REJ09B0140-0700

Everywhere you imagine. RENESAS

The revision list can be viewed directly by clicking the title page. The revision list summarizes the locations of revisions and additions. Details should always be checked by referring to the relevant text.

16

H8S/2215 Group

Hardware Manual

Renesas 16-Bit Single-Chip Microcomputer
H8S Family/H8S/2200 Series

H8S/2215	HD64F2215
	HD64F2215U
	HD6432215B
	HD6432215C
H8S/2215R	HD64F2215R
	HD64F2215RU
H8S/2215T	HD64F2215T
	HD64F2215TU

Hardware Manual

Rev.7.00
Revision Date: Mar. 27, 2007

Renesas Technology
www.renesas.com

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate 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.

General Precautions in the Handling of MPU/MCU Products

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.

Configuration of this Manual

This manual comprises the following items:

1. General Precautions on the Handling of Products
2. Configuration of This Manual
3. Preface
4. Main Revisions in This Edition

   The history of revisions is a summary of sections that have been revised and sections that have been added to earlier versions. This does not include all of the revised contents. For details, confirm by referring to the main description of this manual.
5. Contents
6. Overview
7. Table of Contents
8. Summary
9. Description of Functional Modules
   - CPU and System-Control Modules
   - On-Chip Peripheral Modules

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

   i) Features
   ii) I/O pins
   iii) Description of Registers
   iv) Description of Operation
   v) Usage: Points for Caution

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

10. List of Registers
11. Electrical Characteristics
12. Appendix
    - Product-type codes and external dimensions

Preface

This LSI is a high-performance microcomputer (MCU) made up of the H8S/2000 CPU with Renesas' original architecture as its core, and the peripheral functions required to configure a system.

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

This LSI is equipped with ROM, RAM, a direct memory access controller (DMAC), a bus master for a data transfer controller (DTC), a 16-bit timer pulse unit (TPU), an 8-bit timer (TMR), a watchdog timer (WDT), a universal serial bus (USB), two types of serial communication interfaces (SCIs), an A/D converter, a D/A converter, and I/O ports as on-chip peripheral modules for system configuration.

A single-power flash memory (F-ZTAT™*) version and masked ROM version are available for this LSI's ROM. The F-ZTAT version 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.

This manual describes this LSI's hardware.

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

Target Users: This manual was written for users who will be using the H8S/2215 Group in the design of application systems. Target users are expected to understand the fundamentals of electrical circuits, logical circuits, and microcomputers.

Objective: This manual was written to explain the hardware functions and electrical characteristics of the H8S/2215 Group to the target users.
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

  Read the index that is the final part of the manual to find the page number of the entry on the register. The addresses, bits, and initial values of the registers are summarized in appendix A, I/O Port States in Each Processing State.

  Examples: Register name: The following notation is used for cases when the same or a similar function, e.g. 16-bit timer pulse unit or serial communication, is implemented on more than one channel: XXX_N (XXX is the register name and N is the channel number)

  Bit order: The MSB is on the left and the LSB is on the right.

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/

H8S/2215 Group manuals:

Document Title	Document No.
H8S/2215 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, Optimizing Linkage Editor Compiler Package V.6.01 User's Manual	REJ10B0058
H8S, H8/300 Series Simulator/Debugger (for Windows) User's Manual	ADE-702-037
H8S, H8/300 Series High-performance Embedded Workshop, High-performance Debugging Interface Tutorial	ADE-702-231
High-performance Embedded Workshop (for Windows 95/98 and Windows NT 4.0) User's Manual	ADE-702-201

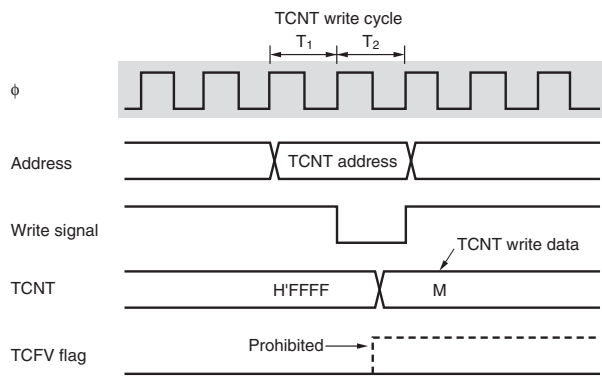
Main Revisions for This Edition

<table>
<tr><th>Item</th><th>Page</th><th>Revision (See Manual for Details)</th></tr>
<tr><td>All</td><td>—</td><td>H8S/2215T (HD64F2215R, HD64F2215T) added
HD6432215A deleted

Description and notes amended
(Before) RESERVE → (After) NC
NC (No Connection): This pin should not be connected; it should be left open.
$\overline{\text{EMLE}}$ pin in H8S/2215R and H8S/2215T.</td></tr>
<tr><td>1.1 Overview</td><td>1</td><td>Description and note * amended
• Various peripheral functions
— User debug interface (H-UDI)*
Note: * Available only in H8S/2215R and H8S/2215T.
• On-chip memory
<table>
<tr><th>ROM</th><th>Part No.</th><th>ROM</th><th>RAM</th><th>Remarks</th></tr>
<tr><td>F-ZTAT Version</td><td>HD64F2215</td><td>256 kbytes</td><td>16 kbytes</td><td>SCI boot version</td></tr>
<tr><td></td><td>HD64F2215U</td><td>256 kbytes</td><td>16 kbytes</td><td>USB boot version</td></tr>
<tr><td></td><td>HD64F2215T</td><td>256 kbytes</td><td>20 kbytes</td><td>SCI boot version</td></tr>
<tr><td></td><td>HD64F2215TU</td><td>256 kbytes</td><td>20 kbytes</td><td>USB boot version</td></tr>
<tr><td></td><td>HD64F2215R</td><td>256 kbytes</td><td>20 kbytes</td><td>SCI boot version</td></tr>
<tr><td></td><td>HD64F2215RU</td><td>256 kbytes</td><td>20 kbytes</td><td>USB boot version</td></tr>
</table></td></tr>
<tr><td>1.2 Internal Block Diagram
Figure 1.1 Internal Block Diagram</td><td>3</td><td>Note 2 amended
Note: 2. The H-UDI function and $\overline{\text{EMLE}}$ pin are only provided in H8S/2215R and H8S/2215T.</td></tr>
<tr><td>1.3 Pin Arrangement
Figure 1.2 Pin Arrangement (TFP-120, TFP-120V)
Figure 1.3 Pin Arrangement (BP-112, BP-112V)</td><td>4, 5</td><td>Note 2 amended
Note: 2. NC in H8S/2215. $\overline{\text{EMLE}}$ pin in H8S/2215R and H8S/2215T.</td></tr>
</table>

Item	Page	Revision (See Manual for Details)
1.4 Pin Function in Each Operating Mode	9	Table amended Pin No. (TFP-120, TFP-120V / BP-112, BP-112V) — Pin Name (Mode 7[*1] / PROM Mode) 93 D8 — P35/SCK1/IRQ5 — NC 94 B9 — P36 (PUPD+)[*3] — NC 95 — — RESERVE — NC
	10	Note 3 added Note: 3. PUPD+ pin in H8S/2215R and H8S/2215T.
1.5 Pin Functions	11	Function description amended Operating mode control (Before) Set the operating mode. Inputs at these pins cannot be modified during operation. → (After) Except for mode changing, be sure to fix the levels of the mode pins (MD2 to MD0) by pulling them down or pulling them up until the power turns off.
	18	USB Pins to be connected to the transceiver (ISP1104) manufactured by NXP.
	21	Notes 1 and 2 amended Notes: 1. Available only in H8S/2215R and H8S/2215T. (NC in H8S/2215.) 2. Available only in H8S/2215. (EMLE pin in H8S/2215R and H8S/2215T.)
2.6. Instruction Set Table 2.1 Instruction Classification	39	Table 2.1 amended LDM[*5], SDM[*5]
		Note 5 added Note: 5. The ER7 register functions as a stack pointer for the LDM and STM instructions, so it cannot be for saving (STM) or restoring (LDM) data.
3.1 Operating Mode Selection Table 3.1 MCU Operating Mode Selection	63	Note * amended Note: * The following applies to the use of mode 7. (1) H8S/2215 ... (2) H8S/2215R or H8S/2215T Development work using E6000 emulator: ...
3.3.4 Mode 7 Table 3.2 USB Support in Mode 7	67	Table 3.2 amended Development Tool — H8S/2215 — H8S/2215R or H8S/2215T E6000 — × — × E10A-USB — —* — ○

Item	Page	Revision (See Manual for Details)
3.3.5 Pin Functions Table 3.3 Pin Functions in Each Operating Mode	68	Note 1 amended Note: 1. The following applies to the use of mode 7. (1) H8S/2215 ... (2) H8S/2215R or H8S/2215T Development work using E6000 emulator: ...
3.4 Memory Map in Each Operating Mode Figure 3.4 Memory Map in Each Operating Mode for HD64F2215R, HD64F2215RU, HD46F2215T, and HD64F2215TU	72	Figure 3.4 title amended
6.9.1 Notes on Bus Release	144	Section 6.9.1 added
8.2.5 DTC Transfer Count Register A (CRA)	208	Description amended ... as an 8-bit transfer counter (1 to 256). In block transfer mode, CRAH holds the block size while CRAL functions as an 8-bit block size counter (1 to 256). CRAL is decremented by 1 ...
8.2.8 DTC Vector Register (DTVECR)	209	Note * added R/W* Note: * Only 1 may be written to the SWDTE bit.
8.4 Location of Register Information and DTC Vector Table Table 8.2 Interrupt Sources, DTC Vector Addresses, and Corresponding DTCE	213	Table 8.2 amended DTC vector address of TXI1 in SIC channel 1 (Before) H'04AB → (After) H'04AC
9.2.5 Overview Table 9.10 P36 Pin Function	238	Table 9.10 amended P36DDR: 0 — Pin function: P36 input P36DDR: 1 — Pin function: P36 output (USB D+ pull-up control output in HD64F2215U, HD64F2215RU, HD64F2215TU)

Item	Page	Revision (See Manual for Details)
9.2.5 Overview Table 9.11 P35 Pin Function Table 9.12 P34 Pin Function	239	Note 2 amended Note: 2. Note on Development Using the E6000 Emulator The H8S/2215 Group does not have … characteristics of the H8S/2215 Group. If it is necessary to …
9.13 Handling of Unused Pins	279	Section 9.13 added
10.3.1 Timer Control Register (TCR)	287	CKEG1 and CKEG0 description amended These bits select the input clock edge. When the internal clock is counted using both edges, the input clock frequency is halved (e.g., ϕ/4 both edges = ϕ/2 rising edge). ... Internal clock edge selection is valid when the input clock is ϕ/4 or slower. If ϕ/1 is selected for the input clock, this setting is ignored and count at falling edge of ϕ is selected.
10.8 Usage Notes Figure 10.53 Contention between TCNT Write and Overflow	346	Figure 10.53 amended 
		Description added Interrupts and Module Stop Mode: ... before entering module stop mode. Module Stop Mode Setting: TPU operation can be disabled or enabled using the module stop control register. The initial setting is for TPU operation to be halted. Register access is enabled by clearing module stop mode. For details, refer to section 22, Power-Down Modes.
11.8.7 Module Stop Mode Setting	366	Section 11.8.7 added
12.5.6 OVF Flag Clearing in Interval Timer Mode	377	Section 12.5.6 added

Item	Page	Revision (See Manual for Details)
13.1 Features	380	Description amended • Average transfer rate generator (SCI_0): In H8S/2215 ... In H8S/2215R and H8S/2215T 921.569 kbps, 720 kbps, ...
13.1.1 Block Diagram Figure 13.2 Block Diagram of SCI_0 (H8S/2215R and H8S/2215T)	382	Figure 13.2 title amended
13.3 Register Descriptions	384, 385	Description amended • Serial extended mode register A_0 (SEMRA_0)(only for channel 0 in H8S/2215R and H8S/2215T) • Serial extended mode register B_0 (SEMRB_0)(only for channel 0 in H8S/2215R and H8S/2215T)
13.3.7 Serial Status Register (SSR)	396	• Normal Serial Communication Interface Mode (When SMIF in SCMR is 0) Note *2 added to TEND description DTC*2
	399	• Smart Card Interface Mode (When SMIF in SCMR is 1) Note *2 added to TEND description DTC*2
13.3.10 Serial Extended Mode Register A_0 (SEMRA_0)(Only for Channel 0 in H8S/2215R and H8S/2215T)	409	Section 13.3.10 title amended
13.3.11 Serial Extended Mode Register B_0 (SEMRB_0)(Only for Channel 0 in H8S/2215R and H8S/2215T)	411	Section 13.3.11 title amended

Item	Page	Revision (See Manual for Details)
13.3.12 Bit Rate Register (BRR) Table 13.3 BRR Settings for Various Bit Rates (Asynchronous Mode)	414	Table 13.3 amended Operating Frequency ϕ (MHz) Bit Rate (bit/s) | 2 (n, N, Error (%)) | 2.097152 (n, N, Error (%)) | 2.4576 (n, N, Error (%)) | 3 (n, N, Error (%)) 38400 — — — — — — 0 1 0.00 — — —
	416	ϕ = 9.8304 to 16 description added to table 13.3
	417	Table 13.3 amended Operating Frequency ϕ (MHz) Bit Rate (bps) | 17.2032 (n, N, Error (%)) | 18 (n, N, Error (%)) | 19.6608 (n, N, Error (%)) | 20 (n, N, Error (%)) 31250 0 16 1.20 0 17 0.00 0 19 –1.17 0 19 0.00 38400 0 13 0.00 0 14 –2.34 0 15 0.00 0 15 1.73
13.10.7 Module Stop Mode Setting	468	Section 13.10.7 added
14.1 Features	469	Description amended • Boundary scan function cannot be performed on the following pins. ... — Boundary scan signals: TCK, TDI, TDO, TMS, $\overline{\text{TRST}}$ — E10A signal ($\overline{\text{EMLE}}$)
14.3.2 IDCODE Register (IDCODE)	474	Description amended … The HD64F2215 and H8S/2215U output fixed code H'0002200F, HD6432215A output fixed code H'0003200F, HD6432215B output fixed code H'001B200F, HD6432215C output fixed code H'001C200F, HD64F2215R and HD64F2215RU output fixed code H'08030447, and HD64F2215T and HD64F2215TU output fixed code H'08031447, respectively, from the TDO. …
Table 14.3 IDCODE Register Configuration		Table 14.3 amended Bits | 31 to 28 | 27 to 12 | 11 to 1 | 0 HD64F2215RU code | 0000 | 1000 0000 0011 0000 | 0100 0100 011 | 1 HD64F2215T and HD64F2215TU code | 0000 | 1000 0000 0011 0001 | 0100 0100 011 | 1 Contents | Version (4 bits) | Part No. (16 bits) | Product No. (11 bits) | Fixed code (1 bit)
14.5 Usage Notes	485	Description added 7. If EXTEST, CLAMP, or ... to the designated mode. 8. When using the boundary scan function, leave the $\overline{\text{EMLE}}$ pin open.

Item	Page	Revision (See Manual for Details)
15.1 Features	488	Description amended • Maximum Configuration, InterfaceNumber, and AlternateSetting Configuration specifications of this LSI H8S/2215: Configuration 1 ... H8S/2215R and H8S/2215T: Configuration 1 ... • 23 kinds of interrupts (H8S/2215) • 25 kinds of interrupts (H8S/2215R and H8S/2215T)
Figure 15.1 Block Diagram of USB	489	Note * amended Note: * Available only in H8S/2215R and H8S/2215T.
15.2 Input/Output Pins Table 15.1 Pin Configuration	490	Table 15.1 amended External transceiver connection signals Signals used to connect with the transceiver (ISP1104) manufactured by NXP.
15.3.1 USB Endpoint Information Registers 00_0 to 22_4 (UEPIR00_0 to UEPIR22_4)	494	• UEPIRnn_0 **Bit** / **Bit Name** / **Initial Value** / **R/W** / **Description** 1 / D33 / — / R/W / H8S/2215 0 / D32 / — / R/W / Interface number to which endpoint belongs (2-bit configuration, settable values: 0 to 2) 00: Control transfer 00 to 10: Other than Control transfer H8S/2215R and H8S/2215T Interface number to which endpoint belongs (2-bit configuration, settable values: 0 to 3) 00: Control transfer 00 to 11: Other than Control transfer
15.3.2 USB Control Register (UCTLR)	501	UCKS3 to UCKS0 description amended ... 0010 (H8S/2215R and H8S/2215T): Uses a clock (48 MHz) generated by doubling the 24-MHz system clock by the PLL circuit. ... ... 0110 (H8S/2215R and H8S/2215T): Uses a clock (48 MHz) generated by doubling the 24-MHz system clock by the PLL circuit. ...
	502	... 1001 (H8S/2215R and H8S/2215T): Uses the clock supplied by the 48-MHz external clock (EXTAL48 pin input) directly. ... ... 1101 (H8S/2215R and H8S/2215T): Uses the USB operating clock (48 MHz) directly. ...

Item	Page	Revision (See Manual for Details)
15.3.10 USB Endpoint Stall Register 1 (UESTL1)	511	SCME description amended Bit / Bit Name / Initial Value / R/W / Description 7 / SCME / 0 / R/W / Reserved The write value should always be 0. 6 to 3 / — / All 0 / R / Reserved These bits are always read as 0 and cannot be modified.
15.3.28 USB Interrupt Flag Register 1 (UIFR1) (Only in H8S/2215R and H8S/2215T)	520	Section 15.3.28 title amended
15.3.30 USB Interrupt Flag Register 2 (UIFR2) (Only in H8S/2215R and H8S/2215T)	523	Section 15.3.30 title amended
15.3.34 USB Interrupt Enable Register 1 (UIER1) (Only in H8S/2215R and H8S/2215T)	528	Section 15.3.34 title amended
15.3.36 USB Interrupt Enable Register 2 (UIER2) (Only in H8S/2215R and H8S/2215T)	529	Section 15.3.36 title amended
15.3.40 USB Interrupt Select Register 1 (UISR1) (Only in H8S/2215R and H8S/2215T)	531	Section 15.3.40 title amended
15.3.42 USB Interrupt Select Register 2 (UISR2) (Only in H8S/2215R and H8S/2215T)	532	Section 15.3.42 title amended
15.4 Interrupt Sources Table 15.5 SCI Interrupt Sources	543	Notes 7 and 8 amended Notes: 7. Available only in H8S/2215R and H8S/2215T. "—" in H8S/2215. 8. Available only in H8S/2215R and H8S/2215T. Reserved in H8S/2215.

Item	Page	Revision (See Manual for Details)
15.5.11 Stall Operations	571	(2) Forcible Stall by Firmware Description amended … handshake to the host (1-3 in figure 15.23). Once an internal … be cleared by the firmware, and also for the …
Figure 15.23 Forcible Stall by Firmware	572	Figure in (1-4) deleted from figure 15.23
15.7 Endpoint Configuration Example Table 15.8 Bit Name Modification List	582	Notes 1 and 2 amended Notes: 1. H'01 in H8S/2215. H'09 in H8S/2215R and H8S/2215T. 2. Available only in H8S/2215R and H8S/2215T. "—" in H8S/2215.
15.8 USB External Circuit Example Figure 15.30 USB External Circuit in Bus-Powered Mode (When On-Chip Transceiver Is Used) Figure 15.31 USB External Circuit in Self-Powered Mode (When On-Chip Transceiver Is Used)	585, 586	Note 3 amended Note: 3. In 64F2215, HD64F2215R, HD64F2215T, HD6432215A, HD6432215B, HD6432215C, Pxx should be assigned to an output port as the D+ pull-up control pin. In 64F2215U, HD64F2215RU, and HD64F2215TU, in which on-chip ROM can be programmed by using the USB, ...
Figure 15.32 USB External Circuit in Bus-Powered Mode (When External Transceiver Is Used)	587	Figure 15.32 and note amended External transceiver (ISP1104 manufactured by NXP) Note: 3. In 64F2215, HD64F2215R, HD64F2215T, HD6432215A, HD6432215B, HD6432215C, Pxx should be assigned to an output port as the D+ pull-up control pin. In 64F2215U, HD64F2215RU, and HD64F2215TU, in which on-chip ROM can be programmed by using the USB, ...
Figure 15.33 USB External Circuit in Self-Powered Mode (When External Transceiver Is Used)	588	Figure 15.33 and note amended External transceiver (ISP1104 manufactured by NXP) Note: 3. In 64F2215, HD64F2215R, HD64F2215T, HD6432215A, HD6432215B, HD6432215C, Pxx should be assigned to an output port as the D+ pull-up control pin. In 64F2215U, HD64F2215RU, and HD64F2215TU, in which on-chip ROM can be programmed by using the USB, ...

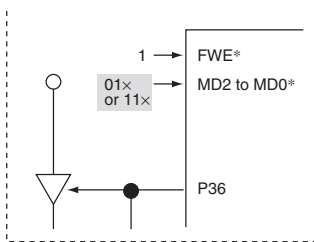
Item	Page	Revision (See Manual for Details)
15.9.1 Operating Frequency	589	Subheading amended and note added • In H8S/2215R and H8S/2215T Note: On the H8S/2215T, use a 16-MHz or 24-MHz system clock for the MCU, even if a 48-MHz oscillator or 48-MHz external clock is used as the USB operation clock.
16.1 Features	599	Description and note * amended • Conversion time: 8.1 µs per channel (at 16-MHz operation), 10.7 µs per channel (at 24-MHz operation)* Note: * Available only in H8S/2215R and H8S/2215T.
16.5.1 Single Mode	606	Description amended ... 1. A/D conversion is started when the ADST bit is set to 1, according to software, TPU, or external trigger input. ...
16.8.6 Module Stop Mode Setting	616	Section 16.8.6 added
17.5 Usage Note 17.5.1 Module Stop Mode Setting	621	Sections 17.5 to 17.5.1 added
18. RAM	623	Description amended This LSI has on-chip high-speed static RAM. … Table amended

Product Class		ROM Type	RAM Size	RAM Address
H8S/2215 Group	HD64F2215R	Flash memory Version	20 kbytes	H'FFA000 to H'FFEFBF H'FFFFC0 to H'FFFFFF
	HD64F2215RU			
	HD64F2215T			
	HD64F2215TU			
	HD64F2215		16 kbytes	H'FFB000 to H'FFEFBF H'FFFFC0 to H'FFFFFF
	HD64F2215U			
	HD6432215A	Masked ROM Version		
	HD6432215B			
	HD6432215C		8 kbytes	H'FFD000 to H'FFEFBF H'FFFFC0 to H'FFFFFF

19.1 Features	625	Description amended • Size	

Product Category		ROM Size	ROM Addresses
H8S/2215 Group	HD64F2215, HD64F2215U, HD64F2215R, HD64F2215RU, HD64F2215T, HD64F2215TU	256 kbytes	H'000000 to H'03FFFF (Modes 6 and 7)

		• Two flash memory operating modes — Boot mode (SCI boot mode: HD64F2215, HD64F2215R, HD64F2215T, USB boot mode: HD64F2215TU, HD64F2215RU, HD64F2215TU)

Item	Page	Revision (See Manual for Details)
19.2 Mode Transitions Figure 19.2 Flash Memory State Transitions	627	Note 3 amended Note: 3. 10x applies only to the HD64F2215RU and HD64F2215TU with 24-MHz system clock.
19.6 On-Board Programming Modes Table 19.3 Setting On-Board Programming Modes	639	Table 19.3 replaced
19.6.1 SCI Boot Mode (HD64F2215, HD64F2215R, HD64F2215T)		Section 19.6.1 title amended
Table 19.5 System Clock Frequencies for Which Automatic Adjustment of LSI Bit Rate Is Possible	642	Table 19.5 amended HD64F2215: 13 to 16 MHz HD64F2215R: 13 to 24 MHz HD64F2215T: 16 MHz and 24 MHz
19.6.2 USB Boot Mode (HD64F2215U, HD64F2215RU, and HD64F2215TU)	643	Description amended • Features — Selection of ... — HD64F2215U: Supports only ... HD64F2215RU and HD64F2215TU: Supports either 16-MHz or 24-MHz system clock, with USB ... • Notes on USB Boot Mode Execution — With the HD64F2215, use a 16-MHz system clock and ...With the HD64F2215RU or HD64F2215TU, use a 16-MHz or 24-MHz system clock and ...
Figure 19.7 System Configuration Diagram when using USB Boot Mode	644	Figure 19.7 amended 

Item	Page	Revision (See Manual for Details)
Section 21 Clock Pulse Generator Figure 21.1 Block Diagram of Clock Pulse Generator	665	Note * amended Note: * ×2 is available only in H8S/2215R and H8S/2215T.
21.2 System Clock Oscillator	669	Description amended The system clock can be supplied by connecting a crystal or ceramic resonator, or by input of an external clock. Suitable resonators differ depending on the product. For details, see table 21.1.
Table 21.1 List of Suitable Resonators		Table 21.1 added
21.2.2 Connecting a Ceramic Resonator (H8S/2215T)	670	Section 21.2.2 added
22.4.3 Setting Oscillation Stabilization Time after Clearing Software Standby Mode Table 22.3 Oscillation Stabilization Time Settings	689	Table and note amended 24 MHz*2 20 MHz*1 Notes: 1. Only in H8S/2215R. 2. Only in H8S/2215R and H8S/2215T.
25.6 A/D Conversion Characteristics Table 25.9 A/D Conversion Characteristics	776	Table 25.9 amended

	Condition A and B			
Item	**Min.**	**Typ.**	**Max.**	**Unit**
Resolution	10	10	10	bits
Conversion time	8.1	—	—	µs
Analog input capacitance	—	—	20	pF
Permissible signal-source impedance	—	—	5	kΩ
Nonlinearity error	—	—	±6.0	LSB
Offset error	—	—	±4.0	LSB
Full-scale error	—	—	±4.0	LSB
Quantization	—	—	±6.5	LSB
Absolute accuracy	—	—	±8.0	LSB

Section 26 Electrical Characteristics (H8S/2215T)	781 to 804	Section 26 added
Appendix B Product Model Lineup	809	Note deleted

Contents

Figures

Section 6 Bus Controller

Section 7 DMA Controller (DMAC)

Section 8 Data Transfer Controller (DTC)

Section 10 16-Bit Timer Pulse Unit (TPU)

Section 11 8-Bit Timers (TMR)

Section 12 Watchdog Timer (WDT)

Section 13 Serial Communication Interface

Section 14 Boundary Scan Function

Section 15 Universal Serial Bus Interface (USB)

Section 16 A/D Converter

Section 17 D/A Converter

Section 19 Flash Memory (F-ZTAT Version)

Section 20 Masked ROM

Section 21 Clock Pulse Generator

Section 22 Power-Down Modes

Section 24 Electrical Characteristics (H8S/2215)

Section 25 Electrical Characteristics (H8S/2215R)

Section 26 Electrical Characteristics (H8S/2215T)

Appendix

Tables

Section 6 Bus Controller

Section 7 DMA Controller (DMAC)

Section 8 Data Transfer Controller (DTC)

Section 9 I/O Ports

Section 26 Electrical Characteristics (H8S/2215T)

Section 1 Overview

1.1 Overview

- High-speed H8S/2000 central processing unit with 16-bit architecture
  - — Upward-compatible with H8/300 and H8/300H CPUs on an object level
  - — Sixteen 16-bit general registers
  - — 65 basic instructions
- Various peripheral functions
  - — DMA controller (DMAC)
  - — Data transfer controller (DTC)
  - — 16-bit timer-pulse unit (TPU)
  - — 8-bit timer (TMR)
  - — Watchdog timer (WDT)
  - — Asynchronous or clocked synchronous serial communication interface (SCI)
  - — Boundary scan
  - — Universal serial bus (USB)
  - — 10-bit A/D converter
  - — 8-bit D/A converter
  - — User debug interface (H-UDI)*
  - — Clock pulse generator

  Note: * Available only in H8S/2215R and H8S/2215T.
- On-chip memory

ROM	Part No.	ROM	RAM	Remarks
F-ZTAT Version	HD64F2215	256 kbytes	16 kbytes	SCI boot version
	HD64F2215U	256 kbytes	16 kbytes	USB boot version
	HD64F2215T	256 kbytes	20 kbytes	SCI boot version
	HD64F2215TU	256 kbytes	20 kbytes	USB boot version
	HD64F2215R	256 kbytes	20 kbytes	SCI boot version
	HD64F2215RU	256 kbytes	20 kbytes	USB boot version
Masked ROM Version	HD6432215B	128 kbytes	16 kbytes	—
	HD6432215C	64 kbytes	8 kbytes	—

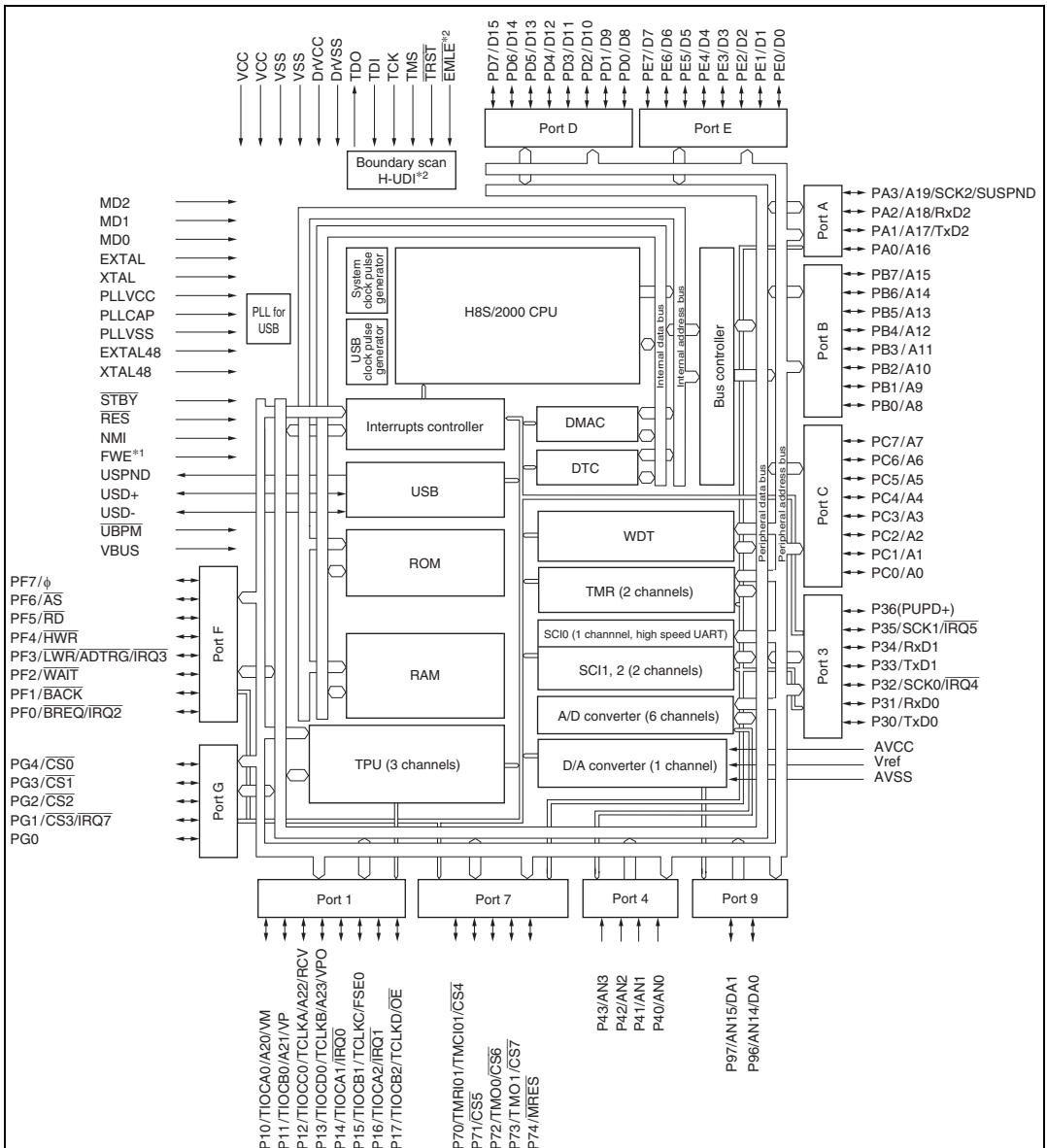
- General I/O ports

	Modes 4 and 5	Mode 6	Mode 7
— I/O pins:	41	41	68
— Input-only pins:	15	23	7

- Supports various power-down states
- Compact package

Package	(Code)	Body Size	Pin Pitch	Remarks
TQFP-120	TFP-120, TFP-120V	14.0 × 14.0 mm	0.4 mm	—
P-LFBGA-112	BP-112, BP-112V	10.0 × 10.0 mm	0.8 mm	—

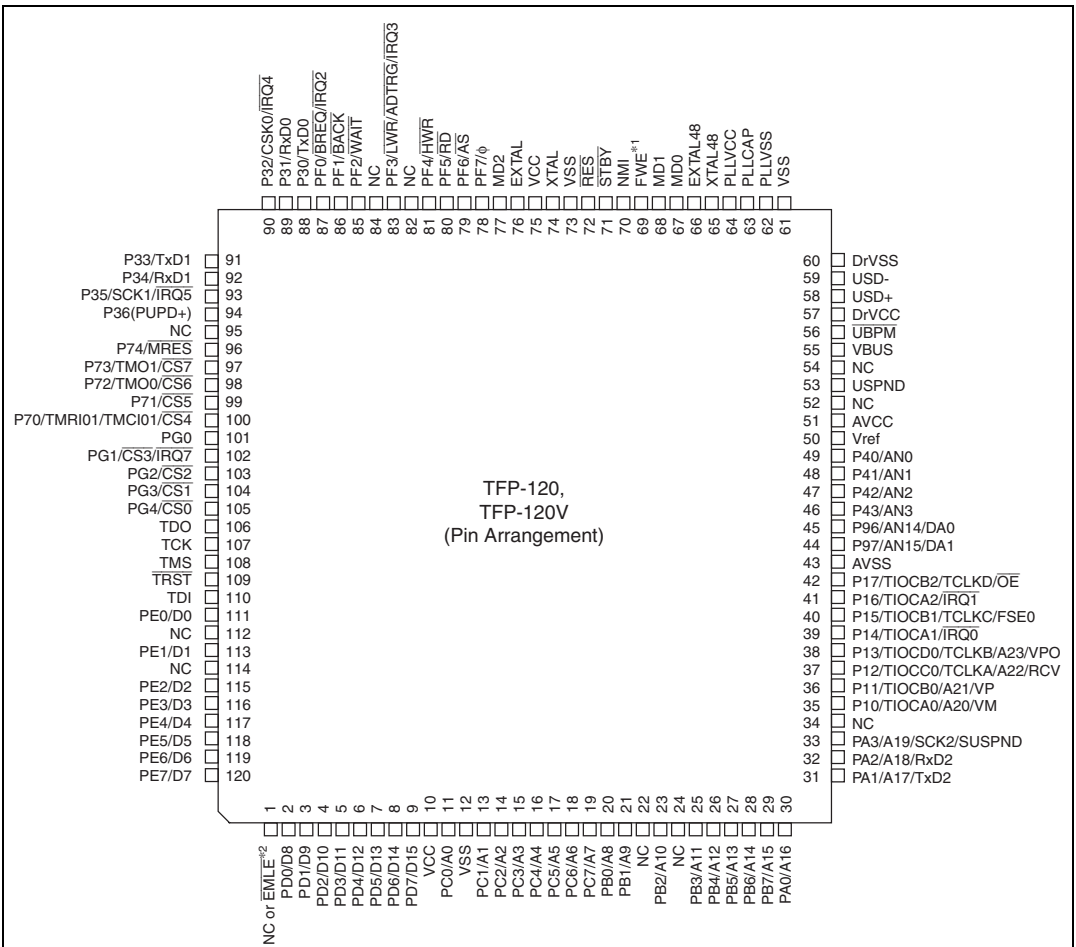
1.2 Internal Block Diagram



Notes: 1. The FWE pin is only provided in the flash memory version.
2. The H-UDI function and EMLE pin are only provided in H8S/2215R and H8S/2215T.

