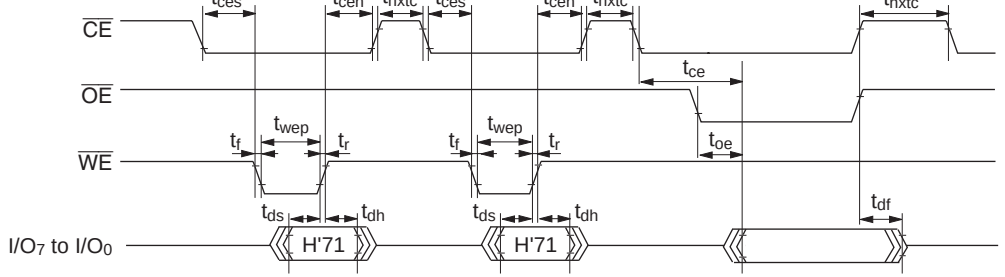
19.11.7 Status Read Mode

- Status read mode is used to identify what type of abnormal end has occurred. Use this mode when an abnormal end occurs in auto-program mode or auto-erase mode.
- The return code is retained until a command write for other than status read mode is performed.

Table 19.21 AC Characteristics in Status Read Mode

Conditions: V_{CC} = 3.3 V ±0.3 V, V_{SS} = 0 V, T_a = 25°C ±5°C

Item	Symbol	Min	Max	Unit
Command write cycle	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/O_3 and I/O_2 are undefined.

Figure 19.27 Status Read Mode Timing Waveforms

Table 19.22 Status Read Mode Return Commands

Pin Name	I/O_7	I/O_6	I/O_5	I/O_4	I/O_3	I/O_2	I/O_1	I/O_0
Attribute	Normal end identification	Command error	Program-ming error	Erase error	—	—	Program-ming or erase count exceeded	Effective address error
Initial value	0	0	0	0	0	0	0	0
Indications	Normal end: 0 Abnormal end: 1	Command error: 1 Otherwise: 0	Program-ming error: 1 Otherwise: 0	Erase error: 1 Otherwise: 0	—	—	Count exceeded: 1 Otherwise: 0	Effective address error: 1 Otherwise: 0

Note: I/O_3 and I/O_2 are undefined.

- The I/O_7 status polling flag indicates the operating status in auto-program or auto-erase mode.
- The I/O_6 status polling flag indicates a normal or abnormal end in auto-program or auto-erase mode.

Table 19.23 Status Polling Output Truth Table

Pin Names	Internal Operation in Progress	Abnormal End	—	Normal End
I/O_7	0	1	0	1
I/O_6	0	0	1	1
I/O_0 to I/O_5	0	0	0	0

19.11.9 PROM Mode Transition Time

Commands cannot be accepted during the oscillation stabilization period or the PROM mode setup period. After the PROM mode setup time, a transition is made to memory read mode.

Table 19.24 Command Wait State Transition Time Specifications

Item	Symbol	Min	Max	Unit
Standby release (oscillation stabilization time)	t_{osc1}	30	—	ms
PROM mode setup time	t_{bmv}	10	—	ms
V_{CC} hold time	t_{dwn}	0	—	ms

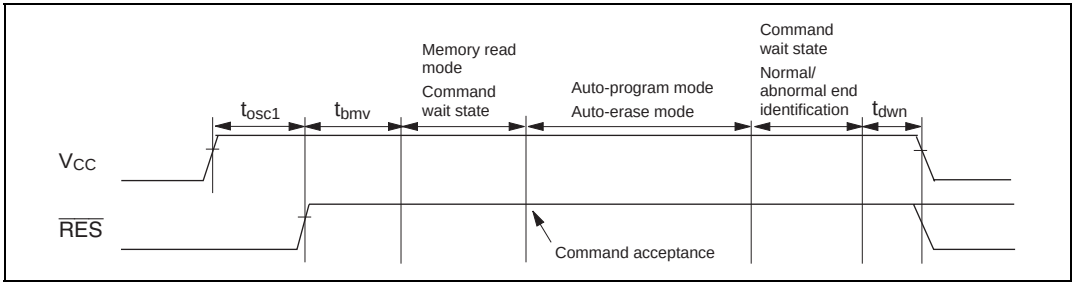


Figure 19.28 Oscillation Stabilization Time, PROM Mode Setup Time, and Power Supply Fall Sequence

- When programming addresses which have previously been programmed, carry out auto-erasing before auto-programming.
- When performing programming using PROM mode on a chip that has been programmed/erased in an on-board programming mode, auto-erasing is recommended before carrying out auto-programming.

Notes: 1. The flash memory is initially in the erased state when the device is shipped by Renesas Technology. For other chips for which the erasure history is unknown, it is recommended that auto-erasing be executed to check and supplement the initialization (erase) level.

2. Auto-programming should be performed once only on the same address block. Additional programming cannot be carried out on address blocks that have already been programmed.

19.12 Flash Memory Programming and Erasing Precautions

Precautions concerning the use of on-board programming mode, the RAM emulation function, and PROM mode are summarized below.

Use the specified voltages and timing for programming and erasing: Applied voltages in excess of the rating can permanently damage the device. Use a PROM programmer that supports the Renesas microcomputer device type with 512-kbyte on-chip flash memory (FZTAT512V3A).

Do not select the HN27C4096 setting for the PROM programmer, and only use the specified socket adapter. Failure to observe these points may result in damage to the device.

Powering on and off: When applying or disconnecting V_{CC} power, fix the $\overline{RES}$ pin low and place the flash memory in the hardware protection state.

The power-on and power-off timing requirements should also be satisfied in the event of a power failure and subsequent recovery.

Use the recommended algorithm when programming and erasing flash memory: The recommended algorithm enables programming and erasing to be carried out without subjecting the device to voltage stress or sacrificing program data reliability. When setting the P or E bit in FLMCR1, the watchdog timer should be set beforehand as a precaution against program runaway, etc.

memory. When the SWE bit is set, data in flash memory can be rewritten, but when SWE = 1, flash memory can only be read in program-verify or erase-verify mode. Access flash memory only for verify operations (verification during programming/erasing). Also, do not clear the SWE bit during programming, erasing, or verifying.

Similarly, when using the RAM emulation function the SWE bit must be cleared before executing a program or reading data in flash memory.

However, the RAM area overlapping flash memory space can be read and written to regardless of whether the SWE bit is set or cleared.

Do not use interrupts while flash memory is being programmed or erased: When flash memory is programmed or erased, all interrupt requests, including NMI, should be disabled to give priority to program/erase operations.

Do not perform additional programming. Erase the memory before reprogramming: In on-board programming, perform only one programming operation on a 128-byte programming unit block. In PROM mode, too, perform only one programming operation on a 128-byte programming unit block. Programming should be carried out with the entire programming unit block erased.

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 programmer socket, socket adapter, and chip are not correctly aligned.

Do not touch the socket adapter or chip during programming: Touching either of these can cause contact faults and write errors.

19.13.1 Features

The H8S/2338 F-ZTAT has 256 kbytes of on-chip flash memory. The features of the flash memory are summarized below.

- 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. Erasing is performed by block erase (in single-block units). To erase the entire flash memory, the individual blocks must be erased sequentially. Block erasing can be performed as required on 4-kbyte, 32-kbyte, and 64-kbyte blocks.
- Programming/erase times

  The flash memory programming time is 10.0 ms (typ.) for simultaneous 128-byte programming, equivalent to 78 µs (typ.) per byte, and the erase time is 50 ms (typ.).
- Reprogramming capability

  The flash memory can be reprogrammed min. 100 times.
- On-board programming modes

  There are two modes in which flash memory can be programmed/erased/verified on-board:
  - Boot mode
  - User program mode
- Automatic bit rate adjustment

  With data transfer in boot mode, the bit rate of the chip can be automatically adjusted to match the transfer bit rate of the host.
- Flash memory emulation by RAM

  Part of the RAM area can be overlapped onto flash memory, to emulate flash memory updates in real time.
- Protect modes

  There are three protect modes, hardware, software, and error protect, which allow protected status to be designated for flash memory program/erase/verify operations.

well as in on-board programming mode.

19.13.2 Overview

Block Diagram

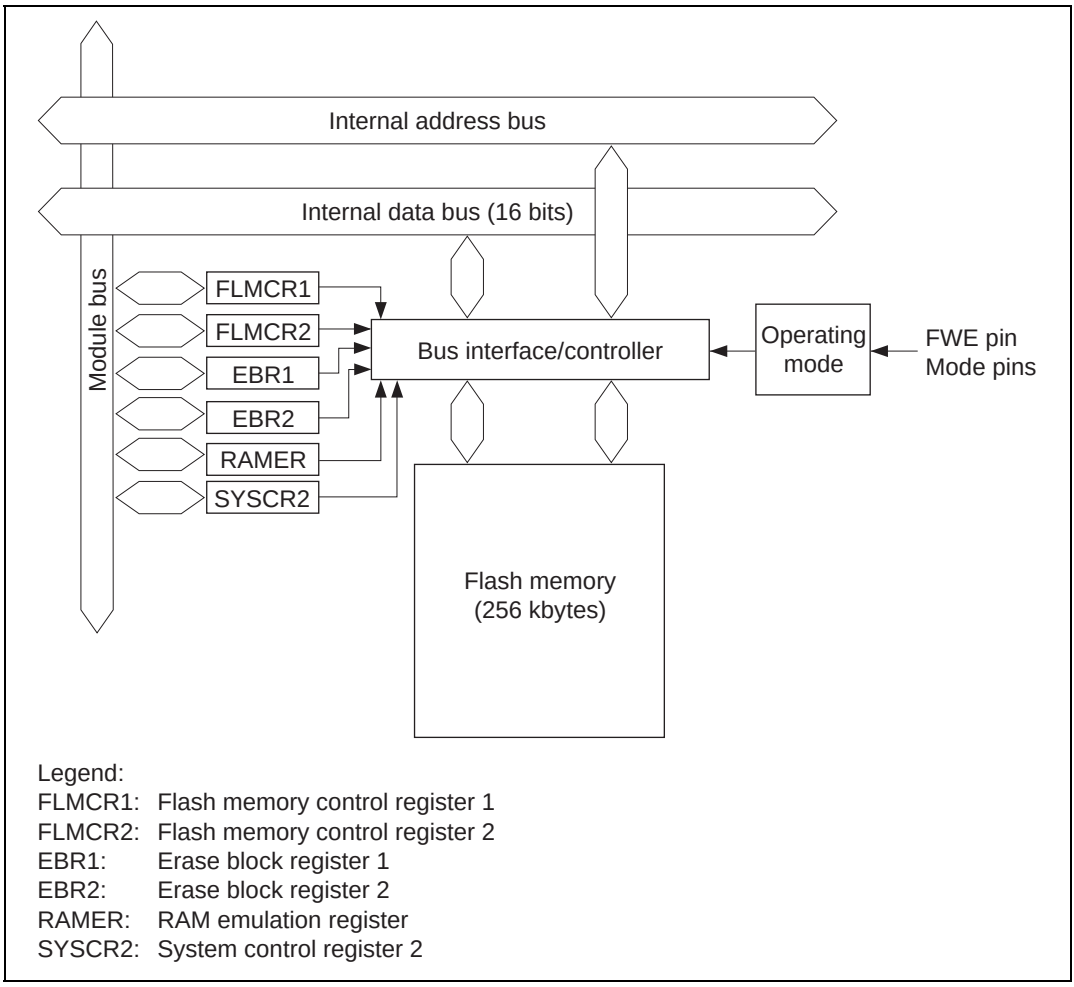
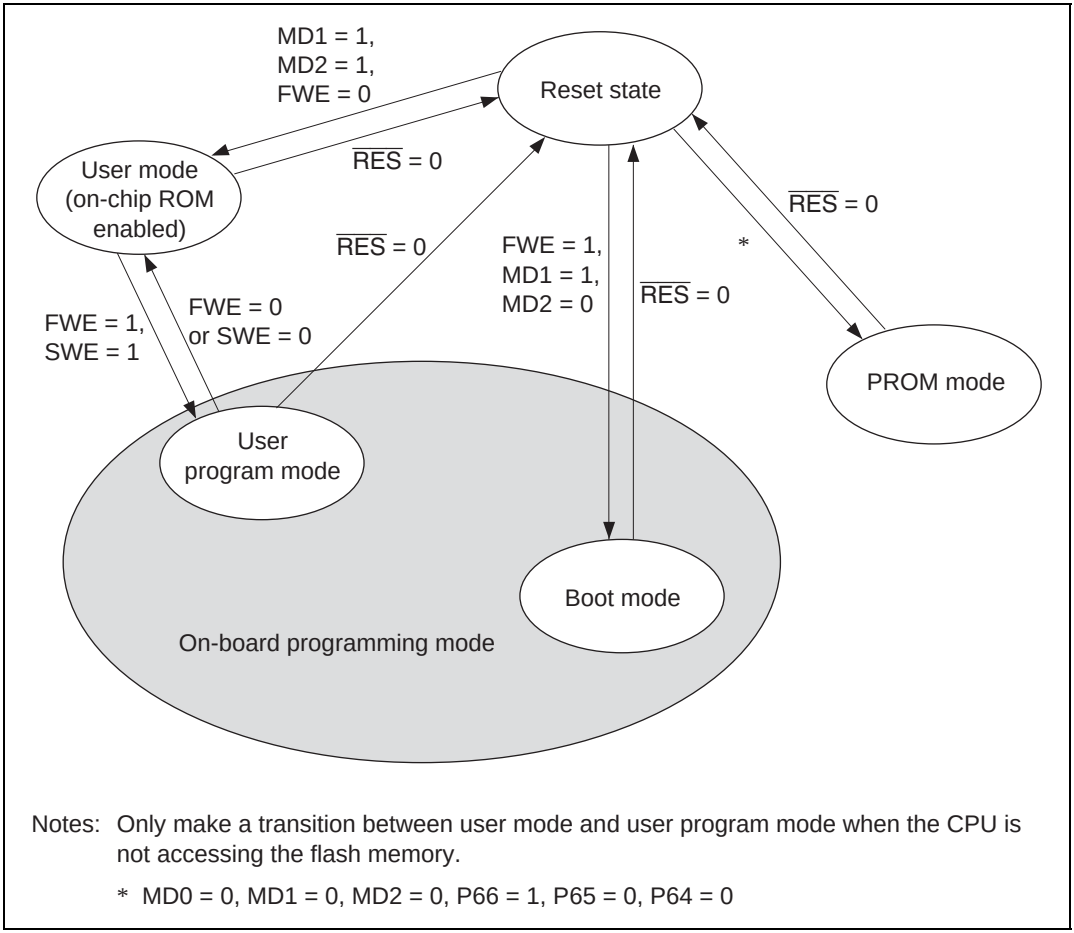


Figure 19.29 Block Diagram of Flash Memory

Mode Transitions: When the mode pins and the FWE pin are set in the reset state and a reset-start is executed, the chip enters one of the operating modes shown in figure 19.30. In user mode, flash memory can be read but not programmed or erased.

Flash memory can be programmed and erased in boot mode, user program mode, and PROM mode.



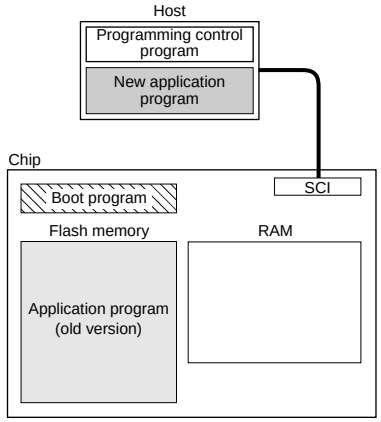
Notes: Only make a transition between user mode and user program mode when the CPU is not accessing the flash memory.

* MD0 = 0, MD1 = 0, MD2 = 0, P66 = 1, P65 = 0, P64 = 0

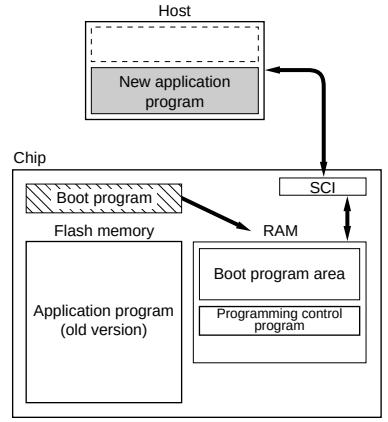
Figure 19.30 Flash Memory Mode Transitions

- Boot mode

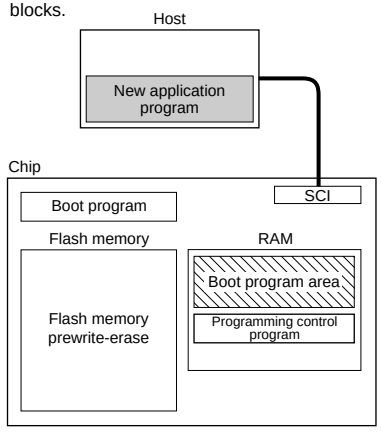
1. Initial state
   The old program version or data remains written in the flash memory. The user should prepare the programming control program and new application program beforehand in the host.



2. Programming control program transfer
   When boot mode is entered, the boot program in the chip (originally incorporated in the chip) is started and the programming control program in the host is transferred to RAM via SCI communication. The boot program required for flash memory erasing is automatically transferred to the RAM boot 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, entire flash memory erasure is performed, without regard to blocks.



4. Writing new application program
   The programming control program transferred from the host to RAM is executed, and the new application program in the host is written into the flash memory.

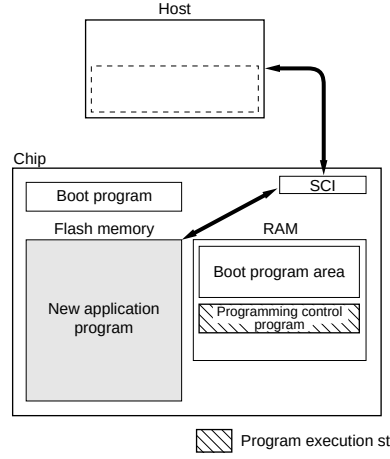
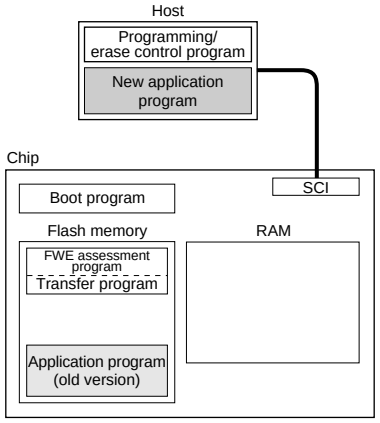
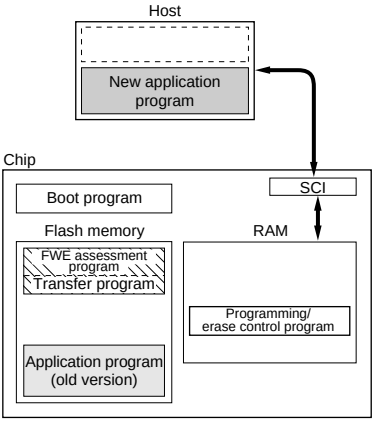


Figure 19.31 Boot Mode

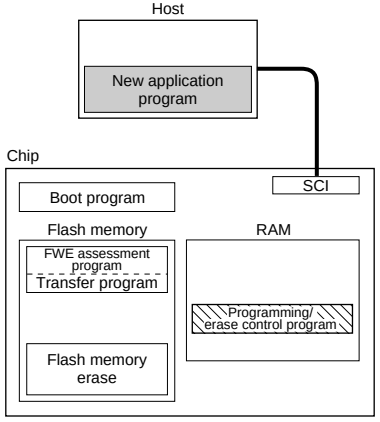
1. Initial state
   (1) The FWE assessment program that confirms that the FWE pin has been driven high, and (2) the program that will transfer the programming/erase control program to on-chip RAM should be written into the flash memory by the user beforehand. (3) The programming/erase control program should be prepared in the host or in the flash memory.



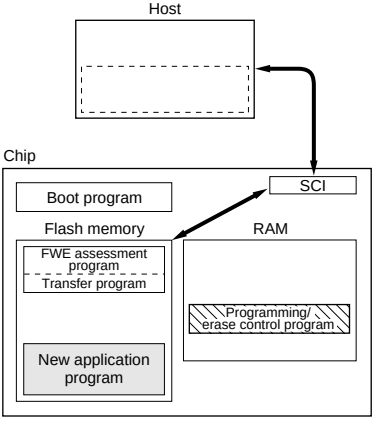
2. Programming/erase control program transfer
   When the FWE pin is driven high, user software confirms this fact, executes the transfer program in the flash memory, and transfers the programming/erase control program to RAM.



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 host is written into the erased flash memory blocks. Do not write to unerased blocks.



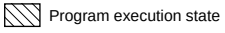


Figure 19.32 User Program Mode (Example)

Reading Overlap RAM Data in User Mode and User Program Mode: Emulation should be performed in user mode or user program mode. When the emulation block set in RAMER is accessed while the emulation function is being executed, data written in the overlap RAM is read.

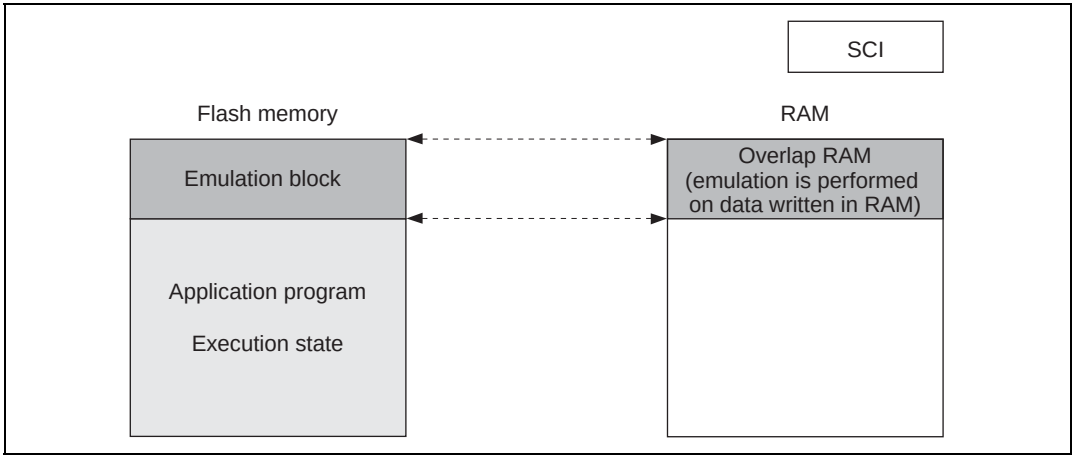


Figure 19.33 Reading Overlap RAM Data in User Mode and User Program Mode

flash memory.

When the programming control program is transferred to RAM, ensure that the transfer destination and the overlap RAM do not overlap, as this will cause data in the overlap RAM to be rewritten.

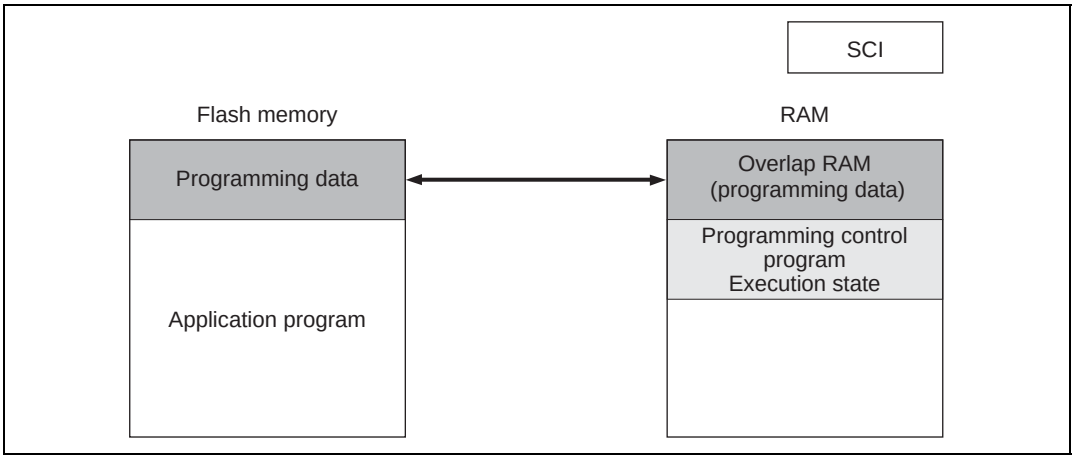


Figure 19.34 Writing Overlap RAM Data in User Program Mode

19.13.6 Differences between Boot Mode and User Program Mode

Table 19.25 Differnces between Boot Mode and User Program Mode

	Boot Mode	User Program Mode
Entire memory erase	Yes	Yes
Block erase	No	Yes
Programming control program*	Program/program-verify	Erase/erase-verify/program/ program-verify/emulation

Note: * To be provided by the user, in accordance with the recommended algorithm.

On-chip 256-kbyte flash memory is divided into three 64-kbyte blocks, one 32-kbyte block, and eight 4-kbyte blocks.

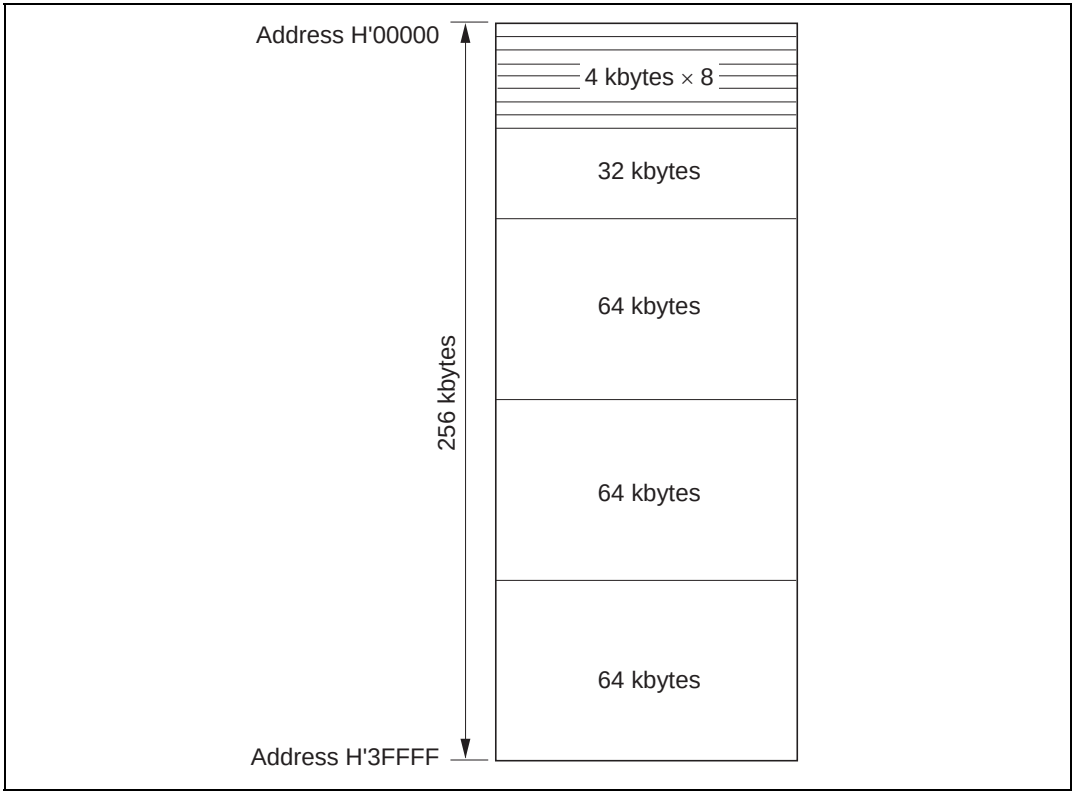


Figure 19.35 Flash Memory Block Configuration

The flash memory is controlled by means of the pins shown in tables 19.26.

Table 19.26 Flash Memory Pins

Pin Name	Abbreviation	I/O	Function
Reset	$\overline{\text{RES}}$	Input	Reset
Flash write enable	FWE	Input	Flash program/erase protection by hardware
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 64	P64	Input	Sets MCU operating mode in PROM mode
Port 65	P65	Input	Sets MCU operating mode in PROM mode
Port 66	P66	Input	Sets MCU operating mode in PROM mode
Transmit data	TxD1	Output	Serial transmit data output
Receive data	RxD1	Input	Serial receive data input

The registers used to control the on-chip flash memory when enabled are shown in table 19.27. In order to access the FLMCR1, FLMCR2, EBR1, and EBR2 registers, the FLSHE bit must be set to 1 in SYSCR2 (except RAMER).

Table 19.27 Flash Memory Registers

Register Name	Abbreviation	R/W	Initial Value	Address*1
Flash memory control register 1	FLMCR1*6	R/W*3	H'00/H'80*4	H'FFC8*2
Flash memory control register 2	FLMCR2*6	R/W*3	H'00	H'FFC9*2
Erase block register 1	EBR1*6	R/W*3	H'00*5	H'FFCA*2
Erase block register 2	EBR2*6	R/W*3	H'00*5	H'FFCB*2
System control register 2	SYSCR2*7	R/W	H'00	H'FF42
RAM emulation register	RAMER	R/W	H'00	H'FEDB

Notes: 1. Lower 16 bits of the address.
2. Flash memory. Registers selection is performed by the FLSHE bit in system control register 2 (SYSCR2).
3. In modes in which the on-chip flash memory is disabled, a read will return H'00, and writes are invalid. Writes are also disabled when the FWE bit is cleared to 0 in FLMCR1.
4. When a high level is input to the FWE pin, the initial value is H'80.
5. When a low level is input to the FWE pin, or if a high level is input and the SWE bit in FLMCR1 is not set, these registers are initialized to H'00.
6. FLMCR1, FLMCR2, EBR1, and EBR2 are 8-bit registers. Only byte accesses are valid for these registers, the access requiring 2 states.
7. The SYSCR2 register can only be used in the F-ZTAT version. In the mask ROM version this register will return an undefined value if read, and cannot be modified.

19.14.1 Flash Memory Control Register 1 (FLMCR1)

Bit	:	7	6	5	4	3	2	1	0
		FWE	SWE	ESU	PSU	EV	PV	E	P
Initial value	:	1/0	0	0	0	0	0	0	0
R/W	:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

FLMCR1 is an 8-bit register used for flash memory operating mode control. Program-verify mode or erase-verify mode is entered by setting SWE to 1 when FWE = 1, then setting the EV or PV bit. Program mode is entered by setting SWE to 1 when FWE = 1, then setting the PSU bit, and finally setting the P bit. Erase mode is entered by setting SWE to 1 when FWE = 1, then setting the ESU bit, and finally setting the E bit. FLMCR1 is initialized by a reset, and in hardware standby mode and software standby mode. Its initial value is H'80 when a high level is input to the FWE pin, and H'00 when a low level is input. When on-chip flash memory is disabled, a read will return H'00, and writes are invalid.

Writes to the SWE bit in FLMCR1 are enabled only when FWE = 1; writes to bits ESU, PSU, EV, and PV only when FWE = 1 and SWE = 1; writes to the E bit only when FWE = 1, SWE = 1, and ESU = 1; and writes to the P bit only when FWE = 1, SWE = 1, and PSU = 1.

Bit 7—Flash Write Enable Bit (FWE): Sets hardware protection against flash memory 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 (SWE): Enables or disables flash memory programming and erasing. This bit should be set when setting FLMCR1 bits 5 to 0, EBR1 bits 7 to 0, and EBR2 bits 3 to 0.

When SWE = 1, the flash memory can only be read in program-verify or erase-verify mode.

0	Writes disabled	(Initial value)
1	Writes enabled [Setting condition] When FWE = 1	

Bit 5—Erase Setup Bit (ESU): Prepares for a transition to erase mode. Do not set the SWE, PSU, EV, PV, E, or P bit at the same time.

Bit 5 ESU	Description	
0	Erase setup cleared	(Initial value)
1	Erase setup [Setting condition] When FWE = 1 and SWE = 1	

Bit 4—Program Setup Bit (PSU): Prepares for a transition to program mode. Do not set the SWE, ESU, EV, PV, E, or P bit at the same time.

Bit 4 PSU	Description	
0	Program setup cleared	(Initial value)
1	Program setup [Setting condition] When FWE = 1 and SWE = 1	

Bit 3—Erase-Verify (EV): Selects erase-verify mode transition or clearing. Do not set the SWE, ESU, PSU, PV, E, or P bit at the same time.

Bit 3 EV	Description	
0	Erase-verify mode cleared	(Initial value)
1	Transition to erase-verify mode [Setting condition] When FWE = 1 and SWE = 1	

Bit 2 PV	Description	
0	Program-verify mode cleared	(Initial value)
1	Transition to program-verify mode [Setting condition] When FWE = 1 and SWE = 1	

Bit 1—Erase (E): Selects erase mode transition or clearing. Do not set the SWE, ESU, PSU, EV, PV, or P bit at the same time.

Bit 1 E	Description	
0	Erase mode cleared	(Initial value)
1	Transition to erase mode [Setting condition] When FWE = 1, SWE = 1, and ESU = 1	

Bit 0—Program (P): Selects program mode transition or clearing. Do not set the SWE, PSU, ESU, EV, PV, or E bit at the same time.

Bit 0 P	Description	
0	Program mode cleared	(Initial value)
1	Transition to program mode [Setting condition] When FWE = 1, SWE = 1, and PSU = 1	

Bit	:	7	6	5	4	3	2	1	0
		FLER	—	—	—	—	—	—	—
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R	—	—	—	—	—	—	—

FLMCR2 is an 8-bit register that controls the flash memory operating modes. FLMCR2 is initialized to H'00 by a reset, and in hardware standby mode and software standby mode.

When on-chip flash memory is disabled, a read will return H'00 and writes are invalid.

Bit 7—Flash Memory Error (FLER): Indicates that an error has occurred during an operation on flash memory (programming or erasing). When FLER is set to 1, flash memory goes to the error-protection state.

Bit 7 FLER	Description	
0	Flash memory is operating normally	(Initial value)
	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 19.17.3, Error Protection	

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

Bit	:	7	6	5	4	3	2	1	0
EBR1		EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0
Initial value	:	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

EBR1 is an 8-bit register that specifies the flash memory erase area block by block. EBR1 is initialized to H'00 by a reset, in hardware standby mode and software standby mode, when a low level is input to the FWE pin, and when a high level is input to the FWE pin and the SWE bit in FLMCR1 is not set. When a bit in EBR1 is set, the corresponding block can be erased. Other blocks are erase-protected. Set only one bit in EBR1 and EBR2 together (setting more than one bit will automatically clear all EBR1 and EBR2 bits to 0). 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 19.28.

19.14.4 Erase Block Registers 2 (EBR2)

Bit	:	7	6	5	4	3	2	1	0
EBR2		—	—	—	—	EB11	EB10	EB9	EB8
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	—	—	—	—	R/W	R/W	R/W	R/W

EBR2 is an 8-bit register that specifies the flash memory erase area block by block. EBR2 is initialized to H'00 by a reset, in hardware standby mode and software standby mode, when a low level is input to the FWE pin, and when a high level is input to the FWE pin and the SWE bit in FLMCR1 is not set. When a bit in EBR2 is set, the corresponding block can be erased. Other blocks are erase-protected. Set only one bit in EBR2 and EBR1 together (setting more than one bit will automatically clear all EBR1 and EBR2 bits to 0). Bits 7 to 4 are reserved; they are always read as 0 and cannot be modified. 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 19.28.

Block (Size)	Address
EB0 (4 kbytes)	H'000000 to H'000FFF
EB1 (4 kbytes)	H'001000 to H'001FFF
EB2 (4 kbytes)	H'002000 to H'002FFF
EB3 (4 kbytes)	H'003000 to H'003FFF
EB4 (4 kbytes)	H'004000 to H'004FFF
EB5 (4 kbytes)	H'005000 to H'005FFF
EB6 (4 kbytes)	H'006000 to H'006FFF
EB7 (4 kbytes)	H'007000 to H'007FFF
EB8 (32 kbytes)	H'008000 to H'00FFFF
EB9 (64 kbytes)	H'010000 to H'01FFFF
EB10 (64 kbytes)	H'020000 to H'02FFFF
EB11 (64 kbytes)	H'030000 to H'03FFFF

19.14.5 System Control Register 2 (SYSCR2)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	FLSHE	—	—	—
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	—	—	—	—	R/W	—	—	—

SYSCR2 is an 8-bit readable/writable register that performs on-chip flash memory control.

SYSCR2 is initialized to H'00 by a reset and in hardware standby mode.

SYSCR2 can only be used in the F-ZTAT versions. In the mask ROM versions this register will return an undefined value if read, and cannot be modified.

Bits 7 to 4—Reserved: These bits cannot be modified and are always read as 0.

enables the flash memory control registers to be read and written to. Clearing FLSHE to 0 designates these registers as unselected (the register contents are retained).

Bit 3 FLSHE	Description
0	Flash control registers are not selected for addresses H'FFFFC8 to H'FFFFCB (Initial value)
1	Flash control registers are selected for addresses H'FFFFC8 to H'FFFFCB

Bits 2 to 0—Reserved: These bits cannot be modified and are always read as 0.

19.14.6 RAM Emulation Register (RAMER)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	RAMS	RAM2	RAM1	RAM0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	—	—	—	—	R/W	R/W	R/W	R/W

RAMER specifies the area of flash memory to be overlapped with part of RAM when emulating real-time flash memory programming. RAMER is initialized to H'00 by a reset and in hardware standby mode. It is not initialized in software standby mode. RAMER settings should be made in user mode or user program mode.

Flash memory area divisions are shown in table 19.29. To ensure correct operation of the emulation function, the ROM for which RAM emulation is performed should not be accessed immediately after this register has been modified. Normal execution of an access immediately after register modification is not guaranteed.

Bits 7 to 4—Reserved: These bits cannot be modified and are always read as 0.

Bit 3 RAMS	Description	
0	Emulation not selected	(Initial value)
	Program/erase-protection of all flash memory blocks is disabled	
1	Emulation selected	
	Program/erase-protection of all flash memory blocks is enabled	

Bits 2 to 0—Flash Memory Area Selection (RAM2 to RAM0): These bits are used together with bit 3 to select the flash memory area to be overlapped with RAM. (See table 19.29.)

Table 19.29 Flash Memory Area Divisions

RAM Area	Block Name	RAMS	RAM2	RAM1	RAM0
H'FFDC00 to H'FFEBFF	RAM area, 4 kbytes	0	*	*	*
H'000000 to H'000FFF	EB0 (4 kbytes)	1	0	0	0
H'001000 to H'001FFF	EB1 (4 kbytes)	1	0	0	1
H'002000 to H'002FFF	EB2 (4 kbytes)	1	0	1	0
H'003000 to H'003FFF	EB3 (4 kbytes)	1	0	1	1
H'004000 to H'004FFF	EB4 (4 kbytes)	1	1	0	0
H'005000 to H'005FFF	EB5 (4 kbytes)	1	1	0	1
H'006000 to H'006FFF	EB6 (4 kbytes)	1	1	1	0
H'007000 to H'007FFF	EB7 (4 kbytes)	1	1	1	1

*: Don't care

When pins are set to on-board programming mode, program/erase/verify operations can be performed on the on-chip flash memory. There are two on-board programming modes: boot mode and user program mode. The pin settings for transition to each of these modes are shown in table 19.30. For a diagram of the transitions to the various flash memory modes, see figure 19.30.

Table 19.30 Setting On-Board Programming Modes

Mode		Pins			
MCU Mode	**CPU Operating Mode**	**FWE**	**MD2**	**MD1**	**MD0**
Boot mode	Advanced expanded mode with on-chip ROM enabled	1	0	1	0
	Advanced single-chip mode				1
User program mode*	Advanced expanded mode with on-chip ROM enabled	1	1	1	0
	Advanced single-chip mode				1

Note: * Normally, user mode should be used. Set the FWE pin to 1 to make a transition to user program mode before performing a program/erase/verify operation.

19.15.1 Boot Mode

When boot mode is used, the flash memory programming control program must be prepared in the host beforehand. The channel 1 SCI to be used is set to asynchronous mode.

When a reset-start is executed after the H8S/2338 F-ZTAT chip's pins have been set to boot mode, the boot program built into the chip is started and the programming control program prepared in the host is serially transmitted to the chip via the SCI. In the chip, the programming control program received via the SCI is written into the programming control program area in on-chip RAM. After the transfer is completed, control branches to the start address of the programming control program area and the programming control program execution state is entered (flash memory programming is performed).

The transferred programming control program must therefore include coding that follows the programming algorithm given later.

The system configuration in boot mode is shown in figure 19.36, and the boot program mode execution procedure in figure 19.37.

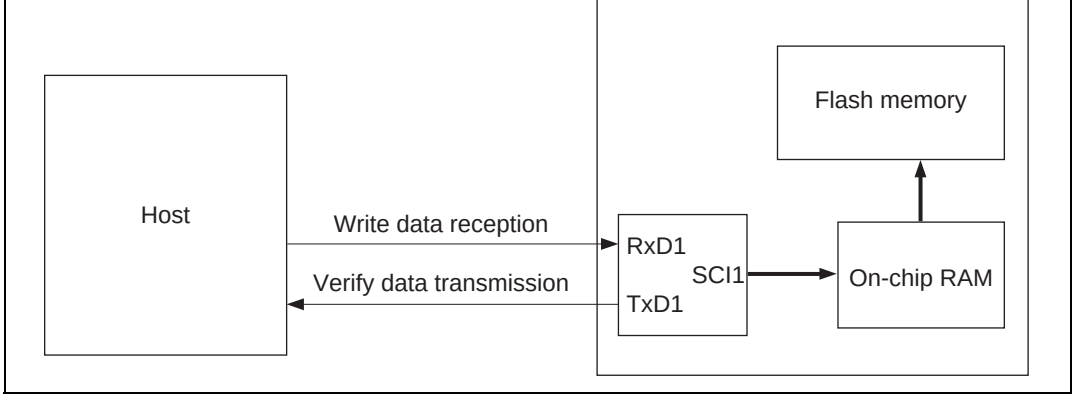
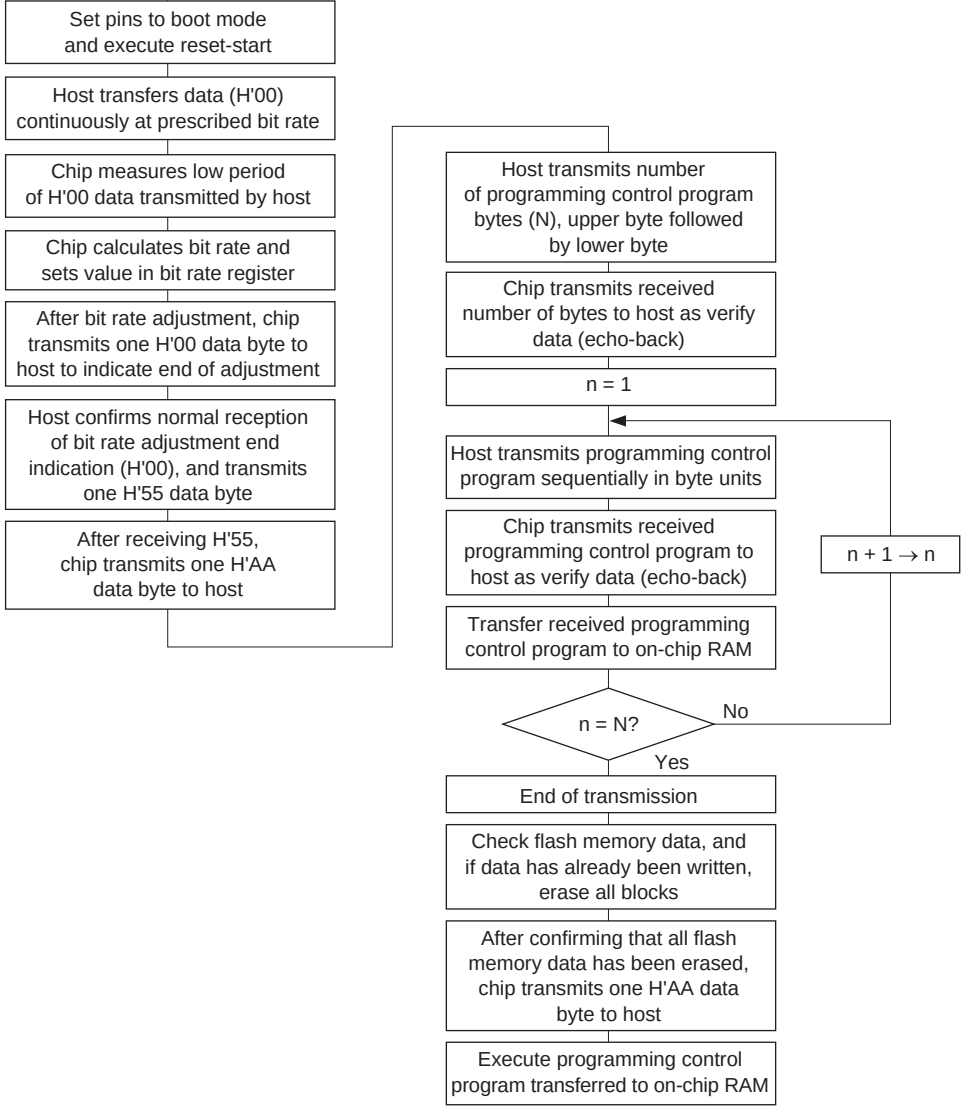


Figure 19.36 System Configuration in Boot Mode



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 operations are halted.

Figure 19.37 Boot Mode Execution Procedure

continuously from the host. The SCI transmit/receive format should be set as follows: 8-bit data, 1 stop bit, no parity. The chip calculates the bit rate of the transmission from the host from the measured low period, and transmits one H'00 byte to the host to indicate the end of bit rate adjustment. The host should confirm that this adjustment end indication (H'00) has been received normally, and transmit one H'55 byte to the chip. If reception cannot be performed normally, initiate boot mode again (reset), and repeat the above operations. Depending on the host's transmission bit rate and the chip's system clock frequency, there will be a discrepancy between the bit rates of the host and the chip. To ensure correct SCI operation, the host's transfer bit rate should be set to 9,600 or 19,200 bps.

Table 19.31 shows typical host transfer bit rates and system clock frequencies for which automatic adjustment of the MCU's bit rate is possible. The boot program should be executed within this system clock range.

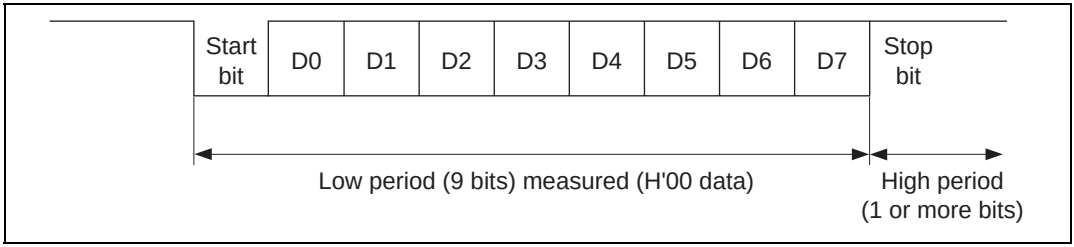
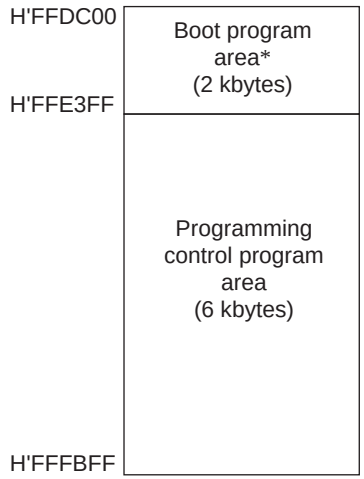


Figure 19.38 Automatic SCI Bit Rate Adjustment

Table 19.31 System Clock Frequencies for which Automatic Adjustment of H8S/2338 F-ZTAT Bit Rate is Possible

Host Bit Rate	System Clock Frequencies for which Automatic Adjustment of H8S/2338 F-ZTAT Bit Rate is Possible
19,200 bps	16 MHz to 25 MHz
9,600 bps	8 MHz to 25 MHz

the programming control program is transferred is H'FFE400 to H'FFFBFF. The boot program area can be used when the programming control program transferred into RAM enters the execution state. A stack area should be set up as required.



Note: * The boot program area cannot be used until a transition is made to the execution state for the programming control program transferred to RAM. Note that the boot program remains stored in this area after a branch is made to the programming control program.

Figure 19.39 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 input at the SCI's RxD1 pin. The reset should end with RxD1 high. After the reset ends, it takes approximately 100 states before the chip is ready to measure the low-level period of the RxD1 pin.
- In boot mode, if any data has been programmed into the flash memory (if all data is not 1), all flash memory blocks are erased. Boot mode is for use when user program mode is unavailable, such as the first time on-board programming is performed, or if the program activated in user program mode is accidentally erased.
- Interrupts cannot be used while the flash memory is being programmed or erased.
- The RxD1 and TxD1 pins should be pulled up on the board.

the RE and TE bits in SCR to 0), but the adjusted bit rate value remains set in BRR. The transmit data output pin, TxD1, goes to the high-level output state (P31DDR = 1, P31DR = 1).

The contents of the CPU's internal general registers are undefined at this time, so these registers must be initialized immediately after branching to the programming control program. In particular, since the stack pointer (SP) is used implicitly in subroutine calls, etc., a stack area must be specified for use by the programming control program.

Initial settings must also be made for the other on-chip registers.

- Boot mode can be entered by making the pin settings shown in table 19.30 and executing a reset-start.

  Boot mode can be cleared by driving the reset pin low, waiting at least 20 states, then setting the FWE pin and mode pins, and executing reset release*1. Boot mode can also be cleared by a WDT overflow reset.

  Do not change the mode pin input levels in boot mode, and do not drive the FWE pin low while the boot program is being executed or while flash memory is being programmed or erased*2.

- If the mode pin input levels are changed (for example, from low to high) during a reset, the state of ports with multiplexed address functions and bus control output pins ($\overline{AS}$, $\overline{RD}$, $\overline{HWR}$) 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 becoming output signal pins during a reset, or to prevent collision with signals outside the microcomputer.

Notes: 1. Mode pins and FWE pin input must satisfy the mode programming setup time (t_{MDS} = 200 ns) with respect to the reset release timing, as shown in figures 19.56 to 19.58.

2. For further information on FWE application and disconnection, see section 19.21, Flash Memory Programming and Erasing Precautions.

3. See section 9, I/O Ports.

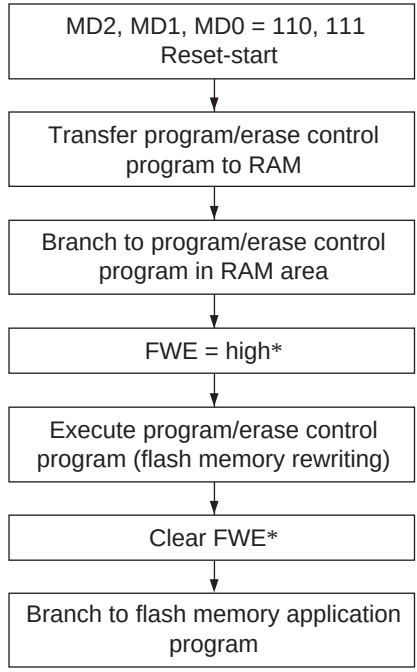
When set to user program mode, the chip can program and erase its flash memory by executing a user program/erase control program. Therefore, on-board reprogramming of the on-chip flash memory can be carried out by providing on-board means of FWE control and supply of programming data, and storing a program/erase control program in part of the program area as necessary.

To select user program mode, select a mode that enables the on-chip flash memory (mode 6 or 7), and apply a high level to the FWE pin. In this mode, on-chip supporting modules other than flash memory operate as they normally would in modes 6 and 7.

The flash memory itself cannot be read while the SWE bit is set to 1 to perform programming or erasing, so the control program that performs programming and erasing should be run in on-chip RAM or external memory. When the program is located in external memory, an instruction for programming the flash memory and the following instruction should be located in on-chip RAM.

Figure 19.40 shows the procedure for executing the program/erase control program when transferred to on-chip RAM.

control program if necessary) beforehand



Notes: Do not apply a constant high level to the FWE pin. Apply a high level to the FWE pin 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 19.21, Flash Memory Programming and Erasing Precautions.

Figure 19.40 User Program Mode Execution Procedure

In the on-board programming modes, flash memory programming and erasing is performed by software, using the CPU. There are four flash memory operating modes: program mode, erase mode, program-verify mode, and erase-verify mode. Transition to these modes can be made for the on-chip ROM area by setting the PSU, ESU, P, E, PV, and EV bits in FLMCR1.

The flash memory cannot be read while being programmed or erased. Therefore, the program that controls flash memory programming/erasing (the programming control program) should be located and executed in on-chip RAM or external memory. When the program is located in external memory, an instruction for programming the flash memory and the following instruction should be located in on-chip RAM. The DMAC or DTC should not be activated before or after the instruction for programming the flash memory is executed.

Notes: 1. Operation is not guaranteed if setting/resetting of the SWE, ESU, PSU, EV, PV, E, and P bits in FLMCR1 is executed by a program in flash memory.
2. When programming or erasing, set FWE to 1 (programming/erasing will not be executed if FWE = 0).
3. Perform programming in the erased state. Do not perform additional programming on previously programmed addresses.

19.16.1 Program Mode

Follow the procedure shown in the program/program-verify flowchart in figure 19.41 to write data or programs to flash memory. Performing program operations according to this flowchart will enable data or programs to be written to flash memory without subjecting the device to voltage stress or sacrificing program data reliability. Programming should be carried out 128 bytes at a time.

For the wait times (x, y, z1, z2, z3, α, ß, γ, ε, η, θ) after bits are set or cleared in flash memory control register 1 (FLMCR1) and the maximum number of programming operations (N), see section 22.2.6, Flash Memory Characteristics.

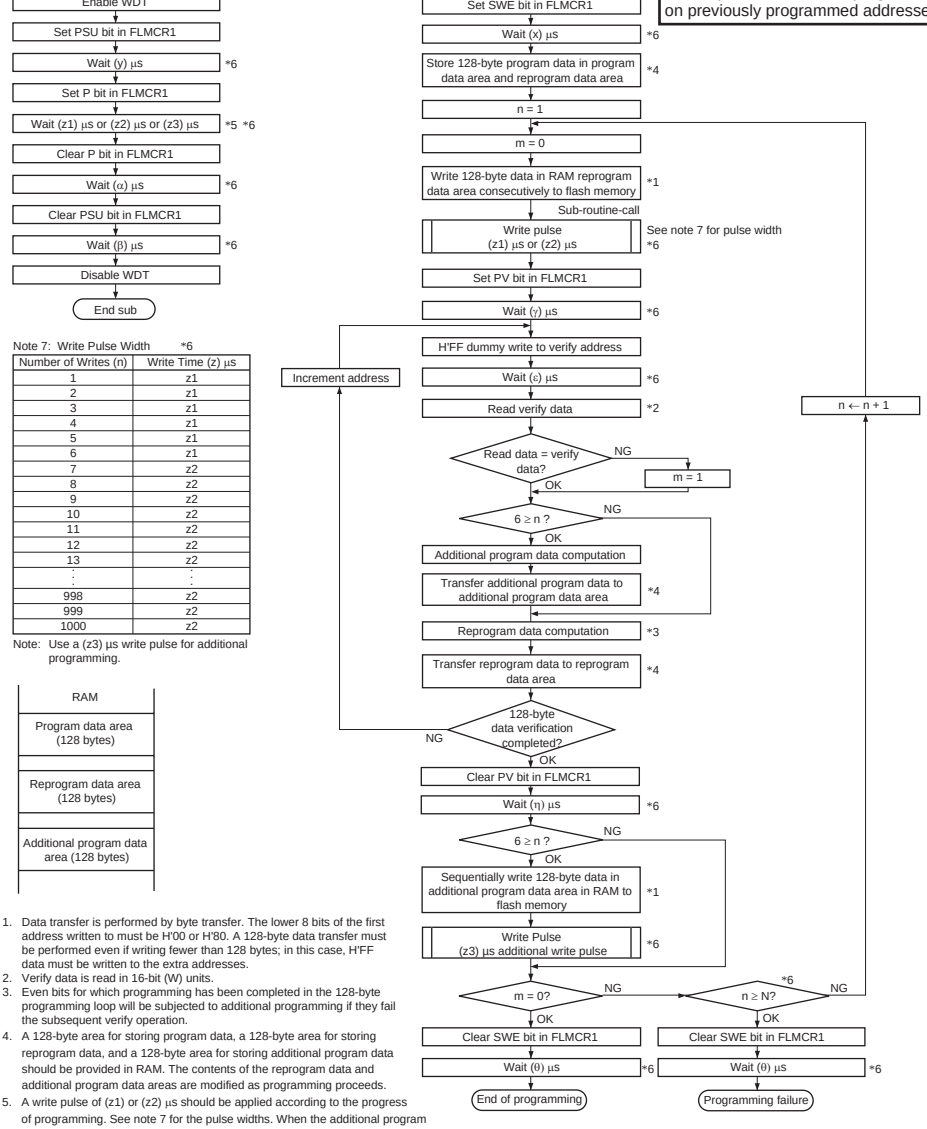
Following the elapse of (x) µs or more after the SWE bit is set to 1 in flash memory control register 1 (FLMCR1), 128-byte program data is stored in the program data area and reprogram data area, and the 128-byte data in the reprogram data area is written consecutively to the write addresses. The lower 8 bits of the first address written to must be H'00 or H'80. 128 consecutive byte data transfers are performed. The program address and program data are latched in the flash memory. 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.

for program mode (program setup) is carried out by setting the PSU bit in FLMCR1, and after the elapse of (y) µs or more, the operating mode is switched to program mode by setting the P bit in FLMCR1. The time during which the P bit is set is the flash memory programming time. Set the programming time according to the table in the programming flowchart.

19.16.2 Program-Verify Mode

In program-verify mode, the data written in program mode is read to check whether it has been correctly written in the flash memory.

After the elapse of a given programming time, the programming mode is exited (the P bit in FLMCR1 is cleared to 0, then the PSU bit is cleared to 0 at least (α) µs later). Next, the watchdog timer is cleared after the elapse of (β) µs or more, and the operating mode is switched to program-verify mode by setting the PV bit in FLMCR1. Before reading in program-verify mode, a dummy write of H'FF data should be made to the addresses to be read. The dummy write should be executed after the elapse of (γ) µs or more. When the flash memory is read in this state (verify data is read in 16-bit units), the data at the latched address is read. Wait at least (ε) µs after the dummy write before performing this read operation. Next, the originally written data is compared with the verify data, and reprogram data is computed (see figure 19.41) and transferred to the reprogram data area. After 128 bytes of data have been verified, exit program-verify mode, wait for at least (η) µs, then clear the SWE bit in FLMCR1 to 0, and wait again for at least (θ) µs. If reprogramming is necessary, set program mode again, and repeat the program/program-verify sequence as before. However, ensure that the program/program-verify sequence is not repeated more than (N) times on the same bits.



on previously programmed addresses.

Note 7: Write Pulse Width *6

Number of Writes (n)	Write Time (z) µs
1	z1
2	z1
3	z1
4	z1
5	z1
6	z1
7	z2
8	z2
9	z2
10	z2
11	z2
12	z2
13	z2
:	:
998	z2
999	z2
1000	z2

Note: Use a (z3) µs write pulse for additional programming.

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 (W) units.
3. Even bits for which programming has been completed in the 128-byte programming loop will be subjected to additional programming 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 program data should be provided in RAM. The contents of the reprogram data and additional program data areas are modified as programming proceeds.
5. A write pulse of (z1) or (z2) µs should be applied according to the progress of programming. See note 7 for the pulse widths. When the additional program data is programmed, a write pulse of (z3) µs should be applied. Reprogram data X' stands for reprogram data to which a write pulse has been applied.
6. For the values of x, y, z1, z2, z3, α, β, γ, ε, η, θ, and N, see section 22.2.6, Flash Memory Characteristics.

Program Data Operation Chart

Original Data (D)	Verify Data (V)	Reprogram Data (X)	Comments
0	0	1	Programming completed
	1	0	Programming incomplete; reprogram
1	0	1	
	1		Still in erased state; no action

Additional Program Data Operation Chart

Reprogram Data (X')	Verify Data (V)	Additional Program Data (Y)	Comments
0	0	0	Additional programming executed
	1	1	Additional programming not executed
1	0		Additional programming not executed
	1		Additional programming not executed

Figure 19.41 Program/Program-Verify Flowchart

Flash memory erasing should be performed block by block following the procedure shown in the erase/erase-verify flowchart (single-block erase) shown in figure 19.42.

For the wait times (x, y, z, α, ß, γ, ε, η, θ) after bits are set or cleared in flash memory control register 1 (FLMCR1) and the maximum number of programming operations (N), see section 22.2.6, Flash Memory Characteristics.

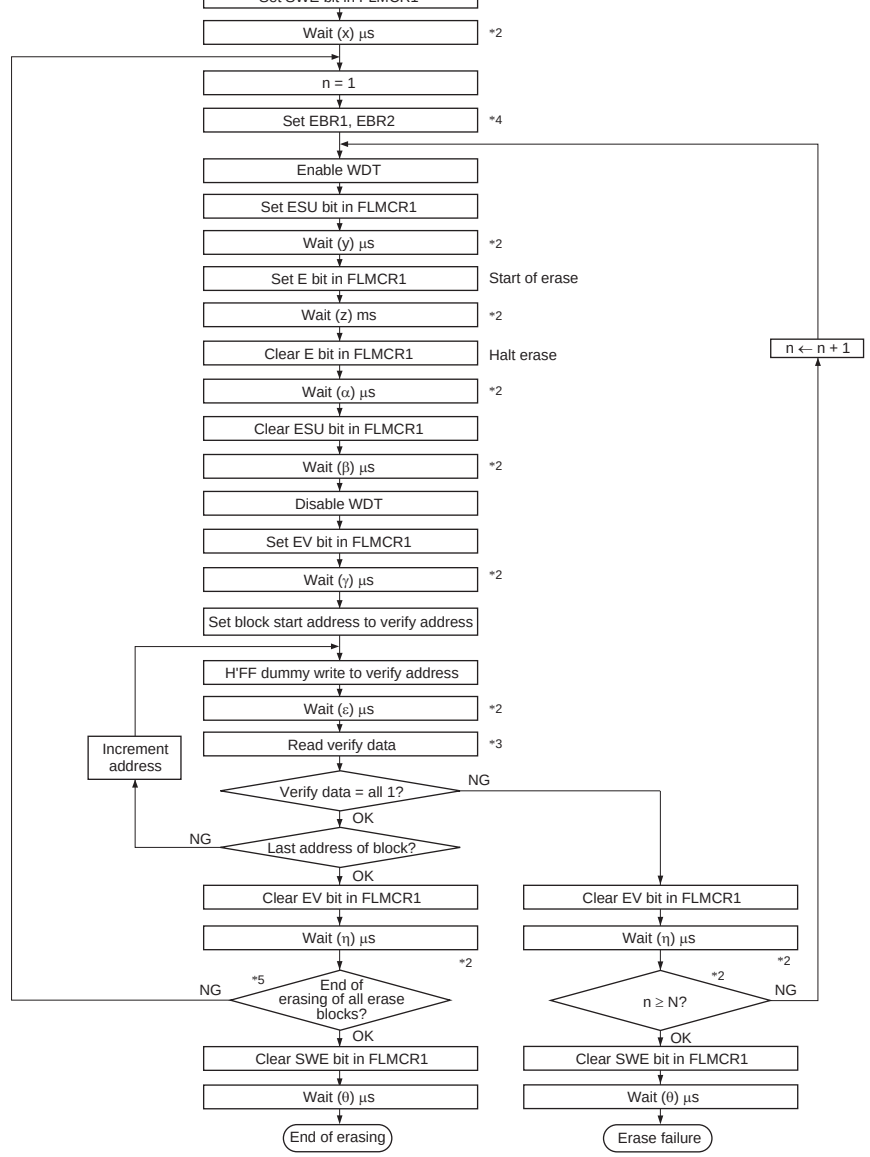
To perform data or program erasure, make a 1 bit setting for the flash memory area to be erased in erase block register 1 or 2 (EBR1 or EBR2) at least (x) µs after setting the SWE bit to 1 in flash memory control register 1 (FLMCR1). Next, the watchdog timer is set to prevent overerasing in the event of program runaway, etc. Set a value greater than (y + z + α + ß) ms as the WDT overflow period. After this, preparation for erase mode (erase setup) is carried out by setting the ESU bit in FLMCR1, and after the elapse of (y) µs or more, the operating mode is switched to erase mode by setting the E bit in FLMCR1. The time during which the E bit is set is the flash memory erase time. Ensure that the erase time does not exceed (z) ms.

Note: With flash memory erasing, prewriting (setting all data in the memory to be erased to 0) is not necessary before starting the erase procedure.

19.16.4 Erase-Verify Mode

In erase-verify mode, data is read after memory has been erased to check whether it has been correctly erased.

After the elapse of the erase time, erase mode is exited (the E bit in FLMCR1 is cleared to 0, then the ESU bit in FLMCR1 is cleared to 0 at least (α) µs later), the watchdog timer is cleared after the elapse of (β) µs or more, and the operating mode is switched to erase-verify mode by setting the EV bit in FLMCR1. Before reading in erase-verify mode, a dummy write of H'FF data should be made to the addresses to be read. The dummy write should be executed after the elapse of (γ) µs or more. When the flash memory is read in this state (verify data is read in 16-bit units), the data at the latched address is read. Wait at least (ε) µs after the dummy write before performing this read operation. If the read data has been erased (all 1), a dummy write is performed to the next address, and erase-verify is performed. If the read data has not been erased, set erase mode again, and repeat the erase/erase-verify sequence in the same way. However, ensure that the erase/erase-verify sequence is not repeated more than (N) times. When verification is completed, exit erase-verify mode, and wait for at least (η) µs. If erasure has been completed on all the erase blocks, clear the SWE bit in FLMCR1 to 0 and wait for at least (θ) µs. If there are any unerased blocks, make a 1 bit setting for the flash memory area to be erased, and repeat the erase/erase-verify sequence in the same way.



Notes: 1. Prewriting (setting erase block data to all 0) is not necessary.
2. The values of x, y, z, α, β, γ, ε, η, θ, and N are shown in the section 22.2.6, Flash Memory Characteristics.
3. Verify data is read in 16-bit (W) units.
4. Set only one bit in EBR1or EBR2. More than one bit cannot be set.
5. Erasing is performed in block units. To erase a number of blocks, the individual blocks must be erased sequentially.

Figure 19.42 Erase/Erase-Verify Flowchart

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

19.17.1 Hardware Protection

Hardware protection refers to a state in which programming/erasing of flash memory is forcibly disabled or aborted. Settings in flash memory control registers 1 and 2 (FLMCR1, FLMCR2) and erase block registers 1 and 2 (EBR1, EBR2) are reset. (See table 19.32.)

Table 19.32 Hardware Protection

		Functions	
Item	**Description**	**Program**	**Erase**
FWE pin protection	• When a low level is input to the FWE pin, FLMCR1, FLMCR2, EBR1, and EBR2 are initialized, and the program/erase-protected state is entered.	Yes	Yes
Reset/standby protection	• In a reset (including a WDT overflow 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 section 22.2.3, AC Characteristics.	Yes	Yes

19.17.2 Software Protection

Software protection can be implemented by setting the SWE bit in flash memory control register 1 (FLMCR1), erase block registers 1 and 2 (EBR1, EBR2), and the RAMS bit in the RAM emulation register (RAMER). When software protection is in effect, setting the P or E bit in FLMCR1 does not cause a transition to program mode or erase mode. (See table 19.33.)

Item	Description	Functions: Program	Functions: Erase
SWE bit protection	• Clearing the SWE bit to 0 in FLMCR1 sets the program/erase-protected state for all blocks • (Execute in on-chip RAM or external memory.)	Yes	Yes
Block specification protection	• Erase protection can be set for individual blocks by settings in erase block registers 1 and 2 (EBR1, EBR2). • Setting EBR1 and EBR2 to H'00 places all blocks in the erase-protected state.	—	Yes
Emulation protection	• Setting the RAMS bit to 1 in the RAM emulation register (RAMER) places all blocks in the program/erase-protected state.	Yes	Yes

19.17.3 Error Protection

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

If the MCU malfunctions during flash memory programming/erasing, the FLER bit is set to 1 in FLMCR2 and the error protection state is entered. The FLMCR1, FLMCR2, EBR1, and EBR2 settings are retained, but program mode or erase mode is aborted at the point at which the error occurred. Program mode or erase mode cannot be re-entered by re-setting the P or E bit. However, PV and EV bit setting is enabled, and a transition can be made to verify mode.

FLER bit setting conditions are as follows:

- When flash memory is read during programming/erasing (including a vector read or instruction fetch)
- Immediately after exception handling (excluding a reset) during programming/erasing
- When a SLEEP instruction (including software standby) is executed during programming/erasing
- When a bus master other than the CPU (the DMAC or DTC) has control of the bus during programming/erasing

Error protection is released only by a reset and in hardware standby mode.

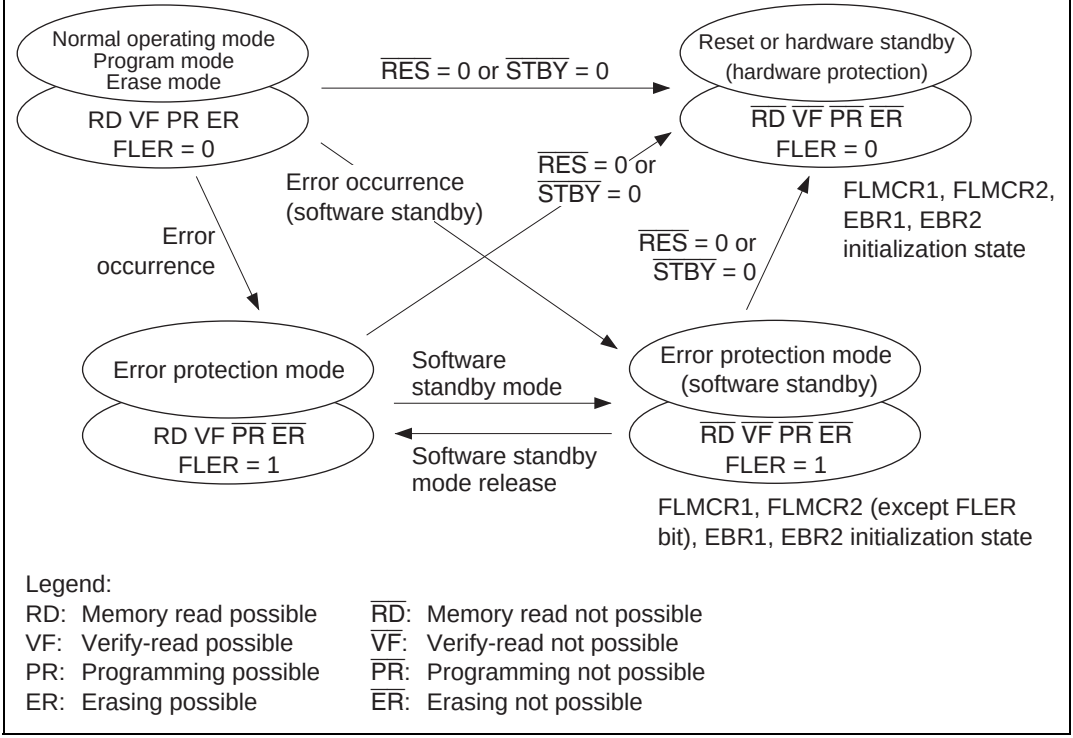


Figure 19.43 Flash Memory State Transitions

19.18.1 Emulation in RAM

Making a setting in the RAM emulation register (RAMER) enables part of RAM to be overlapped onto the flash memory area so that data to be written to flash memory can be emulated in RAM in real time. After the RAMER setting has been made, accesses can be made from the flash memory area or the RAM area overlapping flash memory. Emulation can be performed in user mode and user program mode. Figure 19.44 shows an example of emulation of real-time flash memory programming.

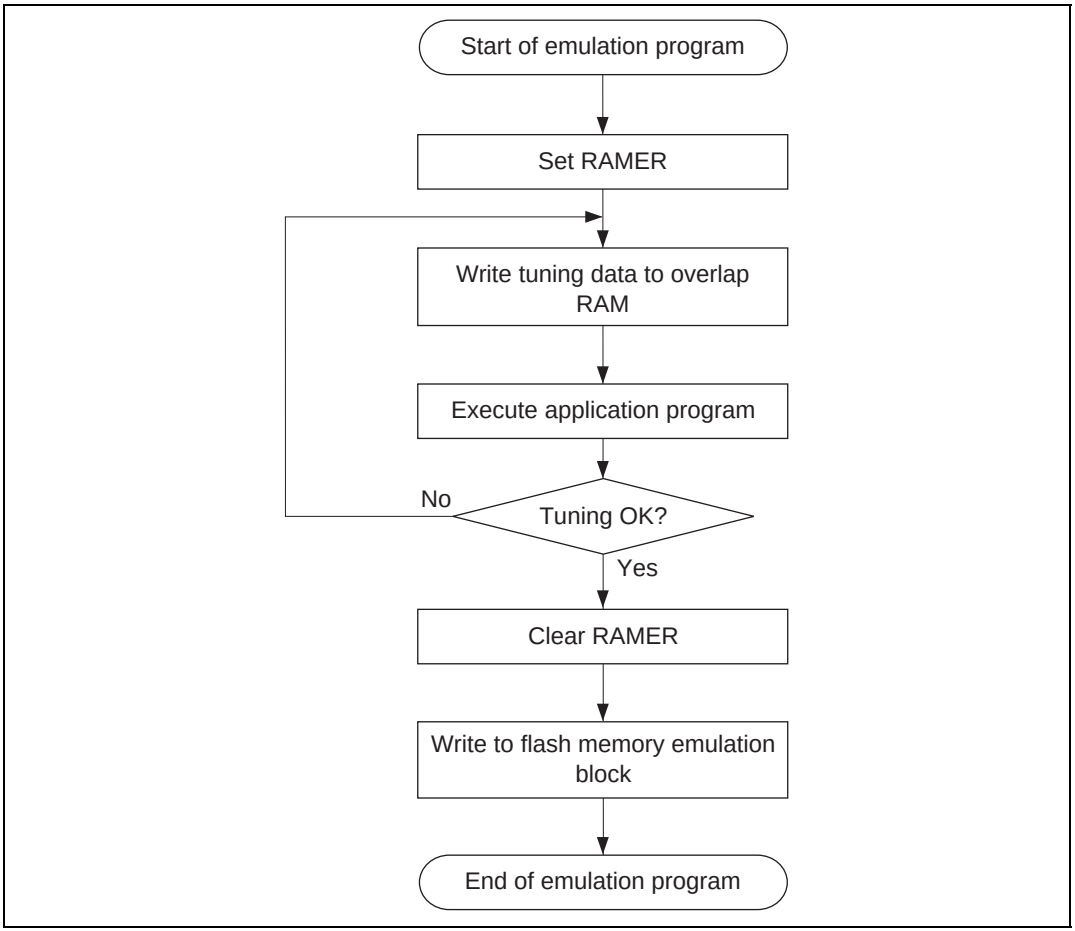


Figure 19.44 Flowchart for Flash Memory Emulation in RAM

An example in which flash memory block area EB1 is overlapped is shown below.

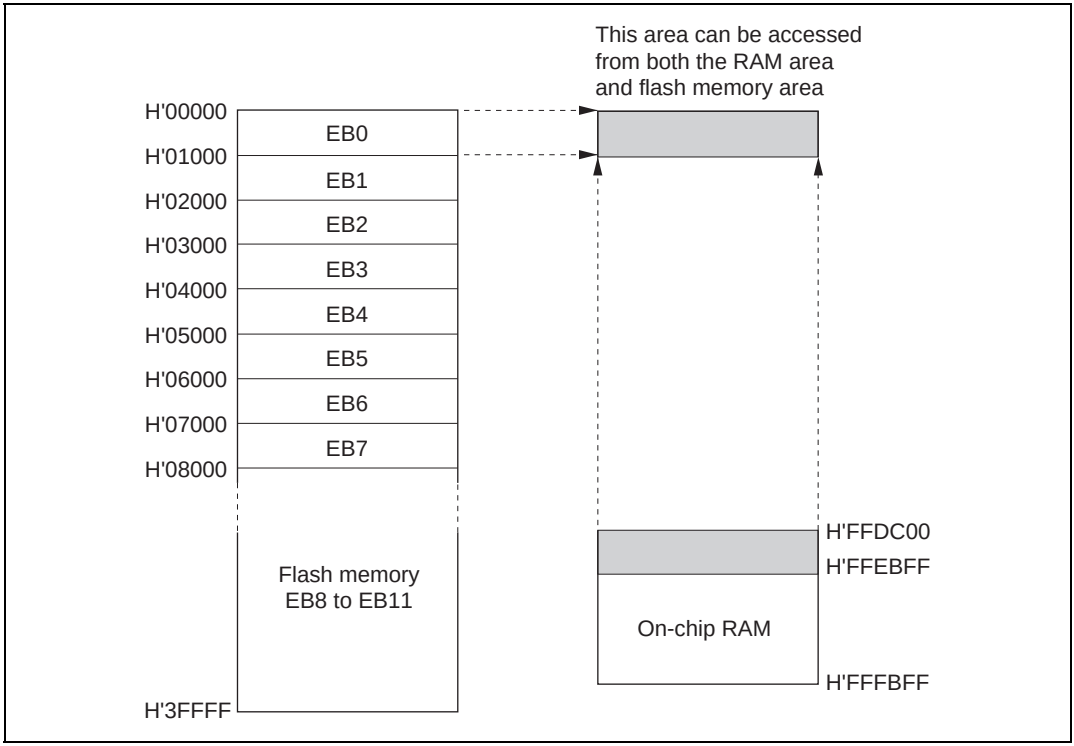


Figure 19.45 Example of RAM Overlap Operation

Example in which Flash Memory Block Area EB1 is Overlapped

1. Set bits RAMS, RAM2, RAM1, and RAM0 in RAMER to 1, 0, 0, 1, to overlap part of RAM onto the area (EB1) 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 RAM overlap.
4. The data written in the overlapping RAM is written into the flash memory space (EB1).

Notes: 1. When the RAMS bit is set to 1, program/erase protection is enabled for all blocks regardless of the value of RAM2, RAM1, and RAM0 (emulation protection). In this state, setting the P or E bit in flash memory control register 1 (FLMCR1) will not cause a transition to program mode or erase mode. When actually programming a flash memory area, the RAMS bit should be cleared to 0.

3. Block area EB0 includes the vector table. When performing RAM emulation, the vector table is needed by the overlap RAM.

19.19 Interrupt Handling when Programming/Erasing Flash Memory

All interrupts, including NMI input, are disabled when flash memory is being programmed or erased (when the P or E bit is set in FLMCR1), and while the boot program is executing in boot mode*1, to give priority to the program or erase operation. There are three reasons for this:

1. Interrupt during programming or erasing might cause a violation of the programming or erasing algorithm, with the result that normal operation could not be assured.
2. In the interrupt exception handling sequence during programming or erasing, the vector would not be read correctly*2, possibly resulting in MCU runaway.
3. If an interrupt occurred during boot program execution, it would not be possible to execute the normal boot mode sequence.

For these reasons, in on-board programming mode alone there are conditions for disabling interrupts, as an exception to the general rule. However, this provision does not guarantee normal erasing and programming or MCU operation. All interrupt requests, including NMI, must therefore be restricted inside and outside the MCU when programming or erasing flash memory. The NMI interrupt is also disabled in the error-protection state while the P or E bit remains set in FLMCR1.

Notes: 1. Interrupt requests must be disabled inside and outside the MCU until the programming control program has completed programming.
2. The vector may not be read correctly in this case for the following two reasons:
   - If flash memory is read while being programmed or erased (while the P or E bit is set in FLMCR1), correct read data will not be obtained (undetermined values will be returned).
   - If the interrupt entry in the vector table has not been programmed yet, interrupt exception handling will not be executed correctly.

19.20.1 PROM Mode Setting

Programs and data can be written and erased in PROM mode as well as in the on-board programming modes. In PROM mode, the on-chip ROM can be freely programmed using a PROM programmer that supports the Renesas microcomputer device type with 256-kbyte on-chip flash memory (FZTAT256V3A). Flash memory read mode, auto-program mode, auto-erase mode, and status read mode are supported with this device type. In auto-program mode, auto-erase mode, and status read mode, a status polling procedure is used, and in status read mode, detailed internal signals are output after execution of an auto-program or auto-erase operation.

Table 19.34 shows PROM mode pin settings.

Table 19.34 PROM Mode Pin Settings

Pin Names	Settings/External Circuit Connection
Mode pins: MD2, MD1, MD0	Low-level input
Mode setting pins: P66, P65, P64	High-level input to P66, low-level input to P65 and P64
FWE pin	High-level input (in auto-program and auto-erase modes)
$\overline{\text{STBY}}$ pin	High-level input (do not select hardware standby mode)
$\overline{\text{RES}}$ pin	Reset circuit
XTAL, EXTAL pins	Oscillator circuit
Other pins requiring setting: P32, P25	High-level input to P32, low-level input to P25

In PROM mode, a socket adapter is connected to the chip as shown in figure 19.47. Figure 19.46 shows the on-chip ROM memory map and figure 19.47 shows the socket adapter pin assignments.

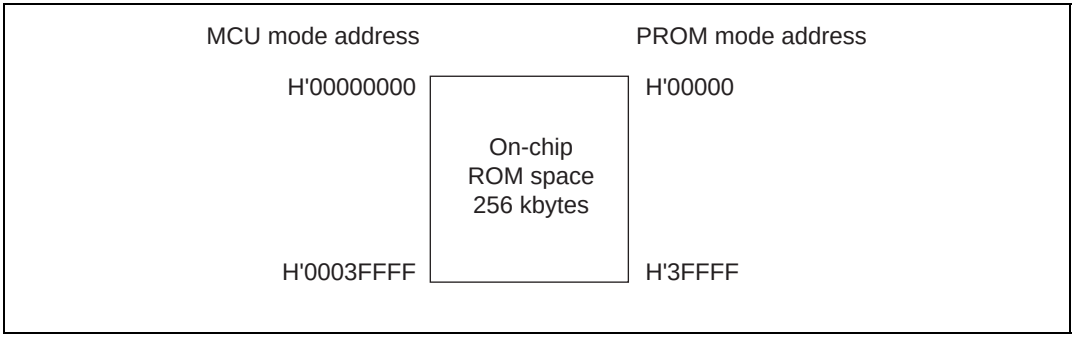
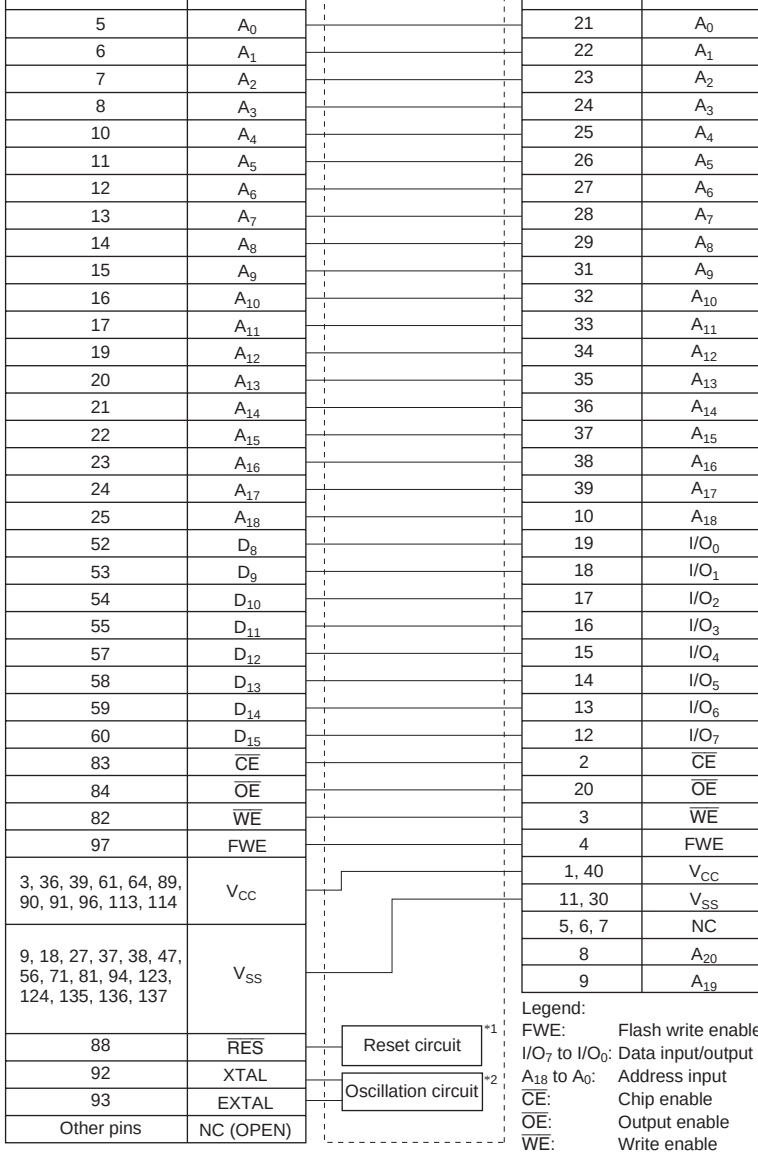


Figure 19.46 Memory Map in PROM Mode



Notes: This figure shows pin assignments, and does not show the entire socket adapter circuit.
1. A reset oscillation stabilization time (t_{osc1}) of at least 10 ms is required.
2. A 12-MHz crystal resonator should be used.