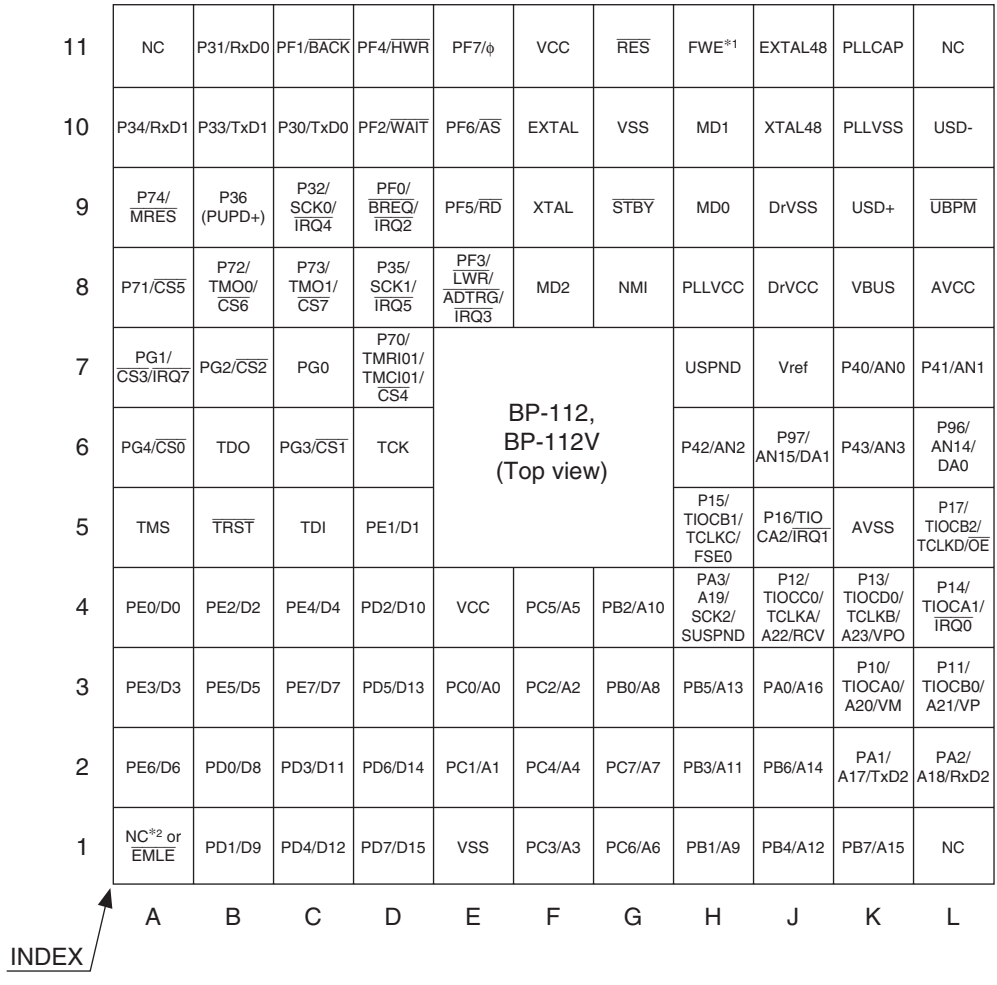
Figure 1.1 Internal Block Diagram

1.3 Pin Arrangement



Notes: NC (No Connection): These pins should not be connected; they should be left open.
1. The FWE pin is only provided in the flash memory version.
2. NC pin in H8S/2215. EMLE pin in H8S/2215R and H8S/2215T.

Figure 1.2 Pin Arrangement (TFP-120, TFP-120V)



	A	B	C	D	E	F	G	H	J	K	L
11	NC	P31/RxD0	PF1/$\overline{\text{BACK}}$	PF4/$\overline{\text{HWR}}$	PF7/ϕ	VCC	$\overline{\text{RES}}$	FWE*1	EXTAL48	PLLCAP	NC
10	P34/RxD1	P33/TxD1	P30/TxD0	PF2/$\overline{\text{WAIT}}$	PF6/$\overline{\text{AS}}$	EXTAL	VSS	MD1	XTAL48	PLLVSS	USD-
9	P74/ $\overline{\text{MRES}}$	P36 (PUPD+)	P32/ SCK0/ $\overline{\text{IRQ4}}$	PF0/ $\overline{\text{BREQ}}$/ $\overline{\text{IRQ2}}$	PF5/$\overline{\text{RD}}$	XTAL	$\overline{\text{STBY}}$	MD0	DrVSS	USD+	$\overline{\text{UBPM}}$
8	P71/$\overline{\text{CS5}}$	P72/ TMO0/ $\overline{\text{CS6}}$	P73/ TMO1/ $\overline{\text{CS7}}$	P35/ SCK1/ $\overline{\text{IRQ5}}$	PF3/ $\overline{\text{LWR}}$/ $\overline{\text{ADTRG}}$/ $\overline{\text{IRQ3}}$	MD2	NMI	PLLVCC	DrVCC	VBUS	AVCC
7	PG1/ $\overline{\text{CS3}}$/$\overline{\text{IRQ7}}$	PG2/$\overline{\text{CS2}}$	PG0	P70/ TMRI01/ TMCI01/ $\overline{\text{CS4}}$	BP-112, BP-112V (Top view)			USPND	Vref	P40/AN0	P41/AN1
6	PG4/$\overline{\text{CS0}}$	TDO	PG3/$\overline{\text{CS1}}$	TCK				P42/AN2	P97/ AN15/DA1	P43/AN3	P96/ AN14/ DA0
5	TMS	$\overline{\text{TRST}}$	TDI	PE1/D1				P15/ TIOCB1/ TCLKC/ FSE0	P16/TIO CA2/$\overline{\text{IRQ1}}$	AVSS	P17/ TIOCB2/ TCLKD/$\overline{\text{OE}}$
4	PE0/D0	PE2/D2	PE4/D4	PD2/D10	VCC	PC5/A5	PB2/A10	PA3/ A19/ SCK2/ SUSPND	P12/ TIOCC0/ TCLKA/ A22/RCV	P13/ TIOCD0/ TCLKB/ A23/VPO	P14/ TIOCA1/ $\overline{\text{IRQ0}}$
3	PE3/D3	PE5/D5	PE7/D7	PD5/D13	PC0/A0	PC2/A2	PB0/A8	PB5/A13	PA0/A16	P10/ TIOCA0/ A20/VM	P11/ TIOCB0/ A21/VP
2	PE6/D6	PD0/D8	PD3/D11	PD6/D14	PC1/A1	PC4/A4	PC7/A7	PB3/A11	PB6/A14	PA1/ A17/TxD2	PA2/ A18/RxD2
1	NC*2 or $\overline{\text{EMLE}}$	PD1/D9	PD4/D12	PD7/D15	VSS	PC3/A3	PC6/A6	PB1/A9	PB4/A12	PB7/A15	NC

Notes: NC (No Connection): These pins should not be connected; they should be left open.
1. The FWE pin is only provided in the flash memory version.
2. NC in H8S/2215. $\overline{\text{EMLE}}$ pin in H8S/2215R and H8S/2215T.

Figure 1.3 Pin Arrangement (BP-112, BP-112V)

1.4 Pin Functions in Each Operating Mode

Pin No.		Pin Name				
TFP-120, TFP-120V	BP-112, BP-112V	Mode 4	Mode 5	Mode 6	Mode 7[*1]	PROM Mode
1	A1	NC or $\overline{\text{EMLE}}$[*2]	NC or $\overline{\text{EMLE}}$[*2]	NC or $\overline{\text{EMLE}}$[*2]	NC or $\overline{\text{EMLE}}$[*2]	NC
2	B2	D8	D8	D8	PD0	D0
3	B1	D9	D9	D9	PD1	D1
4	D4	D10	D10	D10	PD2	D2
5	C2	D11	D11	D11	PD3	D3
6	C1	D12	D12	D12	PD4	D4
7	D3	D13	D13	D13	PD5	D5
8	D2	D14	D14	D14	PD6	D6
9	D1	D15	D15	D15	PD7	D7
10	E4	VCC	VCC	VCC	VCC	VCC
11	E3	A0	A0	PC1/A0	PC0	A0
12	E1	VSS	VSS	VSS	VSS	VSS
13	E2	A1	A1	PC1/A1	PC1	A1
14	F3	A2	A2	PC2/A2	PC2	A2
15	F1	A3	A3	PC3/A3	PC3	A3
16	F2	A4	A4	PC4/A4	PC4	A4
17	F4	A5	A5	PC5/A5	PC5	A5
18	G1	A6	A6	PC6/A6	PC6	A6
19	G2	A7	A7	PC7/A7	PC7	A7
20	G3	PB0/A8	PB0/A8	PB0/A8	PB0	A8
21	H1	PB1/A9	PB1/A9	PB1/A9	PB1	A9
22	—	NC	NC	NC	NC	NC
23	G4	PB2/A10	PB2/A10	PB2/A10	PB2	A10
24	—	NC	NC	NC	NC	NC
25	H2	PB3/A11	PB3/A11	PB3/A11	PB3	A11
26	J1	PB4/A12	PB4/A12	PB4/A12	PB4	A12
27	H3	PB5/A13	PB5/A13	PB5/A13	PB5	A13
28	J2	PB6/A14	PB6/A14	PB6/A14	PB6	A14
29	K1	PB7/A15	PB7/A15	PB7/A15	PB7	A15

Pin No.		Pin Name				
TFP-120, TFP-120V	BP-112, BP-112V	Mode 4	Mode 5	Mode 6	Mode 7[*1]	PROM Mode
30	J3	PA0/A16	PA0/A16	PA0/A16	PA0	A16
31	K2	PA1/A17/TxD2	PA1/A17/TxD2	PA1/A17/TxD2	PA1/TxD2	A17
32	L2	PA2/A18/RxD2	PA2/A18/RxD2	PA2/A18/RxD2	PA2/RxD2	A18
33	H4	PA3/A19/SCK2/ SUSPND	PA3/A19/SCK2/ SUSPND	PA3/A19/SCK2/ SUSPND	PA3/SCK2	NC
34	—	NC	NC	NC	NC	NC
35	K3	P10/TIOCA0/ A20/VM	P10/TIOCA0/ A20/VM	P10/TIOCA0/ A20/VM	P10/TIOCA0	NC
36	L3	P11/TIOCB0/ A21/VP	P11/TIOCB0/ A21/VP	P11/TIOCB0/ A21/VP	P11/TIOCB0	NC
37	J4	P12/TIOCC0/ TCLKA/A22/ RCV	P12/TIOCC0/ TCLKA/A22/ RCV	P12/TIOCC0/ TCLKA/A22/ RCV	P12/TIOCC0/ TCLKA	NC
38	K4	P13/TIOCD0/ TCLKB/A23/ VPO	P13/TIOCD0/ TCLKB/A23/ VPO	P13/TIOCD0/ TCLKB/A23/ VPO	P13/TIOCD0/ TCLKB	NC
39	L4	P14/TIOCA1/ $\overline{\text{IRQ0}}$	P14/TIOCA1/ $\overline{\text{IRQ0}}$	P14/TIOCA1/ $\overline{\text{IRQ0}}$	P14/TIOCA1/ $\overline{\text{IRQ0}}$	VSS
40	H5	P15/TIOCB1/ TCLKC/FSE0	P15/TIOCB1/ TCLKC/FSE0	P15/TIOCB1/ TCLKC/FSE0	P15/TIOCB1/ TCLKC	NC
41	J5	P16/TIOCA2/ $\overline{\text{IRQ1}}$	P16/TIOCA2/ $\overline{\text{IRQ1}}$	P16/TIOCA2/ $\overline{\text{IRQ1}}$	P16/TIOCA2/ $\overline{\text{IRQ1}}$	VSS
42	L5	P17/TIOCB2/ TCLKD/$\overline{\text{OE}}$	P17/TIOCB2/ TCLKD/$\overline{\text{OE}}$	P17/TIOCB2/ TCLKD/$\overline{\text{OE}}$	P17/TIOCB2/ TCLKD/$\overline{\text{OE}}$	NC
43	K5	AVSS	AVSS	AVSS	AVSS	VSS
44	J6	P97/AN15/DA1	P97/AN15/DA1	P97/AN15/DA1	P97/AN15/DA1	NC
45	L6	P96/AN14/DA0	P96/AN14/DA0	P96/AN14/DA0	P96/AN14/DA0	NC
46	K6	P43/AN3	P43/AN3	P43/AN3	P43/AN3	NC
47	H6	P42/AN2	P42/AN2	P42/AN2	P42/AN2	NC
48	L7	P41/AN1	P41/AN1	P41/AN1	P41/AN1	NC
49	K7	P40/AN0	P40/AN0	P40/AN0	P40/AN0	NC
50	J7	Vref	Vref	Vref	Vref	VCC
51	L8	AVCC	AVCC	AVCC	AVCC	VCC
52	—	NC	NC	NC	NC	NC

Pin No.		Pin Name				
TFP-120, TFP-120V	BP-112, BP-112V	Mode 4	Mode 5	Mode 6	Mode 7[*1]	PROM Mode
53	H7	USPND	USPND	USPND	—	NC
54	—	NC	NC	NC	NC	NC
55	K8	VBUS	VBUS	VBUS	VSS	VSS
56	L9	$\overline{\text{UBPM}}$	$\overline{\text{UBPM}}$	$\overline{\text{UBPM}}$	VSS	VSS
57	J8	DrVCC	DrVCC	DrVCC	VSS	VCC
58	K9	USD+	USD+	USD+	—	NC
59	L10	USD-	USD-	USD-	—	NC
60	J9	DrVSS	DrVSS	DrVSS	—	VSS
61	—	VSS	VSS	VSS	VSS	VSS
62	K10	PLLVSS	PLLVSS	PLLVSS	—	VSS
63	K11	PLLCAP	PLLCAP	PLLCAP	NC	NC
64	H8	PLLVCC	PLLVCC	PLLVCC	—	VCC
65	J10	XTAL48	XTAL48	XTAL48	—	NC
66	J11	EXTAL48	EXTAL48	EXTAL48	—	VCC
67	H9	MD0	MD0	MD0	MD0	VSS
68	H10	MD1	MD1	MD1	MD1	VSS
69	H11	FWE	FWE	FWE	FWE	FWE
70	G8	NMI	NMI	NMI	NMI	VCC
71	G9	$\overline{\text{STBY}}$	$\overline{\text{STBY}}$	$\overline{\text{STBY}}$	$\overline{\text{STBY}}$	VCC
72	G11	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$
73	G10	VSS	VSS	VSS	VSS	VSS
74	F9	XTAL	XTAL	XTAL	XTAL	XTAL
75	F11	VCC	VCC	VCC	VCC	VCC
76	F10	EXTAL	EXTAL	EXTAL	EXTAL	EXTAL
77	F8	MD2	MD2	MD2	MD2	VSS
78	E11	PF7/ϕ	PF7/ϕ	PF7/ϕ	PF7/ϕ	NC
79	E10	$\overline{\text{AS}}$	$\overline{\text{AS}}$	$\overline{\text{AS}}$	PF6	NC
80	E9	$\overline{\text{RD}}$	$\overline{\text{RD}}$	$\overline{\text{RD}}$	PF5	NC
81	D11	$\overline{\text{HWR}}$	$\overline{\text{HWR}}$	$\overline{\text{HWR}}$	PF4	NC
82	—	NC	NC	NC	NC	NC

Pin No.		Pin Name				
TFP-120, TFP-120V	BP-112, BP-112V	Mode 4	Mode 5	Mode 6	Mode 7*1	PROM Mode
83	E8	PF3/$\overline{\text{LWR}}$/ $\overline{\text{ADTRG}}$/$\overline{\text{IRQ3}}$	PF3/$\overline{\text{LWR}}$/ $\overline{\text{ADTRG}}$/$\overline{\text{IRQ3}}$	PF3/$\overline{\text{LWR}}$/ $\overline{\text{ADTRG}}$/$\overline{\text{IRQ3}}$	PF3/$\overline{\text{ADTRG}}$/ $\overline{\text{IRQ3}}$	VCC
84	—	NC	NC	NC	NC	NC
85	D10	PF2/$\overline{\text{WAIT}}$	PF2/$\overline{\text{WAIT}}$	PF2/$\overline{\text{WAIT}}$	PF2	NC
86	C11	PF1/$\overline{\text{BACK}}$	PF1/$\overline{\text{BACK}}$	PF1/$\overline{\text{BACK}}$	PF1	NC
87	D9	PF0/$\overline{\text{BREQ}}$/ $\overline{\text{IRQ2}}$	PF0/$\overline{\text{BREQ}}$/ $\overline{\text{IRQ2}}$	PF0/$\overline{\text{BREQ}}$/ $\overline{\text{IRQ2}}$	PF0/$\overline{\text{IRQ2}}$	VCC
88	C10	P30/TxD0	P30/TxD0	P30/TxD0	P30/TxD0	NC
89	B11	P31/RxD0	P31/RxD0	P31/RxD0	P31/RxD0	NC
90	C9	P32/SCK0/$\overline{\text{IRQ4}}$	P32/SCK0/$\overline{\text{IRQ4}}$	P32/SCK0/$\overline{\text{IRQ4}}$	P32/SCK0/$\overline{\text{IRQ4}}$	NC
91	B10	P33/TxD1	P33/TxD1	P33/TxD1	P33/TxD1	NC
92	A10	P34/RxD1	P34/RxD1	P34/RxD1	P34/RxD1	NC
93	D8	P35/SCK1/$\overline{\text{IRQ5}}$	P35/SCK1/$\overline{\text{IRQ5}}$	P35/SCK1/$\overline{\text{IRQ5}}$	P35/SCK1/$\overline{\text{IRQ5}}$	NC
94	B9	P36 (PUPD+)	P36 (PUPD+)	P36 (PUPD+)	P36 (PUPD+)*3	NC
95	—	NC	NC	NC	NC	NC
96	A9	P74/$\overline{\text{MRES}}$	P74/$\overline{\text{MRES}}$	P74/$\overline{\text{MRES}}$	P74/$\overline{\text{MRES}}$	NC
97	C8	P73/TMO1/$\overline{\text{CS7}}$	P73/TMO1/$\overline{\text{CS7}}$	P73/TMO1/$\overline{\text{CS7}}$	P73/TMO1	NC
98	B8	P72/TMO0/$\overline{\text{CS6}}$	P72/TMO0/$\overline{\text{CS6}}$	P72/TMO0/$\overline{\text{CS6}}$	P72/TMO0	NC
99	A8	P71/$\overline{\text{CS5}}$	P71/$\overline{\text{CS5}}$	P71/$\overline{\text{CS5}}$	P71	NC
100	D7	P70/TMRI01/ TMCI01/$\overline{\text{CS4}}$	P70/TMRI01/ TMCI01/$\overline{\text{CS4}}$	P70/TMRI01/ TMCI01/$\overline{\text{CS4}}$	P70/TMRI01/ TMCI01	NC
101	C7	PG0	PG0	PG0	PG0	NC
102	A7	PG1/$\overline{\text{CS3}}$/$\overline{\text{IRQ7}}$	PG1/$\overline{\text{CS3}}$/$\overline{\text{IRQ7}}$	PG1/$\overline{\text{CS3}}$/$\overline{\text{IRQ7}}$	PG1/$\overline{\text{IRQ7}}$	NC
103	B7	PG2/$\overline{\text{CS2}}$	PG2/$\overline{\text{CS2}}$	PG2/$\overline{\text{CS2}}$	PG2	NC
104	C6	PG3/$\overline{\text{CS1}}$	PG3/$\overline{\text{CS1}}$	PG3/$\overline{\text{CS1}}$	PG3	NC
105	A6	PG4/$\overline{\text{CS0}}$	PG4/$\overline{\text{CS0}}$	PG4/$\overline{\text{CS0}}$	PG4	NC
106	B6	TDO	TDO	TDO	TDO	VCC
107	D6	TCK	TCK	TCK	TCK	VCC
108	A5	TMS	TMS	TMS	TMS	VCC
109	B5	$\overline{\text{TRST}}$	$\overline{\text{TRST}}$	$\overline{\text{TRST}}$	$\overline{\text{TRST}}$	$\overline{\text{RES}}$
110	C5	TDI	TDI	TDI	TDI	VCC
111	A4	PE0/D0	PE0/D0	PE0/D0	PE0	NC

Pin No.		Pin Name				
TFP-120, TFP-120V	**BP-112, BP-112V**	**Mode 4**	**Mode 5**	**Mode 6**	**Mode 7**[*1]	**PROM Mode**
112	—	NC	NC	NC	NC	NC
113	D5	PE1/D1	PE1/D1	PE1/D1	PE1	NC
114	—	NC	NC	NC	NC	NC
115	B4	PE2/D2	PE2/D2	PE2/D2	PE2	NC
116	A3	PE3/D3	PE3/D3	PE3/D3	PE3	VCC
117	C4	PE4/D4	PE4/D4	PE4/D4	PE4	VSS
118	B3	PE5/D5	PE5/D5	PE5D5	PE5	$\overline{OE}$
119	A2	PE6/D6	PE6/D6	PE6/D6	PE6	$\overline{WE}$
120	C3	PE7/D7	PE7/D7	PE7/D7	PE7	$\overline{CE}$
—	A1, A11, L1, L11	NC	NC	NC	NC	NC

Notes: NC (No Connection): These pins should not be connected; they should be left open.

1. The USB may be unusable in mode 7 in some cases. See section 3, MCU Operating Mode, for details.
2. NC in H8S/2215. $\overline{EMLE}$ pin in H8S/2215R and H8S/2215T.
3. PUPD+ pin in H8S/2215R and H8S/2215T.

1.5 Pin Functions

Type	Symbol	Pin No. TFP-120, TFP-120V	Pin No. BP-112, BP-112V	I/O	Function
Power Supply	VCC	10 75	E4 F11	Input	Power supply pins. Connect all these pins to the system power supply.
	VSS	12 61 73	E1 G10	Input	Ground pins. Connect all these pins to the system power supply (0 V).
	PLLVCC	64	H8	Input	Power supply pin for internal PLL oscillator. Connect this pin to the system power supply.
	PLLVSS	62	K10	Input	Ground pin for an on-chip PLL oscillator.
	PLLCAP	63	K11	Output	External capacitor pin for an on-chip PLL oscillator.
Clock	XTAL	74	F9	Input	For connection to a crystal resonator. For examples of crystal resonator connection and external clock input, see section 21, Clock Pulse Generator.
	EXTAL	76	F10	Input	For connection to a crystal resonator. (An external clock can be supplied from the EXTAL pin.) For examples of crystal resonator connection and external clock input, see section 21, Clock Pulse Generator.
	XTAL48	65	J10	Input	USB operating clock input pins.
	EXTAL48	66	J11	Input	48-MHz clock for USB communications is input. For examples of using an on-chip PLL, EXTAL48 must be fixed low and XTAL48 must be open.
	ϕ	78	E11	Output	Supplies the system clock to external devices.
Operating Mode Control	MD2 MD1 MD0	77 68 67	F8 H10 H9	Input	Except for mode changing, be sure to fix the levels of the mode pins (MD2 to MD0) by pulling them down or pulling them up until the power turns off.

Type	Symbol	Pin No. TFP-120, TFP-120V	Pin No. BP-112, BP-112V	I/O	Function
System Control	$\overline{\text{RES}}$	72	G11	Input	Reset input pin. When this pin is driven low, the chip is reset.
	$\overline{\text{STBY}}$	71	G9	Input	When this pin is driven low, a transition is made to hardware standby mode.
	$\overline{\text{MRES}}$	96	A9	Input	When this pin is driven low, a transition is made to manual reset mode.
	$\overline{\text{BREQ}}$	87	D9	Input	Used by an external bus master to issue a bus request to this LSI
	$\overline{\text{BACK}}$	86	C11	Output	Indicates that the bus has been released to an external bus master.
	FWE	69	H11	Input	Pin for use by flash memory. This pin is only used in the flash memory version. In the mask ROM version it should be fixed at 0.
	$\overline{\text{EMLE}}$*1	1*1	A1*1	Input	Emulator enable pin. Leave open if the E10A is not used. Drive low level only if E10A is used.
Interrupts	NMI	70	G8	Input	Nonmaskable interrupt pin. If this pin is not used, it should be fixed high.
	$\overline{\text{IRQ7}}$	102	A7	Input	These pins request a maskable interrupt.
	$\overline{\text{IRQ5}}$	93	D8		
	$\overline{\text{IRQ4}}$	90	C9		
	$\overline{\text{IRQ3}}$	83	E8		
	$\overline{\text{IRQ2}}$	87	D9		
	$\overline{\text{IRQ1}}$	41	J5		
	$\overline{\text{IRQ0}}$	39	L4		

Type	Symbol	Pin No. TFP-120, TFP-120V	Pin No. BP-112, BP-112V	I/O	Function
Address bus	A23	38	K4	Output	These pins output an address.
	A22	37	J4		
	A21	36	L3		
	A20	35	K3		
	A19	33	H4		
	A18	32	L2		
	A17	31	K2		
	A16	30	J3		
	A15	29	K1		
	A14	28	J2		
	A13	27	H3		
	A12	26	J1		
	A11	25	H2		
	A10	23	G4		
	A9	21	H1		
	A8	20	G3		
	A7	19	G2		
	A6	18	G1		
	A5	17	F4		
	A4	16	F2		
	A3	15	F1		
	A2	14	F3		
	A1	13	E2		
	A0	11	E3		

Type	Symbol	Pin No. TFP-120, TFP-120V	Pin No. BP-112, BP-112V	I/O	Function
Data bus	D15	9	D1	I/O	These pins constitute a bi-directional data bus.
	D14	8	D2		
	D13	7	D3		
	D12	6	C1		
	D11	5	C2		
	D10	4	D4		
	D9	3	B1		
	D8	2	B2		
	D7	120	C3		
	D6	119	A2		
	D5	118	B3		
	D4	117	C4		
	D3	116	A3		
	D2	115	B4		
	D1	113	D5		
	D0	111	A4		

Type	Symbol	Pin No. TFP-120, TFP-120V	Pin No. BP-112, BP-112V	I/O	Function
Bus Control	$\overline{CS7}$	97	C8	Output	Signals for selecting areas 7 to 0.
	$\overline{CS6}$	98	B8		
	$\overline{CS5}$	99	A8		
	$\overline{CS4}$	100	D7		
	$\overline{CS3}$	102	A7		
	$\overline{CS2}$	103	B7		
	$\overline{CS1}$	104	C6		
	$\overline{CS0}$	105	A6		
	$\overline{AS}$	79	E10	Output	When this pin is low, it indicates that address output on the address bus is enabled.
	$\overline{RD}$	80	E9	Output	When this pin is low, it indicates that the external address space can be read.
	$\overline{HWR}$	81	D11	Output	A strobe signal that writes to external space and indicates that the upper half (D15 to D8) of the data bus is enabled.
	$\overline{LWR}$	83	E8	Output	A strobe signal that writes to external space and indicates that the lower half (D7 to D0) of the data bus is enabled.
	$\overline{WAIT}$	85	D10	Input	Requests insertion of a wait state in the bus cycle when accessing external 3-state address space.

Type	Symbol	Pin No. TFP-120, TFP-120V	Pin No. BP-112, BP-112V	I/O	Function
16-bit timer pulse unit (TPU)	TCLKA	37	J4	Input	TPU external clock input pins.
	TCLKB	38	K4		
	TCLKC	40	H5		
	TCLKD	42	L5		
	TIOCA0	35	K3	I/O	The TGRA_0 to TGRD_0 input capture input/output compare output/PWM output pins.
	TIOCB0	36	L3		
	TIOCC0	37	J4		
	TIOCD0	38	K4		
	TIOCA1	39	L4	I/O	The TGRA_1 to TGRB_1 input capture input/output compare output/PWM output pins.
	TIOCB1	40	H5		
	TIOCA2	41	J5	I/O	The TGRA_2 to TGRB_2 input capture input/output compare output/PWM output pins.
	TIOCB2	42	L5		
8-bit timer (TMR)	TMO1	97	C8	Output	Compare match output pins.
	TMO0	98	B8		
	TMCI01	100	D7	Input	Input pins for the external clock input to the counter.
	TMRI01	100	D7	Input	The counter reset input pins.
Serial Communication interface (SCI)	TxD2	31	K2	Output	Data output pins
	TxD1	91	B10		
	TxD0	88	C10		
	RxD2	32	L2	Input	Data input pins
	RxD1	92	A10		
	RxD0	89	B11		
	SCK2	33	H4	I/O	Clock input/output pins
	SCK1	93	D8		
	SCK0	90	C9		

Type	Symbol	Pin No. TFP-120, TFP-120V	Pin No. BP-112, BP-112V	I/O	Function
A/D converter	AN15	44	J6	Input	Analog input pins for the A/D converter.
	AN14	45	L6		
	AN3	46	K6		
	AN2	47	H6		
	AN1	48	L7		
	AN0	49	K7		
	$\overline{\text{ADTRG}}$	83	E8	Input	Pin for input of an external trigger to start A/D conversion
D/A converter	DA1	44	J6	Output	Analog output pins for the D/A converter.
	DA0	45	L6		
A/D converter D/A converter	AVCC	51	L8	Input	Power supply pin for the A/D and D/A converter. When the D/A converter is not used, connect this pin to the system power supply (VCC).
	AVSS	43	K5	Input	The ground pin for the A/D and D/A converter. Connect this pin to the system power supply (0 V).
	Vref	50	J7	Input	The reference voltage input pin for the A/D and D/A converter. When the A/D and D/A converter is not used, this pin should be connected to the system power supply (VCC).
Boundary scan	TMS	108	A5	Input	Control signal input pin for the boundary scan
	TCK	107	D6	Input	Clock input pin for the boundary scan
	TD0	106	B6	Output	Data output pin for the boundary scan
	TDI	110	C5	Input	Data input pin for the boundary scan
	$\overline{\text{TRST}}$	109	B5	Input	Reset pin for the TAP controller

Type	Symbol	Pin No. TFP-120, TFP-120V	Pin No. BP-112, BP-112V	I/O	Function
USB	USD+	58	K9	I/O	USB data input/output pin
	USD-	59	L10		
	VBUS	55	K8	Input	Connection/disconnection detecting Input/output pin for the USB cable
	USPND	53	H7	Output	USB suspend output This pin is driven high when a transition is made to suspend state.
	VM	35	K3	Input	Pins to be connected to the transceiver (ISP1104) manufactured by NXP.
	VP	36	L3		
	RCV	37	J4		
	VPO	38	K4	Output	
	FSE0	40	H5		
	$\overline{OE}$	42	L5		
	SUSPND	33	H4		
	$\overline{UBPM}$	56	L9	Input	Bus power/self power mode setting Input. When the USB is used in bus power mode, this input pin must be fixed at 0. When the USB is used in self power mode, this input pin must be fixed at 1.
	DrVCC	57	J8	—	Power supply for the on-chip transceiver. Connect this pin to the system power supply.
	DrVSS	60	J9	—	Ground pin for the on-chip transceiver.
	P36 (PUPD+)	94	B9	I/O	Used for D+ pull-up control.

Type	Symbol	Pin No. TFP-120, TFP-120V	Pin No. BP-112, BP-112V	I/O	Function
I/O port	P17	42	L5	I/O	8-bit I/O pins
	P16	41	J5		
	P15	40	H5		
	P14	39	L4		
	P13	38	K4		
	P12	37	J4		
	P11	36	L3		
	P10	35	K3		
	P36	94	B9	I/O	7-bit I/O pins
	P35	93	D8		
	P34	92	A10		
	P33	91	B10		
	P32	90	C9		
	P31	89	B11		
	P30	88	C10		
	P43	46	K6	Input	4-bit input pins
	P42	47	H6		
	P41	48	L7		
	P40	49	K7		
	P74	96	A9	I/O	5-bit I/O pins
	P73	97	C8		
	P72	98	B8		
	P71	99	A8		
	P70	100	D7		
	P97	44	J6	Input	2-bit input pins
	P96	45	L6		
	PA3	33	H4	I/O	4-bit I/O pins
	PA2	32	L2		
	PA1	31	K2		
	PA0	30	J3		

Type	Symbol	Pin No. TFP-120, TFP-120V	Pin No. BP-112, BP-112V	I/O	Function
I/O port	PB7	29	K1	I/O	8-bit I/O pins
	PB6	28	J2		
	PB5	27	H3		
	PB4	26	J1		
	PB3	25	H2		
	PB2	23	G4		
	PB1	21	H1		
	PB0	20	G3		
	PC7	19	G2	I/O	8-bit I/O pins
	PC6	18	G1		
	PC5	17	F4		
	PC4	16	F2		
	PC3	15	F1		
	PC2	14	F3		
	PC1	13	E2		
	PC0	11	E3		
	PD7	9	D1	I/O	8-bit I/O pins
	PD6	8	D2		
	PD5	7	D3		
	PD4	6	C1		
	PD3	5	C2		
	PD2	4	D4		
	PD1	3	B1		
	PD0	2	B2		
	PE7	120	C3	I/O	8-bit I/O pins
	PE6	119	A2		
	PE5	118	B3		
	PE4	117	C4		
	PE3	116	A3		
	PE2	115	B4		
	PE1	113	D5		
	PE0	111	A4		

Type	Symbol	Pin No. TFP-120, TFP-120V	Pin No. BP-112, BP-112V	I/O	Function
I/O port	PF7	78	E11	I/O	8-bit I/O pins
	PF6	79	E10		
	PF5	80	E9		
	PF4	81	D11		
	PF3	83	E8		
	PF2	85	D10		
	PF1	86	C11		
	PF0	87	D9		
	PG4	105	A6	I/O	5-bit I/O pins
	PG3	104	C6		
	PG2	103	B7		
	PG1	102	A7		
	PG0	101	C7		
NC	NC	1*2	A1*2	—	NC (No Connection): These pins should not be connected; they should be left open.
		22	A11		
		24	L1		
		34	L11		
		52			
		54			
		82			
		84			
		95			
		112			
		114			

Notes: 1. Available only in H8S/2215R and H8S/2215T. (NC in H8S/2215.)
2. Available only in H8S/2215 ($\overline{\text{EMLE}}$ pin in H8S/2215R and H8S/2215T).

Section 2 CPU

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

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

2.1 Features

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

— 16 × 16-bit register-register multiply: 20 states
— 32 ÷ 16-bit register-register divide: 20 states

- Two CPU operating modes
  — Normal mode*
  — Advanced mode
  Note: * Normal mode is not available in this LSI.
- Power-down state
  — Transition to power-down state by SLEEP instruction
  — CPU clock speed selection

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

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

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

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

In addition, there are differences in address space, CCR and EXR register functions, power-down modes, etc., depending on the model.

2.1.2 Differences from H8/300 CPU

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

- More general registers and control registers
  - Eight 16-bit extended registers, and one 8-bit and two 32-bit control registers, have been added.
- Extended address space
  - Normal mode supports the same 64-kbyte address space as the H8/300 CPU.
  - Advanced mode supports a maximum 16-Mbyte address space.
- 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.

2.1.3 Differences from H8/300H CPU

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

- Additional control register
  - One 8-bit control registers have 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.

2.2 CPU Operating Modes

The H8S/2000 CPU has two operating modes: normal and advanced. Normal mode supports a maximum 64-kbyte address space. Advanced mode supports a maximum 16-Mbyte total address space. The mode is selected by the mode pins.

2.2.1 Normal Mode

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

- Address Space

  A maximum address space of 64 kbytes can be accessed.
- Extended Registers (En)

  The extended registers (E0 to E7) can be used as 16-bit registers, or as the upper 16-bit segments of 32-bit registers. When En is used as a 16-bit register it can contain any value, even when the corresponding general register (Rn) is used as an address register. If the general register is referenced in the register indirect addressing mode with pre-decrement (@–Rn) or post-increment (@Rn+) and a carry or borrow occurs, however, the value in the corresponding extended register (En) will be affected.
- Instruction Set

  All instructions and addressing modes can be used. Only the lower 16 bits of effective addresses (EA) are valid.
- Exception Vector Table and Memory Indirect Branch Addresses

  In normal mode the top area starting at H'0000 is allocated to the exception vector table. One branch address is stored per 16 bits. The exception vector table in normal mode is shown in figure 2.1. For details of the exception vector table, see section 4, Exception Handling.

  The memory indirect addressing mode (@@aa:8) employed in the JMP and JSR instructions uses an 8-bit absolute address included in the instruction code to specify a memory operand that contains a branch address. In normal mode the operand is a 16-bit (word) operand, providing a 16-bit branch address. Branch addresses can be stored in the top area from H'0000 to H'00FF. Note that this area is also used for the exception vector table.
- Stack Structure

  When the program counter (PC) is pushed onto the stack in a subroutine call, and the PC, condition-code register (CCR), and extended control register (EXR) are pushed onto the stack in exception handling, they are stored as shown in figure 2.2. EXR is not pushed onto the stack in interrupt control mode 0. For details, see section 4, Exception Handling.

Note: Normal mode is not available in this LSI.