Figure 19.47 H8S/2338 F-ZTAT Socket Adapter Pin Assignments

Table 19.35 shows how the different operating modes are set when using PROM mode, and table 19.36 lists the commands used in PROM mode. Details of each mode are given below.

Memory Read Mode: Memory read mode supports byte reads.

Auto-Program Mode: Auto-program mode supports programming of 128 bytes at a time. Status polling is used to confirm the end of auto-programming.

Auto-Erase Mode: Auto-erase mode supports automatic erasing of the entire flash memory. Status polling is used to confirm the end of auto-erasing.

Status Read Mode: Status polling is used for auto-programming and auto-erasing, and normal termination can be confirmed by reading the I/O_6 signal. In status read mode, error information is output if an error occurs.

Table 19.35 Settings for Each Operating Mode in PROM Mode

	Pin Names					
Mode	**FWE**	**$\overline{CE}$**	**$\overline{OE}$**	**$\overline{WE}$**	**I/O_7 to I/O_0**	**A_{18} to A_0**
Read	H or L	L	L	H	Data output	Ain
Output disable	H or L	L	H	H	Hi-Z	X
Command write	H or L*3	L	H	L	Data input	Ain*2
Chip disable*1	H or L	H	X	X	Hi-Z	X

Legend:

H: High level

L: Low level

Hi-Z: High impedance

X: Don't care

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 when making a transition to auto-program or auto-erase mode, input a high level to the FWE pin.

Command Name	Number of Cycles	1st Cycle			2nd Cycle		
		Mode	Address	Data	Mode	Address	Data
Memory read mode	1 + n	Write	X	H'00	Read	RA	Dout
Auto-program mode	129	Write	X	H'40	Write	PA	Din
Auto-erase mode	2	Write	X	H'20	Write	X	H'20
Status read mode	2	Write	X	H'71	Write	X	H'71

Legend:

RA: Read address

PA: Program address

Notes: 1. In auto-program mode, 129 cycles are required for command writing by a simultaneous 128-byte write.

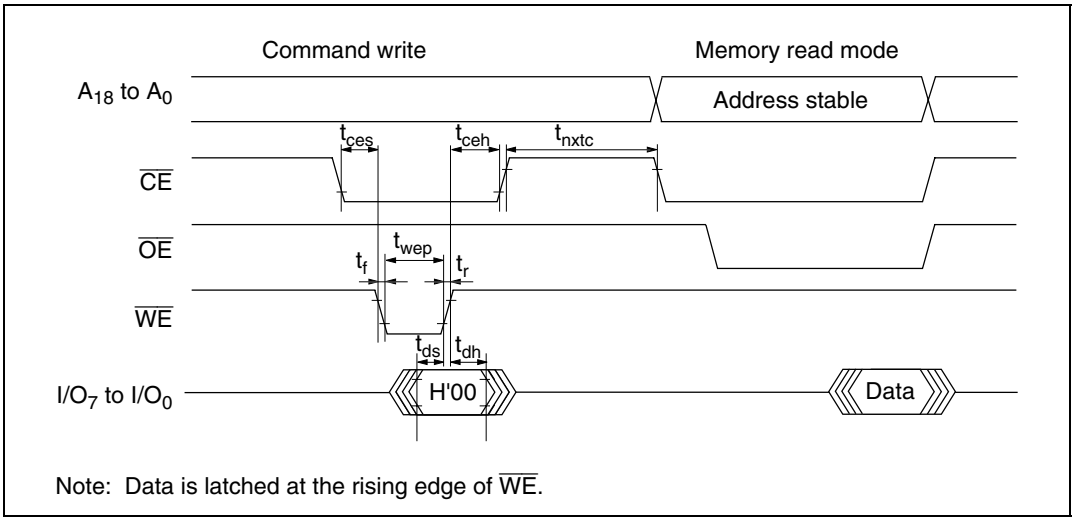
2. In memory read mode, the number of cycles depends on the number of address write cycles (n).

19.20.4 Memory Read Mode

- After the end of an auto-program, auto-erase, or status read operation, the command wait state is entered. To read memory contents, a transition must be made to memory read mode by means of a command write before the read is executed.
- Command writes can be performed in memory read mode, just as in the command wait state.
- Once memory read mode has been entered, consecutive reads can be performed.
- After power-on, memory read mode is entered.

Conditions: V_{CC} = 3.3 V ±0.3 V, V_{SS} = 0 V, T_a = 25°C ±3°C

Item	Symbol	Min	Max	Unit
Command write cycle	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
$\overline{WE}$ rise time	t_r	—	30	ns
$\overline{WE}$ fall time	t_f	—	30	ns



Note: Data is latched at the rising edge of $\overline{WE}$.

Figure 19.48 Memory Read Mode Timing Waveforms after Command Write

Conditions: V_{CC} = 3.3 V ±0.3 V, V_{SS} = 0 V, T_a = 25°C ±3°C

Item	Symbol	Min	Max	Unit
Command write cycle	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
$\overline{WE}$ rise time	t_r	—	30	ns
$\overline{WE}$ fall time	t_f	—	30	ns

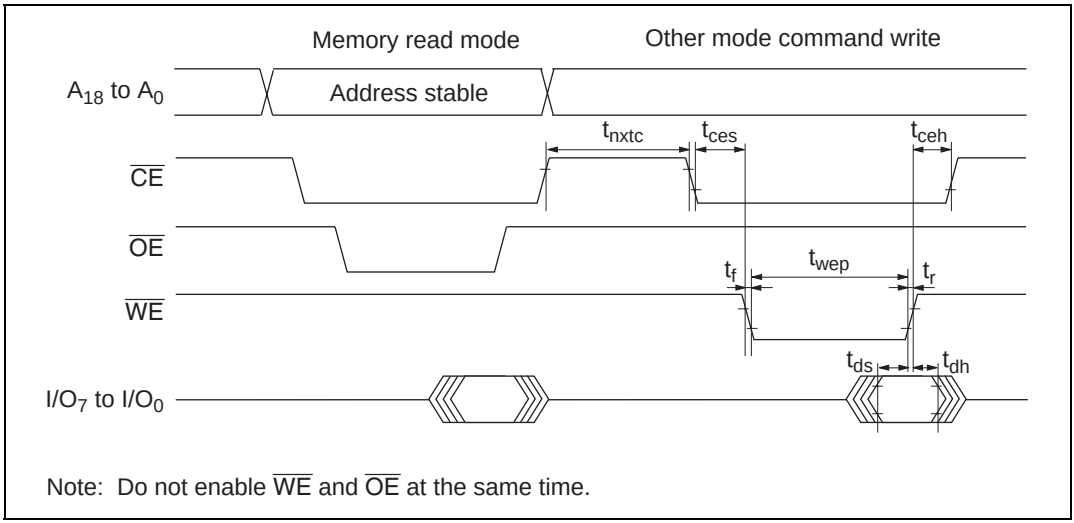


Figure 19.49 Timing Waveforms when Entering Another Mode from Memory Read Mode

Conditions: V_{CC} = 3.3 V ±0.3 V, V_{SS} = 0 V, T_a = 25°C ±3°C

Item	Symbol	Min	Max	Unit
Access time	t_{acc}	—	20	µs
$\overline{CE}$ output delay time	t_{ce}	—	150	ns
$\overline{OE}$ output delay time	t_{oe}	—	150	ns
Output disable delay time	t_{df}	—	100	ns
Data output hold time	t_{oh}	5	—	ns

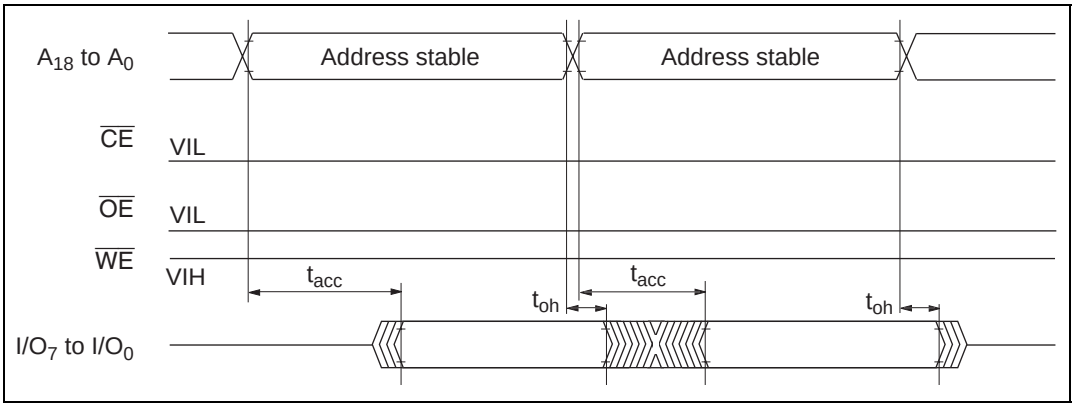


Figure 19.50 Timing Waveforms for $\overline{CE}$/$\overline{OE}$ Enable State Read

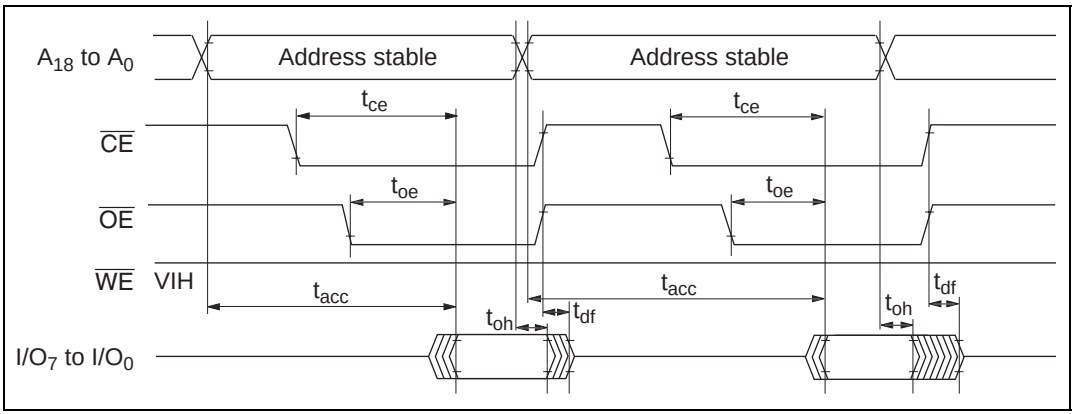


Figure 19.51 Timing Waveforms for $\overline{CE}$/$\overline{OE}$ Clocked Read

- In auto-program mode, 128 bytes are programmed simultaneously. For this purpose, 128 consecutive byte data transfers should be performed.
- 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.
- The lower 7 bits of the transfer address must be held low. If an invalid address is input, memory programming will be started but a programming error will occur.
- Memory address transfer is executed in the second cycle (figure 19.52). Do not perform transfer later than the second cycle.
- Do not perform a command write during a programming operation.
- Perform one auto-programming operation for a 128-byte block for each address. One or more additional programming operations cannot be carried out on address blocks that have already been programmed.
- Confirm normal end of auto-programming by checking I/O_6. Alternatively, status read mode can also be used for this purpose (the I/O_7 status polling pin is used to identify the end of an auto-program operation).
- Status polling I/O_6 and I/O_7 information is retained until the next command write. As long as the next command write has not been performed, reading is possible by enabling $\overline{CE}$ and $\overline{OE}$.

Table 19.40 AC Characteristics in Auto-Program Mode

Conditions: V_{CC} = 3.3 V ±0.3 V, V_{SS} = 0 V, T_a = 25°C ±5°C

Item	Symbol	Min	Max	Unit
Command write cycle	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
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
$\overline{WE}$ rise time	t_r	—	30	ns
$\overline{WE}$ fall time	t_f	—	30	ns
Write setup time	t_{pns}	100	—	ns
Write end setup time	t_{pnh}	100	—	ns

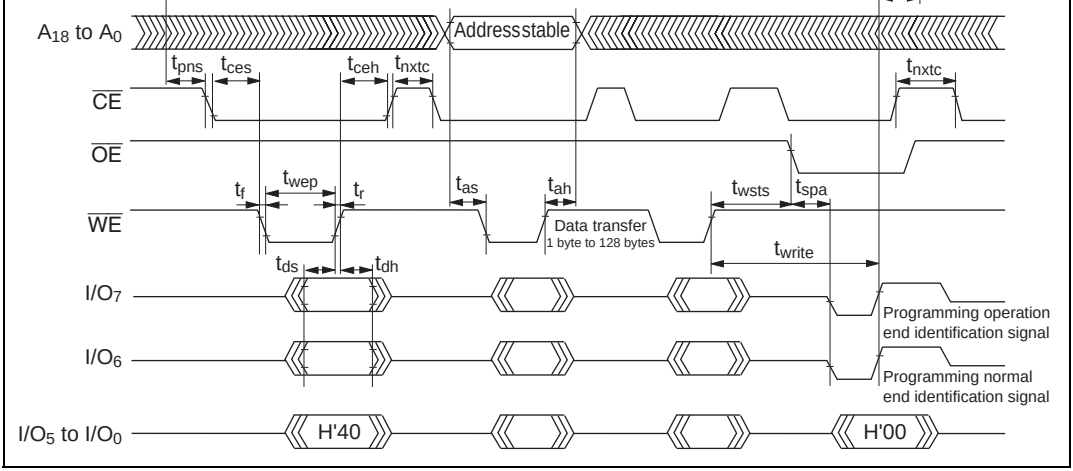


Figure 19.52 Auto-Program Mode Timing Waveforms

19.20.6 Auto-Erase Mode

- Auto-erase mode supports only total memory erasing.
- Do not perform a command write during auto-erasing.
- Confirm normal end of auto-erasing by checking I/O_6. Alternatively, status read mode can also be used for this purpose (the I/O_7 status polling pin is used to identify the end of an auto-erase operation).
- Status polling I/O_6 and I/O_7 pin information is retained until the next command write. As long as the next command write has not been performed, reading is possible by enabling $\overline{CE}$ and $\overline{OE}$.

Table 19.41 AC Characteristics in Auto-Erase Mode

Conditions: V_{CC} = 3.3 V ±0.3 V, V_{SS} = 0 V, T_a = 25°C ±5°C

Item	Symbol	Min	Max	Unit
Command write cycle	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
Status polling start time	t_{ests}	1	—	ms
Status polling access time	t_{spa}	—	150	ns
Memory erase time	t_{erase}	100	40000	ms
$\overline{WE}$ rise time	t_r	—	30	ns
$\overline{WE}$ fall time	t_f	—	30	ns
Erase setup time	t_{ens}	100	—	ns
Erase end setup time	t_{enh}	100	—	ns

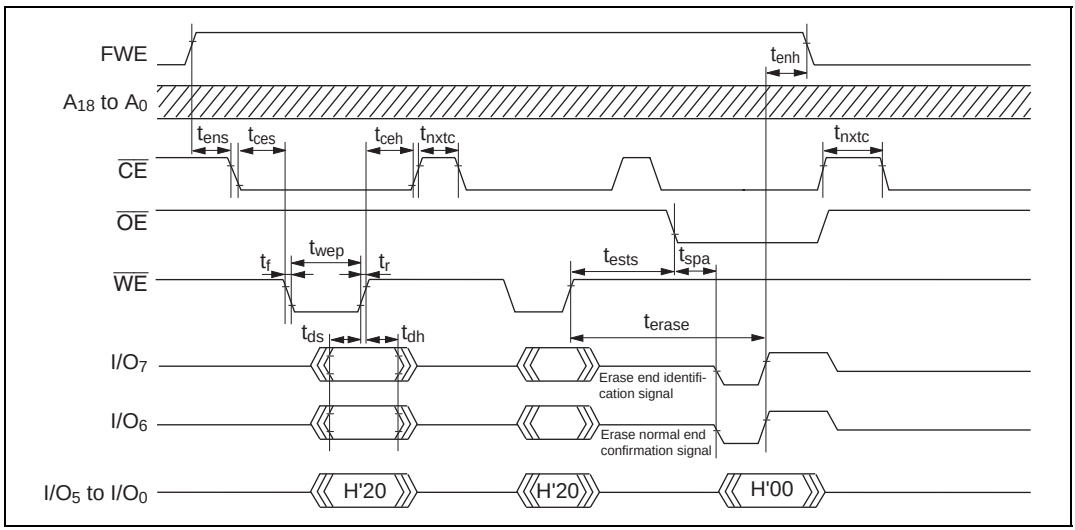


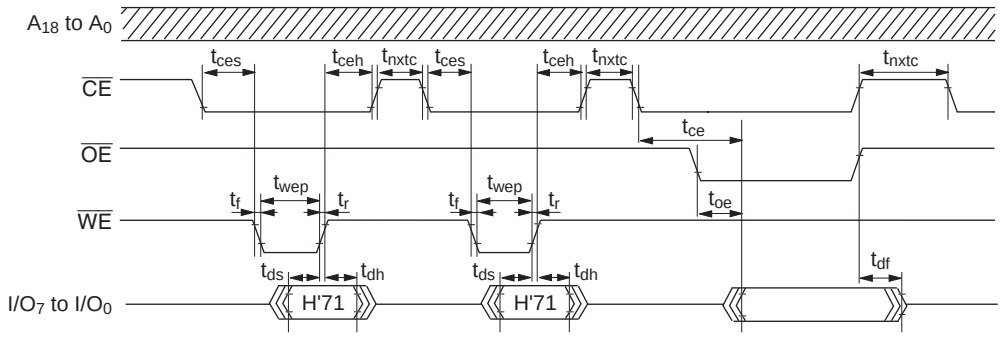
Figure 19.53 Auto-Erase Mode Timing Waveforms

- Status read mode is used to identify what type of abnormal end has occurred. Use this mode when an abnormal end occurs in auto-program mode or auto-erase mode.
- The return code is retained until a command write for other than status read mode is performed.

Table 19.42 AC Characteristics in Status Read Mode

Conditions: V_{CC} = 3.3 V ±0.3 V, V_{SS} = 0 V, T_a = 25°C ±5°C

Item	Symbol	Min	Max	Unit
Command write cycle	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/O_3 and I/O_2 are undefined.

Figure 19.54 Status Read Mode Timing Waveforms

Pin Name	I/O_7	I/O_6	I/O_5	I/O_4	I/O_3	I/O_2	I/O_1	I/O_0
Attribute	Normal end identification	Command error	Program-ming error	Erase error	—	—	Program-ming or erase count exceeded	Effective address error
Initial value	0	0	0	0	0	0	0	0
Indications	Normal end: 0 Abnormal end: 1	Command error: 1 Otherwise: 0	Program-ming error: 1 Otherwise: 0	Erase error: 1 Otherwise: 0	—	—	Count exceeded: 1 Otherwise: 0	Effective address error: 1 Otherwise: 0

Note: I/O_3 and I/O_2 are undefined.

19.20.8 Status Polling

- The I/O_7 status polling flag indicates the operating status in auto-program or auto-erase mode.
- The I/O_6 status polling flag indicates a normal or abnormal end in auto-program or auto-erase mode.

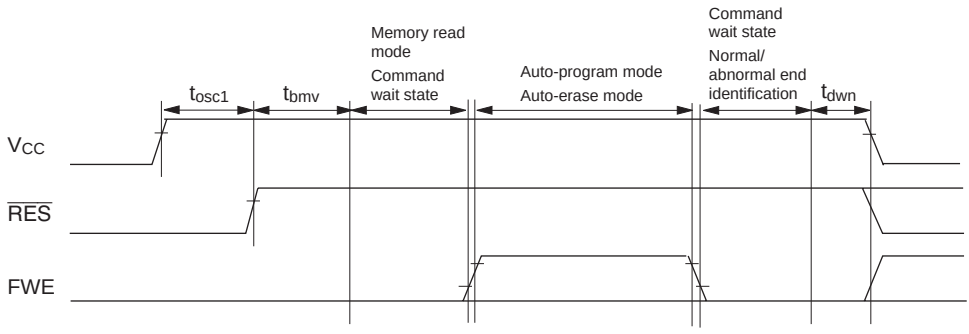
Table 19.44 Status Polling Output Truth Table

Pin Names	Internal Operation in Progress	Abnormal End	—	Normal End
I/O_7	0	1	0	1
I/O_6	0	0	1	1
I/O_0 to I/O_5	0	0	0	0

Commands cannot be accepted during the oscillation stabilization period or the PROM mode setup period. After the PROM mode setup time, a transition is made to memory read mode.

Table 19.45 Command Wait State Transition Time Specifications

Item	Symbol	Min	Max	Unit
Standby release (oscillation stabilization time)	t_{osc1}	30	—	ms
PROM mode setup time	t_{bmv}	10	—	ms
V_{CC} hold time	t_{dwn}	0	—	ms



Note: Except in auto-program mode and auto-erase mode, drive the FWE input pin low.

Figure 19.55 Oscillation Stabilization Time, PROM Mode Setup Time, and Power Supply Fall Sequence

- When programming addresses which have previously been programmed, carry out auto-erasing before auto-programming.
- When performing programming using PROM mode on a chip that has been programmed/erased in an on-board programming mode, auto-erasing is recommended before carrying out auto-programming.

Notes: 1. The flash memory is initially in the erased state when the device is shipped by Renesas Technology. For other chips for which the erasure history is unknown, it is recommended that auto-erasing be executed to check and supplement the initialization (erase) level.

2. Auto-programming should be performed once only on the same address block. Additional programming cannot be carried out on address blocks that have already been programmed.

19.21 Flash Memory Programming and Erasing Precautions

Precautions concerning the use of on-board programming mode, the RAM emulation function, and PROM mode are summarized below.

Use the specified voltages and timing for programming and erasing: Applied voltages in excess of the rating can permanently damage the device. Use a PROM programmer that supports the Renesas microcomputer device type with 256-kbyte on-chip flash memory (FZTAT256V3A).

Do not select the HN27C4096 setting for the PROM programmer, and only use the specified socket adapter. Failure to observe these points may result in damage to the device.

Powering on and off (see figures 19.56 to 19.58): Do not apply a high level to the FWE pin until V_{CC} has stabilized. Also, drive the FWE pin low before turning off V_{CC}.

When applying or disconnecting V_{CC} power, fix the FWE pin low and place the flash memory in the hardware protection state.

The power-on and power-off timing requirements should also be satisfied in the event of a power failure and subsequent recovery.

FWE application/disconnection (see figures 19.56 to 19.58): FWE application should be carried out when MCU operation is in a stable condition. If MCU operation is not stable, fix the FWE pin low and set the protection state.

- 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 stabilization time).
- In boot mode, apply and disconnect FWE during a reset.
- In user program mode, FWE can be switched between high and low level regardless of the reset state. 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 SWE, ESU, PSU, EV, PV, P, and E bits in FLMCR1 are cleared.

  Make sure that the SWE, ESU, PSU, EV, PV, P, and E bits are not set by mistake when applying or disconnecting FWE.

Do not apply a constant high level to the FWE pin: Apply a high level to the FWE pin only when programming or erasing flash memory. A system configuration in which a high level is constantly applied to the FWE pin should be avoided. 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.

Use the recommended algorithm when programming and erasing flash memory: The recommended algorithm enables programming and erasing to be carried out without subjecting the device to voltage stress or sacrificing program data reliability. When setting the P or E bit in FLMCR1, the watchdog timer should be set beforehand as a precaution against program runaway, etc.

Do not set or clear the SWE bit during execution of a program in flash memory: Wait for at least 100 μs after clearing the SWE bit before executing a program or reading data in flash memory. When the SWE bit is set, data in flash memory can be rewritten, but when SWE = 1, flash memory can only be read in program-verify or erase-verify mode. Access flash memory only for verify operations (verification during programming/erasing). Also, do not clear the SWE bit during programming, erasing, or verifying.

Similarly, when using the RAM emulation function while a high level is being input to the FWE pin, the SWE bit must be cleared before executing a program or reading data in flash memory.

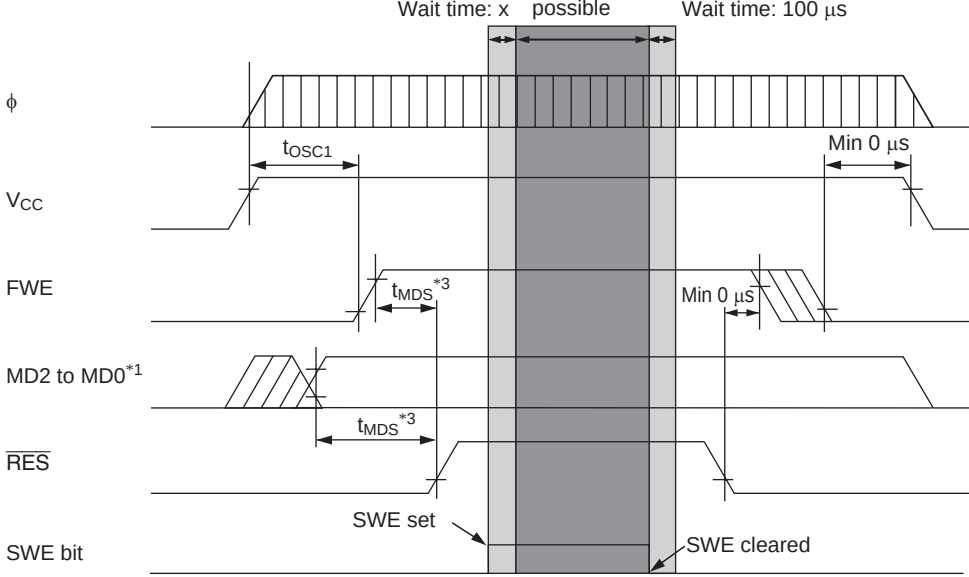
However, the RAM area overlapping flash memory space can be read and written to regardless of whether the SWE bit is set or cleared.

program/erase operations.

Do not perform additional programming. Erase the memory before reprogramming: In on-board programming, perform only one programming operation on a 128-byte programming unit block. In PROM mode, too, perform only one programming operation on a 128-byte programming unit block. Programming should be carried out with the entire programming unit block erased.

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 programmer socket, socket adapter, and chip are not correctly aligned.

Do not touch the socket adapter or chip during programming: Touching either of these can cause contact faults and write errors.

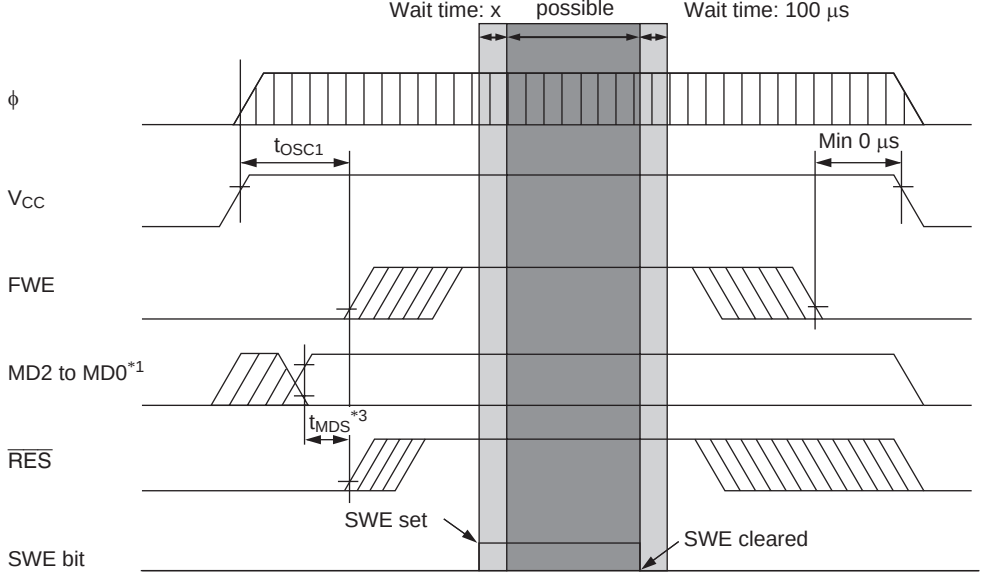


Period during which flash memory access is prohibited
(x: Wait time after setting SWE bit)*2

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 to MD0) must be fixed until power-off by pulling the pins up or down.
2. See section 22.2.6, Flash Memory Characteristics.
3. Mode programming setup time t_{MDS} (min) = 200 ns

Figure 19.56 Power-On/Off Timing (Boot Mode)

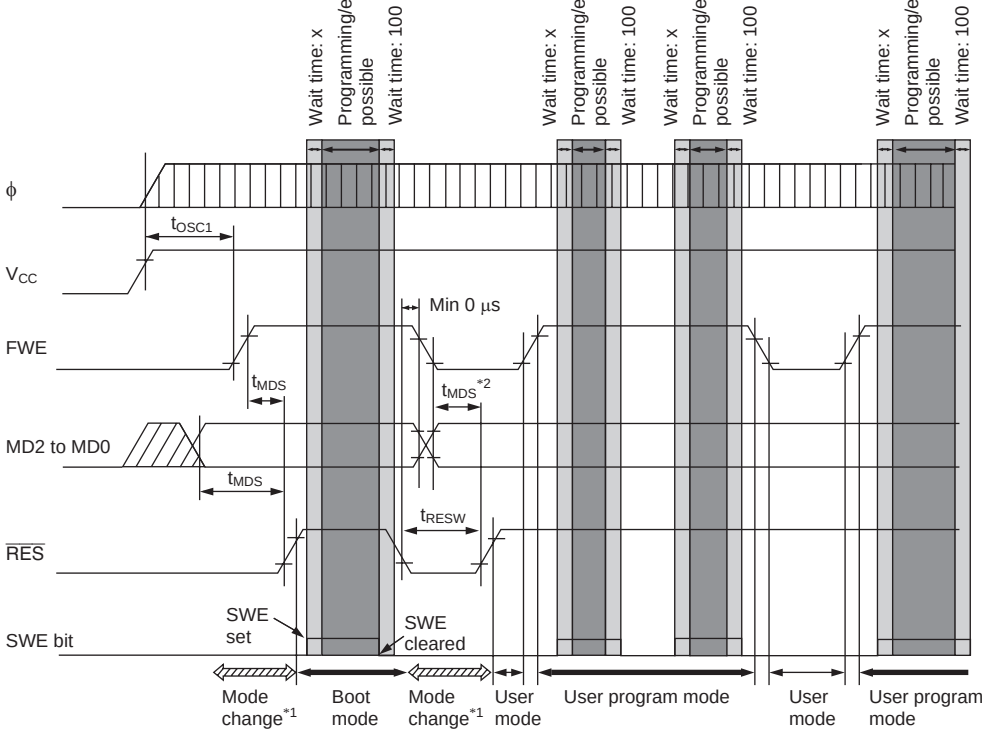


Period during which flash memory access is prohibited
(x: Wait time after setting SWE bit)*2

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 to MD0) must be fixed until power-off by pulling the pins up or down.
2. See section 22.2.6, Flash Memory Characteristics.
3. Mode programming setup time t_{MDS} (min) = 200 ns

Figure 19.57 Power-On/Off Timing (User Program Mode)



Period during which flash memory access is prohibited
(x: Wait time after setting SWE 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 must be carried out by means of $\overline{RES}$ input. The state of ports with multiplexed address functions and bus control output pins ($\overline{AS}$, $\overline{RD}$, $\overline{WR}$) will change during this switchover interval (the interval during which the $\overline{RES}$ pin input is low), 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} (min) of 200 ns is necessary with respect to $\overline{RES}$ clearance timing.
3. See section 22.2.6, Flash Memory Characteristics.

Figure 19.58 Mode Transition Timing
(Example: Boot Mode → User Mode ↔ User Program Mode)

20.1 Overview

The chip has an on-chip clock pulse generator (CPG) that generates the system clock (ϕ), the bus master clock, and internal clocks.

The clock pulse generator consists of an oscillator circuit, a duty adjustment circuit, a medium-speed clock divider, and a bus master clock selection circuit.

In the chip, the CPG has a medium-speed mode in which the bus master runs on a medium-speed clock and the other supporting modules run on the high-speed clock, and a function that allows the medium-speed mode to be disabled and the clock division ratio to be changed for the entire chip. A clock from ϕ/2 to ϕ/32 can be selected.

20.1.1 Block Diagram

Figure 20.1 shows a block diagram of the clock pulse generator.

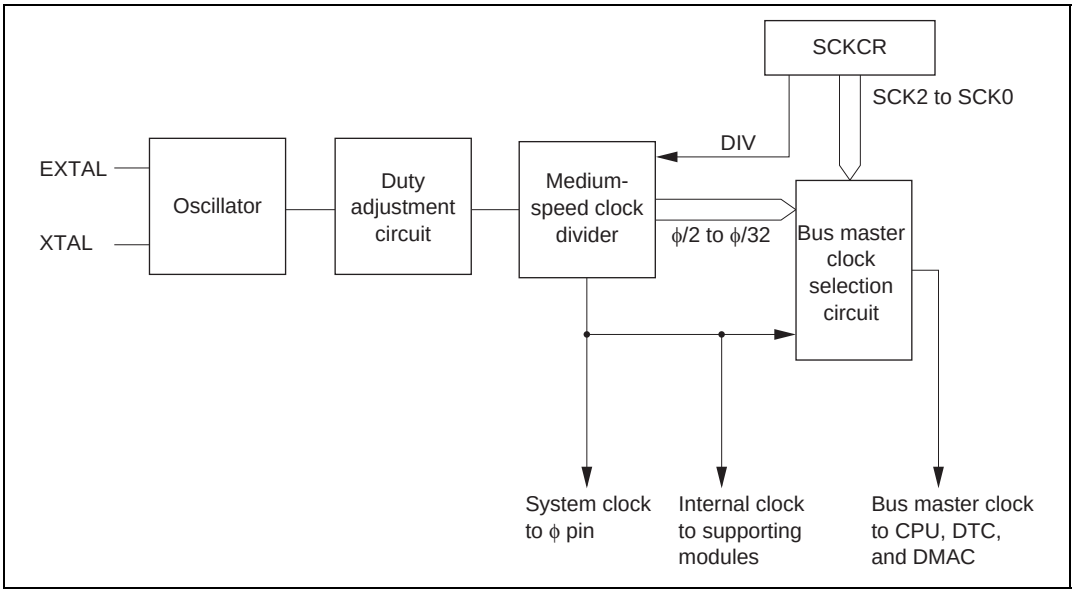


Figure 20.1 Block Diagram of Clock Pulse Generator

The clock pulse generator is controlled by SCKCR. Table 20.1 shows the register configuration.

Table 20.1 Clock Pulse Generator Register

Name	Abbreviation	R/W	Initial Value	Address*
System clock control register	SCKCR	R/W	H'00	H'FF3A

Note: * Lower 16 bits of the address.

20.2 Register Descriptions

20.2.1 System Clock Control Register (SCKCR)

Bit	:	7	6	5	4	3	2	1	0
		PSTOP	—	DIV	—	—	SCK2	SCK1	SCK0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	—	—	R/W	R/W	R/W

SCKCR is an 8-bit readable/writable register that controls φ clock output, the medium-speed mode in which the bus master runs on a medium-speed clock and the other supporting modules run on the high-speed clock, and a function that allows the medium-speed mode to be disabled and the clock division ratio to be changed for the entire chip.

SCKCR is initialized to H'00 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bit 7—φ Clock Output Disable (PSTOP): Controls φ output.

	Description			
Bit 7 PSTOP	**Normal Operation**	**Sleep Mode**	**Software Standby Mode**	**Hardware Standby Mode**
0	φ output (Initial value)	φ output	Fixed high	High impedance
1	Fixed high	Fixed high	Fixed high	High impedance

Bit 6—Reserved: This bit can be read or written to, but only 0 should be written.

the entire chip. In this way, the current dissipation within the chip is reduced in proportion to the division ratio. As the frequency of ϕ changes, the following points must be noted.

- The division ratio set with bits SCK2 to SCK0 should be selected so as to fall within the guaranteed operation range of clock cycle time tcyc given in the AC timing table in the Electrical Characteristics section. Ensure that ϕ min = 2 MHz, and the condition $\phi < 2$ MHz does not arise.

- All internal modules basically operate on ϕ. Note, therefore, that time processing involving the timers, the SCI, etc., will change when the division ratio changes. The wait time when software standby is cleared will also change in line with a change in the division ratio.

- The division ratio can be changed while the chip is operating. The clock output from the ϕ pin will also change when the division ratio is changed. The frequency of the clock output from the ϕ pin in this case will be as follows:

  $$\phi = \text{EXTAL} \times n$$

  Where: EXTAL: Crystal resonator or external clock frequency
  n: Division ratio (n = ϕ/2, ϕ/4, or ϕ/8)

- Do not set the DIV bit and bits SCK2 to SCK0 simultaneously. First set the DIV bit, then bits SCK2 to SCK0.

Bit 5 DIV	Description
0	When bits SCK2 to SCK0 are set to other than high-speed mode, medium-speed mode is set (Initial value)
1	When bits SCK2 to SCK0 are set to other than high-speed mode, a divided clock is supplied to the entire chip

Bits 4 and 3—Reserved: These bits cannot be modified and are always read as 0.

Bits 2 to 0—System Clock Select 2 to 0 (SCK2 to SCK0): When the DIV bit is cleared to 0, these bits select the medium-speed mode; when the DIV bit is set to 1, they select the division ratio of the clock supplied to the entire chip.

0	0	0	Bus master is in high-speed mode (Initial value)	Bus master is in high-speed mode (Initial value)
		1	Medium-speed clock is ϕ/2	Clock supplied to entire chip is ϕ/2
	1	0	Medium-speed clock is ϕ/4	Clock supplied to entire chip is ϕ/4
		1	Medium-speed clock is ϕ/8	Clock supplied to entire chip is ϕ/8
1	0	0	Medium-speed clock is ϕ/16	—
		1	Medium-speed clock is ϕ/32	—
	1	—	—	—

20.3 Oscillator

Clock pulses can be supplied by connecting a crystal resonator, or by input of an external clock.

20.3.1 Connecting a Crystal Resonator

Circuit Configuration: A crystal resonator can be connected as shown in the example in figure 20.2. Select the damping resistance R_d according to table 20.2. An AT-cut parallel-resonance crystal should be used.

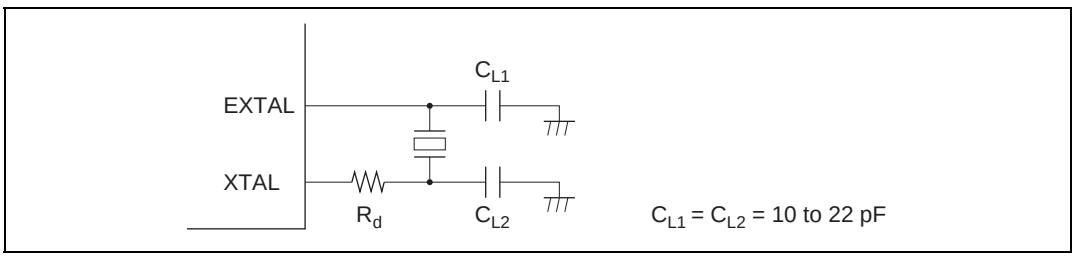


Figure 20.2 Connection of Crystal Resonator (Example)

Table 20.2 Damping Resistance Value

Frequency (MHz)	2	4	8	12	16	20	25
R_d (Ω)	6.8 k	500	200	0	0	0	0

system clock (ϕ).

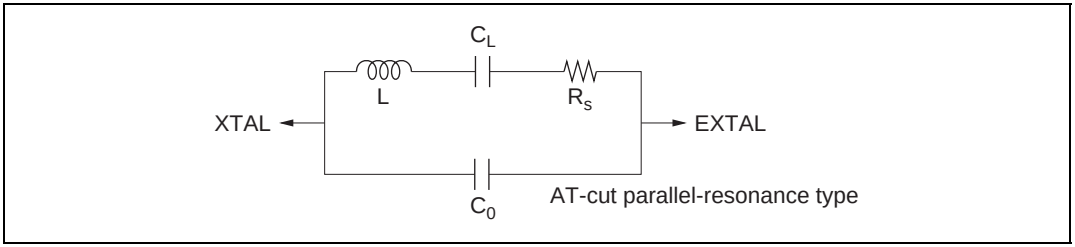


Figure 20.3 Crystal Resonator Equivalent Circuit

Table 20.3 Crystal Resonator Characteristics

Frequency (MHz)	2	4	8	12	16	20	25
R_S max (Ω)	500	120	80	60	50	40	40
C_0 max (pF)	7	7	7	7	7	7	7

Notes on Board Design: When a crystal resonator is connected, the following points should be noted:

Other signal lines should be routed away from the oscillator circuit to prevent induction from interfering with correct oscillation. See figure 20.4.

When designing the board, place the crystal resonator and its load capacitors as close as possible to the XTAL and EXTAL pins.

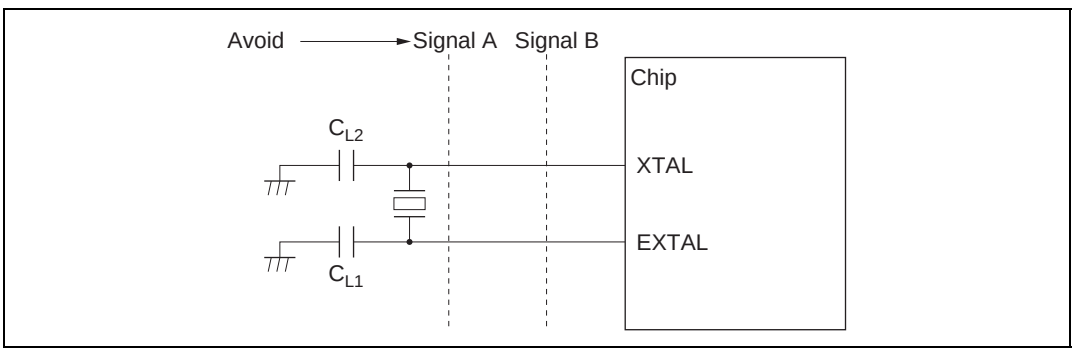


Figure 20.4 Example of Incorrect Board Design

Circuit Configuration: An external clock signal can be input as shown in the examples in figure 20.5. If the XTAL pin is left open, make sure that stray capacitance is no more than 10 pF.

In example (b), make sure that the external clock is held high in standby mode.

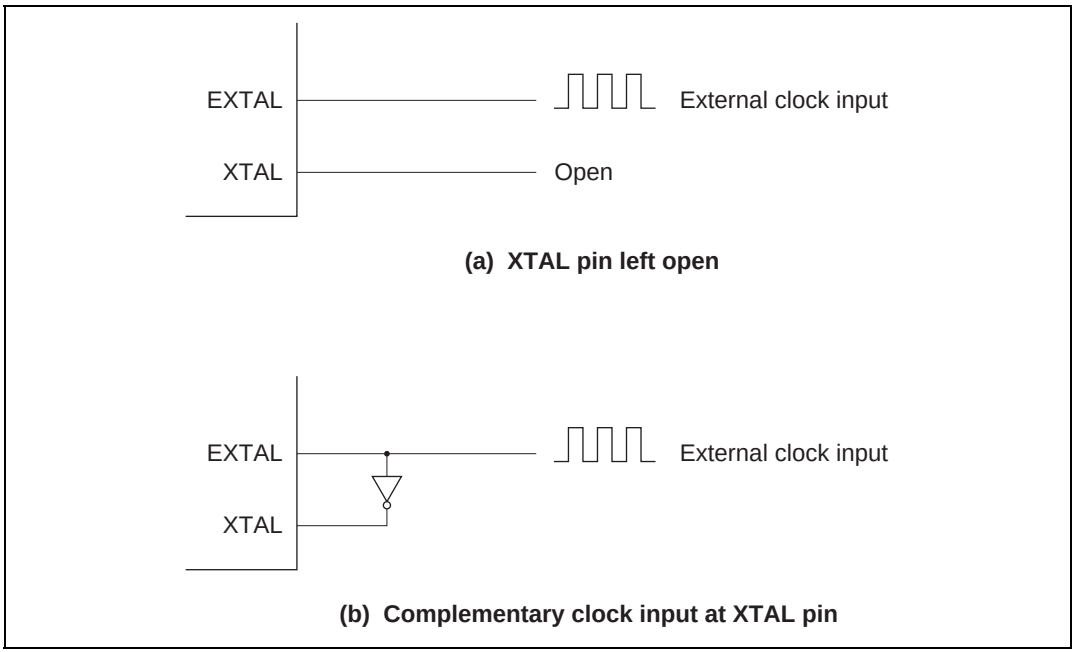


Figure 20.5 External Clock Input (Examples)

External Clock: The external clock signal should have the same frequency as the system clock (ϕ).

Table 20.4 and figure 20.6 show the input conditions for the external clock.

Item	Symbol	V_{CC} = 2.7 V to 3.6 V Min	Max	V_{CC} = 3.0 V to 3.6 V Min	Max	Unit	Test Conditions	
External clock input low pulse width	t_{EXL}	20	—	10	—	ns	Figure 20.6	
External clock input high pulse width	t_{EXH}	20	—	10	—	ns		
External clock rise time	t_{EXr}	—	5	—	5	ns		
External clock fall time	t_{EXf}	—	5	—	5	ns		
Clock low pulse width level	t_{CL}	0.4	0.6	0.4	0.6	t_{cyc}	$\phi \geq 5$ MHz	Figure 22.2
		80	—	80	—	ns	$\phi < 5$ MHz	
Clock high pulse width level	t_{CH}	0.4	0.6	0.4	0.6	t_{cyc}	$\phi \geq 5$ MHz	
		80	—	80	—	ns	$\phi < 5$ MHz	

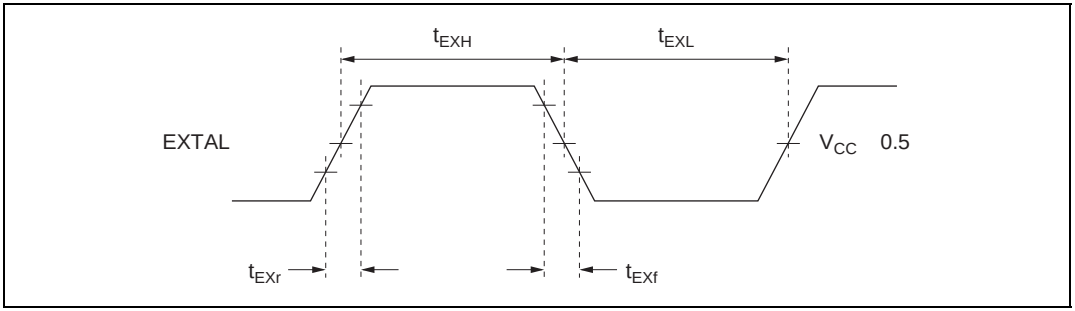


Figure 20.6 External Clock Input Timing

When the oscillator frequency is 5 MHz or higher, the duty adjustment circuit adjusts the duty cycle of the clock signal from the oscillator to generate the system clock (ϕ).

20.5 Medium-Speed Clock Divider

The medium-speed clock divider divides the system clock to generate ϕ/2, ϕ/4, ϕ/8, ϕ/16, and ϕ/32.

20.6 Bus Master Clock Selection Circuit

The bus master clock selection circuit selects the system clock (ϕ) or one of the medium-speed clocks (ϕ/2, ϕ/4, ϕ/8, ϕ/16, or ϕ/32) to be supplied to the bus master, according to the settings of the SCK2 to SCK0 bits in SCKCR.

21.1 Overview

In addition to the normal program execution state, the chip has five power-down modes in which operation of the CPU and oscillator is halted and power dissipation is reduced. Low-power operation can be achieved by individually controlling the CPU, on-chip supporting modules, and so on.

The chip 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

Of these, 2 to 6 are power-down modes. Sleep mode is a CPU mode, medium-speed mode is a CPU and bus master mode, and module stop mode is an on-chip supporting module mode (including bus masters other than the CPU). A combination of these modes can be set.

After a reset, the chip is in high-speed mode.

Table 21.1 shows the conditions for transition to the various modes, the status of the CPU, on-chip supporting modules, etc., and the method of clearing each mode.

Operating Mode	Transition Condition	Clearing Condition	Oscillator	CPU		Modules		I/O Ports
					Registers		Registers	
High speed mode	Control register		Functions	High speed	Function	High speed	Function	High speed
Medium-speed mode	Control register		Functions	Medium speed	Function	High/ medium speed *1	Function	High speed
Sleep mode	Instruction	Interrupt	Functions	Halted	Retained	High speed	Function	High speed
Module stop mode	Control register		Functions	High/ medium speed	Function	Halted	Retained/ reset *2	Retained
Software standby mode	Instruction	External interrupt	Halted	Halted	Retained	Halted	Retained/ reset *2	Retained
Hardware standby mode	Pin		Halted	Halted	Undefined	Halted	Reset	High impedance

Notes: 1. The bus master operates on the medium-speed clock, and other on-chip supporting modules on the high-speed clock.
2. Some SCI registers and the A/D converter are reset, and other on-chip supporting modules retain their states.

21.1.1 Register Configuration

Power-down modes are controlled by the SBYCR, SCKCR, and MSTPCR registers. Table 21.2 summarizes these registers.

Table 21.2 Power-Down Mode Registers

Name	Abbreviation	R/W	Initial Value	Address*
Standby control register	SBYCR	R/W	H'08	H'FF38
System clock control register	SCKCR	R/W	H'00	H'FF3A
Module stop control register H	MSTPCRH	R/W	H'3F	H'FF3C
Module stop control register L	MSTPCRL	R/W	H'FF	H'FF3D

Note: * Lower 16 bits of the address.

21.2.1 Standby Control Register (SBYCR)

Bit	:	7	6	5	4	3	2	1	0
		SSBY	STS2	STS1	STS0	OPE	—	—	IRQ37S
Initial value	:	0	0	0	0	1	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	—	—	R/W

SBYCR is an 8-bit readable/writable register that performs software standby mode control.

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

Bit 7—Software Standby (SSBY): Specifies a transition to software standby mode. Remains set to 1 when software standby mode is released by an external interrupt, and a transition is made to normal operation. The SSBY bit should be cleared by writing 0 to it.

Bit 7 SSBY	Description	
0	Transition to sleep mode after execution of SLEEP instruction	(Initial value)
1	Transition to software standby mode after execution of SLEEP instruction	

Bits 6 to 4—Standby Timer Select 2 to 0 (STS2 to STS0): These bits select the time the MCU waits for the clock to stabilize when software standby mode is cleared by an external interrupt. With crystal oscillation, refer to table 21.4 and make a selection according to the operating frequency so that the standby time is at least 8 ms (the oscillation stabilization time). With an external clock, any selection can be made*.

Note: * Except in the F-ZTAT versions.

0	0	0	Standby time = 8192 states	(Initial value)
		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*	

Note: * Not available in the F-ZTAT versions.

Bit 3—Output Port Enable (OPE): Specifies whether the output of the address bus and bus control signals ($\overline{CS0}$ to $\overline{CS7}$, $\overline{AS}$, $\overline{RD}$, $\overline{HWR}$, $\overline{LWR}$, $\overline{CAS}$) is retained or set to the high-impedance state in software standby mode.

Bit 3 OPE	Description
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 output state (Initial value)

Bits 2 and 1—Reserved: These bits cannot be modified and are always read as 0.

Bit 0—IRQ37 Software Standby Clear Select (IRQ37S): Specifies whether inputs $\overline{IRQ3}$ to $\overline{IRQ7}$ can be used as software standby mode clearing sources in addition to the usual sources, NMI and $\overline{IRQ0}$ to $\overline{IRQ2}$ inputs.

Bit 0 IRQ37S	Description
0	Inputs $\overline{IRQ3}$ to $\overline{IRQ7}$ cannot be used as software standby mode clearing sources (Initial value)
1	Inputs $\overline{IRQ3}$ to $\overline{IRQ7}$ can be used as software standby mode clearing sources

Bit	:	7	6	5	4	3	2	1	0
		PSTOP	—	DIV	—	—	SCK2	SCK1	SCK0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	—	—	R/W	R/W	R/W

SCKCR is an 8-bit readable/writable register that controls ϕ clock output, the medium-speed mode in which the bus master runs on a medium-speed clock and the other supporting modules run on the high-speed clock, and a function that allows the medium-speed mode to be disabled and the clock division ratio to be changed for the entire chip.

SCKCR is initialized to H'00 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bit 7—ϕ Clock Output Disable (PSTOP): Controls ϕ output.

	Description			
Bit 7 PSTOP	**Normal Operating Mode**	**Sleep Mode**	**Software Standby Mode**	**Hardware Standby Mode**
0	ϕ output (Initial value)	ϕ output	Fixed high	High impedance
1	Fixed high	Fixed high	Fixed high	High impedance

Bit 6—Reserved: This bit can be read or written to, but only 0 should be written.

Bit 5—Division Ratio Select (DIV): When the DIV bit is set to 1, the medium-speed mode is disabled and a clock obtained using the division ratio set with bits SCK2 to SCK0 is supplied to the entire chip. In this way, the current dissipation within the chip is reduced in proportion to the division ratio. As the frequency of ϕ changes, the following points must be noted.

- The division ratio set with bits SCK2 to SCK0 should be selected so as to fall within the guaranteed operation range of clock cycle time tcyc given in the AC timing table in the Electrical Characteristics section. Ensure that ϕ min = 2 MHz, and the condition ϕ < 2 MHz does not arise.

- All internal modules basically operate on ϕ. Note, therefore, that time processing involving the timers, the SCI, etc., will change when the division ratio changes. The wait time when software standby is cleared will also change in line with a change in the division ratio.

the φ pin in this case will be as follows:

$$\phi = EXTAL \times n$$

Where: EXTAL: Crystal resonator or external clock frequency
n: Division ratio (n = φ/2, φ/4, or φ/8)

- Do not set the DIV bit and bits SCK2 to SCK0 simultaneously. First set the DIV bit, then bits SCK2 to SCK0.

Bit 5 DIV	Description
0	When bits SCK2 to SCK0 are set to other than high-speed mode, medium-speed mode is set (Initial value)
1	When bits SCK2 to SCK0 are set to other than high-speed mode, a divided clock is supplied to the entire chip

Bits 4 and 3—Reserved: These bits cannot be modified and are always read as 0.

Bits 2 to 0—System Clock Select 2 to 0 (SCK2 to SCK0): When the DIV bit is cleared to 0, these bits select the bus master clock; when the DIV bit is set to 1, they select the division ratio of the clock supplied to the entire chip.

Bit 2 SCK2	Bit 1 SCK1	Bit 0 SCK0	Description	
			DIV = 0	DIV = 1
0	0	0	Bus master is in high-speed mode (Initial value)	Bus master is in high-speed mode (Initial value)
		1	Medium-speed clock is φ/2	Clock supplied to entire chip is φ/2
	1	0	Medium-speed clock is φ/4	Clock supplied to entire chip is φ/4
		1	Medium-speed clock is φ/8	Clock supplied to entire chip is φ/8
1	0	0	Medium-speed clock is φ/16	—
		1	Medium-speed clock is φ/32	—
	1	—	—	—

	MSTPCRH								MSTPCRL							
Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value :	0	0	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/W	R/W	R/W

MSTPCR is a 16-bit readable/writable register that performs module stop mode control.

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

Bits 15 to 0—Module Stop (MSTP15 to MSTP0): These bits specify module stop mode. See table 21.3 for the method of selecting on-chip supporting modules.

Bits 15 to 0 MSTP15 to MSTP0	**Description**
0	Module stop mode cleared
1	Module stop mode set

21.3 Medium-Speed Mode

When the SCK2 to SCK0 bits in SCKCR are set to 1, the operating mode changes to medium-speed mode as soon as the current bus cycle ends. In medium-speed mode, the CPU operates on the operating clock ($\phi/2$, $\phi/4$, $\phi/8$, $\phi/16$, or $\phi/32$) specified by the SCK2 to SCK0 bits. The bus masters other than the CPU (the DMAC and DTC) also operate in medium-speed mode. On-chip supporting modules other than the bus masters always operate on the high-speed clock (ϕ).

In medium-speed mode, a bus access is executed in the specified number of states with respect to the bus master operating clock. For example, if $\phi/4$ is selected as the operating clock, on-chip 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 is made to high-speed mode and medium-speed mode is cleared at the end of the current bus cycle.

If a SLEEP instruction is executed when the SSBY bit in SBYCR is cleared to 0, a transition is made to sleep mode. When sleep mode is cleared by an interrupt, medium-speed mode is restored.

medium-speed mode is restored.

When the $\overline{\text{RES}}$ pin is driven low, a transition is made to the reset state, and medium-speed mode is cleared. The same applies in the case of a reset caused by overflow of the watchdog timer.

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

Figure 21.1 shows the timing for transition to and clearance of medium-speed mode.

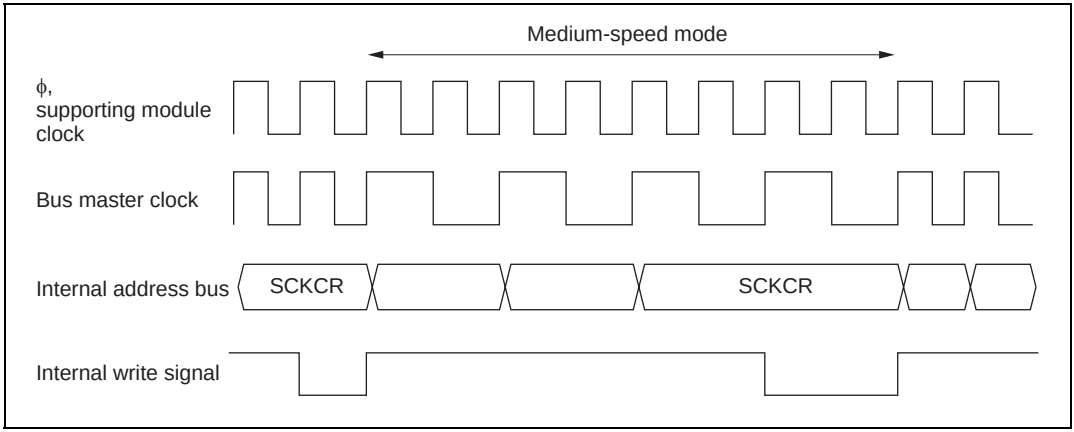


Figure 21.1 Medium-Speed Mode Transition and Clearance Timing

21.4 Sleep Mode

If a SLEEP instruction is executed when the SSBY bit in SBYCR is cleared to 0, the CPU enters sleep mode. In sleep mode, CPU operation stops but the contents of the CPU's internal registers are retained. Other supporting modules do not stop.

Sleep mode is cleared by a reset or any interrupt, and the CPU returns to the normal program execution state via the exception handling state. Sleep mode is not cleared if interrupts are disabled, or if interrupts other than NMI are masked by the CPU.

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

21.5.1 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 the end of the bus cycle and a transition is made to module stop mode. The CPU continues operating independently.

Table 21.3 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 the module starts operating at the end of the bus cycle. In module stop mode, the internal states of modules other than the SCI and A/D converter are retained.

After reset clearance, all modules other than DMAC and DTC are in module stop mode.

When an on-chip supporting module is in module stop mode, read/write access to its registers is disabled.

Do not make a transition to sleep mode with MSTPCR set to H'FFFF or H'EFFF, as this will halt operation of the bus controller.

Register	Bit	Module
MSTPCRH	MSTP15	DMA controller (DMAC)
	MSTP14	Data transfer controller (DTC)
	MSTP13	16-bit timer-pulse unit (TPU)
	MSTP12	8-bit timer module
	MSTP11	Programmable pulse generator (PPG)
	MSTP10	D/A converter (channels 0 and 1)
	MSTP9	A/D converter
	MSTP8	—
MSTPCRL	MSTP7	Serial communication interface (SCI) channel 2
	MSTP6	Serial communication interface (SCI) channel 1
	MSTP5	Serial communication interface (SCI) channel 0
	MSTP4	—
	MSTP3	—
	MSTP2	—
	MSTP1	—
	MSTP0	—

Note: Bits 8 and 4 to 0 can be read or written to, but do not affect operation.

21.5.2 Usage Notes

DMAC/DTC Module Stop: Depending on the operating status of the DMAC or DTC, the MSTP15 and MSTP14 bits may not be set to 1. Setting of the DMAC or DTC module stop mode should be carried out only when the respective module is not activated.

For details, refer to section 7, DMA Controller, and section 8, Data Transfer Controller.

On-Chip Supporting Module Interrupts: Relevant interrupt operations cannot be performed in module stop mode. Consequently, if module stop mode is entered when an interrupt has been requested, it will not be possible to clear the CPU interrupt source or the DMAC or DTC activation source. Interrupts should therefore be disabled before entering module stop mode.

Writing to MSTPCR: MSTPCR should only be written to by the CPU.

21.6.1 Software Standby Mode

If a SLEEP instruction is executed when the SSBY bit in SBYCR is set to 1, software standby mode is entered. In this mode, the CPU, on-chip supporting modules, and oscillator all stop. However, the contents of the CPU's internal registers, RAM data, and the states of on-chip supporting modules other than the SCI and A/D converter, and I/O ports, are retained. Whether the address bus and bus control signals are placed in the high-impedance state or retain the output state can be specified by the OPE bit in SBYCR. See appendix D, Pin States, for details.

In this mode the oscillator stops, and therefore power dissipation is significantly reduced.

21.6.2 Clearing Software Standby Mode

Software standby mode is cleared by an external interrupt (NMI pin, or pins $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ7}}$*), or by means of the $\overline{\text{RES}}$ pin or $\overline{\text{STBY}}$ pin.

Clearing with an Interrupt: When an NMI or IRQ0 to IRQ7* interrupt request signal is input, clock oscillation starts, and after the elapse of the time set in bits STS2 to STS0 in SYSCR, stable clocks are supplied to the entire chip, software standby mode is cleared, and interrupt exception handling is started.

When clearing software standby mode with an IRQ0 to IRQ7* interrupt, set the corresponding enable bit to 1 and ensure that no interrupt with a higher priority than interrupts IRQ0 to IRQ7* is generated. Software standby mode cannot be cleared if the interrupt has been masked on the CPU side or has been designated as a DTC activation source.

Note: * Setting the IRQ37S bit to 1 enables $\overline{\text{IRQ3}}$ to $\overline{\text{IRQ7}}$ to be used as software standby mode clearing sources.

Clearing with the $\overline{\text{RES}}$ Pin: When the $\overline{\text{RES}}$ pin is driven low, clock oscillation is started. At the same time as clock oscillation starts, clocks are supplied to the entire chip. Note that the $\overline{\text{RES}}$ pin must be held low until clock oscillation stabilizes. When the $\overline{\text{RES}}$ pin goes high, the CPU begins reset exception handling.

Clearing with the $\overline{\text{STBY}}$ Pin: When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

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 least 8 ms (the oscillation stabilization time).

Table 21.4 shows the standby times for different operating frequencies and settings of bits STS2 to STS0.

Table 21.4 Oscillation Stabilization Time Settings

STS2	STS1	STS0	Standby Time	25 MHz	20 MHz	16 MHz	12 MHz	10 MHz	8 MHz	6 MHz	4 MHz	2 MHz	Unit
0	0	0	8192 states	0.32	0.41	0.51	0.68	0.8	1.0	1.3	2.0	4.1	ms
		1	16384 states	0.65	0.82	1.0	1.3	1.6	2.0	2.7	4.1	[8.2]	
	1	0	32768 states	1.3	1.6	2.0	2.7	3.3	4.1	5.5	[8.2]	16.4	
		1	65536 states	2.6	3.3	4.1	5.5	6.6	[8.2]	[10.9]	16.4	32.8	
1	0	0	131072 states	5.2	6.6	[8.2]	[10.9]	[13.1]	16.4	21.8	32.8	65.5	
		1	262144 states	[10.4]	[13.1]	16.4	21.8	26.2	32.8	43.6	65.6	131.2	
	1	0	Reserved	—	—	—	—	—	—	—	—	—	—
		1	16 states	0.6	0.8	1.0	1.3	1.6	2.0	2.7	4.0	8.0	µs

[ ]: Recommended time setting

Using an External Clock: Any value can be set. Normally, use of the minimum time is recommended.*

Note: * The 16-state standby time cannot be used in the F-ZTAT versions; a standby time of 8192 states or longer should be used.

21.6.4 Software Standby Mode Application Example

Figure 21.2 shows an example in which a transition is made to software standby mode at the falling edge on the NMI pin, and software standby mode is cleared at the rising edge on the NMI pin.

In this example, an NMI interrupt is accepted with the NMIEG bit in SYSCR cleared to 0 (falling edge specification), then the NMIEG bit is set to 1 (rising edge specification), the SSBY bit is set to 1, and a SLEEP instruction is executed, causing a transition to software standby mode.

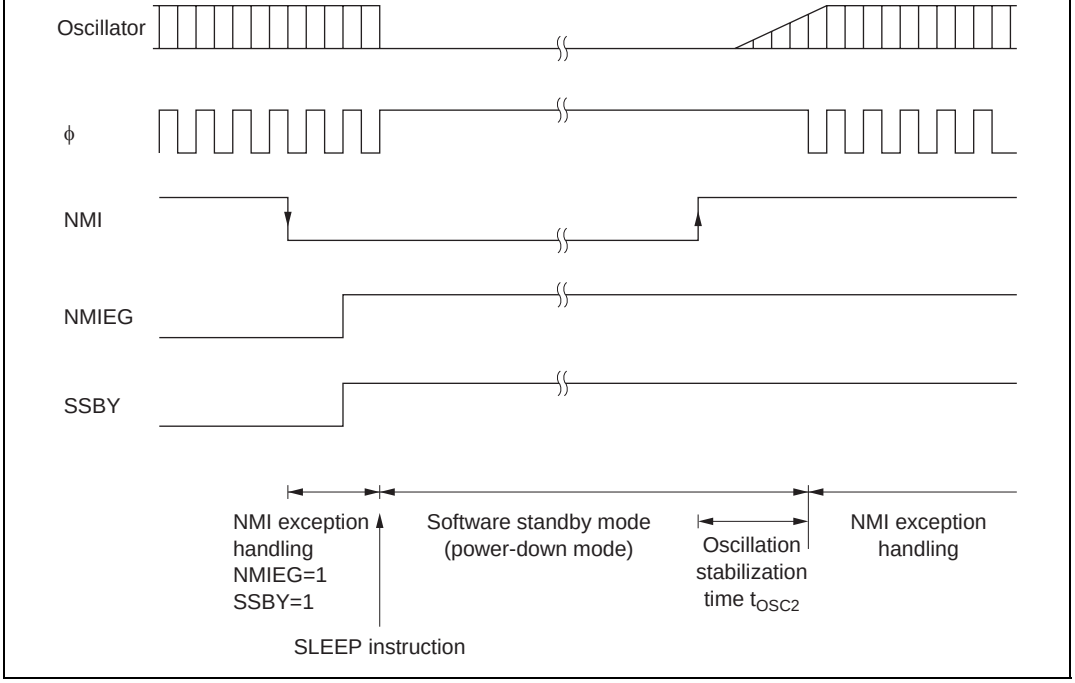


Figure 21.2 Software Standby Mode Application Example

21.6.5 Usage Notes

I/O Port Status: In software standby mode, I/O port states are retained. If the OPE bit is set to 1, the address bus and bus control signal output is also retained. Therefore, there is no reduction in current dissipation for the output current when a high-level signal is output.

Current Dissipation during Oscillation Stabilization Wait Period: Current dissipation increases during the oscillation stabilization wait period.

Write Data Buffer Function: The write data buffer function and software standby mode cannot be used at the same time. When the write data buffer function is used, the WDBE bit in BCRL should be cleared to 0 to cancel the write data buffer function before entering software standby mode. Also check that external writes have finished, by reading external addresses, etc., before executing a SLEEP instruction to enter software standby mode. See section 6.9, Write Data Buffer Function, for details of the write data buffer function.

21.7.1 Hardware Standby Mode

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode from any mode.

In hardware standby mode, all functions enter the reset state and stop operation, resulting in a significant reduction in power dissipation. As long as the prescribed voltage is supplied, on-chip 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 before driving the $\overline{\text{STBY}}$ pin low.

Do not change the state of the mode pins (MD2 to MD0) while the chip is in hardware standby mode.

Hardware standby mode is cleared by means of the $\overline{\text{STBY}}$ pin and the $\overline{\text{RES}}$ pin. When the $\overline{\text{STBY}}$ pin is driven high while the $\overline{\text{RES}}$ pin is low, the reset state is set and clock oscillation is started. Ensure that the $\overline{\text{RES}}$ pin is held low until the clock oscillator stabilizes (at least 8 ms—the oscillation stabilization time—when using a crystal oscillator). When the $\overline{\text{RES}}$ pin is subsequently driven high, a transition is made to the program execution state via the reset exception handling state.

21.7.2 Hardware Standby Mode Timing

Figure 21.3 shows an example of hardware standby mode timing.

When the $\overline{\text{STBY}}$ pin is driven low after the $\overline{\text{RES}}$ pin has been driven low, a transition is made to hardware standby mode. Hardware standby mode is cleared by driving the $\overline{\text{STBY}}$ pin high, waiting for the oscillation stabilization time, then changing the $\overline{\text{RES}}$ pin from low to high.

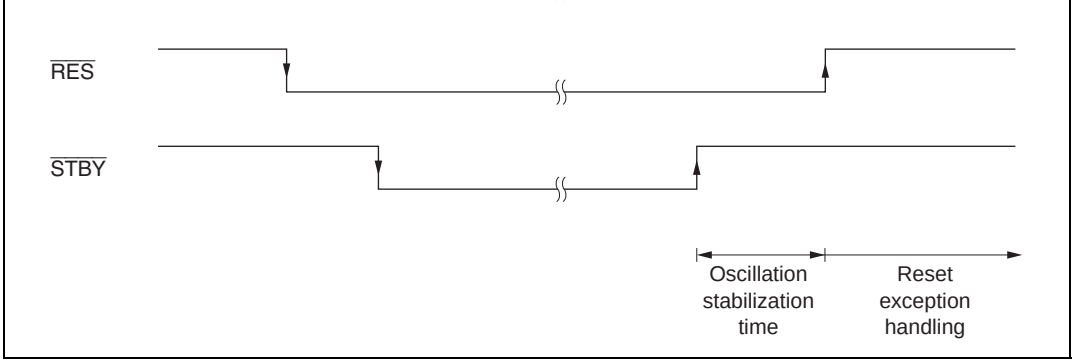


Figure 21.3 Hardware Standby Mode Timing

21.8 ϕ Clock Output Disabling Function

Output of the ϕ clock can be controlled by means of the PSTOP bit in SCKCR, and DDR for the corresponding port. When the PSTOP bit is set to 1, the ϕ clock stops at the end of the bus cycle, and ϕ output goes high. ϕ clock output is enabled when the PSTOP bit is cleared to 0. When DDR for the corresponding port is cleared to 0, ϕ clock output is disabled and input port mode is set. Table 21.5 shows the state of the ϕ pin in each processing state.

Table 21.5 ϕ Pin State in Each Processing State

DDR	0	1	1
PSTOP	—	0	1
Hardware standby mode	High impedance	High impedance	High impedance
Software standby mode	High impedance	Fixed high	Fixed high
Sleep mode	High impedance	ϕ output	Fixed high
Normal operating state	High impedance	ϕ output	Fixed high

22.1 Electrical Characteristics of Mask ROM Version (H8S/2338, H8S/2337) and ROMless Version (H8S/2332)

22.1.1 Absolute Maximum Ratings

Table 22.1 lists the absolute maximum ratings.

Table 22.1 Absolute Maximum Ratings

Item	Symbol	Value	Unit
Power supply voltage	V_{CC}	–0.3 to +4.6	V
Input voltage (except port 4, $P5_4$ to $P5_7$)	V_{in}	–0.3 to V_{CC} +0.3	V
Input voltage (port 4, $P5_4$ to $P5_7$)	V_{in}	–0.3 to AV_{CC} +0.3	V
Reference power supply voltage	V_{ref}	–0.3 to AV_{CC} +0.3	V
Analog power supply voltage	AV_{CC}	–0.3 to +4.6	V
Analog input voltage	V_{AN}	–0.3 to AV_{CC} +0.3	V
Operating temperature	T_{opr}	Regular specifications: –20 to +75	°C
		Wide-range specifications: –40 to +85	°C
Storage temperature	T_{stg}	–55 to +125	°C

Caution: Permanent damage to the chip may result if absolute maximum ratings are exceeded.

Table 22.2 DC Characteristics

Conditions: V_{CC} = 2.7 V to 3.6 V, AV_{CC} = 2.7 V to 3.6 V, V_{ref} = 2.7 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V*1, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Schmitt trigger input voltage	Ports 1, 2, 7, 9, $P5_0$ to $P5_3$, $P6_4$ to $P6_7$, PA_4 to PA_7	VT^-	$V_{CC} \times 0.2$	—	—	V	
		VT^+	—	—	$V_{CC} \times 0.7$	V	
		$VT^+ - VT^-$	$V_{CC} \times 0.07$	—	—	V	
Input high voltage	$\overline{RES}$, $\overline{STBY}$, NMI, MD_2 to MD_0	V_{IH}	$V_{CC} \times 0.9$	—	V_{CC} + 0.3	V	
	EXTAL		$V_{CC} \times 0.7$	—	V_{CC} + 0.3	V	
	Ports 3, 8, B to G, $P5_4$ to $P5_7$, $P6_0$ to $P6_3$, PA_0 to PA_3		2.2	—	V_{CC} + 0.3	V	
	Port 4		2.2	—	AV_{CC} + 0.3	V	
Input low voltage	$\overline{RES}$, $\overline{STBY}$, MD_2 to MD_0	V_{IL}	–0.3	—	$V_{CC} \times 0.1$	V	
	NMI, EXTAL, ports 3, 4, 8, B to G, $P5_4$ to $P5_7$, $P6_0$ to $P6_3$, PA_0 to PA_3		–0.3	—	$V_{CC} \times 0.2$	V	
Output high voltage	All output pins	V_{OH}	V_{CC} – 0.5	—	—	V	I_{OH} = –200 μA
			V_{CC} – 1.0	—	—	V	I_{OH} = –1 mA
Output low voltage	All output pins	V_{OL}	—	—	0.4	V	I_{OL} = 1.6 mA
Input leakage current	$\overline{RES}$	$\lvert I_{in} \rvert$	—	—	10	μA	V_{in} = 0.5 to V_{CC} – 0.5 V
	$\overline{STBY}$, NMI, MD_2 to MD_0		—	—	1	μA	
	Port 4, $P5_4$ to $P5_7$		—	—	1	μA	V_{in} = 0.5 to AV_{CC} – 0.5 V

Three-state leakage current (off state)	Ports 1 to 3, 5 to 9, A to G	$\lvert I_{TSI} \rvert$	—	—	1.0	µA	V_{in} = 0.5 to V_{CC} – 0.5 V
Input pull-up MOS current	Ports A to E	$-I_p$	10	—	300	µA	V_{in} = 0V
Input capacitance	$\overline{RES}$	C_{in}	—	—	30	pF	V_{in} = 0 V, f = 1 MHz, T_a = 25°C
	NMI		—	—	30	pF	
	All input pins except $\overline{RES}$ and NMI		—	—	15	pF	
Current dissipation*2	Normal operation	I_{CC}*4	—	43 (3.0 V)	84	mA	f = 20 MHz
				58 (3.3 V)	105	mA	f = 25 MHz
	Sleep mode		—	34 (3.0 V)	66	mA	f = 20 MHz
				4.6 (3.3 V)	82	mA	f = 25 MHz
	Standby mode*3		—	0.01	10	µA	$T_a \leq$ 50°C
			—	—	80		50°C < T_a
Analog power supply voltage	During A/D and D/A conversion	AI_{CC}	—	0.2 (3.0 V)	2.0	mA	
	Idle		—	0.01	5.0	µA	
Reference power supply voltage	During A/D and D/A conversion	AI_{CC}	—	2.4 (3.0 V)	6.0	mA	
	Idle		—	0.01	5.0	µA	
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Notes: 1. **If the A/D and D/A converters are not used, do not leave the AV_{CC}, V_{ref}, and AV_{SS} pins open.** Connect the AV_{CC} and V_{ref} pins to V_{CC}, and the AV_{SS} pin to V_{SS}.

2. Current dissipation values are for $V_{IH\ min} = V_{CC} - 0.5$ V and $V_{IL\ max} = 0.5$ V with all output pins unloaded and all MOS input pull-ups in the off state.
3. The values are for $V_{RAM} \leq V_{CC} < 2.7$ V, V_{IH} min = $V_{CC} \times 0.9$, and V_{IL} max = 0.3 V.
4. I_{CC} depends on V_{CC} and f as follows:

   I_{CC} max = 1.0 (mA) + 1.15 (mA/(MHz × V)) × V_{CC} × f (normal operation)

   I_{CC} max = 1.0 (mA) + 0.90 (mA/(MHz × V)) × V_{CC} × f (sleep mode)

Conditions: V_{CC} = 2.7 V to 3.6 V, AV_{CC} = 2.7 V to 3.6 V, V_{ref} = 2.7 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Item		Symbol	Min	Typ	Max	Unit
Permissible output low current (per pin)	All output pins	I_{OL}	—	—	2.0	mA
Permissible output low current (total)	Total of all output pins	ΣI_{OL}	—	—	80	mA
Permissible output high current (per pin)	All output pins	$-I_{OH}$	—	—	2.0	mA
Permissible output high current (total)	Total of all output pins	$\Sigma -I_{OH}$	—	—	40	mA

Note: To protect chip reliability, do not exceed the output current values in table 22.3.

22.1.3 AC Characteristics

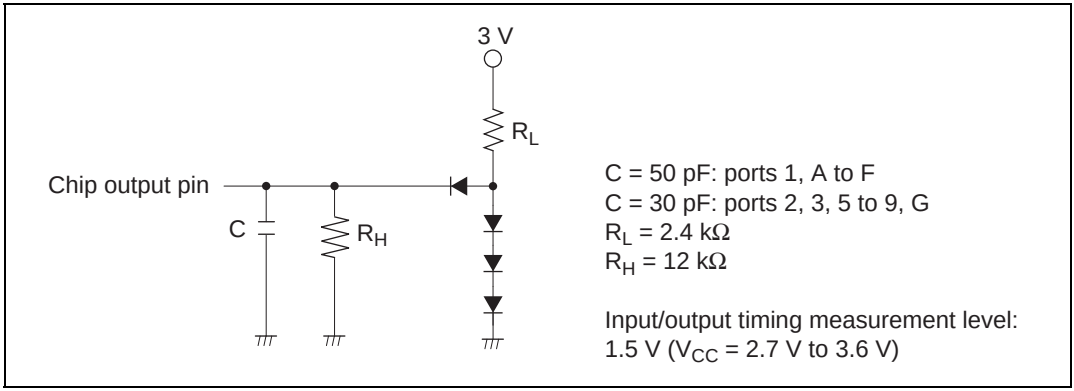


Figure 22.1 Output Load Circuit

Table 22.4 Clock Timing

Condition A: V_{CC} = 2.7 V to 3.6 V, AV_{CC} = 2.7 V to 3.6 V, V_{ref} = 2.7 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 20 MHz, T_a = –20°C to 75°C (regular specifications), T_a = –40°C to 85°C (wide-range specifications)

Condition B: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 25 MHz, T_a = –20°C to 75°C (regular specifications), T_a = –40°C to 85°C (wide-range specifications)

		Condition A		Condition B			
Item	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Unit**	**Test Conditions**
Clock cycle time	t_{cyc}	50	500	40	500	ns	Figure 22.2
Clock pulse high width	t_{CH}	20	—	15	—	ns	
Clock pulse low width	t_{CL}	20	—	15	—	ns	
Clock rise time	t_{Cr}	—	5	—	5	ns	
Clock fall time	t_{Cf}	—	5	—	5	ns	
Reset oscillation stabilization time (crystal)	t_{OSC1}	10	—	10	—	ms	Figure 22.3
Software standby oscillation stabilization time (crystal)	t_{OSC2}	10	—	10	—	ms	
External clock output stabilization delay time	t_{DEXT}	500	—	500	—	μs	Figure 22.3

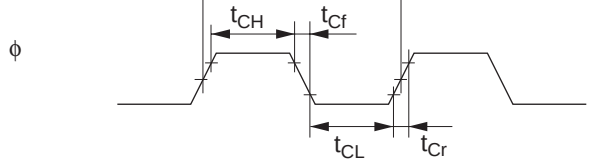


Figure 22.2 System Clock Timing

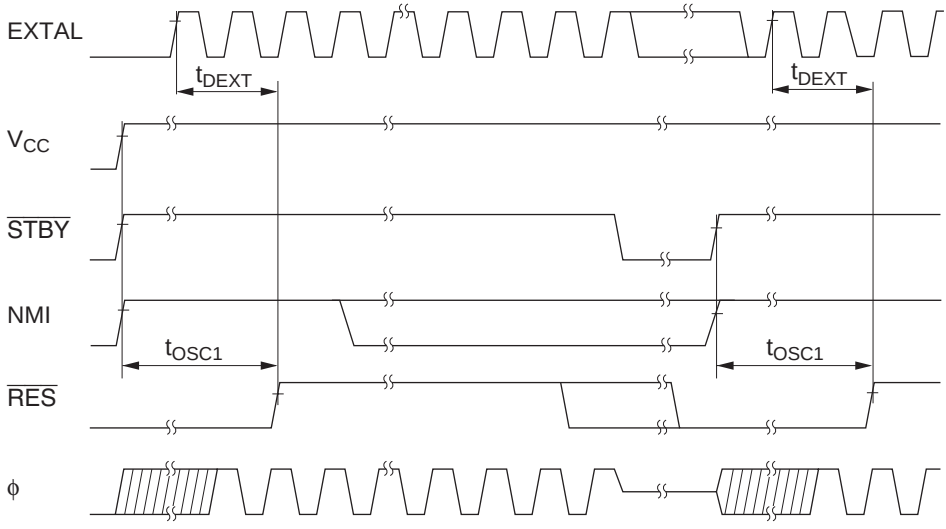


Figure 22.3 Oscillation Stabilization Timing

Table 22.5 Control Signal Timing

Condition A: V_{CC} = 2.7 V to 3.6 V, AV_{CC} = 2.7 V to 3.6 V, V_{ref} = 2.7 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 20 MHz, T_a = –20°C to 75°C (regular specifications), T_a = –40°C to 85°C (wide-range specifications)

Condition B: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 25 MHz, T_a = –20°C to 75°C (regular specifications), T_a = –40°C to 85°C (wide-range specifications)

		Condition A		Condition B			
Item	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Unit**	**Test Conditions**
$\overline{\text{RES}}$ setup time	t_{RESS}	200	—	200	—	ns	Figure 22.4
$\overline{\text{RES}}$ pulse width	t_{RESW}	20	—	20	—	t_{cyc}	
NMI setup time	t_{NMIS}	150	—	150	—	ns	Figure 22.5
NMI hold time	t_{NMIH}	10	—	10	—		
NMI pulse width (in recovery from software standby mode)	t_{NMIW}	200	—	200	—		
$\overline{\text{IRQ}}$ setup time	t_{IRQS}	150	—	150	—	ns	
$\overline{\text{IRQ}}$ hold time	t_{IRQH}	10	—	10	—		
$\overline{\text{IRQ}}$ pulse width (in recovery from software standby mode)	t_{IRQW}	200	—	200	—		

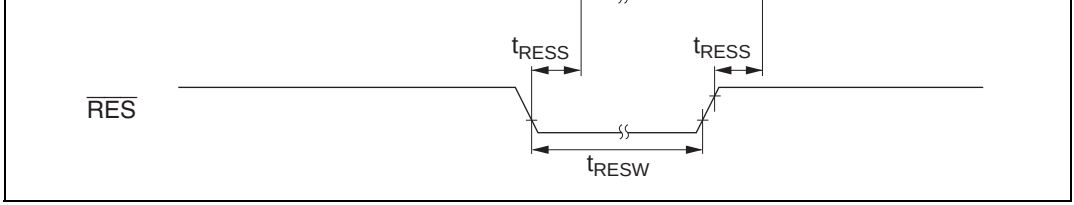


Figure 22.4 Reset Input Timing

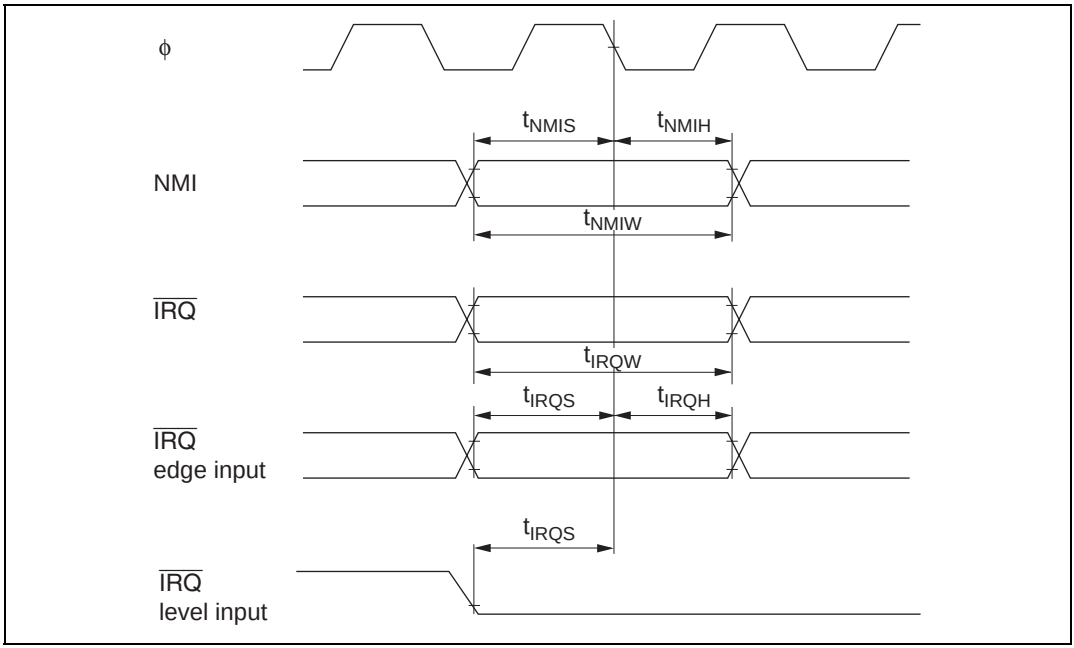


Figure 22.5 Interrupt Input Timing

Table 22.6 Bus Timing

Condition A: V_{CC} = 2.7 V to 3.6 V, AV_{CC} = 2.7 V to 3.6 V, V_{ref} = 2.7 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 20 MHz, T_a = –20°C to 75°C (regular specifications), T_a = –40°C to 85°C (wide-range specifications)

Condition B: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 25 MHz, T_a = –20°C to 75°C (regular specifications), T_a = –40°C to 85°C (wide-range specifications)

		Condition A		Condition B			
Item	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Unit**	**Test Conditions**
Address delay time	t_{AD}	—	20	—	20	ns	Figures 22.6 to 22.13
Address setup time	t_{AS}	0.5 × t_{cyc} – 15	—	0.5 × t_{cyc} – 15	—	ns	
Address hold time	t_{AH}	0.5 × t_{cyc} – 10	—	0.5 × t_{cyc} – 8	—	ns	
Precharge time	t_{PCH}	1.5 × t_{cyc}– 20	—	1.5 × t_{cyc}– 15	—	ns	
$\overline{CS}$ delay time 1	t_{CSD1}	—	20	—	15	ns	
$\overline{CS}$ delay time 2	t_{CSD2}	—	20	—	15	ns	
$\overline{CS}$ delay time 3	t_{CSD3}	—	25	—	20	ns	
$\overline{AS}$ delay time	t_{ASD}	—	20	—	15	ns	
$\overline{RD}$ delay time 1	t_{RSD1}	—	20	—	15	ns	
$\overline{RD}$ delay time 2	t_{RSD2}	—	20	—	15	ns	
$\overline{CAS}$ delay time	t_{CASD}	—	20	—	15	ns	
Read data setup time	t_{RDS}	15	—	15	—	ns	
Read data hold time	t_{RDH}	0	—	0	—	ns	
Read data access time 1	t_{ACC1}	—	1.0 × t_{cyc} – 25	—	1.0 × t_{cyc} – 20	ns	
Read data access time 2	t_{ACC2}	—	1.5 × t_{cyc} – 25	—	1.5 × t_{cyc} – 20	ns	
Read data access time 3	t_{ACC3}	—	2.0 × t_{cyc} – 25	—	2.0 × t_{cyc} – 20	ns	
Read data access time 4	t_{ACC4}	—	2.5 × t_{cyc} – 25	—	2.5 × t_{cyc} – 20	ns	
Read data access time 5	t_{ACC5}	—	3.0 × t_{cyc} – 25	—	3.0 × t_{cyc} – 20	ns	

Read data access time 6	t_{ACC6}	—	1.0 × t_{cyc} – 25	—	1.0 × t_{cyc} – 20	ns	Figures 22.6 to 22.13
$\overline{WR}$ delay time 1	t_{WRD1}	—	20	—	15	ns	
$\overline{WR}$ delay time 2	t_{WRD2}	—	20	—	15	ns	
$\overline{WR}$ pulse width 1	t_{WSW1}	1.0 × t_{cyc} – 20	—	1.0 × t_{cyc} – 15	—	ns	
$\overline{WR}$ pulse width 2	t_{WSW2}	1.5 × t_{cyc} – 20	—	1.5 × t_{cyc} – 15	—	ns	
Write data delay time	t_{WDD}	—	30	—	20	ns	
Write data setup time	t_{WDS}	0.5 × t_{cyc} – 20	—	0.5 × t_{cyc} – 15	—	ns	
Write data hold time	t_{WDH}	0.5 × t_{cyc} – 10	—	0.5 × t_{cyc} – 8	—	ns	
$\overline{WR}$ setup time	t_{WCS}	0.5 × t_{cyc} – 10	—	0.5 × t_{cyc} – 10	—	ns	
$\overline{WR}$ hold time	t_{WCH}	0.5 × t_{cyc} – 10	—	0.5 × t_{cyc} – 10	—	ns	
$\overline{CAS}$ setup time	t_{CSR}	0.5 × t_{cyc} – 10	—	0.5 × t_{cyc} – 8	—	ns	Figure 22.10
$\overline{WAIT}$ setup time	t_{WTS}	30	—	25	—	ns	Figure 22.8
$\overline{WAIT}$ hold time	t_{WTH}	5	—	5	—	ns	
$\overline{BREQ}$ setup time	t_{BRQS}	30	—	30	—	ns	Figure 22.14
$\overline{BACK}$ delay time	t_{BACD}	—	15	—	15	ns	
Bus floating time	t_{BZD}	—	50	—	40	ns	
$\overline{BREQO}$ delay time	t_{BRQOD}	—	30	—	25	ns	Figure 22.15

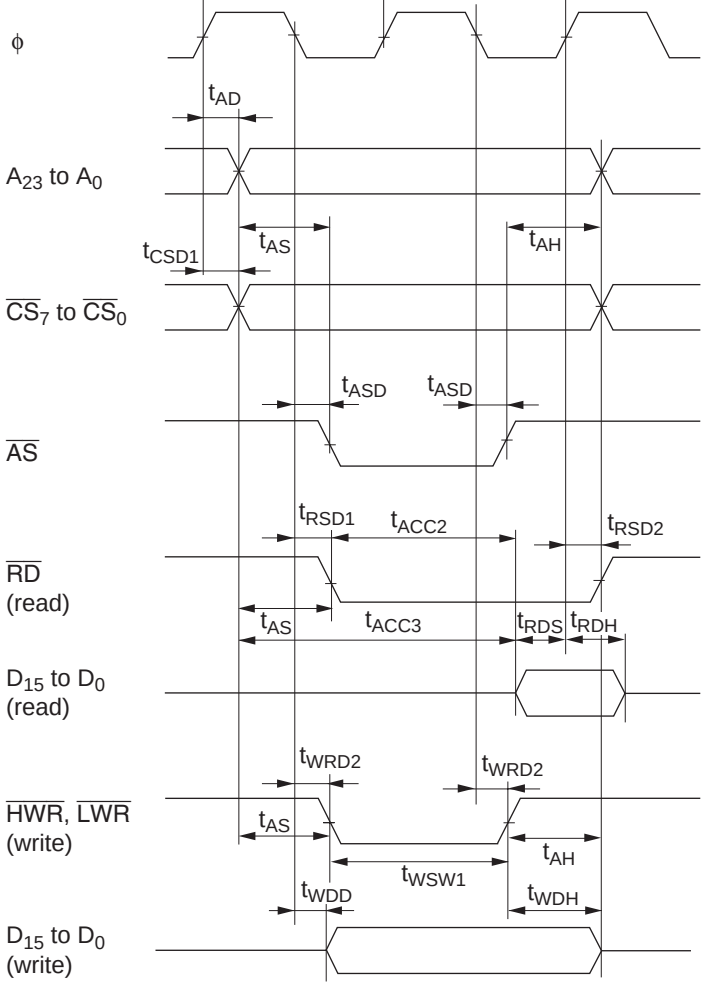


Figure 22.6 Basic Bus Timing (2-State Access)

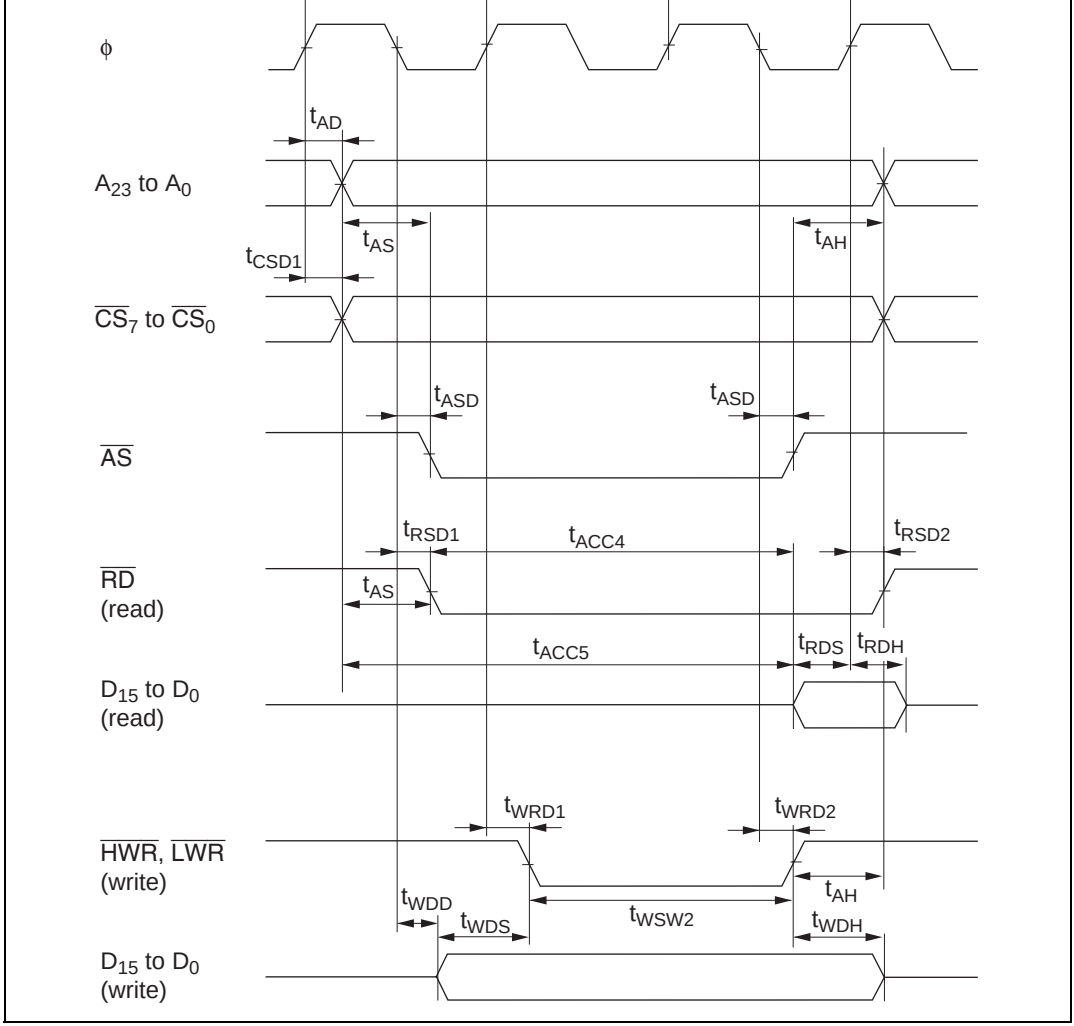


Figure 22.7 Basic Bus Timing (3-State Access)

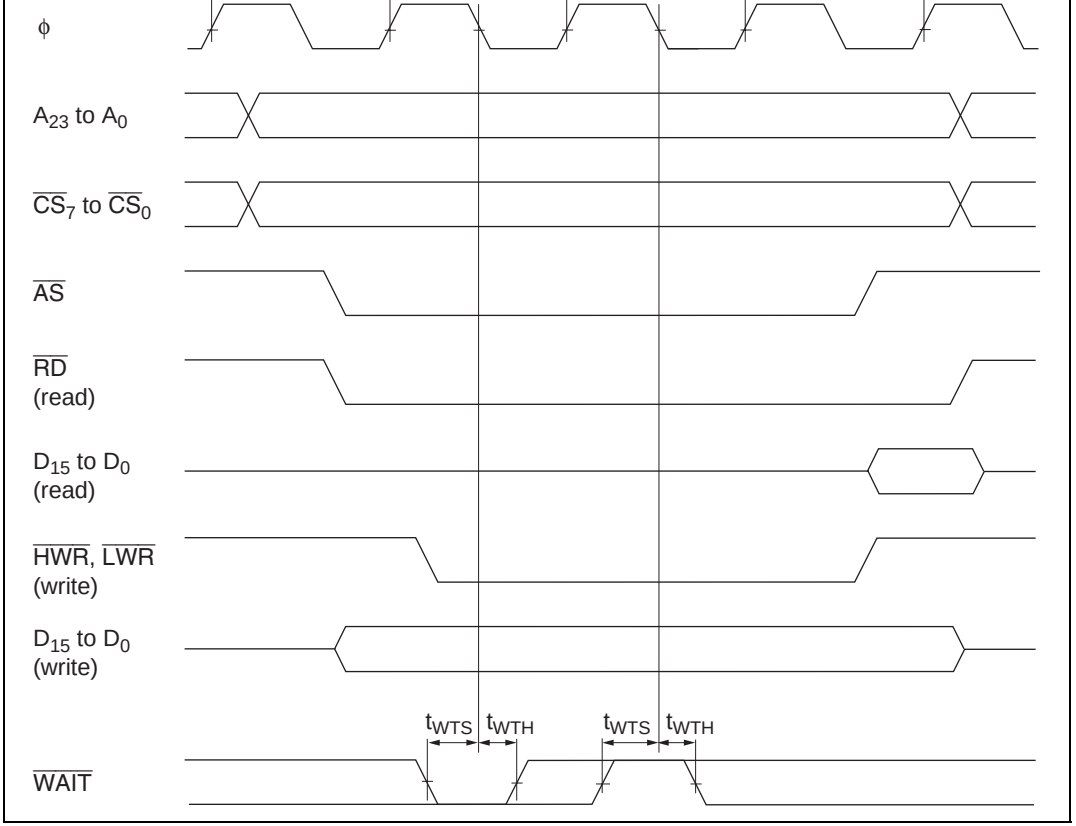


Figure 22.8 Basic Bus Timing (3-State Access, 1 Wait)

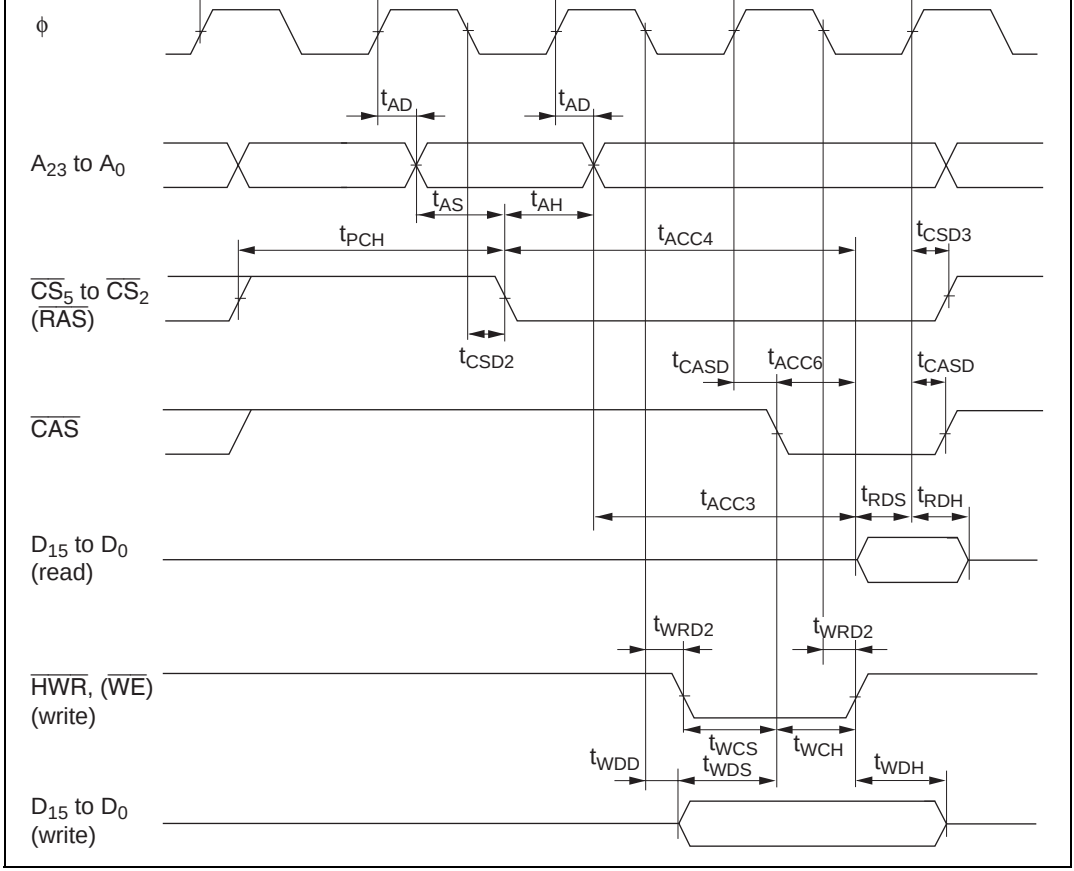


Figure 22.9 DRAM Bus Timing

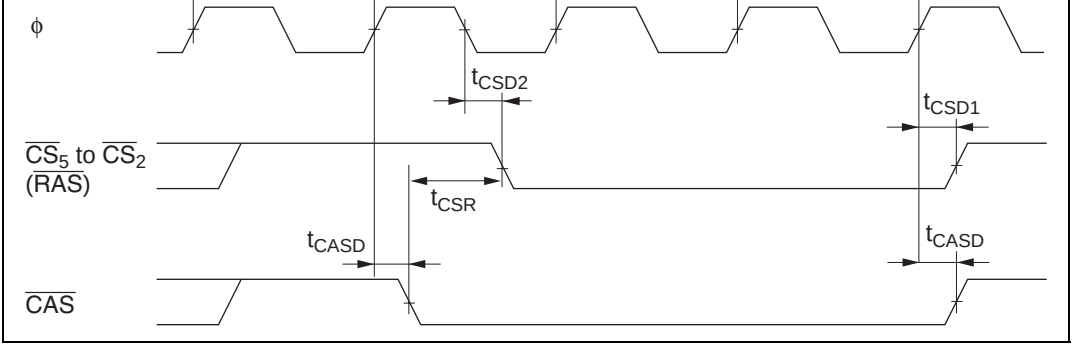


Figure 22.10 CAS-Before-RAS Refresh Timing

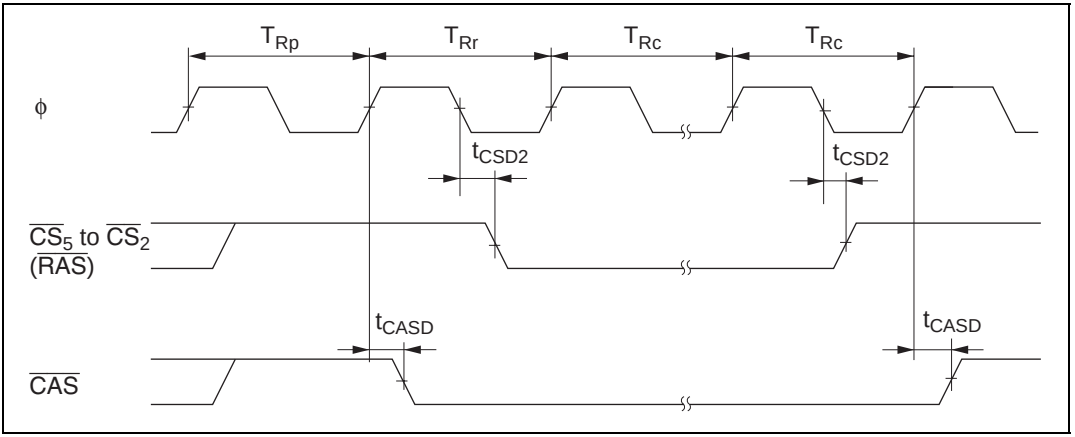


Figure 22.11 Self-Refresh Timing

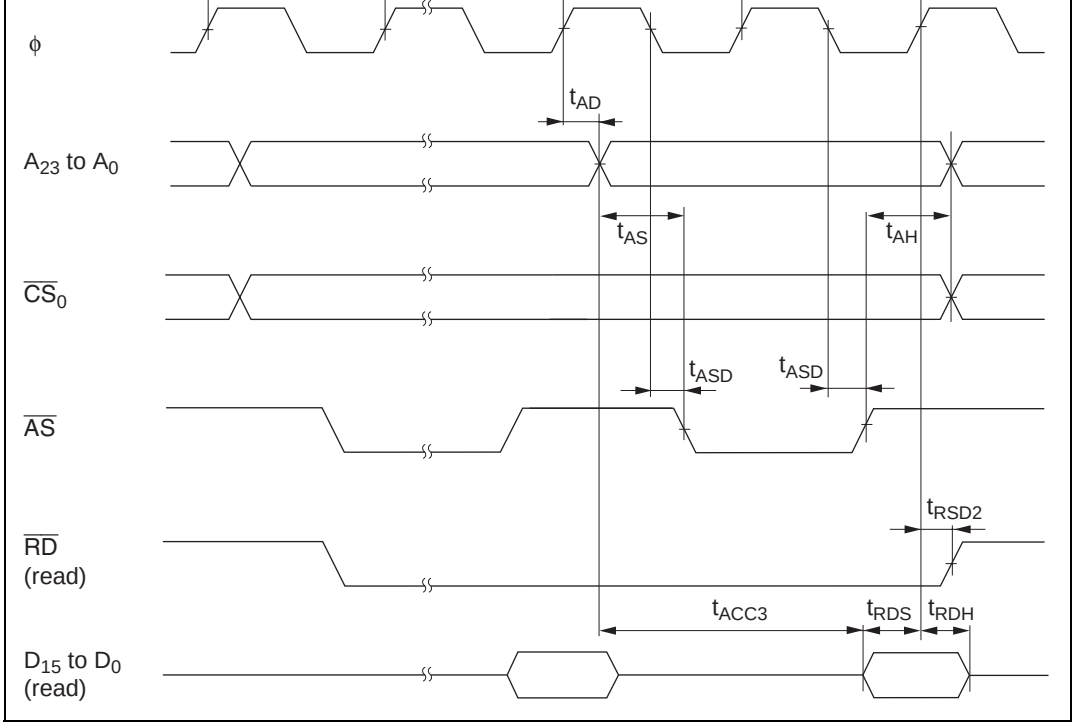


Figure 22.12 Burst ROM Access Timing (2-State Access)

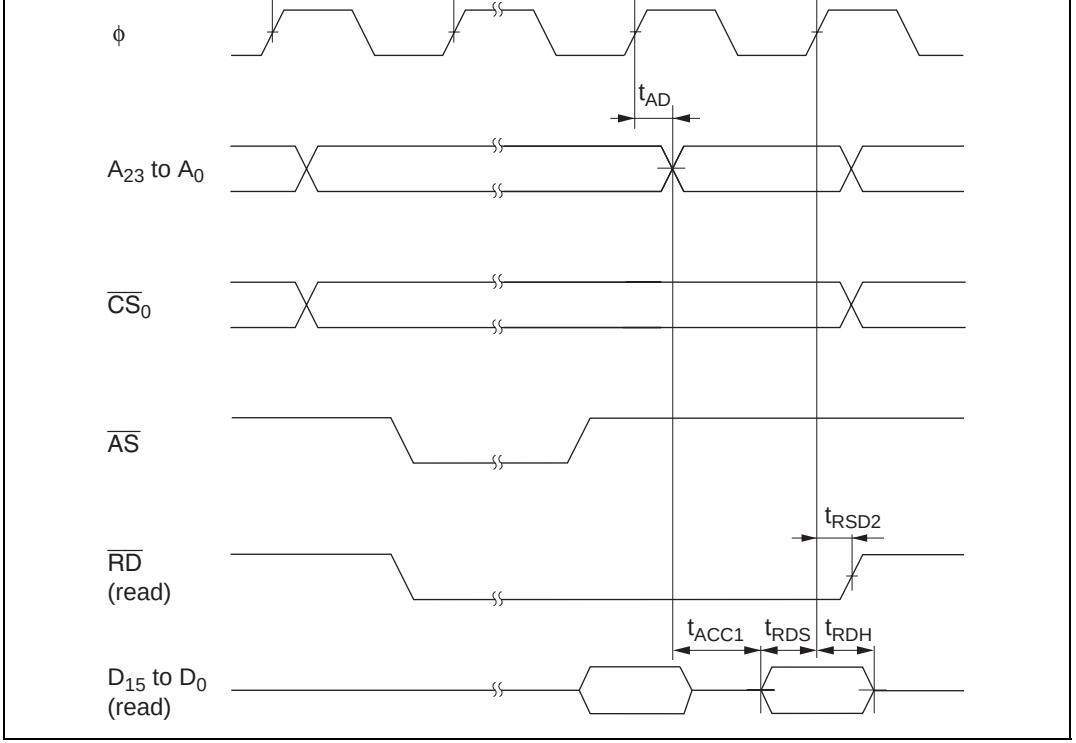


Figure 22.13 Burst ROM Access Timing (1-State Access)

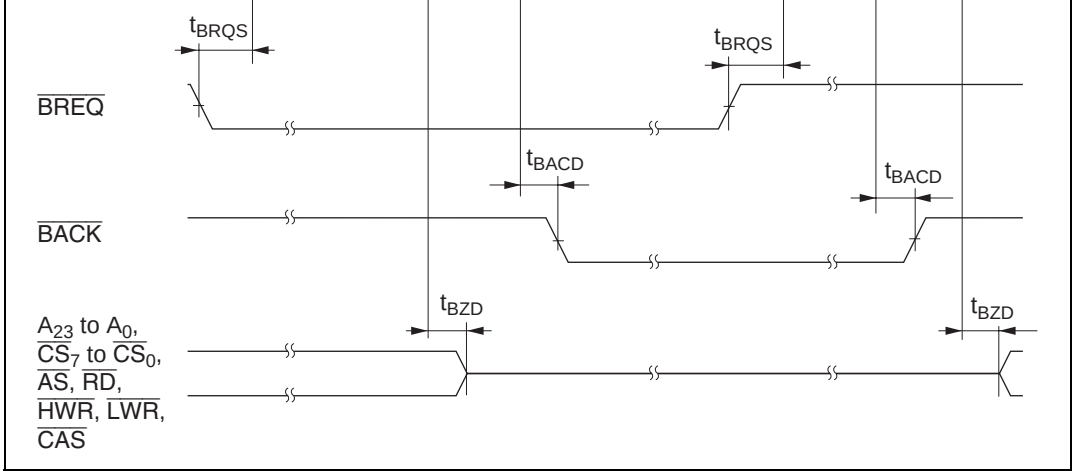


Figure 22.14 External Bus Release Timing

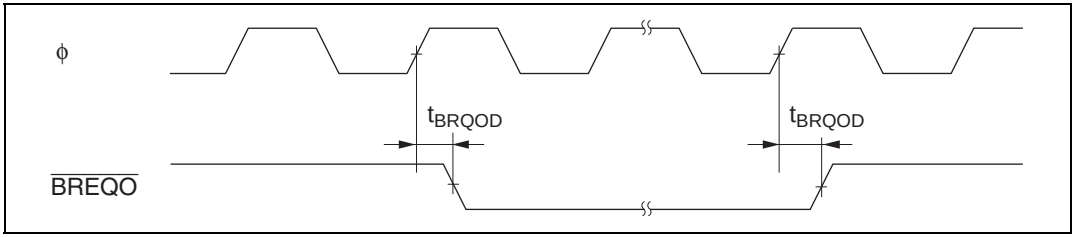


Figure 22.15 External Bus Request Output Timing

Table 22.7 DMAC Timing

Condition A: V_{CC} = 2.7 V to 3.6 V, AV_{CC} = 2.7 V to 3.6 V, V_{ref} = 2.7 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 20 MHz, T_a = –20°C to 75°C (regular specifications), T_a = –40°C to 85°C (wide-range specifications)

Condition B: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 25 MHz, T_a = –20°C to 75°C (regular specifications), T_a = –40°C to 85°C (wide-range specifications)

		Condition A		Condition B			
Item	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Unit**	**Test Conditions**
$\overline{\text{DREQ}}$ setup time	t_{DRQS}	30	—	25	—	ns	Figure 22.19
$\overline{\text{DREQ}}$ hold time	t_{DRQH}	10	—	10	—		
$\overline{\text{TEND}}$ delay time	t_{TED}	—	20	—	18		Figure 22.18
$\overline{\text{DACK}}$ delay time 1	t_{DACD1}	—	20	—	18	ns	Figures 22.16 and 22.17
$\overline{\text{DACK}}$ delay time 2	t_{DACD2}	—	20	—	18		

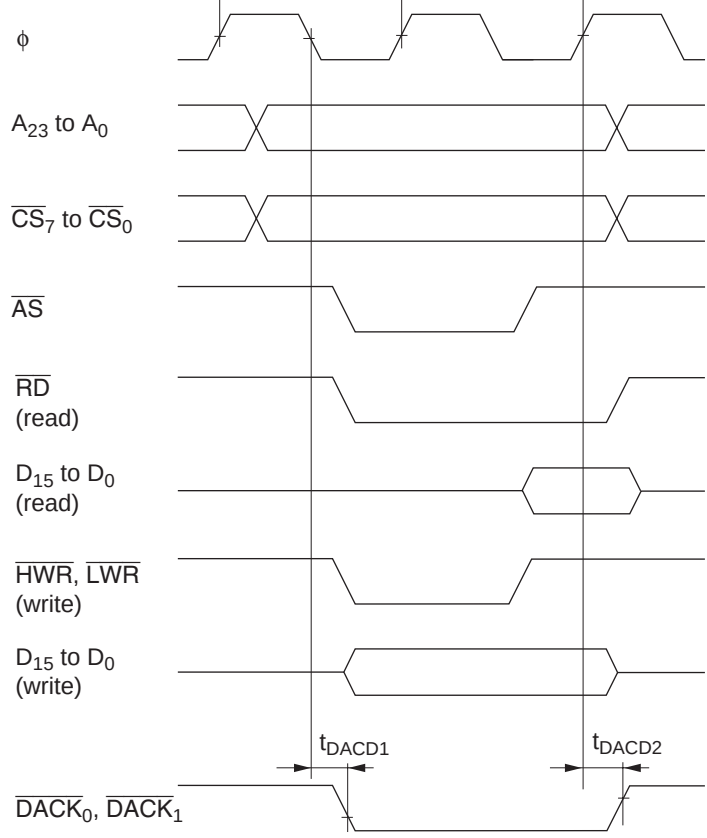


Figure 22.16 DMAC Single Address Transfer Timing (2-State Access)

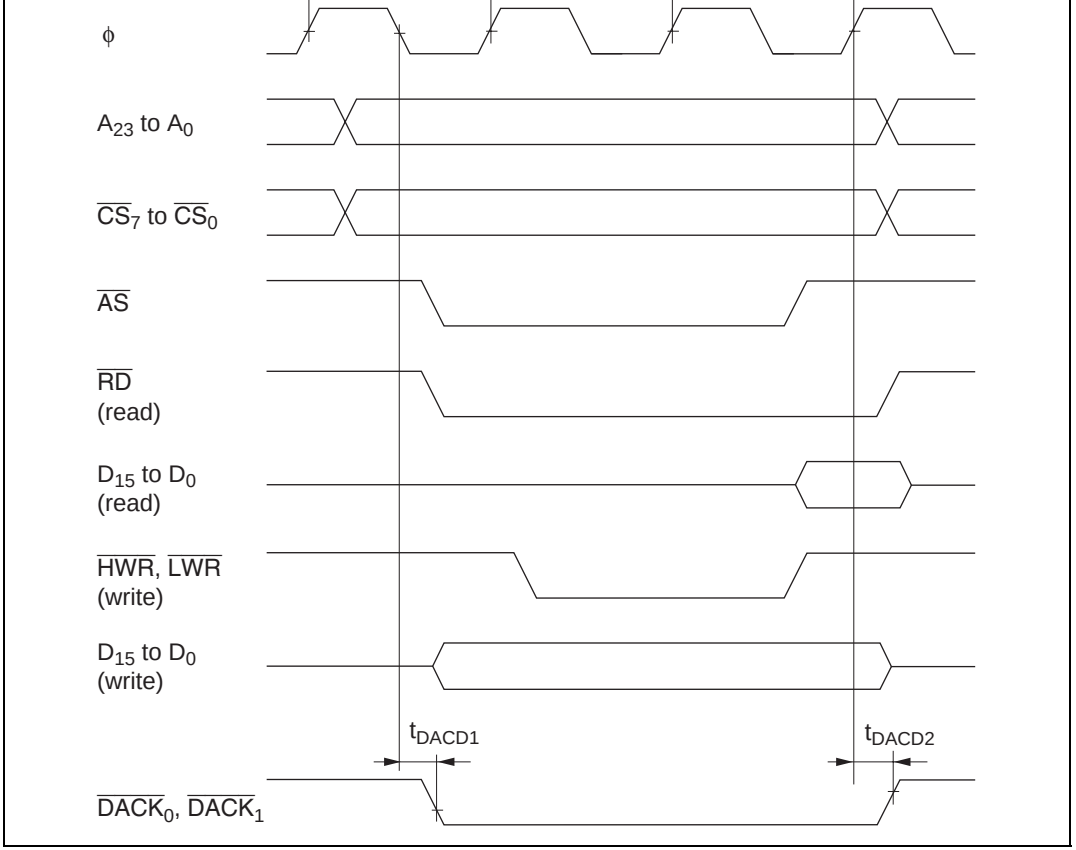


Figure 22.17 DMAC Single Address Transfer Timing (3-State Access)

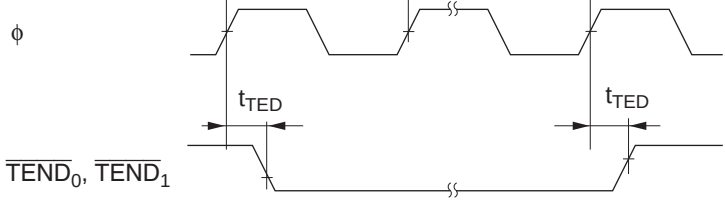


Figure 22.18 DMAC $\overline{\text{TEND}}$ Output Timing

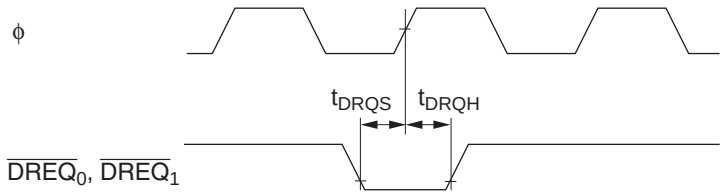


Figure 22.19 DMAC $\overline{\text{DREQ}}$ Input Timing

Table 22.8 Timing of On-Chip Supporting Modules

Condition A: V_{CC} = 2.7 V to 3.6 V, AV_{CC} = 2.7 V to 3.6 V, V_{ref} = 2.7 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 20 MHz, T_a = –20°C to 75°C (regular specifications), T_a = –40°C to 85°C (wide-range specifications)

Condition B: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 25 MHz, T_a = –20°C to 75°C (regular specifications), T_a = –40°C to 85°C (wide-range specifications)

				Condition A		Condition B			
Item			**Symbol**	**Min**	**Max**	**Min**	**Max**	**Unit**	**Test Conditions**
I/O ports	Output data delay time		t_{PWD}	—	50	—	40	ns	Figure 22.20
	Input data setup time		t_{PRS}	30	—	25	—		
	Input data hold time		t_{PRH}	30	—	25	—		
PPG	Pulse output delay time		t_{POD}	—	50	—	40	ns	Figure 22.21
TPU	Timer output delay time		t_{TOCD}	—	50	—	40	ns	Figure 22.22
	Timer input setup time		t_{TICS}	30	—	25	—		
	Timer clock input setup time		t_{TCKS}	30	—	25	—	ns	Figure 22.23
	Timer clock pulse width	Single-edge specification	t_{TCKWH}	1.5	—	1.5	—	t_{cyc}	
		Both-edge specification	t_{TCKWL}	2.5	—	2.5	—		
8-bit timer	Timer output delay time		t_{TMOD}	—	50	—	40	ns	Figure 22.24
	Timer reset input setup time		t_{TMRS}	30	—	25	—	ns	Figure 22.26
	Timer clock input setup time		t_{TMCS}	30	—	25	—	ns	Figure 22.25
	Timer clock pulse width	Single-edge specification	t_{TMCWH}	1.5	—	1.5	—	t_{cyc}	
		Both-edge specification	t_{TMCWL}	2.5	—	2.5	—		
WDT	Overflow output delay time		t_{WOVD}	—	50	—	40	ns	Figure 22.27

SCI	Input clock cycle	Asynchronous	t_{Scyc}	4	—	4	—	t_{cyc}	Figure 22.28
		Synchronous		6	—	6	—		
	Input clock pulse width		t_{SCKW}	0.4	0.6	0.4	0.6	t_{Scyc}	
	Input clock rise time		t_{SCKr}	—	1.5	—	1.5	t_{cyc}	
	Input clock fall time		t_{SCKf}	—	1.5	—	1.5		
	Transmit data delay time		t_{TXD}	—	50	—	40	ns	Figure 22.29
	Receive data setup time (synchronous)		t_{RXS}	50	—	40	—	ns	
	Receive data hold time (synchronous)		t_{RXH}	50	—	40	—	ns	
A/D converter	Trigger input setup time		t_{TRGS}	30	—	30	—	ns	Figure 22.30

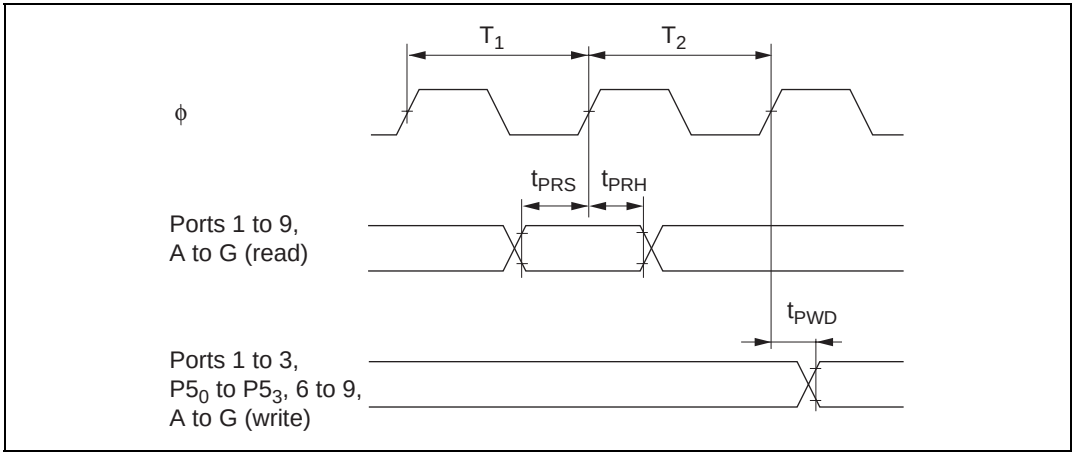


Figure 22.20 I/O Port Input/Output Timing

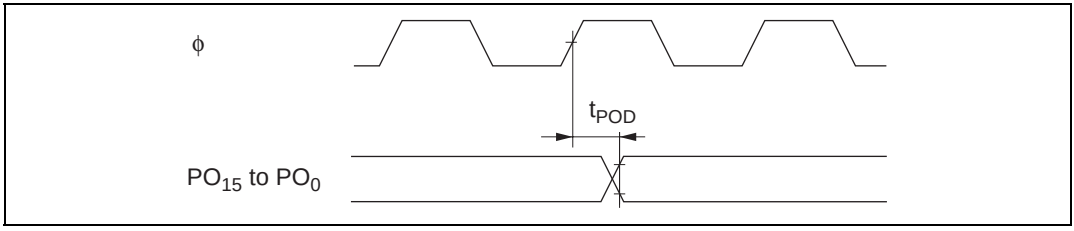


Figure 22.21 PPG Output Timing

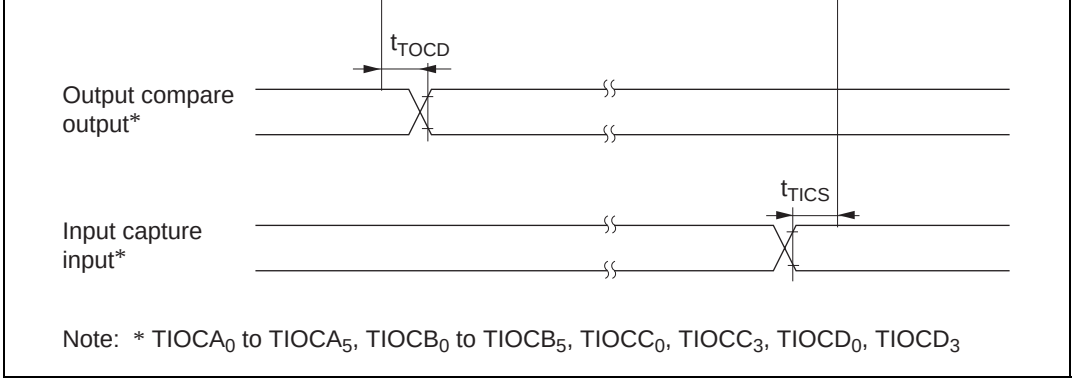


Figure 22.22 TPU Input/Output Timing

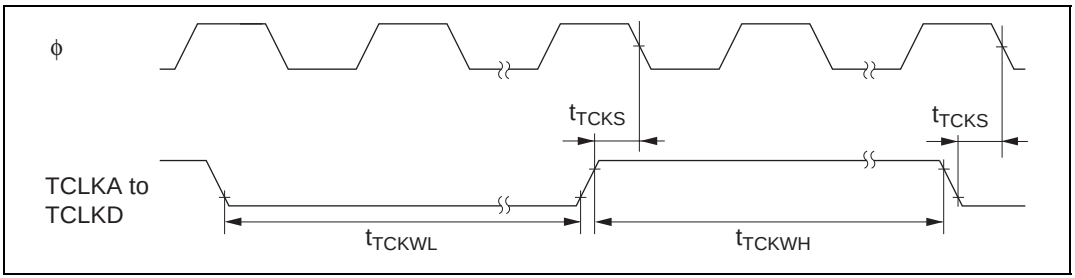


Figure 22.23 TPU Clock Input Timing

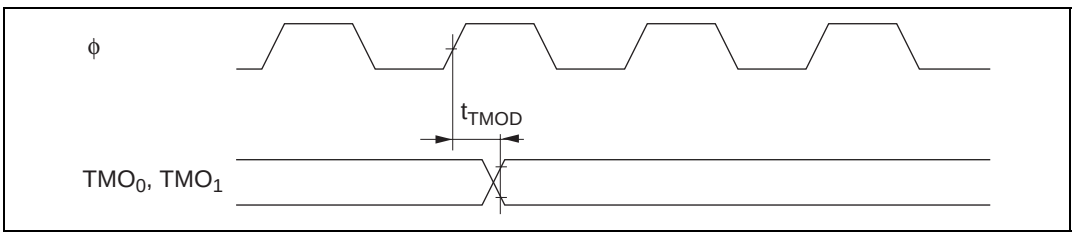


Figure 22.24 8-Bit Timer Output Timing

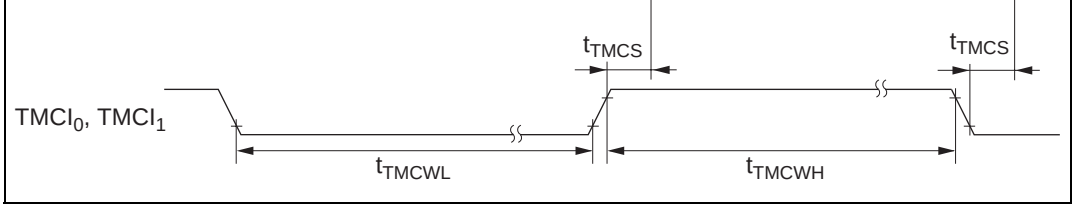


Figure 22.25 8-Bit Timer Clock Input Timing

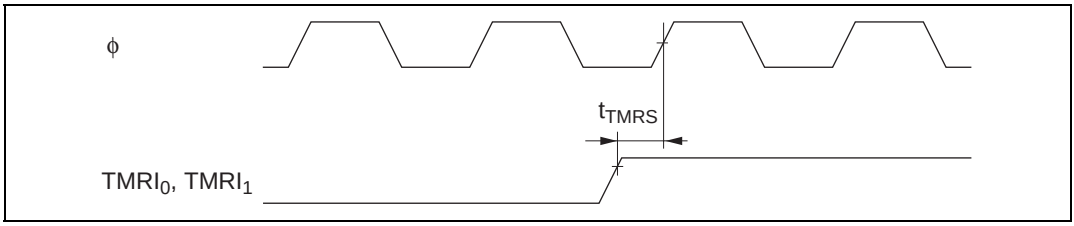


Figure 22.26 8-Bit Timer Reset Input Timing

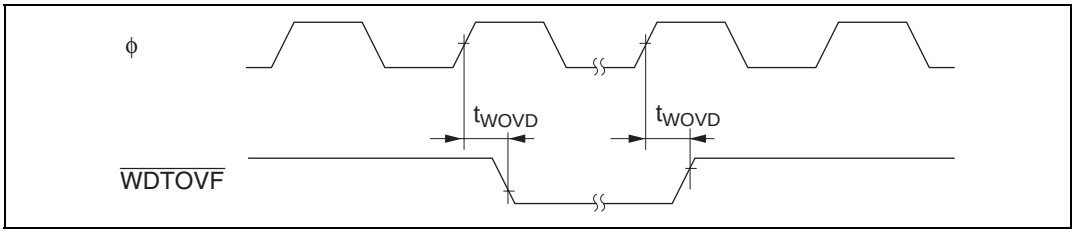


Figure 22.27 WDT Output Timing

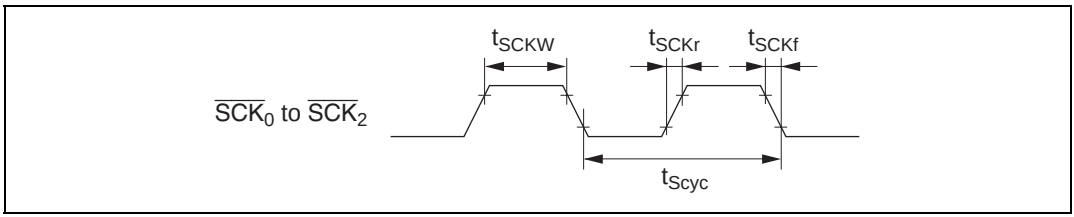


Figure 22.28 SCK Clock Input Timing

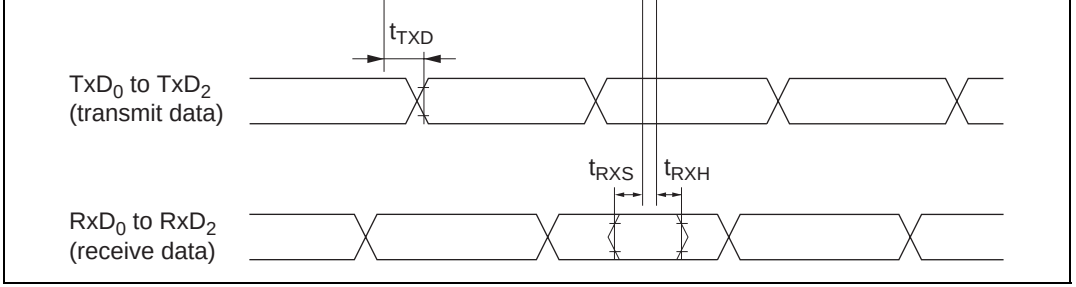


Figure 22.29 SCI Input/Output Timing (Synchronous Mode)

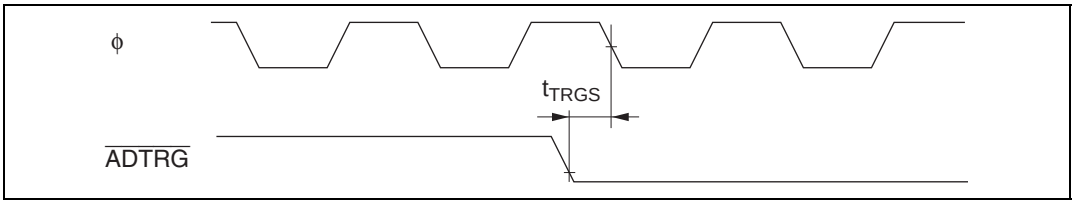


Figure 22.30 A/D Converter External Trigger Input Timing

Table 22.9 A/D Conversion Characteristics

Condition A: V_{CC} = 2.7 V to 3.6 V, AV_{CC} = 2.7 V to 3.6 V, V_{ref} = 2.7 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 20 MHz, T_a = –20°C to 75°C (regular specifications), T_a = –40°C to 85°C (wide-range specifications)

Condition B: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 25 MHz, T_a = –20°C to 75°C (regular specifications), T_a = –40°C to 85°C (wide-range specifications)

	Condition A			Condition B			
Item	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Unit**
Resolution	10	10	10	10	10	10	Bits
Conversion time	6.7	—	—	10.6	—	—	µs
Analog input capacitance	—	—	20	—	—	20	pF
Permissible signal source impedance	—	—	5	—	—	5	kΩ
Nonlinearity error	—	—	±5.5	—	—	±5.5	LSB
Offset error	—	—	±5.5	—	—	±5.5	LSB
Full-scale error	—	—	±5.5	—	—	±5.5	LSB
Quantization error	—	±0.5	—	—	±0.5	—	LSB
Absolute accuracy	—	—	±6.0	—	—	±6.0	LSB

Table 22.10 D/A Conversion Characteristics

Condition A: V_{CC} = 2.7 V to 3.6 V, AV_{CC} = 2.7 V to 3.6 V, V_{ref} = 2.7 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 20 MHz, T_a = –20°C to 75°C (regular specifications), T_a = –40°C to 85°C (wide-range specifications)

Condition B: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 25 MHz, T_a = –20°C to 75°C (regular specifications), T_a = –40°C to 85°C (wide-range specifications)

	Condition A			Condition B				
Item	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Unit**	**Test Conditions**
Resolution	8	8	8	8	8	8	Bits	
Conversion time	—	—	10	—	—	10	µs	20 pF-capacitive load
Absolute accuracy	—	±2.0	±3.0	—	±2.0	±3.0	LSB	2 MΩ resistive load
	—	—	±2.0	—	—	±2.0	LSB	4 MΩ resistive load

22.2.1 Absolute Maximum Ratings

Table 22.11 Absolute Maximum Ratings

Item	Symbol	Value	Unit
Power supply voltage	V_{CC}	–0.3 to +4.3	V
Input voltage (FWE*2, EMLE*3)	V_{in}	–0.3 to V_{CC} +0.3	V
Input voltage (except port 4, $P5_4$ to $P5_7$)	V_{in}	–0.3 to V_{CC} +0.3	V
Input voltage (port 4, $P5_4$ to $P5_7$)	V_{in}	–0.3 to AV_{CC} +0.3	V
Reference power supply voltage	V_{ref}	–0.3 to AV_{CC} +0.3	V
Analog power supply voltage	AV_{CC}	–0.3 to +4.3	V
Analog input voltage	V_{AN}	–0.3 to AV_{CC} +0.3	V
Operating temperature	T_{opr}	Regular specifications: –20 to +75*1	°C
		Wide-range specifications: –40 to +85*1	°C
Storage temperature	T_{stg}	–55 to +125	°C

Caution: Permanent damage to the chip may result if absolute maximum ratings are exceeded.

Notes: 1. The operating temperature ranges for flash memory programming/erasing are as follows:
T_a = 0°C to +75°C (regular specifications), T_a = 0°C to +85°C (wide-range specifications).
2. The FWE pin applies to the H8S/2338 F-ZTAT.
3. The EMLE pin applies to the H8S/2339 F-ZTAT.

Table 22.12 DC Characteristics

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V*1, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Schmitt trigger input voltage	Ports 1, 2, 7, 9, $P5_0$ to $P5_3$, $P6_4$ to $P6_7$, PA_4 to PA_7	VT^-	$V_{CC} \times 0.2$	—	—	V	
		VT^+	—	—	$V_{CC} \times 0.7$	V	
		$VT^+ - VT^-$	$V_{CC} \times 0.07$	—	—	V	
Input high voltage	$\overline{RES}$, $\overline{STBY}$, NMI, MD_2 to MD_0, FWE*2, EMLE*3	V_{IH}	$V_{CC} \times 0.9$	—	V_{CC} + 0.3	V	
	EXTAL		$V_{CC} \times 0.7$	—	V_{CC} + 0.3	V	
	Ports 3, 8, B to G, $P6_0$ to $P6_3$, PA_0 to PA_3		2.2	—	V_{CC} + 0.3	V	
	Port 4, $P5_4$ to $P5_7$		2.2	—	AV_{CC} + 0.3	V	
Input low voltage	$\overline{RES}$, $\overline{STBY}$, MD_2 to MD_0, FWE*2, EMLE*3	V_{IL}	–0.3	—	$V_{CC} \times 0.1$	V	
	NMI, EXTAL, ports 3, 4, 8, B to G, $P5_4$ to $P5_7$, $P6_0$ to $P6_3$, PA_0 to PA_3		–0.3	—	$V_{CC} \times 0.2$	V	
Output high voltage	All output pins	V_{OH}	V_{CC} – 0.5	—	—	V	I_{OH} = –200 μA
			V_{CC} – 1.0	—	—	V	I_{OH} = –1 mA
Output low voltage	All output pins	V_{OL}	—	—	0.4	V	I_{OL} = 1.6 mA
Input leakage current	$\overline{RES}$	$\lvert I_{in} \rvert$	—	—	10.0	μA	V_{in} = 0.5 to V_{CC} – 0.5 V
	$\overline{STBY}$, NMI, MD_2 to MD_0, FWE*2, EMLE*3		—	—	1.0	μA	
	Port 4, $P5_4$ to $P5_7$		—	—	1.0	μA	V_{in} = 0.5 to AV_{CC} – 0.5 V

Three-state leakage current (off state)	Ports 1 to 3, 5 to 9, A to G	$\lvert I_{TSI} \rvert$	—	—	1.0	µA	V_{in} = 0.5 to V_{CC} – 0.5 V
Input pull-up MOS current	Ports A to E	$-I_p$	10	—	300	µA	V_{CC} = 3.0 V to 3.6 V, V_{in} = 0 V
Input capacitance	$\overline{RES}$	C_{in}	—	—	30	pF	V_{in} = 0 V, f = 1 MHz, T_a = 25°C
	NMI		—	—	30	pF	
	All input pins except $\overline{RES}$ and NMI		—	—	15	pF	
Current dissipation*4	Normal operation	I_{CC}*6	—	58 (3.3 V)	105	mA	f = 25 MHz
	Sleep mode		—	46 (3.3 V)	82	mA	f = 25 MHz
	Standby mode*5		—	0.01	10	µA	$T_a \leq 50°C$
			—	—	80		50°C < T_a
Analog power supply voltage	During A/D and D/A conversion	AI_{CC}	—	0.2 (3.0 V)	2.0	mA	
	Idle		—	0.01	5.0	µA	
Reference power supply voltage	During A/D and D/A conversion	AI_{CC}	—	2.4 (3.0 V)	6.0	mA	
	Idle		—	0.01	5.0	µA	
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Notes: 1. **If the A/D and D/A converters are not used, do not leave the AV_{CC}, V_{ref}, and AV_{SS} pins open.** Connect the AV_{CC} and V_{ref} pins to V_{CC}, and the AV_{SS} pin to V_{SS}.

2. The FWE pin applies to the H8S/2338 F-ZTAT.
3. The EMLE pin applies to the H8S/2339 F-ZTAT.
4. Current dissipation values are for $V_{IH\ min} = V_{CC} – 0.5$ V and $V_{IL\ max} = 0.5$ V with all output pins unloaded and all MOS input pull-ups in the off state.
5. The values are for $V_{RAM} \leq V_{CC} < 3.0$ V, V_{IH} min = $V_{CC} \times 0.9$, and V_{IL} max = 0.3 V.
6. I_{CC} depends on V_{CC} and f as follows:
   I_{CC} max = 1.0 (mA) + 1.15 (mA/(MHz × V)) × V_{CC} × f (normal operation)
   I_{CC} max = 1.0 (mA) + 0.90 (mA/(MHz × V)) × V_{CC} × f (sleep mode)

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, $V_{SS} = AV_{SS}$ = 0 V, T_a = –20 to +75°C (regular specifications), T_a = –40 to +85°C (wide-range specifications)

Item		Symbol	Min	Typ	Max	Unit
Permissible output low current (per pin)	All output pins	I_{OL}	—	—	2.0	mA
Permissible output low current (total)	Total of all output pins	$\sum I_{OL}$	—	—	80	mA
Permissible output high current (per pin)	All output pins	$-I_{OH}$	—	—	2.0	mA
Permissible output high current (total)	Total of all output pins	$\sum -I_{OH}$	—	—	40	mA

Note: To protect chip reliability, do not exceed the output current values in table 22.13.

(1) Clock Timing

Table 22.14 Clock Timing

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 25 MHz, T_a = –20°C to 75°C (regular specifications), T_a = –40°C to 85°C (wide-range specifications)

Item	Symbol	Min	Max	Unit	Test Conditions
Clock cycle time	t_{cyc}	40	500	ns	Figure 22.2
Clock pulse high width	t_{CH}	15	—	ns	
Clock pulse low width	t_{CL}	15	—	ns	
Clock rise time	t_{Cr}	—	5	ns	
Clock fall time	t_{Cf}	—	5	ns	
Reset oscillation stabilization time (crystal)	t_{OSC1}	10	—	ms	Figure 22.3
Software standby oscillation stabilization time (crystal)	t_{OSC2}	10	—	ms	
External clock output stabilization delay time	t_{DEXT}	500	—	µs	Figure 22.3

Table 22.15 Control Signal Timing

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 25 MHz, T_a = –20°C to 75°C (regular specifications), T_a = –40°C to 85°C (wide-range specifications)

Item	Symbol	Min	Max	Unit	Test Conditions
$\overline{\text{RES}}$ setup time	t_{RESS}	200	—	ns	Figure 22.4
$\overline{\text{RES}}$ pulse width	t_{RESW}	20	—	t_{cyc}	
NMI setup time	t_{NMIS}	150	—	ns	Figure 22.5
NMI hold time	t_{NMIH}	10	—		
NMI pulse width (in recovery from software standby mode)	t_{NMIW}	200	—		
$\overline{\text{IRQ}}$ setup time	t_{IRQS}	150	—	ns	
$\overline{\text{IRQ}}$ hold time	t_{IRQH}	10	—		
$\overline{\text{IRQ}}$ pulse width (in recovery from software standby mode)	t_{IRQW}	200	—		

Table 22.16 Bus Timing

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 25 MHz, T_a = –20°C to 75°C (regular specifications), T_a = –40°C to 85°C (wide-range specifications)

Item	Symbol	Min	Max	Unit	Test Conditions
Address delay time	t_{AD}	—	20	ns	Figures 22.6 to 22.13
Address setup time	t_{AS}	$0.5 \times t_{cyc} - 15$	—	ns	
Address hold time	t_{AH}	$0.5 \times t_{cyc} - 8$	—	ns	
Precharge time	t_{PCH}	$1.5 \times t_{cyc} - 15$	—	ns	
$\overline{CS}$ delay time 1	t_{CSD1}	—	15	ns	
$\overline{CS}$ delay time 2	t_{CSD2}	—	15	ns	
$\overline{CS}$ delay time 3	t_{CSD3}	—	20	ns	
$\overline{AS}$ delay time	t_{ASD}	—	15	ns	
$\overline{RD}$ delay time 1	t_{RSD1}	—	15	ns	
$\overline{RD}$ delay time 2	t_{RSD2}	—	15	ns	
$\overline{CAS}$ delay time	t_{CASD}	—	15	ns	
Read data setup time	t_{RDS}	15	—	ns	
Read data hold time	t_{RDH}	0	—	ns	
Read data access time 1	t_{ACC1}	—	$1.0 \times t_{cyc} - 20$	ns	
Read data access time 2	t_{ACC2}	—	$1.5 \times t_{cyc} - 20$	ns	
Read data access time 3	t_{ACC3}	—	$2.0 \times t_{cyc} - 20$	ns	
Read data access time 4	t_{ACC4}	—	$2.5 \times t_{cyc} - 20$	ns	
Read data access time 5	t_{ACC5}	—	$3.0 \times t_{cyc} - 20$	ns	
Read data access time 6	t_{ACC6}	—	$1.0 \times t_{cyc} - 20$	ns	
$\overline{WR}$ delay time 1	t_{WRD1}	—	15	ns	
$\overline{WR}$ delay time 2	t_{WRD2}	—	15	ns	
$\overline{WR}$ pulse width 1	t_{WSW1}	$1.0 \times t_{cyc} - 15$	—	ns	
$\overline{WR}$ pulse width 2	t_{WSW2}	$1.5 \times t_{cyc} - 15$	—	ns	
Write data delay time	t_{WDD}	—	20	ns	
Write data setup time	t_{WDS}	$0.5 \times t_{cyc} - 15$	—	ns	
Write data hold time	t_{WDH}	$0.5 \times t_{cyc} - 8$	—	ns	
$\overline{WR}$ setup time	t_{WCS}	$0.5 \times t_{cyc} - 10$	—	ns	
$\overline{WR}$ hold time	t_{WCH}	$0.5 \times t_{cyc} - 10$	—	ns	
$\overline{CAS}$ setup time	t_{CSR}	$0.5 \times t_{cyc} - 8$	—	ns	Figure 22.10
$\overline{WAIT}$ setup time	t_{WTS}	25	—	ns	Figure 22.8
$\overline{WAIT}$ hold time	t_{WTH}	5	—	ns	

$\overline{\text{BACK}}$ delay time	t_{BACD}	—	15	ns	
Bus floating time	t_{BZD}	—	40	ns	
$\overline{\text{BREQO}}$ delay time	t_{BRQOD}	—	25	ns	Figure 22.15

(4) DMAC Timing

Table 22.17 DMAC Timing

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 2.7 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 25 MHz, T_a = –20°C to 75°C (regular specifications), T_a = –40°C to 85°C (wide-range specifications)

Item	Symbol	Min	Max	Unit	Test Conditions
$\overline{\text{DREQ}}$ setup time	t_{DRQS}	25	—	ns	Figure 22.19
$\overline{\text{DREQ}}$ hold time	t_{DRQH}	10	—		
$\overline{\text{TEND}}$ delay time	t_{TED}	—	18		Figure 22.18
$\overline{\text{DACK}}$ delay time 1	t_{DACD1}	—	18	ns	Figures 22.16 and 22.17
$\overline{\text{DACK}}$ delay time 2	t_{DACD2}	—	18		

Table 22.18 Timing of On-Chip Supporting Modules

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 25 MHz, T_a = –20°C to 75°C (regular specifications), T_a = –40°C to 85°C (wide-range specifications)

Item			Symbol	Min	Max	Unit	Test Conditions
I/O ports	Output data delay time		t_{PWD}	—	40	ns	Figure 22.20
	Input data setup time		t_{PRS}	25	—		
	Input data hold time		t_{PRH}	25	—		
PPG	Pulse output delay time		t_{POD}	—	40	ns	Figure 22.21
TPU	Timer output delay time		t_{TOCD}	—	40	ns	Figure 22.22
	Timer input setup time		t_{TICS}	25	—		
	Timer clock input setup time		t_{TCKS}	25	—	ns	Figure 22.23
	Timer clock pulse width	Single-edge specification	t_{TCKWH}	1.5	—	t_{cyc}	
		Both-edge specification	t_{TCKWL}	2.5	—		
8-bit timer	Timer output delay time		t_{TMOD}	—	40	ns	Figure 22.24
	Timer reset input setup time		t_{TMRS}	25	—	ns	Figure 22.26
	Timer clock input setup time		t_{TMCS}	25	—	ns	Figure 22.25
	Timer clock pulse width	Single-edge specification	t_{TMCWH}	1.5	—	t_{cyc}	
		Both-edge specification	t_{TMCWL}	2.5	—		
WDT	Overflow output delay time		t_{WOVD}	—	40	ns	Figure 22.27
SCI	Input clock cycle	Asynchronous	t_{Scyc}	4	—	t_{cyc}	Figure 22.28
		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}	—	40	ns	Figure 22.29
	Receive data setup time (synchronous)		t_{RXS}	40	—	ns	
	Receive data hold time (synchronous)		t_{RXH}	40	—	ns	
A/D converter	Trigger input setup time		t_{TRGS}	30	—	ns	Figure 22.30

Table 22.19 A/D Conversion Characteristics

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 25 MHz, T_a = –20°C to 75°C (regular specifications), T_a = –40°C to 85°C (wide-range specifications)

Item	Min	Typ	Max	Unit
Resolution	10	10	10	Bits
Conversion time	10.6	—	—	µs
Analog input capacitance	—	—	20	pF
Permissible signal source impedance	—	—	5	kΩ
Nonlinearity error	—	—	±5.5	LSB
Offset error	—	—	±5.5	LSB
Full-scale error	—	—	±5.5	LSB
Quantization error	—	—	±0.5	LSB
Absolute accuracy	—	—	±6.0	LSB

22.2.5 D/A Conversion Characteristics

Table 22.20 D/A Conversion Characteristics

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 25 MHz, T_a = –20°C to 75°C (regular specifications), T_a = –40°C to 85°C (wide-range specifications)

Item	Min	Typ	Max	Unit	Test Conditions
Resolution	8	8	8	Bits	
Conversion time	—	—	10	µs	20 pF-capacitive load
Absolute accuracy	—	±2.0	±3.0	LSB	2 MΩ resistive load
	—	—	±2.0	LSB	4 MΩ resistive load

Table 22.21 Flash Memory Characteristics

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, T_a = 0°C to + 75°C (program/erase operating temperature range: regular specifications), T_a = 0°C to + 85°C (program/erase operating temperature range: wide-range specifications)

Item		Symbol		Min	Typ	Max	Unit	Test Conditions
Programming time*1*2*4		t_P		—	10	200	ms/ 128 bytes	
Erase time*1*3*6		t_E		—	50	1000	ms/block	
Reprogramming count		N_{WEC}		100*7	10000*8	—	Times	
Data retention time*9		t_{DRP}		10	—	—	Years	
Programming	Wait time after SWE bit setting*1	x		1	—	—	µs	
	Wait time after PSU bit setting*1	y		50	—	—	µs	
	Wait time after P bit setting*1*4	z	(z1)	—	—	30	µs	$1 \le n \le 6$
			(z2)	—	—	200	µs	$7 \le n \le 1000$
			(z3)	—	—	10	µs	Wait for additional writing
	Wait time after P bit clearing*1	α		5	—	—	µs	
	Wait time after PSU bit clearing*1	β		5	—	—	µs	
	Wait time after PV bit setting*1	γ		4	—	—	µs	
	Wait time after H'FF dummy write*1	ε		2	—	—	µs	
	Wait time after PV bit clearing*1	η		2	—	—	µs	
	Wait time after SWE bit clearing*1	θ		100	—	—	µs	
	Maximum number of writes*1*4	N		—	—	1000*5	Times	
Erasing	Wait time after SWE bit setting*1	x		1	—	—	µs	
	Wait time after ESU bit setting*1	y		100	—	—	µs	
	Wait time after E bit setting*1*6	z		—	—	10	ms	Wait for erasing time
	Wait time after E bit clearing*1	α		10	—	—	µs	
	Wait time after ESU bit clearing*1	β		10	—	—	µs	
	Wait time after EV bit setting*1	γ		20	—	—	µs	
	Wait time after H'FF dummy write*1	ε		2	—	—	µs	
	Wait time after EV bit clearing*1	η		4	—	—	µs	
	Wait time after SWE bit clearing*1	θ		100	—	—	µs	
	Maximum number of erases*1*6	N		—	—	100	Times	

in flash memory control register 1 (FLMCR1). Does not include the program-verify time.)

3. Time to erase one block. (Indicates the time during which the E bit is set in FLMCR1. Does not include the erase-verify time.)
4. Maximum programming time

$$t_P(max) = \sum_{i=1}^{N} \text{wait time after P bit setting (z)}$$

5. The maximum number of writes (N) should be set as shown below according to the actual set value of z so as not to exceed the maximum programming time (t_P(max)).

   The wait time after P bit setting (z) should be changed as follows according to the number of writes (n).

   Number of writes (n)

   $1 \leq n \leq 6$ z = 30 µs

   $7 \leq n \leq 1000$ z = 200 µs

   $1 \leq n \leq 6$ z = 10 µs: For additional writing
6. For the maximum erase time (t_E(max)), the following relationship applies between the wait time after E bit setting (z) and the maximum number of erases (N):

   t_E (max) = Wait time after E bit setting (z) × maximum number of erases (N)
7. Minimum number of times for which all characteristics are guaranteed after rewriting (Guarantee range is 1 to minimum value).
8. Reference value for 25°C (as a guideline, rewriting should normally function up to this value).
9. Data retention characteristic when rewriting is performed within the specification range, including the minimum value.

22.3 Usage Note

Although both the F-ZTAT and mask ROM versions fully meet the electrical specifications listed in this manual, there may be differences in the actual values of the electrical characteristics, operating margins, noise margins, and so forth, due to differences in the fabrication process, the on-chip ROM, and the layout patterns.

If the F-ZTAT version is used to carry out system evaluation and testing, therefore, when switching to the mask ROM version the same evaluation and testing procedures should also be conducted on this version.

A.1 Instruction List

Operand Notation

Rd	General register (destination)*1
Rs	General register (source)*1
Rn	General register*1
ERn	General register (32-bit register)
MAC	Multiply-and-accumulate register (32-bit register)*2
(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 right, or 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

Notes: 1. General registers include 8-bit registers (R0H to R7H, R0L to R7L), 16-bit registers (R0 to R7, E0 to E7), and 32-bit registers (ER0 to ER7).