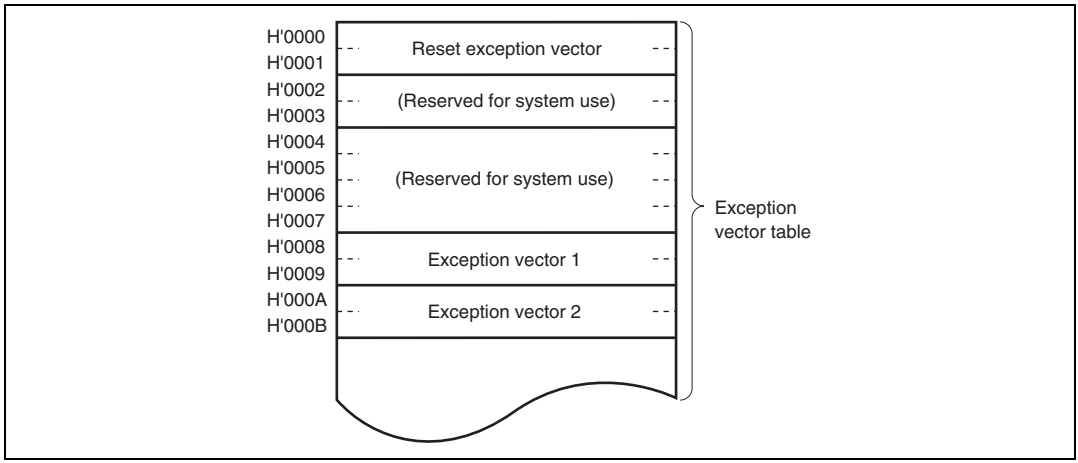
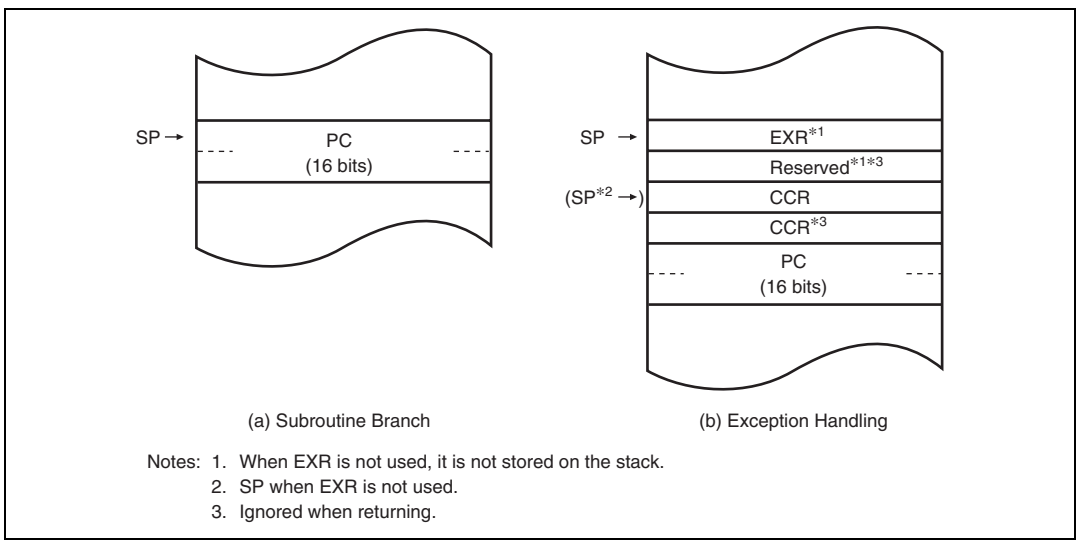


Figure 2.1 Exception Vector Table (Normal Mode)



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

Figure 2.2 Stack Structure in Normal Mode

2.2.2 Advanced Mode

- Address Space

  Linear access is provided to a 16-Mbyte maximum address space.
- 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.
- Exception Vector Table and Memory Indirect Branch Addresses

  In advanced mode the top area starting at H'00000000 is allocated to the exception vector table in units of 32 bits. In each 32 bits, the upper 8 bits are ignored and a branch address is stored in the lower 24 bits (figure 2.3). For details of the exception vector table, see section 4, Exception Handling.

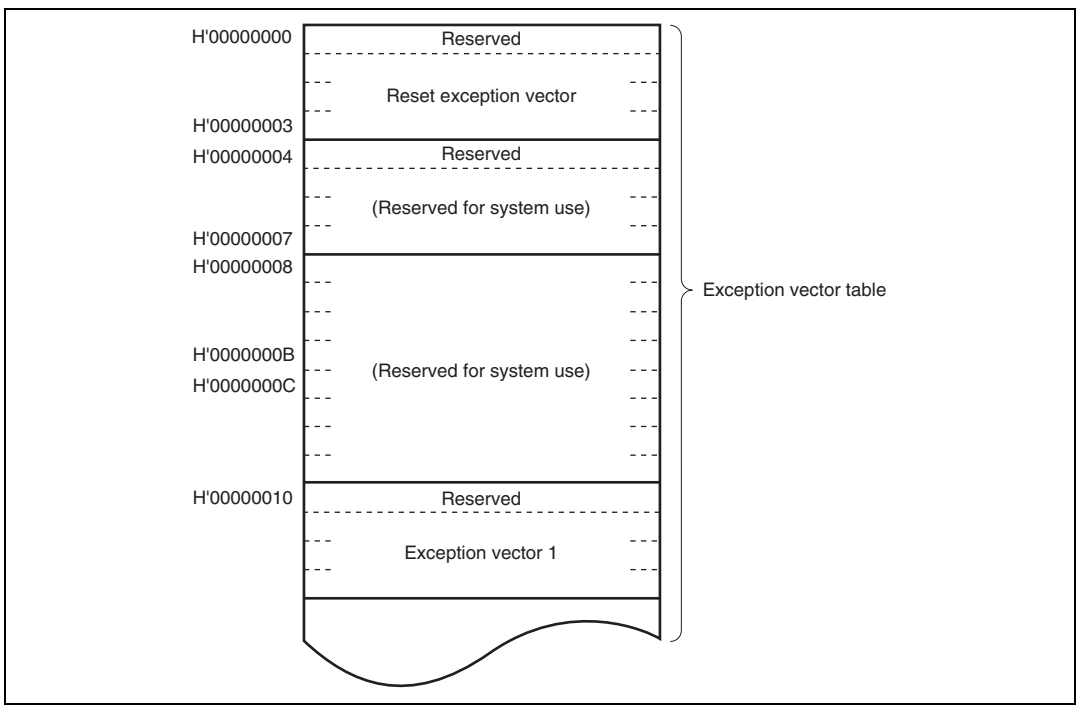


Figure 2.3 Exception Vector Table (Advanced Mode)

The memory indirect addressing mode (@@aa:8) employed in the JMP and JSR instructions uses an 8-bit absolute address included in the instruction code to specify a memory operand that contains a branch address. In advanced mode the operand is a 32-bit longword operand,

providing a 32-bit branch address. The upper 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.

- Stack Structure

  In advanced mode, when the program counter (PC) is pushed onto the stack in a subroutine call, and the PC, condition-code register (CCR), and extended control register (EXR) are pushed onto the stack in exception handling, they are stored as shown in figure 2.4. When EXR is invalid, it is not pushed onto the stack. For details, see section 4, Exception Handling.

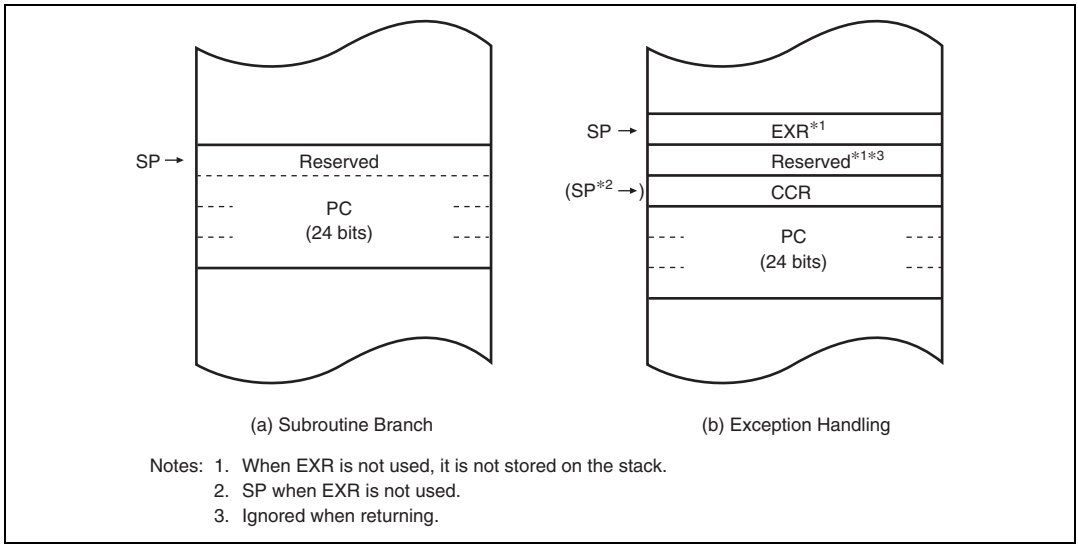


Figure 2.4 Stack Structure in Advanced Mode

2.3 Address Space

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

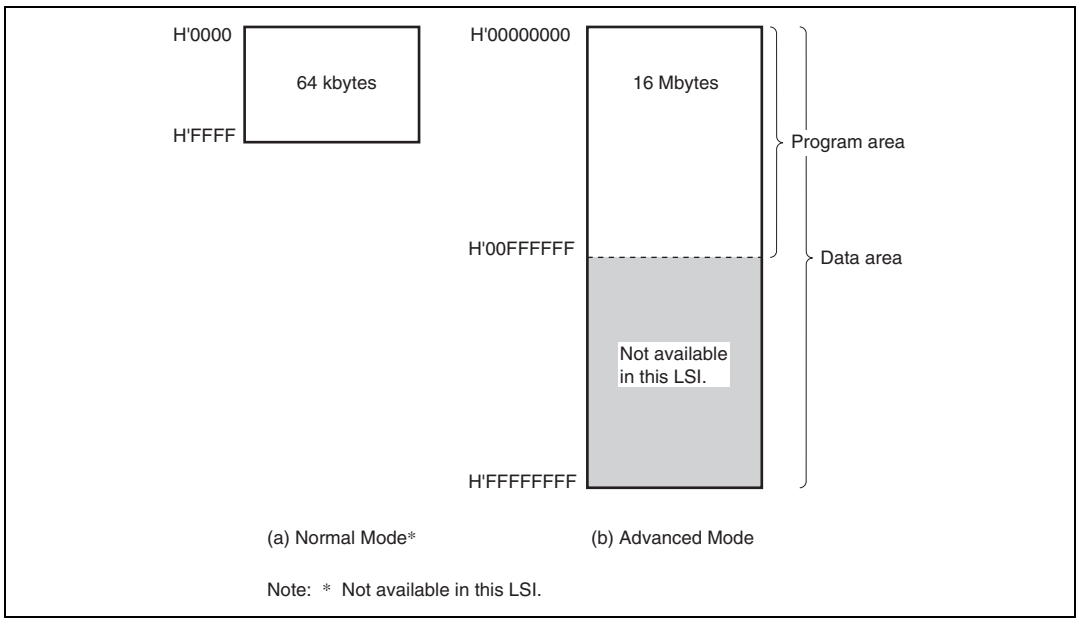


Figure 2.5 Memory Map

2.4 Register Configuration

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

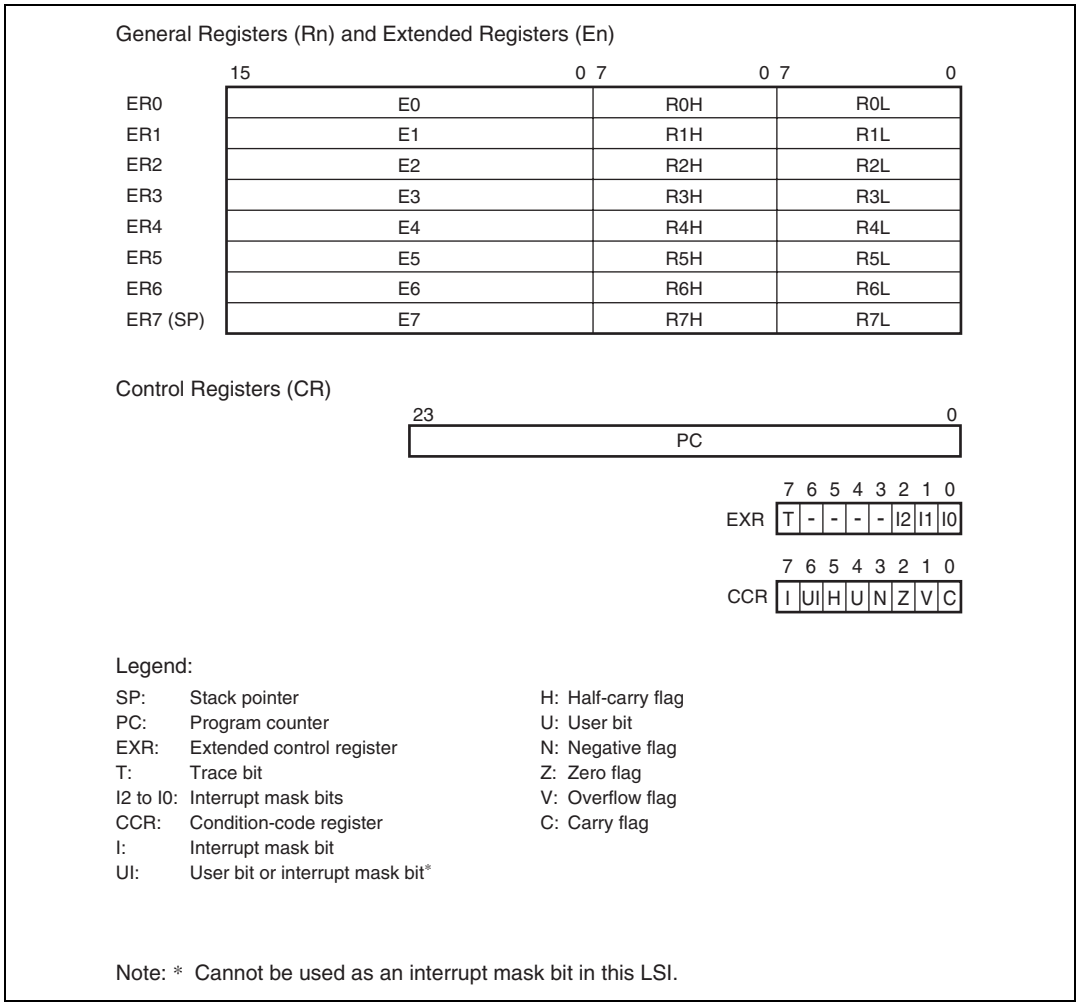


Figure 2.6 CPU Registers

2.4.1 General Registers

The H8S/2000 CPU has eight 32-bit general registers. These general registers are all functionally alike and can be used as both address registers and data registers. When a general register is used as a data register, it can be accessed as a 32-bit, 16-bit, or 8-bit register. Figure 2.7 illustrates the usage of the general registers. When the general registers are used as 32-bit registers or address registers, they are designated by the letters ER (ER0 to ER7).

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.

The usage of each register can be selected independently.

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

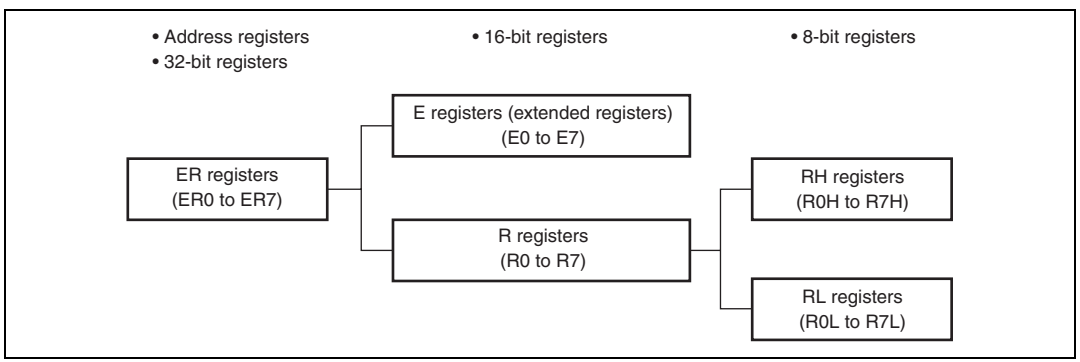


Figure 2.7 Usage of General Registers

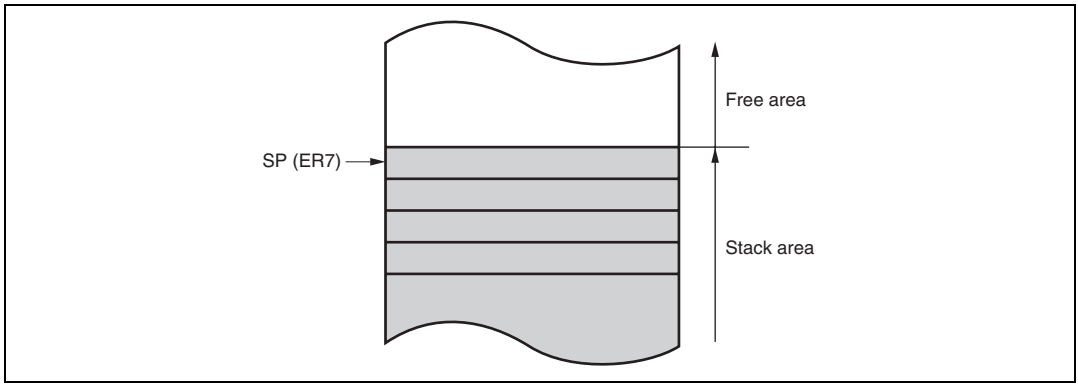


Figure 2.8 Stack

2.4.2 Program Counter (PC)

This 24-bit counter indicates the address of the next instruction the CPU will execute. The length of all CPU instructions is two 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).

2.4.3 Extended Control Register (EXR)

EXR is an 8-bit register that manipulates the LDC, STC, ANDC, ORC, and XORC instructions. When these instructions except for the STC instruction is executed, all interrupts including NMI will be masked for three states after execution is completed.

Bit	Bit Name	Initial Value	R/W	Description
7	T	0	R/W	Trace Bit When this bit is set to 1, a trace exception is generated each time an instruction is executed. When this bit is cleared to 0, instructions are executed in sequence.
6 to 3	—	All 1	—	Reserved These bits are always read as 1.
2 to 0	I2 I1 I0	1	R/W	These bits designate the interrupt mask level (0 to 7). For details, refer to section 5, Interrupt Controller.

2.4.4 Condition-Code Register (CCR)

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

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

Bit	Bit Name	Initial Value	R/W	Description
7	I	1	R/W	Interrupt Mask Bit Masks interrupts other than NMI when set to 1. NMI is accepted regardless of the I bit setting. The I bit is set to 1 by hardware at the start of an exception-handling sequence. For details, refer to section 5, Interrupt Controller.
6	UI	undefined	R/W	User Bit or Interrupt Mask Bit Can be written and read by software using the LDC, STC, ANDC, ORC, and XORC instructions. This bit cannot be used as an interrupt mask bit in this LSI.
5	H	undefined	R/W	Half-Carry Flag When the ADD.B, ADDX.B, SUB.B, SUBX.B, CMP.B or NEG.B instruction is executed, this flag is set to 1 if there is a carry or borrow at bit 3, and cleared to 0 otherwise. When the ADD.W, SUB.W, CMP.W, or NEG.W instruction is executed, the H flag is set to 1 if there is a carry or borrow at bit 11, and cleared to 0 otherwise. When the ADD.L, SUB.L, CMP.L, or NEG.L instruction is executed, the H flag is set to 1 if there is a carry or borrow at bit 27, and cleared to 0 otherwise.
4	U	undefined	R/W	User Bit Can be written and read by software using the LDC, STC, ANDC, ORC, and XORC instructions.
3	N	undefined	R/W	Negative Flag Stores the value of the most significant bit of data as a sign bit.
2	Z	undefined	R/W	Zero Flag Set to 1 to indicate zero data, and cleared to 0 to indicate non-zero data.

Bit	Bit Name	Initial Value	R/W	Description
1	V	undefined	R/W	Overflow Flag Set to 1 when an arithmetic overflow occurs, and cleared to 0 at other times.
0	C	undefined	R/W	Carry Flag Set to 1 when a carry occurs, and cleared to 0 otherwise. Used by: • Add instructions, to indicate a carry • Subtract instructions, to indicate a carry • Shift and rotate instructions, to indicate a carry They carry flag is also used as a bit accumulator by bit manipulation instructions.

2.4.5 Initial Register Values

Reset exception handling loads the CPU's program counter (PC) from the vector table, clears the trace 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 H8S/2000 CPU can process 1-bit, 4-bit (BCD), 8-bit (byte), 16-bit (word), and 32-bit (longword) data. Bit-manipulation instructions operate on 1-bit data by accessing bit n (n = 0, 1, 2, …, 7) of byte operand data. The DAA and DAS decimal-adjust instructions treat byte data as two digits of 4-bit BCD data.

2.5.1 General Register Data Formats

Figure 2.9 shows the data formats in general registers.

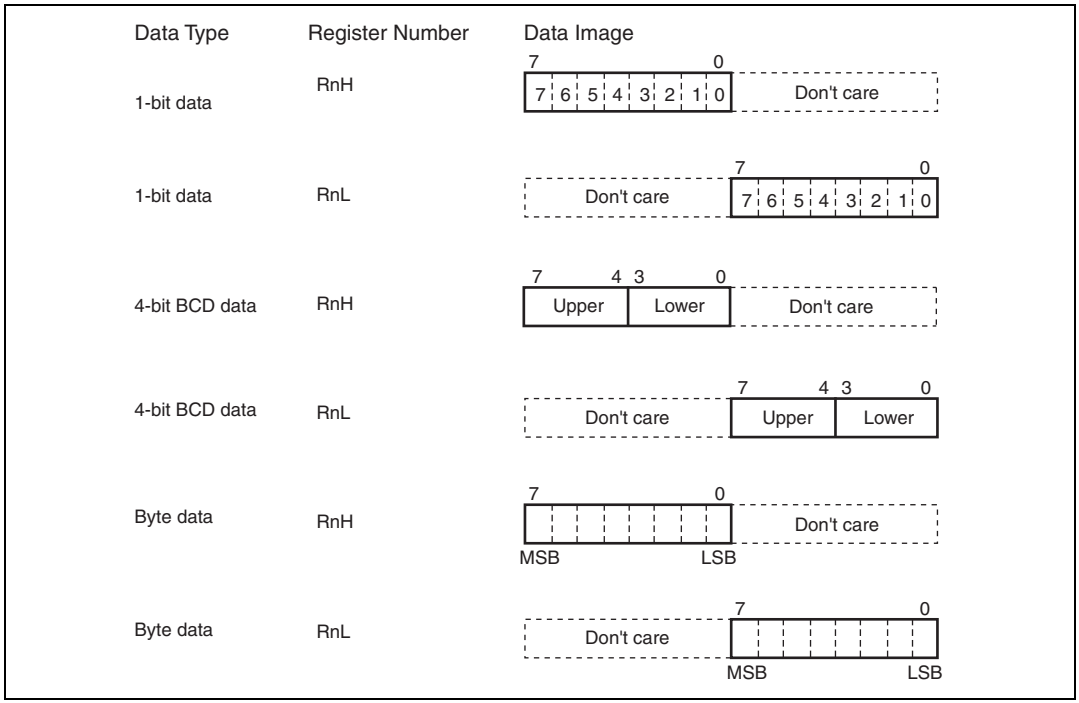


Figure 2.9 General Register Data Formats (1)

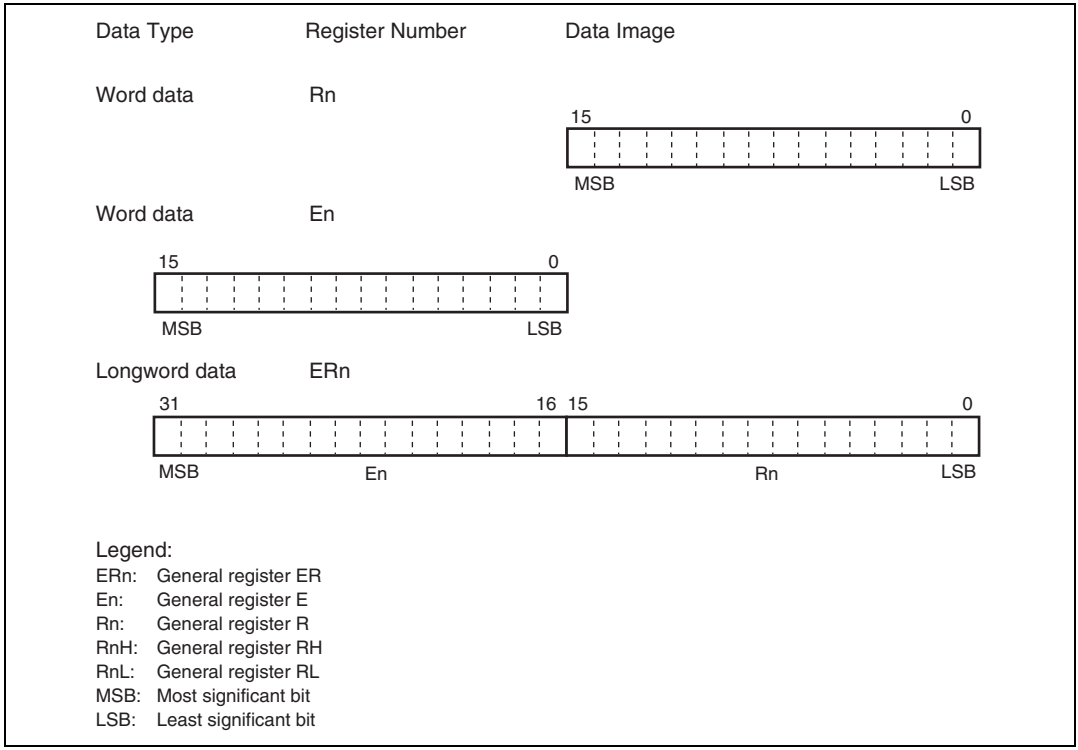


Figure 2.9 General Register Data Formats (2)

2.5.2 Memory Data Formats

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

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

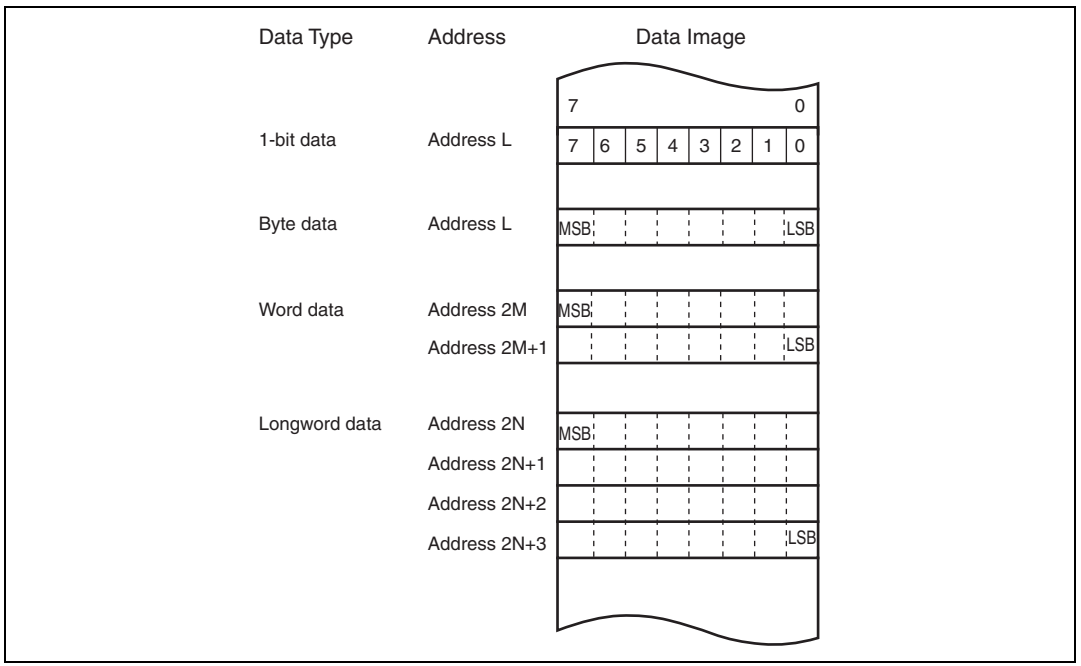


Figure 2.10 Memory Data Formats

2.6 Instruction Set

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

Table 2.1 Instruction Classification

Function	Instructions	Size	Types
Data transfer	MOV	B/W/L	5
	POP*1, PUSH*1	W/L	
	LDM*5, STM*5	L	
	MOVFPE*3, MOVTPE*3	B	
Arithmetic operations	ADD, SUB, CMP, NEG	B/W/L	19
	ADDX, SUBX, DAA, DAS	B	
	INC, DEC	B/W/L	
	ADDS, SUBS	L	
	MULXU, DIVXU, MULXS, DIVXS	B/W	
	EXTU, EXTS	W/L	
	TAS*4	B	
Logic operations	AND, OR, XOR, NOT	B/W/L	4
Shift	SHAL, SHAR, SHLL, SHLR, ROTL, ROTR, ROTXL, ROTXR	B/W/L	8
Bit manipulation	BSET, BCLR, BNOT, BTST, BLD, BILD, BST, BIST, BAND, BIAND, BOR, BIOR, BXOR, BIXOR	B	14
Branch	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 this LSI.
4. Only register ER0, ER1, ER4, or ER5 should be used when using the TAS instruction.
5. The ER7 register functions as a stack pointer for the LDM and STM instructions, so it cannot be for saving (STM) or restoring (LDM) data.

2.6.1 Table of Instructions Classified by Function

Tables 2.3 to 2.10 summarizes the instructions in each functional category. The notation used in tables 2.3 to 2.10 is defined below.

Table 2.2 Operation Notation

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

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

Table 2.3 Data Transfer Instructions

Instruction	Size*1	Function
MOV	B/W/L	(EAs) → Rd, Rs → (EAd) Moves data between two general registers or between a general register and memory, or moves immediate data to a general register.
MOVFPE	B	Cannot be used in this LSI.
MOVTPE	B	Cannot be used in this LSI.
POP	W/L	@SP+ → Rn Pops a general register from the stack. POP.W Rn is identical to MOV.W @SP+, Rn. POP.L ERn is identical to MOV.L @SP+, ERn
PUSH	W/L	Rn → @-SP Pushes a general register onto the stack. PUSH.W Rn is identical to MOV.W Rn, @-SP. PUSH.L ERn is identical to MOV.L ERn, @-SP.
LDM*2	L	@SP+ → Rn (register list) Pops two or more general registers from the stack.
STM*2	L	Rn (register list) → @-SP Pushes two or more general registers onto the stack.

Notes: 1. Size refers to the operand size.
B: Byte
W: Word
L: Longword
2. ER7 is used as a stack pointer in STM and LDM instructions. ER7, therefore, should not be used as a saving (STM) or restoring (LDM) register.

Table 2.4 Arithmetic Operations Instructions (1)

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

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

Table 2.4 Arithmetic Operations Instructions (2)

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

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

Table 2.5 Logic Operations Instructions

Instruction	Size*	Function
AND	B/W/L	Rd ∧ Rs → Rd, Rd ∧ #IMM → Rd Performs a logical AND operation on a general register and another general register or immediate data.
OR	B/W/L	Rd ∨ Rs → Rd, Rd ∨ #IMM → Rd Performs a logical OR operation on a general register and another general register or immediate data.
XOR	B/W/L	Rd ⊕ Rs → Rd, Rd ⊕ #IMM → Rd Performs a logical exclusive OR operation on a general register and another general register or immediate data.
NOT	B/W/L	~ Rd → Rd Takes the one's complement (logical complement) of general register contents.

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

Table 2.6 Shift Instructions

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

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

Table 2.7 Bit Manipulation Instructions (1)

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

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

Table 2.7 Bit Manipulation Instructions (2)

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

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

Table 2.8 Branch Instructions

Instruction	Size	Function
Bcc	—	Branches to a specified address if a specified condition is true. The branching conditions are listed below.

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

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

Table 2.9 System Control Instruction

Instruction	Size*	Function
TRAPA	—	Starts trap-instruction exception handling.
RTE	—	Returns from an exception-handling routine.
SLEEP	—	Causes a transition to a power-down state.
LDC	B/W	(EAs) → CCR, (EAs) → EXR Moves the 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.

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

Table 2.10 Block Data Transfer Instruction

Instruction	Size	Function
EEPMOV.B	—	if R4L ≠ 0 then Repeat @ER5+ → @ER6+ R4L–1 → R4L Until R4L = 0 else next;
EEPMOV.W	—	if R4 ≠ 0 then Repeat @ER5+ → @ER6+ R4–1 → R4 Until R4 = 0 else next;
		Transfer a data block. Starting from the address set in ER5, transfers data for the number of bytes set in R4L or R4 to the address location set in ER6. Execution of the next instruction begins as soon as the transfer is completed.

2.6.2 Basic Instruction Formats

The H8S/2000 CPU instructions consist of 2-byte (1-word) units. An instruction consists of an operation field (op), a register field (r), an effective address extension (EA), and a condition field (cc).

Figure 2.11 shows examples of instruction formats.

- Operation Field

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

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

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

  Specifies the branching condition of Bcc instructions.

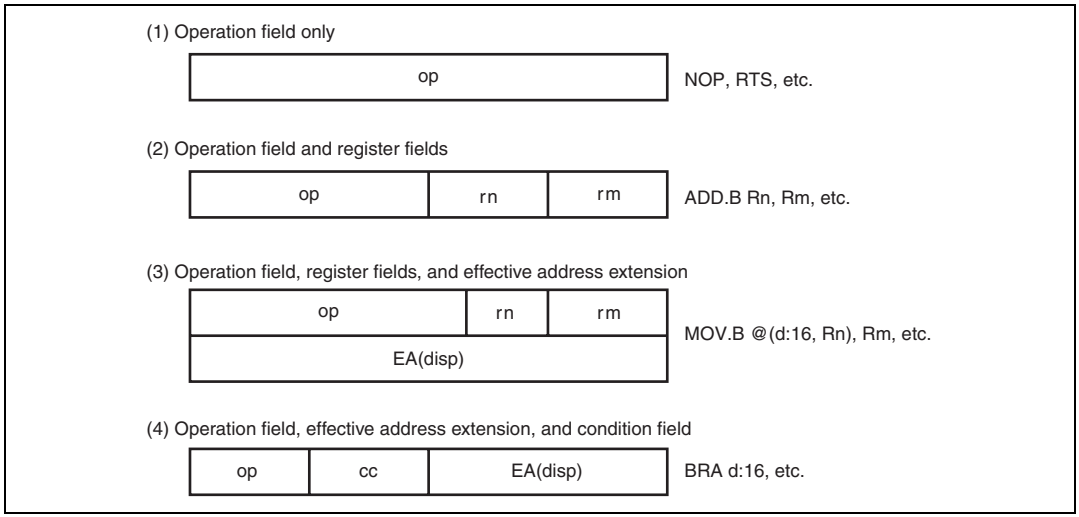


Figure 2.11 Instruction Formats (Examples)

2.7 Addressing Modes and Effective Address Calculation

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

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

2.7.1 Register Direct—Rn

The register field of the instruction 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.

2.7.2 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).

2.7.3 Register Indirect with Displacement—@(d:16, ERn) or @(d:32, ERn)

A 16-bit or 32-bit displacement contained in the instruction is added to an address register (ERn) specified by the register field of the instruction, and the sum gives the address of a memory operand. A 16-bit displacement is sign-extended when added.

2.7.4 Register Indirect with Post-Increment or Pre-Decrement—@ERn+ or @-ERn

Register indirect with post-increment—@ERn+: The register field of the instruction code specifies an address register (ERn) which contains the address of a memory operand. After the operand is accessed, 1, 2, or 4 is added to the address register contents and the sum is stored in the address register. The value added is 1 for byte access, 2 for word 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.

2.7.5 Absolute Address—@aa:8, @aa:16, @aa:24, or @aa:32

The instruction code contains the absolute address of a memory operand. The absolute address may be 8 bits long (@aa:8), 16 bits long (@aa:16), 24 bits long (@aa:24), or 32 bits long (@aa:32). Table 2.12 indicates the accessible absolute address ranges.

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

For a 16-bit absolute address the upper 16 bits are a sign extension. 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 2.12 Absolute Address Access Ranges

Absolute Address		Normal Mode*	Advanced Mode
Data address	8 bits (@aa:8)	H'FF00 to H'FFFF	H'FFFF00 to H'FFFFFF
	16 bits (@aa:16)	H'0000 to H'FFFF	H'000000 to H'007FFF, H'FF8000 to H'FFFFFF
	32 bits (@aa:32)		H'000000 to H'FFFFFF
Program instruction address	24 bits (@aa:24)		

Note: * Not available in this LSI.

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

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

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

2.7.7 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.

2.7.8 Memory Indirect—@@aa:8

This mode can be used by the JMP and JSR instructions. The instruction code contains an 8-bit absolute address specifying a memory operand. 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'0000 to H'00FF in normal mode*, H'000000 to H'0000FF in advanced mode). In normal mode the memory operand is a word operand and the branch address is 16 bits long. In advanced mode the memory operand is a longword operand, the first byte of which is assumed to be 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.

If an odd address is specified in word or longword memory access, or as a branch address, the least significant bit is regarded as 0, causing data to be accessed or instruction code to be fetched at the address preceding the specified address. (For further information, see section 2.5.2, Memory Data Formats.)

Note: * Not available in this LSI.

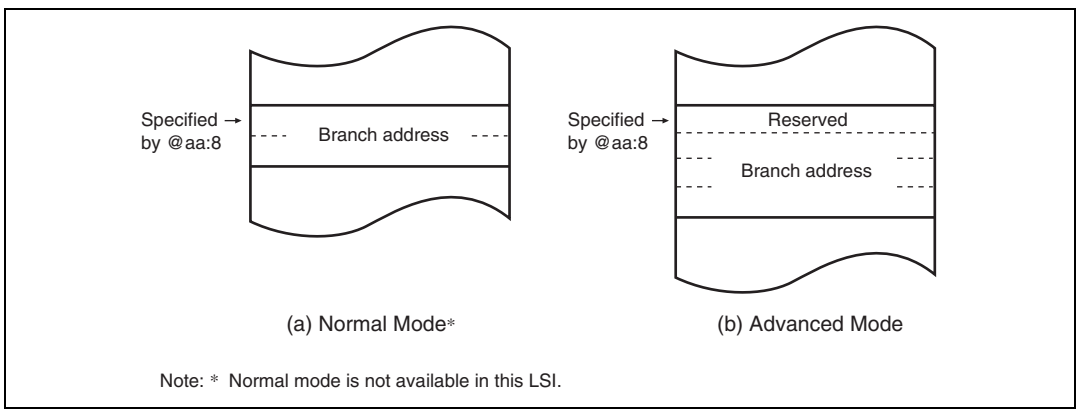


Figure 2.12 Branch Address Specification in Memory Indirect Mode

2.7.9 Effective Address Calculation

Table 2.13 indicates how effective addresses are calculated in each addressing mode. In normal mode the upper 8 bits of the effective address are ignored in order to generate a 16-bit address.