2. The MAC register cannot be used in the chip.

Symbol	
↕	Changes according to the result of the instruction
*	Undetermined (no guaranteed value)
0	Always cleared to 0
1	Always set to 1
—	Not affected by execution of the instruction

(1) Data Transfer Instructions

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	—		I	H	N	Z	V	C	Advanced
MOV	MOV.B #xx:8,Rd	B	2									#xx:8→Rd8	—	—	↕	↕	0	—	1
	MOV.B Rs,Rd	B		2								Rs8→Rd8	—	—	↕	↕	0	—	1
	MOV.B @ERs,Rd	B			2							@ERs→Rd8	—	—	↕	↕	0	—	2
	MOV.B @(d:16,ERs),Rd	B				4						@(d:16,ERs)→Rd8	—	—	↕	↕	0	—	3
	MOV.B @(d:32,ERs),Rd	B				8						@(d:32,ERs)→Rd8	—	—	↕	↕	0	—	5
	MOV.B @ERs+,Rd	B					2					@ERs→Rd8,ERs32+1→ERs32	—	—	↕	↕	0	—	3
	MOV.B @aa:8,Rd	B						2				@aa:8→Rd8	—	—	↕	↕	0	—	2
	MOV.B @aa:16,Rd	B						4				@aa:16→Rd8	—	—	↕	↕	0	—	3
	MOV.B @aa:32,Rd	B						6				@aa:32→Rd8	—	—	↕	↕	0	—	4
	MOV.B Rs,@ERd	B			2							Rs8→@ERd	—	—	↕	↕	0	—	2
	MOV.B Rs,@(d:16,ERd)	B				4						Rs8→@(d:16,ERd)	—	—	↕	↕	0	—	3
	MOV.B Rs,@(d:32,ERd)	B				8						Rs8→@(d:32,ERd)	—	—	↕	↕	0	—	5
	MOV.B Rs,@-ERd	B					2					ERd32-1→ERd32,Rs8→@ERd	—	—	↕	↕	0	—	3
	MOV.B Rs,@aa:8	B						2				Rs8→@aa:8	—	—	↕	↕	0	—	2
	MOV.B Rs,@aa:16	B						4				Rs8→@aa:16	—	—	↕	↕	0	—	3
	MOV.B Rs,@aa:32	B						6				Rs8→@aa:32	—	—	↕	↕	0	—	4
	MOV.W #xx:16,Rd	W	4									#xx:16→Rd16	—	—	↕	↕	0	—	2
	MOV.W Rs,Rd	W		2								Rs16→Rd16	—	—	↕	↕	0	—	1
	MOV.W @ERs,Rd	W			2							@ERs→Rd16	—	—	↕	↕	0	—	2

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)								Operation	Condition Code					No. of States
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa		I	H	N	Z	V	C	Advanced
MOV	MOV.W @(d:16,ERs),Rd	W				4					@(d:16,ERs)→Rd16	—	—	↕	↕	0	—	3
	MOV.W @(d:32,ERs),Rd	W				8					@(d:32,ERs)→Rd16	—	—	↕	↕	0	—	5
	MOV.W @ERs+,Rd	W					2				@ERs→Rd16,ERs32+2→ERs32	—	—	↕	↕	0	—	3
	MOV.W @aa:16,Rd	W						4			@aa:16→Rd16	—	—	↕	↕	0	—	3
	MOV.W @aa:32,Rd	W						6			@aa:32→Rd16	—	—	↕	↕	0	—	4
	MOV.W Rs,@ERd	W			2						Rs16→@ERd	—	—	↕	↕	0	—	2
	MOV.W Rs,@(d:16,ERd)	W				4					Rs16→@(d:16,ERd)	—	—	↕	↕	0	—	3
	MOV.W Rs,@(d:32,ERd)	W				8					Rs16→@(d:32,ERd)	—	—	↕	↕	0	—	5
	MOV.W Rs,@-ERd	W					2				ERd32-2→ERd32,Rs16→@ERd	—	—	↕	↕	0	—	3
	MOV.W Rs,@aa:16	W						4			Rs16→@aa:16	—	—	↕	↕	0	—	3
	MOV.W Rs,@aa:32	W						6			Rs16→@aa:32	—	—	↕	↕	0	—	4
	MOV.L #xx:32,ERd	L	6								#xx:32→ERd32	—	—	↕	↕	0	—	3
	MOV.L ERs,ERd	L		2							ERs32→ERd32	—	—	↕	↕	0	—	1
	MOV.L @ERs,ERd	L			4						@ERs→ERd32	—	—	↕	↕	0	—	4
	MOV.L @(d:16,ERs),ERd	L				6					@(d:16,ERs)→ERd32	—	—	↕	↕	0	—	5
	MOV.L @(d:32,ERs),ERd	L				10					@(d:32,ERs)→ERd32	—	—	↕	↕	0	—	7
	MOV.L @ERs+,ERd	L					4				@ERs→ERd32,ERs32+4→@ERs32	—	—	↕	↕	0	—	5
	MOV.L @aa:16,ERd	L						6			@aa:16→ERd32	—	—	↕	↕	0	—	5
	MOV.L @aa:32,ERd	L						8			@aa:32→ERd32	—	—	↕	↕	0	—	6

Mnemonic	Operand Size	Addressing Mode/ Instruction Length (Bytes)								Operation	Condition Code						No. of States
		#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa		I	H	N	Z	V	C	Advanced
MOV	MOV.L ERs,@ERd	L		4						ERs32→@ERd	—	—	↕	↕	0	—	4
	MOV.L ERs,@(d:16,ERd)	L			6					ERs32→@(d:16,ERd)	—	—	↕	↕	0	—	5
	MOV.L ERs,@(d:32,ERd)	L			10					ERs32→@(d:32,ERd)	—	—	↕	↕	0	—	7
	MOV.L ERs,@-ERd	L				4				ERd32-4→ERd32,ERs32→@ERd	—	—	↕	↕	0	—	5
	MOV.L ERs,@aa:16	L					6			ERs32→@aa:16	—	—	↕	↕	0	—	5
	MOV.L ERs,@aa:32	L					8			ERs32→@aa:32	—	—	↕	↕	0	—	6
POP	POP.W Rn	W							2	@SP→Rn16,SP+2→SP	—	—	↕	↕	0	—	3
	POP.L ERn	L							4	@SP→ERn32,SP+4→SP	—	—	↕	↕	0	—	5
PUSH	PUSH.W Rn	W							2	SP-2→SP,Rn16→@SP	—	—	↕	↕	0	—	3
	PUSH.L ERn	L							4	SP-4→SP,ERn32→@SP	—	—	↕	↕	0	—	5
LDM	LDM @SP+,(ERm-ERn)	L							4	(@SP→ERn32,SP+4→SP) Repeated for each register restored	—	—	—	—	—	—	7/9/11/13
STM	STM (ERm-ERn),@-SP	L							4	(SP-4→SP,ERn32→@SP) Repeated for each register saved	—	—	—	—	—	—	7/9/11/13
MOVFPE	MOVFPE @aa:16,Rd	Cannot be used in the chip															[2]
MOVTPE	MOVTPE Rs,@aa:16	Cannot be used in the chip															[2]

(2) Arithmetic Instructions

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of St
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Advan
ADD	ADD.B #xx:8,Rd	B	2									Rd8+#xx:8→Rd8	—	↕	↕	↕	↕	↕	1
	ADD.B Rs,Rd	B		2								Rd8+Rs8→Rd8	—	↕	↕	↕	↕	↕	1
	ADD.W #xx:16,Rd	W	4									Rd16+#xx:16→Rd16	—	[3]	↕	↕	↕	↕	2
	ADD.W Rs,Rd	W		2								Rd16+Rs16→Rd16	—	[3]	↕	↕	↕	↕	1
	ADD.L #xx:32,ERd	L	6									ERd32+#xx:32→ERd32	—	[4]	↕	↕	↕	↕	3
	ADD.L ERs,ERd	L		2								ERd32+ERs32→ERd32	—	[4]	↕	↕	↕	↕	1
ADDX	ADDX #xx:8,Rd	B	2									Rd8+#xx:8+C→Rd8	—	↕	↕	[5]	↕	↕	1
	ADDX Rs,Rd	B		2								Rd8+Rs8+C→Rd8	—	↕	↕	[5]	↕	↕	1
ADDS	ADDS #1,ERd	L		2								ERd32+1→ERd32	—	—	—	—	—	—	1
	ADDS #2,ERd	L		2								ERd32+2→ERd32	—	—	—	—	—	—	1
	ADDS #4,ERd	L		2								ERd32+4→ERd32	—	—	—	—	—	—	1
INC	INC.B Rd	B		2								Rd8+1→Rd8	—	—	↕	↕	↕	—	1
	INC.W #1,Rd	W		2								Rd16+1→Rd16	—	—	↕	↕	↕	—	1
	INC.W #2,Rd	W		2								Rd16+2→Rd16	—	—	↕	↕	↕	—	1
	INC.L #1,ERd	L		2								ERd32+1→ERd32	—	—	↕	↕	↕	—	1
	INC.L #2,ERd	L		2								ERd32+2→ERd32	—	—	↕	↕	↕	—	1
DAA	DAA Rd	B		2								Rd8 decimal adjust→Rd8	—	*	↕	↕	*	↕	1
SUB	SUB.B Rs,Rd	B		2								Rd8-Rs8→Rd8	—	↕	↕	↕	↕	↕	1
	SUB.W #xx:16,Rd	W	4									Rd16-#xx:16→Rd16	—	[3]	↕	↕	↕	↕	2

Mnemonic	Operand Size	Addressing Mode/ Instruction Length (Bytes)								Operation	Condition Code						No. of States
		#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa		I	H	N	Z	V	C	Advanced
SUB	SUB.W Rs,Rd	W	2							Rd16-Rs16→Rd16	—	[3]	↕	↕	↕	↕	1
	SUB.L #xx:32,ERd	L	6							ERd32-#xx:32→ERd32	—	[4]	↕	↕	↕	↕	3
	SUB.L ERs,ERd	L	2							ERd32-ERs32→ERd32	—	[4]	↕	↕	↕	↕	1
SUBX	SUBX #xx:8,Rd	B	2							Rd8-#xx:8-C→Rd8	—	↕	↕	[5]	↕	↕	1
	SUBX Rs,Rd	B	2							Rd8-Rs8-C→Rd8	—	↕	↕	[5]	↕	↕	1
SUBS	SUBS #1,ERd	L	2							ERd32-1→ERd32	—	—	—	—	—	—	1
	SUBS #2,ERd	L	2							ERd32-2→ERd32	—	—	—	—	—	—	1
	SUBS #4,ERd	L	2							ERd32-4→ERd32	—	—	—	—	—	—	1
DEC	DEC.B Rd	B	2							Rd8-1→Rd8	—	—	↕	↕	↕	—	1
	DEC.W #1,Rd	W	2							Rd16-1→Rd16	—	—	↕	↕	↕	—	1
	DEC.W #2,Rd	W	2							Rd16-2→Rd16	—	—	↕	↕	↕	—	1
	DEC.L #1,ERd	L	2							ERd32-1→ERd32	—	—	↕	↕	↕	—	1
	DEC.L #2,ERd	L	2							ERd32-2→ERd32	—	—	↕	↕	↕	—	1
DAS	DAS Rd	B	2							Rd8 decimal adjust→Rd8	—	*	↕	↕	*	—	1
MULXU	MULXU.B Rs,Rd	B	2							Rd8×Rs8→Rd16 (unsigned multiplication)	—	—	—	—	—	—	12
	MULXU.W Rs,ERd	W	2							Rd16×Rs16→ERd32 (unsigned multiplication)	—	—	—	—	—	—	20
MULXS	MULXS.B Rs,Rd	B	4							Rd8×Rs8→Rd16 (signed multiplication)	—	—	↕	↕	—	—	13
	MULXS.W Rs,ERd	W	4							Rd16×Rs16→ERd32 (signed multiplication)	—	—	↕	↕	—	—	21

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of St
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Advan
DIVXU	DIVXU.B Rs,Rd	B		2								Rd16÷Rs8→Rd16 (RdH: remainder, RdL: quotient) (unsigned division)	—	—	[6]	[7]	—	—	12
	DIVXU.W Rs,ERd	W		2								ERd32÷Rs16→ERd32 (Ed: remainder, Rd: quotient) (unsigned division)	—	—	[6]	[7]	—	—	20
DIVXS	DIVXS.B Rs,Rd	B		4								Rd16÷Rs8→Rd16 (RdH: remainder, RdL: quotient) (signed division)	—	—	[8]	[7]	—	—	13
	DIVXS.W Rs,ERd	W		4								ERd32÷Rs16→ERd32 (Ed: remainder, Rd: quotient) (signed division)	—	—	[8]	[7]	—	—	21
CMP	CMP.B #xx:8,Rd	B	2									Rd8-#xx:8	—	↕	↕	↕	↕	↕	1
	CMP.B Rs,Rd	B		2								Rd8-Rs8	—	↕	↕	↕	↕	↕	1
	CMP.W #xx:16,Rd	W	4									Rd16-#xx:16	—	[3]	↕	↕	↕	↕	2
	CMP.W Rs,Rd	W		2								Rd16-Rs16	—	[3]	↕	↕	↕	↕	1
	CMP.L #xx:32,ERd	L	6									ERd32-#xx:32	—	[4]	↕	↕	↕	↕	3
	CMP.L ERs,ERd	L		2								ERd32-ERs32	—	[4]	↕	↕	↕	↕	1
NEG	NEG.B Rd	B		2								0-Rd8→Rd8	—	↕	↕	↕	↕	↕	1
	NEG.W Rd	W		2								0-Rd16→Rd16	—	↕	↕	↕	↕	↕	1
	NEG.L ERd	L		2								0-ERd32→ERd32	—	↕	↕	↕	↕	↕	1
EXTU	EXTU.W Rd	W		2								0→(<bits 15 to 8> of Rd16)	—	—	0	↕	0	—	1
	EXTU.L ERd	L		2								0→(<bits 31 to 16> of ERd32)	—	—	0	↕	0	—	1

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code					No. of States	
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa			I	H	N	Z	V	C	Advanced
EXTS	EXTS.W Rd	W		2								(<bit 7> of Rd16)→ (<bits 15 to 8> of Rd16)	—	—	↕	↕	0	—	1
	EXTS.L ERd	L		2								(<bit 15> of ERd32)→ (<bits 31 to 16> of ERd32)	—	—	↕	↕	0	—	1
TAS	TAS @ERd*3	B			4							@ERd-0→CCR set, (1)→ (<bit 7> of @ERd)	—	—	↕	↕	0	—	4
MAC	MAC @ERn+, @ERm+	Cannot be used in the chip																	[2]
CLRMAC	CLRMAC																		
LDMAC	LDMAC ERs,MACH																		
	LDMAC ERs,MACL																		
STMAC	STMAC MACH,ERd																		
	STMAC MACL,ERd																		

(3) Logical Instructions

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of St
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Advan
AND	AND.B #xx:8,Rd	B	2									Rd8∧#xx:8→Rd8	—	—	↕	↕	0	—	1
	AND.B Rs,Rd	B		2								Rd8∧Rs8→Rd8	—	—	↕	↕	0	—	1
	AND.W #xx:16,Rd	W	4									Rd16∧#xx:16→Rd16	—	—	↕	↕	0	—	2
	AND.W Rs,Rd	W		2								Rd16∧Rs16→Rd16	—	—	↕	↕	0	—	1
	AND.L #xx:32,ERd	L	6									ERd32∧#xx:32→ERd32	—	—	↕	↕	0	—	3
	AND.L ERs,ERd	L		4								ERd32∧ERs32→ERd32	—	—	↕	↕	0	—	2
OR	OR.B #xx:8,Rd	B	2									Rd8∨#xx:8→Rd8	—	—	↕	↕	0	—	1
	OR.B Rs,Rd	B		2								Rd8∨Rs8→Rd8	—	—	↕	↕	0	—	1
	OR.W #xx:16,Rd	W	4									Rd16∨#xx:16→Rd16	—	—	↕	↕	0	—	2
	OR.W Rs,Rd	W		2								Rd16∨Rs16→Rd16	—	—	↕	↕	0	—	1
	OR.L #xx:32,ERd	L	6									ERd32∨#xx:32→ERd32	—	—	↕	↕	0	—	3
	OR.L ERs,ERd	L		4								ERd32∨ERs32→ERd32	—	—	↕	↕	0	—	2
XOR	XOR.B #xx:8,Rd	B	2									Rd8⊕#xx:8→Rd8	—	—	↕	↕	0	—	1
	XOR.B Rs,Rd	B		2								Rd8⊕Rs8→Rd8	—	—	↕	↕	0	—	1
	XOR.W #xx:16,Rd	W	4									Rd16⊕#xx:16→Rd16	—	—	↕	↕	0	—	2
	XOR.W Rs,Rd	W		2								Rd16⊕Rs16→Rd16	—	—	↕	↕	0	—	1
	XOR.L #xx:32,ERd	L	6									ERd32⊕#xx:32→ERd32	—	—	↕	↕	0	—	3
	XOR.L ERs,ERd	L		4								ERd32⊕ERs32→ERd32	—	—	↕	↕	0	—	2
NOT	NOT.B Rd	B		2								¬ Rd8→Rd8	—	—	↕	↕	0	—	1
	NOT.W Rd	W		2								¬ Rd16→Rd16	—	—	↕	↕	0	—	1
	NOT.L ERd	L		2								¬ ERd32→ERd32	—	—	↕	↕	0	—	1

(4) Shift Instructions

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code					No. of States
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Advanced
SHAL	SHAL.B Rd	B		2									—	—	↕	↕	0	↕	1
	SHAL.B #2,Rd	B		2									—	—	↕	↕	0	↕	1
	SHAL.W Rd	W		2									—	—	↕	↕	0	↕	1
	SHAL.W #2,Rd	W		2									—	—	↕	↕	0	↕	1
	SHAL.L ERd	L		2									—	—	↕	↕	0	↕	1
	SHAL.L #2,ERd	L		2									—	—	↕	↕	0	↕	1
SHAR	SHAR.B Rd	B		2									—	—	↕	↕	0	↕	1
	SHAR.B #2,Rd	B		2									—	—	↕	↕	0	↕	1
	SHAR.W Rd	W		2									—	—	↕	↕	0	↕	1
	SHAR.W #2,Rd	W		2									—	—	↕	↕	0	↕	1
	SHAR.L ERd	L		2									—	—	↕	↕	0	↕	1
	SHAR.L #2,ERd	L		2									—	—	↕	↕	0	↕	1
SHLL	SHLL.B Rd	B		2									—	—	↕	↕	0	↕	1
	SHLL.B #2,Rd	B		2									—	—	↕	↕	0	↕	1
	SHLL.W Rd	W		2									—	—	↕	↕	0	↕	1
	SHLL.W #2,Rd	W		2									—	—	↕	↕	0	↕	1
	SHLL.L ERd	L		2									—	—	↕	↕	0	↕	1
	SHLL.L #2,ERd	L		2									—	—	↕	↕	0	↕	1

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code					No. of States
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Advanced
SHLR	SHLR.B Rd	B		2								0 → MSB → LSB C	—	—	0	↕	0	↕	1
	SHLR.B #2,Rd	B		2									—	—	0	↕	0	↕	1
	SHLR.W Rd	W		2									—	—	0	↕	0	↕	1
	SHLR.W #2,Rd	W		2									—	—	0	↕	0	↕	1
	SHLR.L ERd	L		2									—	—	0	↕	0	↕	1
	SHLR.L #2,ERd	L		2									—	—	0	↕	0	↕	1
ROTXL	ROTXL.B Rd	B		2								C MSB → LSB	—	—	↕	↕	0	↕	1
	ROTXL.B #2,Rd	B		2									—	—	↕	↕	0	↕	1
	ROTXL.W Rd	W		2									—	—	↕	↕	0	↕	1
	ROTXL.W #2,Rd	W		2									—	—	↕	↕	0	↕	1
	ROTXL.L ERd	L		2									—	—	↕	↕	0	↕	1
	ROTXL.L #2,ERd	L		2									—	—	↕	↕	0	↕	1
ROTXR	ROTXR.B Rd	B		2								MSB → LSB C	—	—	↕	↕	0	↕	1
	ROTXR.B #2,Rd	B		2									—	—	↕	↕	0	↕	1
	ROTXR.W Rd	W		2									—	—	↕	↕	0	↕	1
	ROTXR.W #2,Rd	W		2									—	—	↕	↕	0	↕	1
	ROTXR.L ERd	L		2									—	—	↕	↕	0	↕	1
	ROTXR.L #2,ERd	L		2									—	—	↕	↕	0	↕	1

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code					No. of States
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Advanced
ROTL	ROTL.B Rd	B		2									—	—	↕	↕	0	↕	1
	ROTL.B #2,Rd	B		2									—	—	↕	↕	0	↕	1
	ROTL.W Rd	W		2									—	—	↕	↕	0	↕	1
	ROTL.W #2,Rd	W		2									—	—	↕	↕	0	↕	1
	ROTL.L ERd	L		2									—	—	↕	↕	0	↕	1
	ROTL.L #2,ERd	L		2									—	—	↕	↕	0	↕	1
ROTR	ROTR.B Rd	B		2									—	—	↕	↕	0	↕	1
	ROTR.B #2,Rd	B		2									—	—	↕	↕	0	↕	1
	ROTR.W Rd	W		2									—	—	↕	↕	0	↕	1
	ROTR.W #2,Rd	W		2									—	—	↕	↕	0	↕	1
	ROTR.L ERd	L		2									—	—	↕	↕	0	↕	1
	ROTR.L #2,ERd	L		2									—	—	↕	↕	0	↕	1

(5) Bit-Manipulation Instructions

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of St
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Advan
BSET	BSET #xx:3,Rd	B		2								(#xx:3 of Rd8)←1	—	—	—	—	—	—	1
	BSET #xx:3,@ERd	B			4							(#xx:3 of @ERd)←1	—	—	—	—	—	—	4
	BSET #xx:3,@aa:8	B						4				(#xx:3 of @aa:8)←1	—	—	—	—	—	—	4
	BSET #xx:3,@aa:16	B						6				(#xx:3 of @aa:16)←1	—	—	—	—	—	—	5
	BSET #xx:3,@aa:32	B						8				(#xx:3 of @aa:32)←1	—	—	—	—	—	—	6
	BSET Rn,Rd	B		2								(Rn8 of Rd8)←1	—	—	—	—	—	—	1
	BSET Rn,@ERd	B			4							(Rn8 of @ERd)←1	—	—	—	—	—	—	4
	BSET Rn,@aa:8	B						4				(Rn8 of @aa:8)←1	—	—	—	—	—	—	4
	BSET Rn,@aa:16	B						6				(Rn8 of @aa:16)←1	—	—	—	—	—	—	5
	BSET Rn,@aa:32	B						8				(Rn8 of @aa:32)←1	—	—	—	—	—	—	6
BCLR	BCLR #xx:3,Rd	B		2								(#xx:3 of Rd8)←0	—	—	—	—	—	—	1
	BCLR #xx:3,@ERd	B			4							(#xx:3 of @ERd)←0	—	—	—	—	—	—	4
	BCLR #xx:3,@aa:8	B						4				(#xx:3 of @aa:8)←0	—	—	—	—	—	—	4
	BCLR #xx:3,@aa:16	B						6				(#xx:3 of @aa:16)←0	—	—	—	—	—	—	5
	BCLR #xx:3,@aa:32	B						8				(#xx:3 of @aa:32)←0	—	—	—	—	—	—	6
	BCLR Rn,Rd	B		2								(Rn8 of Rd8)←0	—	—	—	—	—	—	1
	BCLR Rn,@ERd	B			4							(Rn8 of @ERd)←0	—	—	—	—	—	—	4
	BCLR Rn,@aa:8	B						4				(Rn8 of @aa:8)←0	—	—	—	—	—	—	4
	BCLR Rn,@aa:16	B						6				(Rn8 of @aa:16)←0	—	—	—	—	—	—	5

Mnemonic	Operand Size	Addressing Mode/ Instruction Length (Bytes)								Operation	Condition Code						No. of States
		#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa		I	H	N	Z	V	C	Advanced
BCLR	BCLR Rn,@aa:32	B					8			(Rn8 of @aa:32)←0	—	—	—	↕	—	—	6
BNOT	BNOT #xx:3,Rd	B	2							(#xx:3 of Rd8)←[¬ (#xx:3 of Rd8)]	—	—	—	↕	—	—	1
	BNOT #xx:3,@ERd	B		4						(#xx:3 of @ERd)←[¬ (#xx:3 of @ERd)]	—	—	—	↕	—	—	4
	BNOT #xx:3,@aa:8	B					4			(#xx:3 of @aa:8)←[¬ (#xx:3 of @aa:8)]	—	—	—	↕	—	—	4
	BNOT #xx:3,@aa:16	B					6			(#xx:3 of @aa:16)←[¬ (#xx:3 of @aa:16)]	—	—	—	↕	—	—	5
	BNOT #xx:3,@aa:32	B					8			(#xx:3 of @aa:32)←[¬ (#xx:3 of @aa:32)]	—	—	—	↕	—	—	6
	BNOT Rn,Rd	B	2							(Rn8 of Rd8)←[¬ (Rn8 of Rd8)]	—	—	—	↕	—	—	1
	BNOT Rn,@ERd	B		4						(Rn8 of @ERd)←[¬ (Rn8 of @ERd)]	—	—	—	↕	—	—	4
	BNOT Rn,@aa:8	B					4			(Rn8 of @aa:8)←[¬ (Rn8 of @aa:8)]	—	—	—	↕	—	—	4
	BNOT Rn,@aa:16	B					6			(Rn8 of @aa:16)←[¬ (Rn8 of @aa:16)]	—	—	—	↕	—	—	5
	BNOT Rn,@aa:32	B					8			(Rn8 of @aa:32)←[¬ (Rn8 of @aa:32)]	—	—	—	↕	—	—	6
BTST	BTST #xx:3,Rd	B	2							¬ (#xx:3 of Rd8)→Z	—	—	—	↕	—	—	1
	BTST #xx:3,@ERd	B		4						¬ (#xx:3 of @ERd)→Z	—	—	—	↕	—	—	3
	BTST #xx:3,@aa:8	B					4			¬ (#xx:3 of @aa:8)→Z	—	—	—	↕	—	—	3
	BTST #xx:3,@aa:16	B					6			¬ (#xx:3 of @aa:16)→Z	—	—	—	↕	—	—	4

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)								Operation	Condition Code					No. of States
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa		I	H	N	Z	V	C	Advanced
BTST	BTST #xx:3,@aa:32	B						8			¬ (#xx:3 of @aa:32)→Z	—	—	—	↕	—	—	5
	BTST Rn,Rd	B		2							¬ (Rn8 of Rd8)→Z	—	—	—	↕	—	—	1
	BTST Rn,@ERd	B			4						¬ (Rn8 of @ERd)→Z	—	—	—	↕	—	—	3
	BTST Rn,@aa:8	B						4			¬ (Rn8 of @aa:8)→Z	—	—	—	↕	—	—	3
	BTST Rn,@aa:16	B						6			¬ (Rn8 of @aa:16)→Z	—	—	—	↕	—	—	4
	BTST Rn,@aa:32	B						8			¬ (Rn8 of @aa:32)→Z	—	—	—	↕	—	—	5
BLD	BLD #xx:3,Rd	B		2							(#xx:3 of Rd8)→C	—	—	—	—	—	↕	1
	BLD #xx:3,@ERd	B			4						(#xx:3 of @ERd)→C	—	—	—	—	—	↕	3
	BLD #xx:3,@aa:8	B						4			(#xx:3 of @aa:8)→C	—	—	—	—	—	↕	3
	BLD #xx:3,@aa:16	B						6			(#xx:3 of @aa:16)→C	—	—	—	—	—	↕	4
	BLD #xx:3,@aa:32	B						8			(#xx:3 of @aa:32)→C	—	—	—	—	—	↕	5
BILD	BILD #xx:3,Rd	B		2							¬ (#xx:3 of Rd8)→C	—	—	—	—	—	↕	1
	BILD #xx:3,@ERd	B			4						¬ (#xx:3 of @ERd)→C	—	—	—	—	—	↕	3
	BILD #xx:3,@aa:8	B						4			¬ (#xx:3 of @aa:8)→C	—	—	—	—	—	↕	3
	BILD #xx:3,@aa:16	B						6			¬ (#xx:3 of @aa:16)→C	—	—	—	—	—	↕	4
	BILD #xx:3,@aa:32	B						8			¬ (#xx:3 of @aa:32)→C	—	—	—	—	—	↕	5
BST	BST #xx:3,Rd	B		2							C→(#xx:3 of Rd8)	—	—	—	—	—	—	1
	BST #xx:3,@ERd	B			4						C→(#xx:3 of @ERd)	—	—	—	—	—	—	4
	BST #xx:3,@aa:8	B						4			C→(#xx:3 of @aa:8)	—	—	—	—	—	—	4

Mnemonic	Operand Size	Addressing Mode/ Instruction Length (Bytes)								Operation	Condition Code						No. of States	
		#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa		I	H	N	Z	V	C	Advanced	
BST	BST #xx:3,@aa:16	B					6			C→(#xx:3 of @aa:16)	—	—	—	—	—	—	5	
	BST #xx:3,@aa:32	B					8			C→(#xx:3 of @aa:32)	—	—	—	—	—	—	6	
BIST	BIST #xx:3,Rd	B	2							¬ C→(#xx:3 of Rd8)	—	—	—	—	—	—	1	
	BIST #xx:3,@ERd	B		4						¬ C→(#xx:3 of @ERd)	—	—	—	—	—	—	4	
	BIST #xx:3,@aa:8	B					4			¬ C→(#xx:3 of @aa:8)	—	—	—	—	—	—	4	
	BIST #xx:3,@aa:16	B					6			¬ C→(#xx:3 of @aa:16)	—	—	—	—	—	—	5	
	BIST #xx:3,@aa:32	B					8			¬ C→(#xx:3 of @aa:32)	—	—	—	—	—	—	6	
BAND	BAND #xx:3,Rd	B	2							C∧(#xx:3 of Rd8)→C	—	—	—	—	—	↕	1	
	BAND #xx:3,@ERd	B		4						C∧(#xx:3 of @ERd)→C	—	—	—	—	—	↕	3	
	BAND #xx:3,@aa:8	B					4			C∧(#xx:3 of @aa:8)→C	—	—	—	—	—	↕	3	
	BAND #xx:3,@aa:16	B					6			C∧(#xx:3 of @aa:16)→C	—	—	—	—	—	↕	4	
	BAND #xx:3,@aa:32	B					8			C∧(#xx:3 of @aa:32)→C	—	—	—	—	—	↕	5	
BIAND	BIAND #xx:3,Rd	B	2							C∧[¬ (#xx:3 of Rd8)]→C	—	—	—	—	—	↕	1	
	BIAND #xx:3,@ERd	B		4						C∧[¬ (#xx:3 of @ERd)]→C	—	—	—	—	—	↕	3	
	BIAND #xx:3,@aa:8	B					4			C∧[¬ (#xx:3 of @aa:8)]→C	—	—	—	—	—	↕	3	
	BIAND #xx:3,@aa:16	B					6			C∧[¬ (#xx:3 of @aa:16)]→C	—	—	—	—	—	↕	4	
	BIAND #xx:3,@aa:32	B					8			C∧[¬ (#xx:3 of @aa:32)]→C	—	—	—	—	—	↕	5	
BOR	BOR #xx:3,Rd	B	2							C∨(#xx:3 of Rd8)→C	—	—	—	—	—	↕	1	
	BOR #xx:3,@ERd	B		4						C∨(#xx:3 of @ERd)→C	—	—	—	—	—	↕	3	

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)								Operation	Condition Code						No. of States
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa		I	H	N	Z	V	C	Advanced
BOR	BOR #xx:3,@aa:8	B						4			C∨(#xx:3 of @aa:8)→C	—	—	—	—	—	↕	3
	BOR #xx:3,@aa:16	B						6			C∨(#xx:3 of @aa:16)→C	—	—	—	—	—	↕	4
	BOR #xx:3,@aa:32	B						8			C∨(#xx:3 of @aa:32)→C	—	—	—	—	—	↕	5
BIOR	BIOR #xx:3,Rd	B		2							C∨[¬ (#xx:3 of Rd8)]→C	—	—	—	—	—	↕	1
	BIOR #xx:3,@ERd	B			4						C∨[¬ (#xx:3 of @ERd)]→C	—	—	—	—	—	↕	3
	BIOR #xx:3,@aa:8	B						4			C∨[¬ (#xx:3 of @aa:8)]→C	—	—	—	—	—	↕	3
	BIOR #xx:3,@aa:16	B						6			C∨[¬ (#xx:3 of @aa:16)]→C	—	—	—	—	—	↕	4
	BIOR #xx:3,@aa:32	B						8			C∨[¬ (#xx:3 of @aa:32)]→C	—	—	—	—	—	↕	5
BXOR	BXOR #xx:3,Rd	B		2							C⊕(#xx:3 of Rd8)→C	—	—	—	—	—	↕	1
	BXOR #xx:3,@ERd	B			4						C⊕(#xx:3 of @ERd)→C	—	—	—	—	—	↕	3
	BXOR #xx:3,@aa:8	B						4			C⊕(#xx:3 of @aa:8)→C	—	—	—	—	—	↕	3
	BXOR #xx:3,@aa:16	B						6			C⊕(#xx:3 of @aa:16)→C	—	—	—	—	—	↕	4
	BXOR #xx:3,@aa:32	B						8			C⊕(#xx:3 of @aa:32)→C	—	—	—	—	—	↕	5
BIXOR	BIXOR #xx:3,Rd	B		2							C⊕[¬ (#xx:3 of Rd8)]→C	—	—	—	—	—	↕	1
	BIXOR #xx:3,@ERd	B			4						C⊕[¬ (#xx:3 of @ERd)]→C	—	—	—	—	—	↕	3
	BIXOR #xx:3,@aa:8	B						4			C⊕[¬ (#xx:3 of @aa:8)]→C	—	—	—	—	—	↕	3
	BIXOR #xx:3,@aa:16	B						6			C⊕[¬ (#xx:3 of @aa:16)]→C	—	—	—	—	—	↕	4
	BIXOR #xx:3,@aa:32	B						8			C⊕[¬ (#xx:3 of @aa:32)]→C	—	—	—	—	—	↕	5

(6) Branch Instructions

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)										Operation	Branching Condition	Condition Code						No. of States	
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa					I	H	N	Z	V	C	Advanced	
Bcc	BRA d:8(BT d:8)	—							2				if condition is true then PC←PC+d else next;	Always	—	—	—	—	—	—	2	
	BRA d:16(BT d:16)	—							4						—	—	—	—	—	—	3	
	BRN d:8(BF d:8)	—							2					Never	—	—	—	—	—	—	2	
	BRN d:16(BF d:16)	—							4						—	—	—	—	—	—	3	
	BHI d:8	—							2					C∨Z=0	—	—	—	—	—	—	2	
	BHI d:16	—							4						—	—	—	—	—	—	3	
	BLS d:8	—							2					C∨Z=1	—	—	—	—	—	—	2	
	BLS d:16	—							4						—	—	—	—	—	—	3	
	BCC d:8(BHS d:8)	—							2					C=0	—	—	—	—	—	—	2	
	BCC d:16(BHS d:16)	—							4						—	—	—	—	—	—	3	
	BCS d:8(BLO d:8)	—							2					C=1	—	—	—	—	—	—	2	
	BCS d:16(BLO d:16)	—							4						—	—	—	—	—	—	3	
	BNE d:8	—							2					Z=0	—	—	—	—	—	—	2	
	BNE d:16	—							4						—	—	—	—	—	—	3	
	BEQ d:8	—							2					Z=1	—	—	—	—	—	—	2	
	BEQ d:16	—							4						—	—	—	—	—	—	3	
	BVC d:8	—							2					V=0	—	—	—	—	—	—	2	
	BVC d:16	—							4						—	—	—	—	—	—	3	

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|	Branching Condition	I	H	N	Z	V	C	Advanced
Bcc	BVS d:8	—							2			V=1	—	—	—	—	—	—	2
	BVS d:16	—							4				—	—	—	—	—	—	3
	BPL d:8	—							2			N=0	—	—	—	—	—	—	2
	BPL d:16	—							4				—	—	—	—	—	—	3
	BMI d:8	—							2			N=1	—	—	—	—	—	—	2
	BMI d:16	—							4				—	—	—	—	—	—	3
	BGE d:8	—							2			N⊕V=0	—	—	—	—	—	—	2
	BGE d:16	—							4				—	—	—	—	—	—	3
	BLT d:8	—							2			N⊕V=1	—	—	—	—	—	—	2
	BLT d:16	—							4				—	—	—	—	—	—	3
	BGT d:8	—							2			Z∨(N⊕V)=0	—	—	—	—	—	—	2
	BGT d:16	—							4				—	—	—	—	—	—	3
	BLE d:8	—							2			Z∨(N⊕V)=1	—	—	—	—	—	—	2
	BLE d:16	—							4				—	—	—	—	—	—	3

Mnemonic	Operand Size	Addressing Mode/ Instruction Length (Bytes)								Operation	Condition Code						No. of States	
		#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa		I	H	N	Z	V	C	Advan	
JMP	JMP @ERn	—		2						PC← ERn	—	—	—	—	—	—	2	
	JMP @aa:24	—					4			PC← aa:24	—	—	—	—	—	—	3	
	JMP @@aa:8	—							2	PC← @aa:8	—	—	—	—	—	—	5	
BSR	BSR d:8	—						2		PC→@-SP,PC← PC+d:8	—	—	—	—	—	—	4	
	BSR d:16	—						4		PC→@-SP,PC← PC+d:16	—	—	—	—	—	—	5	
JSR	JSR @ERn	—		2						PC→@-SP,PC← ERn	—	—	—	—	—	—	4	
	JSR @aa:24	—					4			PC→@-SP,PC← aa:24	—	—	—	—	—	—	5	
	JSR @@aa:8	—							2	PC→@-SP,PC← @aa:8	—	—	—	—	—	—	6	
RTS	RTS	—							2	PC← @SP+	—	—	—	—	—	—	5	

(7) System Control Instructions

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of State
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Advance
TRAPA	TRAPA #xx:2	—										PC→@-SP,CCR→@-SP, EXR→@-SP,<vector>→PC	1	—	—	—	—	—	8 [9]
RTE	RTE	—										EXR←@SP+,CCR←@SP+, PC←@SP+	↕	↕	↕	↕	↕	↕	5 [9]
SLEEP	SLEEP	—										Transition to power-down state	—	—	—	—	—	—	2
LDC	LDC #xx:8,CCR	B	2									#xx:8→CCR	↕	↕	↕	↕	↕	↕	1
	LDC #xx:8,EXR	B	4									#xx:8→EXR	—	—	—	—	—	—	2
	LDC Rs,CCR	B		2								Rs8→CCR	↕	↕	↕	↕	↕	↕	1
	LDC Rs,EXR	B		2								Rs8→EXR	—	—	—	—	—	—	1
	LDC @ERs,CCR	W			4							@ERs→CCR	↕	↕	↕	↕	↕	↕	3
	LDC @ERs,EXR	W			4							@ERs→EXR	—	—	—	—	—	—	3
	LDC @(d:16,ERs),CCR	W				6						@(d:16,ERs)→CCR	↕	↕	↕	↕	↕	↕	4
	LDC @(d:16,ERs),EXR	W				6						@(d:16,ERs)→EXR	—	—	—	—	—	—	4
	LDC @(d:32,ERs),CCR	W				10						@(d:32,ERs)→CCR	↕	↕	↕	↕	↕	↕	6
	LDC @(d:32,ERs),EXR	W				10						@(d:32,ERs)→EXR	—	—	—	—	—	—	6
	LDC @ERs+,CCR	W					4					@ERs→CCR,ERs32+2→ERs32	↕	↕	↕	↕	↕	↕	4
	LDC @ERs+,EXR	W					4					@ERs→EXR,ERs32+2→ERs32	—	—	—	—	—	—	4
	LDC @aa:16,CCR	W						6				@aa:16→CCR	↕	↕	↕	↕	↕	↕	4
	LDC @aa:16,EXR	W						6				@aa:16→EXR	—	—	—	—	—	—	4
	LDC @aa:32,CCR	W						8				@aa:32→CCR	↕	↕	↕	↕	↕	↕	5
	LDC @aa:32,EXR	W						8				@aa:32→EXR	—	—	—	—	—	—	5

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)								Operation	Condition Code						No. of States	
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa		I	H	N	Z	V	C	Advanced	
STC	STC CCR,Rd	B	2								CCR→Rd8	—	—	—	—	—	—	1	
	STC EXR,Rd	B	2								EXR→Rd8	—	—	—	—	—	—	1	
	STC CCR,@ERd	W			4						CCR→@ERd	—	—	—	—	—	—	3	
	STC EXR,@ERd	W			4						EXR→@ERd	—	—	—	—	—	—	3	
	STC CCR,@(d:16,ERd)	W				6					CCR→@(d:16,ERd)	—	—	—	—	—	—	4	
	STC EXR,@(d:16,ERd)	W				6					EXR→@(d:16,ERd)	—	—	—	—	—	—	4	
	STC CCR,@(d:32,ERd)	W				10					CCR→@(d:32,ERd)	—	—	—	—	—	—	6	
	STC EXR,@(d:32,ERd)	W				10					EXR→@(d:32,ERd)	—	—	—	—	—	—	6	
	STC CCR,@-ERd	W					4				ERd32-2→ERd32,CCR→@ERd	—	—	—	—	—	—	4	
	STC EXR,@-ERd	W					4				ERd32-2→ERd32,EXR→@ERd	—	—	—	—	—	—	4	
	STC CCR,@aa:16	W						6			CCR→@aa:16	—	—	—	—	—	—	4	
	STC EXR,@aa:16	W						6			EXR→@aa:16	—	—	—	—	—	—	4	
	STC CCR,@aa:32	W						8			CCR→@aa:32	—	—	—	—	—	—	5	
	STC EXR,@aa:32	W						8			EXR→@aa:32	—	—	—	—	—	—	5	
ANDC	ANDC #xx:8,CCR	B	2								CCR∧#xx:8→CCR	↕	↕	↕	↕	↕	↕	1	
	ANDC #xx:8,EXR	B	4								EXR∧#xx:8→EXR	—	—	—	—	—	—	2	
ORC	ORC #xx:8,CCR	B	2								CCR∨#xx:8→CCR	↕	↕	↕	↕	↕	↕	1	
	ORC #xx:8,EXR	B	4								EXR∨#xx:8→EXR	—	—	—	—	—	—	2	
XORC	XORC #xx:8,CCR	B	2								CCR⊕#xx:8→CCR	↕	↕	↕	↕	↕	↕	1	
	XORC #xx:8,EXR	B	4								EXR⊕#xx:8→EXR	—	—	—	—	—	—	2	
NOP	NOP	—									PC←PC+2	—	—	—	—	—	—	1	

(8) Block Transfer Instructions

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of St
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Advan
EEPMOV	EEPMOV.B	—									4	if R4L≠0 Repeat @ER5→@ER6 ER5+1→ER5 ER6+1→ER6 R4L-1→R4L Until R4L=0 else next;	—	—	—	—	—	—	4+2n
	EEPMOV.W	—									4	if R4≠0 Repeat @ER5→@ER6 ER5+1→ER5 ER6+1→ER6 R4-1→R4 Until R4=0 else next;	—	—	—	—	—	—	4+2n

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
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 chip.
[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.

Table A.2 shows the instruction codes.

Instruction	Mnemonic	Size	Instruction Format																	
			1st byte		2nd byte		3rd byte		4th byte		5th byte		6th byte		7th byte		8th byte		9th byte	10th 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	0		
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	8th byte	9th byte	10
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	8th byte	9th byte	10
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	9th byte	10t
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 IMM 0		
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 IMM 0		
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 IMM 0		
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 IMM 0		
	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 0		

RENESAS

Instruction	Mnemonic	Size	Instruction Format																											
			1st byte				2nd byte				3rd byte				4th byte				5th byte				6th byte				7th byte				8th byte				9th byte				10
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	IMM	0						
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	IMM	0						
	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	0							
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	IMM	0						
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	IMM	0						
	BTST Rn,Rd	B	6	3	rn	rd																															
	BTST Rn,@ERd	B	7	C	0	erd	0				6	3	rn	0																							

Instruction	Mnemonic	Size	Instruction Format																					
			1st byte		2nd byte		3rd byte		4th byte		5th byte		6th byte		7th byte		8th byte		9th byte		10	
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	rn	0				
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	IMM	0			
CLRMAC	CLRMAC	—	Cannot be used in the chip																			
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		8th byte		9th byte		10	
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	disp							
	LDC @(d:32,ERs),EXR	W	0	1	4	1	7	8	0 ers	0	6	B	2	0	disp							
	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	abs											
	LDC @aa:16,EXR	W	0	1	4	1	6	B	0	0	abs											

Instruction	Mnemonic	Size	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte	9th byte	10
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	Cannot be used in the chip									
	LDMAC ERs,MACL	L										
MAC	MAC @ERn+,@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							
	MOV.W @(d:32,ERs),Rd	W	7 8	0:ers 0	6 B	2 rd	disp					

Instruction	Mnemonic	Size	Instruction Format: 1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte	9th byte	10th byte
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	disp			
	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	disp			
	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 chip									
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								
	MULXU.W Rs,ERd	W	5 2	rs 0 erd								

Instruction	Mnemonic	Size	Instruction Format									
			1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte	9th byte	10
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 rn								
	POP.L ERn	L	0 1	0 0	6 D	7 0 ern						
PUSH	PUSH.W Rn	W	6 D	F rn								
	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								

Instruction	Mnemonic	Size	Instruction Format											
			1st byte		2nd byte		3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte	9th byte	10
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	8th byte	9th byte	10
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	disp			
	STC.W EXR,@(d:32,ERd)	W	0 1 4 1	7 8		0 erd 0	6	B A 0	disp			
	STC.W CCR,@-ERd	W	0 1 4 0	6 D		1 erd 0						
	STC.W EXR,@-ERd	W	0 1 4 1	6 D		1 erd 0						

Instruction	Mnemonic	Size	Instruction Format																					
			1st byte		2nd byte		3rd byte		4th byte		5th byte		6th byte		7th byte		8th byte		9th byte		10	
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	Cannot be used in the chip																			
	STMAC MACL,ERd	L																				
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	8th byte	9th byte	10
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:

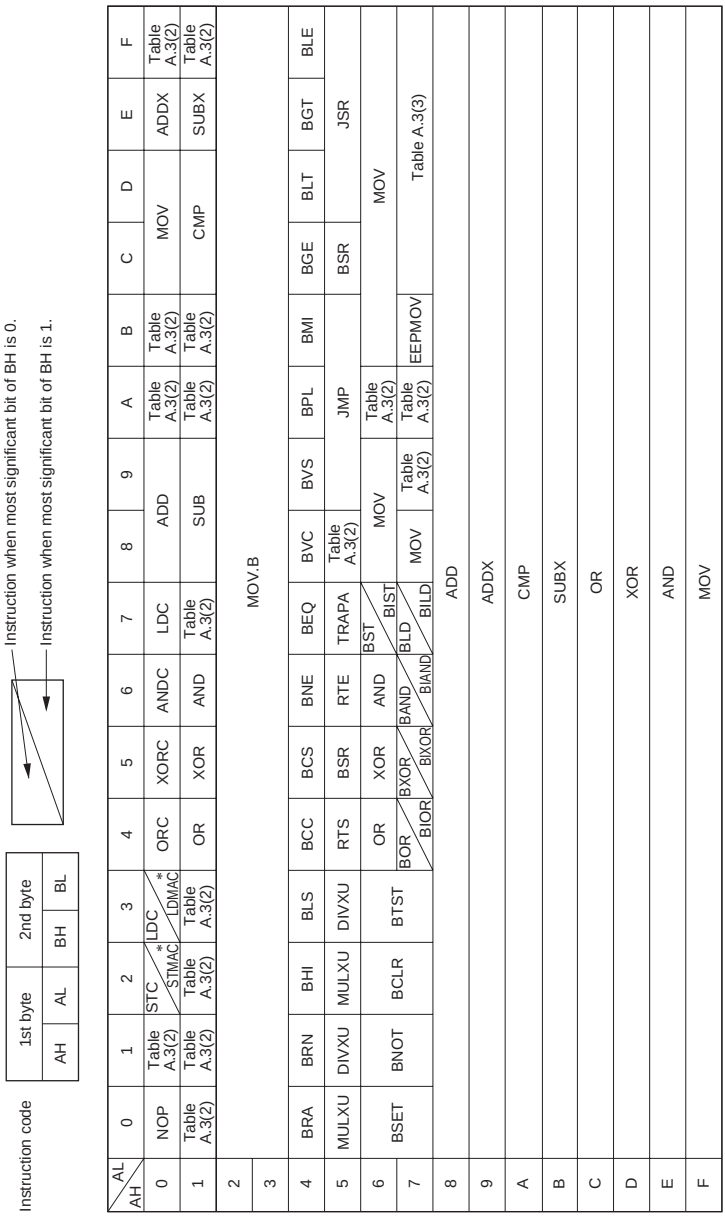
IMM:	Immediate data (2, 3, 8, 16, or 32 bits)
abs:	Absolute address (8, 16, 24, or 32 bits)
disp:	Displacement (8, 16, or 32 bits)
rs, rd, rn:	Register field (4 bits specifying an 8-bit or 16-bit register. The symbols rs, rd, and rn correspond to operand symbols Rs, Rd,an
ers, erd, ern, erm:	Register field (3 bits specifying an address register or 32-bit register. The symbols ers, erd, ern, and erm correspond to operan 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

Table A.3 shows the operation code map.

Table A.3 Operation Code Map (1)



Note: * Cannot be used in the chip.

Instruction code

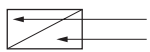
1st byte		2nd byte	
AH	AL	BH	BL

AH AL \ BH	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
01	MOV	LDM / STM			LDC / STC		MAC*		SLEEP		CLRMAC*		Table A.3(3)	Table A.3(3)	TAS	Table A.3(3)
0A	INC								ADD							
0B	ADDS					INC		INC	ADDS					INC		INC
0F	DAA								MOV							
10	SHLL			SHLL				SHLL	SHAL			SHAL				SHAL
11	SHLR			SHLR				SHLR	SHAR			SHAR				SHAR
12	ROTXL			ROTXL				ROTXL	ROTL			ROTL				ROTL
13	ROTXR			ROTXR				ROTXR	ROTR			ROTR				ROTR
17	NOT			NOT		EXTU		EXTU	NEG			NEG		EXTS		EXTS
1A	DEC								SUB							
1B	SUBS					DEC		DEC	SUBS					DEC		DEC
1F	DAS								CMP							
58	BRA	BRN	BHI	BLS	BCC	BCS	BNE	BEQ	BVC	BVS	BPL	BMI	BGE	BLT	BGT	BLE
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									

Note: * Cannot be used in the chip.

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	D	E	F
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 when most significant bit of FH is
Instruction when most significant bit of FH is

EL / AHALBHBLCHCLDHDLEH	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
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 byte	
	AH	AL	BH	BL	CH	CL	DH	DL	EH	EL	FH	FL	GH	GL	HH	HL

Instruction when most significant bit of HH is
Instruction when most significant bit of HH is

GL / AHALBHBL ... FHFLGH	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
6A30aaaaaaaa6*				BTST												
6A30aaaaaaaa7*					BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD								
6A38aaaaaaaa6*	BSET	BNOT	BCLR					BST / BIST								
6A38aaaaaaaa7*																

Note: * aa is the absolute address specification.

The tables in this section can be used to calculate the number of states required for instruction execution by the CPU. Table A.5 indicates the number of instruction fetch, data read/write, and other cycles occurring in each instruction. Table A.4 indicates the number of states required for each cycle. The number of states required for execution of an instruction can be calculated from these two tables as follows:

$$\text{Execution states} = I \times S_I + J \times S_J + K \times S_K + L \times S_L + M \times S_M + N \times S_N$$

Examples: Advanced mode, program code and stack located in external memory, on-chip supporting modules accessed in two states with 8-bit bus width, external devices accessed in three 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 = $2 \times 4 + 2 \times 4 + 2 \times 4 = 24$

		Access Conditions					
		On-Chip Supporting Module		External Device			
				8-Bit Bus		16-Bit Bus	
Cycle		On-Chip Memory	8-Bit Bus	16-Bit Bus	2-State Access	3-State Access	2-State Access	3-State Access
Instruction fetch	S_I	1	4	2	4	6 + 2m	2	3 + m
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	1

Legend:

m: Number of wait states inserted into external device access

Instruction	Mnemonic	Instruction Fetch I	Branch Address Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
ADD	ADD.B #xx:8,Rd	1					
	ADD.B Rs,Rd	1					
	ADD.W #xx:16,Rd	2					
	ADD.W Rs,Rd	1					
	ADD.L #xx:32,ERd	3					
	ADD.L ERs,ERd	1					
ADDS	ADDS #1/2/4,ERd	1					
ADDX	ADDX #xx:8,Rd	1					
	ADDX Rs,Rd	1					
AND	AND.B #xx:8,Rd	1					
	AND.B Rs,Rd	1					
	AND.W #xx:16,Rd	2					
	AND.W Rs,Rd	1					
	AND.L #xx:32,ERd	3					
	AND.L ERs,ERd	2					
ANDC	ANDC #xx:8,CCR	1					
	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					

Instruction	Mnemonic	Fetch I	Read J	Operation K	Access L	Access M	Operation N
Bcc	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					
	BRA d:16 (BT d:16)	2					1
	BRN d:16 (BF d:16)	2					1
	BHI d:16	2					1
	BLS d:16	2					1
	BCC d:16 (BHS d:16)	2					1
	BCS d:16 (BLO d:16)	2					1
	BNE d:16	2					1
	BEQ d:16	2					1
	BVC d:16	2					1
	BVS d:16	2					1
	BPL d:16	2					1
	BMI d:16	2					1
	BGE d:16	2					1
	BLT d:16	2					1
	BGT d:16	2					1
	BLE d:16	2					1
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		

Instruction	Mnemonic	Fetch I	Read J	Operation K	Access L	Access M	Operation N
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		
BIOR	BIOR #xx:8,Rd	1					
	BIOR #xx:8,@ERd	2			1		
	BIOR #xx:8,@aa:8	2			1		
	BIOR #xx:8,@aa:16	3			1		
	BIOR #xx:8,@aa:32	4			1		
BIST	BIST #xx:3,Rd	1					
	BIST #xx:3,@ERd	2			2		
	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		

Instruction	Mnemonic	I	J	K	L	M	N
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	BSET #xx:3,Rd	1					
	BSET #xx:3,@ERd	2			2		
	BSET #xx:3,@aa:8	2			2		
	BSET #xx:3,@aa:16	3			2		
	BSET #xx:3,@aa:32	4			2		
	BSET Rn,Rd	1					
	BSET Rn,@ERd	2			2		
	BSET Rn,@aa:8	2			2		
	BSET Rn,@aa:16	3			2		
	BSET Rn,@aa:32	4			2		
BSR	BSR d:8 Advanced	2		2			
	BSR d:16 Advanced	2		2			1
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		

Instruction	Mnemonic	I	J	K	L	M	N
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	Cannot be used in the chip					
CMP	CMP.B #xx:8,Rd	1					
	CMP.B Rs,Rd	1					
	CMP.W #xx:16,Rd	2					
	CMP.W Rs,Rd	1					
	CMP.L #xx:32,ERd	3					
	CMP.L ERs,ERd	1					
DAA	DAA Rd	1					
DAS	DAS Rd	1					
DEC	DEC.B Rd	1					
	DEC.W #1/2,Rd	1					
	DEC.L #1/2,ERd	1					
DIVXS	DIVXS.B Rs,Rd	2					11
	DIVXS.W Rs,ERd	2					19
DIVXU	DIVXU.B Rs,Rd	1					11
	DIVXU.W Rs,ERd	1					19

Instruction	Mnemonic	I	J	K	L	M	N
		Fetch	Read	Operation	Access	Access	Operation
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					1
	JMP @@aa:8 Advanced	2	2				1
JSR	JSR @ERn Advanced	2		2			
	JSR @aa:24 Advanced	2		2			1
	JSR @@aa:8 Advanced	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	1
	LDC @ERs+,EXR	2				1	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	

Instruction	Mnemonic	Fetch I	Read J	Operation K	Access L	Access M	Operation N
LDM	LDM.L @SP+, (ERn-ERn+1)	2		4			1
	LDM.L @SP+, (ERn-ERn+2)	2		6			1
	LDM.L @SP+, (ERn-ERn+3)	2		8			1
LDMAC	LDMAC ERs,MACH	Cannot be used in the chip					
	LDMAC ERs,MACL						
MAC	MAC @ERn+,@ERm+	Cannot be used in the chip					
MOV	MOV.B #xx:8,Rd	1					
	MOV.B Rs,Rd	1					
	MOV.B @ERs,Rd	1			1		
	MOV.B @(d:16,ERs),Rd	2			1		
	MOV.B @(d:32,ERs),Rd	4			1		
	MOV.B @ERs+,Rd	1			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		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	1
	MOV.W @aa:16,Rd	2				1	
	MOV.W @aa:32,Rd	3				1	
	MOV.W Rs,@ERd	1				1	

Instruction	Mnemonic	Fetch I	Read J	Operation K	Access L	Access M	Operation N
MOV	MOV.W Rs,@(d:16,ERd)	2				1	
	MOV.W Rs,@(d:32,ERd)	4				1	
	MOV.W Rs,@-ERd	1				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 @(d:16,ERs),ERd	3				2	
	MOV.L @(d:32,ERs),ERd	5				2	
	MOV.L @ERs+,ERd	2				2	1
	MOV.L @aa:16,ERd	3				2	
	MOV.L @aa:32,ERd	4				2	
	MOV.L ERs,@ERd	2				2	
	MOV.L ERs,@(d:16,ERd)	3				2	
	MOV.L ERs,@(d:32,ERd)	5				2	
	MOV.L ERs,@-ERd	2				2	1
	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 chip					
MOVTPE	MOVTPE Rs,@:aa:16						
MULXS	MULXS.B Rs,Rd	2					11
	MULXS.W Rs,ERd	2					19
MULXU	MULXU.B Rs,Rd	1					11
	MULXU.W Rs,ERd	1					19
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					

Instruction	Mnemonic	Fetch I	Read J	Operation K	Access L	Access M	Operation N
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	1
	POP.L ERn	2				2	1
PUSH	PUSH.W Rn	1				1	1
	PUSH.L ERn	2				2	1
ROTL	ROTL.B Rd	1					
	ROTL.B #2,Rd	1					
	ROTL.W Rd	1					
	ROTL.W #2,Rd	1					
	ROTL.L ERd	1					
	ROTL.L #2,ERd	1					
ROTR	ROTR.B Rd	1					
	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					

Instruction	Mnemonic	Fetch I	Read J	Operation K	Access L	Access M	Operation N
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			1
RTS	RTS Advanced	2		2			1
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					
SHLL	SHLL.B Rd	1					
	SHLL.B #2,Rd	1					
	SHLL.W Rd	1					
	SHLL.W #2,Rd	1					
	SHLL.L ERd	1					
	SHLL.L #2,ERd	1					
SHLR	SHLR.B Rd	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					1

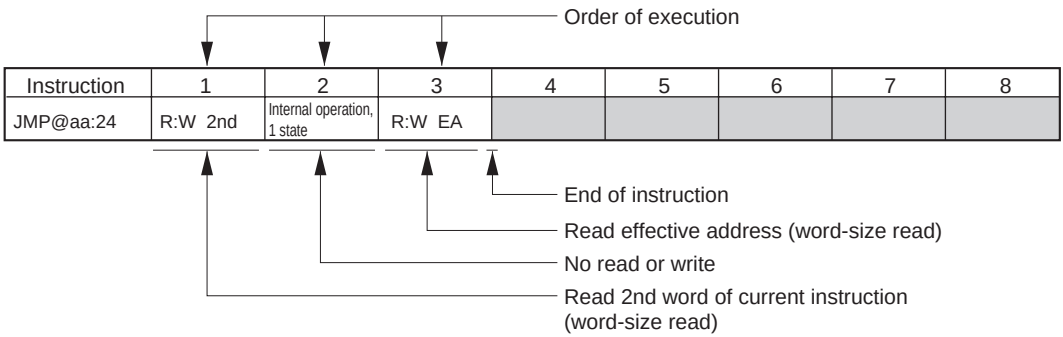
Instruction	Mnemonic	Fetch I	Read J	Operation K	Access L	Access M	Operation N
STC	STC.B CCR,Rd	1					
	STC.B EXR,Rd	1					
	STC.W CCR,@ERd	2				1	
	STC.W EXR,@ERd	2				1	
	STC.W CCR,@(d:16,ERd)	3				1	
	STC.W EXR,@(d:16,ERd)	3				1	
	STC.W CCR,@(d:32,ERd)	5				1	
	STC.W EXR,@(d:32,ERd)	5				1	
	STC.W CCR,@-ERd	2				1	1
	STC.W EXR,@-ERd	2				1	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			1
	STM.L (ERn-ERn+2), @-SP	2		6			1
	STM.L (ERn-ERn+3), @-SP	2		8			1
STMAC	STMAC MACH,ERd	Cannot be used in the chip					
	STMAC MACL,ERd						
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					
TAS	TAS @ERd*3	2			2		
TRAPA	TRAPA #x:2 Advanced	2	2	2/3*1			2

Instruction	Mnemonic	Fetch I	Read J	Operation K	Access L	Access M	Operation N
XOR	XOR.B #xx:8,Rd	1					
	XOR.B Rs,Rd	1					
	XOR.W #xx:16,Rd	2					
	XOR.W Rs,Rd	1					
	XOR.L #xx:32,ERd	3					
	XOR.L ERs,ERd	2					
XORC	XORC #xx:8,CCR	1					
	XORC #xx:8,EXR	2					

Notes: 1. 2 when EXR is invalid, 3 when EXR is valid.
2. When n bytes of data are transferred.
3. Only register ER0, ER1, ER4, or ER5 should be used when using the TAS instruction.

Table A.6 indicates the types of cycles that occur during instruction execution by the CPU. See table A.4 for the number of states per cycle.

How to Read the Table:



Legend

R:B	Byte-size read
R:W	Word-size read
W:B	Byte-size write
W:W	Word-size write
:M	Transfer of the bus is not performed immediately after this cycle
2nd	Address of 2nd word (3rd and 4th bytes)
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

states.

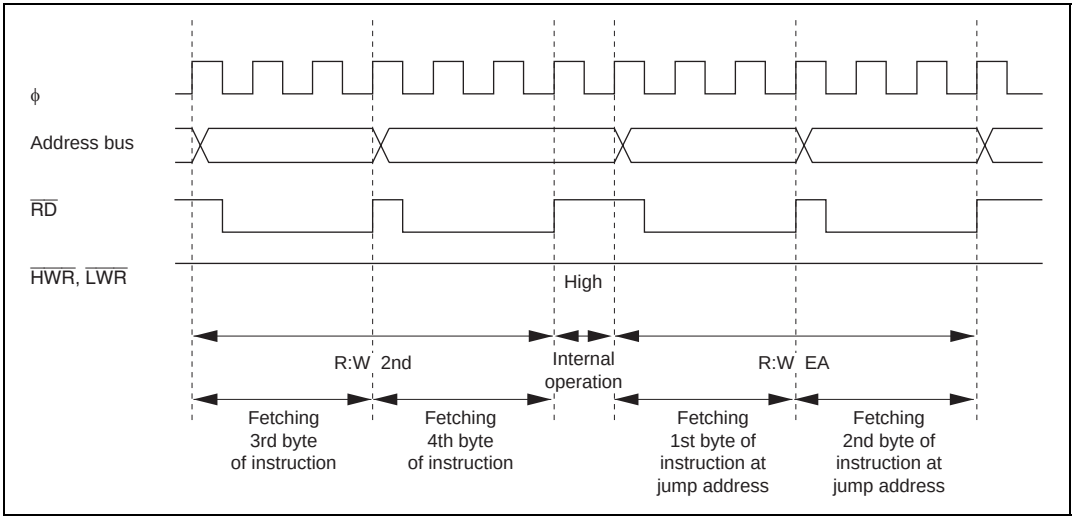


Figure A.1 Address Bus, $\overline{RD}$, $\overline{HWR}$, and $\overline{LWR}$ Timing (8-Bit Bus, Three-State Access, No Wait States)

Instruction	1	2	3	4	5	6	7	8	9
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							
BGT d:8	R:W NEXT	R:W EA							

Instruction	1	2	3	4	5	6	7	8	9
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					
BCLR #xx:3,@aa:8	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BCLR #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B:M EA	R:W:M NEXT	W:B EA				

Instruction	1	2	3	4	5	6	7	8	9
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					
BLD #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT				
BNOT #xx:3,Rd	R:W NEXT								

Instruction	1	2	3	4	5	6	7	8	9
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 Advanced	R:W NEXT	R:W EA	W:W:M stack (H)	W:W stack (L)					
BSR d:16 Advanced	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	8	9
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	Cannot be used in the chip								
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								
EXTU.L ERd	R:W NEXT								
INC.B Rd	R:W NEXT								

Instruction		1	2	3	4	5	6	7	8	9
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	Advanced	R:W NEXT	R:W:M aa:8	R:W aa:8	Internal operation, 1 state	R:W EA				
JSR @ERn	Advanced	R:W NEXT	R:W EA	W:W:M stack (H)	W:W stack (L)					
JSR @aa:24	Advanced	R:W 2nd	Internal operation, 1 state	R:W EA	W:W:M stack (H)	W:W stack (L)				
JSR @@aa:8	Advanced	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	8	9
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	Cannot be used in the chip								
LDMAC ERs,MACL									
MAC @ERn+,@ERm+									
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						
MOV.W @aa:32,Rd	R:W 2nd	R:W 3rd	R:W NEXT	R:B EA					
MOV.W Rs,@ERd	R:W NEXT	W:W EA							

Instruction	1	2	3	4	5	6	7	8	9
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 chip								
MOVTPE Rs,@aa:16									
MULXS.B Rs,Rd	R:W 2nd	R:W NEXT	Internal operation, 11 states						
MULXS.W Rs,ERd	R:W 2nd	R:W NEXT	Internal operation, 19 states						
MULXU.B Rs,Rd	R:W NEXT	Internal operation, 11 states							
MULXU.W Rs,ERd	R:W NEXT	Internal operation, 19 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								
OR.B #xx:8,Rd	R:W NEXT								
OR.B Rs,Rd	R:W NEXT								

Instruction	1	2	3	4	5	6	7	8	9
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								
ROTXR.W #2,Rd	R:W NEXT								
ROTXR.L ERd	R:W NEXT								

Instruction		1	2	3	4	5	6	7	8	9
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	Advanced	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	8	9
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	Cannot be used in the chip								
STMAC MACL,ERd									
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 Advanced	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+2	Internal operation, 1 state	R:W*7
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						

Instruction		1	2	3	4	5	6	7	8	9
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 handling	Advanced	R:W VEC	R:W VEC+2	Internal operation, 1 state	R:W*5					
Interrupt exception handling	Advanced	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+2	Internal operation, 1 state	R:W*7

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 of the instruction. n is the 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 registers.
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 software standby mode the re 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.

This section indicates the effect of each CPU instruction on the condition code. The notation used in the table is defined below.

m = { 31 for longword operands; 15 for word operands; 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

Instruction	H	N	Z	V	C	Definition
ADD	↕	↕	↕	↕	↕	H = Sm–4 · Dm–4 + Dm–4 · $\overline{Rm–4}$ + Sm–4 · $\overline{Rm–4}$ N = Rm Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · R0 V = Sm · Dm · $\overline{Rm}$ + $\overline{Sm}$ · $\overline{Dm}$ · Rm C = Sm · Dm + Dm · $\overline{Rm}$ + Sm · $\overline{Rm}$
ADDS	—	—	—	—	—	
ADDX	↕	↕	↕	↕	↕	H = Sm–4 · Dm–4 + Dm–4 · $\overline{Rm–4}$ + Sm–4 · $\overline{Rm–4}$ N = Rm Z = Z' · $\overline{Rm}$ · ····· · $\overline{R0}$ V = Sm · Dm · Rm + $\overline{Sm}$ · $\overline{Dm}$ · Rm C = Sm · Dm + Dm · 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

CMP	↕	↕	↕	↕	↕	H = Sm–4 · $\overline{Dm–4}$ + $\overline{Dm–4}$ · Rm–4 + Sm–4 · Rm–4 N = Rm Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$ V = $\overline{Sm}$ · Dm · $\overline{Rm}$ + Sm · $\overline{Dm}$ · Rm C = Sm · $\overline{Dm}$ + $\overline{Dm}$ · Rm + Sm · Rm
DAA	*	↕	↕	*	↕	N = Rm Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$ C: decimal arithmetic carry
DAS	*	↕	↕	*	↕	N = Rm Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$ C: decimal arithmetic borrow
DEC	—	↕	↕	↕	—	N = Rm Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$ V = Dm · $\overline{Rm}$
DIVXS	—	↕	↕	—	—	N = Sm · $\overline{Dm}$ + $\overline{Sm}$ · Dm Z = $\overline{Sm}$ · $\overline{Sm–1}$ · ····· · $\overline{S0}$
DIVXU	—	↕	↕	—	—	N = Sm Z = $\overline{Sm}$ · $\overline{Sm–1}$ · ····· · $\overline{S0}$
EEPMOV	—	—	—	—	—	
EXTS	—	↕	↕	0	—	N = Rm Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$
EXTU	—	0	↕	0	—	Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$
INC	—	↕	↕	↕	—	N = Rm Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$ V = $\overline{Dm}$ · Rm
JMP	—	—	—	—	—	
JSR	—	—	—	—	—	
LDC	↕	↕	↕	↕	↕	Stores the corresponding bits of the result. No flags change when the operand is EXR.
LDM	—	—	—	—	—	
LDMAC						Cannot be used in the chip
MAC						

						Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$
MOVFPE						Cannot be used in the chip
MOVTPE						
MULXS	—	↕	↕	—	—	N = R2m Z = $\overline{R2m} \cdot \overline{R2m-1} \cdot \cdots\cdots \cdot \overline{R0}$
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)

						$Z = \overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ C = Dm (1-bit shift) or C = Dm–1 (2-bit shift)
ROTXR	—	↕	↕	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-bit shift) 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						Cannot be used in the chip

						N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ V = $\overline{Sm} \cdot Dm \cdot \overline{Rm} + Sm \cdot \overline{Dm} \cdot Rm$ C = $Sm \cdot \overline{Dm} + \overline{Dm} \cdot Rm + Sm \cdot Rm$
SUBS	—	—	—	—	—	
SUBX	↕	↕	↕	↕	↕	H = $Sm-4 \cdot \overline{Dm-4} + \overline{Dm-4} \cdot Rm-4 + Sm-4 \cdot Rm-4$ N = Rm Z = Z' · $\overline{Rm} \cdot \cdots\cdots \cdot \overline{R0}$ V = $\overline{Sm} \cdot Dm \cdot \overline{Rm} + Sm \cdot \overline{Dm} \cdot Rm$ C = $Sm \cdot \overline{Dm} + \overline{Dm} \cdot Rm + Sm \cdot Rm$
TAS	—	↕	↕	0	—	N = Dm Z = $\overline{Dm} \cdot \overline{Dm-1} \cdot \cdots\cdots \cdot \overline{D0}$
TRAPA	—	—	—	—	—	
XOR	—	↕	↕	0	—	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$
XORC	↕	↕	↕	↕	↕	Stores the corresponding bits of the result. No flags change when the operand is EXR.

B.1 List of Registers (Address Order)

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Data Bus Width
H'F800 to H'FBFF	MRA	SM1	SM0	DM1	DM0	MD1	MD0	DTS	Sz	DTC	16/ 32*1 bits
	SAR										
	MRB	CHNE	DISEL	CHNS	—	—	—	—	—		
	DAR										
	CRA										
	CRB										
H'FE80	TCR3	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU3	16 bits
H'FE81	TMDR3	—	—	BFB	BFA	MD3	MD2	MD1	MD0		
H'FE82	TIOR3H	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0		
H'FE83	TIOR3L	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0		
H'FE84	TIER3	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA		
H'FE85	TSR3	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA		
H'FE86	TCNT3										
H'FE87											
H'FE88	TGR3A										
H'FE89											
H'FE8A	TGR3B										
H'FE8B											
H'FE8C	TGR3C										
H'FE8D											
H'FE8E	TGR3D										
H'FE8F											

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Name	Width
H'FE90	TCR4	—	CCLR1	CCLR0	CKEG	CKEGO	TPSC2	TPSC1	TPSC0	TPU4	16 bits
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	TPU5	16 bits
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'FEB0	P1DDR	P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DDR	Ports	8 bits
H'FEB1	P2DDR	P27DDR	P26DDR	P25DDR	P24DDR	P23DDR	P22DDR	P21DDR	P20DDR		
H'FEB2	P3DDR	—	—	P35DDR	P34DDR	P33DDR	P32DDR	P31DDR	P30DDR		
H'FEB4	P5DDR	—	—	—	—	P53DDR	P52DDR	P51DDR	P50DDR		
H'FEB5	P6DDR	P67DDR	P66DDR	P65DDR	P64DDR	P63DDR	P62DDR	P61DDR	P60DDR		
H'FEB6	P7DDR	—	—	P75DDR	P74DDR	P73DDR	P72DDR	P71DDR	P70DDR		
H'FEB7	P8DDR	—	P86DDR	P85DDR	P84DDR	P83DDR	P82DDR	P81DDR	P80DDR		
H'FEB8	P9DDR	P97DDR	P96DDR	P95DDR	P94DDR	P93DDR	P92DDR	—	—		

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Name	Width
H'FEB9	PADDR	PA7DDR	PA6DDR	PA5DDR	PA4DDR	PA3DDR	PA2DDR	PA1DDR	PA0DDR	Ports	8 bits
H'FEBA	PBDDR	PB7DDR	PB6DDR	PB5DDR	PB4DDR	PB3DDR	PB2DDR	PB1DDR	PB0DDR		
H'FEBB	PCDDR	PC7DDR	PC6DDR	PC5DDR	PC4DDR	PC3DDR	PC2DDR	PC1DDR	PC0DDR		
H'FEBC	PDDDR	PD7DDR	PD6DDR	PD5DDR	PD4DDR	PD3DDR	PD2DDR	PD1DDR	PD0DDR		
H'FEBD	PEDDR	PE7DDR	PE6DDR	PE5DDR	PE4DDR	PE3DDR	PE2DDR	PE1DDR	PE0DDR		
H'FEBE	PFDDR	PF7DDR	PF6DDR	PF5DDR	PF4DDR	PF3DDR	PF2DDR	PF1DDR	PF0DDR		
H'FEBF	PGDDR	—	—	—	PG4DDR	PG3DDR	PG2DDR	PG1DDR	PG0DDR		
H'FEC4	IPRA	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	Interrupt controller	8 bits
H'FEC5	IPRB	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0		
H'FEC6	IPRC	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0		
H'FEC7	IPRD	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0		
H'FEC8	IPRE	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0		
H'FEC9	IPRF	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0		
H'FECA	IPRG	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0		
H'FECB	IPRH	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0		
H'FECC	IPRI	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0		
H'FECD	IPRJ	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0		
H'FECE	IPRK	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0		
H'FED0	ABWCR	ABW7	ABW6	ABW5	ABW4	ABW3	ABW2	ABW1	ABW0	Bus controller	8 bits
H'FED1	ASTCR	AST7	AST6	AST5	AST4	AST3	AST2	AST1	AST0		
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	RMTS2	RMTS1	RMTS0		
H'FED5	BCRL	BRLE	BREQOE	EAE	—	DDS	—	WDBE	WAITE		
H'FED6	MCR	TPC	BE	RCDM	—	MXC1	MXC0	RLW1	RLW0		
H'FED7	DRAMCR	RFSHE	RCW	RMODE	CMF	CMIE	CKS2	CKS1	CKS0		
H'FED8	RTCNT										
H'FED9	RTCOR										
H'FEDB	RAMER*2	—	—	—	—	RAMS	RAM2	RAM1	RAM0	Flash memory	16 bits

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Name	Width
H'FEE0	MAR0AH	—	—	—	—	—	—	—	—	DMAC	16 bits
H'FEE1											
H'FEE2	MAR0AL										
H'FEE3											
H'FEE4	IOR0A										
H'FEE5											
H'FEE6	ETCR0A										
H'FEE7											
H'FEE8	MAR0BH	—	—	—	—	—	—	—	—		
H'FEE9											
H'FEEA	MAR0BL										
H'FEEB											
H'FEEC	IOR0B										
H'FEED											
H'FEEE	ETCR0B										
H'FEEF											
H'FEF0	MAR1AH	—	—	—	—	—	—	—	—		
H'FEF1											
H'FEF2	MAR1AL										
H'FEF3											
H'FEF4	IOAR1A										
H'FEF5											
H'FEF6	ETCR1A										
H'FEF7											
H'FEF8	MAR1BH	—	—	—	—	—	—	—	—		
H'FEF9											
H'FEFA	MAR1BL										
H'FEFB											
H'FEFC	IOAR1B										
H'FEFD											
H'FEFE	ETCR1B										
H'FEFF											

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Name	Width
H'FF00	DMAWER	—	—	—	—	WE1B	WE1A	WE0B	WE0A	DMAC	8 bits
H'FF01	DMATCR	—	—	TEE1	TEE0	—	—	—	—		
H'FF02	DMACR0A	DTSZ	DTID	RPE	DTDIR	DTF3	DTF2	DTF1	DTF0	Short address mode	16 bits
		DTSZ	SAID	SAIDE	BLKDIR	BLKE	—	—	—	Full address mode	
H'FF03	DMACR0B	DTSZ	DTID	RPE	DTDIR	DTF3	DTF2	DTF1	DTF0	Short address mode	
		—	DAID	DAIDE	—	DTF3	DTF2	DTF1	DTF0	Full address mode	
H'FF04	DMACR1A	DTSZ	DTID	RPE	DTDIR	DTF3	DTF2	DTF1	DTF0	Short address mode	
		DTSZ	SAID	SAIDE	BLKDIR	BLKE	—	—	—	Full address mode	
H'FF05	DMACR1B	DTSZ	DTID	RPE	DTDIR	DTF3	DTF2	DTF1	DTF0	Short address mode	
		—	DAID	DAIDE	—	DTF3	DTF2	DTF1	DTF0	Full address mode	
H'FF06	DMABCRH	FAE1	FAE0	SAE1	SAE0	DTA1B	DTA1A	DTA0B	DTA0A	Short address mode	
		FAE1	FAE0	—	—	DTA1	—	DTA0	—	Full address mode	
H'FF07	DMABCRL	DTE1B	DTE1A	DTE0B	DTE0A	DTIE1B	DTIE1A	DTIE0B	DTIE0A	Short address mode	
		DTME1	DTE1	DTME0	DTE0	DTIE1B	DTIE1A	DTIE0B	DTIE0A	Full address mode	

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Name	Width
H'FF2C	ISCRH	IRQ7SCB	IRQ7SCA	IRQ6SCB	IRQ6SCA	IRQ5SCB	IRQ5SCA	IRQ4SCB	IRQ4SCA	Interrupt controller	8 bits
H'FF2D	ISCRL	IRQ3SCB	IRQ3SCA	IRQ2SCB	IRQ2SCA	IRQ1SCB	IRQ1SCA	IRQ0SCB	IRQ0SCA		
H'FF2E	IER	IRQ7E	IRQ6E	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IRQ0E		
H'FF2F	ISR	IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F		
H'FF30 to H'FF35	DTCER	DTCE7	DTCE6	DTCE5	DTCE4	DTCE3	DTCE2	DTCE1	DTCE0	DTC	8 bits
H'FF37	DTVECR	SWDTE	DTVEC6	DTVEC5	DTVEC4	DTVEC3	DTVEC2	DTVEC1	DTVEC0		
H'FF38	SBYCR	SSBY	STS2	STS1	STS0	OPE	—	—	IRQ37S	Power-down mode	8 bits
H'FF39	SYSCR	—	—	INTM1	INTM0	NMIEG	LWROD	IRQPAS	RAME	MCU	8 bits
H'FF3A	SCKCR	PSTOP	—	DIV	—	—	SCK2	SCK1	SCK0	Clock pulse generator	8 bits
H'FF3B	MDCR	—	—	—	—	—	MDS2	MDS1	MDS0	MCU	8 bits
H'FF3C	MSTPCRH	MSTP15	MSTP14	MSTP13	MSTP12	MSTP11	MSTP10	MSTP9	MSTP8	Power-down mode	8 bits
H'FF3D	MSTPCRL	MSTP7	MSTP6	MSTP5	MSTP4	MSTP3	MSTP2	MSTP1	MSTP0		
H'FF42	SYSCR*2	—	—	—	—	FLSHE	—	—	—	MCU	8 bits
H'FF44	Reserved	—	—	—	—	—	—	—	—	Reserved	—
H'FF45	PFCR1	—	—	—	—	A23E	A22E	A21E	A20E	Port	8 bits
H'FF46	PCR	G3CMS1	G3CMS0	G2CMS1	G2CMS0	G1CMS1	G1CMS0	G0CMS1	G0CMS0	PPG	8 bits
H'FF47	PMR	G3INV	G2INV	G1INV	G0INV	G3NOV	G2NOV	G1NOV	G0NOV		
H'FF48	NDERH	NDER15	NDER14	NDER13	NDER12	NDER11	NDER10	NDER9	NDER8		
H'FF49	NDERL	NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1	NDER0		
H'FF4A	PODRH	POD15	POD14	POD13	POD12	POD11	POD10	POD9	POD8		
H'FF4B	PODRL	POD7	POD6	POD5	POD4	POD3	POD2	POD1	POD0		
H'FF4C*3	NDRH	NDR15	NDR14	NDR13	NDR12	NDR11	NDR10	NDR9	NDR8		
H'FF4D*3	NDRL	NDR7	NDR6	NDR5	NDR4	NDR3	NDR2	NDR1	NDR0		
H'FF4E*3	NDRH	—	—	—	—	NDR11	NDR10	NDR9	NDR8		
H'FF4F*3	NDRL	—	—	—	—	NDR3	NDR2	NDR1	NDR0		

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Name	Width
H'FF50	PORT1	P17	P16	P15	P14	P13	P12	P11	P10	Ports	8 bits
H'FF51	PORT2	P27	P26	P25	P24	P23	P22	P21	P20		
H'FF52	PORT3	—	—	P35	P34	P33	P32	P31	P30		
H'FF53	PORT4	P47	P46	P45	P44	P43	P42	P41	P40		
H'FF54	PORT5	P57	P56	P55	P54	P53	P52	P51	P50		
H'FF55	PORT6	P67	P66	P65	P64	P63	P62	P61	P60		
H'FF56	PORT7	—	—	P75	P74	P73	P72	P71	P70		
H'FF57	PORT8	—	P86	P85	P84	P83	P82	P81	P80		
H'FF58	PORT9	P97	P96	P95	P94	P93	P92	—	—		
H'FF59	PORTA	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0		
H'FF5A	PORTB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0		
H'FF5B	PORTC	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0		
H'FF5C	PORTD	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0		
H'FF5D	PORTE	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0		
H'FF5E	PORTF	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0		
H'FF5F	PORTG	—	—	—	PG4	PG3	PG2	PG1	PG0		
H'FF60	P1DR	P17DR	P16DR	P15DR	P14DR	P13DR	P12DR	P11DR	P10DR		
H'FF61	P2DR	P27DR	P26DR	P25DR	P24DR	P23DR	P22DR	P21DR	P20DR		
H'FF62	P3DR	—	—	P35DR	P34DR	P33DR	P32DR	P31DR	P30DR		
H'FF64	P5DR	—	—	—	—	P53DR	P52DR	P51DR	P50DR		
H'FF65	P6DR	P67DR	P66DR	P65DR	P64DR	P63DR	P62DR	P61DR	P60DR		
H'FF66	P7DR	—	—	P75DR	P74DR	P73DR	P72DR	P71DR	P70DR		
H'FF67	P8DR	—	P86DR	P85DR	P84DR	P83DR	P82DR	P81DR	P80DR		
H'FF68	P9DR	P97DR	P96DR	P95DR	P94DR	P93DR	P92DR	—	—		
H'FF69	PADR	PA7DR	PA6DR	PA5DR	PA4DR	PA3DR	PA2DR	PA1DR	PA0DR		
H'FF6A	PBDR	PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR		
H'FF6B	PCDR	PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR		
H'FF6C	PDDR	PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR		
H'FF6D	PEDR	PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR		
H'FF6E	PFDR	PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1DR	PF0DR		
H'FF6F	PGDR	—	—	—	PG4DR	PG3DR	PG2DR	PG1DR	PG0DR		
H'FF70	PAPCR	PA7PCR	PA6PCR	PA5PCR	PA4PCR	PA3PCR	PA2PCR	PA1PCR	PA0PCR		

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Name	Width
H'FF71	PBPCR	PB7PCR	PB6PCR	PB5PCR	PB4PCR	PB3PCR	PB2PCR	PB1PCR	PB0PCR	Ports	8 bits
H'FF72	PCPCR	PC7PCR	PC6PCR	PC5PCR	PC4PCR	PC3PCR	PC2PCR	PC1PCR	PC0PCR		
H'FF73	PDPCR	PD7PCR	PD6PCR	PD5PCR	PD4PCR	PD3PCR	PD2PCR	PD1PCR	PD0PCR		
H'FF74	PEPCR	PE7PCR	PE6PCR	PE5PCR	PE4PCR	PE3PCR	PE2PCR	PE1PCR	PE0PCR		
H'FF76	P3ODR	—	—	P35ODR	P34ODR	P33ODR	P32ODR	P31ODR	P30ODR		
H'FF77	PAODR	PA7ODR	PA6ODR	PA5ODR	PA4ODR	PA3ODR	PA2ODR	PA1ODR	PA0ODR		
H'FF78	SMR0	C/$\overline{A}$/ GM*4	CHR/ BLK*5	PE	O/$\overline{E}$	STOP/ BCP1*6	MP/ BCP0*7	CKS1	CKS0	SCI0, smart card interface 0	8 bits
H'FF79	BRR0										
H'FF7A	SCR0	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0		
H'FF7B	TDR0										
H'FF7C	SSR0	TDRE	RDRF	ORER	FER/ ERS*8	PER	TEND	MPB	MPBT		
H'FF7D	RDR0										
H'FF7E	SCMR0	—	—	—	—	SDIR	SINV	—	SMIF		
H'FF80	SMR1	C/$\overline{A}$/ GM*4	CHR/ BLK*5	PE	O/$\overline{E}$	STOP/ BCP1*6	MP/ BCP0*7	CKS1	CKS0	SCI1, smart card interface 1	8 bits
H'FF81	BRR1										
H'FF82	SCR1	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0		
H'FF83	TDR1										
H'FF84	SSR1	TDRE	RDRF	ORER	FER/ ERS*8	PER	TEND	MPB	MPBT		
H'FF85	RDR1										
H'FF86	SCMR1	—	—	—	—	SDIR	SINV	—	SMIF		
H'FF88	SMR2	C/$\overline{A}$/ GM*4	CHR/ BLK*5	PE	O/$\overline{E}$	STOP/ BCP1*6	MP/ BCP0*7	CKS1	CKS0	SCI2, smart card interface 2	8 bits
H'FF89	BRR2										
H'FF8A	SCR2	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0		
H'FF8B	TDR2										
H'FF8C	SSR2	TDRE	RDRF	ORER	FER/ ERS*8	PER	TEND	MPB	MPBT		
H'FF8D	RDR2										
H'FF8E	SCMR2	—	—	—	—	SDIR	SINV	—	SMIF		

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Name	Width
H'FE90	ADDRAH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D converter	8 bits
H'FE91	ADDRAL	AD1	AD0	—	—	—	—	—	—		
H'FE92	ADDRBH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2		
H'FE93	ADDRBL	AD1	AD0	—	—	—	—	—	—		
H'FE94	ADDRCH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2		
H'FE95	ADDRCL	AD1	AD0	—	—	—	—	—	—		
H'FE96	ADDRDH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2		
H'FE97	ADDRDL	AD1	AD0	—	—	—	—	—	—		
H'FE98	ADCSR	ADF	ADIE	ADST	SCAN	CKS	CH2	CH1	CH0		
H'FE99	ADCR	TRGS1	TRGS0	—	—	CKS1	CH3	—	—		
H'FFA4	DADR0									D/A converter	8 bits
H'FFA5	DADR1										
H'FFA6	DACR01	DAOE1	DAOE0	DAE	—	—	—	—	—		
H'FFA8	DADR2										
H'FFA9	DADR3										
H'FFAA	DACR23	DAOE1	DAOE0	DAE	—	—	—	—	—		
H'FFAC	PFCR2	WAITPS	BREQOPS	CS167E	CS25E	ASOD	—	—	—	Ports	8 bits
H'FFB0	TCR0	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	8-bit timer channel 0, 1	16 bits
H'FFB1	TCR1	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0		
H'FFB2	TCSR0	CMFB	CMFA	OVF	ADTE	OS3	OS2	OS1	OS0		
H'FFB3	TCSR1	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0		
H'FFB4	TCORA0										
H'FFB5	TCORA1										
H'FFB6	TCORB0										
H'FFB7	TCORB1										
H'FFB8	TCNT0										
H'FFB9	TCNT1										
H'FFBC (Read)	TCSR	OVF	WT/$\overline{IT}$	TME	—	—	CKS2	CKS1	CKS0	WDT	16 bits
H'FFBD (Read)	TCNT										
H'FFBF (Read)	RSTCSR	WOVF	RSTE	—	—	—	—	—	—		

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Name	Width
H'FFC0	TSTR	—	—	CST5	CST4	CST3	CST2	CST1	CST0	TPU	16 bits
H'FFC1	TSYR	—	—	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0		
H'FFC8*10	FLMCR1	FWE	SWE	ESU	PSU	EV	PV	E	P	Flash memory (H8S/2338 F-ZTAT)	8 bits
H'FFC9*10	FLMCR2	FLER	—	—	—	—	—	—	—		
H'FFCA*10	EBR1	EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0		
H'FFCB*10	EBR2	—	—	—	—	EB11	EB10	EB9	EB8		
H'FFC8*9	FLMCR1	FWE	SWE	ESU	PSU	EV	PV	E	P	Flash memory (H8S/2339 F-ZTAT)	8 bits
H'FFC9*9	FLMCR2	FLER	—	—	—	—	—	—	—		
H'FFCA*9	EBR1	EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0		
H'FFCB*9	EBR2	—	—	EB13	EB12	EB11	EB10	EB9	EB8		
H'FFD0	TCR0	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU0	16 bits
H'FFD1	TMDR0	—	—	BFB	BFA	MD3	MD2	MD1	MD0		
H'FFD2	TIOR0H	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0		
H'FFD3	TIOR0L	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0		
H'FFD4	TIER0	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA		
H'FFD5	TSR0	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA		
H'FFD6	TCNT0										
H'FFD7											
H'FFD8	TGR0A										
H'FFD9											
H'FFDA	TGR0B										
H'FFDB											
H'FFDC	TGR0C										
H'FFDD											
H'FFDE	TGR0D										
H'FFDF											
H'FFE0	TCR1	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU1	16 bits
H'FFE1	TMDR1	—	—	—	—	MD3	MD2	MD1	MD0		
H'FFE2	TIOR1	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0		
H'FFE4	TIER1	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA		
H'FFE5	TSR1	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA		
H'FFE6	TCNT1										
H'FFE7											

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Name	Width
H'FFE8	TGR1A									TPU1	16 bits
H'FFE9											
H'FFEA	TGR1B										
H'FFEB											
H'FFF0	TCR2	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU2	16 bits
H'FFF1	TMDR2	—	—	—	—	MD3	MD2	MD1	MD0		
H'FFF2	TIOR2	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0		
H'FFF4	TIER2	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA		
H'FFF5	TSR2	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA		
H'FFF6	TCNT2										
H'FFF7											
H'FFF8	TGR2A										
H'FFF9											
H'FFFA	TGR2B										
H'FFFB											

Notes: 1. Located in on-chip RAM. The bus width is 32 bits when the DTC accesses this area as register information, and 16 bits otherwise.
2. Valid only in F-ZTAT version.
3. If the pulse output group 2 and pulse output group 3 output triggers are the same according to the PCR setting, the NDRH address will be H'FF4C, and if different, the address of NDRH for group 2 will be H'FF4E, and that for group 3 will be H'FF4C. Similarly, if the pulse output group 0 and pulse output group 1 output triggers are the same according to the PCR setting, the NDRL address will be H'FF4D, and if different, the address of NDRL for group 0 will be H'FF4F, and that for group 1 will be H'FF4D.
4. Functions as C/$\overline{A}$ for SCI use, and as GM for smart card interface use.
5. Functions as CHR for SCI use, and as BLK for smart card interface use.
6. Functions as STOP for SCI use, and as BCP1 for smart card interface use.
7. Functions as MP for SCI use, and as BCP0 for smart card interface use.
8. Functions as FER for SCI use, and as ERS for smart card interface use.
9. Valid only in H8S/2339 F-ZTAT.
10. Valid only in H8S/2338 F-ZTAT.