Table 2.13 Effective Address Calculation (1)

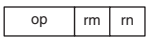
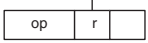
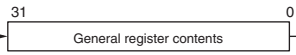
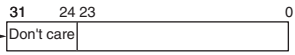
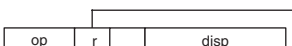
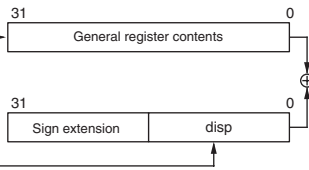
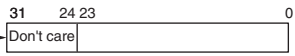
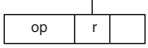
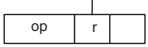
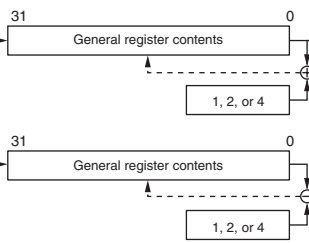
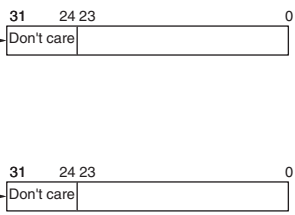
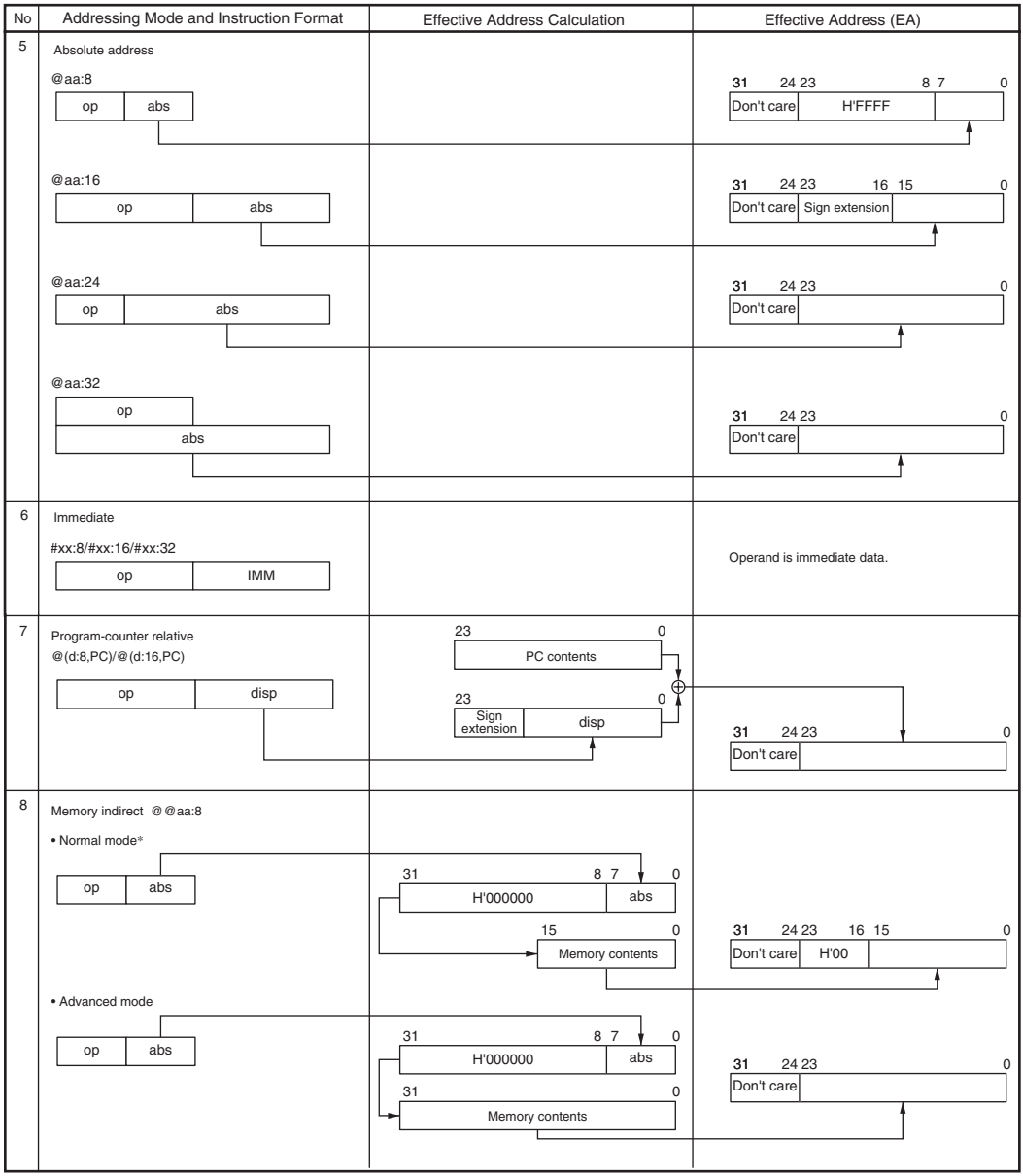
No	Addressing Mode and Instruction Format	Effective Address Calculation	Effective Address (EA)
1	Register direct (Rn) 		Operand is general register contents.
2	Register indirect (@ERn) 		
3	Register indirect with displacement @(d:16,ERn) or @(d:32,ERn) 		
4	Register indirect with post-increment or pre-decrement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

Table 2.13 Effective Address Calculation (2)



Note: * Normal mode is not available in this LSI.

2.8 Processing States

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

- Reset State

  In this state the CPU and internal peripheral modules are all initialized and stop. When the $\overline{\text{RES}}$ input goes low all current processing stops and the CPU enters the reset state. All interrupts are masked in the reset state. Reset exception handling starts when the $\overline{\text{RES}}$ signal changes from low to high. For details, refer to section 4, Exception Handling.

  The reset state can also be entered by a watchdog timer overflow.

- Exception-Handling State

  The exception-handling state is a transient state that occurs when the CPU alters the normal processing flow due to an exception source, such as, a reset, trace, interrupt, or trap instruction. The CPU fetches a start address (vector) from the exception vector table and branches to that address. For further details, refer to section 4, Exception Handling.

- Program Execution State

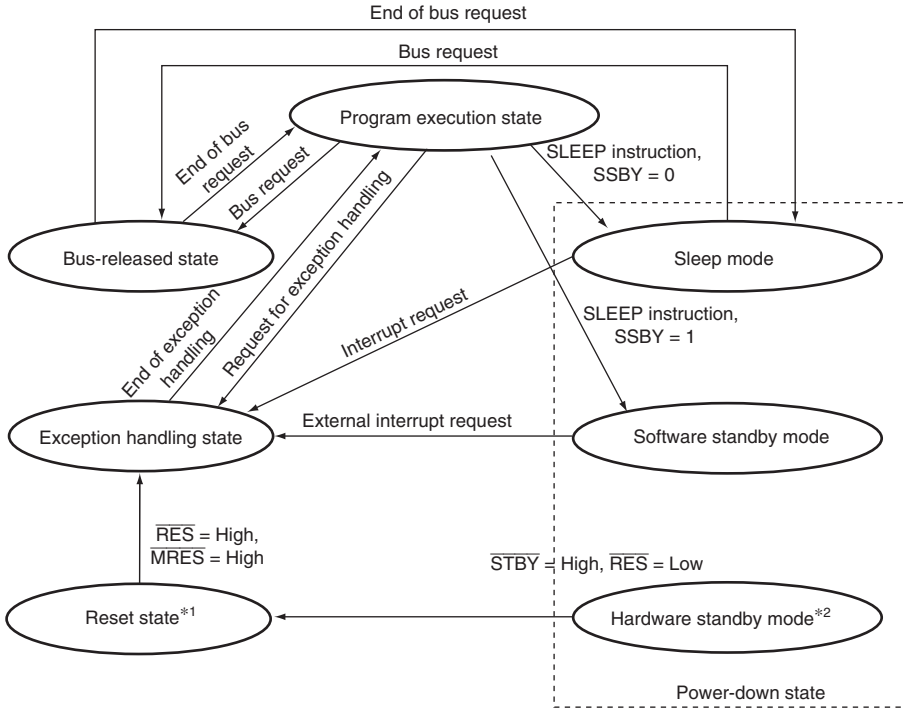
  In this state the CPU executes program instructions in sequence.

- Bus-Released State

  In a product which has a bus master other than the CPU, such as a direct memory access controller (DMAC) and a data transfer controller (DTC), the bus-released state occurs when the bus has been released in response to a bus request from a bus master other than the CPU. While the bus is released, the CPU halts operations.

- Power-Down State

  This is a power-down state in which the CPU stops operating. The program stop state occurs when a SLEEP instruction is executed or the CPU enters hardware standby mode. For details, refer to section 22, Power-Down Modes.



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

Figure 2.13 State Transitions

2.9 Usage Notes

2.9.1 Note on TAS Instruction Usage

Only register ER0, ER1, ER4, or ER5 should be used when using the TAS instruction. The TAS instruction is not generated by the Renesas Technology 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.

2.9.2 STM/LTM Instruction Usage

With the STM or LDM instruction, the ER7 register is used as the stack pointer, and thus cannot be used as a register that allows save (STM) or restore (LDM) operation.

With a single STM or LDM instruction, two to four registers can be saved or restored. The available registers are as follows:

For two registers: ER0 and ER1, ER2 and ER3, or ER4 and ER5

For three registers: ER0 to ER2, or ER4 to ER6

For four registers: ER0 to ER3

For the Renesas Technology H8S or H8/300 Series C/C++ Compiler, the STM/LDM instruction including ER7 is not created.

2.9.3 Note on Bit Manipulation Instructions

Using bit manipulation instructions on registers containing write-only bits can result in the bits that should have been manipulated not being manipulated as intended or in the wrong bits being manipulated.

Reading data from a register containing write-only bits may return fixed or undefined values. Consequently, bit manipulation instructions that use the read values to perform operations (BNOT, BTST, BAND, BIAND, BOR, BIOR, BXOR, BIXOR, BLD, and BILD) will not work properly.

In addition, bit manipulation instructions that write data following operations based on the data values read (BSET, BCLR, BNOT, BST, and BIST) may change the values of bits unrelated to the intended bit manipulation. Therefore, caution is necessary when using bit manipulation instructions on registers containing write-only bits.

The instructions BSET, BCLR, BNOT, BST, and BIST perform the following operations in the order shown:

1. Read data in byte units
2. Perform bit manipulation on the read data according to the instruction
3. Write data in byte units

Example: Using the BCLR instruction to clear pin 14 only of P1DDR for port 1

P1DDR is an 8-bit register that contains write-only bits. It is used to specify the I/O setting of the individual pins in port 1. Reading produces invalid data. Attempting to read from P1DDR returns undefined values.

In this example, the BCLR instruction is used to set pin 14 as an input port. Let us assume that pins 17 to 14 are presently set as output pins and pins 13 to 10 are set as input pins. Thus, the value of P1DDR is initially H'F0.

	P17	**P16**	**P15**	**P14**	**P13**	**P12**	**P11**	**P10**
I/O	Output	Output	Output	Output	Input	Input	Input	Input
P1DDR	1	1	1	1	0	0	0	0

To change pin 14 from an output pin to an input pin, the value of bit 4 in P1DDR must be changed from 1 to 0 (H'F0 to H'E0). Now assume that the BCLR instruction is used to clear bit 4 in P1DDR to 0.

```
BCLR    #4, @P1DDR
```

However, using the above bit manipulation instruction on the write-only register P1DDR can cause problems, as described below.

The BCLR instruction first reads data from P1DDR in byte units, but in this case the read values are undefined. These undefined values can be 0 or 1 for each bit in the register, but there is no way of telling which. Since all of the bits in P1DDR are write-only, undefined values are returned for all of the bits when the register is read. In this example the value of P1DDR is H'F0, but we will assume that the value returned when the register was read is H'F8, which would give bit 3 a value of 1.

	P17	P16	P15	P14	P13	P12	P11	P10
I/O	Output	Output	Output	Output	Input	Input	Input	Input
P1DDR	1	1	1	1	0	0	0	0
Read value	1	1	1	1	1	0	0	0

The BCLR instruction performs bit manipulation on the read value, which is H'F8 in this example. It clears bit 4 to 0.

	P17	P16	P15	P14	P13	P12	P11	P10
I/O	Output	Output	Output	Output	Input	Input	Input	Input
P1DDR	1	1	1	1	0	0	0	0
After bit manipulation	1	1	1	0	1	0	0	0

Following bit manipulation the data is written to P1DDR and the BCLR instruction terminates.

	P17	P16	P15	P14	P13	P12	P11	P10
I/O	Output	Output	Output	Input	Output	Input	Input	Input
P1DDR	1	1	1	0	1	0	0	0
Write value	1	1	1	0	1	0	0	0

The contents of P1DDR should have been overwritten with a value of H'E0, but in fact a value of H'E8 was written to the register. This changed pin 13, which should have been an input pin, to an output pin. In this example we assumed that pin 13 was read as 1. However, since the values returned for pins 17 to 10 are all undefined when read, there is the possibility that individual bit values could be changed from 0 to 1 or from 1 to 0. To prevent this from happening, the recommendations in section 2.9.4, Accessing Registers Containing Write-Only Bits, should be followed when changing the values of registers containing write-only bits.

In addition, the BCLR instruction can be used to clear flags in internal I/O registers to 0. In such cases it is not necessary to read the relevant flag beforehand so long as it is clear that it has been set to 1 by an interrupt processing routine or the like.

2.9.4 Accessing Registers Containing Write-Only Bits

Using data transfer instructions or bit manipulation instructions on registers containing write-only bits can result in undefined values being read. To prevent the reading of undefined values, the procedure described below should be used to access registers containing write-only bits.

In order to write to a register containing write-only bits, set aside a work area in memory (in on-chip RAM, for example) and write the data to be manipulated to it. After accessing and manipulating the data in the work area in memory, write the resulting data to the register containing write-only bits.

Figure 2.14 Example Flowchart of Method for Accessing Registers Containing Write-Only Bits

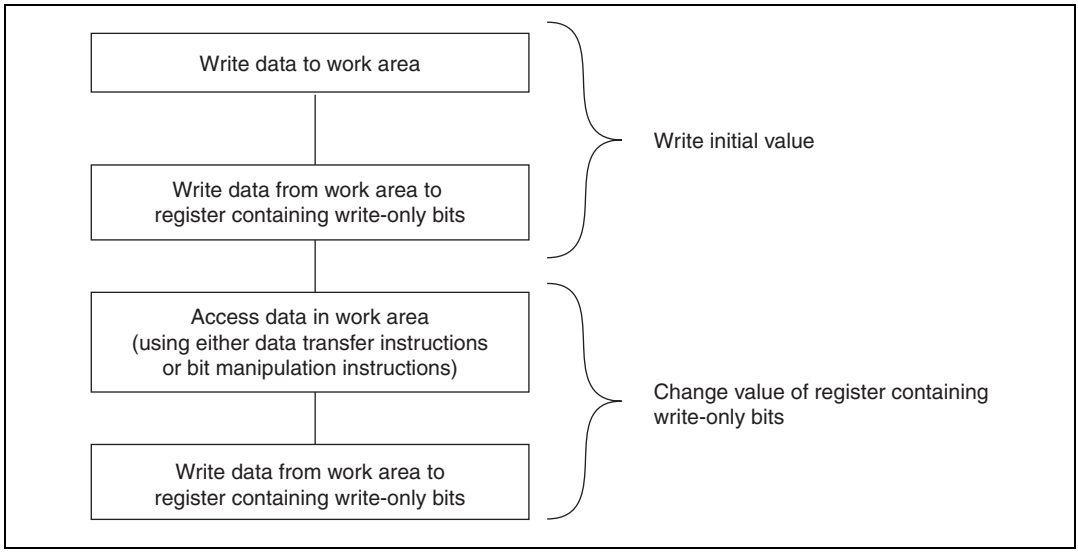


Figure 2.14 Flowchart of Method for Accessing Registers Containing Write-Only Bits

Example: Clearing pin 14 only of P1DDR for port 1

P1DDR is an 8-bit register that contains write-only bits. It is used to specify the I/O setting of the individual pins in port 1. Reading produces invalid data. Attempting to read from P1DDR returns undefined values.

In this example, the BCLR instruction is used to set pin 14 as an input port. To start, the initial value H'F0 to be written to P1DDR is written ahead of time to the work area (RAM0) in memory.

```
MOV.B    #H'F0,  R0L
MOV.B    R0L,    @RAM0
MOV.B    R0L,    @P1DDR
```

	P17	P16	P15	P14	P13	P12	P11	P10
I/O	Output	Output	Output	Output	Input	Input	Input	Input
P1DDR	1	1	1	1	0	0	0	0
RAM0	1	1	1	1	0	0	0	0

To change pin 14 from an output pin to an input pin, the value of bit 4 in P1DDR must be changed from 1 to 0 (H'F0 to H'E0). Here the BCLR instruction will be used to clear bit 4 in P1DDR to 0.

```
BCLR    #4,     @RAM0
```

	P17	P16	P15	P14	P13	P12	P11	P10
I/O	Output	Output	Output	Output	Input	Input	Input	Input
P1DDR	1	1	1	1	0	0	0	0
RAM0	1	1	1	0	0	0	0	0

Since RAM0 is a read/write area of memory, performing the above bit manipulation using the BCLR instruction causes only bit 4 in RAM0 to be cleared to 0. The value of RAM0 is then written to P1DDR.

```
MOV.B   @RAM0,  R0L
MOV.B   R0L,    @P1DDR
```

	P17	P16	P15	P14	P13	P12	P11	P10
I/O	Output	Output	Output	Input	Input	Input	Input	Input
P1DDR	1	1	1	0	0	0	0	0
RAM0	1	1	1	0	0	0	0	0

By using the above procedure to access registers containing write-only bits, it is possible to create programs that are not dependent on the type of instructions used.

Section 3 MCU Operating Modes

3.1 Operating Mode Selection

This LSI supports four operating modes (modes 7 to 4). These modes are depending on the setting of mode pins (MD2 to MD0). Modes 6 to 4 are extended modes in which external memory and external peripheral devices can be accessed. In extended modes, each area can be used as 8-bit or 16-bit address space according to the bus controller settings after program execution. In this case, if an area is specified as 16-bit access space, 16-bit bus mode is employed for all areas; while if an area is specified as 8-bit access space, 8-bit bus mode is employed for all areas. In mode 7, external addresses cannot be used. Do not change the mode pin settings during operation.

Table 3.1 MCU Operating Mode Selection

MCU Operating Mode	MD2	MD1	MD0	CPU Operating Mode	Description	On-Chip ROM	External Data Bus	
							Initial Value	Maximum Value
4	1	0	0	Advanced mode	On-chip ROM disabled, extended mode	Disabled	16 bits	16 bits
5	1	0	1	Advanced mode	On-chip ROM disabled, extended mode	Disabled	8 bits	16 bits
6	1	1	0	Advanced mode	On-chip ROM enabled, extended mode	Enabled	8 bits	16 bits
7*	1	1	1	Advanced mode	Single-chip mode	Enabled	—	—

Note: * The following applies to the use of mode 7.
(1) H8S/2215
The USB cannot be used in mode 7.
(2) H8S/2215R or H8S/2215T
Development work using the E6000 emulator:
The USB cannot be used in mode 7.
Development work using the on-chip (E10A-USB) emulator:
The USB can be used in mode 7.
See section 3.3.4, Mode 7, for details.

3.2 Register Descriptions

The following registers are related to the operating mode.

- Mode control register (MDCR)
- System control register (SYSCR)

3.2.1 Mode Control Register (MDCR)

MDCR is used to monitor the current operating mode of this LSI.

Bit	Bit Name	Initial Value	R/W	Description
7	—	1	—	Reserved This bit is always read as 1 and cannot be modified.
6 to 3	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
2	MDS2	—*	R	Mode select 2 to 0 These bits indicate the input levels at pins MD2 to MD0 (the current operating mode).Bits MDS2 to MDS0 correspond to MD2 to MD0. MDS2 to MDS0 are read-only bits and they cannot be written to. The mode pin (MD2 to MD0) input levels are latched into these bits when MDCR is read. These latches are canceled by a power-on reset, but maintained at manual reset.
1	MDS1	—*	R	
0	MDS0	—*	R	

Note: * Determined by the MD2 to MD0 pin settings.

3.2.2 System Control Register (SYSCR)

SYSCR is used to select the interrupt control mode and the detected edge for NMI, select the $\overline{\text{MRES}}$ input pin enable or disable, and enables or disables on-chip RAM.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R/W	Reserved The write value should always be 0.
6	—	0	—	Reserved These bits are always read as 0 and cannot be modified.
5	INTM1	0	R/W	These bits select the control mode of the interrupt controller. For details of the interrupt control modes, see section 5.6, Interrupt Control Modes and Interrupt Operation. 00: Interrupt control mode 0 01: Setting prohibited 10: Interrupt control mode 2 11: Setting prohibited
4	INTM0	0	R/W	
3	NMIEG	0	R/W	NMI Edge Select Selects the valid edge of the NMI interrupt input. 0: An interrupt is requested at the falling edge of NMI input 1: An interrupt is requested at the rising edge of NMI input
2	MRESE	0	R/W	Manual reset Select Enables or disables the $\overline{\text{MRES}}$ pin input. 0: The $\overline{\text{MRES}}$ pin input (manual reset) is disabled 1: The $\overline{\text{MRES}}$ pin input (manual reset) is enabled The $\overline{\text{MRES}}$ input pin can be used.
1	—	0	—	Reserved These bits are always read as 0 and cannot be modified.
0	RAME	1	R/W	RAM Enable Enables or disables the on-chip RAM. The RAME bit is initialized when the reset status is released. 0: On-chip RAM is disabled 1: On-chip RAM is enabled

3.3 Operating Mode Descriptions

3.3.1 Mode 4

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

Pins P13 to P10, and 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.

Pins P13 to P11 function as input ports immediately after a reset. Pin 10 and ports A and B function as address (A20 to A8) outputs immediately after a reset. Address (A23 to A21) output can be enabled or disabled by bits AE3 to AE0 in the pin function control register (PFCR) regardless of the corresponding data direction register (DDR) values. Pins for which address output is disabled among pins P13 to P10 and in ports A and B become port outputs when the corresponding DDR bits are set to 1.

Port C always has an address (A7 to A0) output function.

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.

3.3.2 Mode 5

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

Pins P13 to P10, and 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.

Pins P13 to P11 function as input ports immediately after a reset. Pin 10 and ports A and B function as address (A20 to A8) outputs immediately after a reset. Address (A23 to A21) output can be enabled or disabled by bits AE3 to AE0 in the pin function control register (PFCR) regardless of the corresponding data direction register (DDR) values. Pins for which address output is disabled among pins P13 to P10 and in ports A and B become port outputs when the corresponding DDR bits are set to 1.

Port C always has an address (A7 to A0) output function.

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

3.3.3 Mode 6

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

Pins P13 to P10, and ports A, B and C function as input ports immediately after a reset. Address (A23 to A8) output can be enabled or disabled by bits AE3 to AE0 in the pin function control register (PFCR) regardless of the corresponding data direction register (DDR) values. Pins for which address output is disabled among pins P13 to P10 and in ports A and B become port outputs when the corresponding DDR bits are set to 1.

Port C is an input port immediately after a reset. Addresses A7 to A0 are output by setting the corresponding DDR bits to 1.

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

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

3.3.4 Mode 7

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

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

In addition, the USB is not supported in some cases due to development tool issues, as summarized in table 3.2.

Table 3.2 USB Support in Mode 7

Development Tool	H8S/2215	H8S/2215R or H8S/2215T
E6000	×	×
E10A-USB	—*	○

Note: * The H8S/2215 does not have an on-chip emulator function.

3.3.5 Pin Functions

The pin functions of ports 1, and A to F vary depending on the operating mode. Table 3.3 shows the functions in modes 4 to 7.

Table 3.3 Pin Functions in Each Operating Mode

Port		Mode 4	Mode 5	Mode 6	Mode 7[*1]
Port 1	P13 to P11	P*/A	P*/A	P*/A	P
	P10	P/A*	P/A*	P*/A	P
Port A	PA3 to PA0	P/A*	P/A*	P*/A	P
Port B		P/A*	P/A*	P*/A	P
Port C		A	A	P*/A	P
Port D		D	D	D	P
Port E		P/D*	P*/D	P*/D	P
Port F	PF7	P/C*	P/C*	P/C*	P*/C
	PF6 to PF4	C	C	C	P
	PF3	P/C*	P*/C	P*/C	
	PF2 to PF0	P*/C	P*/C	P*/C	

Legend:

P: I/O port

A: Address bus output

D: Data bus I/O

C: Control signals, clock I/O

*: After reset

Note: 1. The following applies to the use of mode 7.

(1) H8S/2215

The USB cannot be used in mode 7.

(2) H8S/2215R or H8S/2215T

Development work using the E6000 emulator:

The USB cannot be used in mode 7.

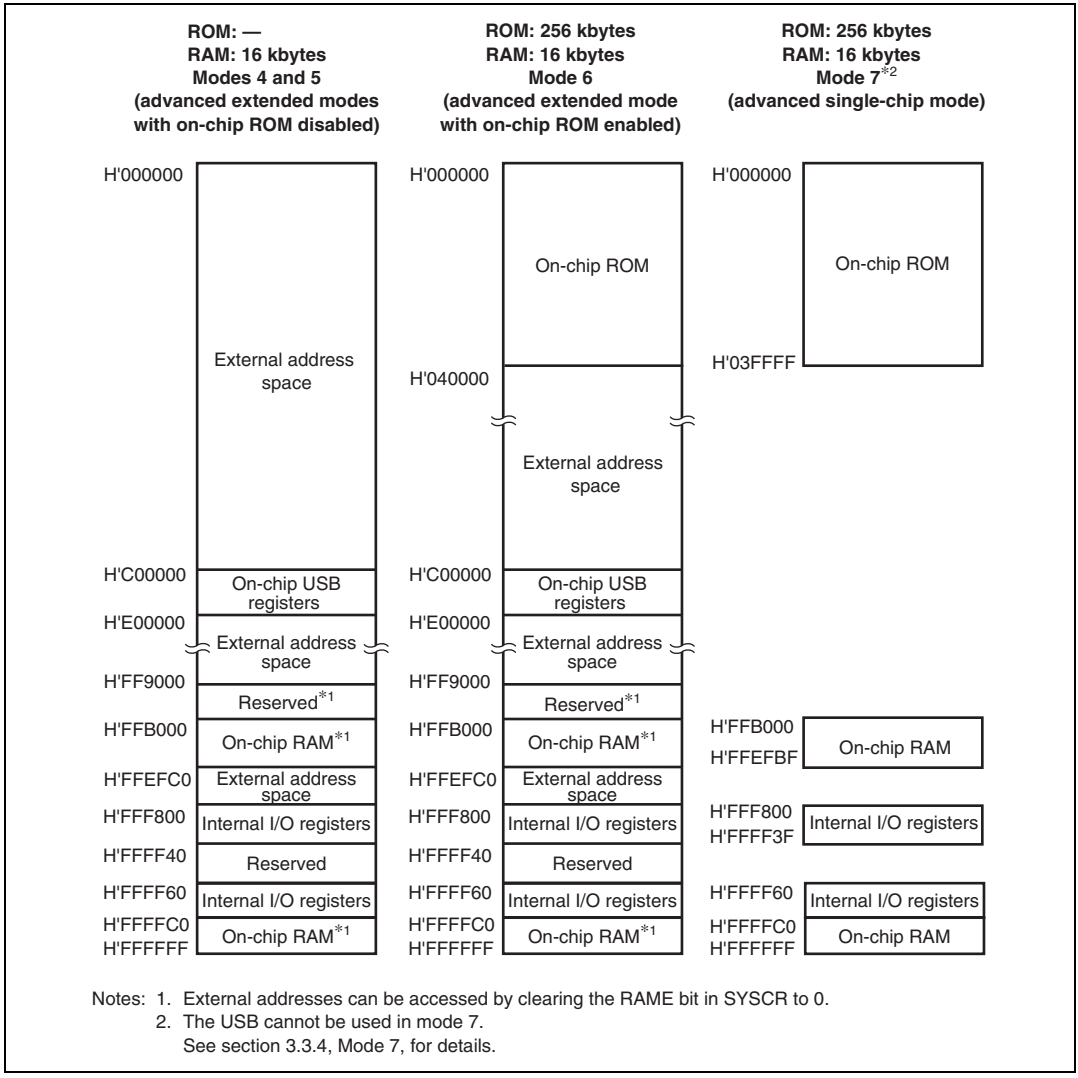
Development work using the on-chip (E10A-USB) emulator:

The USB can be used in mode 7.

See section 3.3.4, Mode 7, for details.

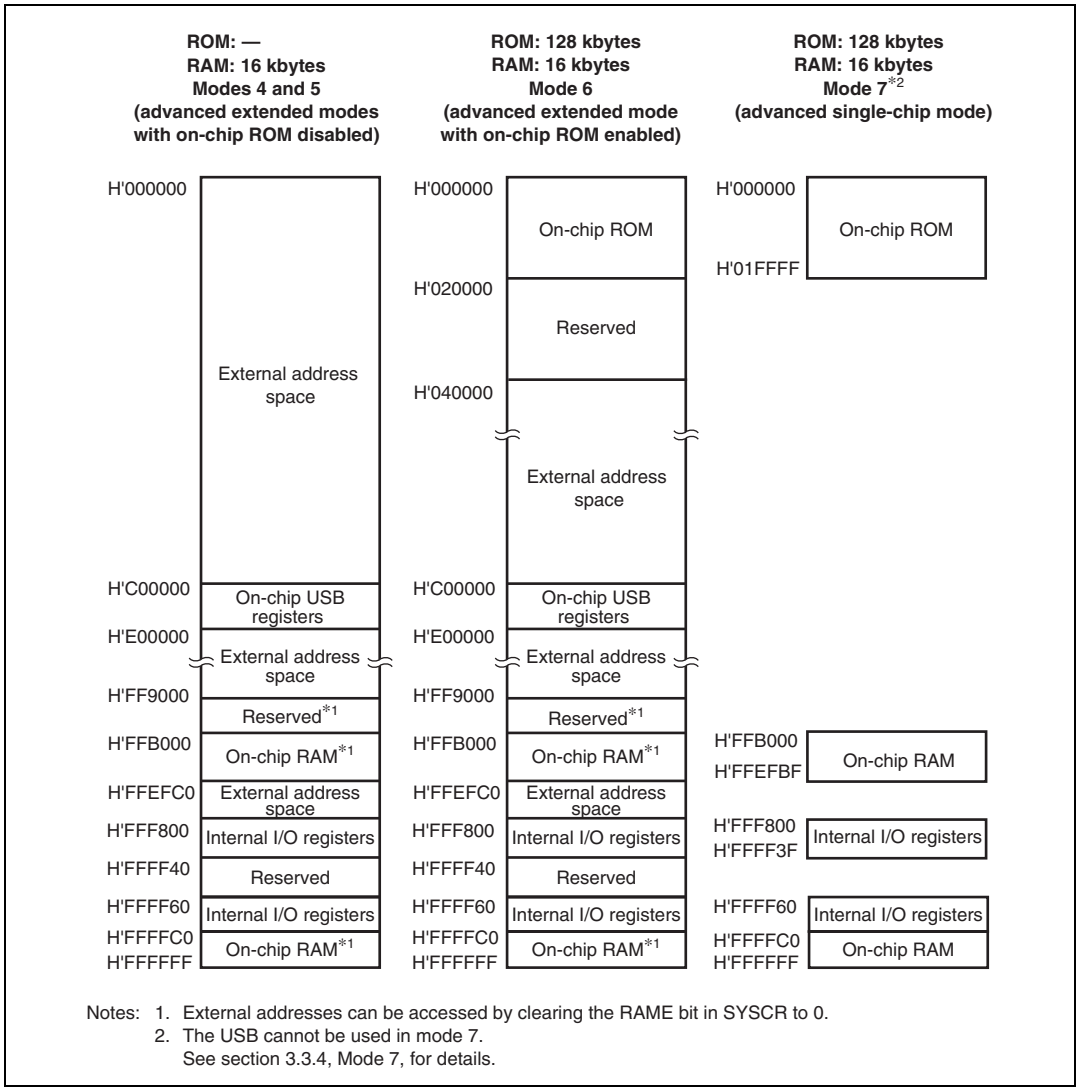
3.4 Memory Map in Each Operating Mode

Figures 3.1 to 3.4 show the memory map in each operating mode for HD64F2215, HD64F2215U, HD6432215B, and HD6432215C.



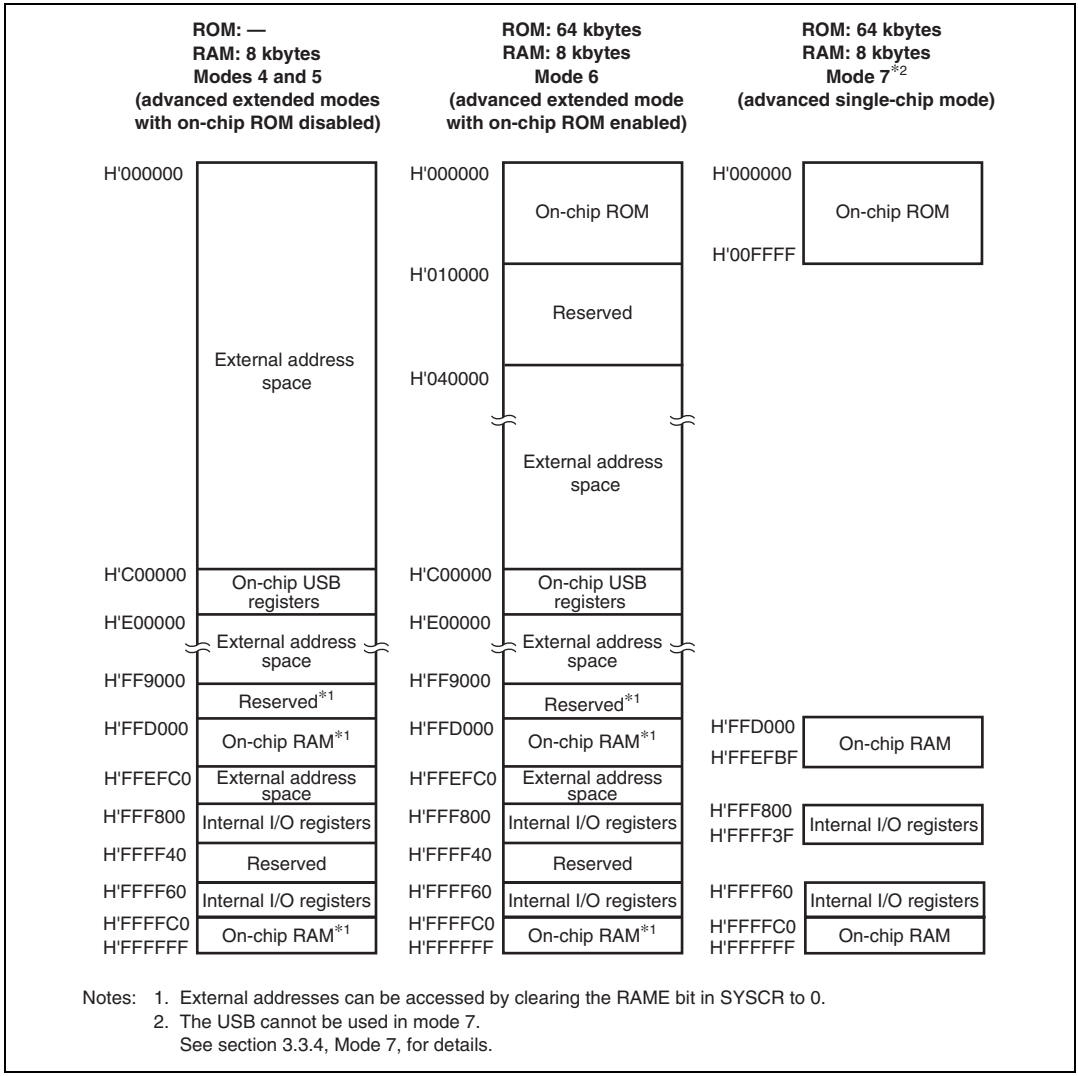
Notes: 1. External addresses can be accessed by clearing the RAME bit in SYSCR to 0.
2. The USB cannot be used in mode 7.
See section 3.3.4, Mode 7, for details.

Figure 3.1 Memory Map in Each Operating Mode for HD64F2215 and HD64F2215U



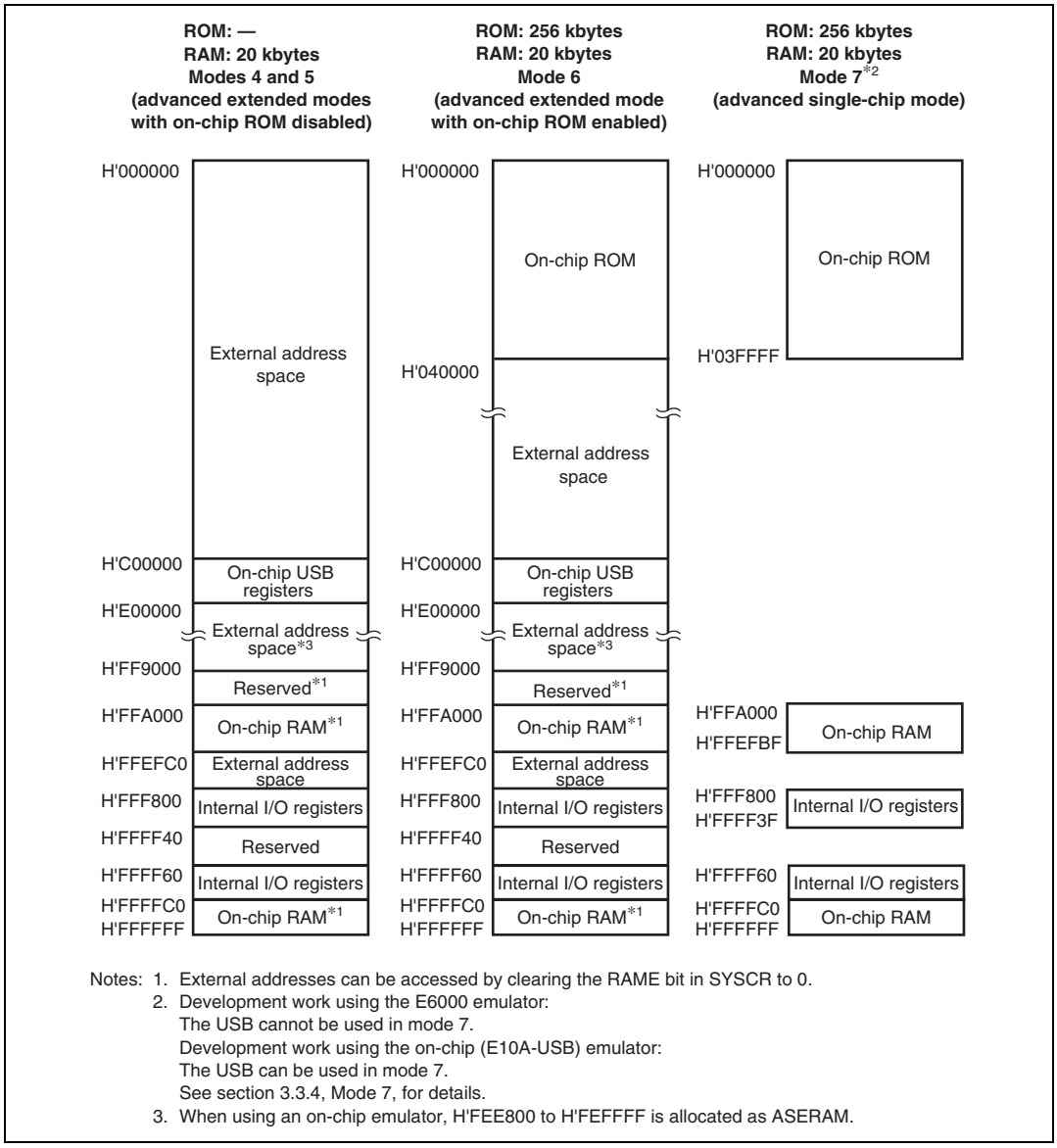
Notes: 1. External addresses can be accessed by clearing the RAME bit in SYSCR to 0.
2. The USB cannot be used in mode 7.
See section 3.3.4, Mode 7, for details.

Figure 3.2 Memory Map in Each Operating Mode for HD6432215B



Notes: 1. External addresses can be accessed by clearing the RAME bit in SYSCR to 0.
2. The USB cannot be used in mode 7.
See section 3.3.4, Mode 7, for details.

Figure 3.3 Memory Map in Each Operating Mode for HD6432215C



Notes: 1. External addresses can be accessed by clearing the RAME bit in SYSCR to 0.
2. Development work using the E6000 emulator:
The USB cannot be used in mode 7.
Development work using the on-chip (E10A-USB) emulator:
The USB can be used in mode 7.
See section 3.3.4, Mode 7, for details.
3. When using an on-chip emulator, H'FEE800 to H'FEFFFF is allocated as ASERAM.

Figure 3.4 Memory Map in Each Operating Mode for HD64F2215R, HD64F2215RU, HD64F2215T, and HD64F2215TU

Section 4 Exception Handling

4.1 Exception Handling Types and Priority

As table 4.1 indicates, exception handling may be caused by a reset, trace, 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. Exception sources, the stack structure, and operation of the CPU vary depending on the interrupt control mode. For details on the interrupt control mode, refer to section 5, Interrupt Controller.

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 $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ pin, or when the watchdog timer overflows. The CPU enters the reset state when the $\overline{\text{RES}}$ pin is low. The CPU enters the manual reset state when the $\overline{\text{MRES}}$ pin is low.
	Trace	Starts when execution of the current instruction or exception handling ends, if the trace (T) bit in the EXR is set to 1. This is enabled only in trace interrupt control mode 2. Trace exception processing is not performed after RTE instruction execution.
	Interrupt	Starts when execution of the current instruction or exception handling ends, if an interrupt request has been issued. Note that after executing the ANDC, ORC, XORC, or LDC instruction or at the completion of reset exception processing, no interrupt is detected.
Low	Trap instruction (TRAPA)	Started by execution of a trap instruction (TRAPA). Trap exception processing is always accepted in program execution state.

4.2 Exception Sources and Exception Vector Table

Different vector addresses are assigned to different exception sources. Table 4.2 lists the exception sources and their vector addresses. Since the usable modes differ depending on the product, for details on each product, refer to section 3, MCU Operating Modes.

Table 4.2 Exception Handling Vector Table

Exception Source		Vector Number	Vector Address[*1] Normal Mode[*2]	Vector Address[*1] Advanced Mode
Power-on reset		0	H'0000 to H'0001	H'0000 to H'0003
Manual reset		1	H'0002 to H'0003	H'0004 to H'0007
Reserved for system use		2	H'0004 to H'0005	H'0008 to H'000B
		3	H'0006 to H'0007	H'000C to H'000F
		4	H'0008 to H'0019	H'0010 to H'0013
Trace		5	H'000A to H000B	H'0014 to H0017
Direct transitions[*2]		6	H'000C to H000D	H'0018 to H001B
External interrupt (NMI)		7	H'000E to H'000F	H'001C to H'001F
Trap instruction	#0	8	H'0010 to H'0011	H'0020 to H'0023
	#1	9	H'0012 to H'0013	H'0024 to H'0027
	#2	10	H'0014 to H'0015	H'0028 to H'002B
	#3	11	H'0016 to H'0017	H'002C to H'002F
Reserved for system use		12	H'0018 to H'0019	H'0030 to H'0033
		13	H'001A to H'001B	H'0034 to H'0037
		14	H'001C to H'001D	H'0038 to H'003B
		15	H'001E to H'001F	H'003C to H'003F
External interrupt	IRQ0	16	H'0020 to H'0021	H'0040 to H'0043
External interrupt	IRQ1	17	H'0022 to H'0023	H'0044 to H'0047
External interrupt	IRQ2	18	H'0024 to H'0025	H'0048 to H'004B
External interrupt	IRQ3	19	H'0026 to H'0027	H'004C to H'004F
External interrupt	IRQ4	20	H'0028 to H'0029	H'0050 to H'0053
External interrupt	IRQ5	21	H'002A to H'002B	H'0054 to H'0057
USB interrupt	IRQ6	22	H'002C to H'002D	H'0058 to H'005B
External interrupt	IRQ7	23	H'002E to H'002F	H'005C to H'005F
Internal interrupt[*3]		24	H'0030 to H'0031	H'0060 to H'0063
		⎪	⎪	⎪
		127	H'00FE to H'00FF	H'01FC to H'01FF

Notes: 1. Lower 16 bits of the address.
2. Not available in this LSI.
3. For details of internal interrupt vectors, see section 5.5, Interrupt Exception Handling Vector Table.

4.3 Reset

A reset has the highest exception priority.

When the $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ pin goes low, all processing halts and this LSI enters the reset state. To ensure that this LSI is reset, hold the $\overline{\text{RES}}$ pin low for at least 20 ms at power-on.

A reset initializes the internal state of the CPU and the registers of on-chip peripheral modules.

This LSI can also be reset by overflow of the watchdog timer. For details, see section 12, Watchdog Timer (WDT).

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

Note: $\overline{\text{TRST}}$ should be brought low level at power-on. For details, see section 14, Boundary Scan Function.

4.3.1 Reset Types

A reset can be of either of two types: a power-on reset or a manual reset. Reset types are shown in table 4.3. A power-on reset should be used when powering on.

The internal state of the CPU is initialized by either type of reset. A power-on reset also initializes all the registers in the on-chip peripheral modules, while a manual reset initializes all the registers in the on-chip peripheral modules except for the bus controller and I/O ports, which retain their previous states.

With a manual reset, since the on-chip peripheral modules are initialized, ports used as on-chip peripheral module I/O pins are switched to I/O ports controlled by DDR and DR.

Table 4.3 Reset Types

	Reset Transition Condition		Internal State	
Type	**$\overline{\text{MRES}}$**	**$\overline{\text{RES}}$**	**CPU**	**On-Chip Peripheral Modules**
Power-on reset	×	Low	Initialized	Initialized
Manual reset	Low	High	Initialized	Initialized, except for bus controller and I/O ports

Legend:
×: Don't care

A reset caused by the watchdog timer can also be of either of two types: a power-on reset or a manual reset.

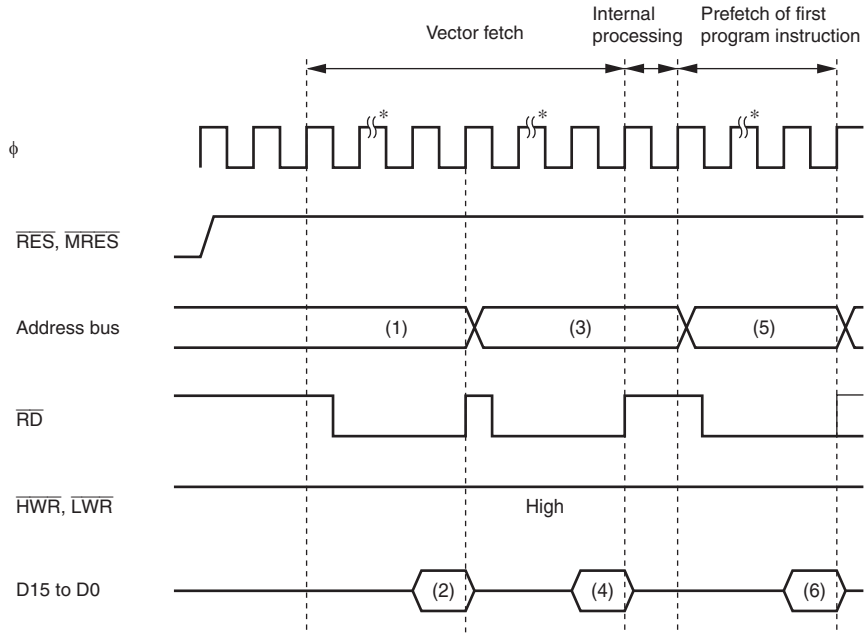
When the $\overline{\text{MRES}}$ pin is used, $\overline{\text{MRES}}$ pin input must be enabled by setting the MRESE bit to 1 in SYSCR.

4.3.2 Reset Exception Handling

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

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

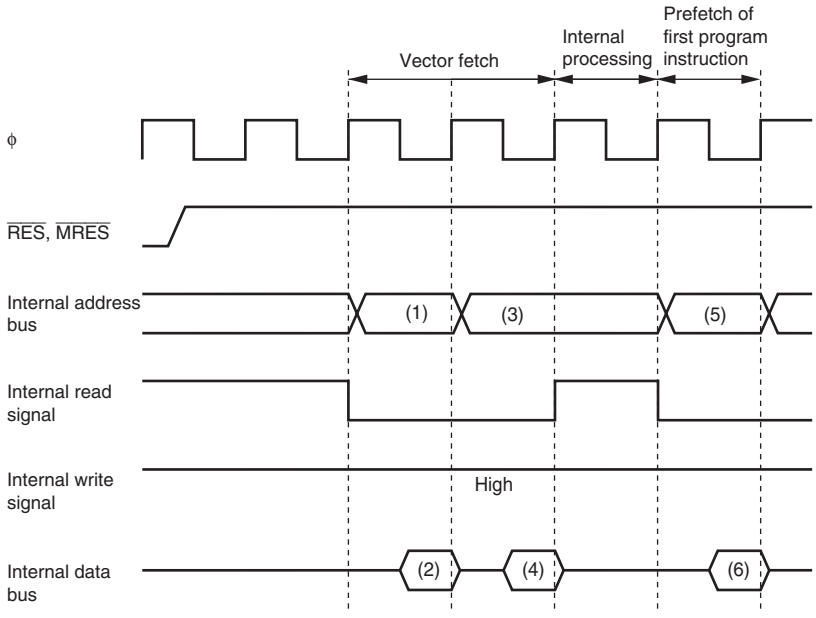
Figures 4.1 and 4.2 show examples of the reset sequence.



(1) (3) Reset exception handling vector address (for power-on reset, (1) = H'000000, (3) = H'000002; for manual reset, (1) = H'000004, (3) = H'000006)
(2) (4) Start address (contents of reset exception handling vector address)
(5) Start address ((5) = (2) (4))
(6) First program instruction

Note: * Three program wait states are inserted.

Figure 4.1 Reset Sequence (Mode 4)



(1) (3) Reset exception handling vector address (for power-on reset, (1) = H'000000, (3) = H'000002; for manual reset, (1) = H'000004, (3) = H'000006)
(2) (4) Start address (contents of reset exception handling vector address)
(5) Start address ((5) = (2) (4))
(6) First program instruction

Figure 4.2 Reset Sequence (Modes 6, 7)

4.3.3 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 SP).

4.3.4 State of On-Chip Peripheral Modules after Reset Release

After reset release, MSTPCRA to MSTPCRC are initialized to H'3F, H'FF, and H'FF, respectively, and all modules except the DMAC and DTC enter module stop mode. Consequently, on-chip peripheral module registers cannot be read from or written to. Register reading and writing is enabled when module stop mode is exited.

4.4 Traces

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 not affected by interrupt masking. Table 4.4 shows the state of CCR and EXR after execution of trace exception handling. Trace mode is canceled by clearing the T bit in EXR to 0. 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.

Interrupts are accepted even within the trace exception handling routine.

Table 4.4 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.

4.5 Interrupts

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. The source to start interrupt exception handling and the vector address differ depending on the product. For details, refer to section 5, Interrupt Controller.

The interrupt exception handling is as follows:

1. The values in the program counter (PC), condition code register (CCR), and extended control register (EXR) are saved in the stack.
2. The interrupt mask bit is updated and the T bit is cleared.
3. A vector address corresponding to the interrupt source is generated, the start address is loaded from the vector table to the PC, and program execution starts from that address.

4.6 Trap Instruction

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 trap instruction exception handling is as follows:

1. The values in the program counter (PC), condition code register (CCR), and extended control register (EXR) are saved in the stack.
2. The interrupt mask bit is updated and the T bit is cleared.
3. A vector address corresponding to the interrupt source is generated, the start address is loaded from the vector table to the PC, and program execution starts from that address.

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

Table 4.5 shows the status of CCR and EXR after execution of trap instruction exception handling.

Table 4.5 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.

4.7 Stack Status after Exception Handling

Figure 4.3 shows the stack after completion of trap instruction exception handling and interrupt exception handling.

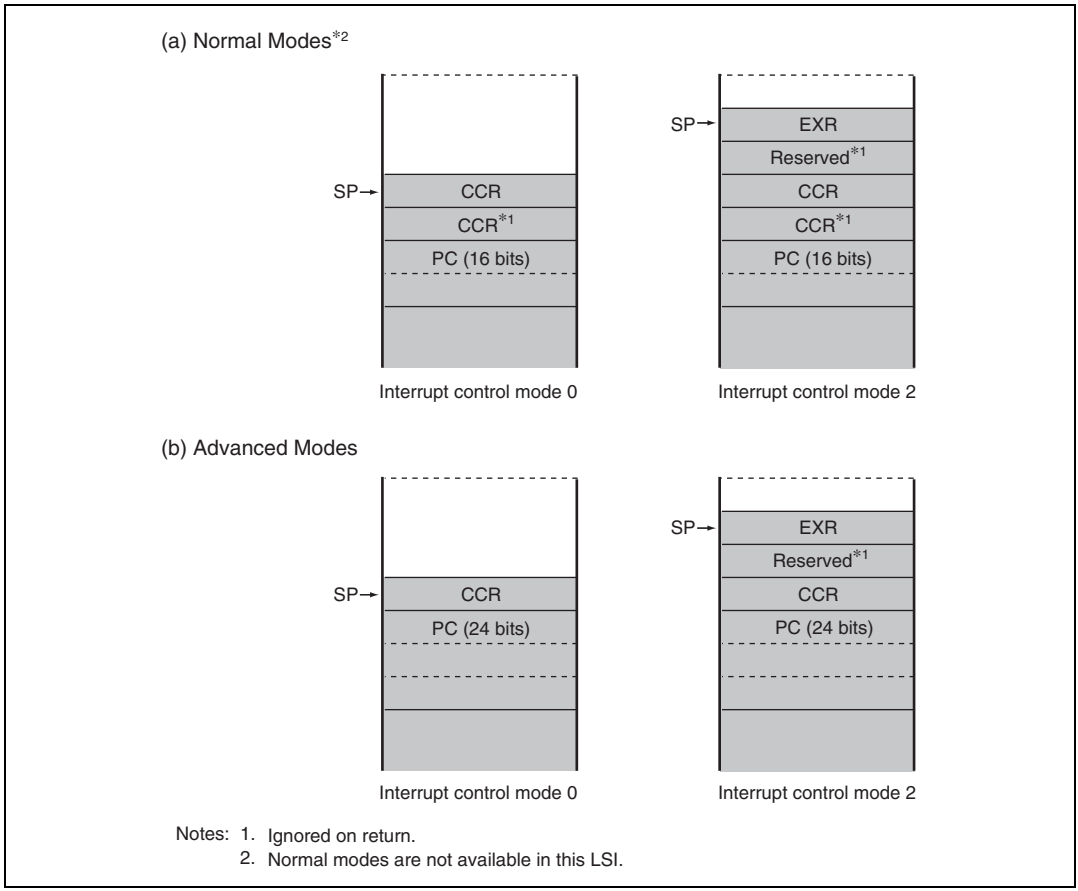


Figure 4.3 Stack Status after Exception Handling

4.8 Notes on Use of the Stack

When accessing word data or longword data, this LSI 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 4.4 shows an example of what happens when the SP value is odd.

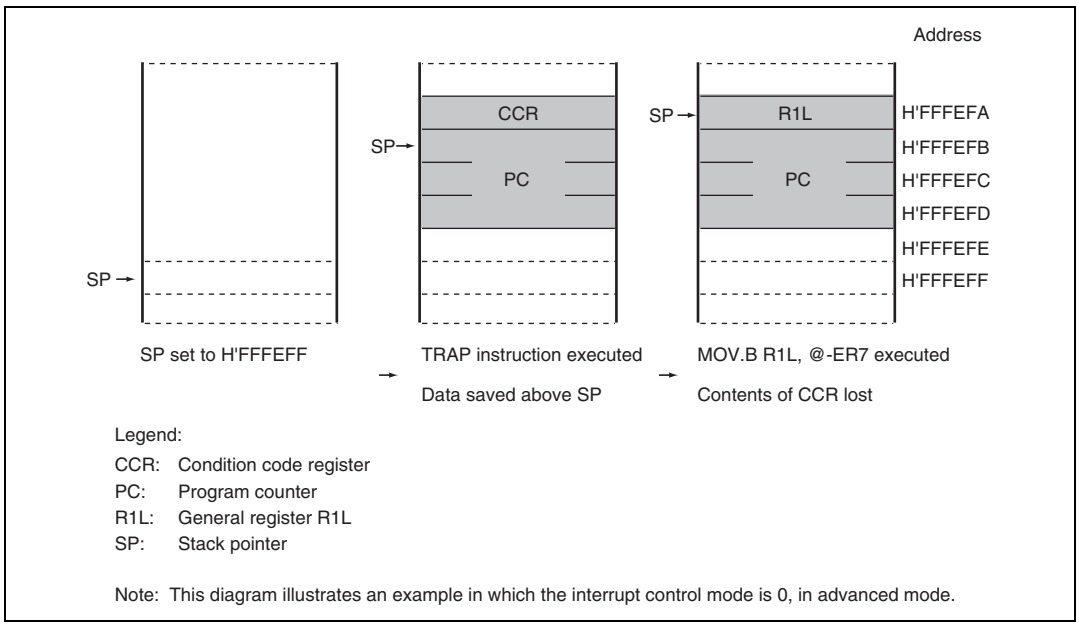


Figure 4.4 Operation when SP Value Is Odd

Section 5 Interrupt Controller

5.1 Features

- Two interrupt control modes
  - Any 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 IPR
  - An interrupt priority register (IPR) is 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.
- Eight external interrupts (NMI, $\overline{\text{IRQ7}}$, and $\overline{\text{IRQ5}}$ to $\overline{\text{IRQ0}}$)
  - 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 $\overline{\text{IRQ7}}$ and $\overline{\text{IRQ5}}$ to $\overline{\text{IRQ0}}$. $\overline{\text{IRQ6}}$ is an interrupt only for the on-chip USB.
- DTC and DMAC control
  - DTC or DMAC activation is performed by means of interrupts.

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

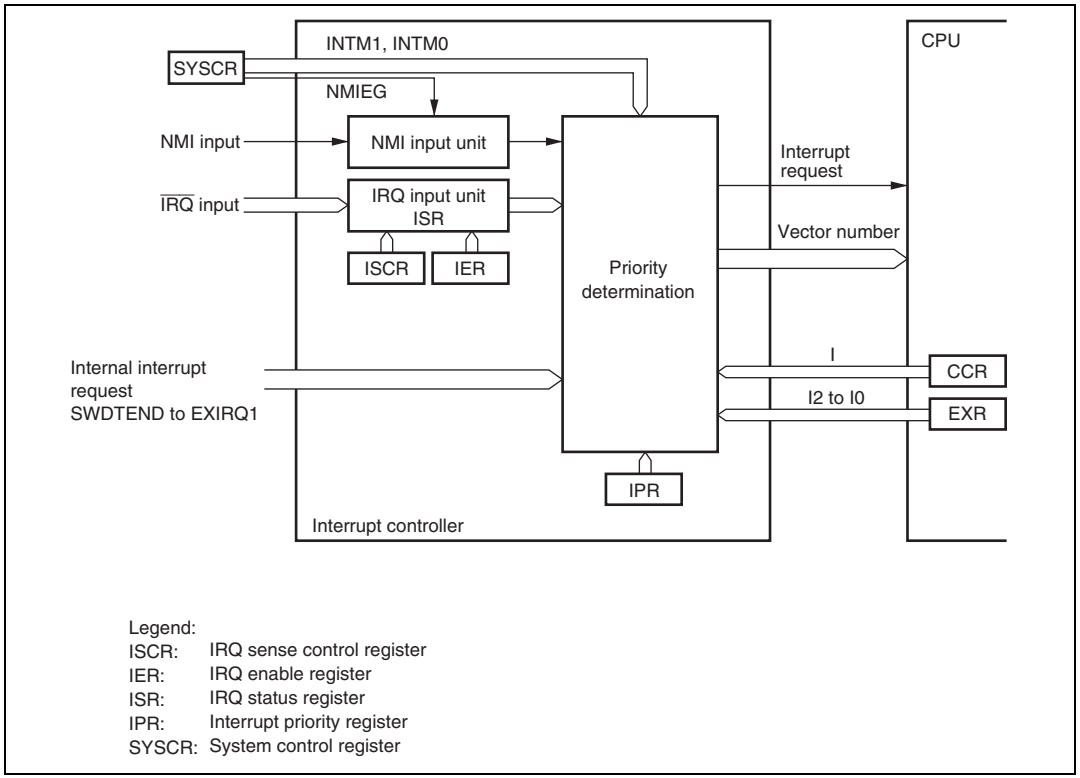


Figure 5.1 Block Diagram of Interrupt Controller

5.2 Input/Output Pins

Table 5.1 summarizes the pins of the interrupt controller.

Table 5.1 Pin Configuration

Name	I/O	Function
NMI	Input	Nonmaskable external interrupt Rising or falling edge can be selected
$\overline{\text{IRQ7}}$	Input	Maskable external interrupts
$\overline{\text{IRQ5}}$	Input	Rising, falling, or both edges, or level sensing, ($\overline{\text{IRQ6}}$ is an interrupt signal only for the on-chip USB) can be selected
$\overline{\text{IRQ4}}$	Input	
$\overline{\text{IRQ3}}$	Input	
$\overline{\text{IRQ2}}$	Input	
$\overline{\text{IRQ1}}$	Input	
$\overline{\text{IRQ0}}$	Input	

5.3 Register Descriptions

The interrupt controller has the following registers. For details on the system control register, refer to section 3.2.2, System Control Register (SYSCR).

- System control register (SYSCR)
- IRQ sense control register H (ISCRH)
- IRQ sense control register L (ISCRL)
- IRQ enable register (IER)
- IRQ status register (ISR)
- Interrupt priority register A (IPRA)
- Interrupt priority register B (IPRB)
- Interrupt priority register C (IPRC)
- Interrupt priority register D (IPRD)
- Interrupt priority register E (IPRE)
- Interrupt priority register F (IPRF)
- Interrupt priority register G (IPRG)
- Interrupt priority register I (IPRI)
- Interrupt priority register J (IPRJ)
- Interrupt priority register K (IPRK)
- Interrupt priority register M (IPRM)

5.3.1 Interrupt Priority Registers A to G, I to K, M (IPRA to IPRG, IPRI to IPRK, IPRM)

The IPR registers set priorities (levels 7 to 0) for interrupts other than NMI.

The correspondence between interrupt sources and IPR settings is shown in table 5.2. Setting a value in the range from H'0 to H'7 in the 3-bit groups of bits 6 to 4 and 2 to 0 sets the priority of the corresponding interrupt.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved These bits are always read as 0 and cannot be modified.
6	IPR6	1	R/W	Sets the priority of the corresponding interrupt source.
5	IPR5	1	R/W	000: Priority level 0 (Lowest)
4	IPR4	1	R/W	001: Priority level 1 010: Priority level 2 011: Priority level 3 100: Priority level 4 101: Priority level 5 110: Priority level 6 111: Priority level 7 (Highest)
3	—	0	—	Reserved These bits are always read as 0 and cannot be modified.
2	IPR2	1	R/W	Sets the priority of the corresponding interrupt source.
1	IPR1	1	R/W	000: Priority level 0 (Lowest)
0	IPR0	1	R/W	001: Priority level 1 010: Priority level 2 011: Priority level 3 100: Priority level 4 101: Priority level 5 110: Priority level 6 111: Priority level 7 (Highest)

5.3.2 IRQ Enable Register (IER)

IER controls enabling and disabling of interrupt requests IRQ7 to IRQ0.

Bit	Bit Name	Initial Value	R/W	Description
7	IRQ7E	0	R/W	IRQ7 Enable The IRQ7 interrupt request is enabled when this bit is 1.
6	IRQ6E	0	R/W	IRQ6 Enable* The IRQ6 interrupt request is enabled when this bit is 1.
5	IRQ5E	0	R/W	IRQ5 Enable The IRQ5 interrupt request is enabled when this bit is 1.
4	IRQ4E	0	R/W	IRQ4 Enable The IRQ4 interrupt request is enabled when this bit is 1.
3	IRQ3E	0	R/W	IRQ3 Enable The IRQ3 interrupt request is enabled when this bit is 1.
2	IRQ2E	0	R/W	IRQ2 Enable The IRQ2 interrupt request is enabled when this bit is 1.
1	IRQ1E	0	R/W	IRQ1 Enable The IRQ1 interrupt request is enabled when this bit is 1.
0	IRQ0E	0	R/W	IRQ0 Enable The IRQ0 interrupt request is enabled when this bit is 1.

Note: * IRQ6 is an interrupt only for the on-chip USB.

5.3.3 IRQ Sense Control Registers H and L (ISCRH, ISCRL)

The ISCR registers select the source that generates an interrupt request at pins $\overline{\text{IRQ7}}$, and $\overline{\text{IRQ5}}$ to $\overline{\text{IRQ0}}$.

Bit	Bit Name	Initial Value	R/W	Description
15	IRQ7SCB	0	R/W	IRQ7 Sense Control B
14	IRQ7SCA	0	R/W	IRQ7 Sense Control A 00: Interrupt request generated at $\overline{\text{IRQ7}}$ input low level 01: Interrupt request generated at falling edge of $\overline{\text{IRQ7}}$ input 10: Interrupt request generated rising edge of $\overline{\text{IRQ7}}$ input 11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ7}}$ input
13	IRQ6SCB	0	R/W	IRQ6* Sense Control B
12	IRQ6SCA	0	R/W	IRQ6* Sense Control A 00: Setting prohibited when using on-chip USB suspend or resume interrupt 01: Interrupt request generated at falling edge of $\overline{\text{IRQ6}}$ input 1×: Setting prohibited
11	IRQ5SCB	0	R/W	IRQ5 Sense Control B
10	IRQ5SCA	0	R/W	IRQ5 Sense Control A 00: Interrupt request generated at $\overline{\text{IRQ5}}$ input low level 01: Interrupt request generated at falling edge of $\overline{\text{IRQ5}}$ input 10: Interrupt request generated at rising edge of $\overline{\text{IRQ5}}$ input 11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ5}}$ input

Legend:

×: Don't care

Note: * IRQ6 is an interrupt only for the on-chip USB.

Bit	Bit Name	Initial Value	R/W	Description
9	IRQ4SCB	0	R/W	IRQ4 Sense Control B
8	IRQ4SCA	0	R/W	IRQ4 Sense Control A 00: Interrupt request generated at $\overline{\text{IRQ4}}$ input low level 01: Interrupt request generated at falling edge of $\overline{\text{IRQ4}}$ input 10: Interrupt request generated at rising edge of $\overline{\text{IRQ4}}$ input 11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ4}}$ input
7	IRQ3SCB	0	R/W	IRQ3 Sense Control B
6	IRQ3SCA	0	R/W	IRQ3 Sense Control A 00: Interrupt request generated at $\overline{\text{IRQ3}}$ input low level 01: Interrupt request generated at falling edge of $\overline{\text{IRQ3}}$ input 10: Interrupt request generated at rising edge of $\overline{\text{IRQ3}}$ input 11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ3}}$ input
5	IRQ2SCB	0	R/W	IRQ2 Sense Control B
4	IRQ2SCA	0	R/W	IRQ2 Sense Control A 00: Interrupt request generated at $\overline{\text{IRQ2}}$ input low level 01: Interrupt request generated at falling edge of $\overline{\text{IRQ2}}$ input 10: Interrupt request generated at rising edge of $\overline{\text{IRQ2}}$ input 11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ2}}$ input
3	IRQ1SCB	0	R/W	IRQ1 Sense Control B
2	IRQ1SCA	0	R/W	IRQ1 Sense Control A 00: Interrupt request generated at $\overline{\text{IRQ1}}$ input low level 01: Interrupt request generated at falling edge of $\overline{\text{IRQ1}}$ input 10: Interrupt request generated at rising edge of $\overline{\text{IRQ1}}$ input 11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ1}}$ input

Bit	Bit Name	Initial Value	R/W	Description
1	IRQ0SCB	0	R/W	IRQ0 Sense Control B
0	IRQ0SCA	0	R/W	IRQ0 Sense Control A 00: Interrupt request generated at $\overline{\text{IRQ0}}$ input low level 01: Interrupt request generated at falling edge of $\overline{\text{IRQ0}}$ input 10: Interrupt request generated at rising edge of $\overline{\text{IRQ0}}$ input 11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ0}}$ input

5.3.4 IRQ Status Register (ISR)

ISR indicates the status of IRQ7 to IRQ0 interrupt requests. Only 0 should be written to these bits for clearing the flag.

Bit	Bit Name	Initial Value	R/W	Description
7	IRQ7F	0	R/(W)*	[Setting condition] • When the interrupt source selected by the ISCR registers occurs [Clearing conditions] • Cleared by reading IRQnF flag when IRQnF = 1, then writing 0 to IRQnF flag • When interrupt exception handling is executed when low-level detection is set and, $\overline{\text{IRQn}}$ input is high • When IRQn interrupt exception handling is executed when falling, rising, or both-edge detection is set • When the DTC is activated by an IRQn interrupt, and the DISEL bit in MRB of the DTC is cleared to 0
6	IRQ6F	0	R/(W)*	
5	IRQ5F	0	R/(W)*	
4	IRQ4F	0	R/(W)*	
3	IRQ3F	0	R/(W)*	
2	IRQ2F	0	R/(W)*	
1	IRQ1F	0	R/(W)*	
0	IRQ0F	0	R/(W)*	

Note: * The write value should always be 0 to clear the flag.

5.4 Interrupt Sources

5.4.1 External Interrupts

There are eight external interrupts: NMI, IRQ7, and IRQ5 to IRQ0. These interrupts can be used to restore this LSI from software standby mode. IRQ6 is an interrupt only for the on-chip USB. However, IRQ6 is functionally same as IRQ7 restore this LSI from software standby mode. IRQ6 is functionally same as IRQ7 and IRQ5 to IRQ0.

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

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 IRQn interrupts is shown in figure 5.2.

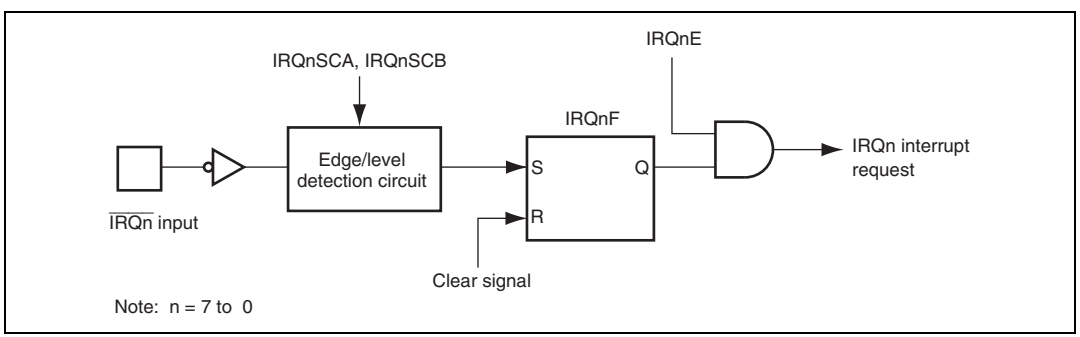


Figure 5.2 Block Diagram of IRQn Interrupts

The setting for IRQnF is shown in figure 5.3.

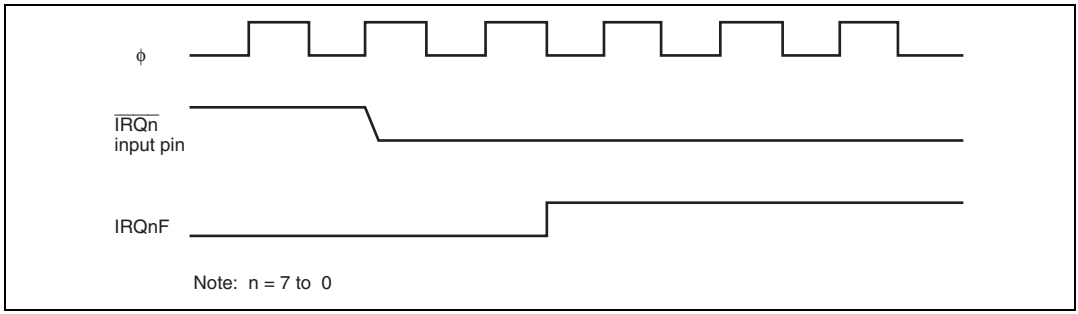


Figure 5.3 Set Timing for IRQnF

The detection of IRQn interrupts does not depend on whether the relevant pin has been set for input or output. However, when a pin is used as an external interrupt input pin, do not clear the corresponding DDR to 0; and use the pin as an I/O pin for another function. IRQnF interrupt request flag is set when the setting condition is satisfied, regardless of IER settings. Accordingly, refer to only necessary flags.

5.4.2 Internal Interrupts

The sources for internal interrupts from on-chip peripheral modules have the following features:

- For each on-chip peripheral module there are flags that indicate the interrupt request status, and enable bits that 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 or DTC can be activated by a TPU, SCI, or other interrupt request.
- When the DMAC or DTC is activated by an interrupt request, it is not affected by the interrupt control mode or CPU interrupt mask bit.

5.5 Interrupt Exception Handling Vector Table

Table 5.2 shows interrupt exception handling sources, vector addresses, and interrupt priorities.

For default priorities, the lower the vector number, the higher the priority. Priorities among modules can be set by means of the IPR. Modules set at the same priority will conform to their default priorities. Priorities within a module are fixed.