Module	Register	Abbreviation	R/W	Initial Value	Address*1
Interrupt controller	System control register	SYSCR	R/W	H'01	H'FF39
	IRQ sense control register H	ISCRH	R/W	H'00	H'FF2C
	IRQ sense control register L	ISCRL	R/W	H'00	H'FF2D
	IRQ enable register	IER	R/W	H'00	H'FF2E
	IRQ status register	ISR	R/(W)*2	H'00	H'FF2F
	Interrupt priority register A	IPRA	R/W	H'77	H'FEC4
	Interrupt priority register B	IPRB	R/W	H'77	H'FEC5
	Interrupt priority register C	IPRC	R/W	H'77	H'FEC6
	Interrupt priority register D	IPRD	R/W	H'77	H'FEC7
	Interrupt priority register E	IPRE	R/W	H'77	H'FEC8
	Interrupt priority register F	IPRF	R/W	H'77	H'FEC9
	Interrupt priority register G	IPRG	R/W	H'77	H'FECA
	Interrupt priority register H	IPRH	R/W	H'77	H'FECB
	Interrupt priority register I	IPRI	R/W	H'77	H'FECC
	Interrupt priority register J	IPRJ	R/W	H'77	H'FECD
	Interrupt priority register K	IPRK	R/W	H'77	H'FECE
Bus controller	Bus width control register	ABWCR	R/W	H'FF/H'00*5	H'FED0
	Access state control register	ASTCR	R/W	H'FF	H'FED1
	Wait control register H	WCRH	R/W	H'FF	H'FED2
	Wait control register L	WCRL	R/W	H'FF	H'FED3
	Bus control register H	BCRH	R/W	H'D0	H'FED4
	Bus control register L	BCRL	R/W	H'3C	H'FED5
	Memory control register	MCR	R/W	H'00	H'FED6
	DRAM control register	DRAMCR	R/W	H'00	H'FED7
	Refresh timer counter	RTCNT	R/W	H'00	H'FED8
	Refresh time constant register	RTCOR	R/W	H'FF	H'FED9

	DTC mode register B	MRB	—*3	Undefined	—*4
	DTC source address register	SAR	—*3	Undefined	—*4
	DTC destination address register	DAR	—*3	Undefined	—*4
	DTC transfer count register A	CRA	—*3	Undefined	—*4
	DTC transfer count register B	CRB	—*3	Undefined	—*4
	DTC enable register	DTCER	R/W	H'00	H'FF30 to H'FF35
	DTC vector register	DTVECR	R/W	H'00	H'FF37
	Module stop control register	MSTPCR	R/W	H'3FFF	H'FF3C
DMAC0	Memory address register 0A	MAR0A	R/W	Undefined	H'FEE0
	I/O address register 0A	IOAR0A	R/W	Undefined	H'FEE4
	Transfer count register 0A	ETCR0A	R/W	Undefined	H'FEE6
	Memory address register 0B	MAR0B	R/W	Undefined	H'FEE8
	I/O address register 0B	IOAR0B	R/W	Undefined	H'FEEC
	Transfer count register 0B	ETCR0B	R/W	Undefined	H'FEEE
DMAC1	Memory address register 1A	MAR1A	R/W	Undefined	H'FEF0
	I/O address register 1A	IOAR1A	R/W	Undefined	H'FEF4
	Transfer count register 1A	ETCR1A	R/W	Undefined	H'FEF6
	Memory address register 1B	MAR1B	R/W	Undefined	H'FEF8
	I/O address register 1B	IOAR1B	R/W	Undefined	H'FEFC
	Transfer count register 1B	ETCR1B	R/W	Undefined	H'FEFE
Both DMAC channels	DMA write enable register	DMAWER	R/W	H'00	H'FF00
	DMA terminal control register	DMATCR	R/W	H'00	H'FF01
	DMA control register 0A	DMACR0A	R/W	H'00	H'FF02
	DMA control register 0B	DMACR0B	R/W	H'00	H'FF03
	DMA control register 1A	DMACR1A	R/W	H'00	H'FF04
	DMA control register 1B	DMACR1B	R/W	H'00	H'FF05
	DMA band control register	DMABCR	R/W	H'0000	H'FF06
	Module stop control register	MSTPCR	R/W	H'3FFF	H'FF3C

	Timer mode register 0	TMDR0	R/W	H'C0	H'FFD1
	Timer I/O control register 0H	TIOR0H	R/W	H'00	H'FFD2
	Timer I/O control register 0L	TIOR0L	R/W	H'00	H'FFD3
	Timer interrupt enable register 0	TIER0	R/W	H'40	H'FFD4
	Timer status register 0	TSR0	R/(W)*2	H'C0	H'FFD5
	Timer counter 0	TCNT0	R/W	H'0000	H'FFD6
	Timer general register 0A	TGR0A	R/W	H'FFFF	H'FFD8
	Timer general register 0B	TGR0B	R/W	H'FFFF	H'FFDA
	Timer general register 0C	TGR0C	R/W	H'FFFF	H'FFDC
	Timer general register 0D	TGR0D	R/W	H'FFFF	H'FFDE
TPU1	Timer control register 1	TCR1	R/W	H'00	H'FFE0
	Timer mode register 1	TMDR1	R/W	H'C0	H'FFE1
	Timer I/O control register 1	TIOR1	R/W	H'00	H'FFE2
	Timer interrupt enable register 1	TIER1	R/W	H'40	H'FFE4
	Timer status register 1	TSR1	R/(W)*2	H'C0	H'FFE5
	Timer counter 1	TCNT1	R/W	H'0000	H'FFE6
	Timer general register 1A	TGR1A	R/W	H'FFFF	H'FFE8
	Timer general register 1B	TGR1B	R/W	H'FFFF	H'FFEA
TPU2	Timer control register 2	TCR2	R/W	H'00	H'FFF0
	Timer mode register 2	TMDR2	R/W	H'C0	H'FFF1
	Timer I/O control register 2	TIOR2	R/W	H'00	H'FFF2
	Timer interrupt enable register 2	TIER2	R/W	H'40	H'FFF4
	Timer status register 2	TSR2	R/(W)*2	H'C0	H'FFF5
	Timer counter 2	TCNT2	R/W	H'0000	H'FFF6
	Timer general register 2A	TGR2A	R/W	H'FFFF	H'FFF8
	Timer general register 2B	TGR2B	R/W	H'FFFF	H'FFFA

	Timer mode register 3	TMDR3	R/W	H'C0	H'FE81
	Timer I/O control register 3H	TIOR3H	R/W	H'00	H'FE82
	Timer I/O control register 3L	TIOR3L	R/W	H'00	H'FE83
	Timer interrupt enable register 3	TIER3	R/W	H'40	H'FE84
	Timer status register 3	TSR3	R/(W)*2	H'C0	H'FE85
	Timer counter 3	TCNT3	R/W	H'0000	H'FE86
	Timer general register 3A	TGR3A	R/W	H'FFFF	H'FE88
	Timer general register 3B	TGR3B	R/W	H'FFFF	H'FE8A
	Timer general register 3C	TGR3C	R/W	H'FFFF	H'FE8C
	Timer general register 3D	TGR3D	R/W	H'FFFF	H'FE8E
TPU4	Timer control register 4	TCR4	R/W	H'00	H'FE90
	Timer mode register 4	TMDR4	R/W	H'C0	H'FE91
	Timer I/O control register 4	TIOR4	R/W	H'00	H'FE92
	Timer interrupt enable register 4	TIER4	R/W	H'40	H'FE94
	Timer status register 4	TSR4	R/(W)*2	H'C0	H'FE95
	Timer counter 4	TCNT4	R/W	H'0000	H'FE96
	Timer general register 4A	TGR4A	R/W	H'FFFF	H'FE98
	Timer general register 4B	TGR4B	R/W	H'FFFF	H'FE9A
TPU5	Timer control register 5	TCR5	R/W	H'00	H'FEA0
	Timer mode register 5	TMDR5	R/W	H'C0	H'FEA1
	Timer I/O control register 5	TIOR5	R/W	H'00	H'FEA2
	Timer interrupt enable register 5	TIER5	R/W	H'40	H'FEA4
	Timer status register 5	TSR5	R/(W)*2	H'C0	H'FEA5
	Timer counter 5	TCNT5	R/W	H'0000	H'FEA6
	Timer general register 5A	TGR5A	R/W	H'FFFF	H'FEA8
	Timer general register 5B	TGR5B	R/W	H'FFFF	H'FEAA
ALL TPU channels	Timer start register	TSTR	R/W	H'00	H'FFC0
	Timer syncro register	TSYR	R/W	H'00	H'FFC1
	Module stop control register	MSTPCR	R/W	H'3FFF	H'FF3C

	PPG output mode register	PMR	R/W	H'F0	H'FF47
	Next data enable register H	NDERH	R/W	H'00	H'FF48
	Next data enable register L	NDERL	R/W	H'00	H'FF49
	Output data register H	PODRH	R/(W)*6	H'00	H'FF4A
	Output data register L	PODRL	R/(W)*6	H'00	H'FF4B
	Next data register H	NDRH	R/W	H'00	H'FF4C*7 H'FF4E
	Next data register L	NDRL	R/W	H'00	H'FF4D*7 H'FF4F
	Port 1 data direction register	P1DDR	W	H'00	H'FEB0
	Port 2 data direction register	P2DDR	W	H'00	H'FEB1
	Module stop control register	MSTPCR	R/W	H'3FFF	H'FF3C
8-bit timer 0	Timer control register 0	TCR0	R/W	H'00	H'FFB0
	Timer control/status register 0	TCSR0	R/(W)*8	H'00	H'FFB2
	Timer constant register A0	TCORA0	R/W	H'FF	H'FFB4
	Timer constant register B0	TCORB0	R/W	H'FF	H'FFB6
	Timer counter 0	TCNT0	R/W	H'00	H'FFB8
8-bit timer 1	Timer control register 1	TCR1	R/W	H'00	H'FFB1
	Timer control/status register 1	TCSR1	R/(W)*8	H'10	H'FFB3
	Timer constant register A1	TCORA1	R/W	H'FF	H'FFB5
	Timer constant register B1	TCORB1	R/W	H'FF	H'FFB7
	Timer counter 1	TCNT1	R/W	H'00	H'FFB9
Both 8-bit timer channels	Module stop control register	MSTPCR	R/W	H'3FFF	H'FF3C

					Write*9 H'FFBC: Read
	Timer counter	TCNT	R/W	H'00	H'FFBC: Write*9 H'FFBD: Read
	Reset control/status register	RSTCSR	R/(W)*10	H'1F	H'FFBE: Write*9 H'FFBF: Read
SCI0	Serial mode register 0	SMR0	R/W	H'00	H'FF78
	Bit rate register 0	BRR0	R/W	H'FF	H'FF79
	Serial control register 0	SCR0	R/W	H'00	H'FF7A
	Transmit data register 0	TDR0	R/W	H'FF	H'FF7B
	Serial status register 0	SSR0	R/(W)*2	H'84	H'FF7C
	Receive data register 0	RDR0	R	H'00	H'FF7D
	Smart card mode register 0	SCMR0	R/W	H'F2	H'FF7E
SCI1	Serial mode register 1	SMR1	R/W	H'00	H'FF80
	Bit rate register 1	BRR1	R/W	H'FF	H'FF81
	Serial control register 1	SCR1	R/W	H'00	H'FF82
	Transmit data register 1	TDR1	R/W	H'FF	H'FF83
	Serial status register 1	SSR1	R/(W)*2	H'84	H'FF84
	Receive data register 1	RDR1	R	H'00	H'FF85
	Smart card mode register 1	SCMR1	R/W	H'F2	H'FF86
SCI2	Serial mode register 2	SMR2	R/W	H'00	H'FF88
	Bit rate register 2	BRR2	R/W	H'FF	H'FF89
	Serial control register 2	SCR2	R/W	H'00	H'FF8A
	Transmit data register 2	TDR2	R/W	H'FF	H'FF8B
	Serial status register 2	SSR2	R/(W)*2	H'84	H'FF8C
	Receive data register 2	RDR2	R	H'00	H'FF8D
	Smart card mode register 2	SCMR2	R/W	H'F2	H'FF8E
All SCI channels	Module stop control register	MSTPCR	R/W	H'3FFF	H'FF3C

	Bit rate register 0	BRR0	R/W	H'FF	H'FF79
	Serial control register 0	SCR0	R/W	H'00	H'FF7A
	Transmit data register 0	TDR0	R/W	H'FF	H'FF7B
	Serial status register 0	SSR0	R/(W)*2	H'84	H'FF7C
	Receive data register 0	RDR0	R	H'00	H'FF7D
	Smart card mode register 0	SCMR0	R/W	H'F2	H'FF7E
SMCI1	Serial mode register 1	SMR1	R/W	H'00	H'FF80
	Bit rate register 1	BRR1	R/W	H'FF	H'FF81
	Serial control register 1	SCR1	R/W	H'00	H'FF82
	Transmit data register 1	TDR1	R/W	H'FF	H'FF83
	Serial status register 1	SSR1	R/(W)*2	H'84	H'FF84
	Receive data register 1	RDR1	R	H'00	H'FF85
	Smart card mode register 1	SCMR1	R/W	H'F2	H'FF86
SMCI2	Serial mode register 2	SMR2	R/W	H'00	H'FF88
	Bit rate register 2	BRR2	R/W	H'FF	H'FF89
	Serial control register 2	SCR2	R/W	H'00	H'FF8A
	Transmit data register 2	TDR2	R/W	H'FF	H'FF8B
	Serial status register 2	SSR2	R/(W)*2	H'84	H'FF8C
	Receive data register 2	RDR2	R	H'00	H'FF8D
	Smart card mode register 2	SCMR2	R/W	H'00	H'FF8E
All SMCI channels	Module stop control register	MSTPCR	R/W	H'3FFF	H'FF3C
ADC	A/D data register AH	ADDRAH	R	H'00	H'FF90
	A/D data register AL	ADDRAL	R	H'00	H'FF91
	A/D data register BH	ADDRBH	R	H'00	H'FF92
	A/D data register BL	ADDRBL	R	H'00	H'FF93
	A/D data register CH	ADDRCH	R	H'00	H'FF94
	A/D data register CL	ADDRCL	R	H'00	H'FF95
	A/D data register DH	ADDRDH	R	H'00	H'FF96
	A/D data register DL	ADDRDL	R	H'00	H'FF97
	A/D control/status register	ADCSR	R/(W)*10	H'00	H'FF98
	A/D control register	ADCR	R/W	H'3F	H'FF99
	Module stop control register	MSTPCR	R/W	H'3FFF	H'FF3C

	D/A data register 1	DADR1	R/W	H'00	H'FFA5
	D/A control register 01	DACR01	R/W	H'1F	H'FFA6
DAC2, 3	D/A data register 2	DADR2	R/W	H'00	H'FDA8
	D/A data register 3	DADR3	R/W	H'00	H'FDA9
	D/A control register 23	DACR23	R/W	H'1F	H'FDAA
All DAC channels	Module stop control register	MSTPCR	R/W	H'3FFF	H'FF3C
On-chip RAM	System control register	SYSCR	R/W	H'01	H'FF39
Flash memory	Flash memory control register 1	FLMCR1*15	R/W*12	H'00/H'80*13	H'FFC8*11
	Flash memory control register 2	FLMCR2*15	R/W*12	H'00	H'FFC9*11
	Erase block register 1	EBR1*15	R/W*12	H'00*14	H'FFCA*11
	Erase block register 2	EBR2*15	R/W*12	H'00*14	H'FFCB*11
	RAM emulation register	RAMER*21	R/W	H'00	H'FEDB
	System control register 2	SYSCR2*16	R/W	H'00	H'FF42
Clock pulse generator	System clock control register	SCKCR	R/W	H'00	H'FF3A
Power-down mode	Standby control register	SBYCR	R/W	H'08	H'FF38
	System clock control register	SCKCR	R/W	H'00	H'FF3A
	Module stop control register H	MSTPCRH	R/W	H'3F	H'FF3C
	Module stop control register L	MSTPCRL	R/W	H'FF	H'FF3D
Port 1	Port 1 data direction register	P1DDR	W	H'00	H'FEB0
	Port 1 data register	P1DR	R/W	H'00	H'FF60
	Port 1 register	PORT1	R	Undefined	H'FF50
Port 2	Port 2 data direction register	P2DDR	W	H'00	H'FEB1
	Port 2 data register	P2DR	R/W	H'00	H'FF61
	Port 2 register	PORT2	R	Undefined	H'FF51
Port 3	Port 3 data direction register	P3DDR	W	H'00	H'FEB2
	Port 3 data register	P3DR	R/W	H'00	H'FF62
	Port 3 register	PORT3	R	Undefined	H'FF52
	Port 3 open drain control register	P3ODR	R/W	H'00	H'FF76
Port 4	Port 4 register	PORT4	R	Undefined	H'FF53

	Port 5 data register	P5DR	R/W	H'0*17	H'FF64
	Port 5 register	PORT5	R	Undefined	H'FF54
	Port function control register 2	PFCR2	R/W	H'30	H'FFAC
	System control register	SYSCR	R/W	H'01	H'FF39
Port 6	Port 6 data direction register	P6DDR	W	H'00	H'FEB5
	Port 6 data register	P6DR	R/W	H'00	H'FF65
	Port 6 register	PORT6	R	Undefined	H'FF55
	Port function control register 2	PFCR2	R/W	H'30	H'FFAC
Port 7	Port 7 data direction register	P7DDR	W	H'00	H'FEB6
	Port 7 data register	P7DR	R/W	H'00	H'FF66
	Port 7 register	PORT7	R	Undefined	H'FF56
Port 8	Port 8 data direction register	P8DDR	W	H'00*18	H'FEB7
	Port 8 data register	P8DR	R/W	H'00*18	H'FF67
	Port 8 register	PORT8	R	Undefined*18	H'FF57
	Port function control register 2	PFCR2	R/W	H'30*18	H'FFAC
Port 9	Port 9 data direction register	P9DDR	W	H'00*19	H'FEB8
	Port 9 data register	P9DR	R/W	H'00*19	H'FF68
	Port 9 register	PORT9	R	Undefined	H'FF58
	System control register	SYSCR	R/W	H'01	H'FF39
Port A	Port A data direction register	PADDR	W	H'00	H'FEB9
	Port A data register	PADR	R/W	H'00	H'FF69
	Port A register	PORTA	R	Undefined	H'FF59
	Port A MOS pull-up control register	PAPCR	R/W	H'00	H'FF70
	Port A open drain control register	PAODR	R/W	H'00	H'FF77
	Port function control register 1	PFCR1	R/W	H'0F	H'FF45
Port B	Port B data direction register	PBDDR	W	H'00	H'FEBA
	Port B data register	PBDR	R/W	H'00	H'FF6A
	Port B register	PORTB	R	Undefined	H'FF5A
	Port B MOS pull-up control register	PBPCR	R/W	H'00	H'FF71

	Port C data register	PCDR	R/W	H'00	H'FF6B
	Port C register	PORTC	R	Undefined	H'FF5B
	Port C MOS pull-up control register	PCPCR	R/W	H'00	H'FF72
Port D	Port D data direction register	PDDDR	W	H'00	H'FEBC
	Port D data register	PDDR	R/W	H'00	H'FF6C
	Port D register	PORTD	R	Undefined	H'FF5C
	Port D MOS pull-up control register	PDPCR	R/W	H'00	H'FF73
Port E	Port E data direction register	PEDDR	W	H'00	H'FEBD
	Port E data register	PEDR	R/W	H'00	H'FF6D
	Port E register	PORTE	R	Undefined	H'FF5D
	Port E MOS pull-up control register	PEPCR	R/W	H'00	H'FF74
Port F	Port F data direction register	PFDDR	W	H'80/H'00*5	H'FEBE
	Port F data register	PFDR	R/W	H'00	H'FF6E
	Port F register	PORTF	R	Undefined	H'FF5E
	Port function control register 2	PFCR2	R/W	H'30	H'FFAC
	System control register	SYSCR	R/W	H'01	H'FF39
Port G	Port G data direction register	PGDDR	W	H'10/H'00 *5 *20	H'FEBF
	Port G data register	PGDR	R/W	H'00*20	H'FF6F
	Port G register	PORTG	R	Undefined*20	H'FF5F
	Port function control register 2	PFCR2	R/W	H'30	H'FFAC

Notes:
1. Lower 16 bits of the address.
2. Only 0 can be written for flag clearing.
3. Registers in the DTC cannot be read or written to directly.
4. Located as register information in on-chip RAM addresses H'EBC0 to H'EFBF. Cannot be located in external memory space. Do not clear the RAME bit in SYSCR to 0 when using the DTC.
5. The initial value differs depending on the MCU operating mode.
6. Bits used for pulse output cannot be written to.
7. If the pulse output group 2 and pulse output group 3 output triggers are the same according to the PCR setting, the NDRH address will be H'FF4C, and if different, the address of NDRH for group 2 will be H'FF4E, and that for group 3 will be H'FF4C. Similarly, if the pulse output group 0 and pulse output group 1 output triggers are the same according to the PCR setting, the NDRL address will be H'FF4D, and if different, the address of NDRL for group 0 will be H'FF4F, and that for group 1 will be H'FF4D.
8. Only 0 can be written to bits 7 to 5, to clear the flags.

11. Flash memory registers selection is performed by means of the FLSHE bit in system control register 2 (SYSCR2).
12. In modes in which the on-chip flash memory is disabled, a read will return H'00, and writes are invalid. Writes are also disabled when the FWE bit in FLMCR1 is cleared to 0. (Not applies to H8S/2339 F-ZTAT.)
13. In H8S/2338 F-ZTAT, when a high level is input to the FWE pin, the initial value is H'80. In H8S/2339 F-ZTAT, the initial value is H'80.
14. In H8S/2338 F-ZTAT, when a low level is input to the FWE pin, or if a high level is input but the SWE bit in FLMCR1 is not set, these registers are initialized to H'00. In H8S/2339 F-ZTAT, when the SWE bit in FLMCR1 is not set, these registers are initialized to H'00.
15. FLMCR1, FLMCR2, EBR1, and EBR2 are 8-bit registers. Only byte access can be used on these registers, with the access requiring two states.
16. The SYSCR2 register can only be used in the F-ZTAT version. In the mask ROM version this register will return an undefined value if read, and cannot be written to.
17. Value of bits 3 to 0.
18. Value of bits 6 to 0.
19. Value of bits 7 to 2.
20. Value of bits 4 to 0.
21. Valid only in flash memory versions.

Bit	7	6	5	4	3	2	1	0
	SM1	SM0	DM1	DM0	MD1	MD0	DTS	Sz
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	—	—	—	—	—	—	—	—

DTC Data Transfer Size

0	Byte-size transfer
1	Word-size transfer

DTC Transfer Mode Select

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

DTC Mode

0	0	Normal mode
	1	Repeat mode
1	0	Block transfer mode
	1	—

Destination Address Mode

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)

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)

Bit	7	6	5	4	3	2	1	0
	CHNE	DISEL	CHNS	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	—	—	—	—	—	—	—	—

Reserved
Only 0 should be written to these bits

DTC Chain Transfer Enable, DTC Chain Transfer Select

CHNE	CHNS	Description
0	—	No chain transfer. (At end of DTC data transfer, DTC waits for activation)
1	0	Chain transfer every time
1	1	Chain transfer only when transfer counter = 0

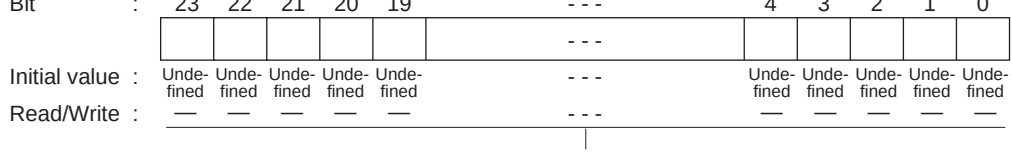
DTC Interrupt Select

0	After DTC data transfer ends, the CPU interrupt is disabled unless the transfer counter is 0
1	After DTC data transfer ends, the CPU interrupt is enabled

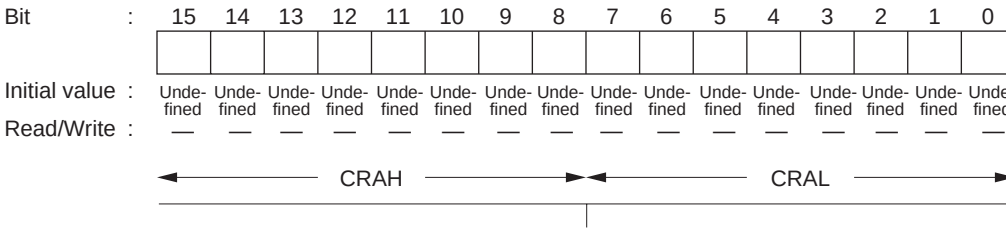
SAR—DTC Source Address Register H'F800 to H'FBFF DTC

Bit	23	22	21	20	19	- - -	4	3	2	1	0
						- - -					
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	- - -	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	—	—	—	—	—	- - -	—	—	—	—	—

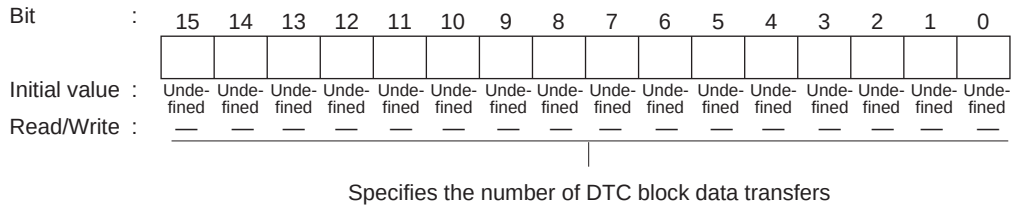
Specifies DTC transfer data source address



CRA—DTC Transfer Count Register A H'F800 to H'FBFF DTC



CRB—DTC Transfer Count Register B H'F800 to H'FBFF DTC



Bit	:	7	6	5	4	3	2	1	0
		CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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 TCLKA pin input
		1	Internal clock: counts on ϕ/1024
	1	0	Internal clock: counts on ϕ/256
		1	Internal clock: counts on ϕ/4096

Clock Edge

0	0	Count at rising edge
	1	Count at falling edge
1	—	Count at both edges

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 TSYR to 1.
2. When TGRC or TGRD is used as a buffer register, TCNT is not cleared because the buffer register setting has priority, and compare match/input capture does not occur.

Bit	:	7	6	5	4	3	2	1	0
		—	—	BFB	BFA	MD3	MD2	MD1	MD0
Initial value	:	1	1	0	0	0	0	0	0
Read/Write	:	—	—	R/W	R/W	R/W	R/W	R/W	R/W

Mode

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	*	*	*	—

* : Don't care

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 should always be written to MD2.

Buffer Operation A

0	TGRA operates normally
1	TGRA and TGRC used together for buffer operation

Buffer Operation B

0	TGRB operates normally
1	TGRB and TGRD used together for buffer operation

Bit	:	7	6	5	4	3	2	1	0
Initial value	:	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
		0	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TGR3A I/O Control

0	0	0	0	TGR3A 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	TGR3A is input capture register	Capture input source is $TIOCA_3$ 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

* : Don't care

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 $TIOCB_3$ 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 ϕ/1 is used as the TCNT4 count clock, this setting is invalid and input capture does not occur.

Bit	7	6	5	4	3	2	1	0
	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TGR3C I/O Control

0	0	0	0	TGR3C is output compare register*1	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	TGR3C is input capture register*1	Capture input source is $TIOCC_3$ 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

* : Don't care

Note: 1. When the BFA bit in TMDR3 is set to 1 and TGR3C is used as a buffer register, this setting is invalid and input capture/output compare does not occur.

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 $TIOCD_3$ 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 does not occur.
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 does not occur.

Note: When TGRC or TGRD is designated for buffer operation, this setting is invalid and the register operates as a buffer register.

Bit	:	7	6	5	4	3	2	1	0
		TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
Initial value	:	0	1	0	0	0	0	0	0
Read/Write	:	R/W	—	—	R/W	R/W	R/W	R/W	R/W

TGR Interrupt Enable A

0	Interrupt request (TGIA) by TGFA bit disabled
1	Interrupt request (TGIA) by TGFA bit enabled

TGR Interrupt Enable B

0	Interrupt request (TGIB) by TGFB bit disabled
1	Interrupt request (TGIB) by TGFB bit enabled

TGR Interrupt Enable C

0	Interrupt request (TGIC) by TGFC bit disabled
1	Interrupt request (TGIC) by TGFC bit enabled

TGR Interrupt Enable D

0	Interrupt request (TGID) by TGFD bit disabled
1	Interrupt request (TGID) by TGFD bit enabled

Overflow Interrupt Enable

0	Interrupt request (TCIV) by TCFV disabled
1	Interrupt request (TCIV) by TCFV enabled

A/D Conversion Start Request Enable

0	A/D conversion start request generation disabled
1	A/D conversion start request generation enabled

Bit :	7	6	5	4	3	2	1	0
	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
Initial value :	1	1	0	0	0	0	0	0
Read/Write :	—	—	—	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

TGR Input Capture/Output Compare Flag A

0	[Clearing conditions] • When DTC is activated by TGIA interrupt while DISEL bit of MRB in DTC is 0 • When DMAC is activated by TGIA interrupt while DTA bit of DMABCR in DMAC is 1 • When 0 is written to TGFA after reading TGFA = 1
1	[Setting conditions] • When TCNT=TGRA while TGRA is functioning as output compare register • When TCNT value is transferred to TGRA by input capture signal while TGRA is functioning as input capture register

TGR Input Capture/Output Compare Flag B

0	[Clearing conditions] • When DTC is activated by TGIB interrupt while DISEL bit of MRB in DTC 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 while DISEL bit of MRB in DTC is 0 • When 0 is written to TGFC after reading TGFC = 1
1	[Setting conditions] • When TCNT = TGRC while TGRC is functioning as output compare register • When TCNT value is transferred to TGRC by input capture signal while TGRC is functioning as input capture register

TGR Input Capture/Output Compare Flag D

0	[Clearing conditions] • When DTC is activated by TGID interrupt while DISEL bit of MRB in DTC is 0 • When 0 is written to TGFD after reading TGFD = 1
1	[Setting conditions] • When TCNT = TGRD while TGRD is functioning as output compare register • When TCNT value is transferred to TGRD by input capture signal while TGRD is functioning as input capture register

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.

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
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	R/W	R/W

Up-counter

TGR3A—Timer General Register 3A **H'FE88** **TPU3**
TGR3B—Timer General Register 3B **H'FE8A** **TPU3**
TGR3C—Timer General Register 3C **H'FE8C** **TPU3**
TGR3D—Timer General Register 3D **H'FE8E** **TPU3**

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	:	1	1	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	R/W	R/W

Bit :	7	6	5	4	3	2	1	0
	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
Initial value :	0	0	0	0	0	0	0	0
Read/Write :	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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 TCLKA pin input
		1	External clock: counts on TCLKC pin input
	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.

Clock Edge

0	0	Count at rising edge
	1	Count at falling edge
1	—	Count at both edges

Note: This setting is ignored when channel 4 is in phase counting mode.

Counter Clear

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*

Note: * Synchronous operation setting is performed by setting the SYNC bit in TSYR to 1.

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	MD3	MD2	MD1	MD0
Initial value	:	1	1	0	0	0	0	0	0
Read/Write	:	—	—	—	—	R/W	R/W	R/W	R/W

Mode

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	*	*	*	—

* : Don't care

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

Bit	:	7	6	5	4	3	2	1	0
		IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TGR4A I/O Control

0	0	0	0	TGR4A 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	TGR4A is input capture register	Capture input source is $TIOCA_4$ pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Capture input source is TGR3A compare match/ input capture	Input capture at generation of TGR3A compare match/input capture

* : Don't care

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 $TIOCB_4$ 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

Bit	7	6	5	4	3	2	1	0
	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
Initial value :	0	1	0	0	0	0	0	0
Read/Write :	R/W	—	R/W	R/W	—	—	R/W	R/W

TGR Interrupt Enable A

0	Interrupt request (TGIA) by TGFA bit disabled
1	Interrupt request (TGIA) by TGFA bit enabled

TGR Interrupt Enable B

0	Interrupt request (TGIB) by TGFB bit disabled
1	Interrupt request (TGIB) by TGFB bit enabled

Overflow Interrupt Enable

0	Interrupt request (TCIV) by TCFV disabled
1	Interrupt request (TCIV) by TCFV enabled

Underflow Interrupt Enable

0	Interrupt request (TCIU) by TCFU disabled
1	Interrupt request (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

Bit	:	7	6	5	4	3	2	1	0
		TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA
Initial value	:	1	1	0	0	0	0	0	0
Read/Write	:	R	—	R/(W)*	R/(W)*	—	—	R/(W)*	R/(W)*

TGR Input Capture/Output Compare Flag A

0	[Clearing conditions] • When DTC is activated by TGIA interrupt while DISEL bit of MRB in DTC is 0 • When DMAC is activated by TGIA interrupt while DTA bit of DMABCR in DMAC is 1 • When 0 is written to TGFA after reading TGFA = 1
1	[Setting conditions] • When TCNT = TGRA while TGRA is functioning as output compare register • When TCNT value is transferred to TGRA by input capture signal while TGRA is functioning as input capture register

TGR Input Capture/Output Compare Flag B

0	[Clearing conditions] • When DTC is activated by TGIB interrupt while DISEL bit of MRB in DTC 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

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)

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'FFFF)

Count Direction Flag

0	TCNT counts down
1	TCNT counts up

Note: * Can only be written with 0 for flag clearing.

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
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	R/W	R/W

Up/down-counter*

Note: * This timer counter can be used as an up/down-counter only in phase counting mode or when performing overflow/underflow counting on another channel. In other cases it functions as an up-counter.

TGR4A—Timer General Register 4A **H'FE98** **TPU4**
TGR4B—Timer General Register 4B **H'FE9A** **TPU4**

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	:	1	1	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	R/W	R/W

Bit :	7	6	5	4	3	2	1	0
	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
Initial value :	0	0	0	0	0	0	0	0
Read/Write :	—	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 pin input
		1	External clock: counts on TCLKC pin input
	1	0	Internal clock: counts on ϕ/256
		1	External clock: counts on TCLKD pin input

Note: This setting is ignored when channel 5 is in phase counting mode.

Clock Edge

0	0	Count at rising edge
	1	Count at falling edge
1	—	Count at both edges

Note: This setting is ignored when channel 5 is in phase counting mode.

Counter Clear

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*

Note: * Synchronous operation setting is performed by setting the SYNC bit in TSYR to 1.

Bit :	7	6	5	4	3	2	1	0
	—	—	—	—	MD3	MD2	MD1	MD0
Initial value :	1	1	0	0	0	0	0	0
Read/Write :	—	—	—	—	R/W	R/W	R/W	R/W

Mode

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	*	*	*	—

* : Don't care

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

Bit	:	7	6	5	4	3	2	1	0
		IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TGR5A I/O Control

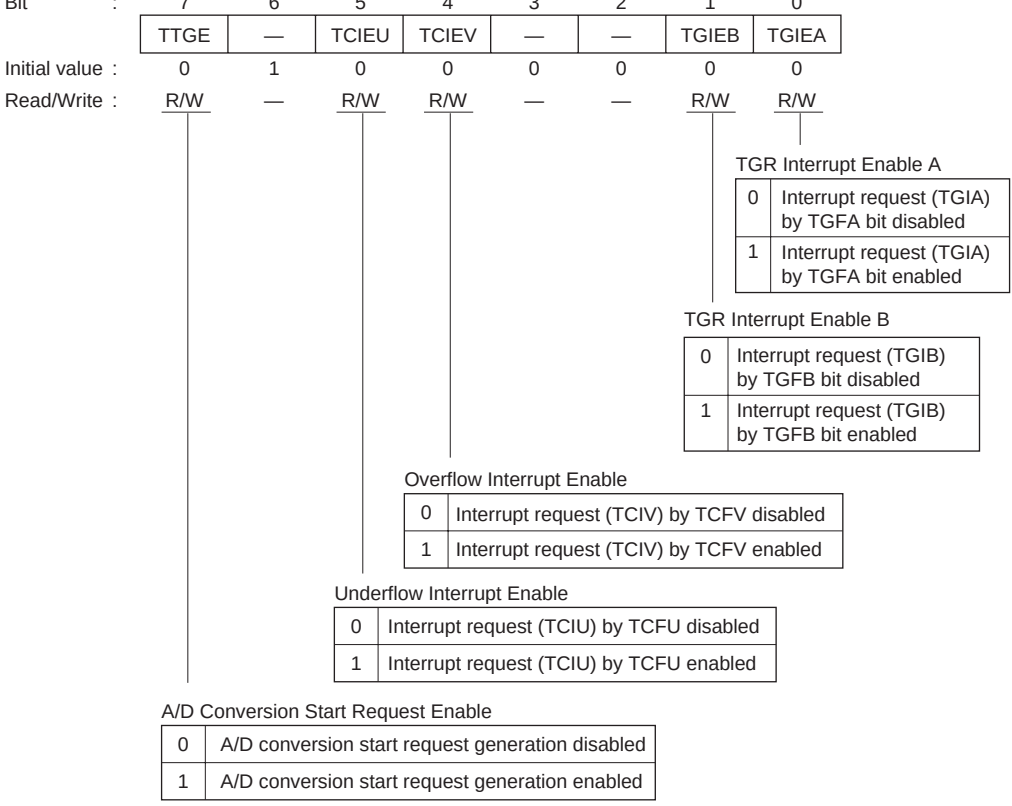
0	0	0	0	TGR5A is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	*	0	0	TGR5A is input capture register	Capture input source is $TIOCA_5$ pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges

* : Don't care

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 $TIOCB_5$ pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges

* : Don't care


Bit : 7 6 5 4 3 2 1 0
TTGE — TCIEU TCIEV — — TGIEB TGIEA
Initial value : 0 1 0 0 0 0 0 0
Read/Write : R/W — R/W R/W — — R/W R/W
TGR Interrupt Enable A
0 Interrupt request (TGIA) by TGFA bit disabled
1 Interrupt request (TGIA) by TGFA bit enabled
TGR Interrupt Enable B
0 Interrupt request (TGIB) by TGFB bit disabled
1 Interrupt request (TGIB) by TGFB bit enabled
Overflow Interrupt Enable
0 Interrupt request (TCIV) by TCFV disabled
1 Interrupt request (TCIV) by TCFV enabled
Underflow Interrupt Enable
0 Interrupt request (TCIU) by TCFU disabled
1 Interrupt request (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

Bit	7	6	5	4	3	2	1	0
	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA
Initial value :	1	1	0	0	0	0	0	0
Read/Write :	R	—	R/(W)*	R/(W)*	—	—	R/(W)*	R/(W)*

TGR Input Capture/Output Compare Flag A

0	[Clearing conditions] • When DTC is activated by TGIA interrupt while DISEL bit of MRB in DTC is 0 • When DMAC is activated by TGIA interrupt while DTA bit of DMABCR in DMAC is 1 • When 0 is written to TGFA after reading TGFA = 1
1	[Setting conditions] • When TCNT = TGRA while TGRA is functioning as output compare register • When TCNT value is transferred to TGRA by input capture signal while TGRA is functioning as input capture register

TGR Input Capture/Output Compare Flag B

0	[Clearing conditions] • When DTC is activated by TGIB interrupt while DISEL bit of MRB in DTC 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

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)

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'FFFF)

Count Direction Flag

0	TCNT counts down
1	TCNT counts up

Note: * Can only be written with 0 for flag clearing.

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
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	R/W	R/W

Up/down-counter*

Note: * This timer counter can be used as an up/down-counter only in phase counting mode or when performing overflow/underflow counting on another channel. In other cases it functions as an up-counter.

TGR5A—Timer General Register 5A **H'FEA8** **TPU5**

TGR5B—Timer General Register 5B **H'FEAA** **TPU5**

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	:	1	1	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	R/W	R/W

P1DDR—Port 1 Data Direction Register **H'FEB0** **Port 1**

Bit	:	7	6	5	4	3	2	1	0
		P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DDR
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	W	W	W	W	W	W	W	W

Specify input or output for individual port 1 pins

Bit	:	7	6	5	4	3	2	1	0
		P27DDR	P26DDR	P25DDR	P24DDR	P23DDR	P22DDR	P21DDR	P20DDR
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	W	W	W	W	W	W	W	W

Specify input or output for individual port 2 pins

P3DDR—Port 3 Data Direction Register H'FEB2 Port 3

Bit	:	7	6	5	4	3	2	1	0
		—	—	P35DDR	P34DDR	P33DDR	P32DDR	P31DDR	P30DDR
Initial value	:	Undefined	Undefined	0	0	0	0	0	0
Read/Write	:	—	—	W	W	W	W	W	W

Specify input or output for individual port 3 pins

P5DDR—Port 5 Data Direction Register H'FEB4 Port 5

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	P53DDR	P52DDR	P51DDR	P50DDR
Initial value	:	Undefined	Undefined	Undefined	Undefined	0	0	0	0
Read/Write	:	—	—	—	—	W	W	W	W

Specify input or output for individual port 5 pins

Bit :	7	6	5	4	3	2	1	0
	P67DDR	P66DDR	P65DDR	P64DDR	P63DDR	P62DDR	P61DDR	P60DDR
Initial value :	0	0	0	0	0	0	0	0
Read/Write :	W	W	W	W	W	W	W	W

Specify input or output for individual port 6 pins

P7DDR—Port 7 Data Direction Register H'FEB6 Port 7

Bit :	7	6	5	4	3	2	1	0
	—	—	P75DDR	P74DDR	P73DDR	P72DDR	P71DDR	P70DDR
Initial value :	Undefined	Undefined	0	0	0	0	0	0
Read/Write :	—	—	W	W	W	W	W	W

Specify input or output for individual port 7 pins

P8DDR—Port 8A Data Direction Register H'FEB7 Port 8

Bit :	7	6	5	4	3	2	1	0
	—	P86DDR	P85DDR	P84DDR	P83DDR	P82DDR	P81DDR	P80DDR
Initial value :	Undefined	0	0	0	0	0	0	0
Read/Write :	—	W	W	W	W	W	W	W

Specify input or output for individual port 8 pins

Bit	:	7	6	5	4	3	2	1	0
		P97DDR	P96DDR	P95DDR	P94DDR	P93DDR	P92DDR	—	—
Initial value	:	0	0	0	0	0	0	Undefined	Undefined
Read/Write	:	W	W	W	W	W	W	—	—

Specify input or output for individual port 9 pins

PADDR—Port A Data Direction Register H'FEB9 Port A

Bit	:	7	6	5	4	3	2	1	0
		PA7DDR	PA6DDR	PA5DDR	PA4DDR	PA3DDR	PA2DDR	PA1DDR	PA0DDR
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	W	W	W	W	W	W	W	W

Specify input or output for individual port A pins

PBDDR—Port B Data Direction Register H'FEBA Port B

Bit	:	7	6	5	4	3	2	1	0
		PB7DDR	PB6DDR	PB5DDR	PB4DDR	PB3DDR	PB2DDR	PB1DDR	PB0DDR
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	W	W	W	W	W	W	W	W

Specify input or output for individual port B pins

Bit	:	7	6	5	4	3	2	1	0
		PC7DDR	PC6DDR	PC5DDR	PC4DDR	PC3DDR	PC2DDR	PC1DDR	PC0DDR
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	W	W	W	W	W	W	W	W

Specify input or output for individual port C pins

PDDDR—Port D Data Direction Register **H'FEBC** **Port D**

Bit	:	7	6	5	4	3	2	1	0
		PD7DDR	PD6DDR	PD5DDR	PD4DDR	PD3DDR	PD2DDR	PD1DDR	PD0DDR
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	W	W	W	W	W	W	W	W

Specify input or output for individual port D pins

PEDDR—Port E Data Direction Register **H'FEBD** **Port E**

Bit	:	7	6	5	4	3	2	1	0
		PE7DDR	PE6DDR	PE5DDR	PE4DDR	PE3DDR	PE2DDR	PE1DDR	PE0DDR
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	W	W	W	W	W	W	W	W

Specify input or output for individual port E pins

Bit	:	7	6	5	4	3	2	1	0
		PF7DDR	PF6DDR	PF5DDR	PF4DDR	PF3DDR	PF2DDR	PF1DDR	PF0DDR
Modes 4 to 6									
Initial value	:	1	0	0	0	0	0	0	0
Read/Write	:	W	W	W	W	W	W	W	W
Mode 7									
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	W	W	W	W	W	W	W	W

Specify input or output for individual port F pins

PGDDR—Port G Data Direction Register H'FEBF Port G

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	PG4DDR	PG3DDR	PG2DDR	PG1DDR	PG0DDR
Modes 4 and 5									
Initial value	:	Undefined	Undefined	Undefined	1	0	0	0	0
Read/Write	:	—	—	—	W	W	W	W	W
Modes 6 and 7									
Initial value	:	Undefined	Undefined	Undefined	0	0	0	0	0
Read/Write	:	—	—	—	W	W	W	W	W

Specify input or output for individual port G pins

IPRC — Interrupt Priority Register C	**H'FEC6**	**Interrupt Controller**
IPRD — Interrupt Priority Register D	**H'FEC7**	**Interrupt Controller**
IPRE — Interrupt Priority Register E	**H'FEC8**	**Interrupt Controller**
IPRF — Interrupt Priority Register F	**H'FEC9**	**Interrupt Controller**
IPRG — Interrupt Priority Register G	**H'FECA**	**Interrupt Controller**
IPRH — Interrupt Priority Register H	**H'FECB**	**Interrupt Controller**
IPRI — Interrupt Priority Register I	**H'FECC**	**Interrupt Controller**
IPRJ — Interrupt Priority Register J	**H'FECD**	**Interrupt Controller**
IPRK — Interrupt Priority Register K	**H'FECE**	**Interrupt Controller**

Bit	:	7	6	5	4	3	2	1	0
		—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0
Initial value	:	0	1	1	1	0	1	1	1
Read/Write	:	—	R/W	R/W	R/W	—	R/W	R/W	R/W

Set priority (levels 7 to 0) for interrupt sources

Correspondence between Interrupt Sources and IPR Settings

Register	Bits	
	6 to 4	2 to 0
IPRA	IRQ0	IRQ1
IPRB	IRQ2	IRQ4
	IRQ3	IRQ5
IPRC	IRQ6	DTC
	IRQ7	
IPRD	WDT	Refresh timer
IPRE	—*	A/D converter
IPRF	TPU channel 0	TPU channel 1
IPRG	TPU channel 2	TPU channel 3
IPRH	TPU channel 4	TPU channel 5
IPRI	8-bit timer channel 0	8-bit timer channel 1
IPRJ	DMAC	SCI channel 0
IPRK	SCI channel 1	SCI channel 2

Note: * Reserved bits.

Bit	:	7	6	5	4	3	2	1	0
		ABW7	ABW6	ABW5	ABW4	ABW3	ABW2	ABW1	ABW0
Modes 5 to 7									
Initial value	:	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
Mode 4									
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	R/W	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 access
1	Area n is designated for 8-bit access

(n = 7 to 0)

ASTCR—Access State Control Register **H'FED1** **Bus Controller**

Bit	:	7	6	5	4	3	2	1	0
		AST7	AST6	AST5	AST4	AST3	AST2	AST1	AST0
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

Area 7 to 0 Access State Control

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

(n = 7 to 0)

Bit :	7	6	5	4	3	2	1	0
	W71	W70	W61	W60	W51	W50	W41	W40
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

Area 4 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 5 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 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

Bit	:	7	6	5	4	3	2	1	0
		W31	W30	W21	W20	W11	W10	W01	W00
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

Area 0 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 1 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 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

Bit	7	6	5	4	3	2	1	0
	ICIS1	ICIS0	BRSTRM	BRSTS1	BRSTS0	RMTS2	RMTS1	RMTS0
Initial value :	1	1	0	1	0	0	0	0
Read/Write :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

RAM Type Select

RMTS2	RMTS1	RMTS0	Area 5	Area 4	Area 3	Area 2
0	0	0	Normal space			
		1	Normal space			DRAM space
	1	0	Normal space		DRAM space	
		1	DRAM space			
1	—	—	—			

Note: When areas selected in DRAM space are all 8-bit space, the PF_2 pin can be used as an I/O port or $\overline{BREQO}$.

Burst Cycle Select 0

0	Max. 4 words in burst access
1	Max. 8 words in burst access

Burst Cycle Select 1

0	Burst cycle comprises 1 state
1	Burst cycle comprises 2 states

Area 0 Burst ROM Enable

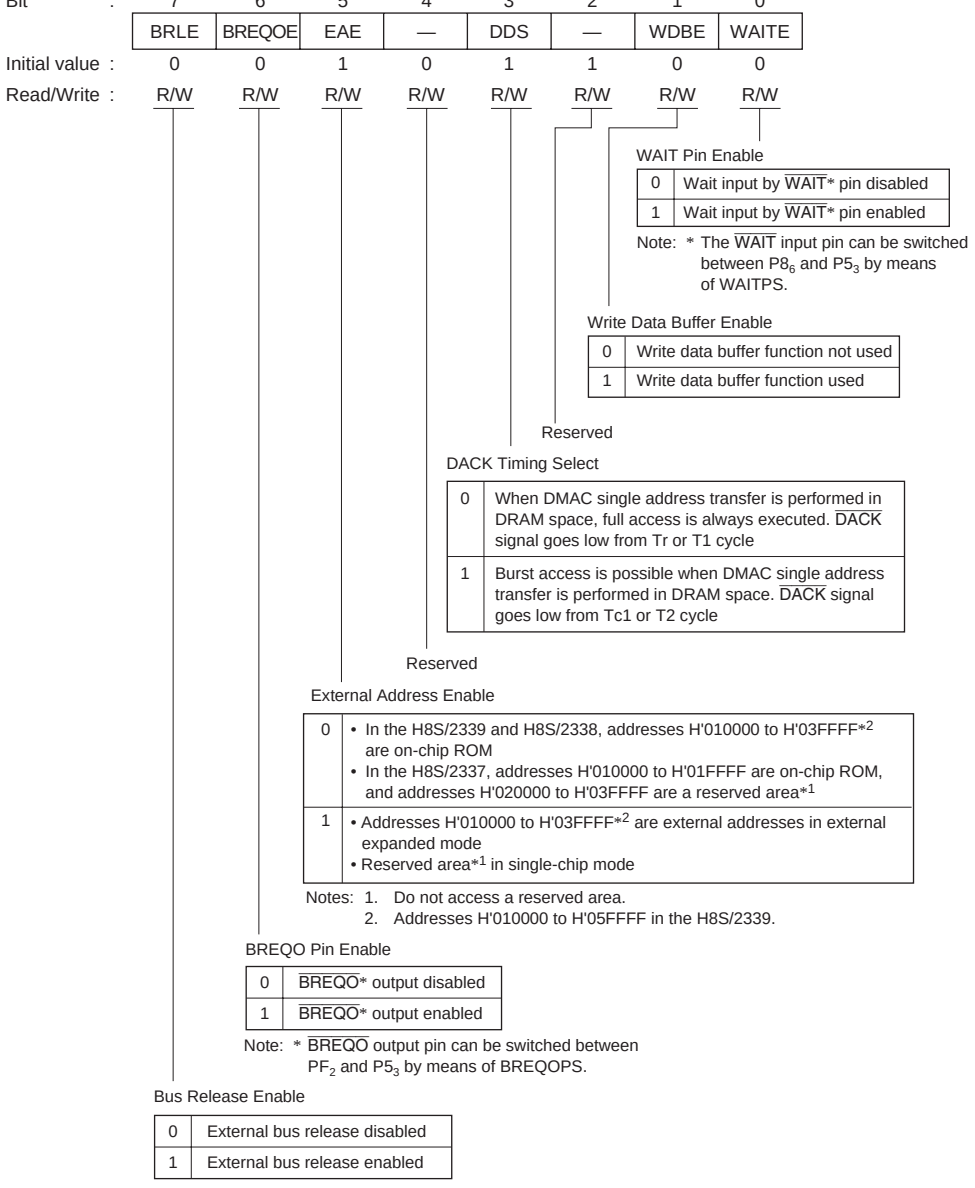
0	Basic bus interface
1	Burst ROM interface

Idle Cycle Insert 0

0	Idle cycle not inserted in case of successive external read and external write cycles
1	Idle cycle inserted in case of successive external read and external write cycles

Idle Cycle Insert 1

0	Idle cycle not inserted in case of successive external read cycles in different areas
1	Idle cycle inserted in case of successive external read cycles in different areas


Bit : 7 6 5 4 3 2 1 0
BRLE BREQOE EAE — DDS — WDBE WAITE
Initial value : 0 0 1 0 1 1 0 0
Read/Write : R/W R/W R/W R/W R/W R/W R/W R/W
WAIT Pin Enable
0 Wait input by WAIT* pin disabled
1 Wait input by WAIT* pin enabled
Note: * The WAIT input pin can be switched between P8₆ and P5₃ by means of WAITPS.
Write Data Buffer Enable
0 Write data buffer function not used
1 Write data buffer function used
Reserved
DACK Timing Select
0 When DMAC single address transfer is performed in DRAM space, full access is always executed. DACK signal goes low from Tr or T1 cycle
1 Burst access is possible when DMAC single address transfer is performed in DRAM space. DACK signal goes low from Tc1 or T2 cycle
Reserved
External Address Enable
0 • In the H8S/2339 and H8S/2338, addresses H'010000 to H'03FFFF*2 are on-chip ROM
• In the H8S/2337, addresses H'010000 to H'01FFFF are on-chip ROM, and addresses H'020000 to H'03FFFF are a reserved area*1
1 • Addresses H'010000 to H'03FFFF*2 are external addresses in external expanded mode
• Reserved area*1 in single-chip mode
Notes: 1. Do not access a reserved area.
2. Addresses H'010000 to H'05FFFF in the H8S/2339.
BREQO Pin Enable
0 BREQO* output disabled
1 BREQO* output enabled
Note: * BREQO output pin can be switched between PF₂ and P5₃ by means of BREQOPS.
Bus Release Enable
0 External bus release disabled
1 External bus release enabled

Bit :	7	6	5	4	3	2	1	0
	TPC	BE	RCDM	—	MXC1	MXC0	RLW1	RLW0
Initial value :	0	0	0	0	0	0	0	0
Read/Write :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Refresh Cycle Wait Control

0	0	No wait state inserted
	1	1 wait state inserted
1	0	2 wait states inserted
	1	3 wait states inserted

Multiplex Shift Count

0	0	8-bit shift
	1	9-bit shift
1	0	10-bit shift
	1	—

Reserved

RAS Down Mode

0	RAS up mode selected for DRAM interface
1	RAS down mode selected for DRAM interface

Burst Access Enable

0	Burst disabled (always full access)
1	For DRAM space access, access in fast page mode

TP Cycle Control

0	1-state precharge cycle is inserted
1	2-state precharge cycle is inserted

Bit	:	7	6	5	4	3	2	1	0
		RFSHE	RCW	RMODE	CMF	CMIE	CKS2	CKS1	CKS0
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Refresh Counter Clock Select

0	0	0	Count operation disabled
		1	Count uses ϕ/2
	1	0	Count uses ϕ/8
		1	Count uses ϕ/32
1	0	0	Count uses ϕ/128
		1	Count uses ϕ/512
	1	0	Count uses ϕ/2048
		1	Count uses ϕ/4096

Compare Match Interrupt Enable

0	Interrupt request (CMI) by CMF flag disabled
1	Interrupt request (CMI) by CMF flag enabled

Compare Match Flag

0	[Clearing condition] When 0 is written to CMF after reading CMF = 1
1	[Setting condition] When RTCNT = RTCOR

Refresh Mode

0	Self-refreshing is not performed in software standby mode
1	Self-refreshing is performed in software standby mode

RAS-CAS Wait

0	Wait state insertion in CAS-before-RAS refreshing disabled $\overline{RAS}$ falls in T_{Rr} cycle
1	One wait state inserted in CAS-before-RAS refreshing $\overline{RAS}$ falls in T_{Rc1} cycle

Refresh Control

0	Refresh control is not performed
1	Refresh control is performed

Bit	:	7	6	5	4	3	2	1	0
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Internal clock count value

RTCOR—Refresh Time Constant Register H'FED9 Bus Controller

Bit	:	7	6	5	4	3	2	1	0
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

Sets the period for compare match operations with RTCNT

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	RAMS	RAM2	RAM1	RAM0
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	—	—	—	—	R/W	R/W	R/W	R/W

RAM Select, Flash Memory Area Select

RAMS	RAM2	RAM1	RAM0	RAM Area	Block Name
0	*	*	*	H'FFDC00 to H'FFEBFF	RAM area, 4 kbytes
1	0	0	0	H'000000 to H'000FFF	EB0 (4 kbytes)
			1	H'001000 to H'001FFF	EB1 (4 kbytes)
		1	0	H'002000 to H'002FFF	EB2 (4 kbytes)
			1	H'003000 to H'003FFF	EB3 (4 kbytes)
	1	0	0	H'004000 to H'004FFF	EB4 (4 kbytes)
			1	H'005000 to H'005FFF	EB5 (4 kbytes)
		1	0	H'006000 to H'006FFF	EB6 (4 kbytes)
			1	H'007000 to H'007FFF	EB7 (4 kbytes)

*: Don't care

Bit	:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MAR0AH	:	—	—	—	—	—	—	—	—								
Initial value	:	0	0	0	0	0	0	0	0	*	*	*	*	*	*	*	*
Read/Write	:	—	—	—	—	—	—	—	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAR0AL	:																
Initial value	:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
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	R/W	R/W

* : Undefined

In short address mode: Specifies transfer source/transfer destination address
In full address mode: Transfer destination address

IOAR0A—I/O Address Register 0A H'FEE4 DMAC

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IOAR0A	:																
Initial value	:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
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	R/W	R/W

* : Undefined

In short address mode: Specifies transfer source/transfer destination address
In full address mode: Not used

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETCR0A	:																
Initial value	:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
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	R/W	R/W

Sequential mode
Idle mode
Normal mode: Transfer counter

Repeat mode: Transfer number storage register; Transfer counter

Block transfer mode: Block size storage register; Block size counter

* : Undefined

MAR0BH—Memory Address Register 0BH **H'FEE8** **DMAC**
MAR0BL—Memory Address Register 0BL **H'FEEA** **DMAC**

Bit	:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MAR0BH	:	—	—	—	—	—	—	—	—								
Initial value	:	0	0	0	0	0	0	0	0	*	*	*	*	*	*	*	*
Read/Write	:	—	—	—	—	—	—	—	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAR0BL	:																
Initial value	:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
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	R/W	R/W

* : Undefined

In short address mode: Specifies transfer source/transfer destination address
In full address mode: Transfer destination address

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IOAR0B	:																
Initial value	:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
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	R/W	R/W

* : Undefined

In short address mode: Specifies transfer source/transfer destination address
In full address mode: Not used

ETCR0B—Transfer Count Register 0B H'FEEE DMAC

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETCR0B	:																
Initial value	:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
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	R/W	R/W

Sequential mode and idle mode	Transfer counter
Repeat mode	Transfer number storage register / Transfer counter
Block transfer mode	Block transfer counter

* : Undefined

Note: Not used in normal mode.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MAR1AH	—	—	—	—	—	—	—	—								
Initial value	0	0	0	0	0	0	0	0	*	*	*	*	*	*	*	*
Read/Write	—	—	—	—	—	—	—	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAR1AL																
Initial value	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
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	R/W	R/W

* : Undefined

In short address mode: Specifies transfer source/transfer destination address
In full address mode: Transfer destination address

IOAR1A—I/O Address Register 1A H'FEF4 DMAC

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IOAR1A																
Initial value	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
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	R/W	R/W

* : Undefined

In short address mode: Specifies transfer source/transfer destination address
In full address mode: Not used

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETCR1A	:																
Initial value	:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
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	R/W	R/W

Sequential mode
Idle mode
Normal mode — Transfer counter

Repeat mode — Transfer number storage register; Transfer counter

Block transfer mode — Block size storage register; Block size counter

* : Undefined

MAR1BH — Memory Address Register 1BH **H'FEF8** **DMAC**
MAR1BL — Memory Address Register 1BL **H'FEFA** **DMAC**

Bit	:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MAR1BH	:	—	—	—	—	—	—	—	—								
Initial value	:	0	0	0	0	0	0	0	0	*	*	*	*	*	*	*	*
Read/Write	:	—	—	—	—	—	—	—	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAR1BL	:																
Initial value	:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
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	R/W	R/W

* : Undefined

In short address mode: Specifies transfer source/transfer destination address
In full address mode: Transfer destination address

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IOAR1B	:																
Initial value	:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
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	R/W	R/W

* : Undefined

In short address mode: Specifies transfer source/transfer destination address
In full address mode: Not used

ETCR1B—Transfer Count Register 1B H'FEFE DMAC

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETCR1B	:																
Initial value	:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
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	R/W	R/W

Sequential mode and idle mode: Transfer counter

Repeat mode: Transfer number storage register | Transfer counter

Block transfer mode: Block transfer counter

* : Undefined

Note: Not used in normal mode.

Bit	:	7	6	5	4	3	2	1	0
DMAWER	:	—	—	—	—	WE1B	WE1A	WE0B	WE0A
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	—	—	—	—	R/W	R/W	R/W	R/W

Write Enable 0A

0	Writes to all bits in DMACR0A, and bits 8, 4, and 0 in DMABCR are disabled
1	Writes to all bits in DMACR0A, and bits 8, 4, and 0 in DMABCR are enabled

Write Enable 0B

0	Writes to all bits in DMACR0B, bits 9, 5, and 1 in DMABCR, and bit 4 in DMATCR are disabled
1	Writes to all bits in DMACR0B, bits 9, 5, and 1 in DMABCR, and bit 4 in DMATCR are enabled

Write Enable 1A

0	Writes to all bits in DMACR1A, and bits 10, 6, and 2 in DMABCR are disabled
1	Writes to all bits in DMACR1A, and bits 10, 6, and 2 in DMABCR are enabled

Write Enable 1B

0	Writes to all bits in DMACR1B, bits 11, 7, and 3 in DMABCR, and bit 5 in DMATCR are disabled
1	Writes to all bits in DMACR1B, bits 11, 7, and 3 in DMABCR, and bit 5 in DMATCR are enabled

Bit	:	7	6	5	4	3	2	1	0
DMATCR	:	—	—	TEE1	TEE0	—	—	—	—
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	—	—	R/W	R/W	—	—	—	—

Transfer End Enable 0

0	$\overline{\text{TEND}}_0$ pin output disabled
1	$\overline{\text{TEND}}_0$ pin output enabled

Transfer End Enable 1

0	$\overline{\text{TEND}}_1$ pin output disabled
1	$\overline{\text{TEND}}_1$ pin output enabled

Full address mode

Bit	15	14	13	12	11	10	9	8
DMACRA	DTSZ	SAID	SAIDE	BLKDIR	BLKE	—	—	—
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Reserved

Only 0 should be written to these bits

Block Direction/Block Enable

0	0	Transfer in normal mode
	1	Transfer in block transfer mode, destination side is block area
1	0	Transfer in normal mode
	1	Transfer in block transfer mode, source side is block area

Source Address Increment/Decrement

0	0	MARA is fixed
	1	MARA is incremented after a data transfer
1	0	MARA is fixed
	1	MARA is decremented after a data transfer

Data Transfer Size

0	Byte-size transfer
1	Word-size transfer

DMACRB :	—	DAID	DAIDE	—	DTF3	DTF2	DTF1	DTF0
Initial value :	0	0	0	0	0	0	0	0
Read/Write :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Reserved
Only 0 should be writtento this bit

Reserved
Only 0 should be writtento this bit

Data Transfer Factor

DTF 3	DTF 2	DTF 1	DTF 0	Block Transfer Mode	Normal Mode
0	0	0	0	—	—
			1	Activated by A/D converter conversion end interrupt	—
		1	0	Activated by $\overline{\text{DREQ}}$ pin falling edge input	Activated by $\overline{\text{DREQ}}$ pin falling edge input
			1	Activated by $\overline{\text{DREQ}}$ pin low-level input	Activated by $\overline{\text{DREQ}}$ pin low-level input
	1	0	0	Activated by SCI channel 0 transmission-data-empty interrupt	—
			1	Activated by SCI channel 0 reception-data-full interrupt	—
		1	0	Activated by SCI channel 1 transmission-data-empty interrupt	Auto-request (cycle steal)
			1	Activated by SCI channel 1 reception-data-full interrupt	Auto-request (burst)
1	0	0	0	Activated by TPU channel 0 compare match/input capture A interrupt	—
			1	Activated by TPU channel 1 compare match/input capture A interrupt	—
		1	0	Activated by TPU channel 2 compare match/input capture A interrupt	—
			1	Activated by TPU channel 3 compare match/input capture A interrupt	—
	1	0	0	Activated by TPU channel 4 compare match/input capture A interrupt	—
			1	Activated by TPU channel 5 compare match/input capture A interrupt	—
		1	0	—	—
			1	—	—

Destination Address Increment/Decrement

0	0	MARB is fixed
	1	MARB is incremented after a data transfer
1	0	MARB is fixed
	1	MARB is decremented after a data transfer

DMACR :	DTS2	DTID	RPE	DTDIR	DTF3	DTF2	DTF1	DTF0
Initial value :	0	0	0	0	0	0	0	0
Read/Write :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data Transfer Direction

0	Dual address mode: Transfer with MAR as source address and IOAR as destination address Single address mode: Transfer with MAR as source address and $\overline{\text{DACK}}$ pin as write strobe
1	Dual address mode: Transfer with IOAR as source address and MAR as destination address Single address mode: Transfer with $\overline{\text{DACK}}$ pin as read strobe and MAR as destination address

Repeat Enable

0	Transfer in sequential mode
1	Transfer in repeat mode or idle mode

Data Transfer Increment/Decrement

0	MAR is incremented after a data transfer
1	MAR is decremented after a data transfer

Data Transfer Size

0	Byte-size transfer
1	Word-size transfer

Data Transfer Factor

				Channel A	Channel B
0	0	0	0	—	
			1	Activated by A/D converter conversion end interrupt	
		1	0	—	Activated by $\overline{\text{DREQ}}$ pin falling edge input
			1	—	Activated by $\overline{\text{DREQ}}$ pin low-level input
	1	0	0	Activated by SCI channel 0 transmission complete interrupt	
			1	Activated by SCI channel 0 reception complete interrupt	
		1	0	Activated by SCI channel 1 transmission complete interrupt	
			1	Activated by SCI channel 1 reception complete interrupt	
1	0	0	0	Activated by TPU channel 0 compare match/input capture A interrupt	
			1	Activated by TPU channel 1 compare match/input capture A interrupt	
		1	0	Activated by TPU channel 2 compare match/input capture A interrupt	
			1	Activated by TPU channel 3 compare match/input capture A interrupt	
	1	0	0	Activated by TPU channel 4 compare match/input capture A interrupt	
			1	Activated by TPU channel 5 compare match/input capture A interrupt	
		1	0	—	
			1	—	

Full address mode

Bit	15	14	13	12	11	10	9	8
DMABCRH	FAE1	FAE0	—	—	DTA1	—	DTA0	—
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Reserved
Only 0 should be writtento these bits

Reserved
Only 0 should be writtento this bit

Reserved
Only 0 should be writtento this bit

Channel 0 Data Transfer Acknowledge

0	Clearing of selected internal interrupt source at time of DMA transfer is disabled
1	Clearing of selected internal interrupt source at time of DMA transfer is enabled

Channel 1 Data Transfer Acknowledge

0	Clearing of selected internal interrupt source at time of DMA transfer is disabled
1	Clearing of selected internal interrupt source at time of DMA transfer is enabled

Channel 0 Full Address Enable

0	Short address mode
1	Full address mode

Channel 1 Full Address Enable

0	Short address mode
1	Full address mode

Bit	7	6	5	4	3	2	1	0
DMABCRL	DTME1	DTE1	DTME0	DTE0	DTIE1B	DTIE1A	DTIE0B	DTIE0A
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Channel 0 Data Transfer Interrupt Enable A

0	Transfer end interrupt disabled
1	Transfer end interrupt enabled

Channel 0 Data Transfer Interrupt Enable B

0	Transfer suspended interrupt disabled
1	Transfer suspended interrupt enabled

Channel 1 Data Transfer Interrupt Enable A

0	Transfer end interrupt disabled
1	Transfer end interrupt enabled

Channel 1 Data Transfer Interrupt Enable B

0	Transfer suspended interrupt disabled
1	Transfer suspended interrupt enabled

Channel 0 Data Transfer Enable

0	Data transfer disabled
1	Data transfer enabled

Channel 0 Data Transfer Master Enable

0	Data transfer disabled. In burst mode, cleared to 0 by an NMI interrupt
1	Data transfer enabled

Channel 1 Data Transfer Enable

0	Data transfer disabled
1	Data transfer enabled

Channel 1 Data Transfer Master Enable

0	Data transfer disabled. In burst mode, cleared to 0 by an NMI interrupt
1	Data transfer enabled

Short address mode

Bit	:	15	14	13	12	11	10	9	8
DMABCRH	:	FAE1	FAE0	SAE1	SAE0	DTA1B	DTA1A	DTA0B	DTA0A
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Channel 0A Data Transfer Acknowledge

0	Clearing of selected internal interrupt source at time of DMA transfer is disabled
1	Clearing of selected internal interrupt source at time of DMA transfer is enabled

Channel 0B Data Transfer Acknowledge

0	Clearing of selected internal interrupt source at time of DMA transfer is disabled
1	Clearing of selected internal interrupt source at time of DMA transfer is enabled

Channel 1A Data Transfer Acknowledge

0	Clearing of selected internal interrupt source at time of DMA transfer is disabled
1	Clearing of selected internal interrupt source at time of DMA transfer is enabled

Channel 1B Data Transfer Acknowledge

0	Clearing of selected internal interrupt source at time of DMA transfer is disabled
1	Clearing of selected internal interrupt source at time of DMA transfer is enabled

Channel 0B Single Address Enable

0	Transfer in dual address mode
1	Transfer in single address mode

Channel 1B Single Address Enable

0	Transfer in dual address mode
1	Transfer in single address mode

Channel 0 Full Address Enable

0	Short address mode
1	Full address mode

Channel 1 Full Address Enable

0	Short address mode
1	Full address mode

Bit	7	6	5	4	3	2	1	0
DMABCRL	DTE1B	DTE1A	DTE0B	DTE0A	DTIE1B	DTIE1A	DTIE0B	DTIE0A
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Channel 0A Data Transfer Interrupt Enable

0	Transfer end interrupt disabled
1	Transfer end interrupt enabled

Channel 0B Data Transfer Interrupt Enable

0	Transfer end interrupt disabled
1	Transfer end interrupt enabled

Channel 1A Data Transfer Interrupt Enable

0	Transfer end interrupt disabled
1	Transfer end interrupt enabled

Channel 1B Data Transfer Interrupt Enable

0	Transfer end interrupt disabled
1	Transfer end interrupt enabled

Channel 0A Data Transfer Enable

0	Data transfer disabled
1	Data transfer enabled

Channel 0B Data Transfer Enable

0	Data transfer disabled
1	Data transfer enabled

Channel 1A Data Transfer Enable

0	Data transfer disabled
1	Data transfer enabled

Channel 1B Data Transfer Enable

0	Data transfer disabled
1	Data transfer enabled

ISCRH

Bit	:	15	14	13	12	11	10	9	8
		IRQ7SCB	IRQ7SCA	IRQ6SCB	IRQ6SCA	IRQ5SCB	IRQ5SCA	IRQ4SCB	IRQ4SCA
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IRQ7 to IRQ4 Sense Control

ISCRL

Bit	:	7	6	5	4	3	2	1	0
		IRQ3SCB	IRQ3SCA	IRQ2SCB	IRQ2SCA	IRQ1SCB	IRQ1SCA	IRQ0SCB	IRQ0SCA
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IRQ3 to IRQ0 Sense Control

IRQnSCB	IRQnSCA	Interrupt Request Generation
0	0	$\overline{IRQ}_n$ input low level
	1	Falling edge of $\overline{IRQ}_n$ input
1	0	Rising edge of $\overline{IRQ}_n$ input
	1	Both falling and rising edges of $\overline{IRQ}_n$ input

(n = 7 to 0)

Bit	:	7	6	5	4	3	2	1	0
		IRQ7E	IRQ6E	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IRQ0E
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IRQn Enable

0	IRQn interrupt disabled
1	IRQn interrupt enabled

(n = 7 to 0)

ISR—IRQ Status Register **H'FF2F** **Interrupt Controller**

Bit	:	7	6	5	4	3	2	1	0
		IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Indicate the status of IRQ7 to IRQ0 interrupt requests

Note: * Can only be written with 0 for flag clearing.

Bit	:	7	6	5	4	3	2	1	0
		DTCE7	DTCE6	DTCE5	DTCE4	DTCE3	DTCE2	DTCE1	DTCE0
Initial value :		0	0	0	0	0	0	0	0
Read/Write :		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

DTC Activation Enable

0	DTC activation by this interrupt is disabled [Clearing conditions] • When the DISEL bit is 1 and 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 ended

Correspondence between Interrupt Sources and DTCER

	Bits							
Register	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DTCERA	IRQ0	IRQ1	IRQ2	IRQ3	IRQ4	IRQ5	IRQ6	IRQ7
DTCERB	—	ADI	TGI0A	TGI0B	TGI0C	TGI0D	TGI1A	TGI1B
DTCERC	TGI2A	TGI2B	TGI3A	TGI3B	TGI3C	TGI3D	TGI4A	TGI4B
DTCERD	—	—	TGI5A	TGI5B	CMIA0	CMIB0	CMIA1	CMIB1
DTCERE	DMTEND0A	DMTEND0B	DMTEND1A	DMTEND1B	RXI0	TXI0	RXI1	TXI1
DTCERF	RXI2	TXI2	—	—	—	—	—	—

Note: For DTCE bit setting, read/write operations must be performed using bit-manipulation instructions such as BSET and BCLR. For the initial setting only, however, when multiple activation sources are set at one time, it is possible to disable interrupts and write after executing a dummy read on the relevant register.

Bit :	7	6	5	4	3	2	1	0
	SWDTE	DTVEC6	DTVEC5	DTVEC4	DTVEC3	DTVEC2	DTVEC1	DTVEC0
Initial value :	0	0	0	0	0	0	0	0
Read/Write :	R/W	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Sets vector number for DTC software activation

DTC Software Activation Enable

0	DTC software activation is disabled [Clearing conditions] • When the DISEL bit is 0 and the specified number of transfers have not ended • When SWDTEND is requested to the CPU, then 0 is written to the SWDTE bit
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

Note: * DTVEC6 to DTVEC0 bits can be written to when SWDTE = 0.

Bit :	7	6	5	4	3	2	1	0
	SSBY	STS2	STS1	STS0	OPE	—	—	IRQ37S
Initial value :	0	0	0	0	1	0	0	0
Read/Write :	R/W	R/W	R/W	R/W	R/W	—	—	R/W

IRQ37 Software Standby Clear Select

0	$\overline{IRQ}_3$ to $\overline{IRQ}_7$ cannot be used as software standby mode clearing sources
1	$\overline{IRQ}_3$ to $\overline{IRQ}_7$ can be used as software standby mode clearing sources

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 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

Bit	7	6	5	4	3	2	1	0
	—	—	INTM1	INTM0	NMIEG	LWROD	IRQPAS	RAME
Initial value	0	0	0	0	0	0	0	1
Read/Write	R/W	—	R/W	R/W	R/W	R/W	R/W	R/W

RAM Enable

0	On-chip RAM disabled
1	On-chip RAM enabled

IRQ Port Switching Select

0	$\overline{IRQ_4}$ to $\overline{IRQ_7}$ can be input from $P9_4$ to $P9_7$
1	$\overline{IRQ_4}$ to $\overline{IRQ_7}$ can be input from $P5_0$ to $P5_3$

Note: $\overline{IRQ_4}$ to $\overline{IRQ_7}$ input is always performed from only one of the ports.

$\overline{LWR}$ Output Disable

0	PF_3 is designated as $\overline{LWR}$ output pin
1	PF_3 is designated as I/O port, and does not function as $\overline{LWR}$ output pin

NMI Input Edge Select

0	Falling edge
1	Rising edge

Interrupt Control Mode Selection

0	0	Interrupt control mode 0
	1	Setting prohibited
1	0	Interrupt control mode 2
	1	Setting prohibited

Reserved
Only 0 should be written to this bit

Bit	7	6	5	4	3	2	1	0
	PSTOP	—	DIV	—	—	SCK2	SCK1	SCK0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	—	—	R/W	R/W	R/W

Division Ratio Select

Reserved
Only 0 should be written to this bit

System Clock Select

SCK2	SCK1	SCK0	DIV = 0	DIV = 1
0	0	0	Bus master is in high-speed mode	Bus master is in high-speed mode
		1	Medium-speed clock is ϕ/2	Clock supplied to entire chip is ϕ/2
	1	0	Medium-speed clock is ϕ/4	Clock supplied to entire chip is ϕ/4
		1	Medium-speed clock is ϕ/8	Clock supplied to entire chip is ϕ/8
1	0	0	Medium-speed clock is ϕ/16	—
		1	Medium-speed clock is ϕ/32	—
	1	—	—	—

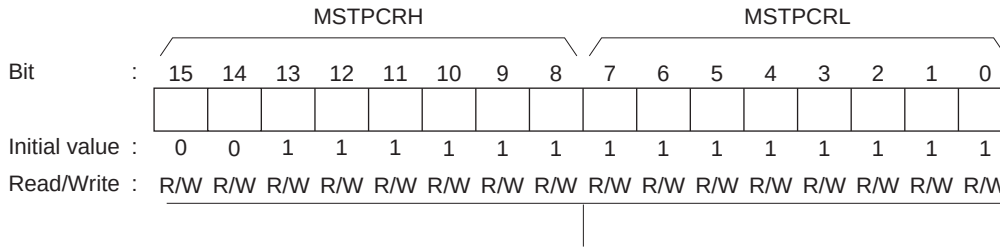
ϕ Clock Output Control

PSTOP	Normal Operation	Sleep Mode	Software Standby Mode	Hardware Standby Mode
0	ϕ output	ϕ output	Fixed high	High impedance
1	Fixed high	Fixed high	Fixed high	High impedance

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	—	MDS2	MDS1	MDS0
Initial value	:	1	0	0	0	0	—*	—*	—*
Read/Write	:	—	—	—	—	—	R	R	R

Current mode pin operating mode

Note: * Determined by pins MD_2 to MD_0



Specifies module stop mode

0	Module stop mode cleared
1	Module stop mode set

MSTP Bits and On-Chip Supporting Modules

Register	Bits	Module
MSTPCRH	MSTP15	DMAC
	MSTP14	DTC
	MSTP13	TPU
	MSTP12	8-bit timer
	MSTP11	PPG
	MSTP10	DA0,1
	MSTP9	A/D
	MSTP8	—
MSTPCRL	MSTP7	SCI2
	MSTP6	SCI1
	MSTP5	SCI0
	MSTP4	DA2, 3
	MSTP3	—
	MSTP2	—
	MSTP1	—
	MSTP0	—

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	FLSHE	—	—	—
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	—	—	—	—	R/W	—	—	— (R/W)

In H8S/2339 only this bit is R/W, and should only be written with 0

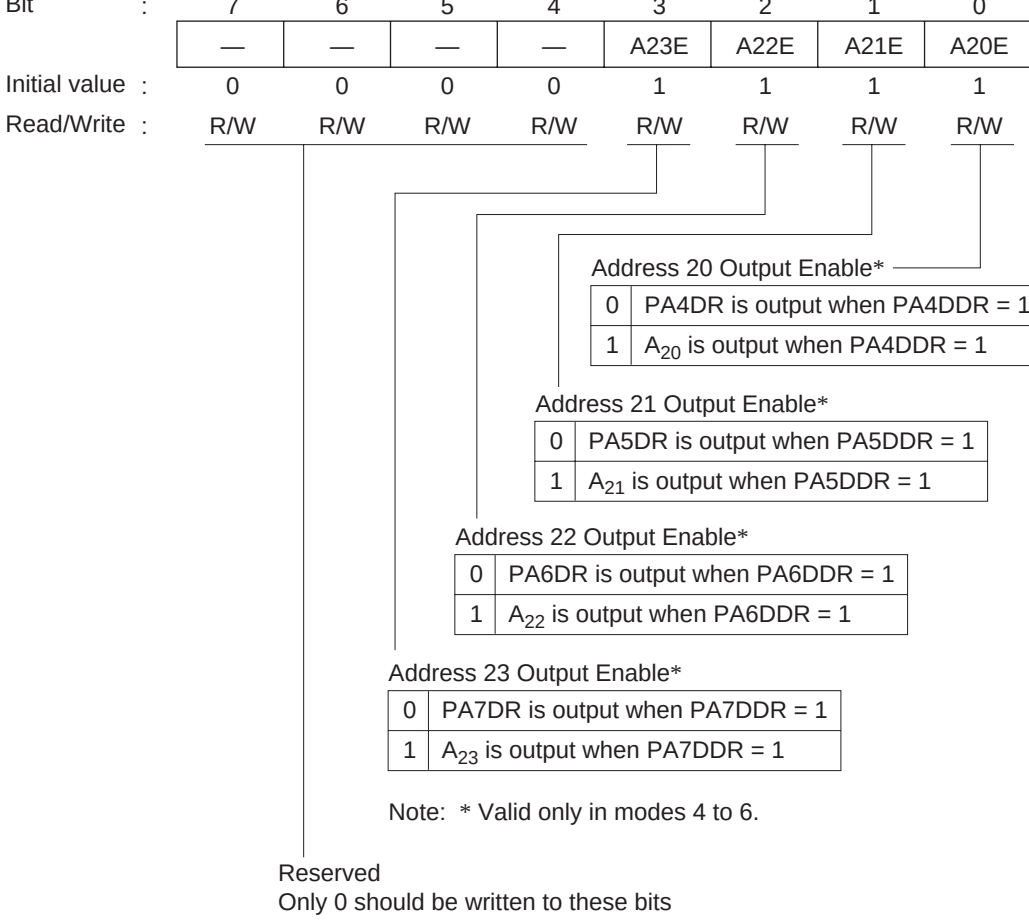
Flash Memory Control Register Enable

0	Flash control register is not selected for addresses H'FFFFC8 to HFFFFCB
1	Flash control register is selected for addresses H'FFFFC8 to HFFFFCB

Reserved Register H'FF44

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	—	—	—	—
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	—	—	R/W	—	—	—	—	—

Reserved
Only 0 should be written to these bits


Bit : 7 6 5 4 3 2 1 0
— — — — A23E A22E A21E A20E
Initial value : 0 0 0 0 1 1 1 1
Read/Write : R/W R/W R/W R/W R/W R/W R/W R/W
Address 20 Output Enable*
0 PA4DR is output when PA4DDR = 1
1 A20 is output when PA4DDR = 1
Address 21 Output Enable*
0 PA5DR is output when PA5DDR = 1
1 A21 is output when PA5DDR = 1
Address 22 Output Enable*
0 PA6DR is output when PA6DDR = 1
1 A22 is output when PA6DDR = 1
Address 23 Output Enable*
0 PA7DR is output when PA7DDR = 1
1 A23 is output when PA7DDR = 1
Note: * Valid only in modes 4 to 6.
Reserved
Only 0 should be written to these bits

Bit :	7	6	5	4	3	2	1	0
	G3CMS1	G3CMS0	G2CMS1	G2CMS0	G1CMS1	G1CMS0	G0CMS1	G0CMS0
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

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

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

Output Trigger for Pulse Output Group 2

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

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

Bit	7	6	5	4	3	2	1	0
	G3INV	G2INV	G1INV	G0INV	G3NOV	G2NOV	G1NOV	G0NOV
Initial value	1	1	1	1	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Pulse Output Group n Normal/Non-Overlap Operation Select

0	Normal operation in pulse output group n (output values updated at compare match A in the selected TPU channel)
1	Non-overlapping operation in pulse output group n (independent 1 and 0 output at compare match A or B in the selected TPU channel)

(n = 3 to 0)

Pulse Output Group n Direct/Inverse Output

0	Inverse output for pulse output group n (low-level output at pin for a 1 in PODRH)
1	Direct output for pulse output group n (high-level output at pin for a 1 in PODRH)

(n = 3 to 0)

NDERH

Bit	7	6	5	4	3	2	1	0
	NDER15	NDER14	NDER13	NDER12	NDER11	NDER10	NDER9	NDER8
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Pulse Output Enable/Disable

0	Pulse outputs PO_{15} to PO_8 are disabled
1	Pulse outputs PO_{15} to PO_8 are enabled

NDERL

Bit	7	6	5	4	3	2	1	0
	NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1	NDER0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Pulse Output Enable/Disable

0	Pulse outputs PO_7 to PO_0 are disabled
1	Pulse outputs PO_7 to PO_0 are enabled

PODRH

Bit	7	6	5	4	3	2	1	0
	POD15	POD14	POD13	POD12	POD11	POD10	POD9	POD8
Initial value :	0	0	0	0	0	0	0	0
Read/Write :	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Stores output data for use in pulse output

PODRL

Bit	7	6	5	4	3	2	1	0
	POD7	POD6	POD5	POD4	POD3	POD2	POD1	POD0
Initial value :	0	0	0	0	0	0	0	0
Read/Write :	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Stores output data for use in pulse output

Note: * A bit that has been set for pulse output by NDER is read-only.

(1) When pulse output group output triggers are the same

(a) Address: H'FF4C

Bit	7	6	5	4	3	2	1	0
	NDR15	NDR14	NDR13	NDR12	NDR11	NDR10	NDR9	NDR8
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Stores the next data for pulse output groups 3 and 2

(b) Address: H'FF4E

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	1	1	1	1	1	1	1	1
Read/Write	—	—	—	—	—	—	—	—

(2) When pulse output group output triggers are different

(a) Address: H'FF4C

Bit	7	6	5	4	3	2	1	0
	NDR15	NDR14	NDR13	NDR12	—	—	—	—
Initial value	0	0	0	0	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	—	—	—	—

Stores the next data for pulse output group 3

(b) Address: H'FF4E

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	NDR11	NDR10	NDR9	NDR8
Initial value	1	1	1	1	0	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W

Stores the next data for pulse output group 2

(1) When pulse output group output triggers are the same

(a) Address: H'FF4D

Bit	7	6	5	4	3	2	1	0
	NDR7	NDR6	NDR5	NDR4	NDR3	NDR2	NDR1	NDR0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Stores the next data for pulse output groups 1 and 0

(b) Address: H'FF4F

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	1	1	1	1	1	1	1	1
Read/Write	—	—	—	—	—	—	—	—

(2) When pulse output group output triggers are different

(a) Address: H'FF4D

Bit	7	6	5	4	3	2	1	0
	NDR7	NDR6	NDR5	NDR4	—	—	—	—
Initial value	0	0	0	0	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	—	—	—	—

Stores the next data for pulse output group 1

(b) Address: H'FF4F

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	NDR3	NDR2	NDR1	NDR0
Initial value	1	1	1	1	0	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W

Stores the next data for pulse output group 0

Bit	:	7	6	5	4	3	2	1	0
		P17	P16	P15	P14	P13	P12	P11	P10
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
Read/Write	:	R	R	R	R	R	R	R	R

State of port 1 pins

Note: * Determined by the state of pins $P1_7$ to $P1_0$.