Table 5.2 Interrupt Sources, Vector Addresses, and Interrupt Priorities

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address*		Priority
			Advanced Mode	IPR	
External pins	NMI	7	H'001C		High
	IRQ0	16	H'0040	IPRA6 to IPRA4	
	IRQ1	17	H'0044	IPRA2 to IPRA0	
	IRQ2	18	H'0048	IPRB6 to IPRB4	
	IRQ3	19	H'004C		
	IRQ4	20	H'0050	IPRB2 to IPRB0	
	IRQ5	21	H'0054		
USB	IRQ6	22	H'0058	IPRC6 to IPRC4	
External pins	IRQ7	23	H'005C		
DTC	SWDTEND	24	H'0060	IPRC2 to IPRC0	
Watchdog Timer	WOVI	25	H'0064	IPRD6 to IPRD4	
A/D	ADI	28	H'0070		
TPU channel 0	TGI0A	32	H'0080	IPRF6 to IPRF4	
	TGI0B	33	H'0084		
	TGI0C	34	H'0088		
	TGI0D	35	H'008C		
	TGI0V	36	H'0090		
TPU channel 1	TGI1A	40	H'00A0	IPRF2 to IPRF0	
	TGI1B	41	H'00A4		
	TGI1V	42	H'00A8		
	TGI1U	43	H'00AC		
TPU channel 2	TGI2A	44	H'00B0	IPRG6 to IPRG4	
	TGI2B	45	H'00B4		
	TGI2V	46	H'00B8		
	TGI2U	47	H'00BC		
8-bit timer channel 0	CMIA0 (compare match A)	64	H'0100	IPRI6 to IPRI4	
	CMIB0 (compare match B)	65	H'0104		
	OVI0 (overflow)	66	H'0108		Low

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address*	IPR	Priority
8-bit timer channel 1	CMIA1 (compare match A)	68	H'0110	IPRI2 to IPRI0	High
	CMIB1 (compare match B)	69	H'0114		
	OVI1 (overflow)	70	H'0118		
DMAC	DEND0A	72	H'0120	IPRJ6 to IPRJ4	
	DEND0B	73	H'0124		
	DEND1A	74	H'0128		
	DEND1B	75	H'012C		
SCI channel 0	ERI0	80	H'0140	IPRJ2 to IPRJ0	
	RXI0	81	H'0144		
	TXI0	82	H'0148		
	TEI0	83	H'014C		
SCI channel 1	ERI1	84	H'0150	IPRK6 to IPRK4	
	RXI1	85	H'0154		
	TXI1	86	H'0158		
	TEI1	87	H'015C		
SCI channel 2	ERI2	88	H'0160	IPRK2 to IPRK0	
	RXI2	89	H'0164		
	TXI2	90	H'0168		
	TEI2	91	H'016C		
USB	EXIRQ0	104	H'01A0	IPRM6 to IPRM4	
	EXIRQ1	105	H'01A4		Low

Note: * Lower 16 bits of the start address.

5.6 Interrupt Control Modes and Interrupt Operation

The interrupt controller has two modes: interrupt control mode 0 and interrupt control mode 2.

Interrupt operations differ depending on the interrupt control mode. The interrupt control mode is selected by SYSCR. Table 5.3 shows the differences between interrupt control mode 0 and interrupt control mode 2.

Table 5.3 Interrupt Control Modes

Interrupt Control Mode	Priority Setting Register	Interrupt Mask Bits	Description
0	Default	I	The priority of interrupt sources are fixed at the default settings. Interrupt sources except for NMI is marked by the I bit.
2	IPR	I2 to I0	8-level interrupt mask control is performed by bits I2 to I0. 8 priority levels other than NMI can be set with IPR.

5.6.1 Interrupt Control Mode 0

In interrupt control mode 0, interrupt requests except for NMI is masked by the I bit of CCR in the CPU. Figure 5.4 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. If the I bit is set to 1, only an NMI interrupt is accepted, and other interrupt requests are held pending. If the I bit is cleared, an interrupt request is accepted.
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 the CPU accepts an interrupt request, it starts interrupt exception handling after execution of the current instruction has been completed.
5. The PC and CCR are saved to the stack area by interrupt exception handling. The PC saved on the stack shows the address of the first instruction to be executed after returning from the interrupt handling routine.
6. Next, the I bit in CCR is set to 1. This masks all interrupts except NMI.

7. The CPU generates a vector address for the accepted interrupt and starts execution of the interrupt handling routine at the address indicated by the contents of the vector address in the vector table.

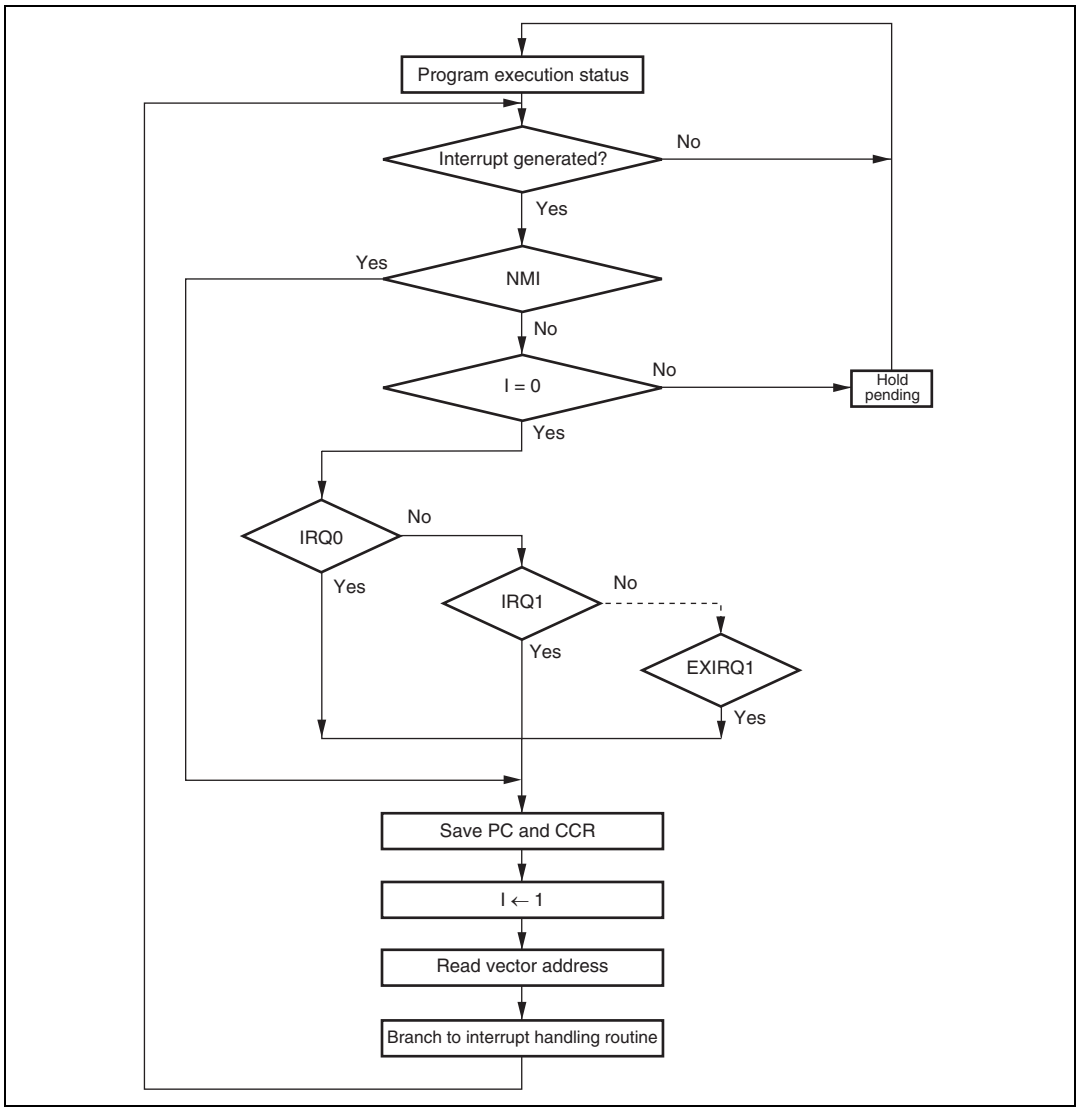


Figure 5.4 Flowchart of Procedure Up to Interrupt Acceptance in Interrupt Control Mode 0

5.6.2 Interrupt Control Mode 2

In interrupt control mode 2, mask control is done in eight levels for interrupt requests except for NMI by comparing the EXR interrupt mask level (I2 to I0 bits) in the CPU and the IPR setting.

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. 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 5.2 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 the CPU accepts an interrupt request, it starts interrupt exception handling 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. The CPU generates a vector address for the accepted interrupt and starts execution of the interrupt handling routine at the address indicated by the contents of the vector address in the vector table.

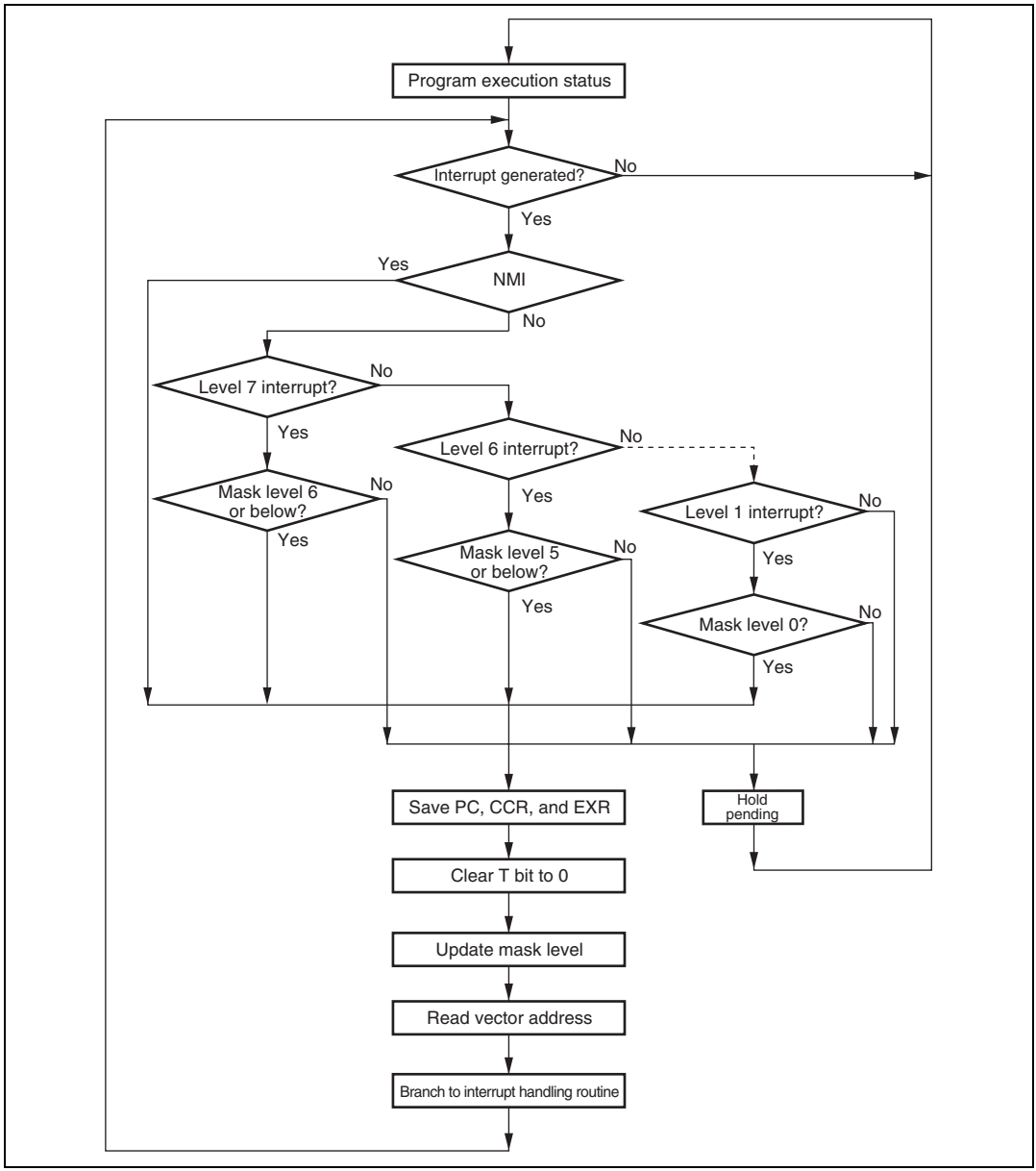


Figure 5.5 Flowchart of Procedure Up to Interrupt Acceptance in Interrupt Control Mode 2

5.6.3 Interrupt Exception Handling Sequence

Figure 5.6 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.

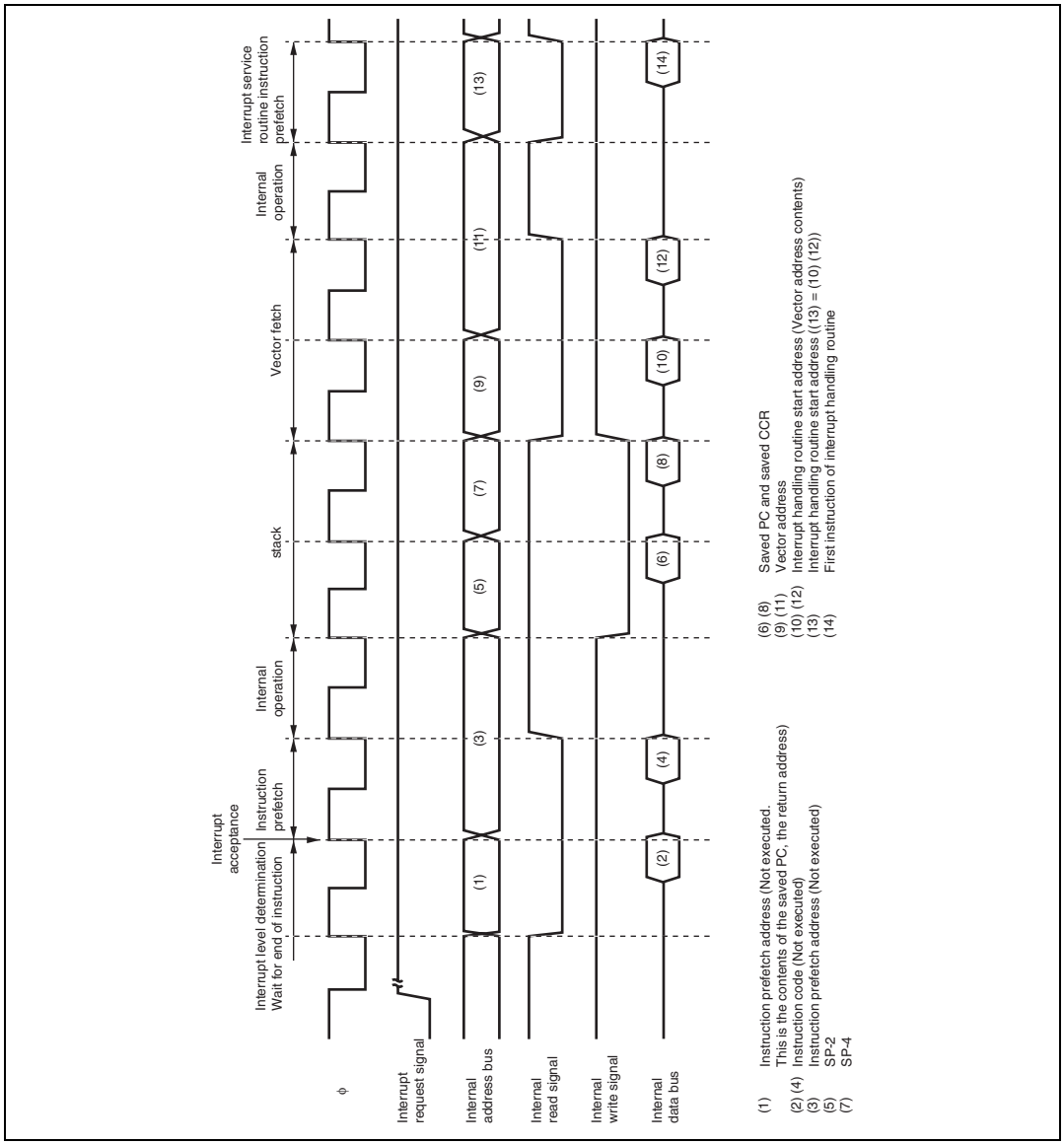


Figure 5.6 Interrupt Exception Handling

5.6.4 Interrupt Response Times

Table 5.4 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 5.4 are explained in table 5.5.

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

Table 5.4 Interrupt Response Times

		Normal Mode*5		Advanced Mode	
No.	Execution State	Interrupt Control Mode 0	Interrupt Control Mode 2	Interrupt Control Mode 0	Interrupt Control Mode 2
1	Interrupt priority determination*1	3	3	3	3
2	Number of wait states until executing instruction ends*2	1 to 19+2·S_I	1 to 19+2·S_I	1 to 19+2·S_I	1 to 19+2·S_I
3	PC, CCR, EXR stack save	2·S_K	3·S_K	2·S_K	3·S_K
4	Vector fetch	S_I	S_I	2·S_I	2·S_I
5	Instruction fetch*3	2·S_I	2·S_I	2·S_I	2·S_I
6	Internal processing*4	2	2	2	2
Total	(using on-chip memory)	11 to 31	12 to 32	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.
5. Not available in this LSI.

Table 5.5 Number of States in Interrupt Handling Routine Execution Statuses

		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.

5.6.5 DTC Activation by Interrupt

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

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

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

Figure 5.7 shows a block diagram of the interrupt controller of DTC and DMAC.

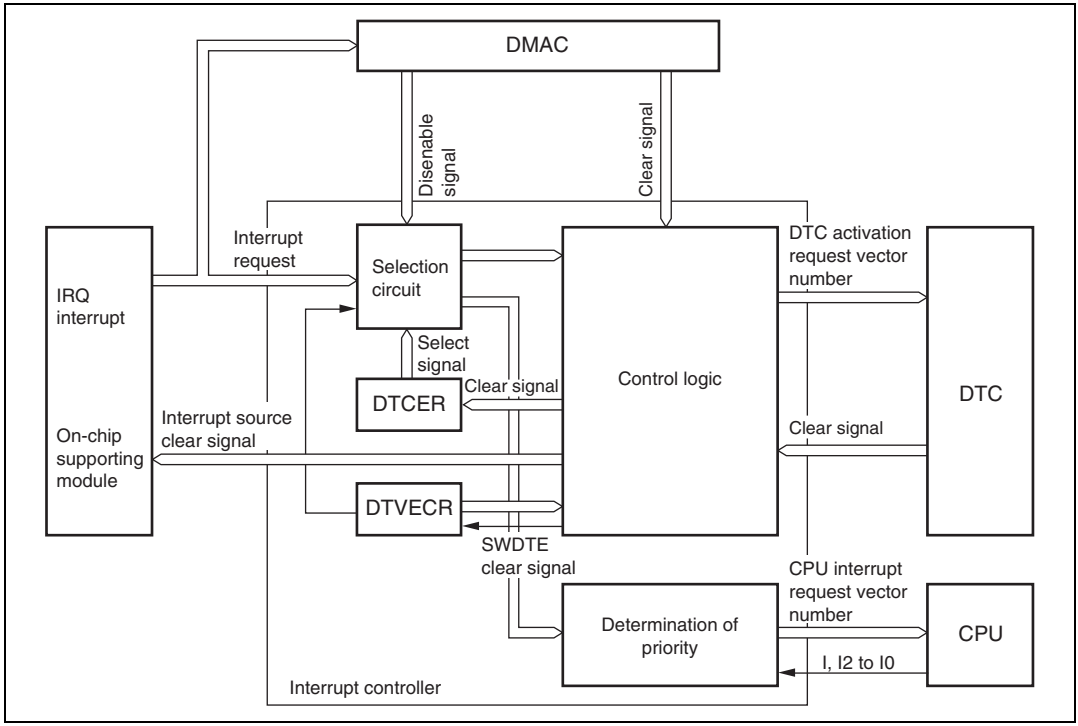


Figure 5.7 Interrupt Control for DTC and DMAC

Selection of Interrupt Source: An activation factor is directly input to each channel of the DMAC. The activation factors for each channel of the DMAC are selected by the DTF3 to DTF0 bits of DMACR. The DTA bit of DMABCR can be used to select whether the selected activation factors are managed by the DMAC. By setting the DTA bit to 1, the interrupt factor which was the activation factor for that DMAC cannot act as the DTC activation factor or the CPU interrupt factor.

Interrupt factors other than the interrupts managed by the DMAC are selected as DTC activation request or CPU interrupt request by the DTCERA to DTCERF of DTC and the DTCE bit of DTCERI.

By specifying the DISEL bit of the DTC's MRB, it is possible to clear the DTCE bit to 0 after DTC data transfer, and request a CPU interrupt.

If DTC carries out the designate number of data transfers and the transfer counter reads 0, after DTC data transfer, the DTCE bit is also cleared to 0, CPU interrupt requested.

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 8.4, Location of Register Information and DTC Vector Table. The activation source is directly input to each channel of DMAC.

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 the DMAC activation factor and as the DTC activation factor or CPU interrupt factor, these operate independently. They operate in accordance with the respective operating states and bus priorities.

Table 5.6 shows the interrupt factor clear control and selection of interrupt factors by specification of the DTA bit of DMAC's DMABCR, DTC's DTCERA to DTCERF's DTCE bit, and the DISEL bit of DTC's MRB.

Table 5.6 Interrupt Source Selection and Clearing Control

Settings					
DMAC	**DTC**		**Interrupt Sources Selection/Clearing Control**		
DTA	**DTCE**	**DISEL**	**DMAC**	**DTC**	**CPU**
0	0	*	Δ	X	O
	1	0	Δ	O	X
		1	Δ	Δ	O
1	*	*	O	X	X

Legend:

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

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

X: The relevant bit cannot be used.

*: Don't care

Notes on Use: The SCI interrupt source is cleared when the DMAC or DTC reads or writes to the prescribed register, and is not dependent upon the DTA bit, DTCE bit, or DISEL bit.

5.7 Usage Notes

5.7.1 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.

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 to 0.

Figure 5.8 shows an example in which the TGIEA bit in the TPU's TIER_0 is cleared to 0.

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

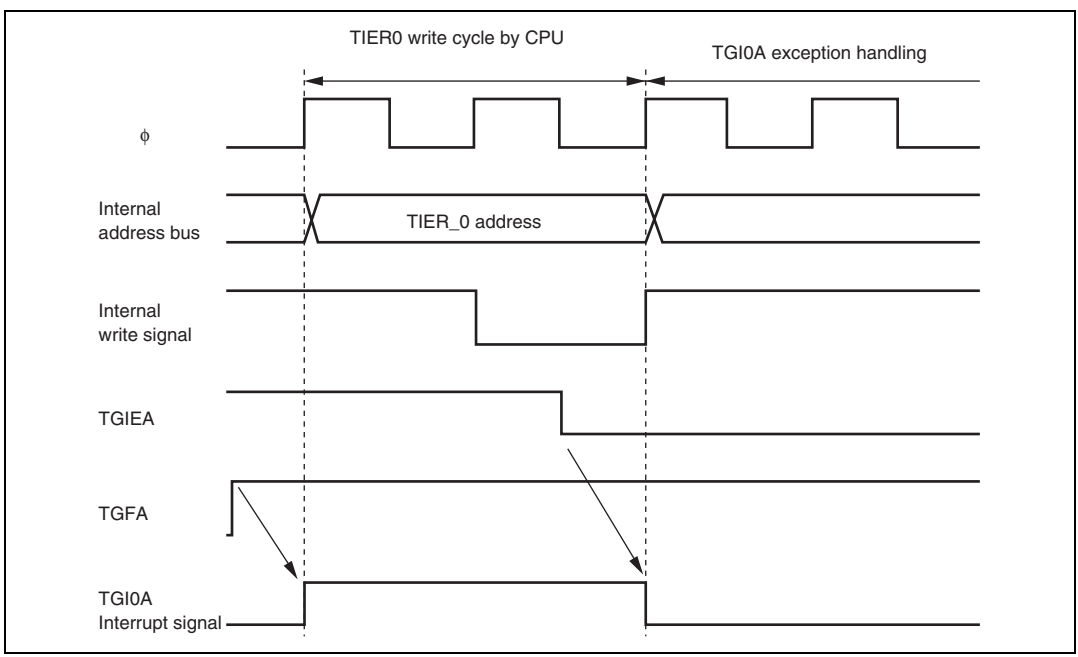


Figure 5.8 Contention between Interrupt Generation and Disabling

5.7.2 Instructions that Disable Interrupts

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.

5.7.3 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.