PORT2—Port 2 Register **H'FF51** **Port 2**

Bit	:	7	6	5	4	3	2	1	0
		P27	P26	P25	P24	P23	P22	P21	P20
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
Read/Write	:	R	R	R	R	R	R	R	R

State of port 2 pins

Note: * Determined by the state of pins $P2_7$ to $P2_0$.

PORT3—Port 3 Register **H'FF52** **Port 3**

Bit	:	7	6	5	4	3	2	1	0
		—	—	P35	P34	P33	P32	P31	P30
Initial value	:	Undefined	Undefined	—*	—*	—*	—*	—*	—*
Read/Write	:	—	—	R	R	R	R	R	R

State of port 3 pins

Note: * Determined by the state of pins $P3_5$ to $P3_0$.

Bit	:	7	6	5	4	3	2	1	0
		P47	P46	P45	P44	P43	P42	P41	P40
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
Read/Write	:	R	R	R	R	R	R	R	R

State of port 4 pins

Note: * Determined by the state of pins $P4_7$ to $P4_0$.

PORT5—Port 5 Register **H'FF54** **Port 5**

Bit	:	7	6	5	4	3	2	1	0
		P57	P56	P55	P54	P53	P52	P51	P50
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
Read/Write	:	R	R	R	R	R	R	R	R

State of port 5 pins

Note: * Determined by the state of pins $P5_7$ to $P5_0$.

PORT6—Port 6 Register **H'FF55** **Port 6**

Bit	:	7	6	5	4	3	2	1	0
		P67	P66	P65	P64	P63	P62	P61	P60
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
Read/Write	:	R	R	R	R	R	R	R	R

State of port 6 pins

Note: * Determined by the state of pins $P6_7$ to $P6_0$.

Bit :	7	6	5	4	3	2	1	0
	—	—	P75	P74	P73	P72	P71	P70
Initial value :	Undefined	Undefined	—*	—*	—*	—*	—*	—*
Read/Write :	—	—	R	R	R	R	R	R

State of port 7 pins

Note: * Determined by the state of pins $P7_5$ to $P7_0$.

PORT8—Port 8 Register **H'FF57** **Port 8**

Bit :	7	6	5	4	3	2	1	0
	—	P86	P85	P84	P83	P82	P81	P80
Initial value :	Undefined	—*	—*	—*	—*	—*	—*	—*
Read/Write :	—	R	R	R	R	R	R	R

State of port 8 pins

Note: * Determined by the state of pins $P8_6$ to $P8_0$.

PORT9—Port 9 Register **H'FF58** **Port 9**

Bit :	7	6	5	4	3	2	1	0
	P97	P96	P95	P94	P93	P92	—	—
Initial value :	—*	—*	—*	—*	—*	—*	Undefined	Undefined
Read/Write :	R	R	R	R	R	R	—	—

State of port 9 pins

Note: * Determined by the state of pins $P9_7$ to $P9_2$.

Bit	:	7	6	5	4	3	2	1	0
		PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
Read/Write	:	R	R	R	R	R	R	R	R

State of port A pins

Note: * Determined by the state of pins PA_7 to PA_0.

PORTB—Port B Register **H'FF5A** **Port B**

Bit	:	7	6	5	4	3	2	1	0
		PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
Read/Write	:	R	R	R	R	R	R	R	R

State of port B pins

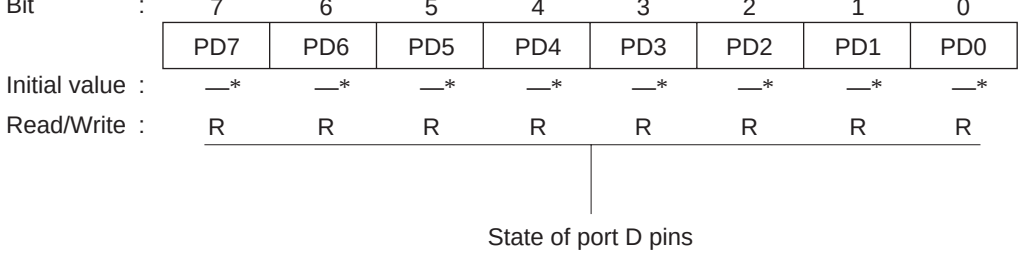
Note: * Determined by the state of pins PB_7 to PB_0.

PORTC—Port C Register **H'FF5B** **Port C**

Bit	:	7	6	5	4	3	2	1	0
		PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
Read/Write	:	R	R	R	R	R	R	R	R

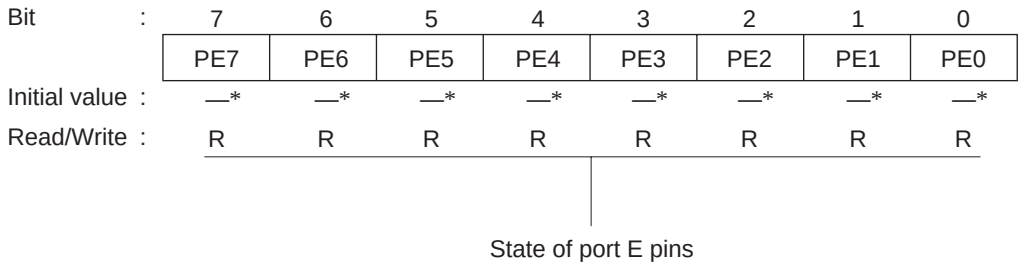
State of port C pins

Note: * Determined by the state of pins PC_7 to PC_0.



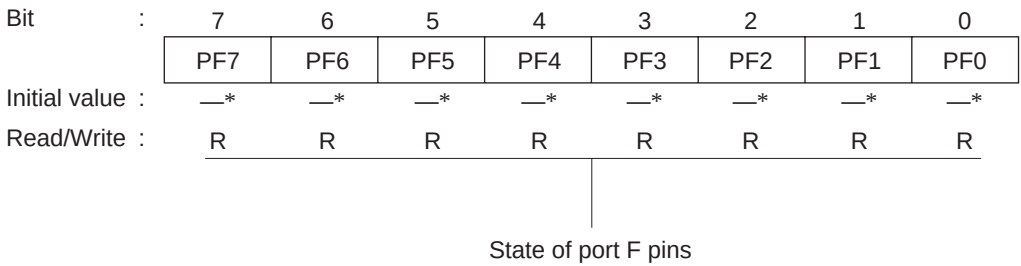
Note: * Determined by the state of pins PD_7 to PD_0.

PORTE—Port E Register **H'FF5D** **Port E**



Note: * Determined by the state of pins PE_7 to PE_0.

PORTF—Port F Register **H'FF5E** **Port F**



Note: * Determined by the state of pins PF_7 to PF_0.

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	PG4	PG3	PG2	PG1	PG0
Initial value	:	Undefined	Undefined	Undefined	—*	—*	—*	—*	—*
Read/Write	:	—	—	—	R	R	R	R	R

State of port G pins

Note: * Determined by the state of pins PG_4 to PG_0.

P1DR—Port 1 Data Register **H'FF60** **Port 1**

Bit	:	7	6	5	4	3	2	1	0
		P17DR	P16DR	P15DR	P14DR	P13DR	P12DR	P11DR	P10DR
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Stores output data for port 1 pins ($P1_7$ to $P1_0$)

P2DR—Port 2 Data Register **H'FF61** **Port 2**

Bit	:	7	6	5	4	3	2	1	0
		P27DR	P26DR	P25DR	P24DR	P23DR	P22DR	P21DR	P20DR
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Stores output data for port 2 pins ($P2_7$ to $P2_0$)

Bit	:	7	6	5	4	3	2	1	0
		—	—	P35DR	P34DR	P33DR	P32DR	P31DR	P30DR
Initial value	:	Undefined	Undefined	0	0	0	0	0	0
Read/Write	:	—	—	R/W	R/W	R/W	R/W	R/W	R/W

Stores output data for port 3 pins ($P3_5$ to $P3_0$)

P5DR—Port 5 Data Register **H'FF64** **Port 5**

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	P53DR	P52DR	P51DR	P50DR
Initial value	:	Undefined	Undefined	Undefined	Undefined	0	0	0	0
Read/Write	:	—	—	—	—	R/W	R/W	R/W	R/W

Stores output data for port 5 pins ($P5_3$ to $P5_0$)

P6DR—Port 6 Data Register **H'FF65** **Port 6**

Bit	:	7	6	5	4	3	2	1	0
		P67DR	P66DR	P65DR	P64DR	P63DR	P62DR	P61DR	P60DR
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Stores output data for port 6 pins ($P6_7$ to $P6_0$)

Bit	:	7	6	5	4	3	2	1	0
		—	—	P75DR	P74DR	P73DR	P72DR	P71DR	P70DR
Initial value	:	Undefined	Undefined	0	0	0	0	0	0
Read/Write	:	—	—	R/W	R/W	R/W	R/W	R/W	R/W

Stores output data for port 7 pins ($P7_5$ to $P7_0$)

P8DR—Port 8 Data Register **H'FF67** **Port 8**

Bit	:	7	6	5	4	3	2	1	0
		—	P86DR	P85DR	P84DR	P83DR	P82DR	P81DR	P80DR
Initial value	:	Undefined	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 8 pins ($P8_6$ to $P8_0$)

P9DR—Port 9 Data Register **H'FF68** **Port 9**

Bit	:	7	6	5	4	3	2	1	0
		P97DR	P96DR	P95DR	P94DR	P93DR	P92DR	—	—
Initial value	:	0	0	0	0	0	0	Undefined	Undefined
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	—	—

Stores output data for port 9 pins ($P9_7$ to $P9_2$)

Bit	:	7	6	5	4	3	2	1	0
		PA7DR	PA6DR	PA5DR	PA4DR	PA3DR	PA2DR	PA1DR	PA0DR
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Stores output data for port A pins (PA_7 to PA_0)

PBDR—Port B Data Register **H'FF6A** **Port B**

Bit	:	7	6	5	4	3	2	1	0
		PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Stores output data for port B pins (PB_7 to PB_0)

PCDR—Port C Data Register **H'FF6B** **Port C**

Bit	:	7	6	5	4	3	2	1	0
		PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Stores output data for port C pins (PC_7 to PC_0)

Bit	:	7	6	5	4	3	2	1	0
		PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Stores output data for port D pins (PD_7 to PD_0)

PEDR—Port E Data Register **H'FF6D** **Port E**

Bit	:	7	6	5	4	3	2	1	0
		PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Stores output data for port E pins (PE_7 to PE_0)

PFDR—Port F Data Register **H'FF6E** **Port F**

Bit	:	7	6	5	4	3	2	1	0
		PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1DR	PF0DR
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Stores output data for port F pins (PF_7 to PF_0)

Bit	7	6	5	4	3	2	1	0
	—	—	—	PG4DR	PG3DR	PG2DR	PG1DR	PG0DR
Initial value	Undefined	Undefined	Undefined	0	0	0	0	0
Read/Write	—	—	—	R/W	R/W	R/W	R/W	R/W

Stores output data for port G pins (PG_4 to PG_0)

PAPCR—Port A MOS Pull-Up Control Register H'FF70 **Port A**

Bit	7	6	5	4	3	2	1	0
	PA7PCR	PA6PCR	PA5PCR	PA4PCR	PA3PCR	PA2PCR	PA1PCR	PA0PCR
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Controls the MOS input pull-up function incorporated into port A on a bit-by-bit basis

PBPCR—Port B MOS Pull-Up Control Register H'FF71 **Port B**

Bit	7	6	5	4	3	2	1	0
	PB7PCR	PB6PCR	PB5PCR	PB4PCR	PB3PCR	PB2PCR	PB1PCR	PB0PCR
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Controls the MOS input pull-up function incorporated into port B on a bit-by-bit basis

Bit	:	7	6	5	4	3	2	1	0
		PC7PCR	PC6PCR	PC5PCR	PC4PCR	PC3PCR	PC2PCR	PC1PCR	PC0PCR
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Controls the MOS input pull-up function incorporated into port C on a bit-by-bit basis

PDPCR—Port D MOS Pull-Up Control Register H'FF73 **Port D**

Bit	:	7	6	5	4	3	2	1	0
		PD7PCR	PD6PCR	PD5PCR	PD4PCR	PD3PCR	PD2PCR	PD1PCR	PD0PCR
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Controls the MOS input pull-up function incorporated into port D on a bit-by-bit basis

PEPCR—Port E MOS Pull-Up Control Register H'FF74 **Port E**

Bit	:	7	6	5	4	3	2	1	0
		PE7PCR	PE6PCR	PE5PCR	PE4PCR	PE3PCR	PE2PCR	PE1PCR	PE0PCR
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Controls the MOS input pull-up function incorporated into port E on a bit-by-bit basis

Bit	:	7	6	5	4	3	2	1	0
		—	—	P35ODR	P34ODR	P33ODR	P32ODR	P31ODR	P30ODR
Initial value	:	Undefined	Undefined	0	0	0	0	0	0
Read/Write	:	—	—	R/W	R/W	R/W	R/W	R/W	R/W

Controls the PMOS on/off status for each port 3 pin ($P3_5$ to $P3_0$)

PAODR—Port A Open Drain Control Register H'FF77 Port A

Bit	:	7	6	5	4	3	2	1	0
		PA7ODR	PA6ODR	PA5ODR	PA4ODR	PA3ODR	PA2ODR	PA1ODR	PA0ODR
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Controls the PMOS on/off status for each port A pin (PA_7 to PA_0)

Bit :	7	6	5	4	3	2	1	0
	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0
Initial value :	0	0	0	0	0	0	0	0
Read/Write :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock Select

0	0	ϕ clock
	1	ϕ/4 clock
1	0	ϕ/16 clock
	1	ϕ/64 clock

Multiprocessor Mode

0	Multiprocessor function disabled
1	Multiprocessor format selected

Stop Bit Length

0	1 stop bit
1	2 stop bits

Parity Mode

0	Even parity
1	Odd parity

Parity Enable

0	Parity bit addition and checking disabled
1	Parity bit addition and checking enabled

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

Bit :	7	6	5	4	3	2	1	0
	GM	BLK	PE	O/$\overline{E}$	BCP1	BCP0	CKS1	CKS0
Initial value :	0	0	0	0	0	0	0	0
Read/Write :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock Select

0	0	ϕ clock
	1	ϕ/4 clock
1	0	ϕ/16 clock
	1	ϕ/64 clock

Base Clock Pulse

BCP1	BCP0	Base Clock Pulse
0	0	32 clocks
	1	64 clocks
1	0	372 clocks
	1	256 clocks

Parity Mode

0	Even parity
1	Odd parity

Parity Enable
(Set to 1 when using the smart card interface)

0	Setting prohibited
1	Parity bit addition and checking enabled

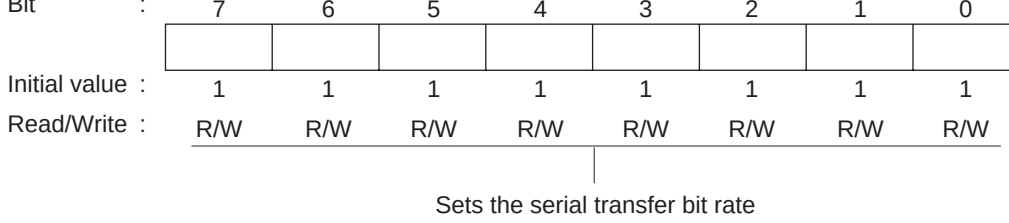
Block Transfer Mode Select

0	Normal smart card interface mode
1	Block transfer mode

GSM Mode

0	Normal smart card interface mode operation • TEND flag generated 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 generated 11.0 etu after beginning of start bit • Fixed high/low-level control possible (set in SCR) in addition to clock output on/off control

Note: etu: Elementary time unit (time for transfer of one bit)



Note: For details, see section 14.2.8, Bit Rate Register (BRR).

Bit	7	6	5	4	3	2	1	0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock Enable

0	0	Asynchronous mode	Internal clock/SCK pin functions as I/O port
		Synchronous mode	Internal clock/SCK pin functions as serial clock output
	1	Asynchronous mode	Internal clock/SCK pin functions as clock output*1
		Synchronous mode	Internal clock/SCK pin functions as serial clock output
1	0	Asynchronous mode	External clock/SCK pin functions as clock input*2
		Synchronous mode	External clock/SCK pin functions as serial clock input
	1	Asynchronous mode	External clock/SCK pin functions as clock input*2
		Synchronous mode	External clock/SCK pin functions as serial clock input

Notes: 1. Outputs a clock of the same frequency as the bit rate.
2. Inputs a clock with a frequency 16 times the bit rate.

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-data-full interrupt (RXI) requests, receive-error interrupt (ERI) requests, and setting of the RDRF, FER, and ORER flags in SSR are disabled until data with the multiprocessor bit set 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) request disabled
1	Receive-data-full interrupt (RXI) request and receive-error interrupt (ERI) request enabled

Transmit Interrupt Enable

0	Transmit-data-empty interrupt (TXI) request disabled
1	Transmit-data-empty interrupt (TXI) request enabled

Bit :	7	6	5	4	3	2	1	0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
Initial value :	0	0	0	0	0	0	0	0
Read/Write :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock Enable

SCMR	SMR	SCR setting		SCK pin function
SMIF	GM	CKE1	CKE0	
0	See SCI specification			
1	0	0	0	Operates as port I/O pin
1	0	0	1	Clock output as SCK output pin
1	1	0	0	Fixed-low output as SCK output pin
1	1	0	1	Clock output as SCK output pin
1	1	1	0	Fixed-high output as SCK output pin
1	1	1	1	Clock output as SCK 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-data-full interrupt (RXI) requests, receive error interrupt (ERI) requests, and setting of the RDRF, FER, and ORER flags in SSR are disabled until data with the multiprocessor bit set to 1 is received

Receive Enable

0	Reception disabled
1	Reception enabled

Transmit Enable

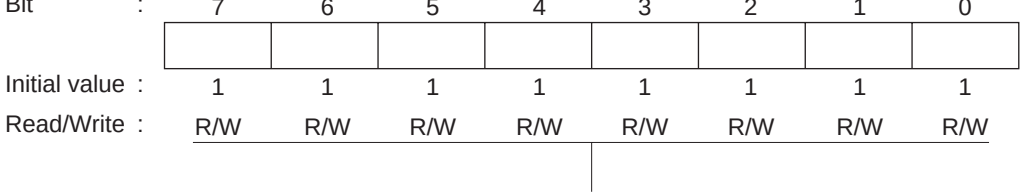
0	Transmission disabled
1	Transmission enabled

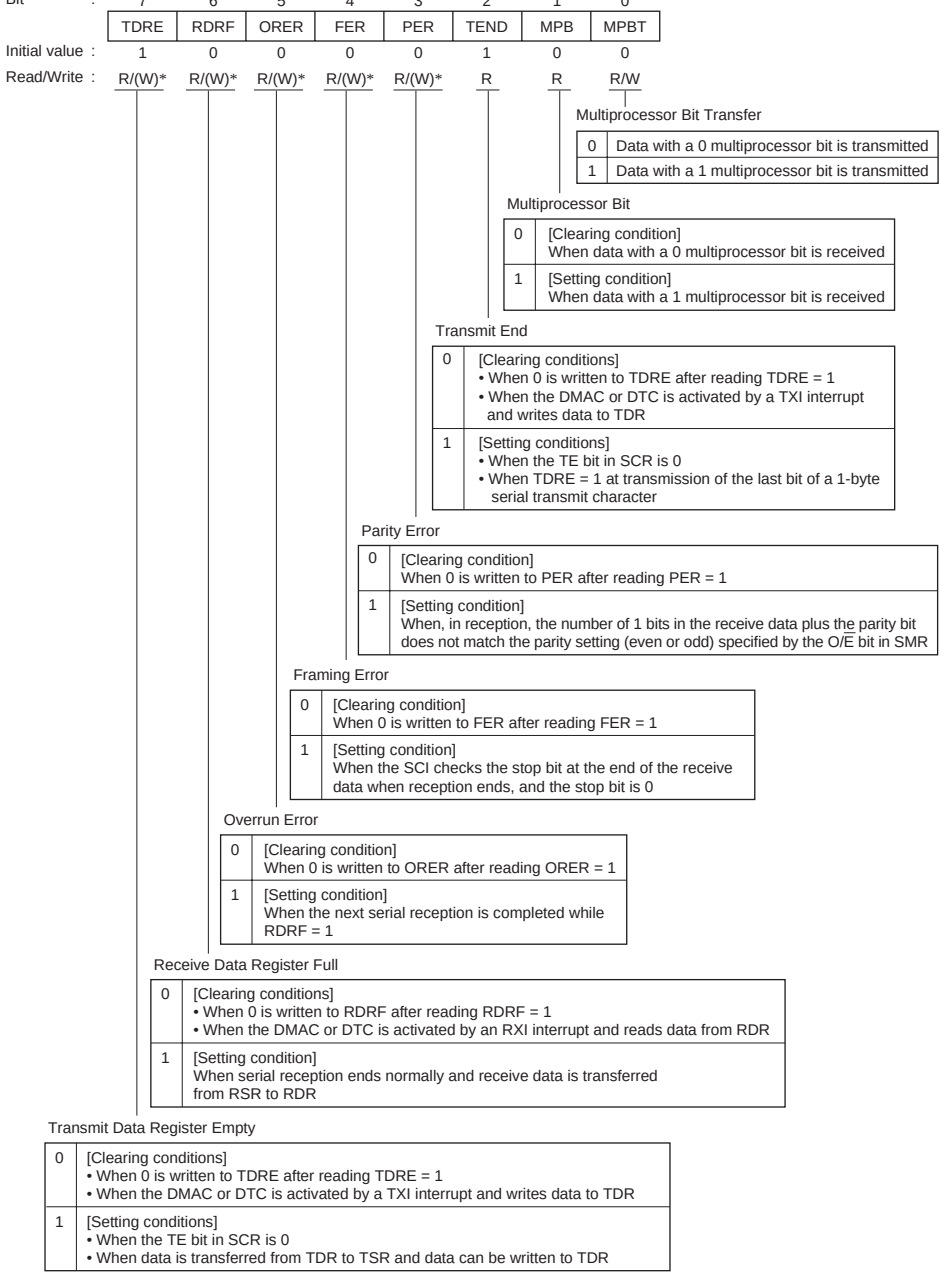
Receive Interrupt Enable

0	Receive-data-full interrupt (RXI) request and receive-error interrupt (ERI) request disabled
1	Receive-data-full interrupt (RXI) request and receive-error interrupt (ERI) request enabled

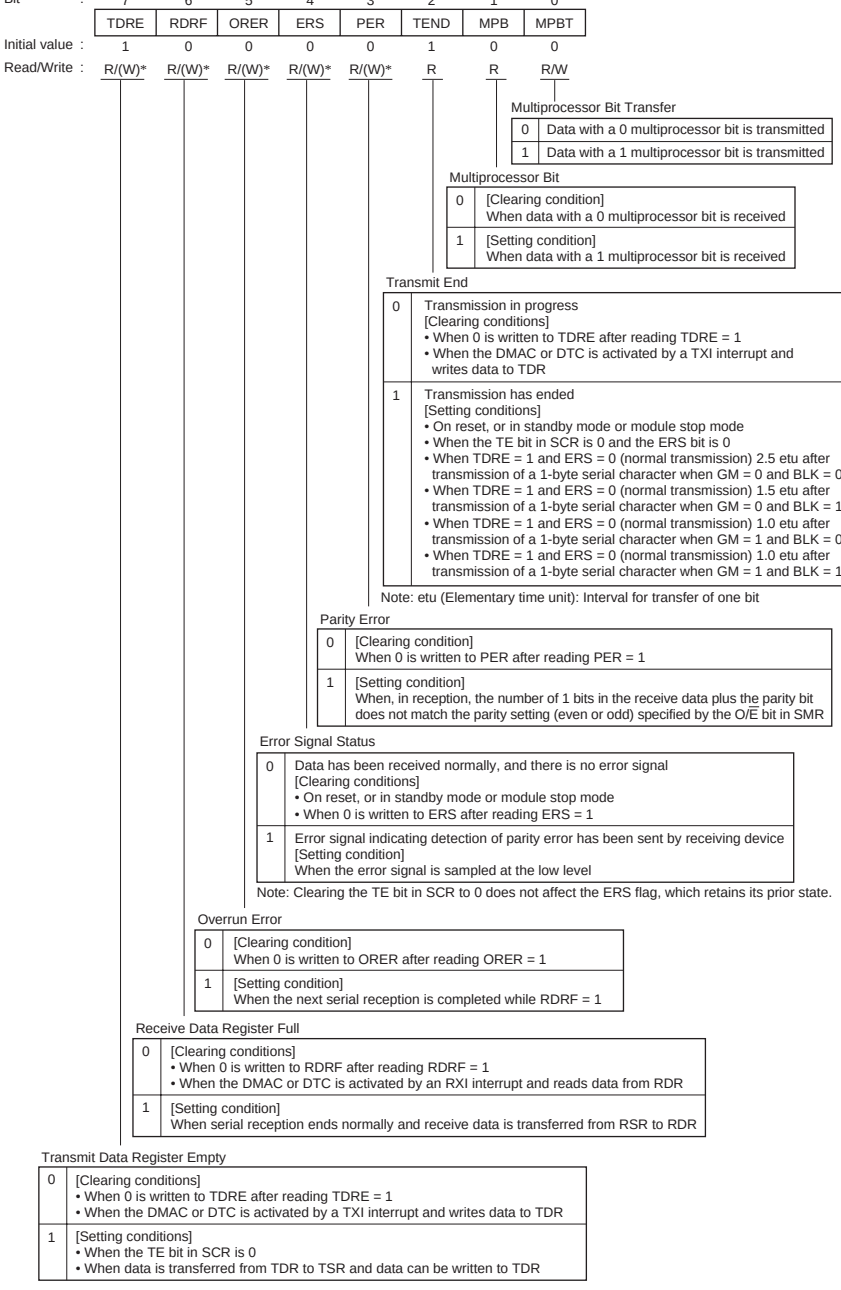
Transmit Interrupt Enable

0	Transmit-data-empty interrupt (TXI) request disabled
1	Transmit-data-empty interrupt (TXI) request enabled


Bit :
7
6
5
4
3
2
1
0
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
Stores data for serial transmission



Note: * Can only be written with 0 for flag clearing.



Note: * Can only be written with 0 for flag clearing.

Bit	:	7	6	5	4	3	2	1	0
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	R	R	R	R	R	R	R	R

Stores received serial data

SCMR0—Smart Card Mode Register 0 H'FF7E SCI0, Smart Card Interface 0

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	SDIR	SINV	—	SMIF
Initial value	:	1	1	1	1	0	0	1	0
Read/Write	:	—	—	—	—	R/W	R/W	—	R/W

Smart Card Interface Mode Select

0	Smart card interface function is disabled
1	Smart card interface function is enabled

Smart Card Data Invert

0	TDR contents are transmitted as they are Receive data is stored in RDR as it is
1	TDR contents are inverted before being transmitted Receive data is stored in RDR in inverted form

Smart Card Data 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

Bit	:	7	6	5	4	3	2	1	0
		C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock Select

0	0	ϕ clock
	1	ϕ/4 clock
1	0	ϕ/16 clock
	1	ϕ/64 clock

Multiprocessor Mode

0	Multiprocessor function disabled
1	Multiprocessor format selected

Stop Bit Length

0	1 stop bit
1	2 stop bits

Parity Mode

0	Even parity
1	Odd parity

Parity Enable

0	Parity bit addition and checking disabled
1	Parity bit addition and checking enabled

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

Bit :	7	6	5	4	3	2	1	0
	GM	BLK	PE	O/$\overline{E}$	BCP1	BCP0	CKS1	CKS0
Initial value :	0	0	0	0	0	0	0	0
Read/Write :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock Select

0	0	ϕ clock
	1	ϕ/4 clock
1	0	ϕ/16 clock
	1	ϕ/64 clock

Base Clock Pulse

BCP1	BCP0	Base Clock Pulse
0	0	32 clocks
	1	64 clocks
1	0	372 clocks
	1	256 clocks

Parity Mode
(Set to 1 when using the smart card interface)

0	Even parity
1	Odd parity

Parity Enable

0	Setting prohibited
1	Parity bit addition and checking enabled

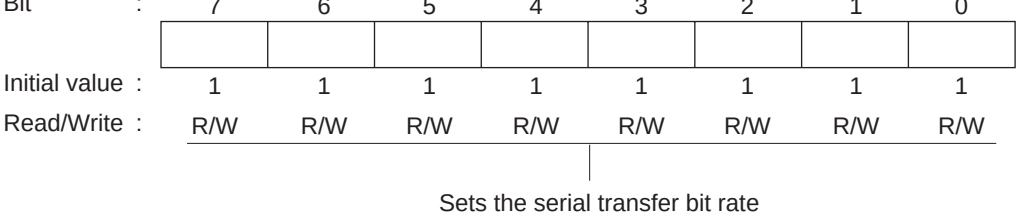
Block Transfer Mode Select

0	Normal smart card interface mode
1	Block transfer mode

GSM Mode

0	Normal smart card interface mode operation • TEND flag generated 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 generated 11.0 etu after beginning of start bit • Fixed high/low-level control possible (set in SCR) in addition to clock output on/off control

Note: etu: Elementary time unit (time for transfer of one bit)



Note: For details, see section 14.2.8, Bit Rate Register (BRR).

Bit :	7	6	5	4	3	2	1	0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
Initial value :	0	0	0	0	0	0	0	0
Read/Write :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock Enable

0	0	Asynchronous mode	Internal clock/SCK pin functions as I/O port
		Synchronous mode	Internal clock/SCK pin functions as serial clock output
	1	Asynchronous mode	Internal clock/SCK pin functions as clock output*1
		Synchronous mode	Internal clock/SCK pin functions as serial clock output
1	0	Asynchronous mode	External clock/SCK pin functions as clock input*2
		Synchronous mode	External clock/SCK pin functions as serial clock input
	1	Asynchronous mode	External clock/SCK pin functions as clock input*2
		Synchronous mode	External clock/SCK pin functions as serial clock input

Notes: 1. Outputs a clock of the same frequency as the bit rate.
2. Inputs a clock with a frequency 16 times the bit rate.

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-data-full interrupt (RXI) requests, receive error interrupt (ERI) requests, and setting of the RDRF, FER, and ORER flags in SSR are disabled until data with the multiprocessor bit set 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) request disabled
1	Receive-data-full interrupt (RXI) request and receive-error interrupt (ERI) request enabled

Transmit Interrupt Enable

0	Transmit-data-empty interrupt (TXI) request disabled
1	Transmit-data-empty interrupt (TXI) request enabled

Bit	7	6	5	4	3	2	1	0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock Enable

SCMR	SMR	SCR setting		SCK pin function
SMIF	GM	CKE1	CKE0	
0	See SCI specification			
1	0	0	0	Operates as port I/O pin
1	0	0	1	Clock output as SCK output pin
1	1	0	0	Fixed-low output as SCK output pin
1	1	0	1	Clock output as SCK output pin
1	1	1	0	Fixed-high output as SCK output pin
1	1	1	1	Clock output as SCK 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-data-full interrupt (RXI) requests, receive error interrupt (ERI) requests, and setting of the RDRF, FER, and ORER flags in SSR are disabled until data with the multiprocessor bit set to 1 is received

Receive Enable

0	Reception disabled
1	Reception enabled

Transmit Enable

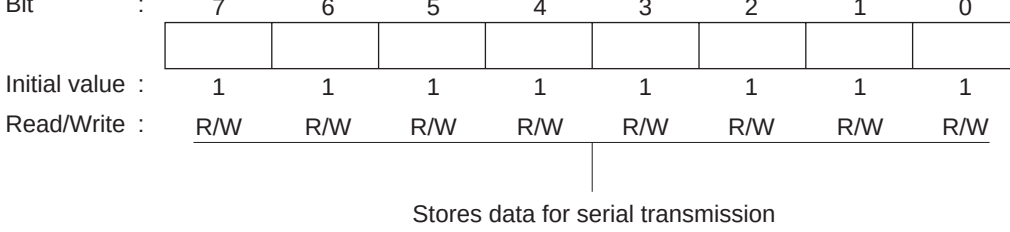
0	Transmission disabled
1	Transmission enabled

Receive Interrupt Enable

0	Receive-data-full interrupt (RXI) request and receive-error interrupt (ERI) request disabled
1	Receive-data-full interrupt (RXI) request and receive-error interrupt (ERI) request enabled

Transmit Interrupt Enable

0	Transmit-data-empty interrupt (TXI) request disabled
1	Transmit-data-empty interrupt (TXI) request enabled


Bit :
7
6
5
4
3
2
1
0
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
Stores data for serial transmission

Bit	7	6	5	4	3	2	1	0
	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
Initial value :	1	0	0	0	0	1	0	0
Read/Write :	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/W

Multiprocessor Bit Transfer

0	Data with a 0 multiprocessor bit is transmitted
1	Data with a 1 multiprocessor bit is transmitted

Multiprocessor Bit

0	[Clearing condition] When data with a 0 multiprocessor bit is received
1	[Setting condition] When data with a 1 multiprocessor bit is received

Transmit End

0	[Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DMAC or DTC is activated by a TXI interrupt and writes data to TDR
1	[Setting conditions] • When the TE bit in SCR is 0 • When TDRE = 1 at transmission of the last bit of a 1-byte serial transmit character

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 plus the parity bit does not match the parity setting (even or 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 the stop bit at the end of the receive data 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

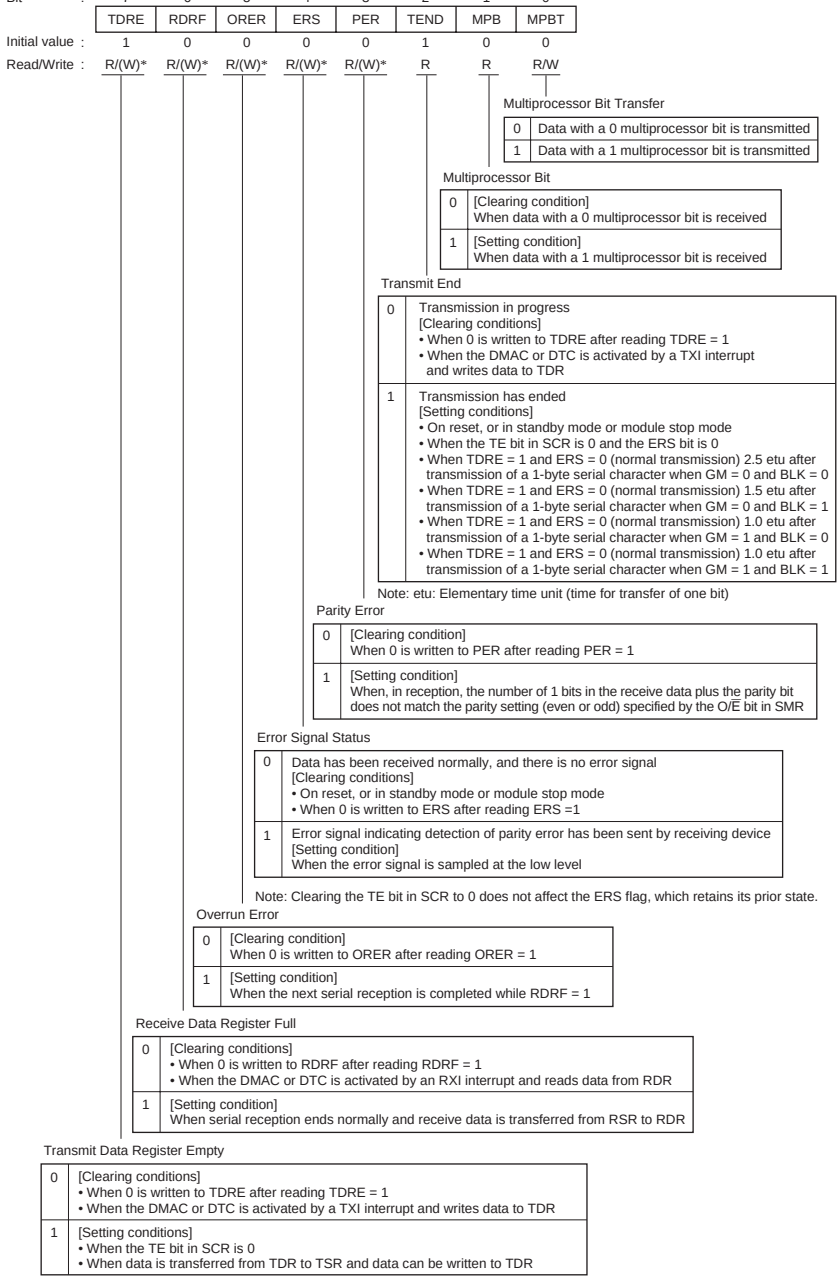
Receive Data Register Full

0	[Clearing conditions] • When 0 is written to RDRF after reading RDRF = 1 • When the DMAC or 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 DMAC or 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

Note: * Can only be written with 0 for flag clearing.



Note: * Can only be written with 0 for flag clearing.

Bit	:	7	6	5	4	3	2	1	0
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	R	R	R	R	R	R	R	R

Stores received serial data

SCMR1—Smart Card Mode Register 1 H'FF86 SCI1, Smart Card Interface 1

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	SDIR	SINV	—	SMIF
Initial value	:	1	1	1	1	0	0	1	0
Read/Write	:	—	—	—	—	R/W	R/W	—	R/W

Smart Card Interface Mode Select

0	Smart card interface function is disabled
1	Smart card interface function is enabled

Smart Card Data Invert

0	TDR contents are transmitted as they are Receive data is stored in RDR as it is
1	TDR contents are inverted before being transmitted Receive data is stored in RDR in inverted form

Smart Card Data 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

Bit :	7	6	5	4	3	2	1	0
	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0
Initial value :	0	0	0	0	0	0	0	0
Read/Write :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock Select

0	0	ϕ clock
	1	ϕ/4 clock
1	0	ϕ/16 clock
	1	ϕ/64 clock

Multiprocessor Mode

0	Multiprocessor function disabled
1	Multiprocessor format selected

Stop Bit Length

0	1 stop bit
1	2 stop bits

Parity Mode

0	Even parity
1	Odd parity

Parity Enable

0	Parity bit addition and checking disabled
1	Parity bit addition and checking enabled

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

Bit	7	6	5	4	3	2	1	0
	GM	BLK	PE	O/$\overline{E}$	BCP1	BCP0	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock Select

0	0	ϕ clock
	1	ϕ/4 clock
1	0	ϕ/16 clock
	1	ϕ/64 clock

Base Clock Pulse

BCP1	BCP0	Base Clock Pulse
0	0	32 clocks
	1	64 clocks
1	0	372 clocks
	1	256 clocks

Parity Mode
(Set to 1 when using the smart card interface)

0	Even parity
1	Odd parity

Parity Enable

0	Setting prohibited
1	Parity bit addition and checking enabled

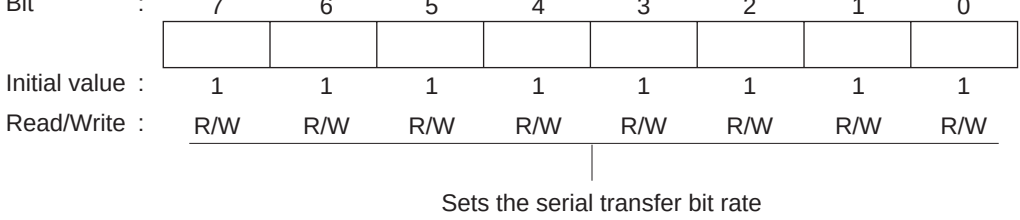
Block Transfer Mode Select

0	Normal smart card interface mode
1	Block transfer mode

GSM Mode

0	Normal smart card interface mode operation • TEND flag generated 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 generated 11.0 etu after beginning of start bit • Fixed high/low-level control possible (set in SCR) in addition to clock output on/off control

Note: etu: Elementary time unit (time for transfer of one bit)



Note: For details, see section 14.2.8, Bit Rate Register (BRR).

Bit :	7	6	5	4	3	2	1	0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
Initial value :	0	0	0	0	0	0	0	0
Read/Write :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock Enable

0	0	Asynchronous mode	Internal clock/SCK pin functions as I/O port
		Synchronous mode	Internal clock/SCK pin functions as serial clock output
	1	Asynchronous mode	Internal clock/SCK pin functions as clock output*1
		Synchronous mode	Internal clock/SCK pin functions as serial clock output
1	0	Asynchronous mode	External clock/SCK pin functions as clock input*2
		Synchronous mode	External clock/SCK pin functions as serial clock input
	1	Asynchronous mode	External clock/SCK pin functions as clock input*2
		Synchronous mode	External clock/SCK pin functions as serial clock input

Notes: 1. Outputs a clock of the same frequency as the bit rate.
2. Inputs a clock with a frequency 16 times the bit rate.

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-data-full interrupt (RXI) requests, receive error interrupt (ERI) requests, and setting of the RDRF, FER, and ORER flags in SSR are disabled until data with the multiprocessor bit set 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) request disabled
1	Receive-data-full interrupt (RXI) request and receive-error interrupt (ERI) request enabled

Transmit Interrupt Enable

0	Transmit-data-empty interrupt (TXI) request disabled
1	Transmit-data-empty interrupt (TXI) request enabled

Bit	7	6	5	4	3	2	1	0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock Enable

SMCR	SMR	SCR setting		SCK pin function
SMIF	GM	CKE1	CKE0	
0	See SCI specification			
1	0	0	0	Operates as port I/O pin
1	0	0	1	Clock output as SCK output pin
1	1	0	0	Fixed-low output as SCK output pin
1	1	0	1	Clock output as SCK output pin
1	1	1	0	Fixed-high output as SCK output pin
1	1	1	1	Clock output as SCK 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-data-full interrupt (RXI) requests, receive error interrupt (ERI) requests, and setting of the RDRF, FER, and ORER flags in SSR are disabled until data with the multiprocessor bit set to 1 is received

Receive Enable

0	Reception disabled
1	Reception enabled

Transmit Enable

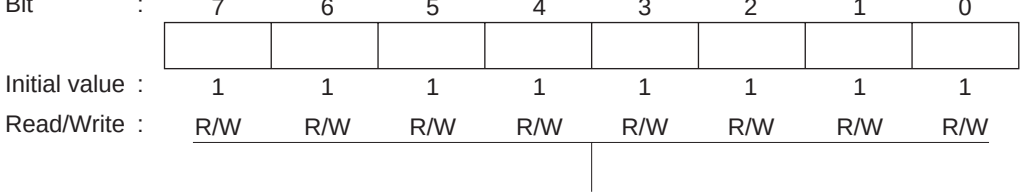
0	Transmission disabled
1	Transmission enabled

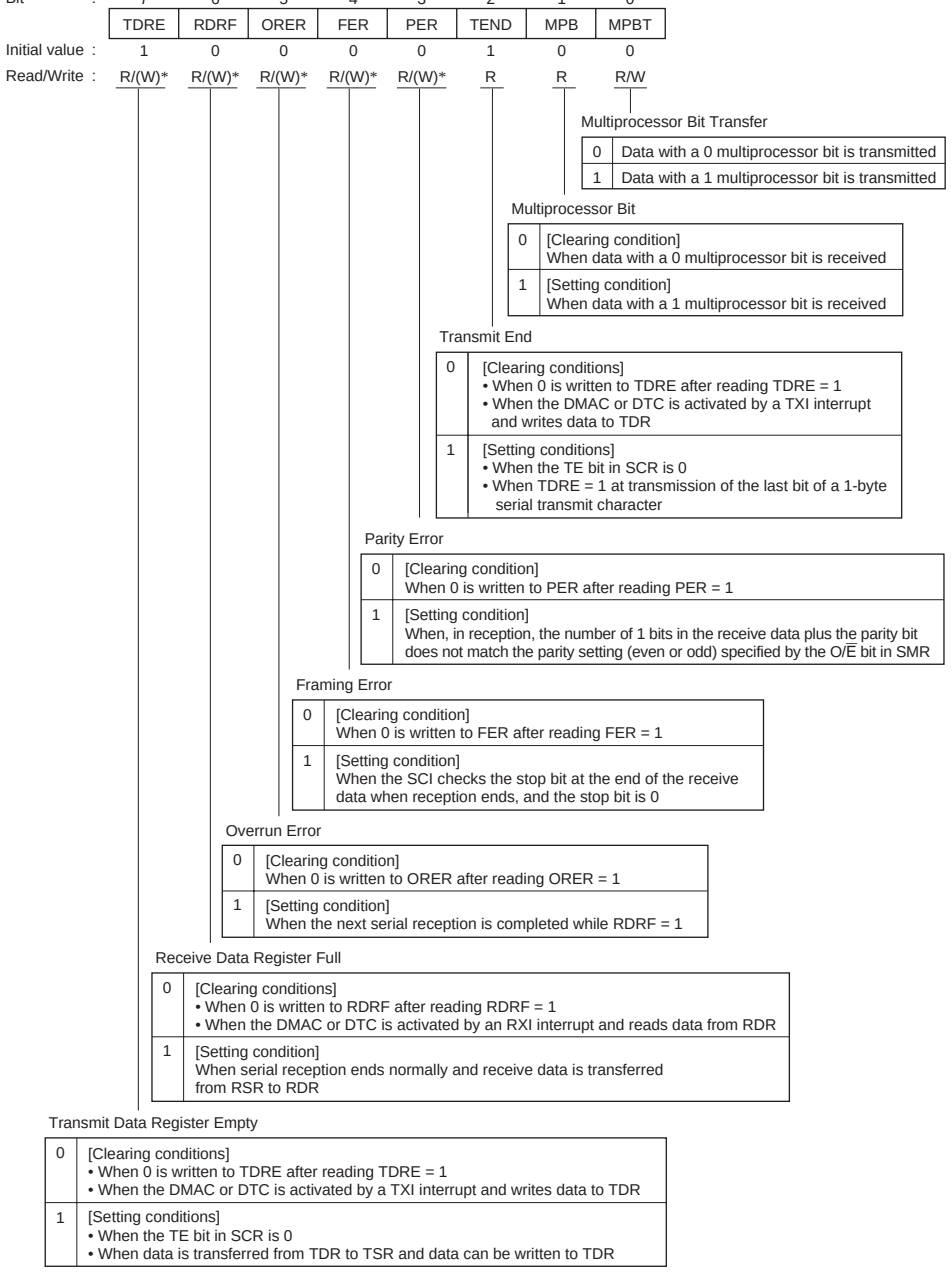
Receive Interrupt Enable

0	Receive-data-full interrupt (RXI) request and receive-error interrupt (ERI) request disabled
1	Receive-data-full interrupt (RXI) request and receive-error interrupt (ERI) request enabled

Transmit Interrupt Enable

0	Transmit-data-empty interrupt (TXI) request disabled
1	Transmit-data-empty interrupt (TXI) request enabled


Bit :
7
6
5
4
3
2
1
0
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
Stores data for serial transmission



Note: * Can only be written with 0 for flag clearing.

Bit	TDRE	RDRF	ORER	ERS	PER	TEND	MPB	MPBT
Initial value :	1	0	0	0	0	1	0	0
Read/Write :	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/W

Multiprocessor Bit Transfer

0	Data with a 0 multiprocessor bit is transmitted
1	Data with a 1 multiprocessor bit is transmitted

Multiprocessor Bit

0	[Clearing condition] When data with a 0 multiprocessor bit is received
1	[Setting condition] When data with a 1 multiprocessor bit is received

Transmit End

0	Transmission in progress [Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DMAC or DTC is activated by a TXI interrupt and writes data to TDR
1	Transmission has ended [Setting conditions] • On reset, or in standby mode or module stop mode • When the TE bit in SCR is 0 and the ERS bit is 0 • When TDRE = 1 and ERS = 0 (normal transmission) 2.5 etu after transmission of a 1-byte serial character when GM = 0 and BLK = 0 • When TDRE = 1 and ERS = 0 (normal transmission) 1.5 etu after transmission of a 1-byte serial character when GM = 0 and BLK = 1 • When TDRE = 1 and ERS = 0 (normal transmission) 1.0 etu after transmission of a 1-byte serial character when GM = 1 and BLK = 0 • When TDRE = 1 and ERS = 0 (normal transmission) 1.0 etu after transmission of a 1-byte serial character when GM = 1 and BLK = 1

Note: etu: Elementary time unit (time for transfer of one 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 plus the parity bit does not match the parity setting (even or odd) specified by the O/$\overline{E}$ bit in SMR

Error Signal Status

0	Data has been received normally, and there is no error signal [Clearing conditions] • On reset, or in standby mode or module stop mode • When 0 is written to ERS after reading ERS =1
1	Error signal indicating detection of parity error has been sent by receiving device [Setting condition] When the error signal is sampled at the low level

Note: Clearing the TE bit in SCR to 0 does not affect the ERS flag, which retains its prior 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 DMAC or 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 DMAC or 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

Note: * Can only be written with 0 for flag clearing.

Bit	:	7	6	5	4	3	2	1	0
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	R	R	R	R	R	R	R	R

Stores received serial data

SCMR2—Smart Card Mode Register 2 **H'FF8E** **SCI2, Smart Card Interface 2**

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	SDIR	SINV	—	SMIF
Initial value	:	1	1	1	1	0	0	1	0
Read/Write	:	—	—	—	—	R/W	R/W	—	R/W

Smart Card Interface Mode Select

0	Smart card interface function is disabled
1	Smart card interface function is enabled

Smart Card Data Invert

0	TDR contents are transmitted as they are Receive data is stored in RDR as it is
1	TDR contents are inverted before being transmitted Receive data is stored in RDR in inverted form

Smart Card Data 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

ADDRBH—A/D Data Register BH	**H'FF92**	**A/D Converter**
ADDRBL—A/D Data Register BL	**H'FF93**	**A/D Converter**
ADDRCH—A/D Data Register CH	**H'FF94**	**A/D Converter**
ADDRCL—A/D Data Register CL	**H'FF95**	**A/D Converter**
ADDRDH—A/D Data Register DH	**H'FF96**	**A/D Converter**
ADDRDL—A/D Data Register DL	**H'FF97**	**A/D Converter**

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	—	—	—	—	—	—
Initial value	:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Read/Write	:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Stores the results of A/D conversion

Analog Input Channel				A/D Data Register
Channel Set 0 (CH3 = 1)		Channel Set 1 (CH3 = 0)		
Group 0	Group 1	Group 0	Group 1	
AN0	AN4	Setting prohibited	AN12	ADDRA
AN1	AN5	Setting prohibited	AN13	ADDRB
AN2	AN6	Setting prohibited	AN14	ADDRC
AN3	AN7	Setting prohibited	AN15	ADDRD

Bit	:	7	6	5	4	3	2	1	0
		ADF	ADIE	ADST	SCAN	CKS	CH2	CH1	CH0
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	R/(W)*	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Channel Select
Note: CH2 to CH0 are used in combination with CH3 bit in ADCR and SCAN bit to select the analog input channels. See ADCR—A/D Control Register H'FF99 A/D Converter.

Clock Select
Note: CKS is used in combination with bit 3 (CKS1) in ADCR. See ADCR—A/D Control Register H'FF99 A/D Converter.

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 automatically when conversion ends • Scan mode: A/D conversion is started. Conversion continues sequentially on the selected channels until ADST is cleared to 0 by software, a reset, or transition to standby mode or module stop mode

A/D Interrupt Enable

0	A/D conversion end interrupt request disabled
1	A/D conversion end interrupt request enabled

A/D End Flag

0	[Clearing conditions] • When 0 is written to the ADF flag after reading ADF = 1 • When the DMAC or 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.

Bit	:	7	6	5	4	3	2	1	0
		TRGS1	TRGS0	—	—	CKS1	CH3	—	—
Initial value	:	0	0	1	1	1	1	1	1
Read/Write	:	R/W	R/W	—	—	R/W	R/W	—	—

Channel Select

Selects the analog input channels. Ensure that conversion is halted (ADST = 0) before making a channel selection.

Channel Selection				Description	
CH3	CH2*	CH1*	CH0*	Single Mode (SCAN = 0)	Scan Mode (SCAN = 1)
0	0	0	0	Setting prohibited	Setting prohibited
			1		
		1	0		
			1		
	1	0	0	AN12	AN12
			1	AN13	AN12, AN13
		1	0	AN14	AN12 to AN14
			1	AN15	AN12 to AN15
1	0	0	0	AN0 (Initial value)	AN0
			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

Note: * CH2, CH1, and CH0 are bits in ADCSR.

Clock Select

Bit 3	ADCSR Bit 3	Description
CKS1	CKS	
0	0	Conversion time = 530 states (max.)
	1	Conversion time = 68 states (max.)
1	0	Conversion time = 266 states (max.) (Initial value)
	1	Conversion time = 134 states (max.)

Timer Trigger Select

TRGS1	TRGS1	Description
0	0	A/D conversion start by external trigger is disabled
	1	A/D conversion start by external trigger (TPU) is enabled
1	0	A/D conversion start by external trigger (8-bit timer) is enabled
	1	A/D conversion start by external trigger pin ($\overline{\text{ADTRG}}$) is enabled


Bit : 7 6 5 4 3 2 1 0
Initial value : 0 0 0 0 0 0 0 0
Read/Write : R/W R/W R/W R/W R/W R/W R/W R/W
Stores data for D/A conversion

Bit :	7	6	5	4	3	2	1	0
	DAOE1	DAOE0	DAE	—	—	—	—	—
Initial value :	0	0	0	1	1	1	1	1
Read/Write :	R/W	R/W	R/W	—	—	—	—	—

D/A Output Enable 0

0	Analog output DA_0 is disabled
1	Channel 0 D/A conversion is enabled Analog output DA_0 is enabled

D/A Output Enable 1

0	Analog output DA_1 is disabled
1	Channel 1 D/A conversion is enabled Analog output DA_1 is enabled

D/A Conversion Control

DAOE1	DAOE0	DAE	Description
0	0	*	Channel 0 and 1 D/A conversion disabled
	1	0	Channel 0 D/A conversion enabled
			Channel 1 D/A conversion disabled
		1	Channel 0 and 1 D/A conversion enabled
1	0	0	Channel 0 D/A conversion disabled
			Channel 1 D/A conversion enabled
		1	Channel 0 and 1 D/A conversion enabled
	1	*	Channel 0 and 1 D/A conversion enabled

* : Don't care


Bit : 7 6 5 4 3 2 1 0
Initial value : 0 0 0 0 0 0 0 0
Read/Write : R/W R/W R/W R/W R/W R/W R/W R/W
Stores data for D/A conversion

Bit	:	7	6	5	4	3	2	1	0
		DAOE1	DAOE0	DAE	—	—	—	—	—
Initial value	:	0	0	0	1	1	1	1	1
Read/Write	:	R/W	R/W	R/W	—	—	—	—	—

D/A Output Enable 0

0	Analog output DA_2 is disabled
1	Channel 2 D/A conversion is enabled Analog output DA_2 is enabled

D/A Output Enable 1

0	Analog output DA_3 is disabled
1	Channel 3 D/A conversion is enabled Analog output DA_3 is enabled

D/A Conversion Control

DAOE1	DAOE0	DAE	Description
0	0	*	Channel 2 and 3 D/A conversion disabled
	1	0	Channel 2 D/A conversion enabled
			Channel 3 D/A conversion disabled
		1	Channel 2 and 3 D/A conversion enabled
1	0	0	Channel 2 D/A conversion disabled
			Channel 3 D/A conversion enabled
		1	Channel 2 and 3 D/A conversion enabled
	1	*	Channel 2 and 3 D/A conversion enabled

* : Don't care

Bit	:	7	6	5	4	3	2	1	0
		WAITPS	BREQOPS	CS167E	CS25E	ASOD	—	—	—
Initial value	:	0	0	1	1	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R	R	R

AS Output Disable

0	PF_6 is designated as $\overline{AS}$ output pin
1	PF_6 is designated as I/O port, and does not function as $\overline{AS}$ output pin

Note: This bit is valid in modes 4 to 6.

CS25 Enable

0	$\overline{CS}_2$, $\overline{CS}_3$, $\overline{CS}_4$, and $\overline{CS}_5$ output disabled (can be used as I/O ports)
1	$\overline{CS}_2$, $\overline{CS}_3$, $\overline{CS}_4$, and $\overline{CS}_5$ output enabled

Note: Clear the DDR bits to 0 before changing the CS25E setting.

CS167 Enable

0	$\overline{CS}_1$, $\overline{CS}_6$, and $\overline{CS}_7$ output disabled (can be used as I/O ports)
1	$\overline{CS}_1$, $\overline{CS}_6$, and $\overline{CS}_7$ output enabled

Note: Clear the DDR bits to 0 before changing the CS167E setting.

BREQO Pin Select

0	$\overline{BREQO}$ output is PF_2 pin
1	$\overline{BREQO}$ output is $P5_3$ pin

Note: Set BREQOPS before setting the BREQOE bit in BCRL to 1.

WAIT Pin Select

0	$\overline{WAIT}$ input is $P8_6$ pin
1	$\overline{WAIT}$ input is $P5_3$ pin

Note: Set WAITPS before setting the WAITE bit in BCRL to 1.

Bit	7	6	5	4	3	2	1	0
	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0
Initial value :	0	0	0	0	0	0	0	0
Read/Write :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock Select

0	0	0	Clock input disabled
		1	Internal clock: counted at falling edge of ϕ/8
	1	0	Internal clock: counted at falling edge of ϕ/64
		1	Internal clock: counted at falling edge of ϕ/8192
1	0	0	For channel 0: Count at TCNT1 overflow signal* For channel 1: Count at TCNT0 compare match A*
		1	External clock: counted at rising edge
	1	0	External clock: counted at falling edge
		1	External clock: counted at both rising and falling edges

Note: * If the count input of channel 0 is the TCNT1 overflow signal and that of channel 1 is the TCNT0 compare match signal, no incrementing clock is generated. Do not use this setting.

Counter Clear

0	0	Clear is disabled
	1	Clear by compare match A
1	0	Clear by compare match B
	1	Clear by rising edge of external reset input

Timer Overflow Interrupt Enable

0	OVF interrupt requests (OVI) are disabled
1	OVF interrupt requests (OVI) are enabled

Compare Match Interrupt Enable A

0	CMFA interrupt requests (CMIA) are disabled
1	CMFA interrupt requests (CMIA) are enabled

Compare Match Interrupt Enable B

0	CMFB interrupt requests (CMIB) are disabled
1	CMFB interrupt requests (CMIB) are enabled

TCSR0 Bit	:	7	6	5	4	3	2	1	0
		CMFB	CMFA	OVF	ADTE	OS3	OS2	OS1	OS0
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	R/(W)*	R/(W)*	R/(W)*	R/W	R/W	R/W	R/W	R/W

TCSR1 Bit	:	7	6	5	4	3	2	1	0
		CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0
Initial value	:	0	0	0	1	0	0	0	0
Read/Write	:	R/(W)*	R/(W)*	R/(W)*	—	R/W	R/W	R/W	R/W

Output Select

0	0	No change when compare match A occurs
	1	0 is output when compare match A occurs
1	0	1 is output when compare match A occurs
	1	Output is inverted when compare match A occurs (toggle output)

Output Select

0	0	No change when compare match B occurs
	1	0 is output when compare match B occurs
1	0	1 is output when compare match B occurs
	1	Output is inverted when compare match B occurs (toggle output)

A/D Trigger Enable (TCSR0 only)

0	A/D converter start requests by compare match A are disabled
1	A/D converter start requests by compare match A are enabled

Timer Overflow Flag

0	[Clearing condition] When 0 is written to OVF after reading OVF = 1
1	[Setting condition] When TCNT overflows (changes from H'FF to H'00)

Compare Match Flag A

0	[Clearing conditions] • When 0 is written to CMFA after reading CMFA = 1 • When the DTC is activated by a CMIA interrupt, while the DISEL bit of MRB in DTC is 0
1	[Setting condition] When TCNT matches TCORA

Compare Match Flag B

0	[Clearing conditions] • When 0 is written to CMFB after reading CMFB = 1 • When the DTC is activated by a CMIB interrupt, while the DISEL bit of MRB in DTC is 0
1	[Setting condition] When TCNT matches TCORB

Note: * Only 0 can be written to bits 7 to 5, to clear these flags.

	TCORA0								TCORA1							
Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value :	1	1	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	R/W	R/W

TCORB0—Time Constant Register B0 **H'FFB6** **8-Bit Timer Channel 0**
TCORB1—Time Constant Register B1 **H'FFB7** **8-Bit Timer Channel 1**

	TCORB0								TCORB1							
Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value :	1	1	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	R/W	R/W

TCNT0—Timer Counter 0 **H'FFB8** **8-Bit Timer Channel 0**
TCNT1—Timer Counter 1 **H'FFB9** **8-Bit Timer Channel 1**

	TCNT0								TCNT1							
Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
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	R/W	R/W

Bit	:	7	6	5	4	3	2	1	0
		OVF	WT/$\overline{IT}$	TME	—	—	CKS2	CKS1	CKS0
Initial value	:	0	0	0	1	1	0	0	0
Read/Write*1	:	R/(W)*2	R/W	R/W	—	—	R/W	R/W	R/W

Clock Select

CKS2	CKS1	CKS0	Clock	Overflow period* (when ϕ = 20 MHz)
0	0	0	ϕ/2 (Initial value)	25.6µs
		1	ϕ/64	819.2µs
	1	0	ϕ/128	1.6ms
		1	ϕ/512	6.6ms
1	0	0	ϕ/2048	26.2ms
		1	ϕ/8192	104.9ms
	1	0	ϕ/32768	419.4ms
		1	ϕ/131072	1.68s

Note: * The overflow period is the time from when TCNT starts counting up from H'00 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 request (WOVI) when TCNT overflows
1	Watchdog timer mode: Generates the $\overline{WDTOVF}$*1 signal when TCNT overflows*2

Notes: 1. The $\overline{WDTOVF}$ pin function is not available in the F-ZTAT versions.
2. For details of the case where TCNT overflows in watchdog timer mode, see section 13.2.3, Reset Control/Status Register (RSTCSR).

Overflow Flag

0	[Clearing condition] When 0 is written to OVF after reading OVF = 1
1	[Setting condition] When TCNT overflows from H'FF to H'00 in interval timer mode

Notes: 1. The method for writing to TCSR is different from that for general registers to prevent accidental overwriting. For details, see section 13.2.4, Notes on Register Access.
2. Can only be written with 0 for flag clearing.

Bit	:	7	6	5	4	3	2	1	0
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

RSTCSR—Reset Control/Status Register **H'FFBE (W), H'FFBF (R)** **WDT**

Bit	:	7	6	5	4	3	2	1	0
		WOVF	RSTE	—	—	—	—	—	—
Initial value	:	0	0	0	1	1	1	1	1
Read/Write	:	R/(W)*	R/W	R/W	—	—	—	—	—

Reserved
This bit should be written with 0.

Reset Enable

0	Reset signal is not generated if TCNT overflows*
1	Reset signal is generated if TCNT overflows

Note: * The modules in the H8S/2339 Group are not reset, but TCNT and TCSR in WDT are reset.

Watchdog Timer Overflow Flag

0	[Clearing condition] When 0 is written to WOVF after reading RSTCSR when WOVF = 1
1	[Setting condition] When TCNT overflows (changes from H'FF to H'00) during watchdog timer operation

Notes: The method for writing to RSTCSR is different from that for general registers to prevent accidental overwriting. For details, see section 13.2.4, Notes on Register Access.

* Can only be written with 0 for flag clearing.

Bit	:	7	6	5	4	3	2	1	0
		—	—	CST5	CST4	CST3	CST2	CST1	CST0
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	—	—	R/W	R/W	R/W	R/W	R/W	R/W

Counter Start

0	TCNTn count operation is stopped
1	TCNTn performs count operation

(n = 5 to 0)

Note: If 0 is written to the CST bit during operation with the TIOC pin designated for output, the counter stops but the TIOC pin output compare output level is retained. If TIOR is written to when the CST bit is cleared to 0, the pin output level will be changed to the set initial output value.

TSYR—Timer Synchro Register H'FFC1 TPU

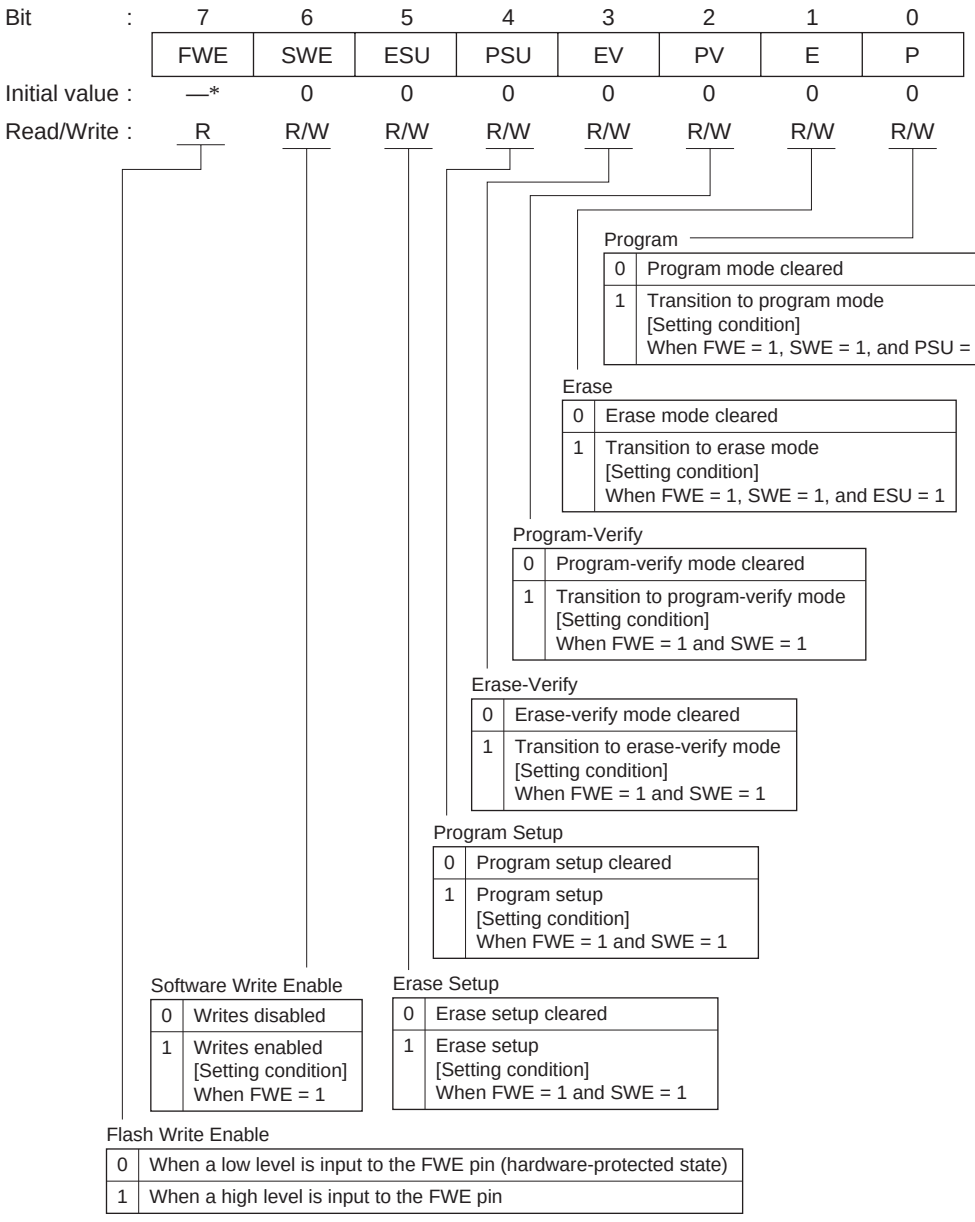
Bit	:	7	6	5	4	3	2	1	0
		—	—	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	—	—	R/W	R/W	R/W	R/W	R/W	R/W

Timer Synchronization

0	TCNTn operates independently (TCNT presetting/ clearing is unrelated to other channels)
1	TCNTn performs synchronous operation TCNT synchronous presetting/synchronous clearing is possible

(n = 5 to 0)

Notes: 1. To set synchronous operation, the SYNC bits for at least two channels must be set to 1.
2. To set synchronous clearing, in addition to the SYNC bit, the TCNT clearing source must also be set by means of bits CCLR2 to CCLR0 in TCR.



Note: * Determined by the state of the FWE pin (H8S/2338 F-ZTAT).
The FWE pin is fixed to 1 in the H8S/2339 F-ZTAT.

Bit	7	6	5	4	3	2	1	0
	FLER	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	—	—	—	—	—	—	—

Flash Memory Error

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 for H8S/2339 F-ZTAT, and section 19.17.3, Error Protection for H8S/2338 F-ZTAT.

EBR1—Erase Block Register 1 **H'FFCA** **Flash Memory**
EBR2—Erase Block Register 2 **H'FFCB** **Flash Memory**
(Valid only in F-ZTAT version)

Bit	7	6	5	4	3	2	1	0
EBR1	EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1	0
EBR2	—	—	EB13*	EB12*	EB11	EB10	EB9	EB8
Initial value	0	0	0	0	0	0	0	0
Read/Write	—	—	— (R/W)*	— (R/W)*	R/W	R/W	R/W	R/W

Note: * Valid only in H8S/2339 F-ZTAT.

Bit	7	6	5	4	3	2	1	0
	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
Initial value	0	0	0	0	0	0	0	0
Read/Write	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 pin input
		1	External clock: counts on TCLKB pin input
	1	0	External clock: counts on TCLKC pin input
		1	External clock: counts on TCLKD pin input

Clock Edge

0	0	Count at rising edge
	1	Count at falling edge
1	—	Count at both edges

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 TSYR to 1.
2. When TGRC or TGRD is used as a buffer register, TCNT is not cleared because the buffer register setting has priority, and compare match/input capture does not occur.

Bit :	7	6	5	4	3	2	1	0
	—	—	BFB	BFA	MD3	MD2	MD1	MD0
Initial value :	1	1	0	0	0	0	0	0
Read/Write :	—	—	R/W	R/W	R/W	R/W	R/W	R/W

Mode

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	*	*	*	—

* : Don't care

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 should always be written to MD2.

TGRA Buffer Operation

0	TGRA operates normally
1	TGRA and TGRC used together for buffer operation

TGRB Buffer Operation

0	TGRB operates normally
1	TGRB and TGRD used together for buffer operation

Bit	7	6	5	4	3	2	1	0
	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TGR0A I/O Control

0	0	0	0	TGR0A 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	TGR0A is input capture register	Capture input source is $TIOCA_0$ 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

* : Don't care

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 0 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 $TIOCB_0$ 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 does not occur.

Bit	7	6	5	4	3	2	1	0
	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TGR0C I/O Control

0	0	0	0	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 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	TGR0C is input capture register	Capture input source is $TIOCC_0$ 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

* : Don't care

Note: When the BFA bit in TMDR0 is set to 1 and TGR0C is used as a buffer register, this setting is invalid and input capture/output compare does not occur.

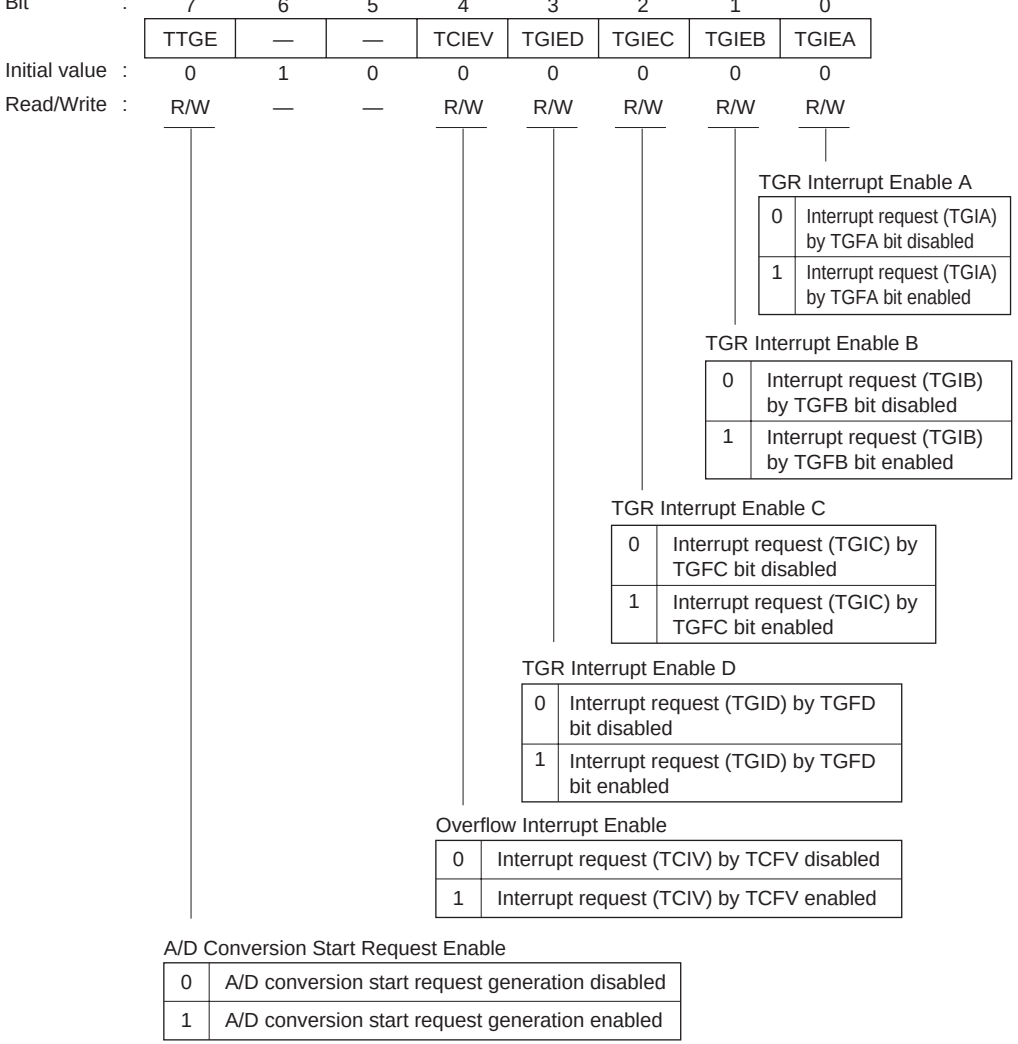
TGR0D I/O Control

0	0	0	0	TGR0D 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	TGR0D is input capture register *2	Capture input source is $TIOCD_0$ 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

Notes: 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 does not occur.
2. When the BFB bit in TMDR0 is set to 1 and TGR0D is used as a buffer register, this setting is invalid and input capture/output compare does not occur.

Note: When TGRC or TGRD is designated for buffer operation, this setting is invalid and the register operates as a buffer register.


Bit : 7 6 5 4 3 2 1 0
TTGE — — TCIEV TGIED TGIEC TGIEB TGIEA
Initial value : 0 1 0 0 0 0 0 0
Read/Write : R/W — — R/W R/W R/W R/W R/W
TGR Interrupt Enable A
0 Interrupt request (TGIA) by TGFA bit disabled
1 Interrupt request (TGIA) by TGFA bit enabled
TGR Interrupt Enable B
0 Interrupt request (TGIB) by TGFB bit disabled
1 Interrupt request (TGIB) by TGFB bit enabled
TGR Interrupt Enable C
0 Interrupt request (TGIC) by TGFC bit disabled
1 Interrupt request (TGIC) by TGFC bit enabled
TGR Interrupt Enable D
0 Interrupt request (TGID) by TGFD bit disabled
1 Interrupt request (TGID) by TGFD bit enabled
Overflow Interrupt Enable
0 Interrupt request (TCIV) by TCFV disabled
1 Interrupt request (TCIV) by TCFV enabled
A/D Conversion Start Request Enable
0 A/D conversion start request generation disabled
1 A/D conversion start request generation enabled

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
Initial value	:	1	1	0	0	0	0	0	0
Read/Write	:	—	—	—	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

TGR Input Capture/Output Compare Flag A

0	[Clearing conditions] • When DTC is activated by TGIA interrupt while DISEL bit of MRB in DTC is 0 • When DMAC is activated by TGIA interrupt while DTA bit of DMABCR in DMAC is 1 • When 0 is written to TGFA after reading TGFA = 1
1	[Setting conditions] • When TCNT = TGRA while TGRA is functioning as output compare register • When TCNT value is transferred to TGRA by input capture signal while TGRA is functioning as input capture register

TGR Input Capture/Output Compare Flag B

0	[Clearing conditions] • When DTC is activated by TGIB interrupt while DISEL bit of MRB in DTC 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 while DISEL bit of MRB in DTC is 0 • When 0 is written to TGFC after reading TGFC = 1
1	[Setting conditions] • When TCNT = TGRC while TGRC is functioning as output compare register • When TCNT value is transferred to TGRC by input capture signal while TGRC is functioning as input capture register

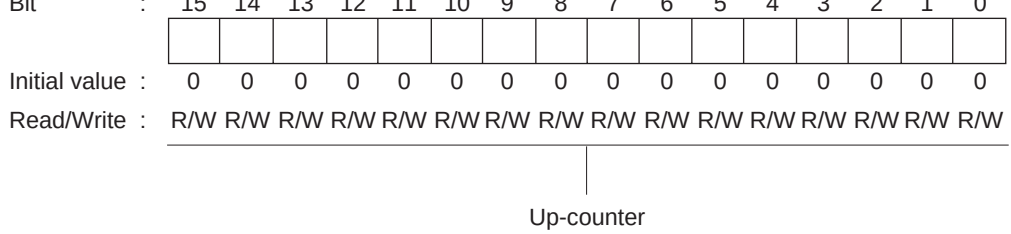
TGR Input Capture/Output Compare Flag D

0	[Clearing conditions] • When DTC is activated by TGID interrupt while DISEL bit of MRB in DTC is 0 • When 0 is written to TGFD after reading TGFD = 1
1	[Setting conditions] • When TCNT = TGRD while TGRD is functioning as output compare register • When TCNT value is transferred to TGRD by input capture signal while TGRD is functioning as input capture register

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'FFD8** **TPU0**
TGR0B—Timer General Register 0B **H'FFDA** **TPU0**
TGR0C—Timer General Register 0C **H'FFDC** **TPU0**
TGR0D—Timer General Register 0D **H'FFDE** **TPU0**

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	:	1	1	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	R/W	R/W

Bit	:	7	6	5	4	3	2	1	0
		—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	—	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 pin input
		1	External clock: counts on TCLKB pin input
	1	0	Internal clock: counts on ϕ/256
		1	Counts on TCNT2 overflow/underflow

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

Clock Edge*

0	0	Count at rising edge
	1	Count at falling edge
1	—	Count at both edges

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

Counter Clear

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*

Note: * Synchronous operation setting is performed by setting the SYNC bit in TSYR to 1.

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	MD3	MD2	MD1	MD0
Initial value	:	1	1	0	0	0	0	0	0
Read/Write	:	—	—	—	—	R/W	R/W	R/W	R/W

Mode

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	*	*	*	—

* : Don't care

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

Bit	:	7	6	5	4	3	2	1	0
		IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TGR1A I/O Control

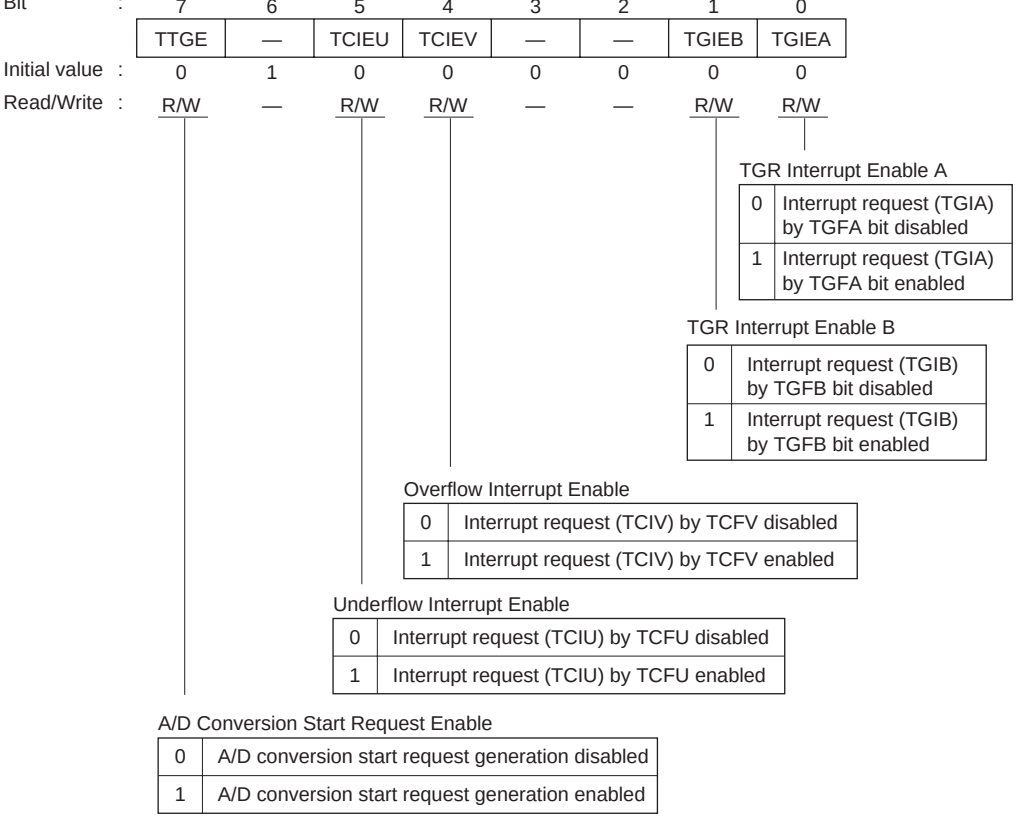
0	0	0	0	TGR1A is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	0	0	0	TGR1A is input capture register	Capture input source is $TIOCA_1$ pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Capture input source is TGR0A compare match/ input capture	Input capture at generation of channel 0/TGR0A compare match/ input capture

* : Don't care

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 $TIOCB_1$ 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


Bit : 7 6 5 4 3 2 1 0
TTGE — TCIEU TCIEV — — TGIEB TGIEA
Initial value : 0 1 0 0 0 0 0 0
Read/Write : R/W — R/W R/W — — R/W R/W
TGR Interrupt Enable A
0 Interrupt request (TGIA) by TGFA bit disabled
1 Interrupt request (TGIA) by TGFA bit enabled
TGR Interrupt Enable B
0 Interrupt request (TGIB) by TGFB bit disabled
1 Interrupt request (TGIB) by TGFB bit enabled
Overflow Interrupt Enable
0 Interrupt request (TCIV) by TCFV disabled
1 Interrupt request (TCIV) by TCFV enabled
Underflow Interrupt Enable
0 Interrupt request (TCIU) by TCFU disabled
1 Interrupt request (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

Bit	:	7	6	5	4	3	2	1	0
		TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA
Initial value	:	1	1	0	0	0	0	0	0
Read/Write	:	R	—	R/(W)*	R/(W)*	—	—	R/(W)*	R/(W)*

TGR Input Capture/Output Compare Flag A

0	[Clearing conditions] • When DTC is activated by TGIA interrupt while DISEL bit of MRB in DTC is 0 • When DMAC is activated by TGIA interrupt while DTA bit of DMABCR in DMAC is 1 • When 0 is written to TGFA after reading TGFA = 1
1	[Setting conditions] • When TCNT = TGRA while TGRA is functioning as output compare register • When TCNT value is transferred to TGRA by input capture signal while TGRA is functioning as input capture register

TGR Input Capture/Output Compare Flag B

0	[Clearing conditions] • When DTC is activated by TGIB interrupt while DISEL bit of MRB in DTC 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

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)

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'FFFF)

Count Direction Flag

0	TCNT counts down
1	TCNT counts up

Note: * Can only be written with 0 for flag clearing.

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
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	R/W	R/W

Up/down-counter*

Note: * This timer counter can be used as an up/down-counter only in phase counting mode or when performing overflow/underflow counting on another channel. In other cases it functions as an up-counter.

TGR1A—Timer General Register 1A **H'FFE8** **TPU1**
TGR1B—Timer General Register 1B **H'FFEA** **TPU1**

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	:	1	1	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	R/W	R/W

Bit	:	7	6	5	4	3	2	1	0
		—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	—	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 pin input
		1	External clock: counts on TCLKB pin input
	1	0	External clock: counts on TCLKC pin input
		1	Internal clock: counts on ϕ/1024

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

Clock Edge*

0	0	Count at rising edge
	1	Count at falling edge
1	—	Count at both edges

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

Counter Clear

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*

Note: * Synchronous operation setting is performed by setting the SYNC bit in TSYR to 1.

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	MD3	MD2	MD1	MD0
Initial value	:	1	1	0	0	0	0	0	0
Read/Write	:	—	—	—	—	R/W	R/W	R/W	R/W

Mode

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	*	*	*	—

* : Don't care

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

Bit	:	7	6	5	4	3	2	1	0
		IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TGR2A I/O Control

0	0	0	0	TGR2A 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	TGR2A is input capture register	Capture input source is $TIOCA_2$ pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges

* : Don't care

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 $TIOCB_2$ pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges

* : Don't care

Bit	:	7	6	5	4	3	2	1	0
		TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
Initial value	:	0	1	0	0	0	0	0	0
Read/Write	:	R/W	—	R/W	R/W	—	—	R/W	R/W

TGR Interrupt Enable A

0	Interrupt request (TGIA) by TGFA bit disabled
1	Interrupt request (TGIA) by TGFA bit enabled

TGR Interrupt Enable B

0	Interrupt request (TGIB) by TGFB bit disabled
1	Interrupt request (TGIB) by TGFB bit enabled

Overflow Interrupt Enable

0	Interrupt request (TCIV) by TCFV disabled
1	Interrupt request (TCIV) by TCFV enabled

Underflow Interrupt Enable

0	Interrupt request (TCIU) by TCFU disabled
1	Interrupt request (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

Bit	:	7	6	5	4	3	2	1	0
		TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA
Initial value	:	1	1	0	0	0	0	0	0
Read/Write	:	R	—	R/(W)*	R/(W)*	—	—	R/(W)*	R/(W)*

TGR Input Capture/Output Compare Flag A

0	[Clearing conditions] • When DTC is activated by TGIA interrupt while DISEL bit of MRB in DTC is 0 • When DMAC is activated by TGIA interrupt while DTA bit of DMABCR in DMAC is 1 • When 0 is written to TGFA after reading TGFA = 1
1	[Setting conditions] • When TCNT = TGRA while TGRA is functioning as output compare register • When TCNT value is transferred to TGRA by input capture signal while TGRA is functioning as input capture register

TGR Input Capture/Output Compare Flag B

0	[Clearing conditions] • When DTC is activated by TGIB interrupt while DISEL bit of MRB in DTC 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

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)

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'FFFF)

Count Direction Flag

0	TCNT counts down
1	TCNT counts up

Note: * Can only be written with 0 for flag clearing.

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
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	R/W	R/W

Up/down-counter*

Note: * This timer counter can be used as an up/down-counter only in phase counting mode or when performing overflow/underflow counting on another channel. In other cases it functions as an up-counter.

TGR2A—Timer General Register 2A **H'FFF8** **TPU2**
TGR2B—Timer General Register 2B **H'FFFA** **TPU2**

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	:	1	1	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	R/W	R/W

C.1 Port 1

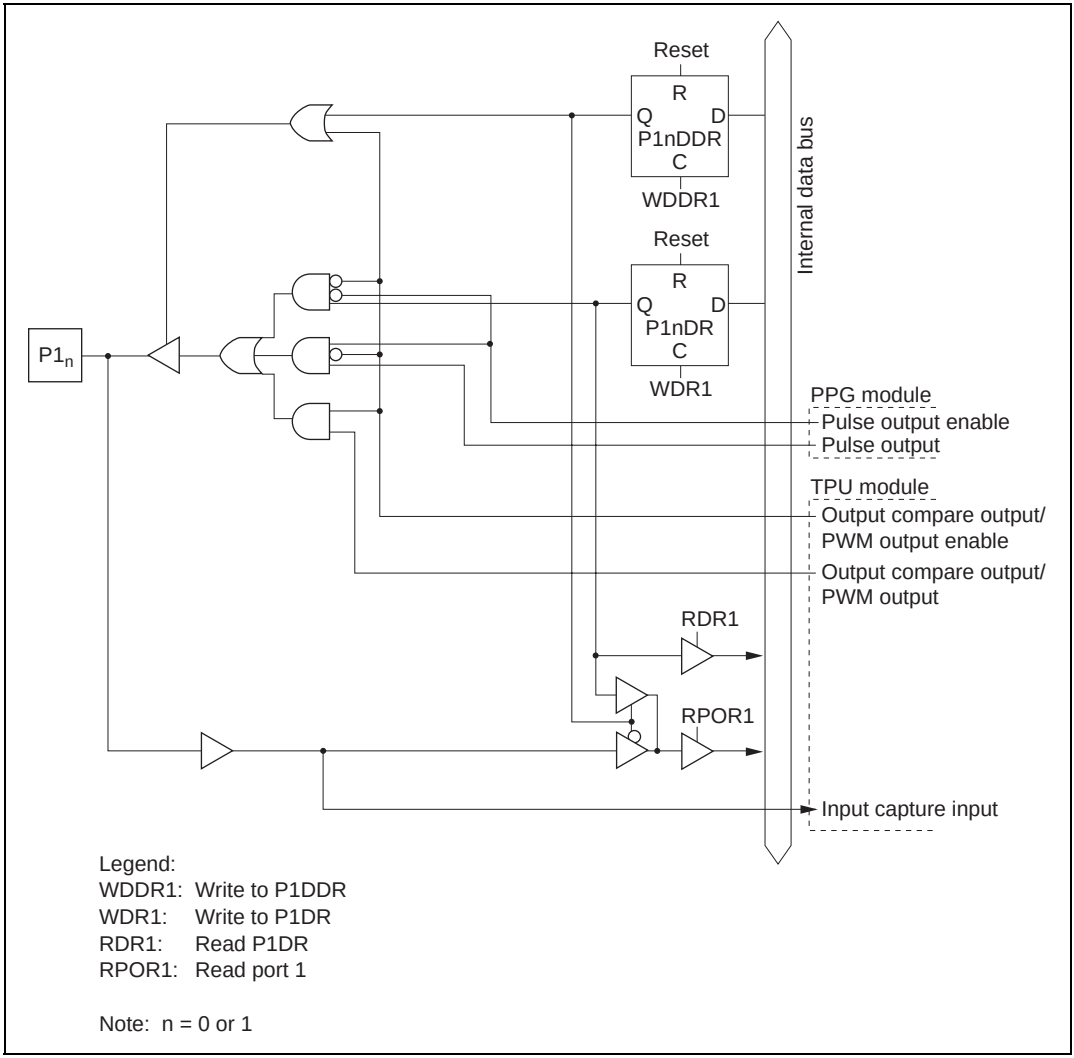


Figure C.1 (a) Port 1 Block Diagram (Pins $P1_0$ and $P1_1$)

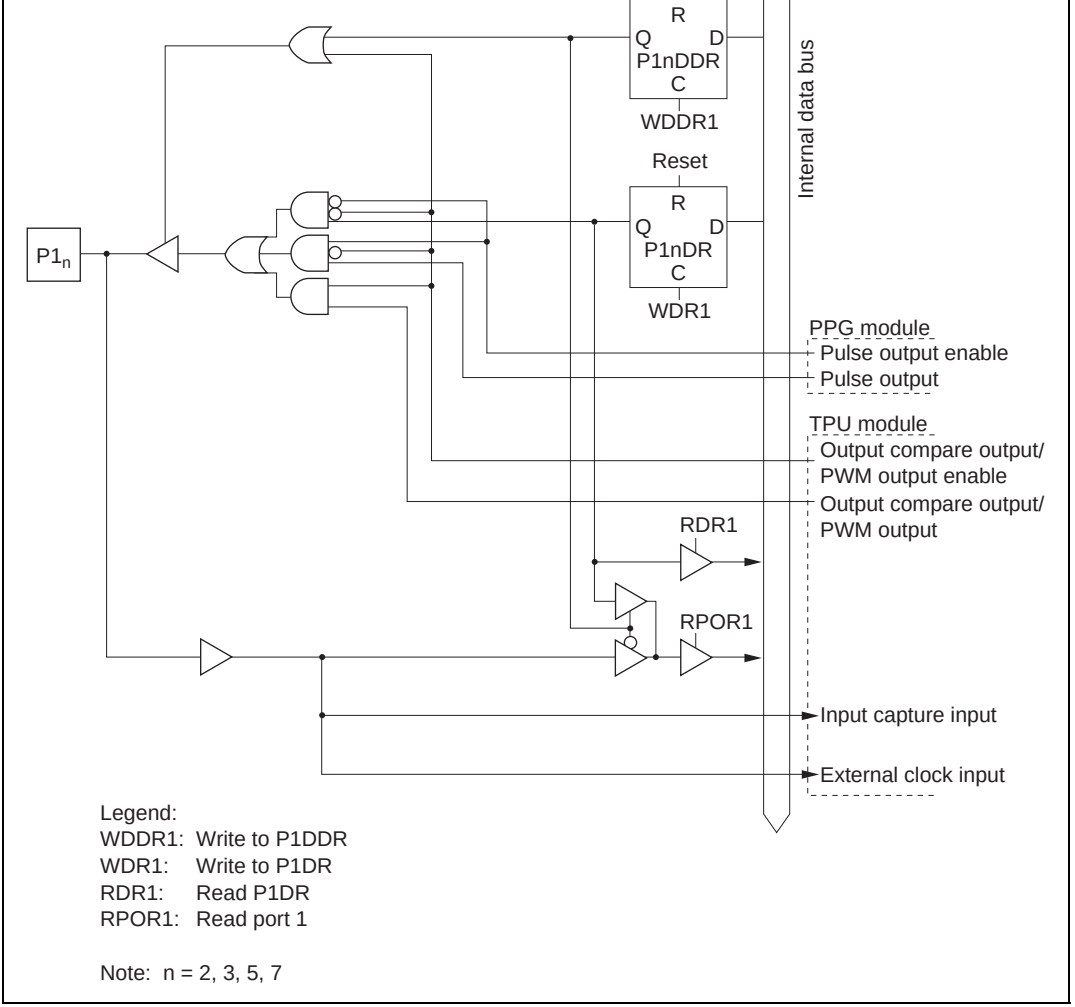


Figure C.1 (b) Port 1 Block Diagram (Pins $P1_2$, $P1_3$, $P1_5$, and $P1_7$)

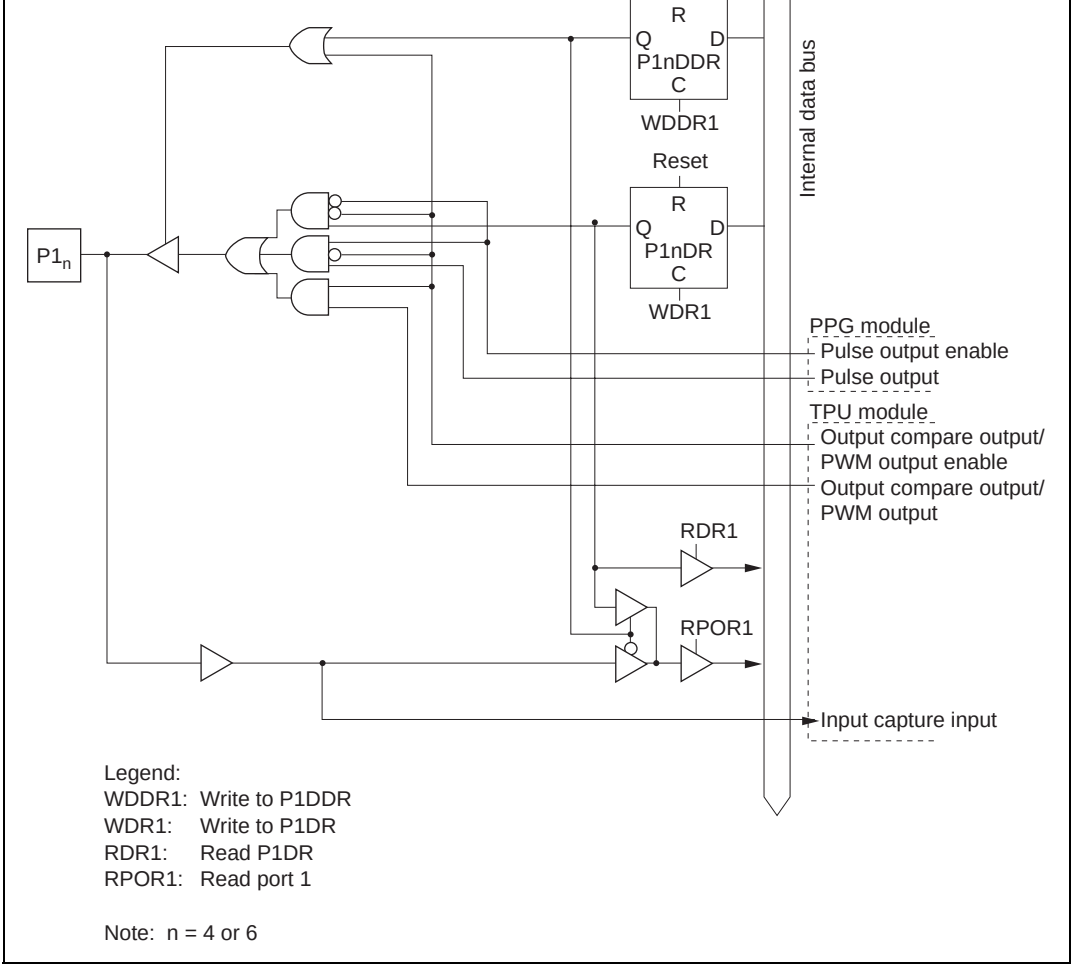


Figure C.1 (c) Port 1 Block Diagram (Pins $P1_4$ and $P1_6$)

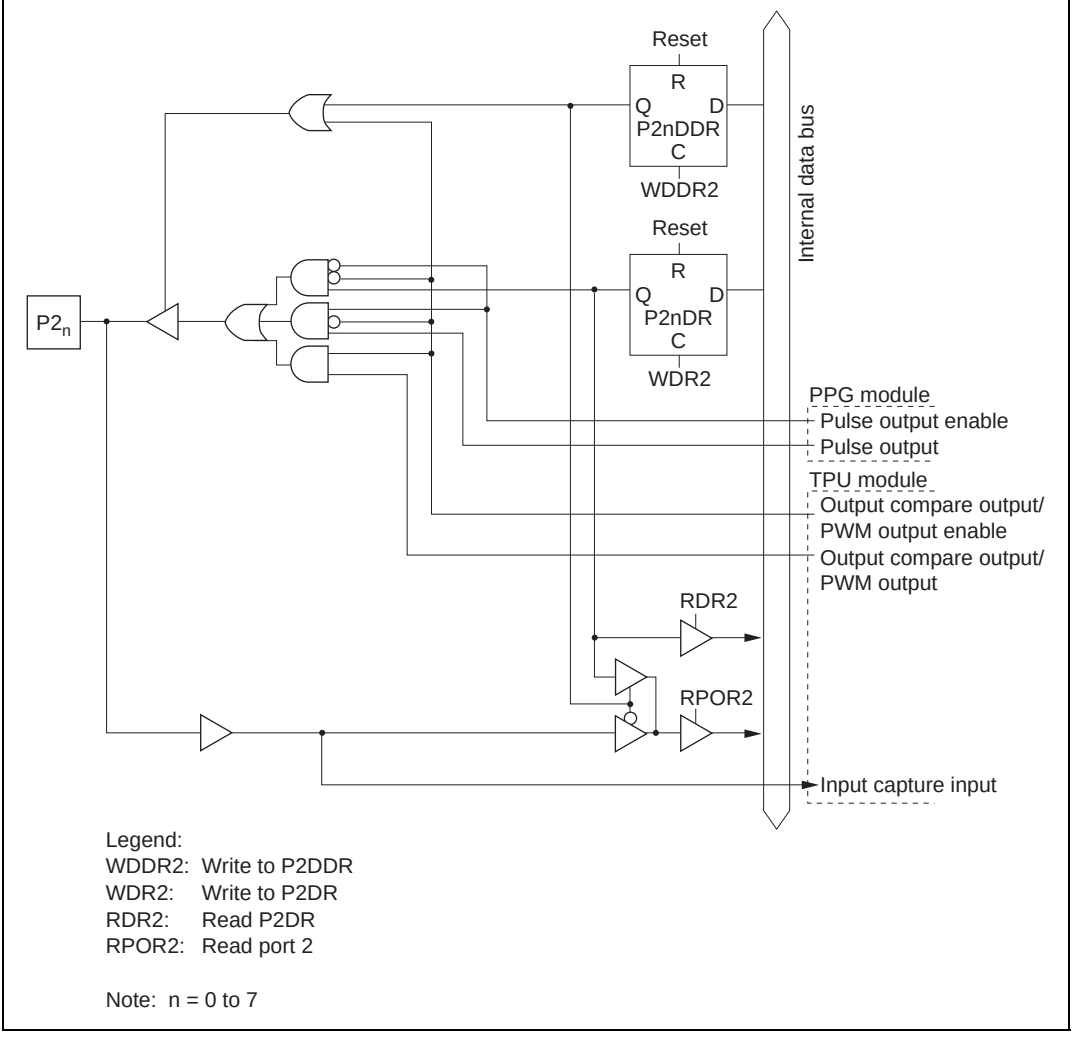


Figure C.2 Port 2 Block Diagram (Pins $P2_0$ to $P2_7$)

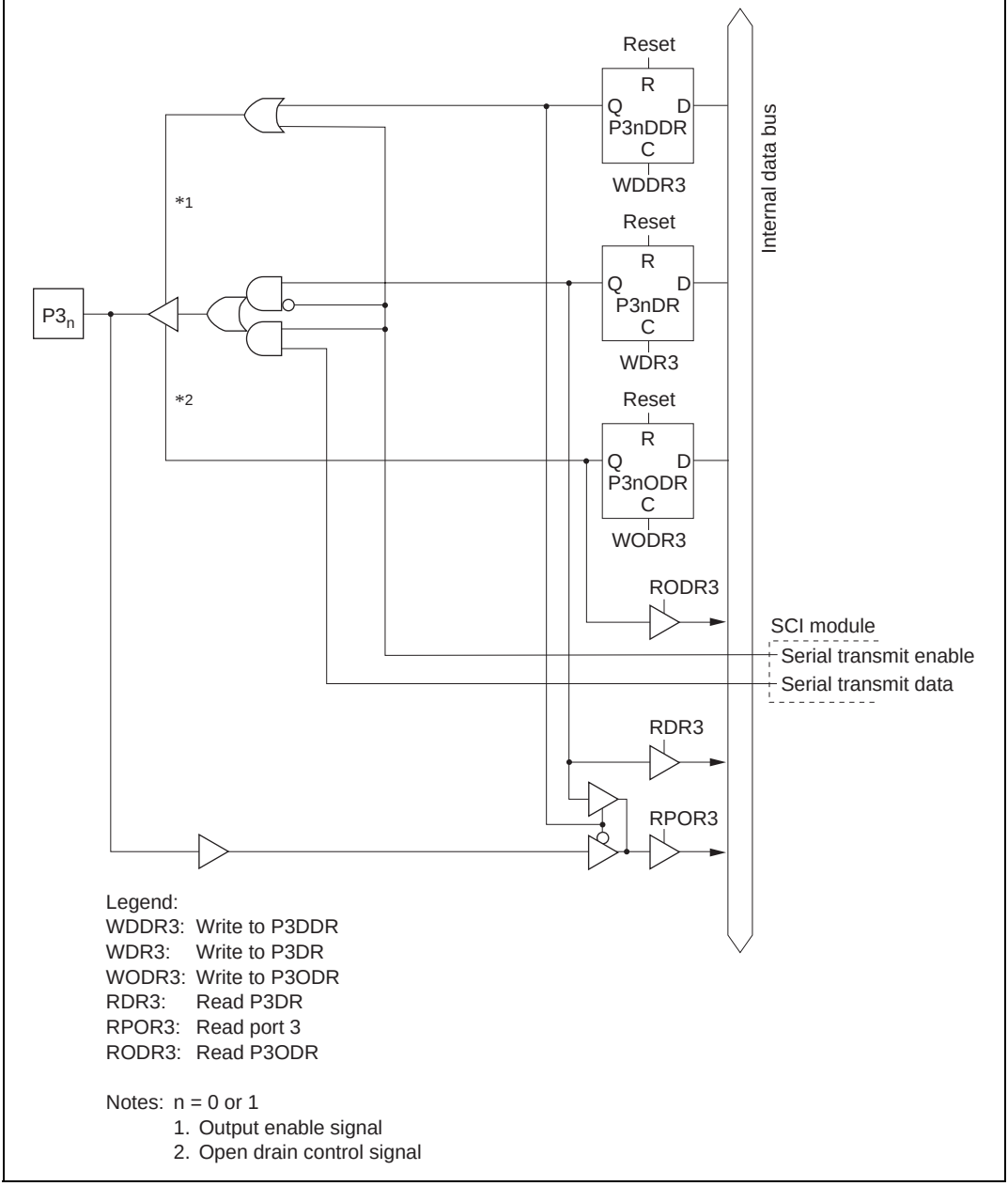


Figure C.3 (a) Port 3 Block Diagram (Pins $P3_0$ and $P3_1$)

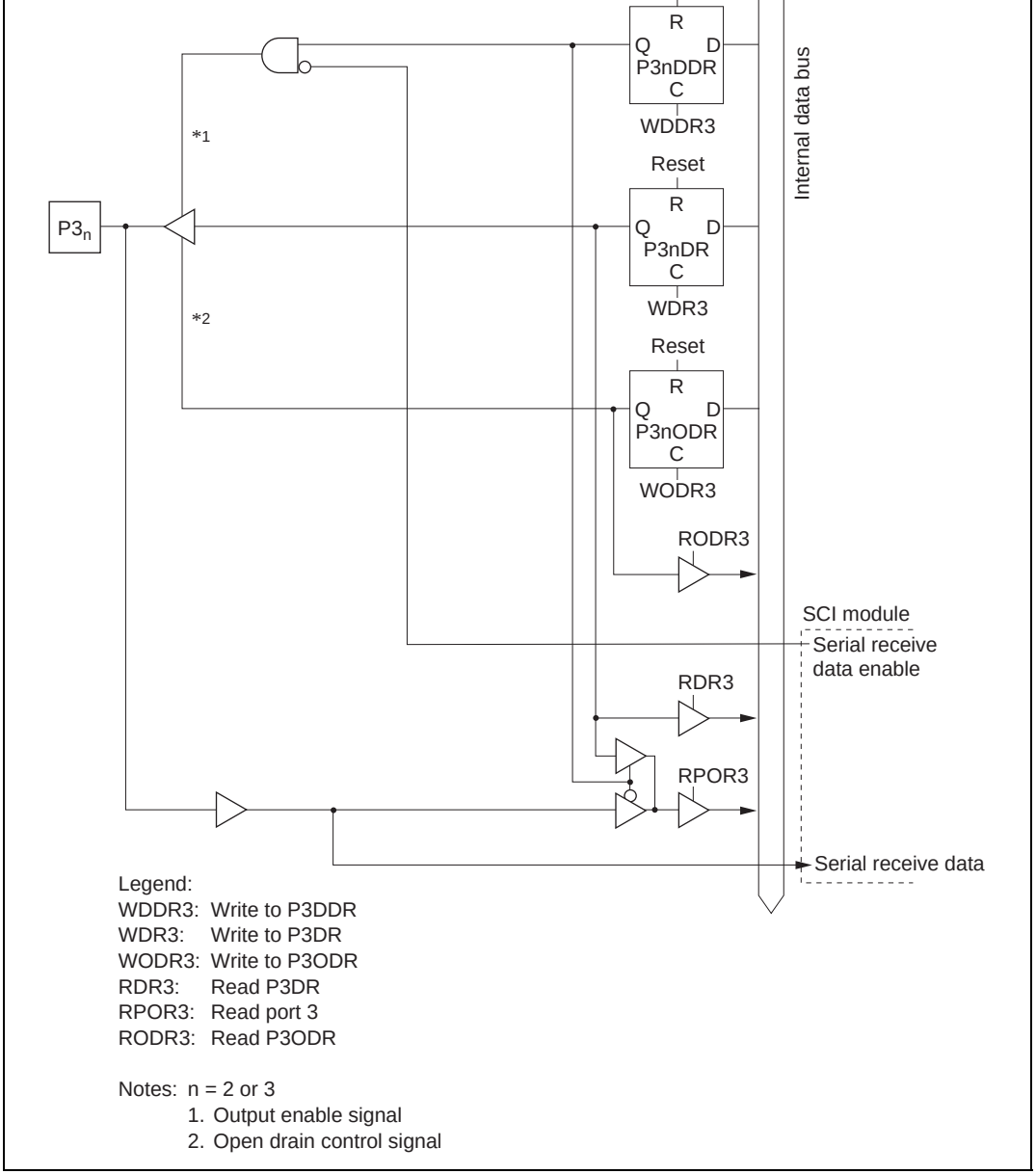
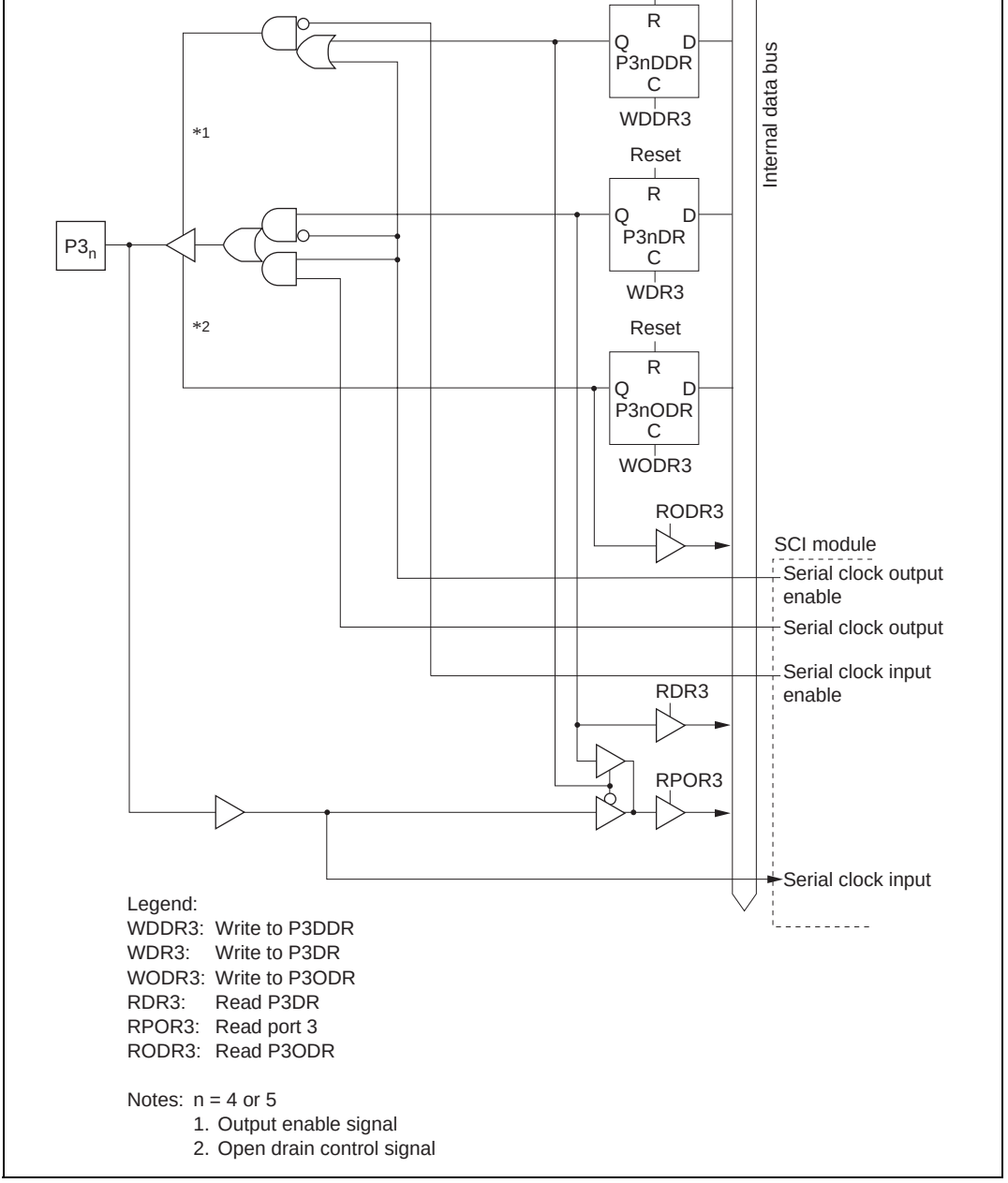


Figure C.3 (b) Port 3 Block Diagram (Pins $P3_2$ and $P3_3$)



Legend:
WDDR3: Write to P3DDR
WDR3: Write to P3DR
WODR3: Write to P3ODR
RDR3: Read P3DR
RPOR3: Read port 3
RODR3: Read P3ODR

Notes: n = 4 or 5
1. Output enable signal
2. Open drain control signal

Figure C.3 (c) Port 3 Block Diagram (Pins $P3_4$ and $P3_5$)

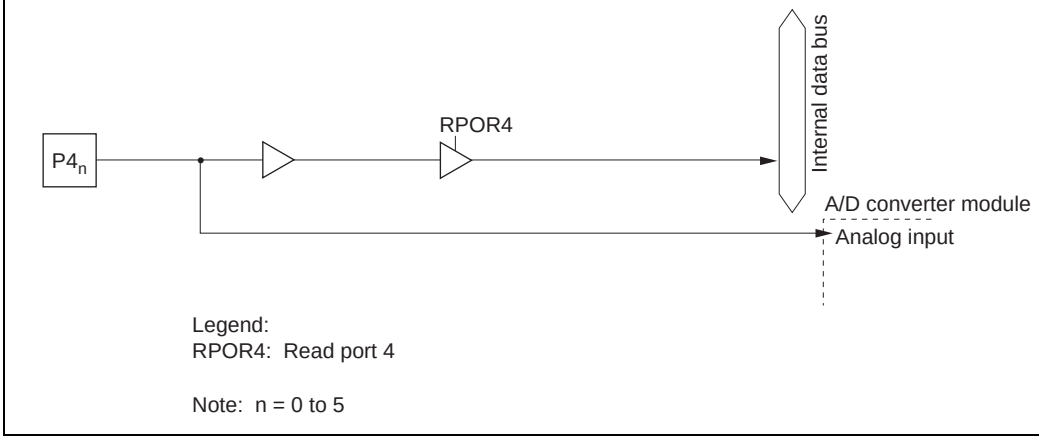


Figure C.4 (a) Port 4 Block Diagram (Pins $P4_0$ to $P4_5$)

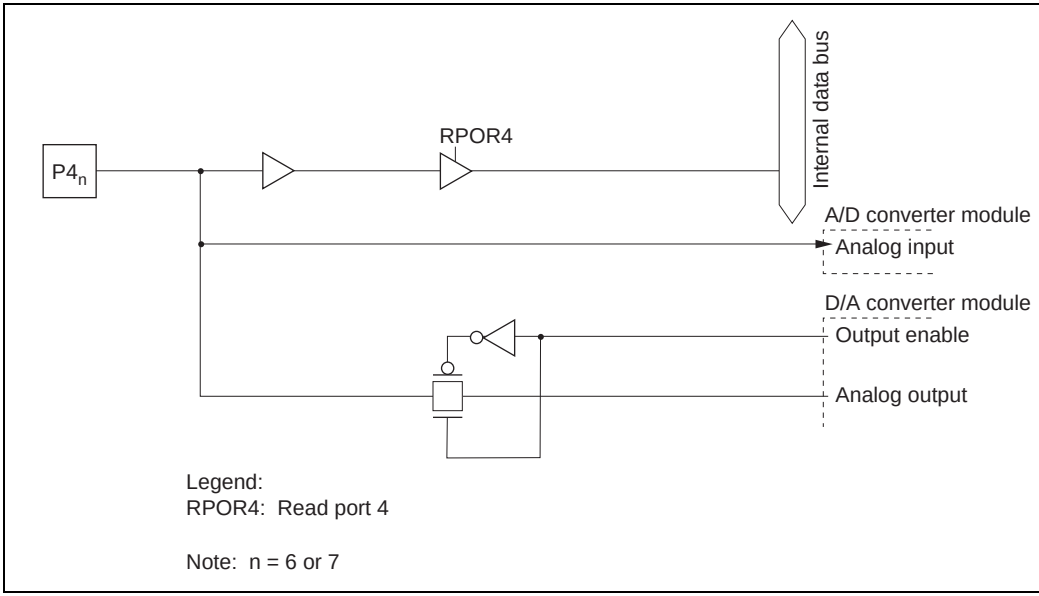


Figure C.4 (b) Port 4 Block Diagram (Pins $P4_6$ and $P4_7$)

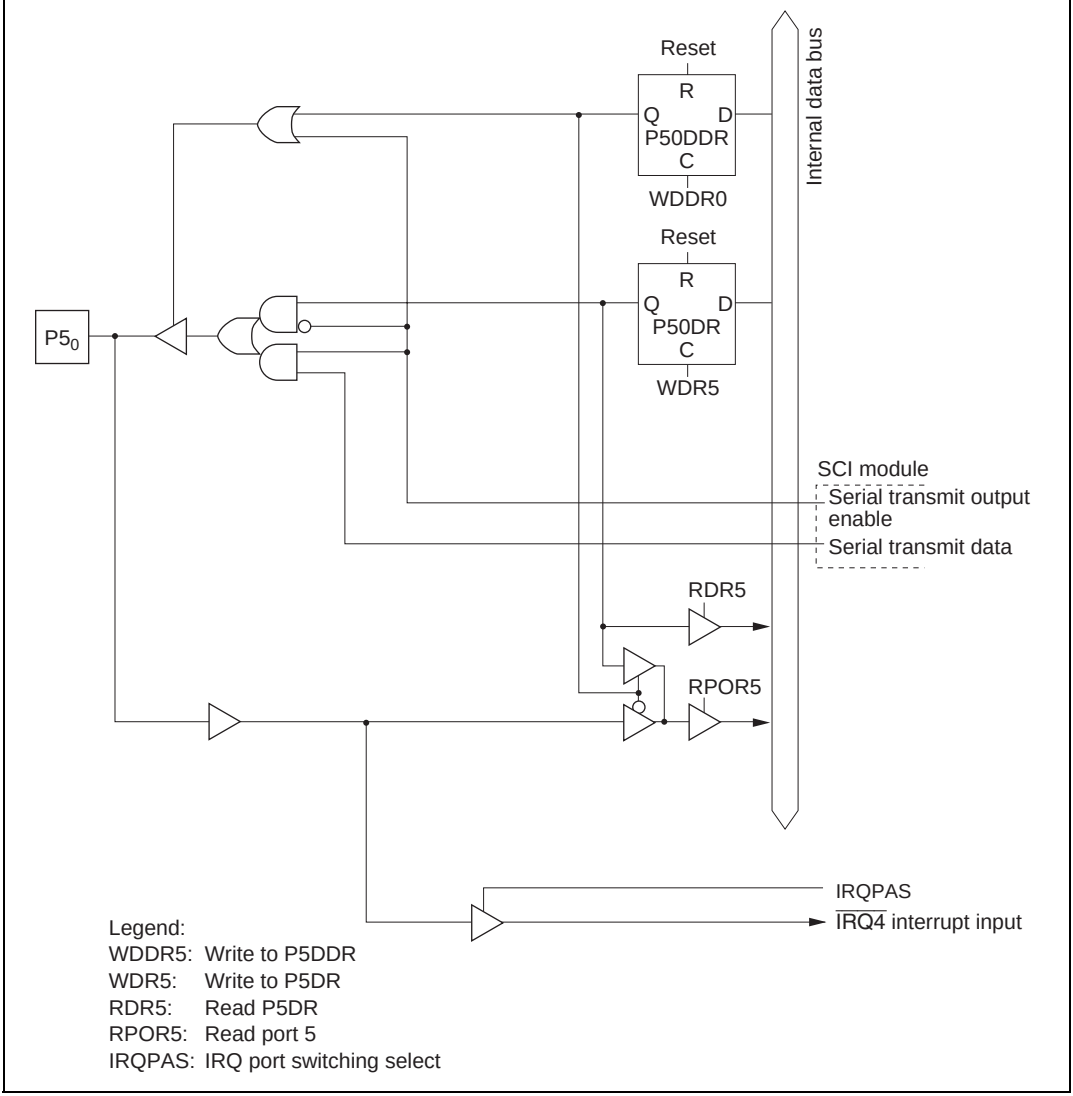


Figure C.5 (a) Port 5 Block Diagram (Pin $P5_0$)

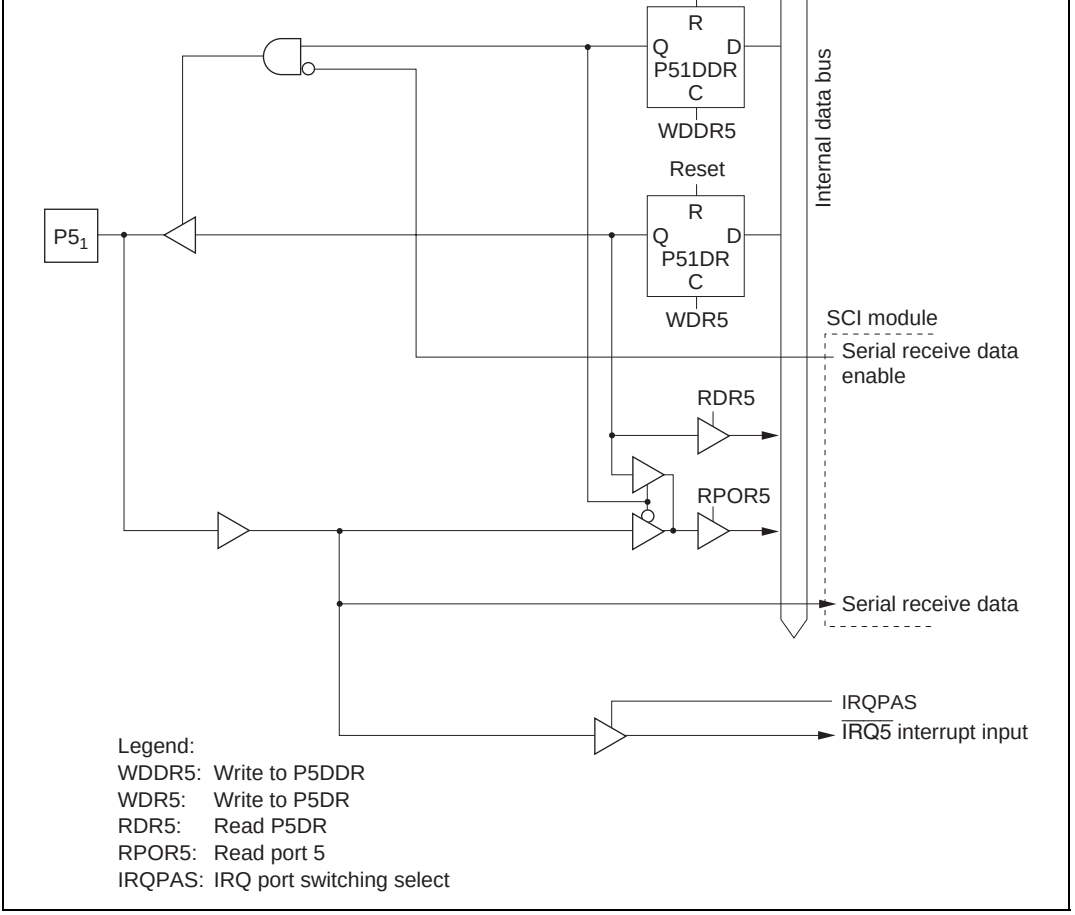


Figure C.5 (b) Port 5 Block Diagram (Pin P5$_1$)

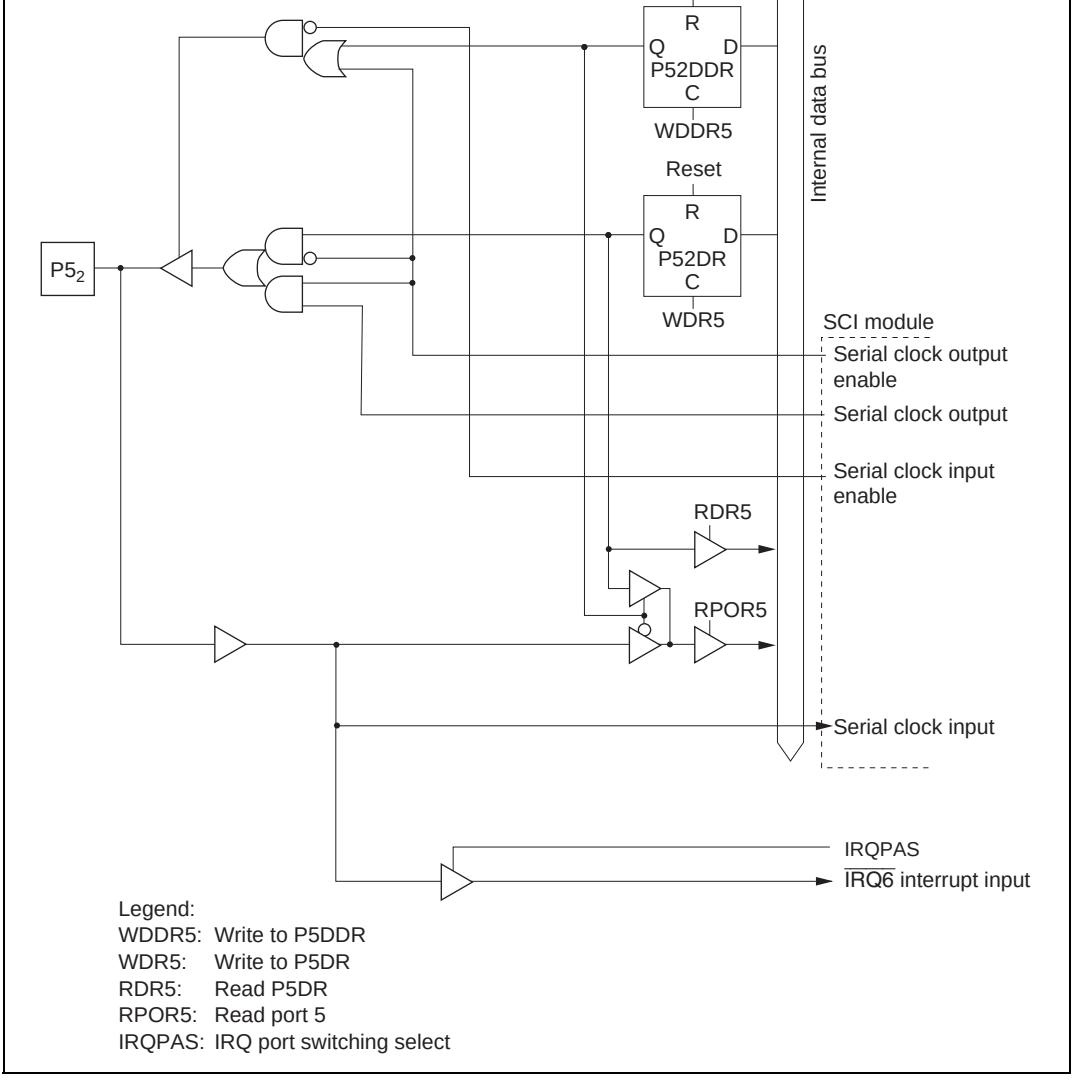


Figure C.5 (c) Port 5 Block Diagram (Pin $P5_2$)

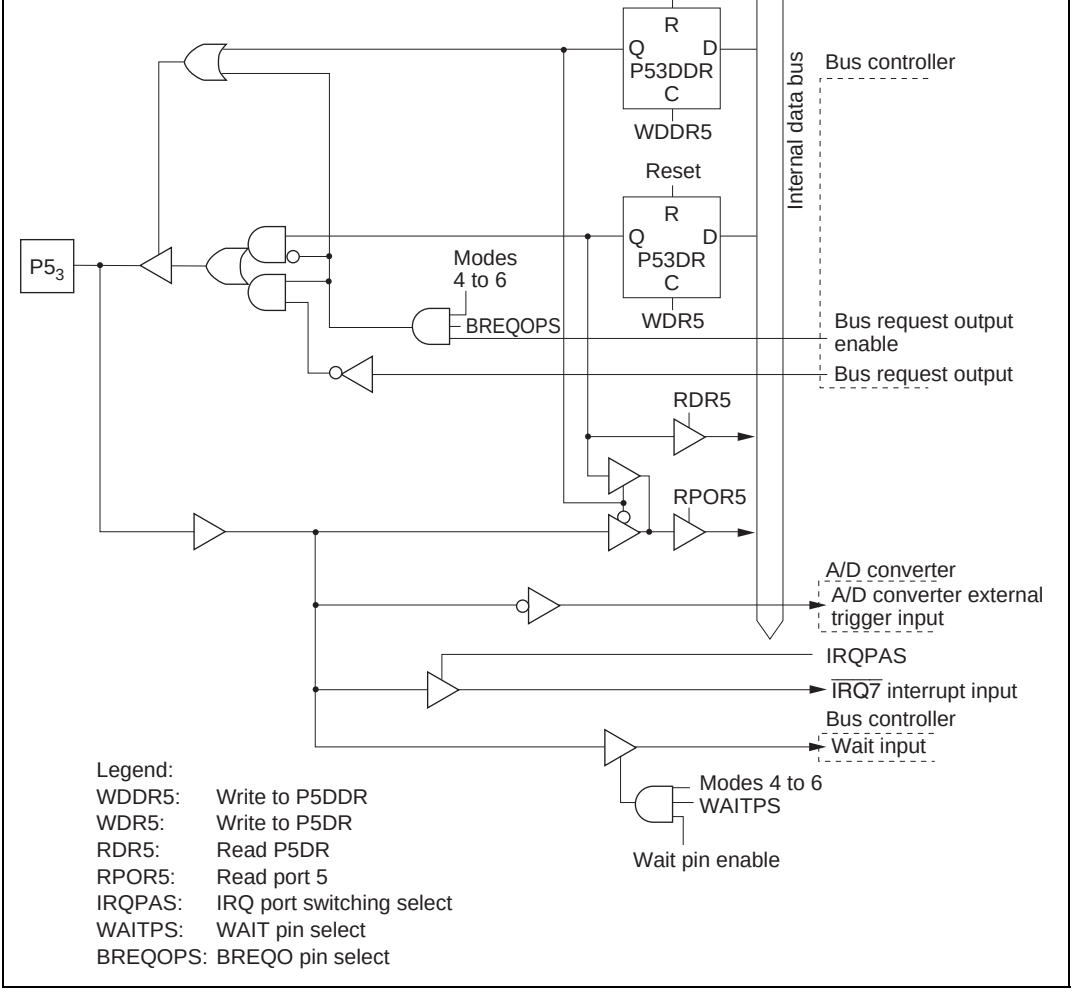


Figure C.5 (d) Port 5 Block Diagram (Pin P5$_3$)

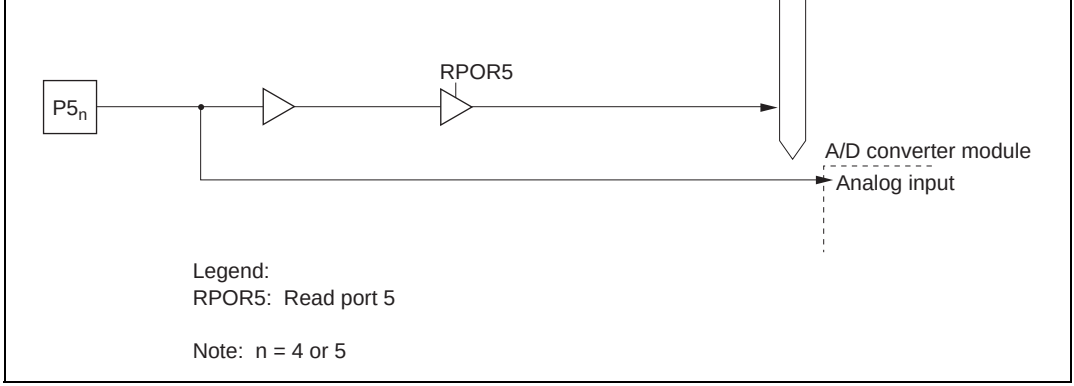


Figure C.5 (e) Port 5 Block Diagram (Pins $P5_4$ and P_5)

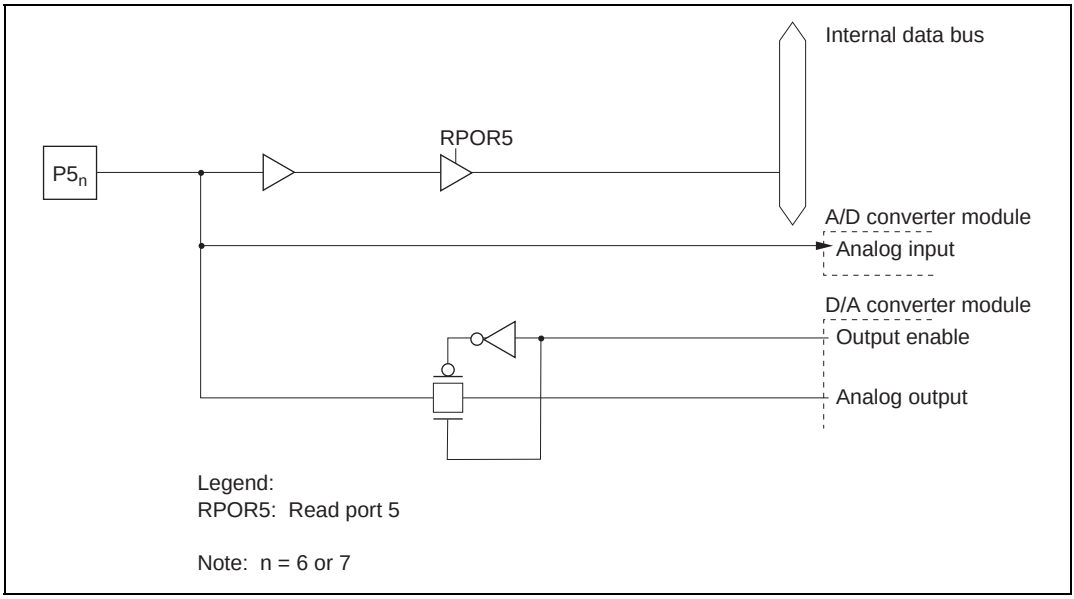


Figure C.5 (f) Port 5 Block Diagram (Pins $P5_6$ and P_7)

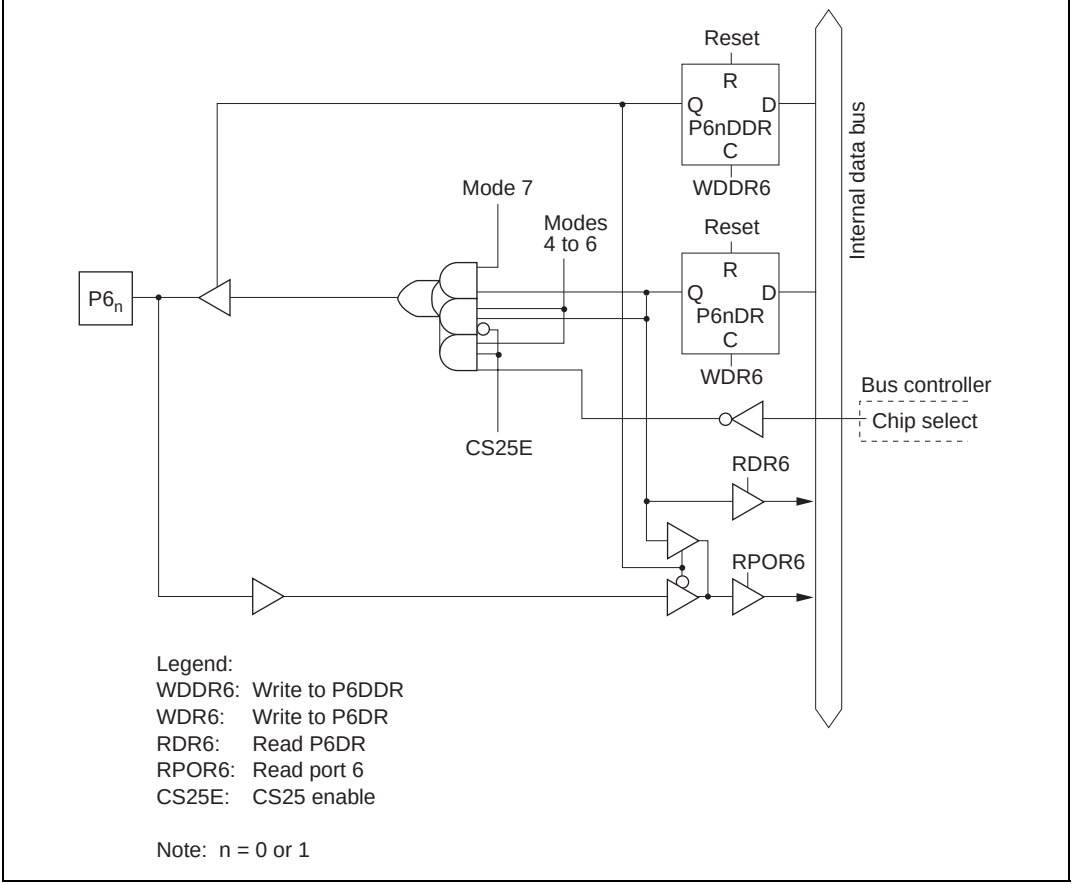


Figure C.6 (a) Port 6 Block Diagram (Pins $P6_0$ and $P6_1$)

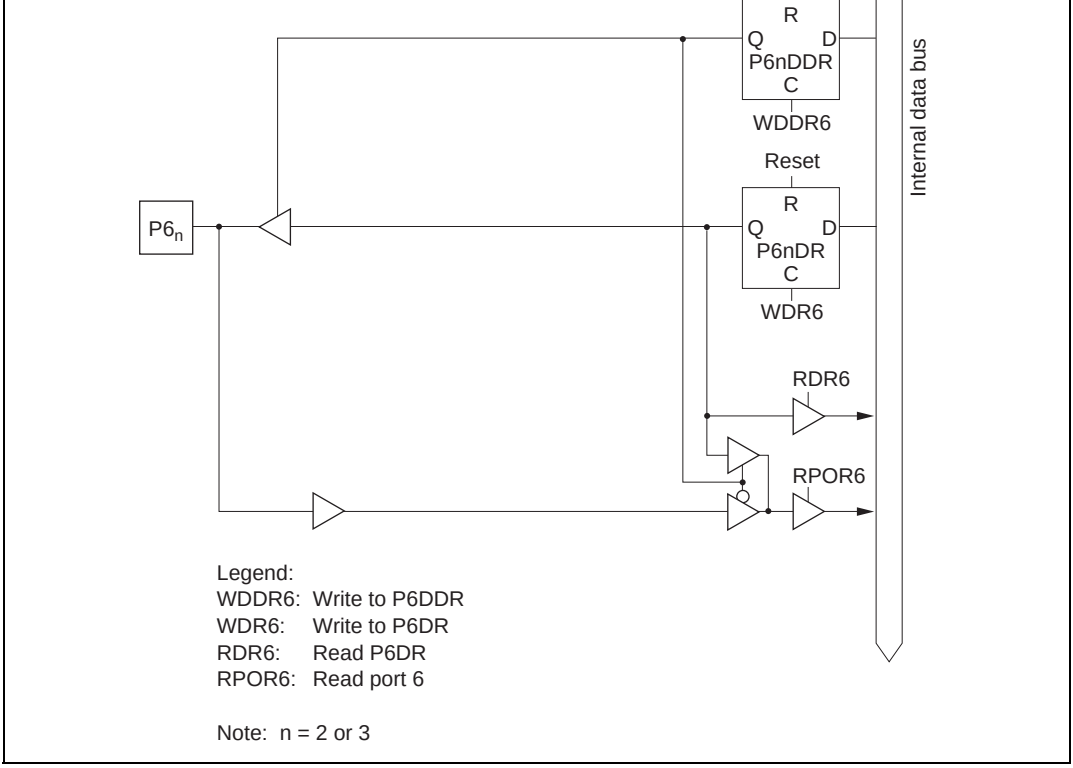


Figure C.6 (b) Port 6 Block Diagram (Pins $P6_2$ and $P6_3$)

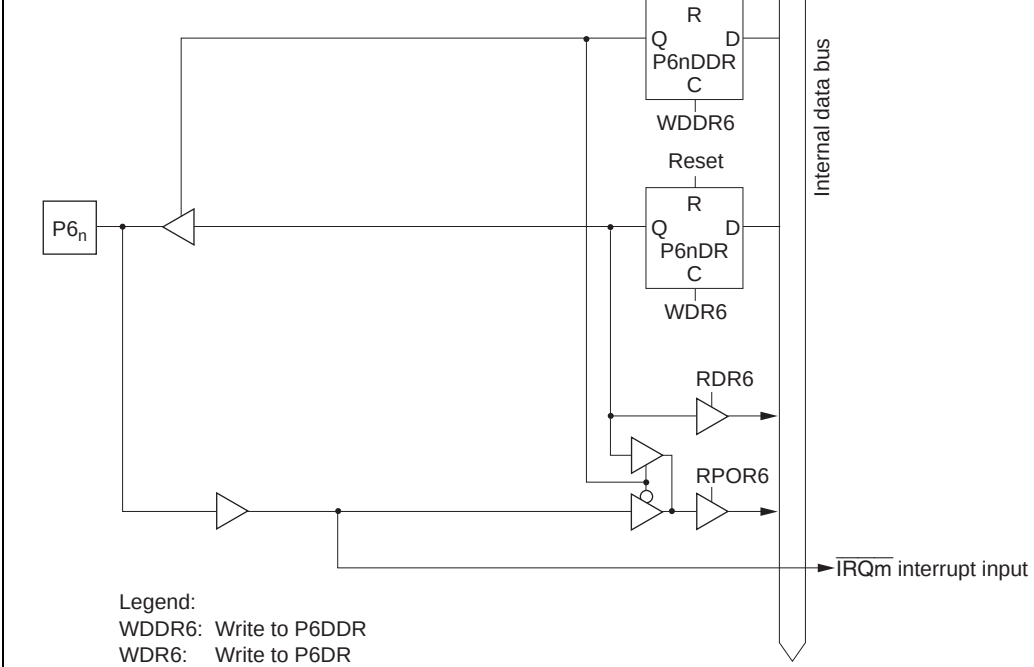


Figure C.6 (c) Port 6 Block Diagram (Pins $P6_4$ and $P6_5$)

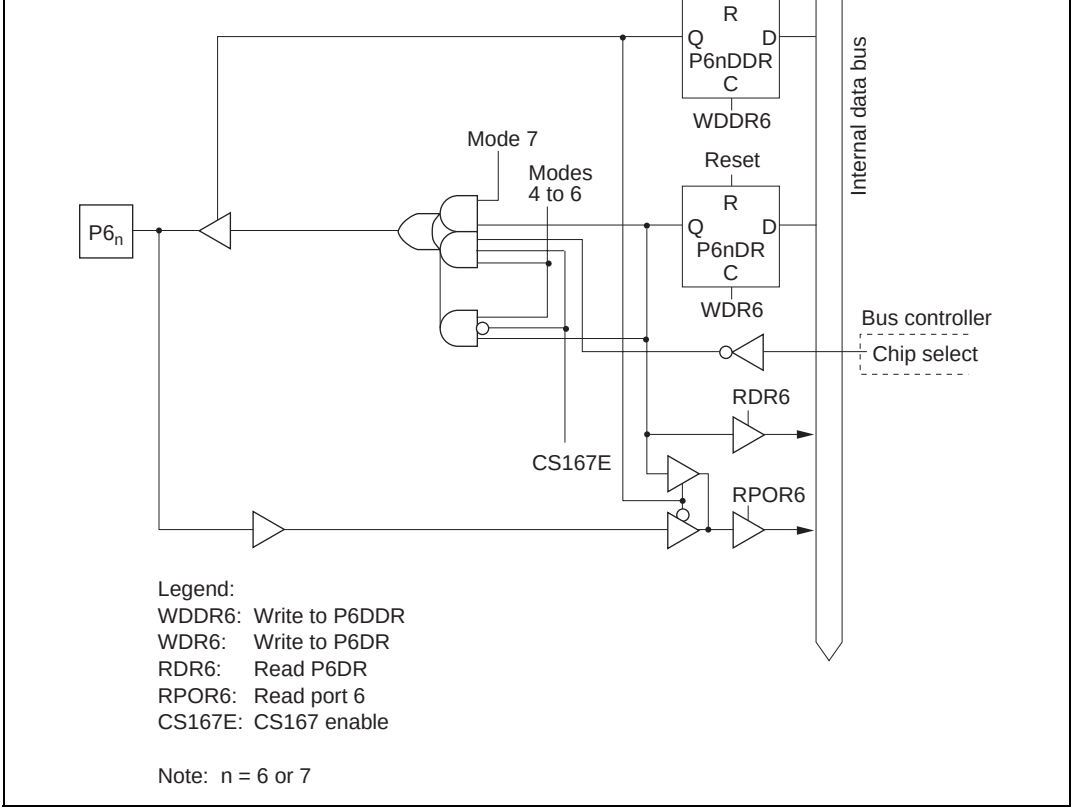


Figure C.6 (d) Port 6 Block Diagram (Pins $P6_6$ and $P6_7$)

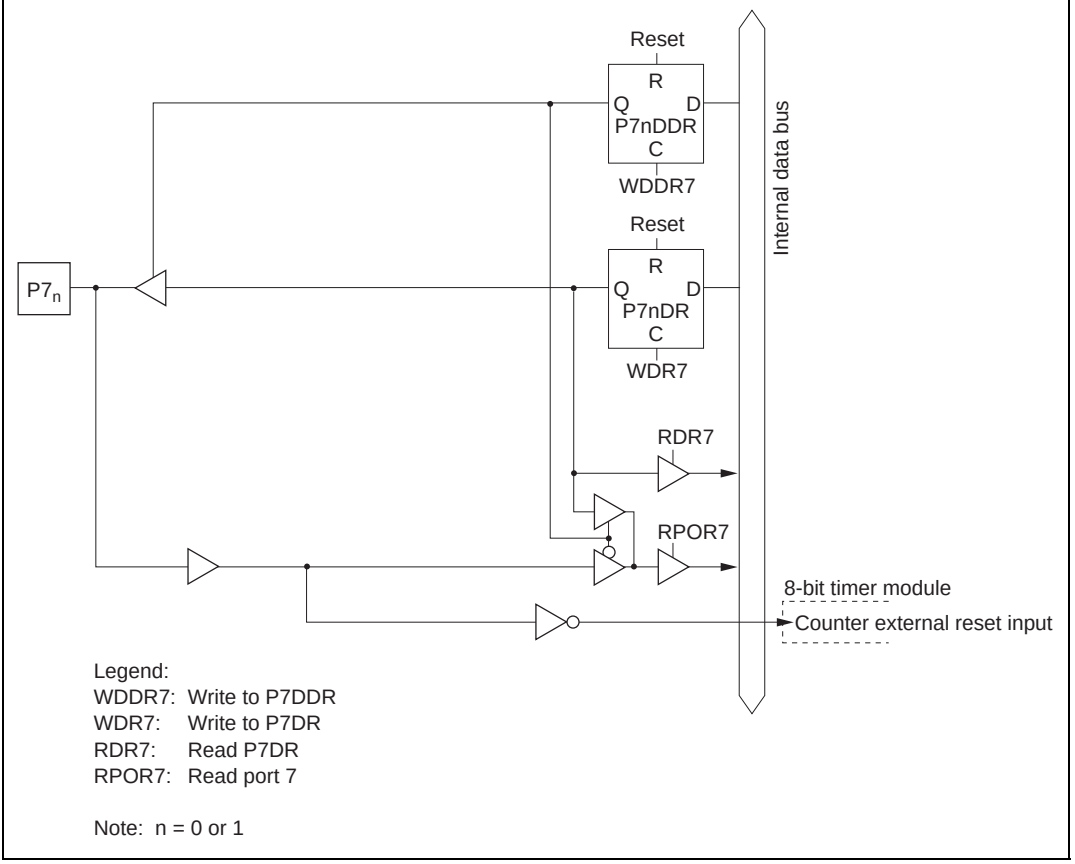


Figure C.7 (a) Port 7 Block Diagram (Pins $P7_0$ and $P7_1$)

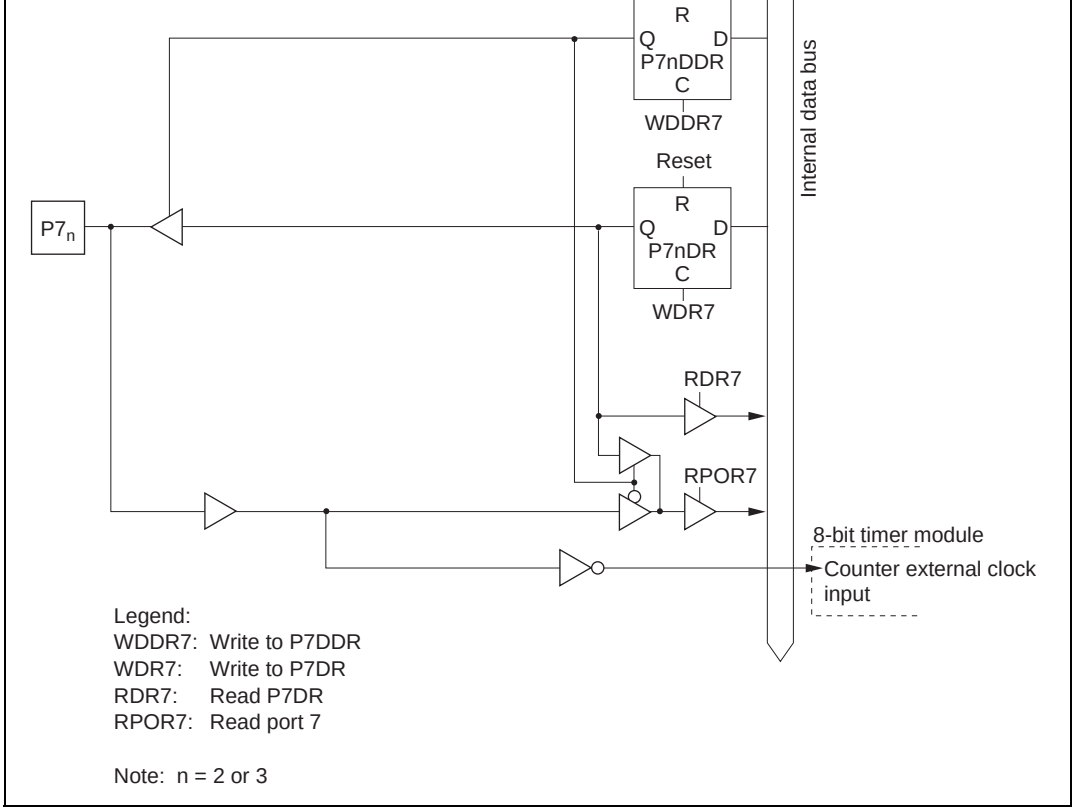


Figure C.7 (b) Port 7 Block Diagram (Pins $P7_2$ and $P7_3$)

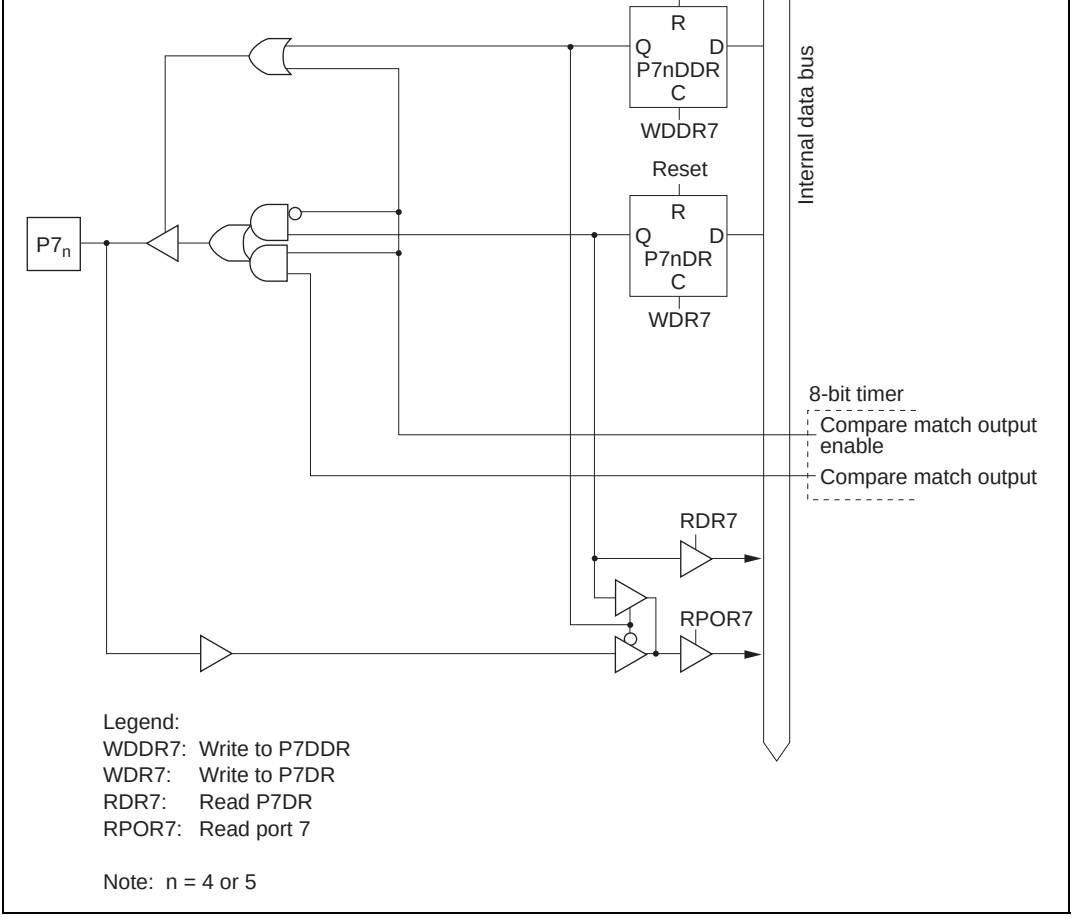


Figure C.7 (c) Port 7 Block Diagram (Pins $P7_4$ and $P7_5$)

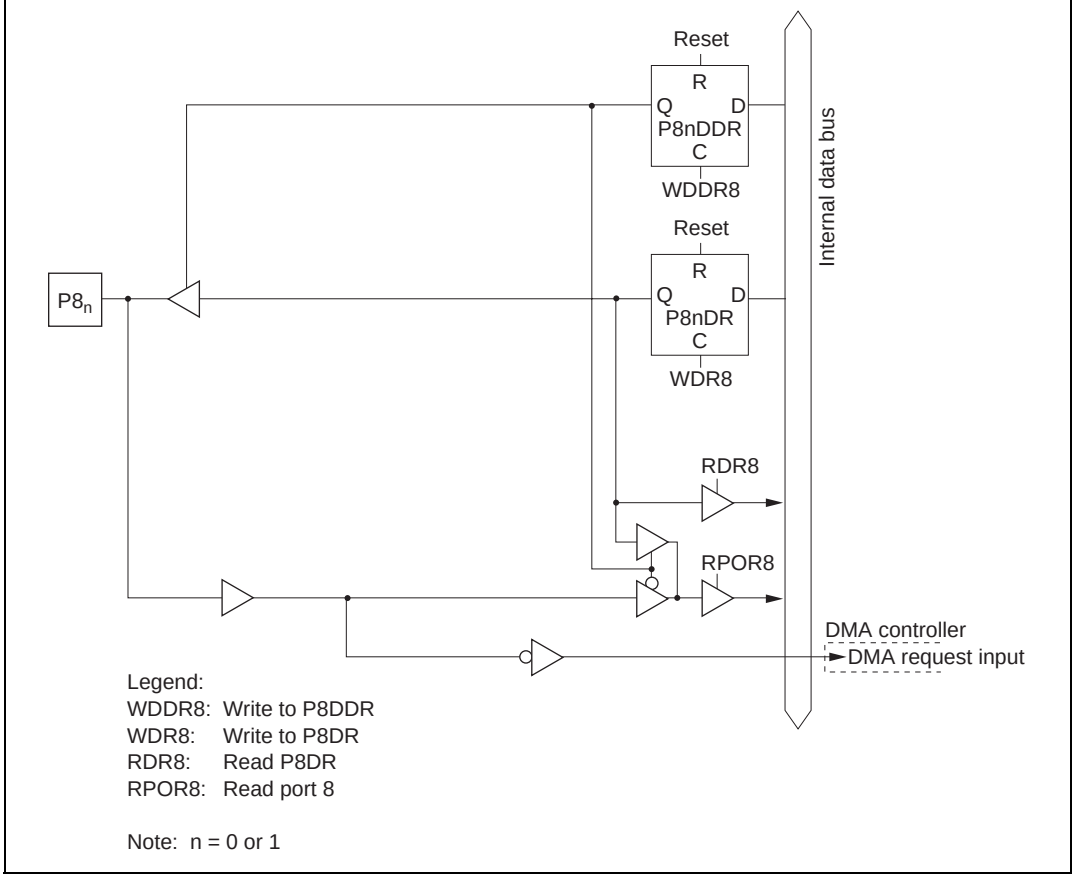


Figure C.8 (a) Port 8 Block Diagram (Pins $P8_0$ and $P8_1$)

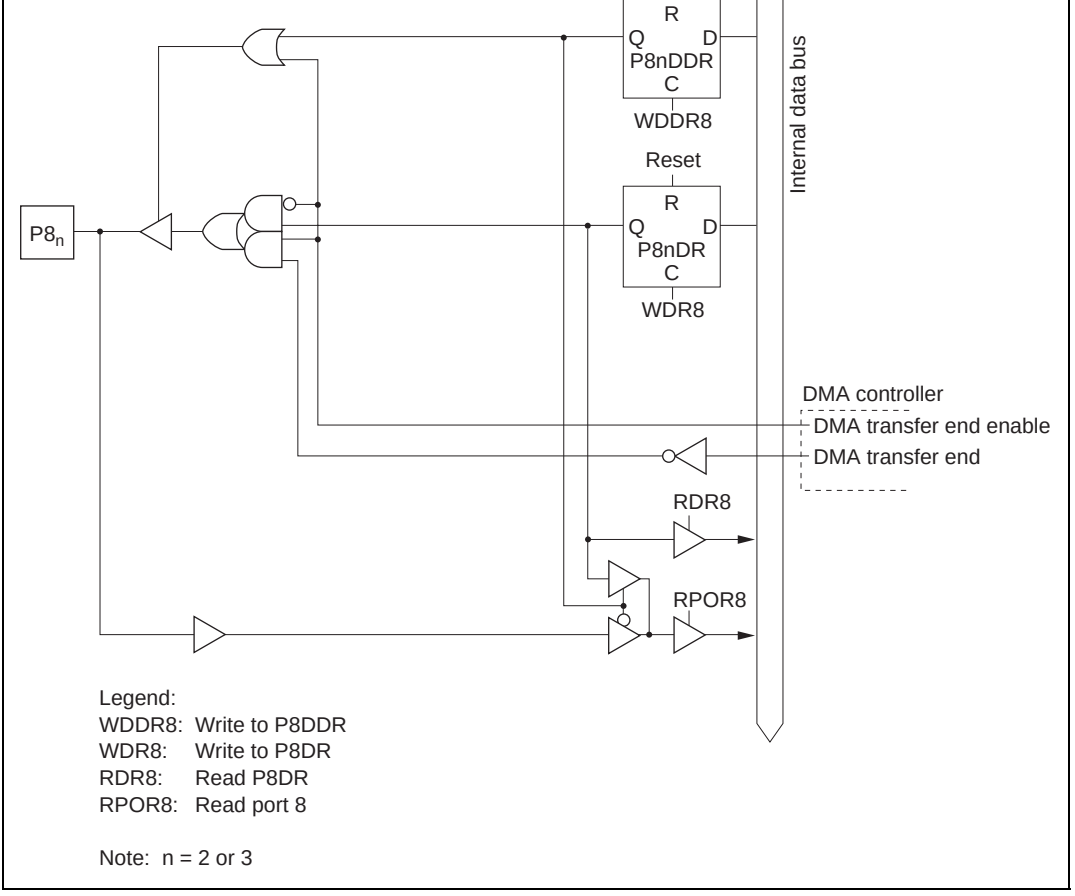


Figure C.8 (b) Port 8 Block Diagram (Pins $P8_2$ and $P8_3$)

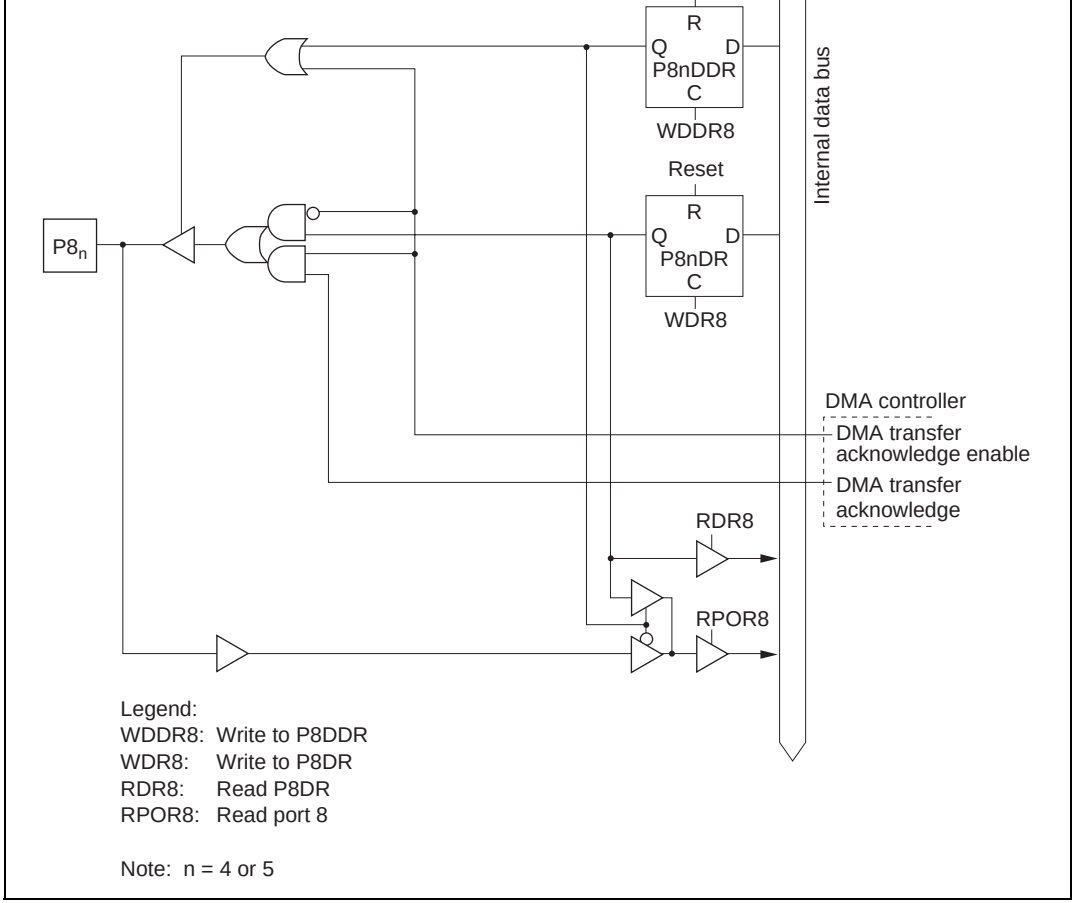


Figure C.8 (c) Port 8 Block Diagram (Pins $P8_4$ and $P8_5$)

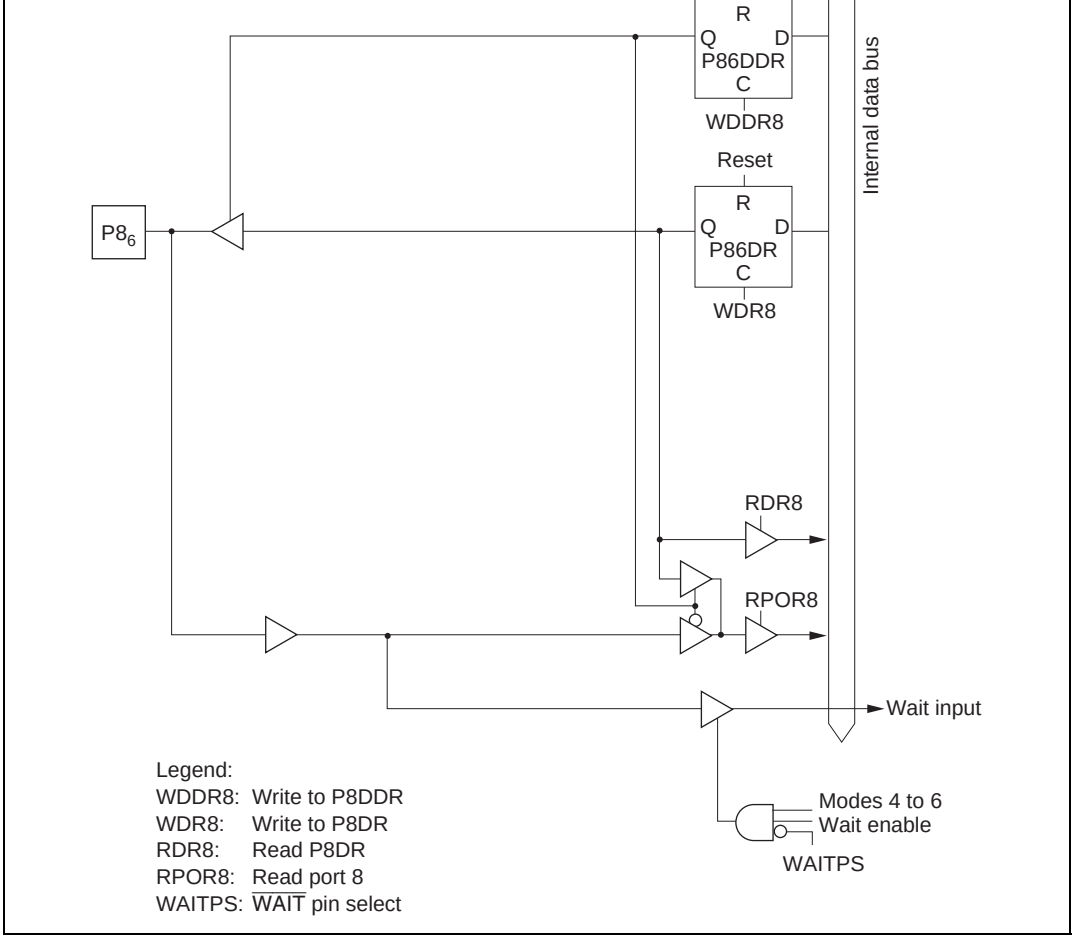


Figure C.8 (d) Port 8 Block Diagram (Pin P8$_6$)