5.7.4 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
```

Section 6 Bus Controller

This LSI has a built-in bus controller (BSC) that manages the external address space divided into eight areas. 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 Features

- Manages external address space in area units
  - Manages the external space as eight areas of 2 Mbytes
  - Bus specifications can be set independently for each area
  - Burst ROM interface can be set
- Basic bus interface*
  - Chip select ($\overline{CS0}$ to $\overline{CS7}$) 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
- Burst ROM interface
  - Burst ROM interface can be selected for area 0
  - One or two states can be selected for the burst cycle
- Idle cycle insertion
  - Idle cycle can be inserted between consecutive read accesses to different areas
  - Idle cycle can be inserted before a write access to an external area immediately after a read access to an external area
- Bus arbitration
  - The on-chip bus arbiter arbitrates bus mastership among CPU, DMAC, and DTC
- Other features
  - External bus release function

Note: * Chip select $\overline{CS6}$ in area 6 is for the on-chip USB. Therefore it cannot be used as an external area. 8-bit bus mode, 3-state access, and no program wait state should be set for area 6.

Figure 6.1 shows a block diagram of the bus controller.

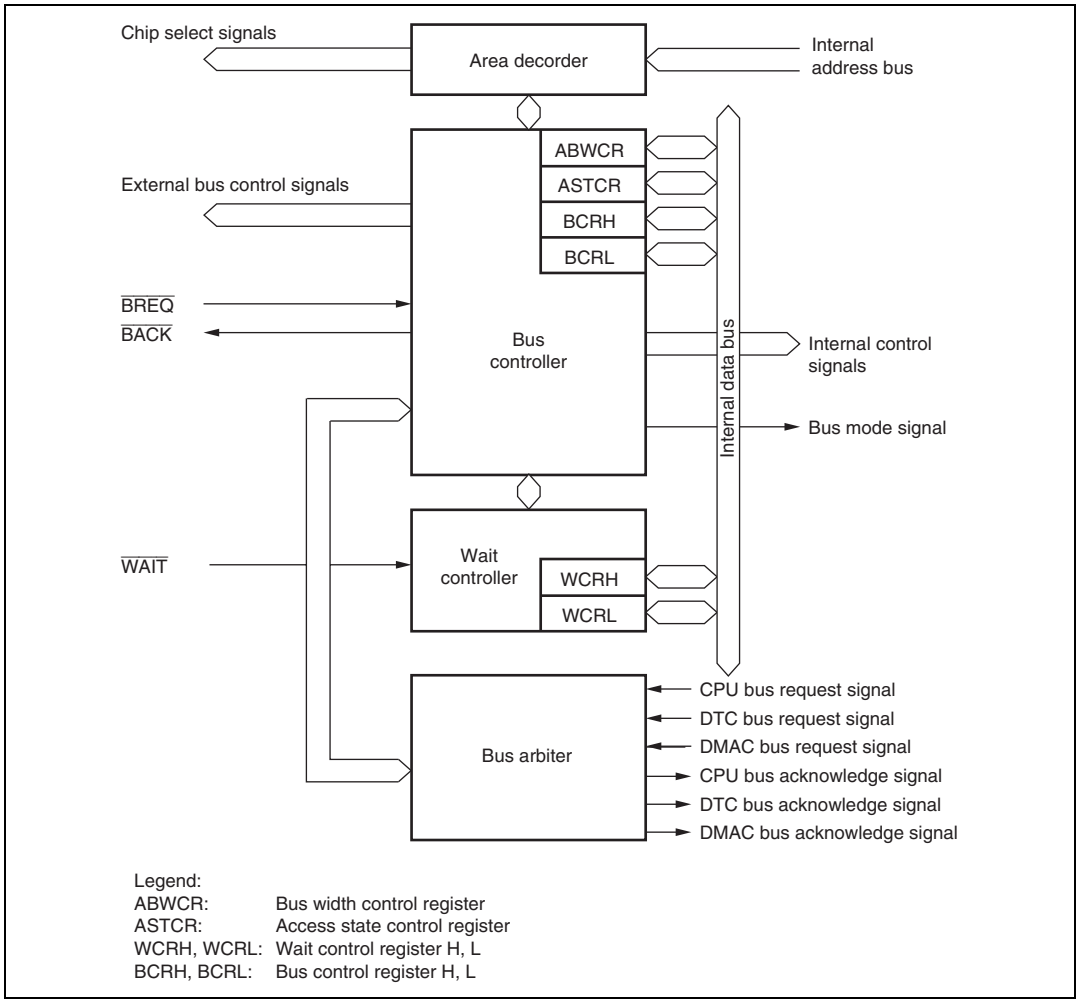


Figure 6.1 Block Diagram of Bus Controller

6.2 Input/Output Pins

Table 6.1 summarizes the pins of the bus controller.

Table 6.1 Pin Configuration

Name	Symbol	I/O	Function
Address strove	$\overline{AS}$	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	$\overline{HWR}$	Output	Strobe signal indicating that external space is to be written, and upper half (D15 to D8) of data bus is enabled.
Low write	$\overline{LWR}$	Output	Strobe signal indicating that external space is to be written, and lower half (D7 to D0) of data bus is enabled.
Chip select 0 to 7	$\overline{CS0}$ to $\overline{CS7}$	Output	Strobe signal indicating that areas 0 to 7 are selected.
Wait	$\overline{WAIT}$	Input	Wait request signal when accessing external 3-state access space.
Bus request	$\overline{BREQ}$	Input	Request signal that releases bus to external device.
Bus request acknowledge	$\overline{BACK}$	Output	Acknowledge signal indicating that bus has been released.

6.3 Register Descriptions

The following shows the registers of the bus controller.

- Bus width control register (ABWCR)
- Access state control register (ASTCR)
- Wait control register H (WCRH)
- Wait control register L (WCRL)
- Bus control register H (BCRH)
- Bus control register L (BCRL)
- Pin function control register (PFCR)

6.3.1 Bus Width Control Register (ABWCR)

ABWCR designates each area for either 8-bit access or 16-bit access.

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

Bit	Bit Name	Initial Value	R/W	Description
7	ABW7	1/0*1	R/W	Area 7 to 0 Bus Width Controls
6	ABW6*2	1/0*1	R/W	These bits select whether the corresponding area is to be designated for 8-bit access or 16-bit access.
5	ABW5	1/0*1	R/W	
4	ABW4	1/0*1	R/W	0: Area n is designated for 16-bit access
3	ABW3	1/0*1	R/W	1: Area n is designated for 8-bit access
2	ABW2	1/0*1	R/W	Note: n = 7 to 0
1	ABW1	1/0*1	R/W	
0	ABW0	1/0*1	R/W	

Notes: 1. In modes 5 to 7, initial value of each bit is 1. In mode 4, initial value of each bit is 0.
2. The on-chip USB is allocated to area 6. Therefore this bit should be set to 1.

6.3.2 Access State Control Register (ASTCR)

ASTCR designates each area as either a 2-state access space or a 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 except for the on-chip USB is fixed regardless of the settings in ASTCR.

Bit	Bit Name	Initial Value	R/W	Description
7	AST7	1	R/W	Area 7 to 0 Access State Controls
6	AST6*	1	R/W	These bits select whether the corresponding area is to be designated as a 2-state access space or a 3-state access space. Wait state insertion is enabled or disabled at the same time.
5	AST5	1	R/W	
4	AST4	1	R/W	
3	AST3	1	R/W	0: Area n is designated for 2-state access Wait state insertion in area n external space is disabled
2	AST2	1	R/W	
1	AST1	1	R/W	
0	AST0	1	R/W	1: Area n is designated for 3-state access Wait state insertion in area n external space is enabled
				Note: n = 7 to 0

Note: * The on-chip USB is allocated to area 6. Therefore this bit should be set to 1.

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

WCRH and WCRL select the number of program wait states for each area.

Program waits are not inserted in the case of on-chip memory or internal I/O registers except for the on-chip USB.

- WCRH

Bit	Bit Name	Initial Value	R/W	Description
7	W71	1	R/W	Area 7 Wait Control 1 and 0
6	W70	1	R/W	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. 00: Program wait not inserted when external space area 7 is accessed 01: 1 program wait state inserted when external space area 7 is accessed 10: 2 program wait states inserted when external space area 7 is accessed 11: 3 program wait states inserted when external space area 7 is accessed
5	W61*	1	R/W	Area 6 Wait Control 1 and 0
4	W60*	1	R/W	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. 00: Program wait not inserted when external space area 6 is accessed 01: 1 program wait state inserted when external space area 6 is accessed 10: 2 program wait states inserted when external space area 6 is accessed 11: 3 program wait states inserted when external space area 6 is accessed

Bit	Bit Name	Initial Value	R/W	Description
3	W51	1	R/W	Area 5 Wait Control 1 and 0
2	W50	1	R/W	These bits select the number of program wait states when area 5 in external space is accessed while the AST5 bit in ASTCR is set to 1. 00: Program wait not inserted when external space area 5 is accessed 01: 1 program wait state inserted when external space area 5 is accessed 10: 2 program wait states inserted when external space area 5 is accessed 11: 3 program wait states inserted when external space area 5 is accessed
1	W41	1	R/W	Area 4 Wait Control 1 and 0
0	W40	1	R/W	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. 00: Program wait not inserted when external space area 4 is accessed 01: 1 program wait state inserted when external space area 4 is accessed 10: 2 program wait states inserted when external space area 4 is accessed 11: 3 program wait states inserted when external space area 4 is accessed

Note: * The on-chip USB is allocated to area 6. Therefore these bits should be set to 0.

- WCRL

Bit	Bit Name	Initial Value	R/W	Description
7	W31	1	R/W	Area 3 Wait Control 1 and 0
6	W30	1	R/W	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. 00: Program wait not inserted when external space area 3 is accessed 01: 1 program wait state inserted when external space area 3 is accessed 10: 2 program wait states inserted when external space area 3 is accessed 11: 3 program wait states inserted when external space area 3 is accessed
5	W21	1	R/W	Area 2 Wait Control 1 and 0
4	W20	1	R/W	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. 00: Program wait not inserted when external space area 2 is accessed 01: 1 program wait state inserted when external space area 2 is accessed 10: 2 program wait states inserted when external space area 2 is accessed 11: 3 program wait states inserted when external space area 2 is accessed
3	W11	1	R/W	Area 1 Wait Control 1 and 0
2	W10	1	R/W	These bits select the number of program wait states when area 1 in external space is accessed while the AST1 bit in ASTCR is set to 1. 00: Program wait not inserted when external space area 1 is accessed 01: 1 program wait state inserted when external space area 1 is accessed 10: 2 program wait states inserted when external space area 1 is accessed 11: 3 program wait states inserted when external space area 1 is accessed

Bit	Bit Name	Initial Value	R/W	Description
1	W01	1	R/W	Area 0 Wait Control 1 and 0
0	W00	1	R/W	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. 00: Program wait not inserted when external space area 0 is accessed 01: 1 program wait state inserted when external space area 0 is accessed 10: 2 program wait states inserted when external space area 0 is accessed 11: 3 program wait states inserted when external space area 0 is accessed

6.3.4 Bus Control Register H (BCRH)

BCRH selects enabling or disabling of idle cycle insertion, and the memory interface for area 0.

Bit	Bit Name	Initial Value	R/W	Description
7	ICIS1	1	R/W	Idle Cycle Insert 1 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. 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
6	ICIS0	1	R/W	Idle Cycle Insert 0 Selects whether or not one idle cycle state is to be inserted between bus cycles when successive external read and write cycles are performed. 0: Idle cycle not inserted in case of successive external read and write cycles 1: Idle cycle inserted in case of successive external read and write cycles
5	BRSTRM	0	R/W	Burst ROM enable Selects whether area 0 is used as a burst ROM interface. 0: Area 0 is basic bus interface 1: Area 0 is burst ROM interface
4	BRSTS1	1	R/W	Burst Cycle Select 1 Selects the number of burst cycles for the burst ROM interface. 0: Burst cycle comprises 1 state 1: Burst cycle comprises 2 states
3	BRSTS0	0	R/W	Burst Cycle Select 0 Selects the number of words that can be accessed in a burst ROM interface burst access. 0: Max. 4 words in burst access 1: Max. 8 words in burst access
2 to 0	—	All 0	R/W	Reserved The write value should always be 0.

6.3.5 Bus Control Register L (BCRL)

BCRL performs selection of the external bus-released state protocol, and enabling or disabling of $\overline{\text{WAIT}}$ pin input.

Bit	Bit Name	Initial Value	R/W	Description
7	BRLE	0	R/W	Bus release enable Enables or disables external bus release. 0: External bus release is disabled. $\overline{\text{BREQ}}$ and $\overline{\text{BACK}}$ can be used as I/O ports. 1: External bus release is enabled.
6	—	0	R/W	Reserved The write value should always be 0.
5	—	0	—	Reserved This bit is always read as 0 and cannot be modified.
4	—	0	R/W	Reserved The write value should always be 0.
3	—	1	R/W	Reserved The write value should always be 1.
2, 1	—	All 0	R/W	Reserved The write value should always be 0.
0	WAITE	0	R/W	WAIT pin enable Selects enabling or disabling of wait input by the $\overline{\text{WAIT}}$ pin. 0: Wait input by $\overline{\text{WAIT}}$ pin disabled. $\overline{\text{WAIT}}$ pin can be used as I/O port. 1: Wait input by $\overline{\text{WAIT}}$ pin enabled.

6.3.6 Pin Function Control Register (PFCR)

PFCR performs address output control in external extended mode.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R/W	Reserved The write value should always be 0.
3	AE3	1/0*	R/W	Address Output Enable 3 to 0 These bits select enabling or disabling of address outputs A8 to A23 in ROMless extended mode and modes with ROM. When a pin is enabled for address output, the address is output regardless of the corresponding DDR setting. When a pin is disabled for address output, it becomes an output port when the corresponding DDR bit is set to 1. 0000: A8 to A23 output disabled (Initial value in modes 6, 7) 0001: A8 output enabled; A9 to A23 output disabled 0010: A8, A9 output enabled; A10 to A23 output disabled 0011: A8 to A10 output enabled; A11 to A23 output disabled 0100: A8 to A11 output enabled; A12 to A23 output disabled 0101: A8 to A12 output enabled; A13 to A23 output disabled 0110: A8 to A13 output enabled; A14 to A23 output disabled 0111: A8 to A14 output enabled; A15 to A23 output disabled 1000: A8 t o A15 output enabled; A16 to A23 output disabled 1001: A8 to A16 output enabled; A17 to A23 output disabled 1010: A8 to A17 output enabled; A18 to A23 output disabled 1011: A8 to A18 output enabled; A19 to A23 output disabled 1100: A8 to A19 output enabled; A20 to A23 output disabled 1101: A8 to A20 output enabled; A21 to A23 output disabled (Initial value in modes 4, 5) 1110: A8 to A21 output enabled; A22, A23 output disabled 1111: A8 to A23 output enabled
2	AE2	1/0*	R/W	
1	AE1	0	R/W	
0	AE0	1/0*	R/W	

Note: * In modes 4 and 5, initial value of each bit is 1. In modes 6 and 7, initial value of each bit is 0.

6.4 Bus Control

6.4.1 Area Divisions

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. In normal mode*, it controls a 64-kbyte address space comprising part of area 0.

Figure 6.2 shows an outline of the memory map.

Chip select signals ($\overline{CS0}$ to $\overline{CS7}$) can be output for each area.

Note: * Not available in this LSI.

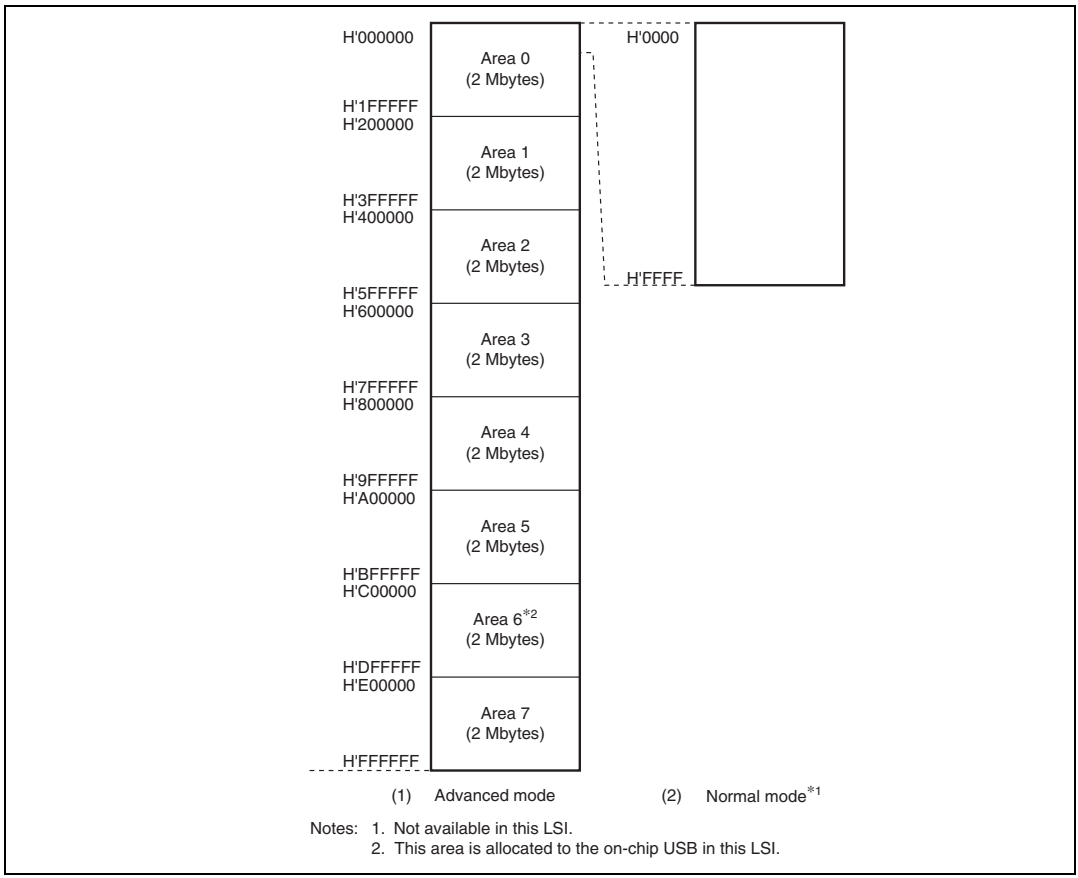


Figure 6.2 Overview of Area Divisions

6.4.2 Bus Specifications

The external space bus specifications consist of three elements: bus width, number of access states, and number of program wait states.

The bus width and number of access states for on-chip memory and internal I/O registers except for the on-chip USB 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 a 16-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 set. When the burst ROM interface is designated, 16-bit bus mode is always set. 8-bit bus mode should be set for area 6 in this LSI.

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 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.

Area 6 should be set to function as a 3-state access space in this LSI.

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.

The number of program wait states in area 6 should be set to 0 in this LSI.

Table 6.2 Bus Specifications for Each Area (Basic Bus Interface)

ABWCR	ASTCR	WCRH, WCRL		Bus Specifications (Basic Bus Interface)		
ABWn	ASTn	Wn1	Wn0	Bus Width	Number of Access States	Number of 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

6.4.3 Bus Interface for Each Area

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 (see section 6.6, Basic Bus 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 ROM-disabled extended mode, all of area 0 is external space. In ROM-enabled extended mode, the space excluding on-chip ROM is external space.

When area 0 external space is accessed, the $\overline{CS0}$ signal can be output.