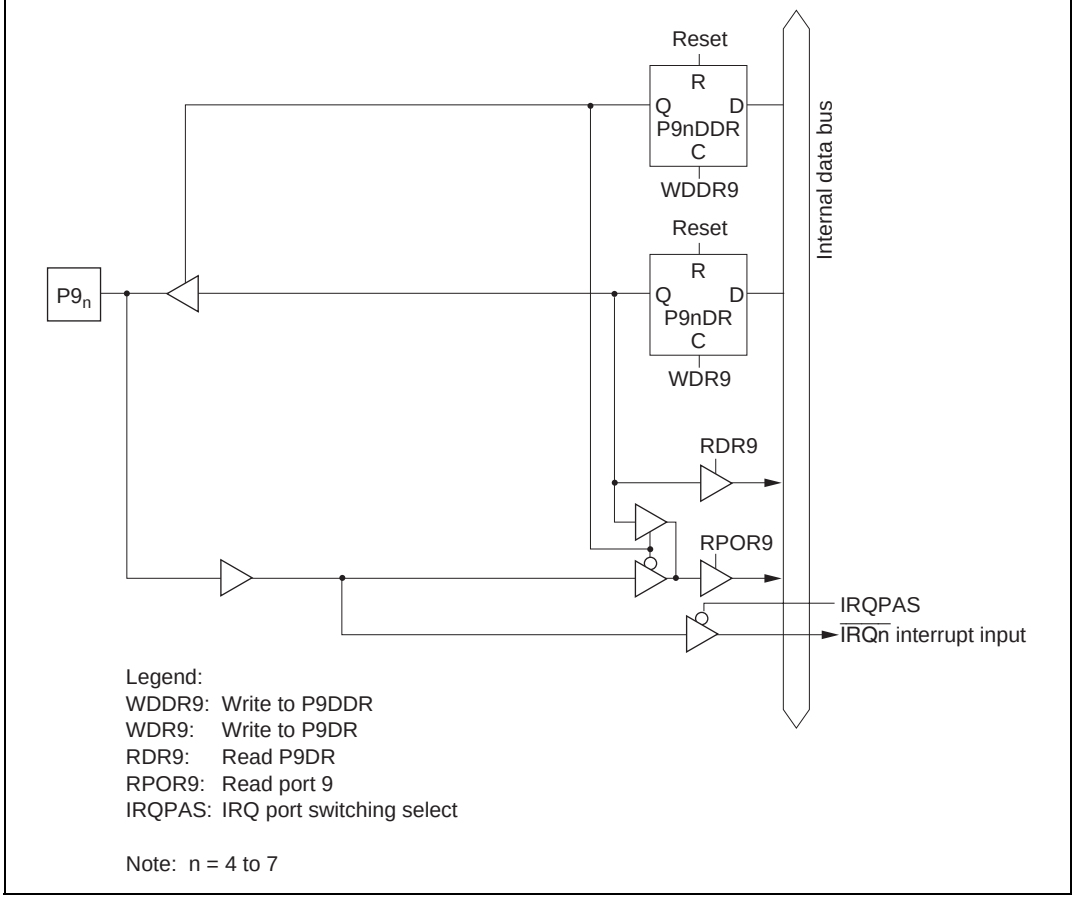


Figure C.9 (a) Port 9 Block Diagram (Pins $P9_4$ to $P9_7$)

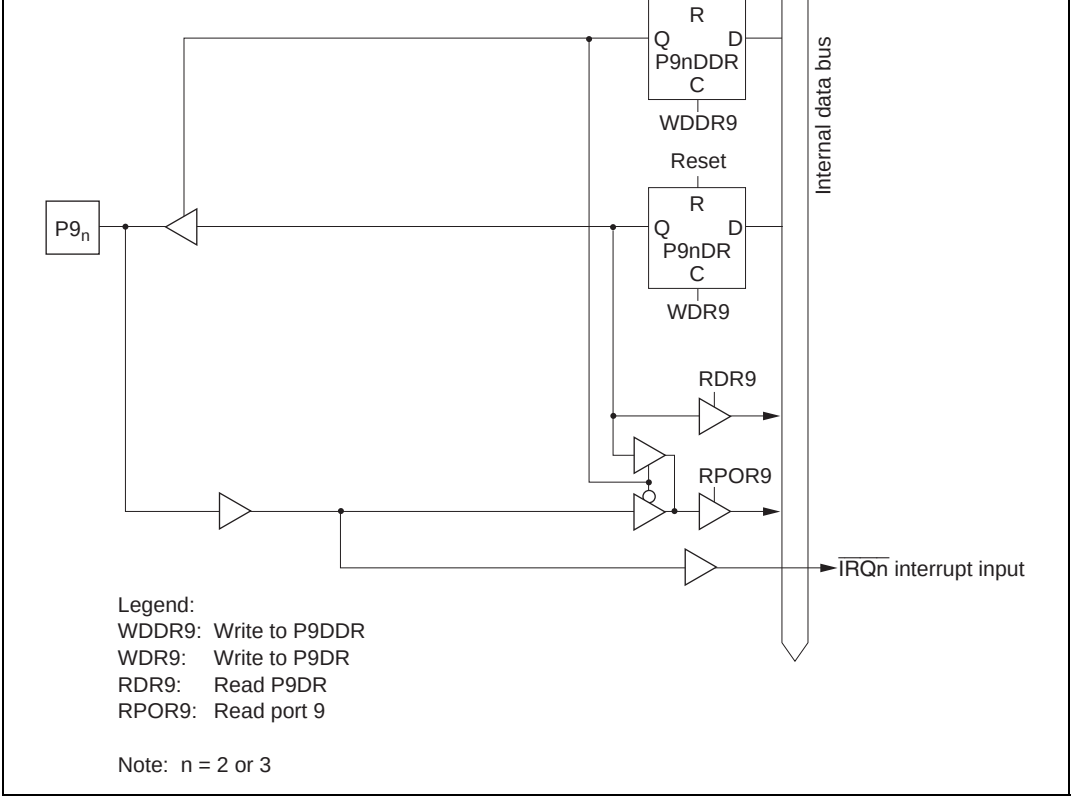


Figure C.9 (b) Port 9 Block Diagram (Pins $P9_2$ and $P9_3$)

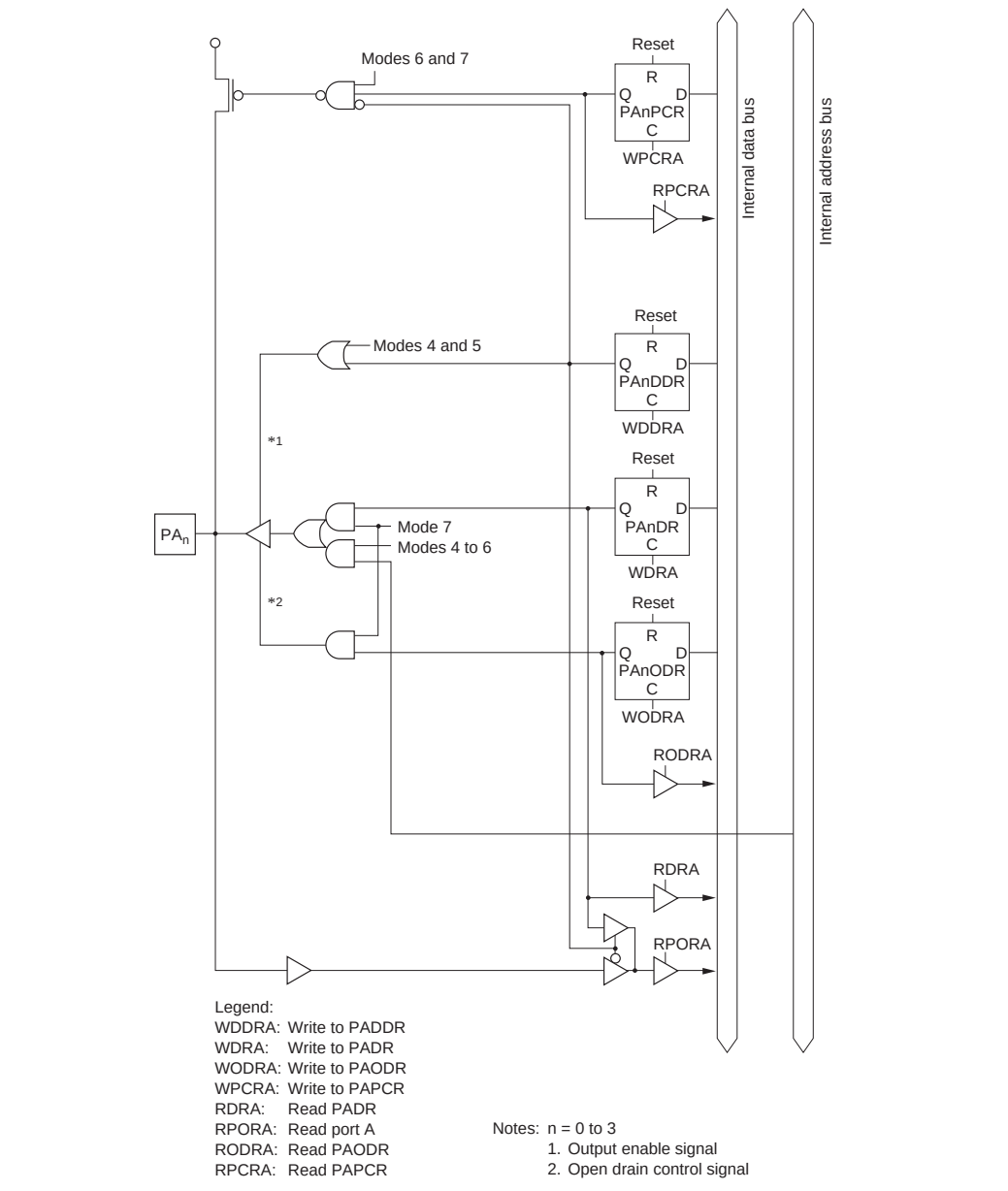


Figure C.10 (a) Port A Block Diagram (Pins PA_0 to PA_3)

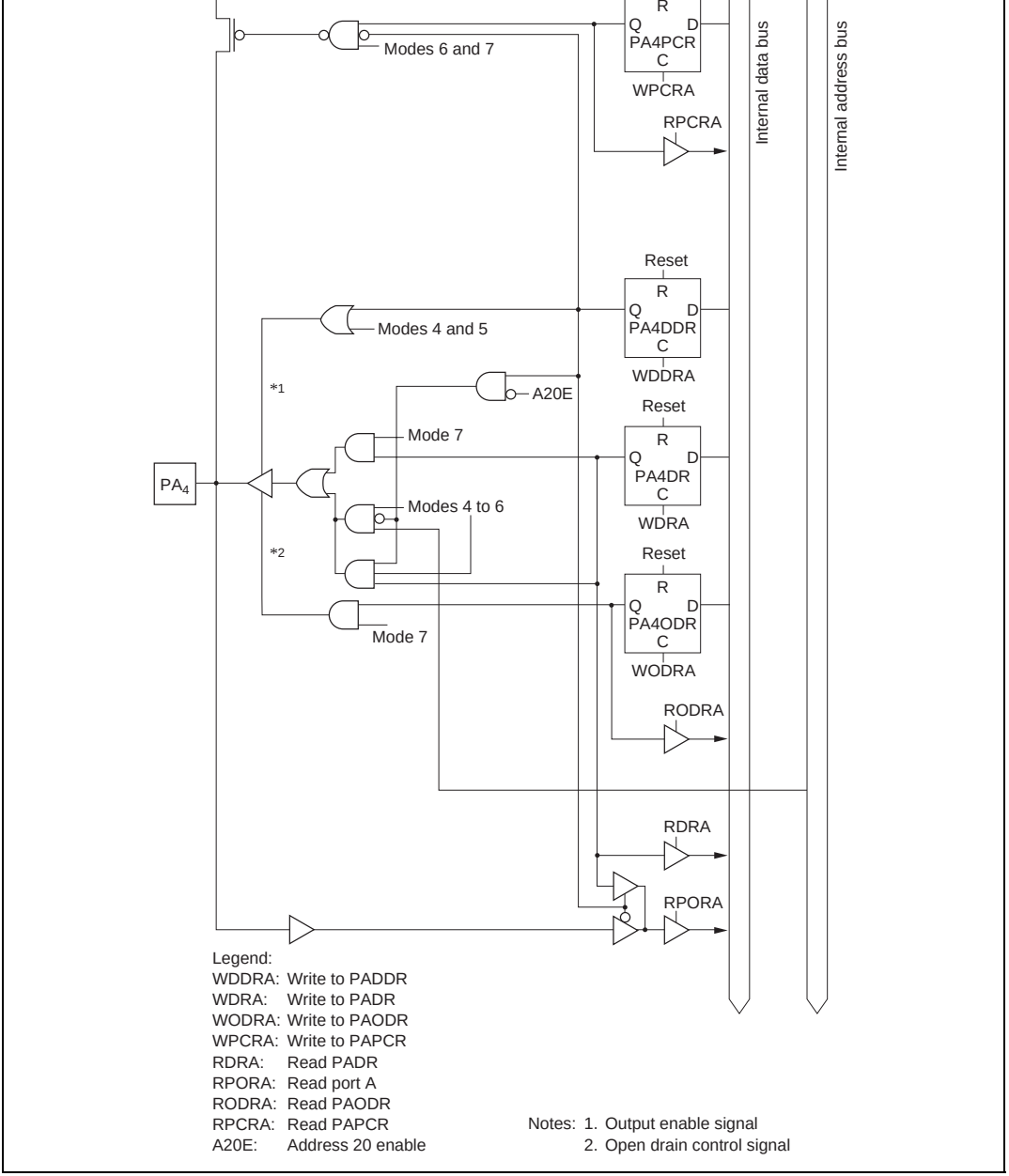


Figure C.10 (b) Port A Block Diagram (Pin PA_4)

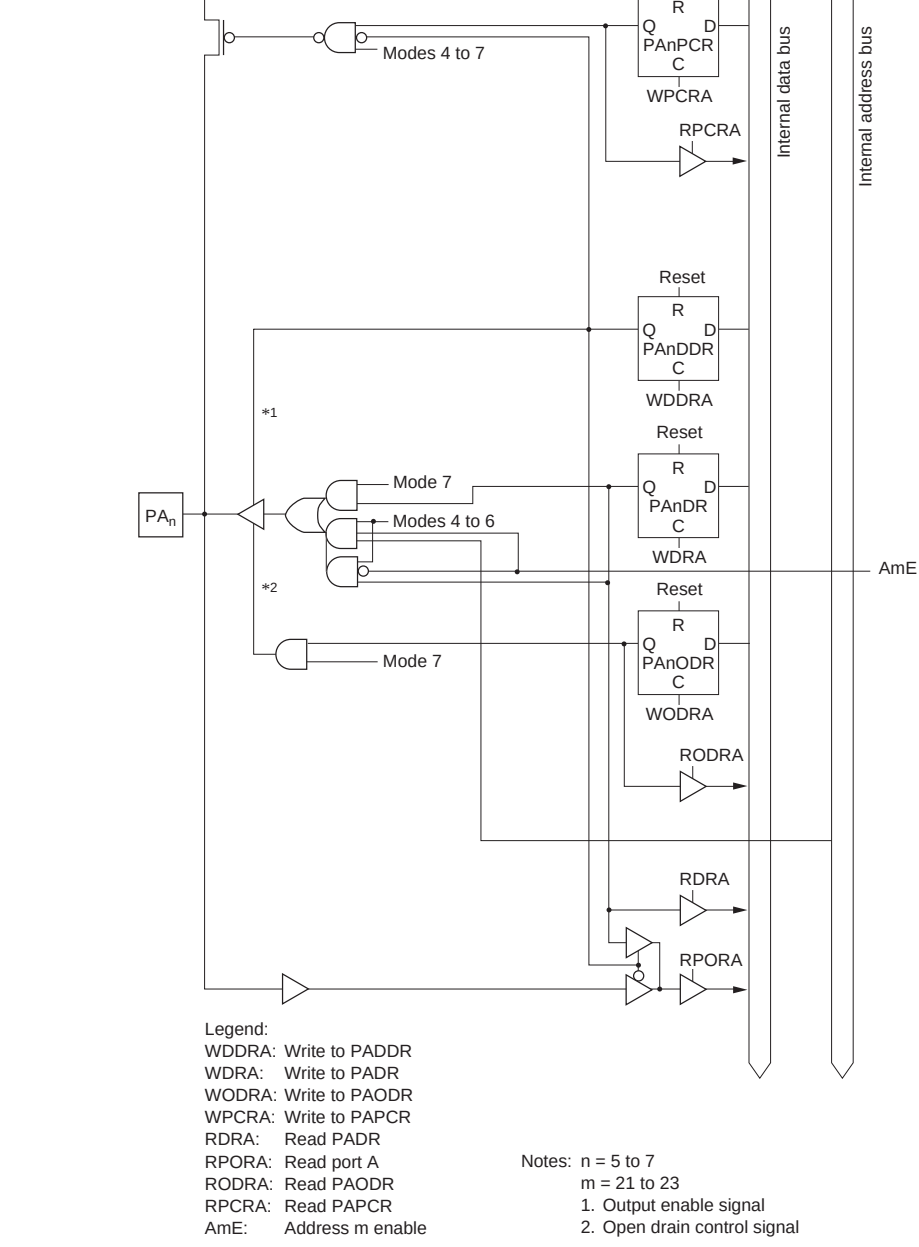


Figure C.10 (c) Port A Block Diagram (Pins PA_5 to PA_7)

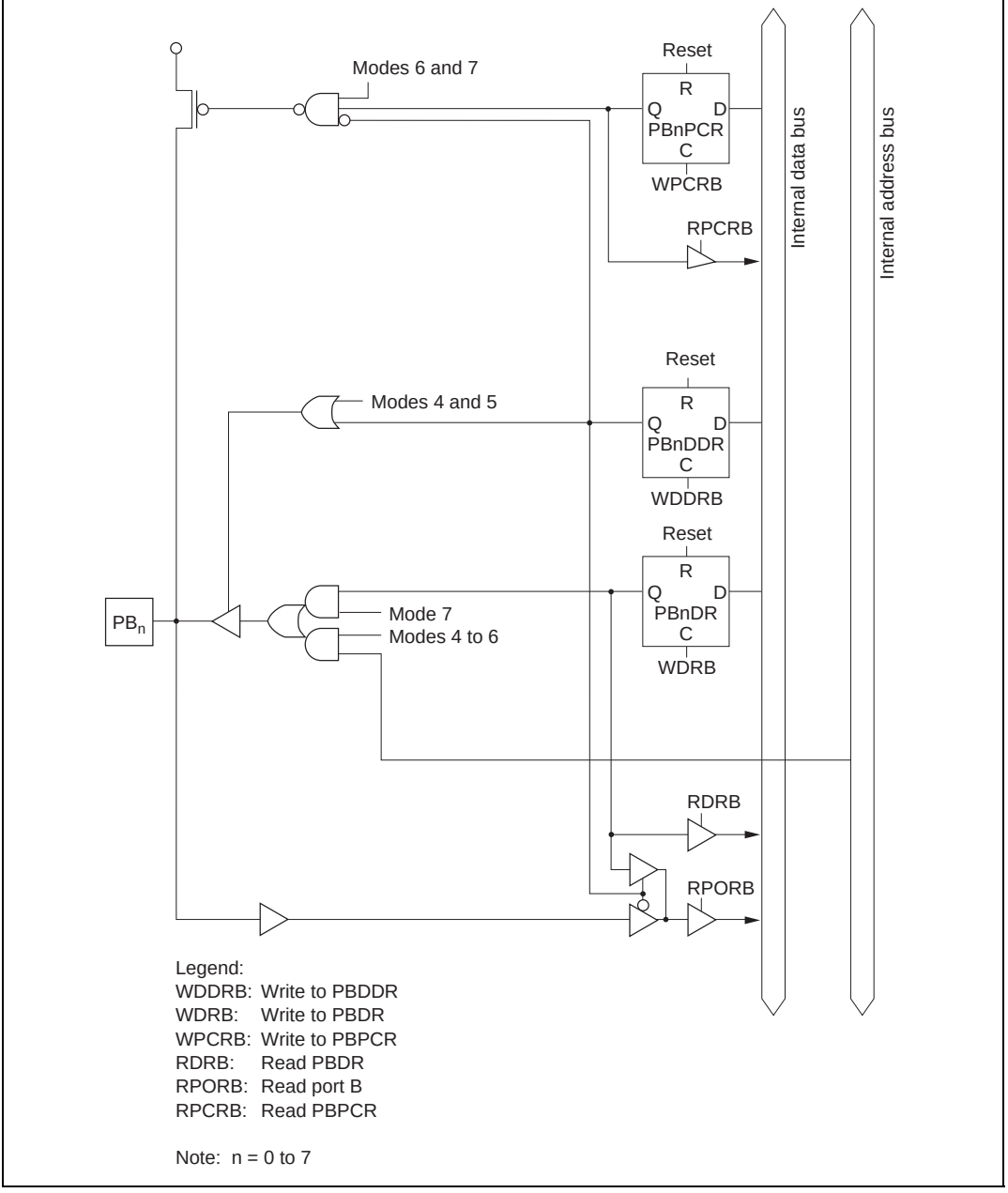


Figure C.11 Port B Block Diagram (Pins PB_0 to PB_7)

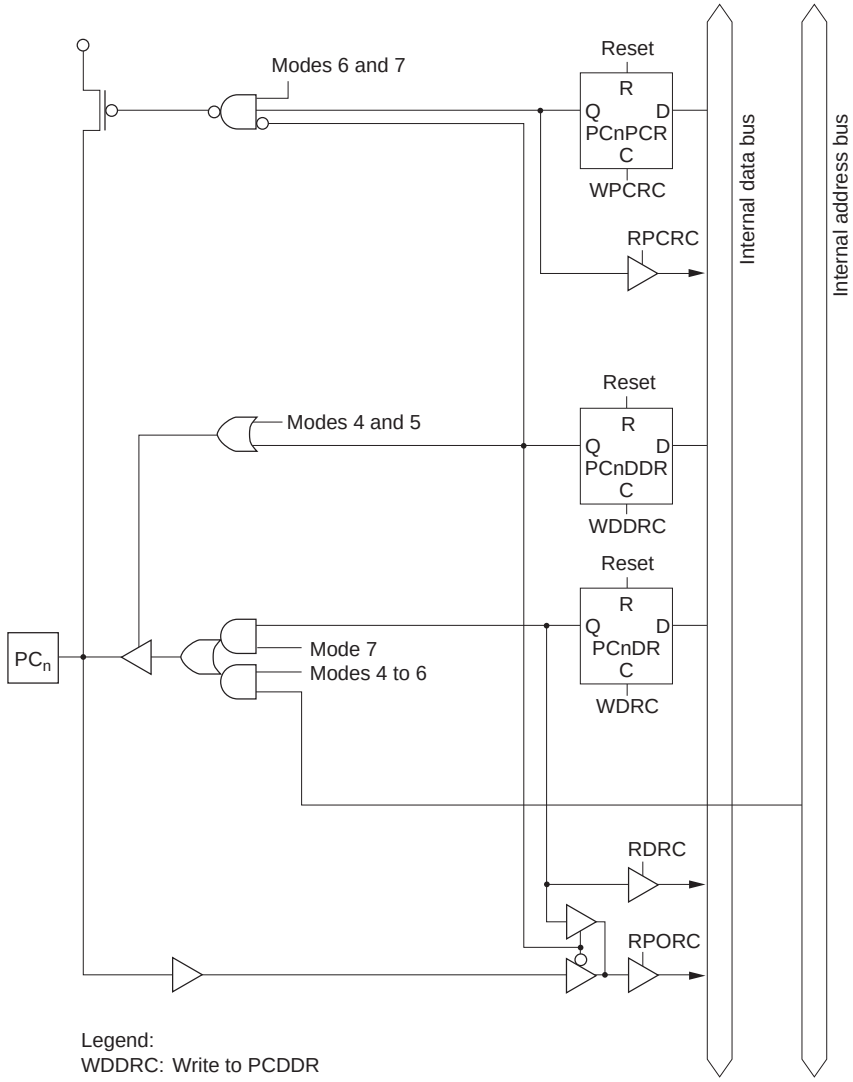


Figure C.12 Port C Block Diagram (Pins PC_0 to PC_7)

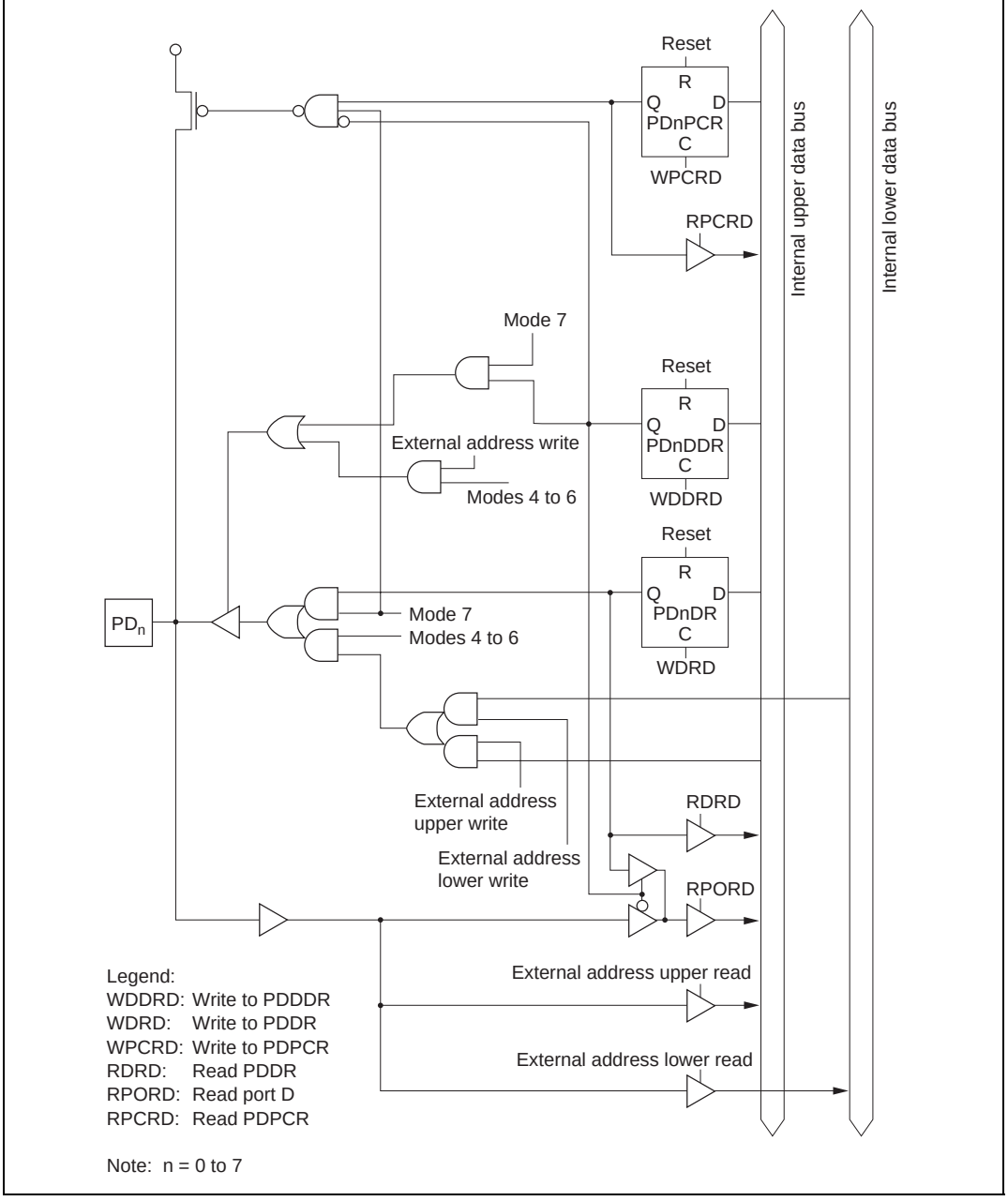


Figure C.13 Port D Block Diagram (Pins PD_0 to PD_7)

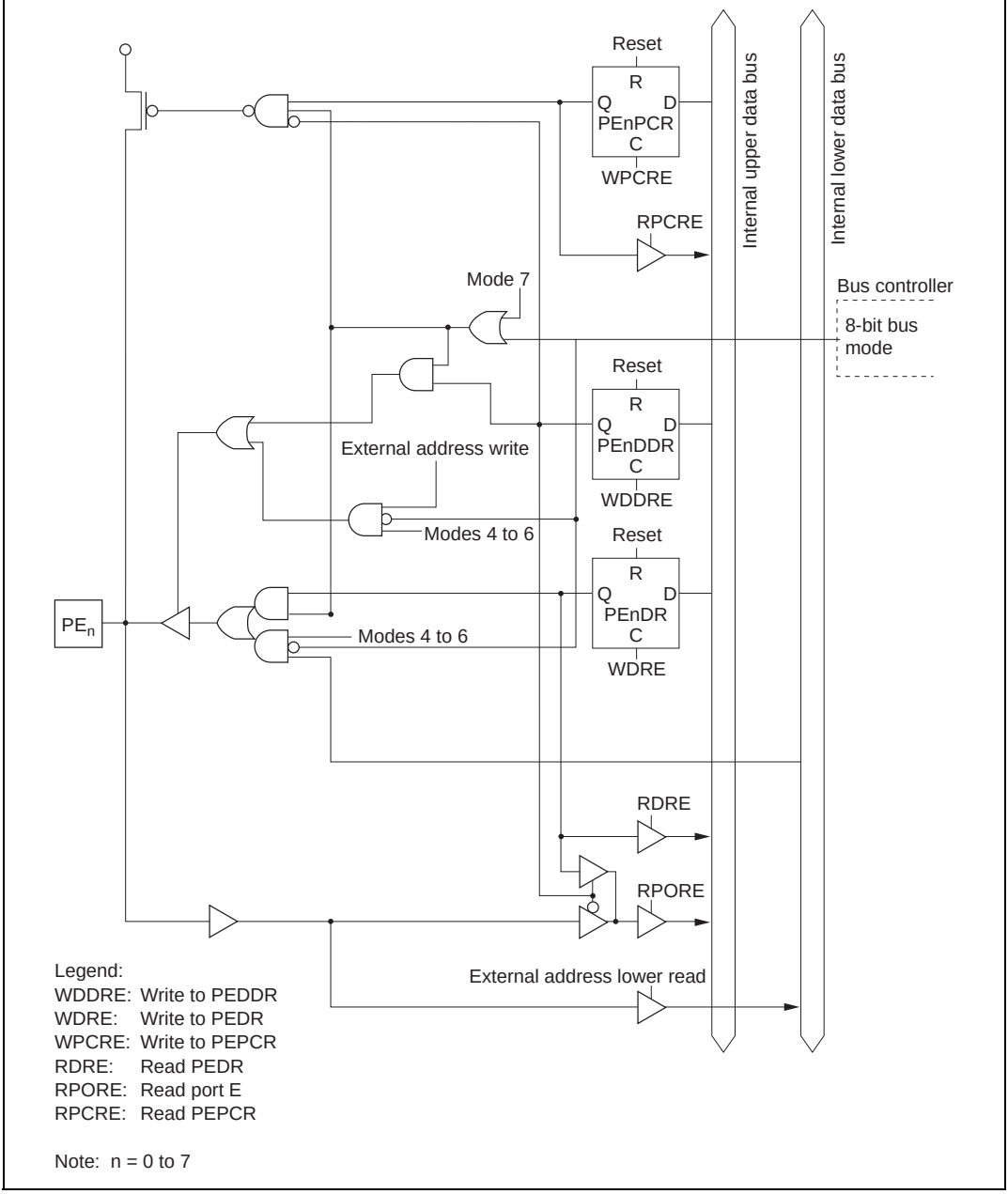


Figure C.14 Port E Block Diagram (Pins PE_0 to PE_7)

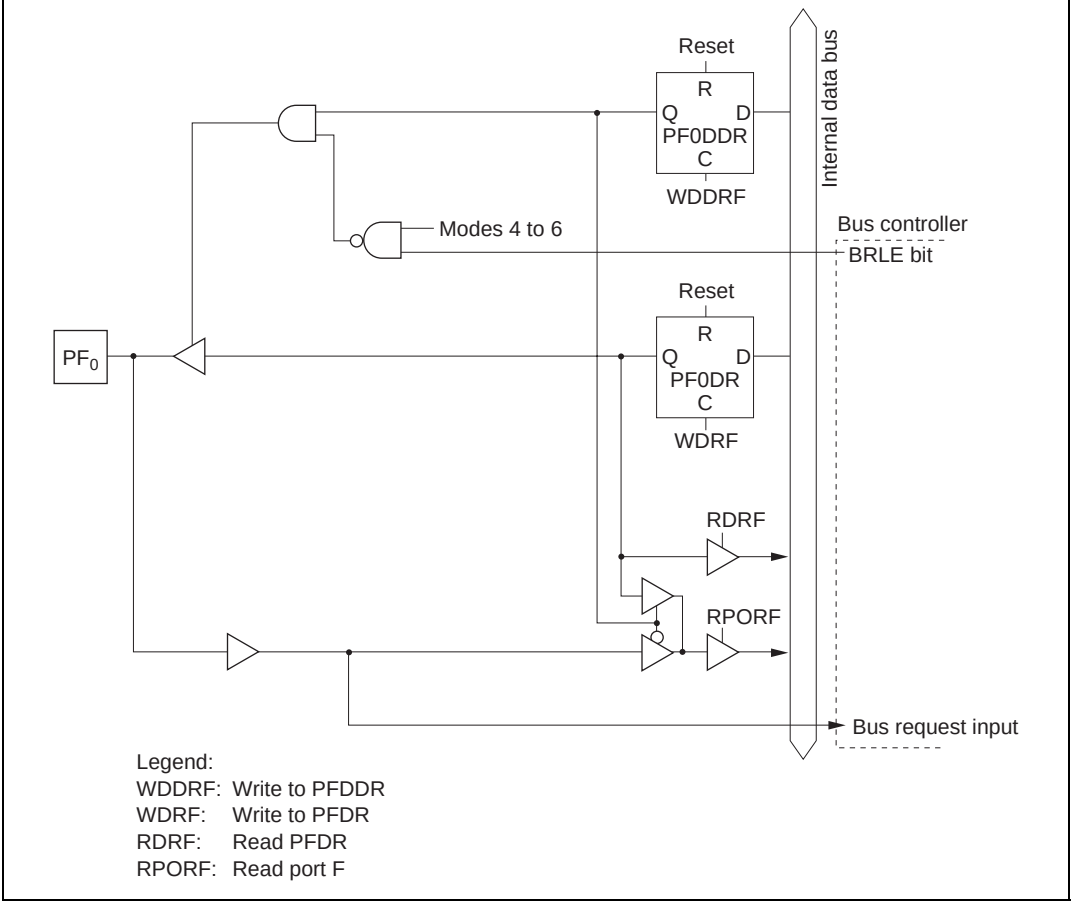


Figure C.15 (a) Port F Block Diagram (Pin PF_0)

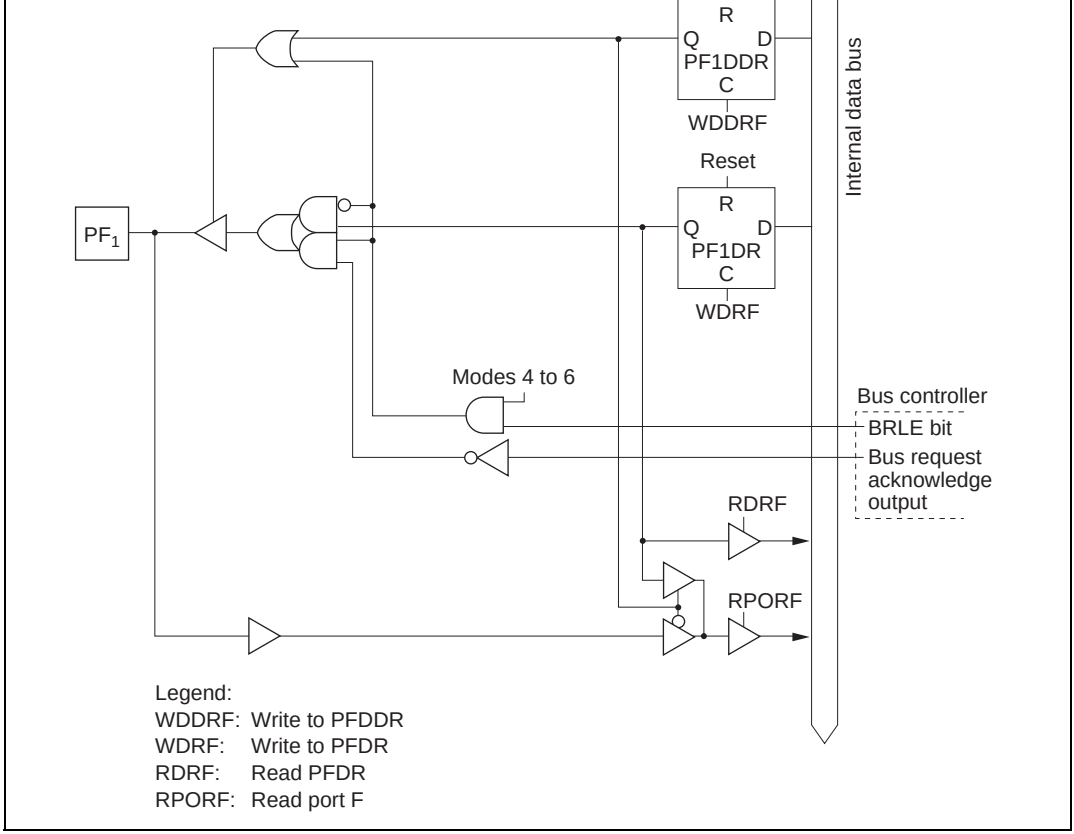


Figure C.15 (b) Port F Block Diagram (Pin PF_1)

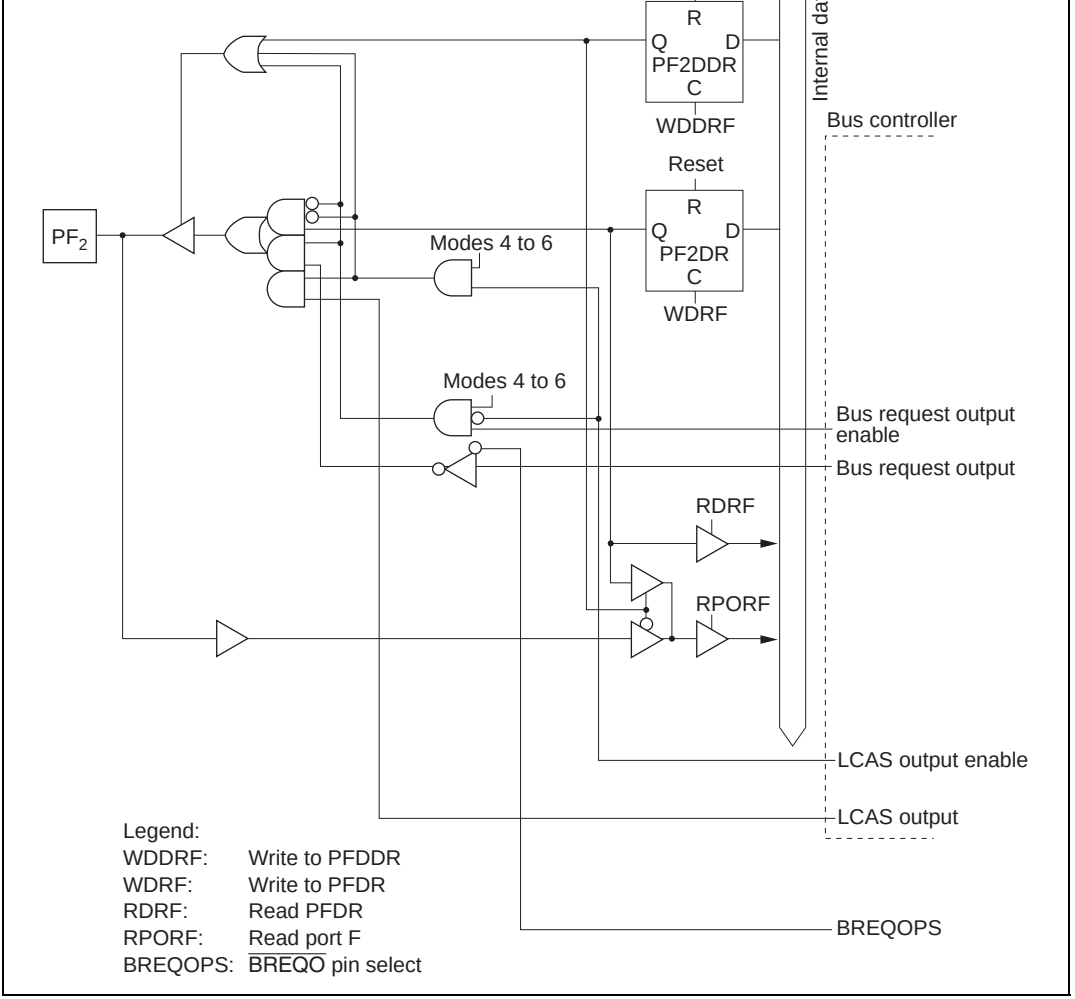


Figure C.15 (c) Port F Block Diagram (Pin PF_2)

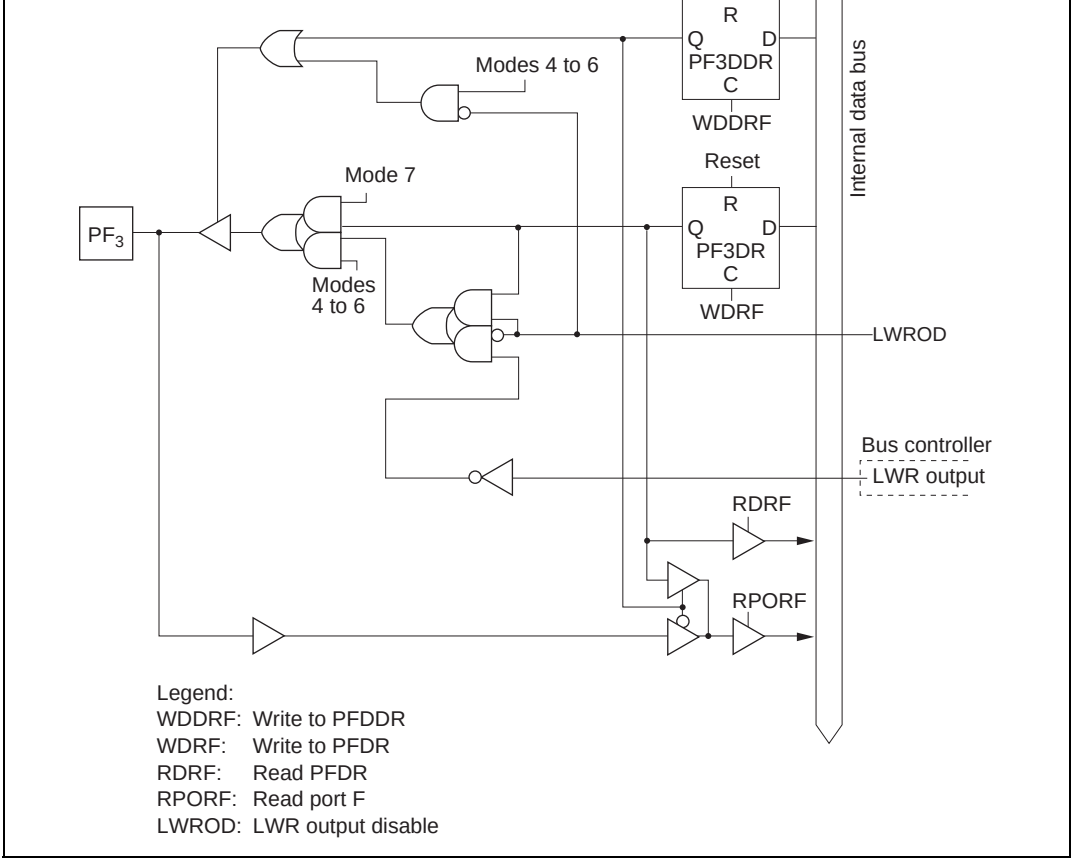


Figure C.15 (d) Port F Block Diagram (Pin PF_3)

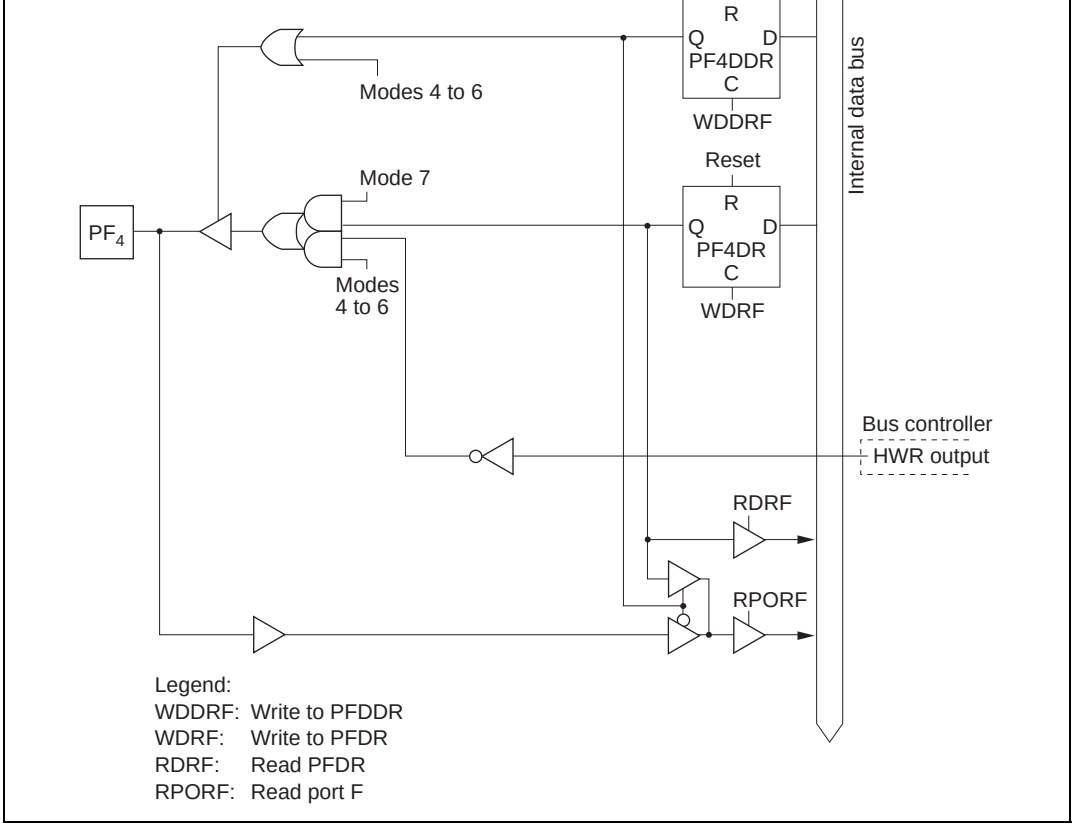


Figure C.15 (e) Port F Block Diagram (Pin PF_4)

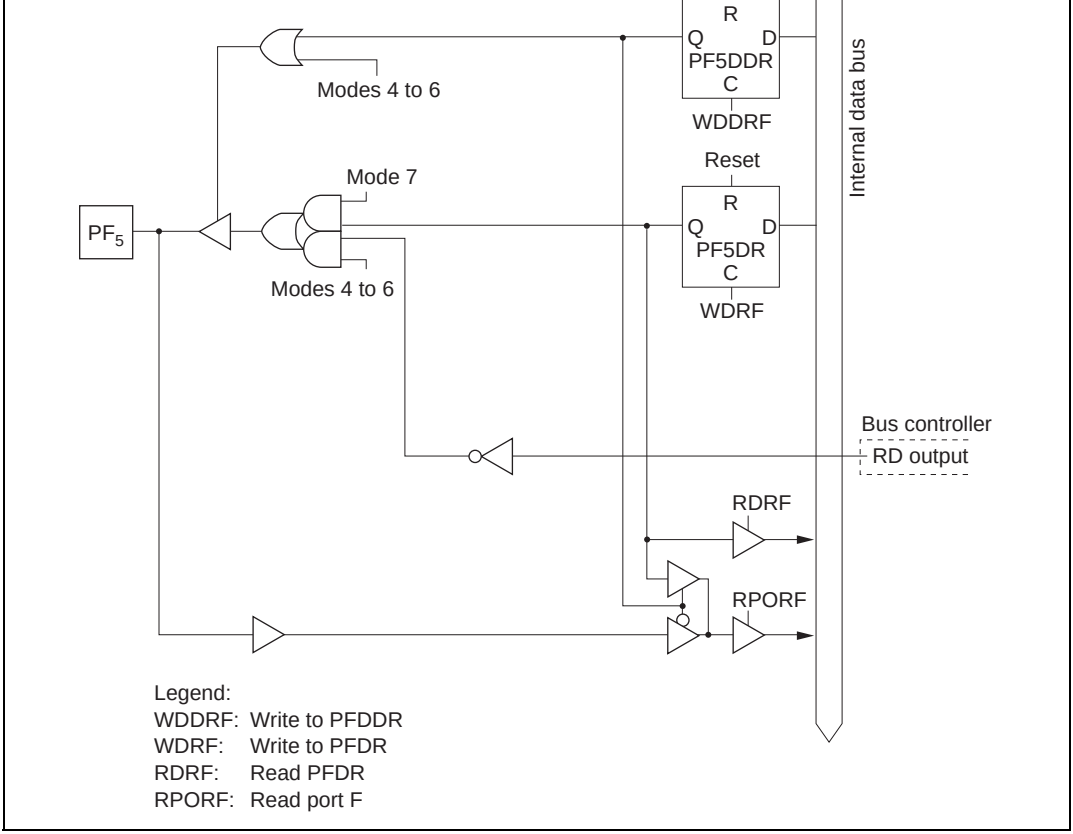


Figure C.15 (f) Port F Block Diagram (Pin PF_5)

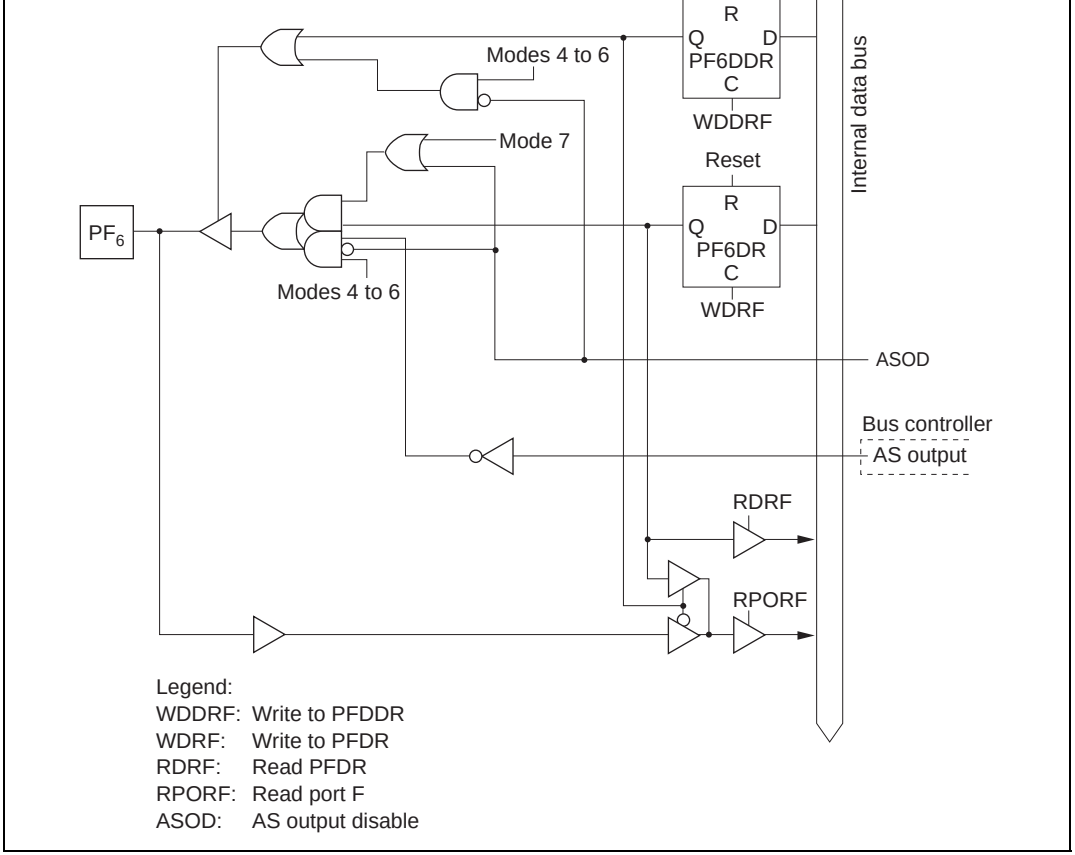


Figure C.15 (g) Port F Block Diagram (Pin PF_6)

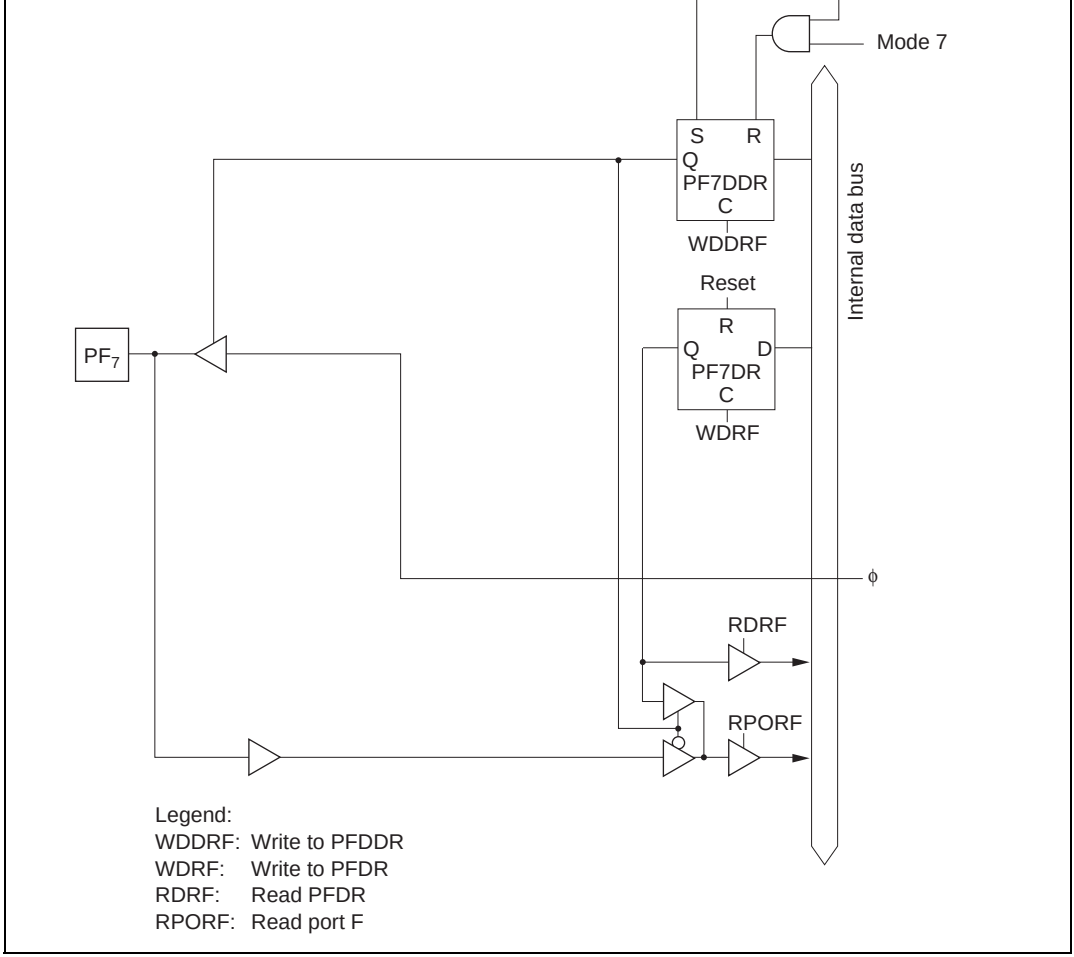


Figure C.15 (h) Port F Block Diagram (Pin PF_7)

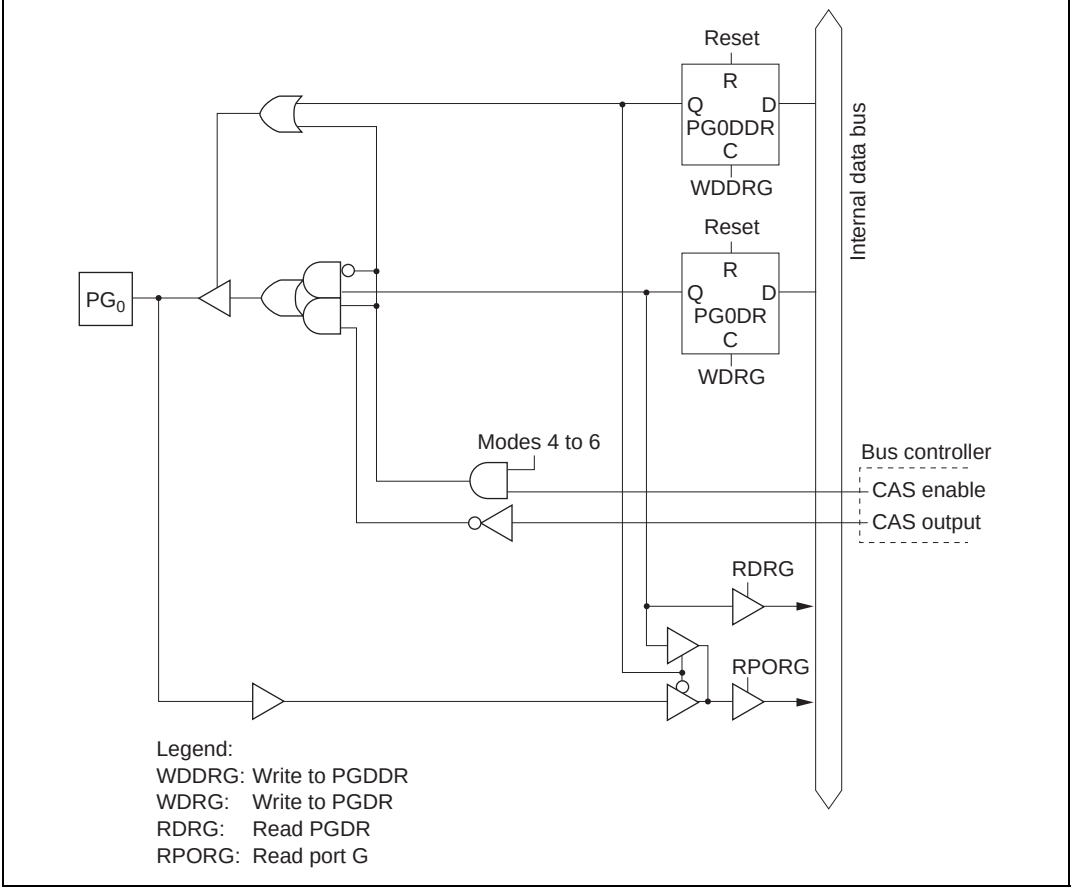


Figure C.16 (a) Port G Block Diagram (Pin PG_0)

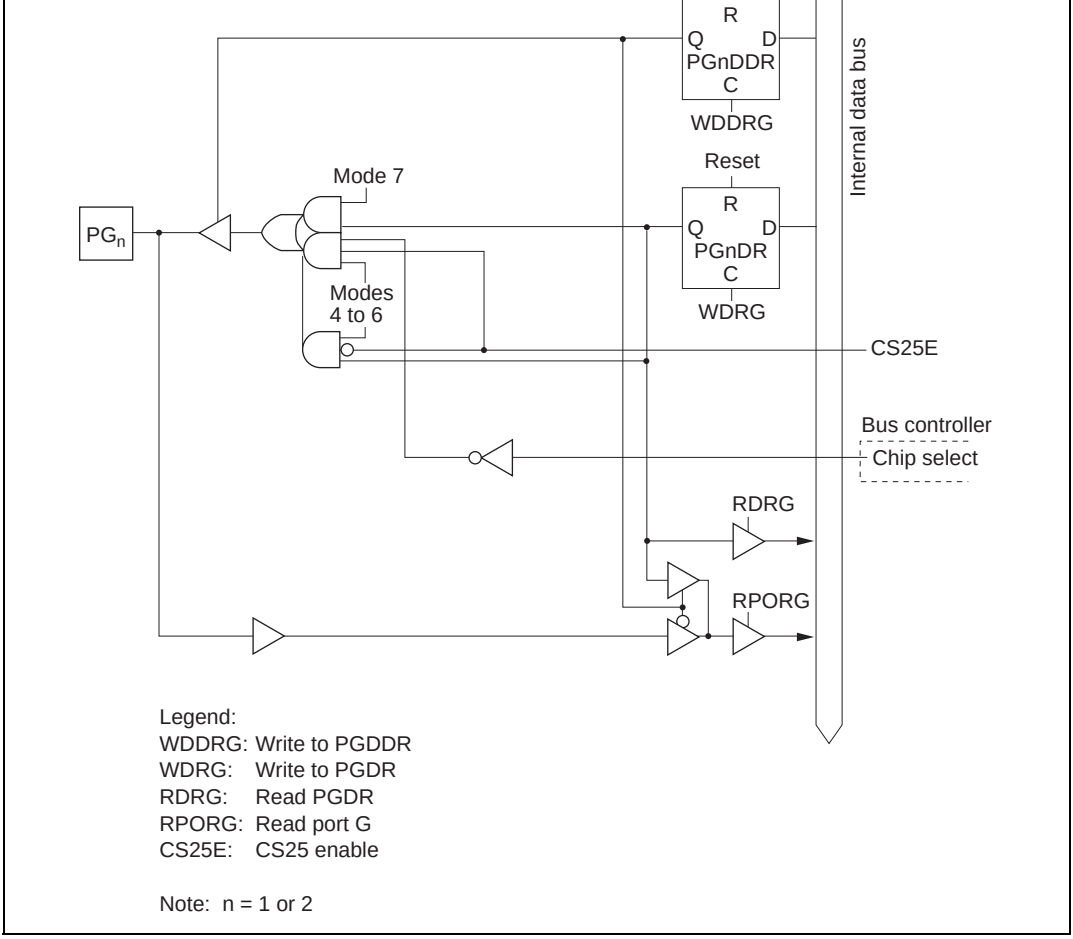


Figure C.16 (b) Port G Block Diagram (Pins PG_1 and PG_2)

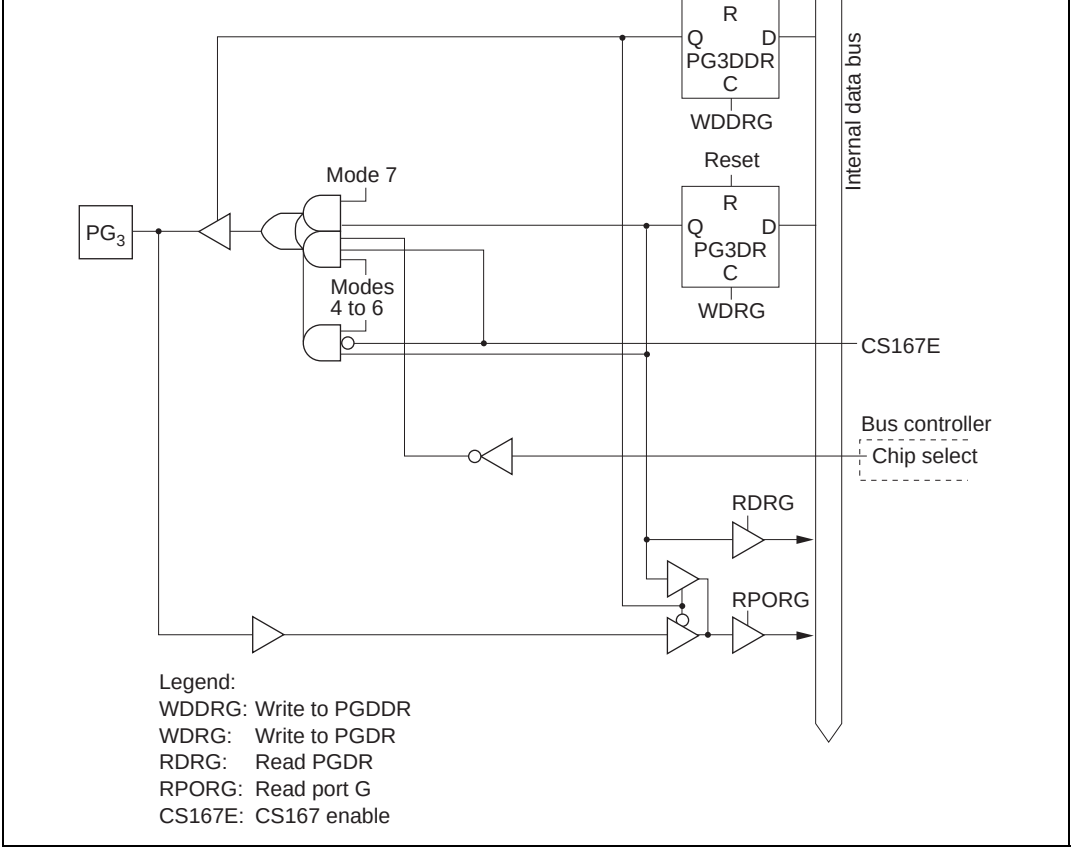


Figure C.16 (c) Port G Block Diagram (Pin PG_3)

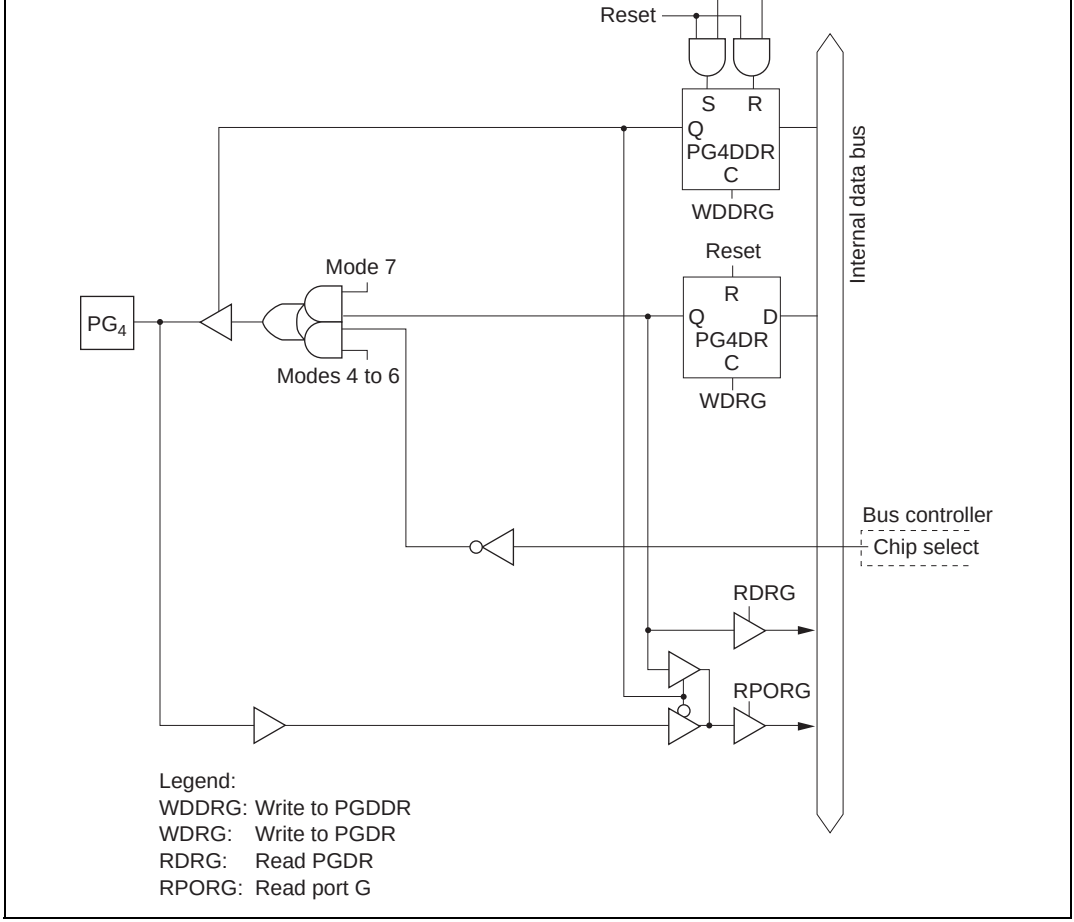


Figure C.16 (d) Port G Block Diagram (Pin PG_4)

D.1 Port States in Each Mode

Table D.1 I/O Port States in Each Processing State

Port Name Pin Name	MCU Operating Mode	Reset	Hardware Standby Mode	Software Standby Mode	Bus-Released State	Program Execution State Sleep Mode
Port 1	4 to 7	T	T	kept	kept	I/O port
Port 2	4 to 7	T	T	kept	kept	I/O port
Port 3	4 to 7	T	T	kept	kept	I/O port
$P4_7/DA_1$	4 to 7	T	T	[DAOE1 = 1] kept [DAOE1 = 0] T	kept	I/O port
$P4_6/DA_0$	4 to 7	T	T	[DAOE0 = 1] kept [DAOE0 = 0] T	kept	I/O port
$P4_5$ to $P4_0$	4 to 7	T	T	T	T	Input port
$P5_7/DA_3$	4 to 7	T	T	[DAOE3 = 1] kept [DAOE3 = 0] T	kept	I/O port
$P5_6/DA_2$	4 to 7	T	T	[DAOE2 = 1] kept [DAOE2 = 0] T	kept	I/O port
$P5_5$, $P5_4$	4 to 7	T	T	T	T	Input port

Pin Name	Mode	Reset	Mode	Standby Mode	Sleep Mode	
$P5_3$/$\overline{WAIT}$/ $\overline{BREQO}$	4 to 6	T	T	[BREQOE · BREQOPS + WAITE · WAITPS = 0] kept	[BREQOE · BREQOPS + WAITE · WAITPS = 0] kept	[BREQOE · BREQOPS + WAITE · WAITPS = 0] I/O port
				[BREQOE · BREQOPS = 1] kept	[BREQOE · BREQOPS = 1] $\overline{BREQO}$	[BREQOE · BREQOPS = 1] $\overline{BREQO}$
				[BREQOE · BREQOPS = 0] and [WAITE · WAITPS · $\overline{DDR}$ = 1] T	[BREQOE · BREQOPS = 0] and [WAITE · WAITPS · $\overline{DDR}$ = 1] T	[BREQOE · BREQOPS = 0] and [WAITE · WAITPS · $\overline{DDR}$ = 1] $\overline{WAIT}$
	7	T	T	kept	kept	I/O port
$P5_2$ to $P5_0$	4 to 7	T	T	kept	kept	I/O port
$P6_7$/$\overline{CS_7}$ $P6_6$/$\overline{CS_6}$	4 to 6	T	T	[CS167E = 0], [CS167E · $\overline{DDR}$ = 1] kept	[CS167E = 0] kept	[CS167E = 0] I/O port
				[CS167E · DDR · $\overline{OPE}$ = 1] T	[CS167E · $\overline{DDR}$ = 1] kept	[CS167E · $\overline{DDR}$ = 1] Input port
				[CS167E · DDR · OPE = 1] H	[CS167E · DDR = 1] T	[CS167E · DDR = 1] $\overline{CS_7}$, $\overline{CS_6}$
	7	T	T	kept	kept	I/O port
$P6_5$ to $P6_2$	4 to 7	T	T	kept	kept	I/O port
$P6_1$/$\overline{CS_5}$ $P6_0$/$\overline{CS_4}$	4 to 6	T	T	[CS25E · DDR · $\overline{OPE}$ = 1] T	[CS25E = 0] kept	[CS25E = 0] I/O port
				[CS25E · DDR · OPE = 1] H	[CS25E · $\overline{DDR}$ = 1] kept	[CS25E · $\overline{DDR}$ = 1] Input port
				[CS25E = 0], [CS25E · $\overline{DDR}$ = 1] kept	[CS25E · DDR = 1] T	[CS25E · DDR = 1] $\overline{CS_5}$, $\overline{CS_4}$
	7	T	T	kept	kept	I/O port
Port 7	4 to 7	T	T	kept	kept	I/O port

Pin Name	Mode	Reset	Mode	Standby Mode	Sleep	Sleep Mode
$P8_6$/$\overline{WAIT}$	4 to 6	T	T	[WAITE · $\overline{WAITPS}$ · $\overline{DDR}$ = 1] T [WAITE · $\overline{WAITPS}$ = 0] kept	[WAITE · $\overline{WAITPS}$ · $\overline{DDR}$ = 1] T [WAITE · $\overline{WAITPS}$ = 0] kept	[WAITE · $\overline{WAITPS}$ · $\overline{DDR}$ = 1] $\overline{WAIT}$ [WAITE · $\overline{WAITPS}$ = 0] I/O port
	7	T	T	kept	kept	I/O port
$P8_5$ to $P8_0$	4 to 7	T	T	kept	kept	I/O port
Port 9	4 to 7	T	T	kept	kept	I/O port
PA_7/A_{23} PA_6/A_{22} PA_5/A_{21}	4 to 6	T	T	[AnE = 0] kept [AnE · $\overline{DDR}$ = 1] T [AnE · DDR · $\overline{OPE}$ = 1] T [AnE · DDR · OPE = 1] kept	[AnE = 0] kept [AnE · $\overline{DDR}$ = 1] T [AnE · DDR = 1] T	[AnE = 0] I/O port [AnE · $\overline{DDR}$ = 1] Input port [AnE · DDR = 1] Address output
	7	T	T	kept	kept	I/O port
PA_4/A_{20}	4, 5	L	T	[$\overline{A20E}$ · DDR = 1] kept [A20E · $\overline{OPE}$ = 1] T [A20E · OPE = 1] kept	[$\overline{A20E}$ · DDR = 1] kept [A20E + $\overline{A20E}$ · $\overline{DDR}$ = 1] T	[$\overline{A20E}$ · DDR = 1] Output port [A20E + $\overline{A20E}$ · $\overline{DDR}$ = 1] Address output
	6	T	T	[A20E = 0], [A20E · $\overline{DDR}$ = 1] kept [A20E · DDR · $\overline{OPE}$ = 1] T [A20E · DDR · OPE = 1] kept	[A20E = 0], [A20E · $\overline{DDR}$ = 1] kept [A20E · DDR = 1] T	[A20E = 0] I/O port [A20E · $\overline{DDR}$ = 1] Input port [A20E · DDR = 1] Address output
	7	T	T	kept	kept	I/O port

Pin Name	Mode		Reset	Hardware Standby Mode	Software Standby Mode	Sleep Mode
PA_3/A_{19} PA_2/A_{18} PA_1/A_{17} PA_0/A_{16}	4, 5		L	T	[OPE = 0] T [OPE = 1] kept	T	Address output
	6		T	T	[DDR · OPE = 0] T [DDR · OPE = 1] kept	T	[DDR = 0] Input port [DDR = 1] Address output
	7		T	T	kept	kept	I/O port
Port B	4, 5		L	T	[OPE = 0] T [OPE = 1] kept	T	Address output
	6		T	T	[DDR · OPE = 0] T [DDR · OPE = 1] kept	T	[DDR = 0] Input port [DDR = 1] Address output
	7		T	T	kept	kept	I/O port
Port C	4, 5		L	T	[OPE = 0] T [OPE = 1] kept	T	Address output
	6		T	T	[DDR · OPE = 0] T [DDR · OPE = 1] kept	T	[DDR = 0] Input port [DDR = 1] Address output
	7		T	T	kept	kept	I/O port
Port D	4 to 6		T	T	T	T	Data bus
	7		T	T	kept	kept	I/O port
Port E	4 to 6	8-bit bus	T	T	kept	kept	I/O port
		16-bit bus	T	T	T	T	Data bus
	7			T	kept	kept	I/O port

Pin Name	Mode	Reset	Mode	Standby Mode	State	Sleep Mode
PF_7/ϕ	4 to 6	Clock output	T	[DDR = 0] Input port [DDR = 1] H	[DDR = 0] Input port [DDR = 1] Clock output	[DDR = 0] Input port [DDR = 1] Clock output
	7	T	T	[DDR = 0] Input port [DDR = 1] H	[DDR = 0] Input port [DDR = 1] Clock output	[DDR = 0] Input port [DDR = 1] Clock output
$PF_6/\overline{AS}$	4 to 6	H	T	[ASOD = 1] kept [$\overline{ASOD} \cdot \overline{OPE}$ = 1] T [$\overline{ASOD} \cdot$ OPE = 1] H	[ASOD = 1] kept [ASOD = 0] T	[ASOD = 1] I/O port [ASOD = 0] $\overline{AS}$
	7	T	T	kept	kept	I/O port
$PF_5/\overline{RD}$ $PF_4/\overline{HWR}$	4 to 6	H	T	[OPE = 0] T [OPE = 1] H	T	$\overline{RD}$, $\overline{HWR}$
	7	T	T	kept	kept	I/O port
$PF_3/\overline{LWR}$	4 to 6	H	T	[LWROD = 1] kept [$\overline{LWROD} \cdot \overline{OPE}$ = 1] T [$\overline{LWROD} \cdot$ OPE = 1] H	[LWROD = 1] kept [LWROD = 0] T	[LWROD = 1] I/O port [LWROD = 0] $\overline{LWR}$
	7	T	T	kept	kept	I/O port

Pin Name	Mode	Reset	Mode	Standby Mode	Sleep Mode	Sleep Mode
$PF_2/\overline{LCAS}/\overline{BREQO}$	4 to 6	T	T	[BREQOE · $\overline{BREQO}$ · $\overline{PS}$ + LCASE = 0] kept	[BREQOE · $\overline{BREQO}$ · $\overline{PS}$ + LCASE = 0] kept	[BREQOE · $\overline{BREQO}$ · $\overline{PS}$ + LCASE = 0] I/O port
				[BREQOE · $\overline{BREQO}$ · $\overline{PS}$ = 1] kept	[BREQOE · $\overline{BREQO}$ · $\overline{PS}$ = 1] $\overline{BREQO}$	[BREQOE · $\overline{BREQO}$ · $\overline{PS}$ = 1] $\overline{BREQO}$
				[LCASE = 1, OPE = 0] T	[LCASE = 1] T	[LCASE = 1] $\overline{LCAS}$
				[LCASE = 1, OPE = 1] H		
	7	T	T	kept	kept	I/O port
$PF_1/\overline{BACK}$	4 to 6	T	T	[BRLE=0] kept	L	[BRLE = 0] I/O port
				[BRLE=1] $\overline{BACK}$		[BRLE = 1] $\overline{BACK}$
	7	T	T	kept	kept	I/O port
$PF_0/\overline{BREQ}$	4 to 6	T	T	[BRLE=0] kept	T	[BRLE = 0] I/O port
				[BRLE=1] T		[BRLE = 1] $\overline{BREQ}$
	7	T	T	kept	kept	I/O port
$PG_4/\overline{CS_0}$	4, 5	H	T	[DDR · OPE = 0] T	T	[DDR = 0] Input port
	6	T		[DDR · OPE = 1] H		[DDR = 1] $\overline{CS_0}$
	7	T	T	kept	kept	I/O port

Pin Name	Mode	Reset	Mode	Standby Mode	Sleep State	Sleep Mode
PG$_3$/$\overline{CS_1}$	4 to 6	T	T	[CS167E = 0] kept	[CS167E = 0] kept	[CS167E = 0] I/O port
				[CS167E · $\overline{DDR}$ = 1] T	[CS167E = 1] T	[CS167E · $\overline{DDR}$ = 1] Input port
				[CS167E · DDR · $\overline{OPE}$ = 1] T		[CS167E · DDR = 1] $\overline{CS_1}$
				[CS167E · DDR · OPE = 1] H		
	7	T	T	kept	kept	I/O port
PG$_2$/$\overline{CS_2}$ PG$_1$/$\overline{CS_3}$	4 to 6	T	T	[CS25E = 0] kept	[CS25E = 0] kept	[CS25E = 0] I/O port
				[CS25E · $\overline{DDR}$ = 1] T	[CS25E = 1] T	[CS25E · $\overline{DDR}$ = 1] Input port
				[CS25E · DDR · $\overline{OPE}$ = 1] T		[CS25E · DDR = 1] $\overline{CS_2}$ to $\overline{CS_3}$
				[CS25E · DDR · OPE = 1] H		
	7	T	T	kept	kept	I/O port
PG0/$\overline{CAS}$	4 to 6	T	T	[DRAME = 0] kept	T	[DRAME = 0] Input port
				[DRAME · $\overline{OPE}$ = 1] T		[DRAME = 1] $\overline{CAS}$
				[DRAME · OPE = 1] $\overline{CAS}$		
	7	T	T	kept	kept	I/O port
$\overline{WDTOVF}$	4 to 7	H	H	H	H	H*

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
DRAME: DRAM space setting

A20E: Address 20 enable
BREQOPS: BREQO pin select
ASOD: AS output disable
WAITPS: WAIT pin select
CS167E: CS167 enable
CS25E: CS25 enable
LWROD: LWR output disable

Note: * A low level is output if a WDT overflow occurs while WT/IT is set to 1.

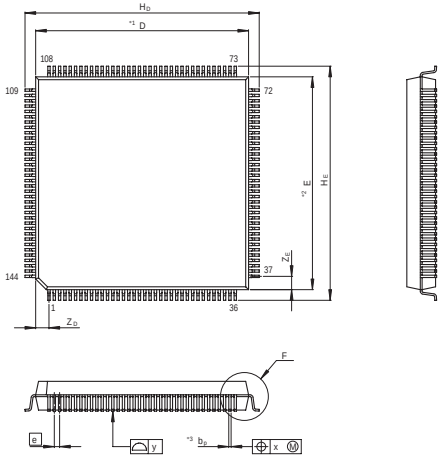
Table E.1 H8S/2339 Group Product Lineup

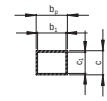
Product Type		Model	Marking	Package (Package Code)
H8S/2339	F-ZTAT version	HD64F2339	HD64F2339VFC	144-pin QFP (FP-144G)
		HD64F2339E*	HD64F2339EVFC	144-pin QFP (FP-144G)
H8S/2338	Mask ROM version	HD6432338	HD6432338FC	144-pin QFP (FP-144G)
	F-ZTAT version	HD64F2338	HD64F2338VFC	144-pin QFP (FP-144G)
H8S/2337	Mask ROM version	HD6432337	HD6432337FC	144-pin QFP (FP-144G)
H8S/2332	ROMless version	HD6412332	HD6412332VFC	144-pin QFP (FP-144G)

Note: * The on-chip debug function can be used with the E10-A emulator (E10-A compatible version). However, some function modules and pin functions are unavailable when the on-chip debug function is in use. Refer to figure 1.3.

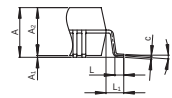
The package dimension that is shown in the Renesas Semiconductor Package Data Book has priority.

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-QFP144-20x20-0.50	PRQP0144KA-A	FP-144G/FP-144GV	2.4g





Terminal cross section



Detail F

NOTE)
1. DIMENSIONS"*1"AND"*2" DO NOT INCLUDE MOLD FLASH
2. DIMENSION"*3"DOES NOT INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	—	20	—
E	—	20	—
A_2	—	2.70	—
H_D	21.8	22.0	22.2
H_E	21.8	22.0	22.2
A	—	—	3.05
A_1	0.00	0.10	0.25
b_p	0.17	0.22	0.27
b_1	—	0.20	—
c	0.12	0.17	0.22
c_1	—	0.15	—
θ	0°	—	8°
e	—	0.5	—
x	—	—	0.10
y	—	—	0.10
Z_D	—	1.25	—
Z_E	—	1.25	—
L	0.4	0.5	0.6
L_1	—	1.0	—

Figure F.1 FP-144G Package Dimensions

Renesas 16-Bit Single-Chip Microcomputer
Hardware Manual
H8S/2339 Group

Publication Date: 1st Edition, March 1999
Rev.4.00, September 7, 2007
Published by: Sales Strategic Planning Div.
Renesas Technology Corp.
Edited by: Customer Support Department
Global Strategic Communication Div.
Renesas Solutions Corp.

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan

RENESAS SALES OFFICES

http://www.renesas.com

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

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

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

Renesas Technology (Shanghai) Co., Ltd.
Unit 204, 205, AZIACenter, No.1233 Lujiazui Ring Rd, Pudong District, Shanghai, China 200120
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7898

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

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

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

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

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

Colophon 6.0

H8S/2339 Group
Hardware Manual

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

REJ09B0245-0400

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

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

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

Other Similar products are found below :

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