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

Areas 1 to 6: In external extended mode, all of areas 1 to 6 are external spaces. When areas 1 to 6 external space are accessed, the $\overline{CS1}$ to $\overline{CS6}$ pin signals respectively can be output. Only the basic bus interface can be used for areas 1 to 6. Area 6 is only for the on-chip USB. For details, see section 15, Universal Serial Bus Interface (USB).

Area 7: Area 7 includes the on-chip RAM and internal l/O registers. In external extended mode, the space excluding the on-chip RAM and internal l/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 $\overline{CS7}$ signal can be output.
Only the basic bus interface can be used for the area 7.

6.4.4 Chip Select Signals

This LSI can output chip select signals ($\overline{CS0}$ to $\overline{CS7}$) 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 $\overline{CSn}$ (n = 0 to 7) output timing. Enabling or disabling of the $\overline{CSn}$ signal is performed by setting the data direction register (DDR) for the port corresponding to the particular $\overline{CSn}$ pin.

In ROM-disabled extended mode, the $\overline{CS0}$ pin is placed in the output state after a power-on reset. Pins $\overline{CS1}$ to $\overline{CS7}$ are placed in the input state after a power-on reset, and so the corresponding DDR should be set to 1 when outputting signals $\overline{CS1}$ to $\overline{CS7}$.

In ROM-enabled extended mode, pins $\overline{CS0}$ to $\overline{CS7}$ are all placed in the input state after a power-on reset, and so the corresponding DDR should be set to 1 when outputting signals $\overline{CS0}$ to $\overline{CS7}$. For details, see section 9, I/O Ports.

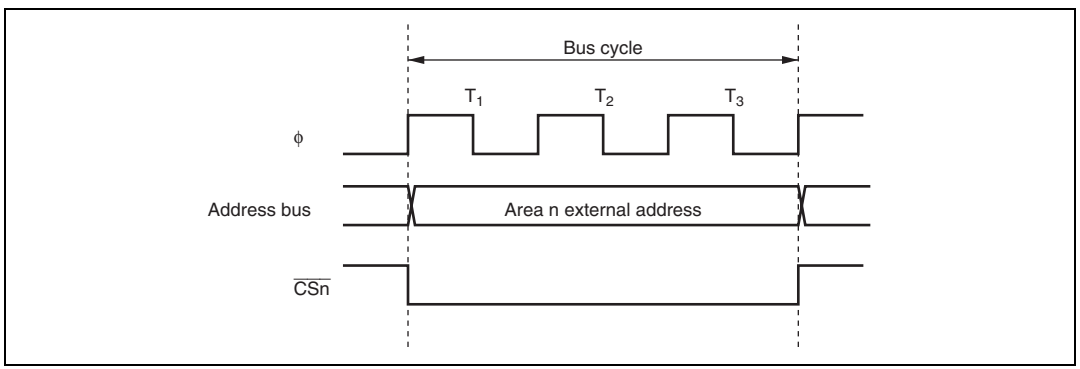


Figure 6.3 $\overline{CSn}$ Signal Output Timing (n = 0 to 7)

6.5 Basic Timing

The CPU is driven by a system clock (ϕ), 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 peripheral modules, and the external address space.

6.5.1 On-Chip Memory (ROM, RAM) Access Timing

On-chip memory is accessed in one state. The data bus is 16 bits wide, permitting both byte and word transfer instruction. Figure 6.4 shows the on-chip memory access cycle. Figure 6.5 shows the pin states.

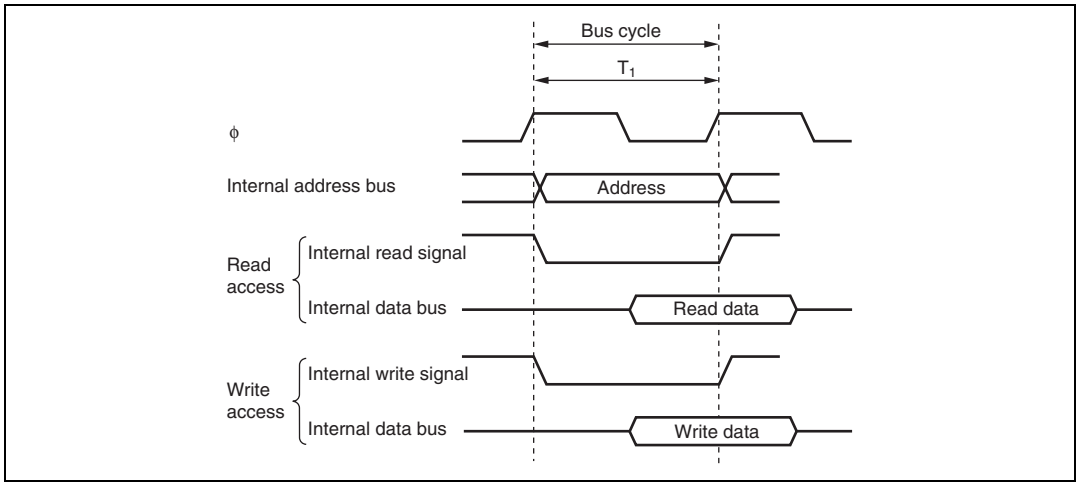


Figure 6.4 On-Chip Memory Access Cycle

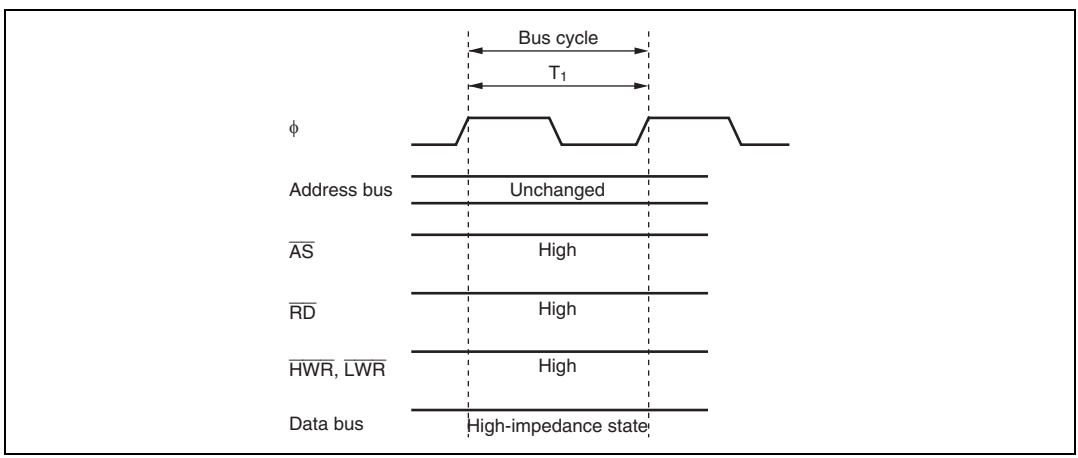


Figure 6.5 Pin States during On-Chip Memory Access

6.5.2 On-Chip Peripheral Module Access Timing

The on-chip peripheral modules are accessed in two states except on-chip USB. The data bus is either 8 bits or 16 bits wide, depending on the particular internal I/O register being accessed. Figure 6.6 shows the access timing for the on-chip peripheral modules. Figure 6.7 shows the pin states.

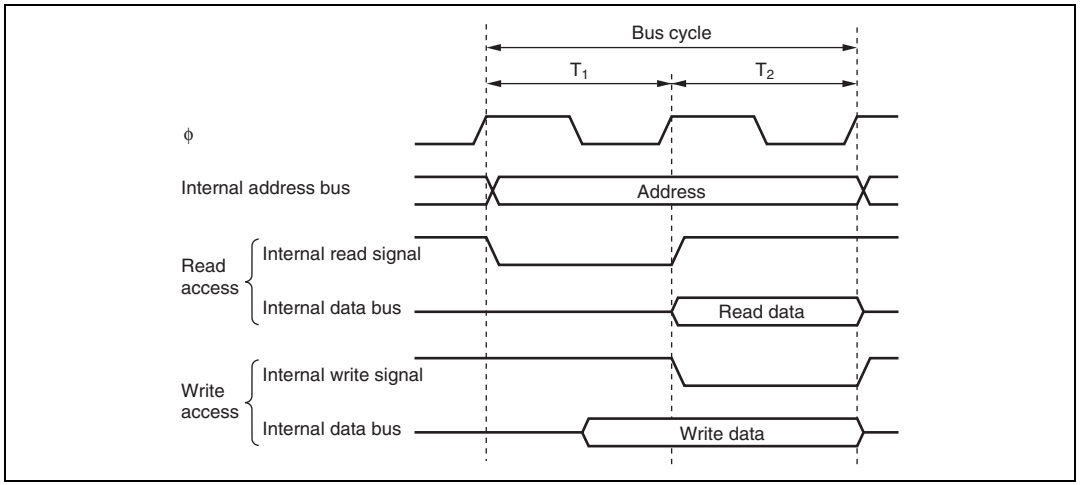


Figure 6.6 On-Chip Peripheral Module Access Cycle

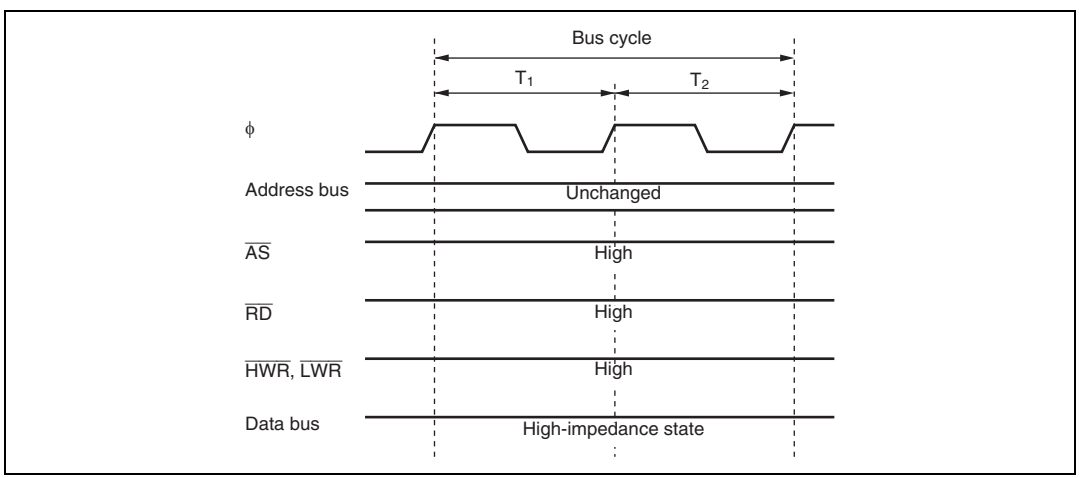


Figure 6.7 Pin States during On-Chip Peripheral Module Access

6.5.3 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 6.6.3, Basic Timing.

6.6 Basic Bus Interface

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

6.6.1 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 (D15 to D8) or lower data bus (D7 to D0) is used according to the bus specifications for the area being accessed (8-bit access space or 16-bit access space) and the data size.

8-Bit Access Space: Figure 6.8 illustrates data alignment control for the 8-bit access space. With the 8-bit access space, the upper data bus (D15 to D8) 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.

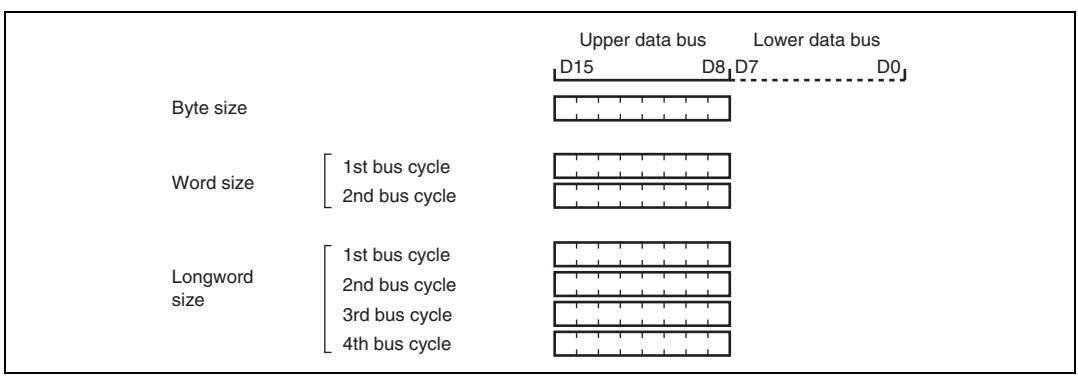


Figure 6.8 Access Sizes and Data Alignment Control (8-Bit Access Space)

16-Bit Access Space: Figure 6.9 illustrates data alignment control for the 16-bit access space. With the 16-bit access space, the upper data bus (D15 to D8) and lower data bus (D7 to D0) are used 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.

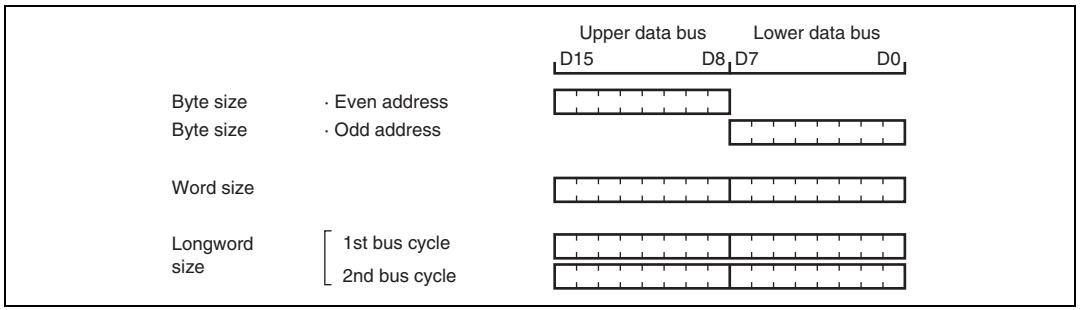


Figure 6.9 Access Sizes and Data Alignment Control (16-Bit Access Space)

6.6.2 Valid Strobes

Table 6.3 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 6.3 Data Buses Used and Valid Strobes

Area	Access Size	Read/ Write	Address	Valid Strobe	Upper Data Bus (D15 to D8)	Lower Data Bus (D7 to D0)
8-bit access space	Byte	Read	—	$\overline{RD}$	Valid	Invalid
		Write	—	$\overline{HWR}$		Hi-Z
16-bit access space	Byte	Read	Even	$\overline{RD}$	Valid	Invalid
			Odd		Invalid	Valid
		Write	Even	$\overline{HWR}$	Valid	Hi-Z
			Odd	$\overline{LWR}$	Hi-Z	Valid
	Word	Read	—	$\overline{RD}$	Valid	Valid
		Write	—	$\overline{HWR}$, $\overline{LWR}$	Valid	Valid

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

6.6.3 Basic Timing

8-Bit 2-State Access Space: Figure 6.10 shows the bus timing for an 8-bit 2-state access space. When an 8-bit access space is accessed, the upper half (D15 to D8) of the data bus is used.

Wait states cannot be inserted.

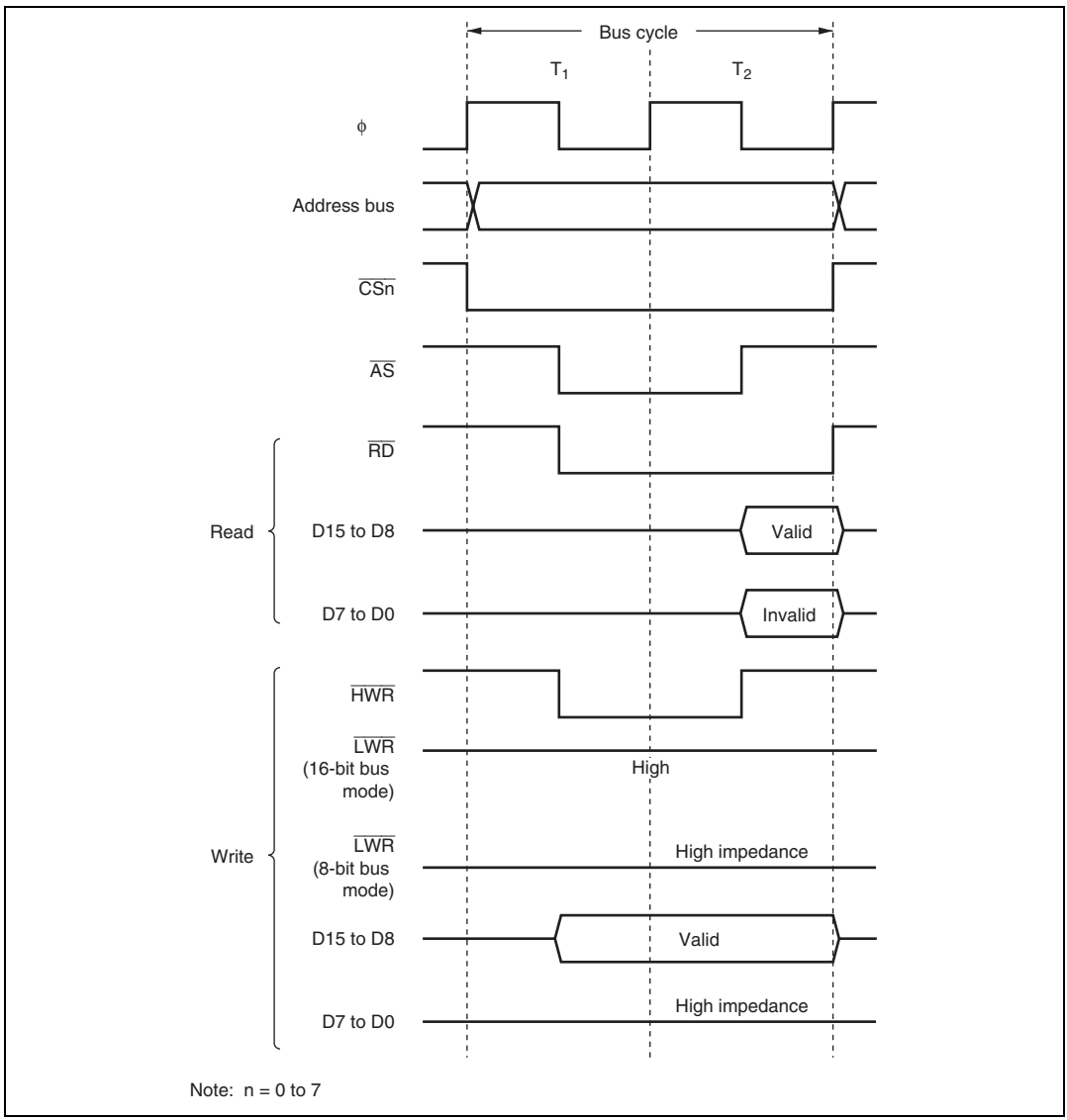


Figure 6.10 Bus Timing for 8-Bit 2-State Access Space

8-Bit 3-State Access Space (Except Area 6): Figure 6.11 shows the bus timing for an 8-bit 3-state access space. When an 8-bit access space is accessed, the upper half (D15 to D8) of the data bus is used.

Wait states can be inserted.

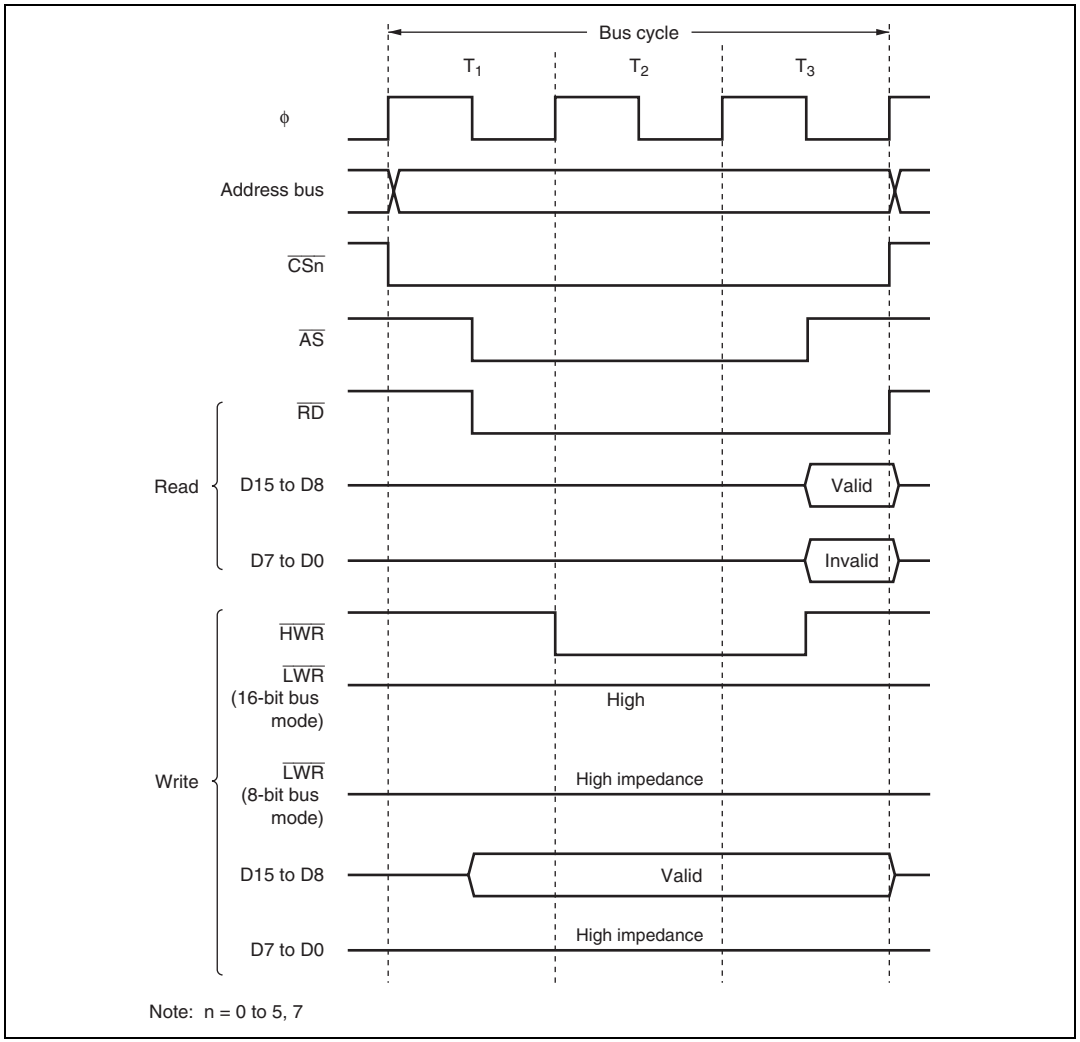


Figure 6.11 Bus Timing for 8-Bit 3-State Access Space (Except Area 6)

8-Bit 3-State Access Space (Area 6): Figure 6.12 shows the bus timing for area 6. When area 6 is accessed, the data bus cannot be used.

Wait states cannot be inserted.

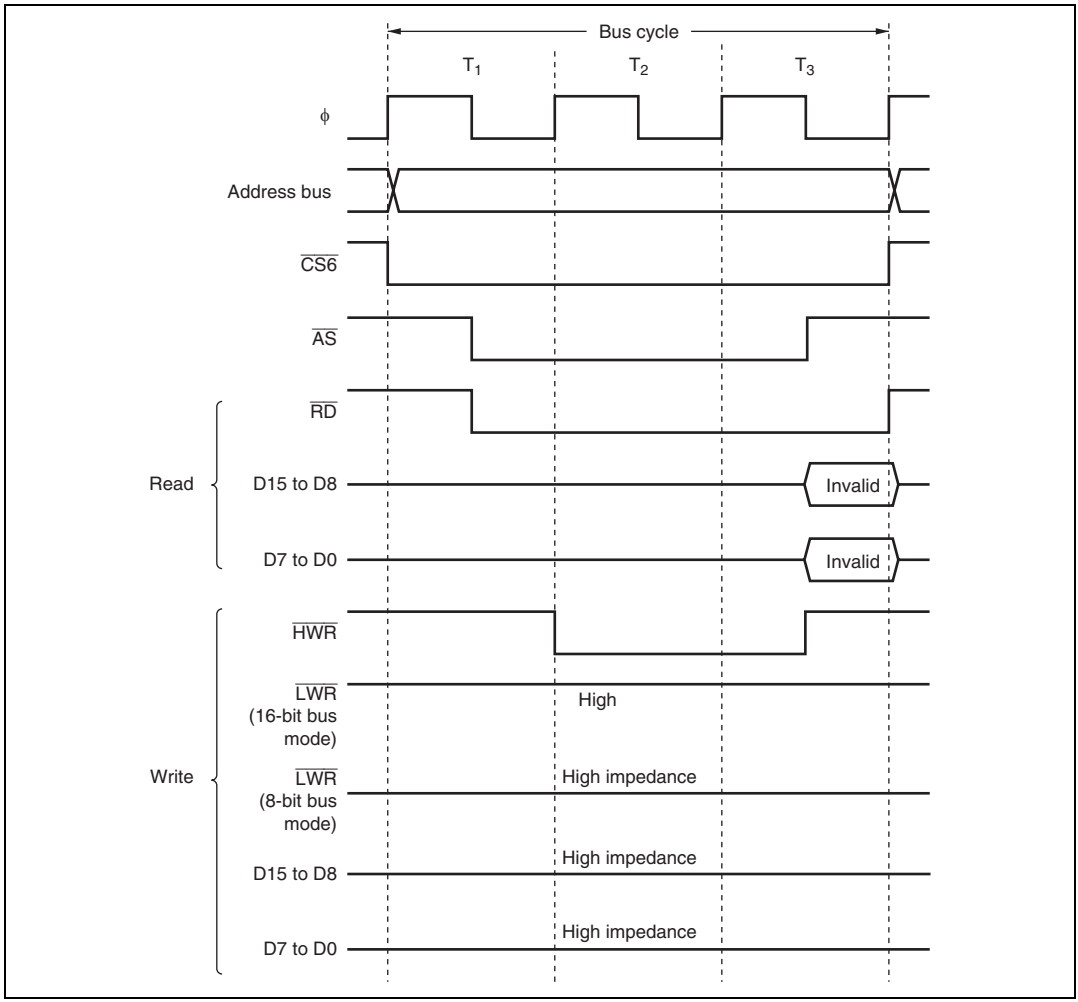


Figure 6.12 Bus Timing for Area 6

16-Bit 2-State Access Space: Figures 6.13 to 6.15 show bus timings for a 16-bit 2-state access space. When a 16-bit access space is accessed, the upper half (D15 to D8) of the data bus is used for the even address, and the lower half (D7 to D0) for the odd address.

Wait states cannot be inserted.

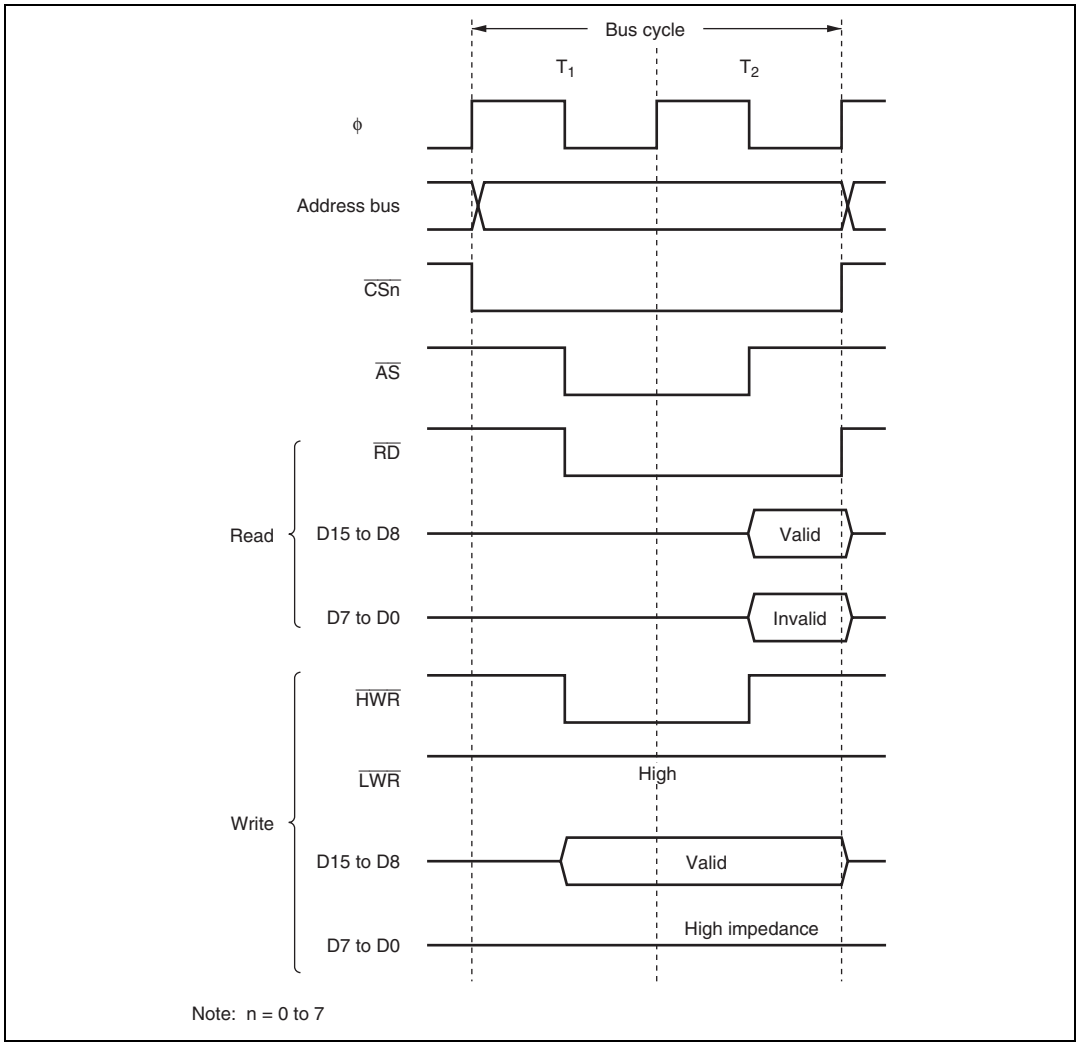


Figure 6.13 Bus Timing for 16-Bit 2-State Access Space (1) (Even Address Byte Access)

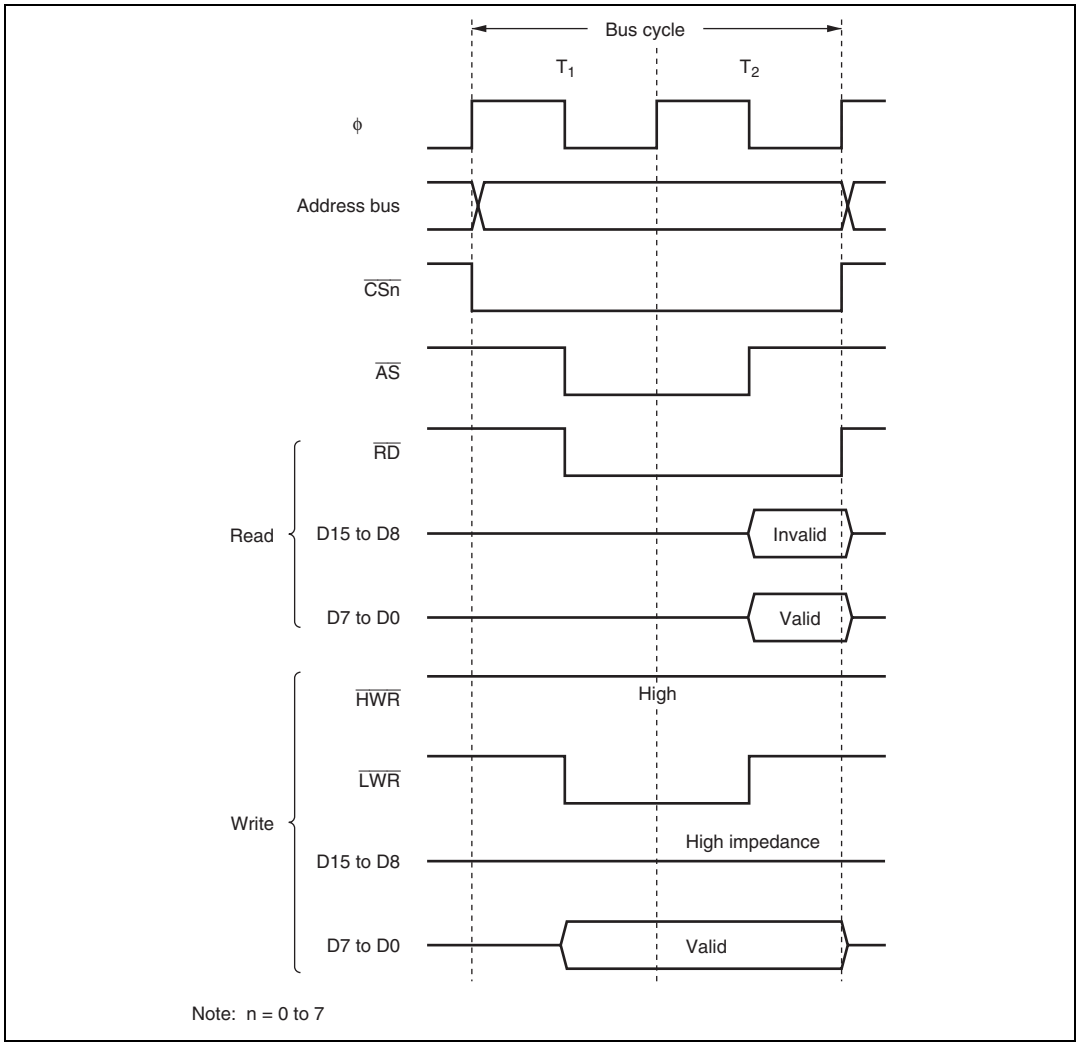


Figure 6.14 Bus Timing for 16-Bit 2-State Access Space (2) (Odd Address Byte Access)

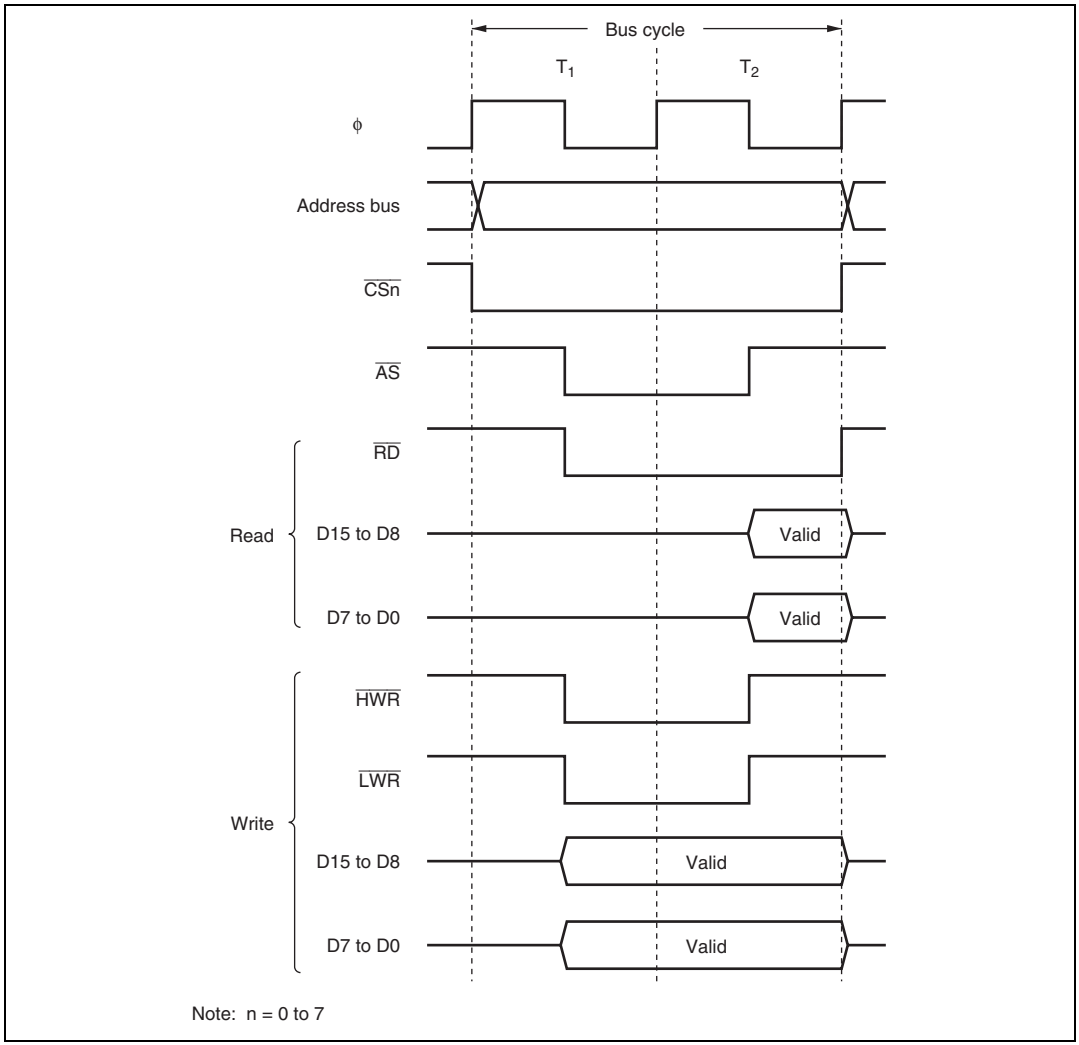


Figure 6.15 Bus Timing for 16-Bit 2-State Access Space (3) (Word Access)

16-Bit 3-State Access Space: Figures 6.16 to 6.18 show bus timings for a 16-bit 3-state access space. When a 16-bit access space is accessed, the upper half (D15 to D8) of the data bus is used for the even address, and the lower half (D7 to D0) for the odd address.

Wait states can be inserted.

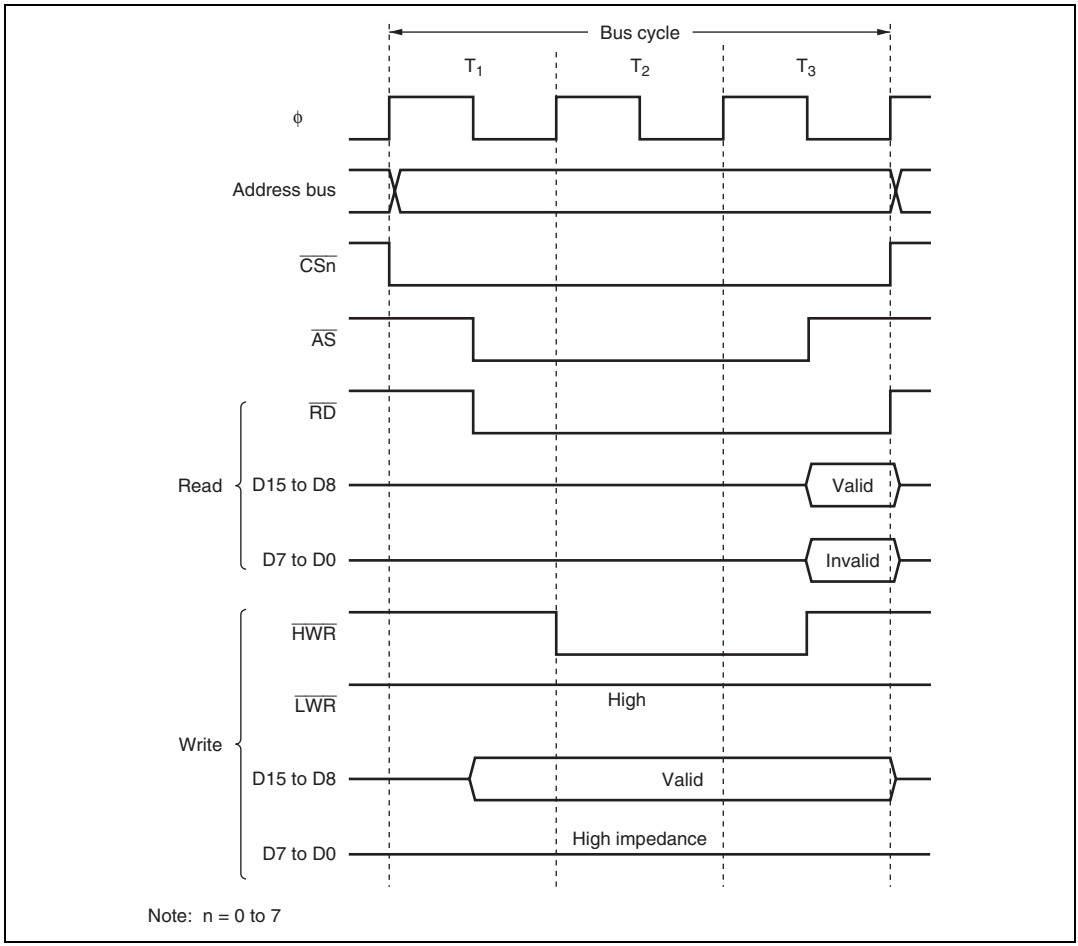


Figure 6.16 Bus Timing for 16-Bit 3-State Access Space (1) (Even Address Byte Access)

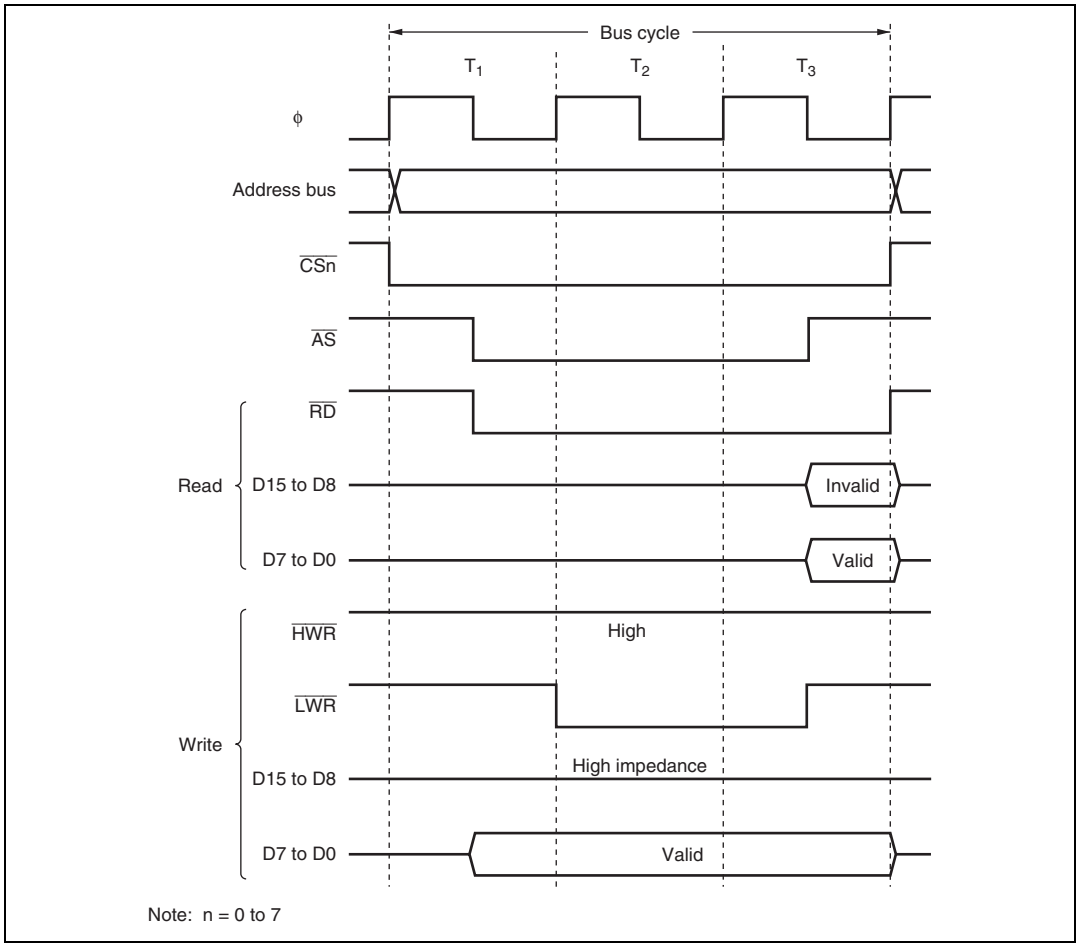


Figure 6.17 Bus Timing for 16-Bit 3-State Access Space (2) (Odd Address Byte Access)

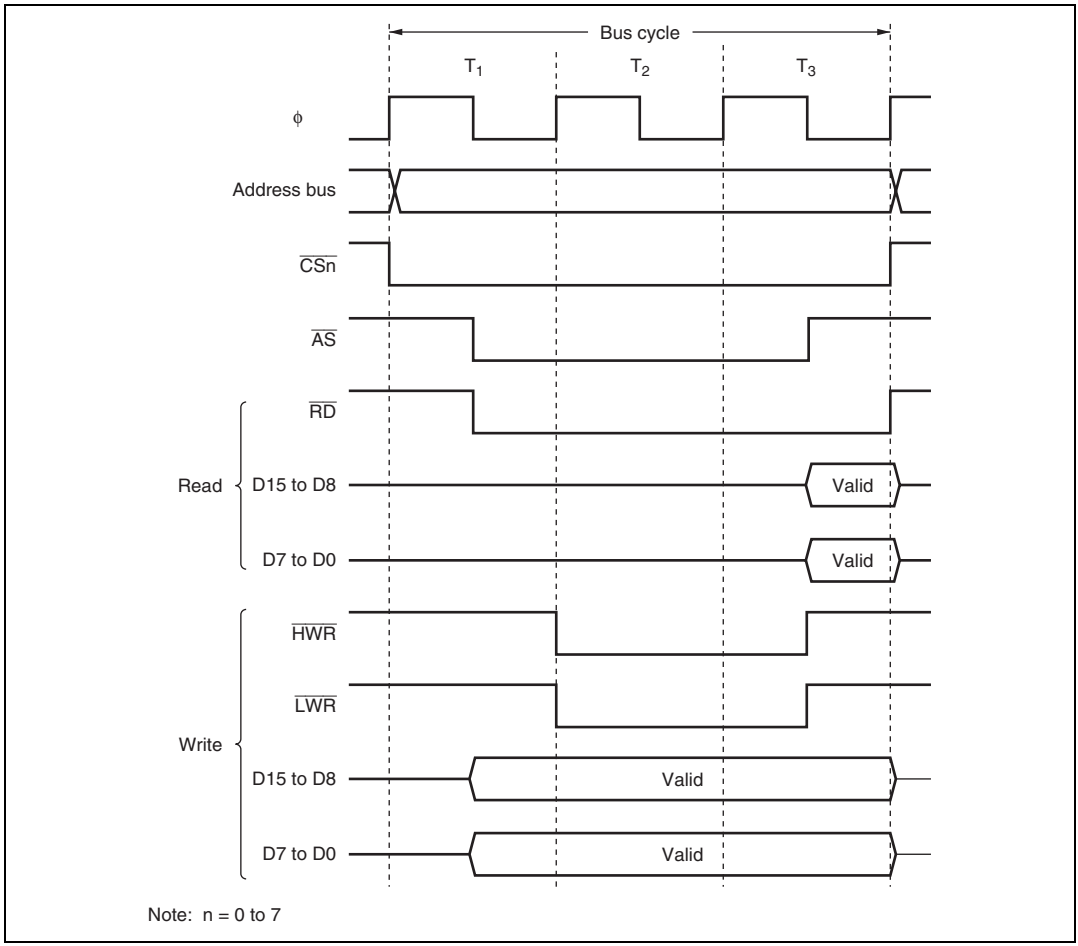


Figure 6.18 Bus Timing for 16-Bit 3-State Access Space (3) (Word Access)

6.6.4 Wait Control

When accessing external space, this LSI 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 of WCRH and WCRL.

Pin Wait Insertion: Setting the WAITE bit in BCRH to 1 enables wait insertion by means of the $\overline{\text{WAIT}}$ pin. When external space is accessed in this state, program wait insertion is first carried out according to 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, a T_w state is inserted. If the $\overline{\text{WAIT}}$ pin is held low, T_w states are inserted until it goes high.

Figure 6.19 shows an example of wait state insertion timing.

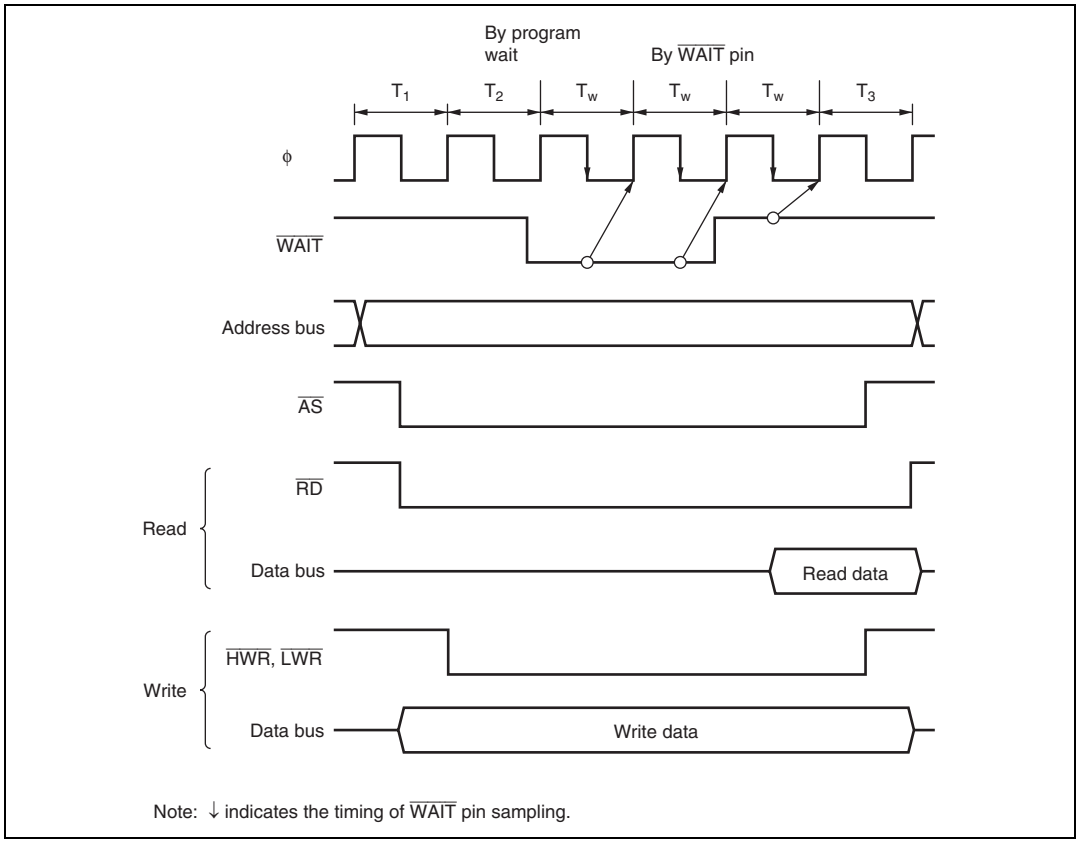


Figure 6.19 Example of Wait State Insertion Timing

6.7 Burst ROM Interface

With this LSI, external space area 0 can be designated as burst ROM space, and burst ROM interfacing can be performed. The burst ROM space interface enables 16-bit configuration 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 of 4 words or 8 words can be performed for CPU instruction fetches only. One or two states can be selected for burst access.

6.7.1 Basic Timing

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

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

The basic access timing for burst ROM space is shown in figures 6.20 and 6.21. The timing shown in figure 6.20 is for the case where the AST0 and BRSTS1 bits are both set to 1, and that in figure 6.21 is for the case where both these bits are cleared to 0.

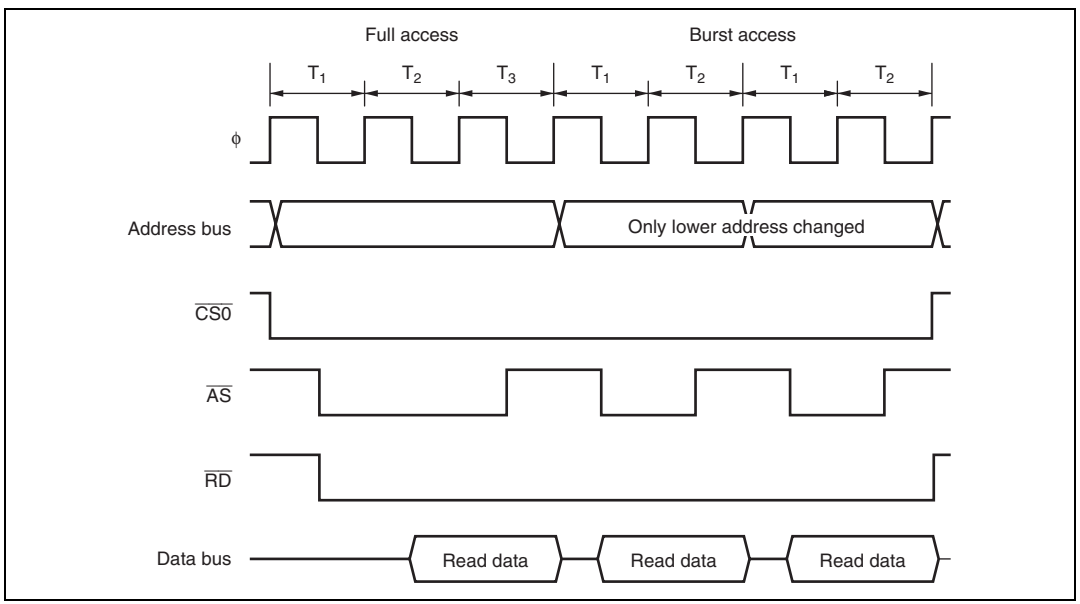


Figure 6.20 Example of Burst ROM Access Timing (When AST0 = BRSTS1 = 1)

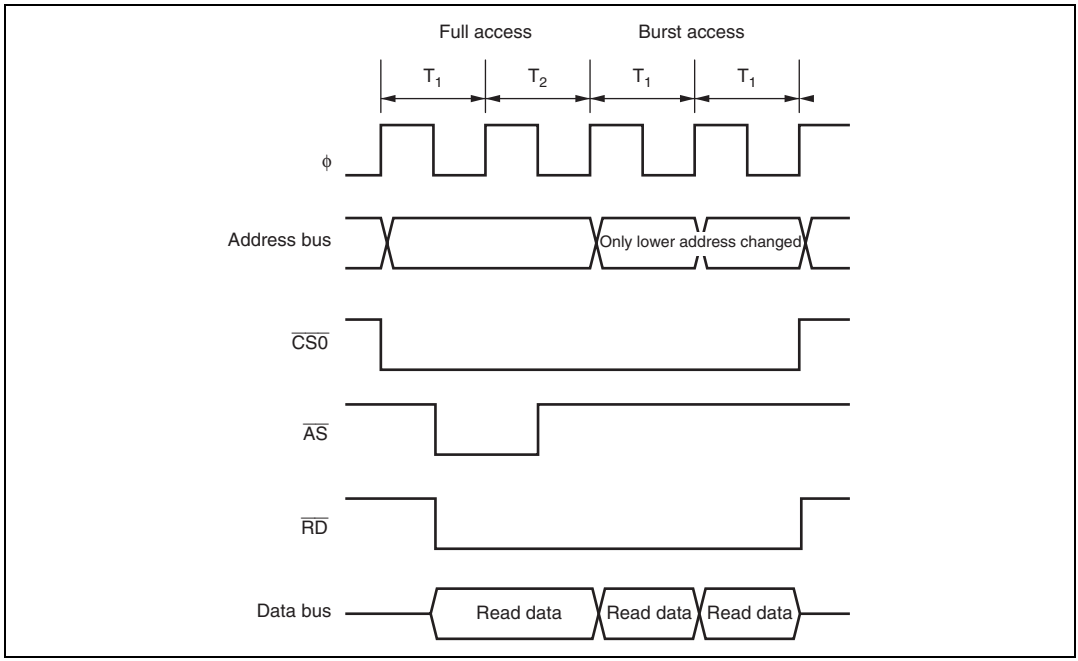


Figure 6.21 Example of Burst ROM Access Timing (When AST0 = BRSTS1 = 0)

6.7.2 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) of the burst ROM interface. See section 6.6.4, Wait Control.

Wait states cannot be inserted in a burst cycle.

6.8 Idle Cycle

When this LSI 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 between 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, with a long output floating time, and high-speed memory, I/O interfaces, and so on.

Consecutive Reads between Different Areas: If consecutive reads between 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.

Figure 6.22 shows an example of the operation in this case. In this example, bus cycle A is a read cycle from ROM with a long output floating time, and bus cycle B is a read cycle from SRAM, each being located in a different area. In (a), an idle cycle is not inserted, and a collision occurs in 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.

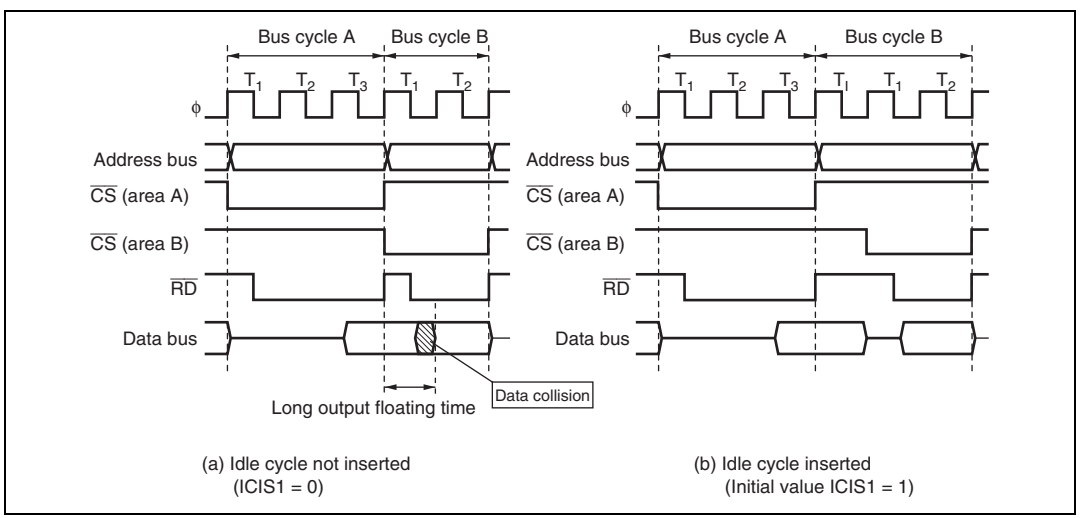


Figure 6.22 Example of Idle Cycle Operation (1)

Write after Read: If an external write occurs after an external read while the ICIS0 bit in BCRH is set to 1, an idle cycle is inserted at the start of the write cycle.

Figure 6.23 shows an example of the operation in this case. In this example, bus cycle A is a read cycle from 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 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.

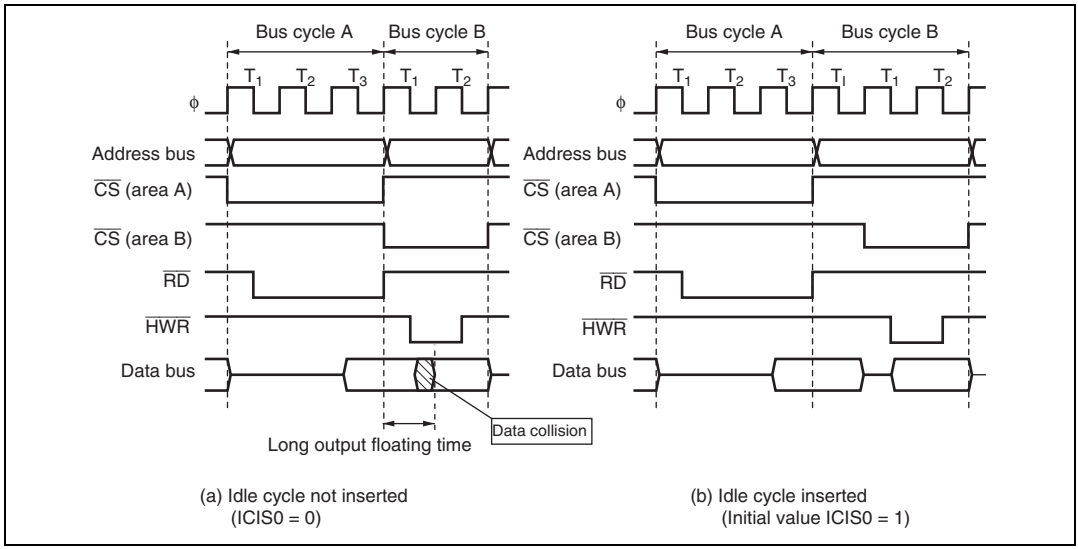


Figure 6.23 Example of Idle Cycle Operation (2)

Relationship between Chip Select ($\overline{CS}$) Signal and Read ($\overline{RD}$) Signal: Depending on the system's load conditions, the $\overline{RD}$ signal may lag behind the $\overline{CS}$ signal. An example is shown in figure 6.24.

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.

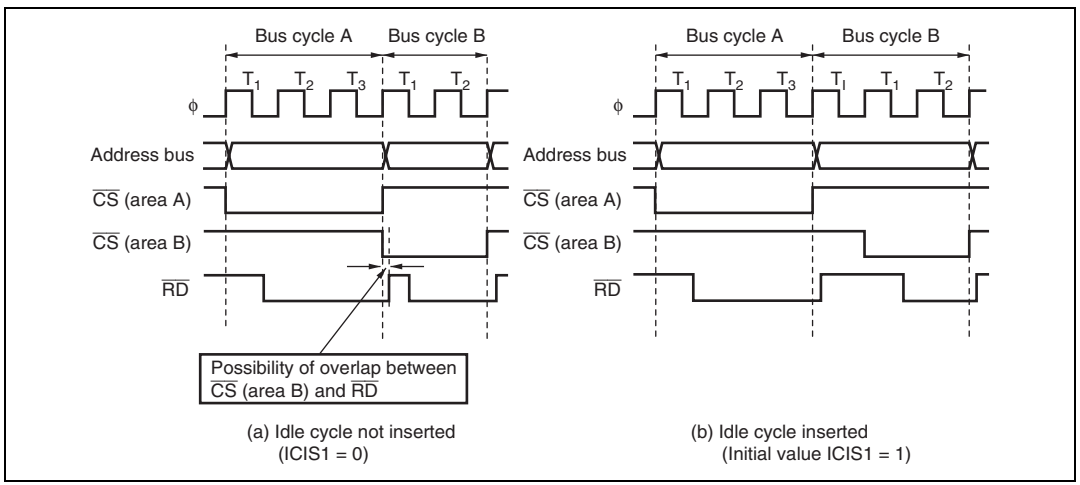


Figure 6.24 Relationship between Chip Select ($\overline{CS}$) and Read ($\overline{RD}$)

Table 6.4 shows pin states in an idle cycle.

Table 6.4 Pin States in Idle Cycle

Pins	Pin State
A23 to A0	Contents of next bus cycle
D15 to D0	High impedance
$\overline{CSn}$	High
$\overline{AS}$	High
$\overline{RD}$	High
$\overline{HWR}$	High
$\overline{LWR}$	High

6.9 Bus Release

This LSI 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.

In external extended 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 this LSI. 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.

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

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

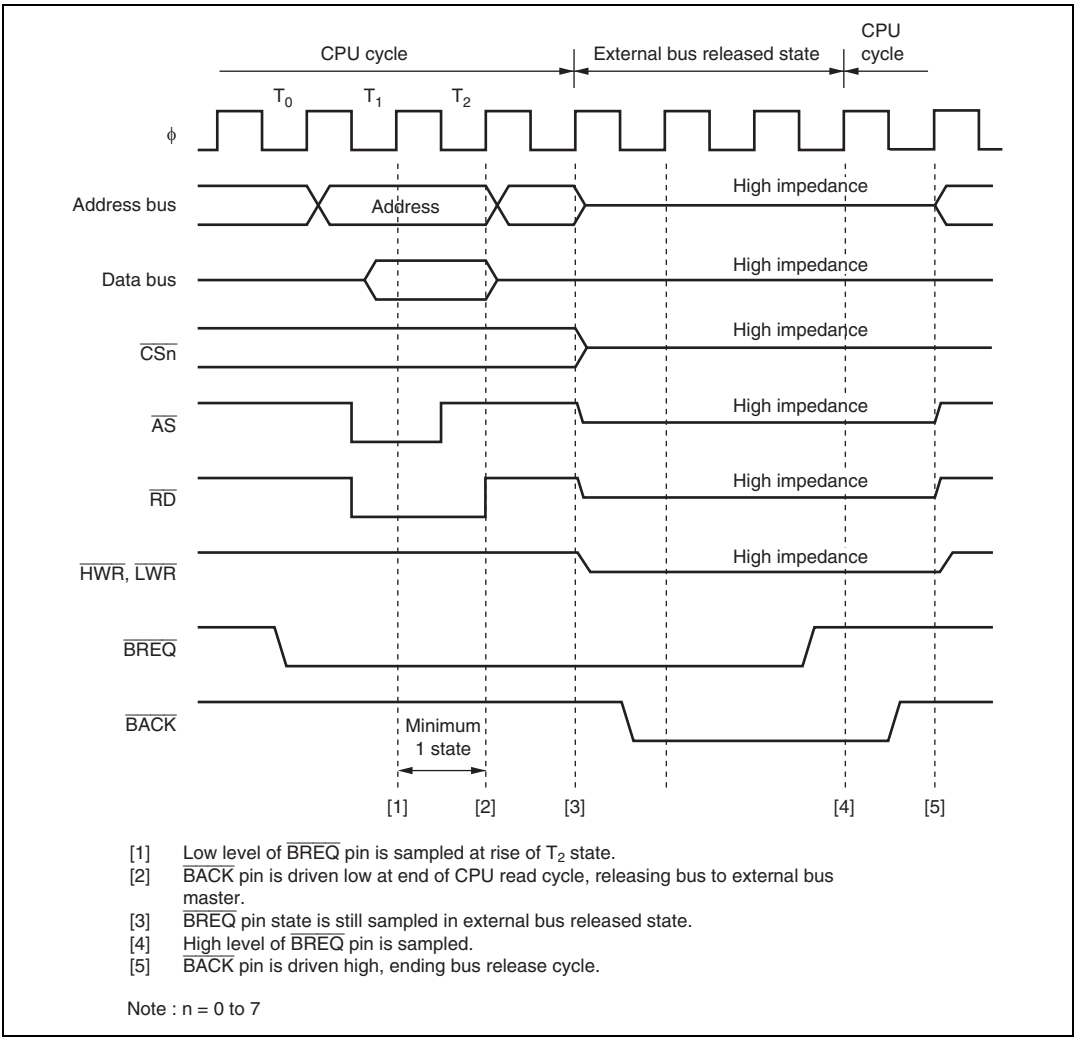
(High) External bus release > Internal bus master external access (Low)

Table 6.5 shows pin states in the external bus released state.

Table 6.5 Pin States in Bus Released State

Pins	Pin State
A23 to A0	High impedance
D15 to D0	High impedance
$\overline{\text{CSn}}$	High impedance
$\overline{\text{AS}}$	High impedance
$\overline{\text{RD}}$	High impedance
$\overline{\text{HWR}}$	High impedance
$\overline{\text{LWR}}$	High impedance

Figure 6.25 shows the timing for transition to the bus-released state.



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

Note : n = 0 to 7

Figure 6.25 Bus-Released State Transition Timing

6.9.1 Notes on Bus Release

The external bus release function halts when a transition is made to sleep mode while MSTPCR is set to H'FFFFFF. To use the external bus release function in sleep mode, do not set MSTPCR to H'FFFFFF.

6.10 Bus Arbitration

This LSI has a bus arbiter that arbitrates bus master operations.

There are three bus masters, the CPU, DMAC, and DTC, 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.

6.10.1 Operation

The bus arbiter detects 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 internal bus access by an internal bus master, and external bus release, can be executed in parallel.

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

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

6.10.2 Bus Transfer Timing

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 DMAC and DTC, 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 operations.
- 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 a USB request in normal mode, and in short address mode or 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.

6.10.3 External Bus Release Usage Note

External bus release can be performed on completion of an external bus cycle. The $\overline{CS}$ signal remains low until the end of the external bus cycle. Therefore, when external bus release is performed, the $\overline{CS}$ signal may change from the low level to the high-impedance state.

6.11 Resets and the Bus Controller

In a power-on reset, this LSI, including the bus controller, enters the reset state at that point, and an executing bus cycle is discontinued.

In a manual reset, the bus controller's registers and internal state are maintained, and an executing external bus cycle is completed. In this case, $\overline{WAIT}$ input is ignored and write data is not guaranteed.

Section 7 DMA Controller (DMAC)

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

7.1 Features

The features of the DMAC are listed below.

- Choice of short address mode or full address mode
  (1) Short address mode
  — Maximum of 4 channels can be used
  — Choice of dual 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
  — Choice of sequential mode, idle mode, or repeat mode for dual address mode
  (2) 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, USB request, auto-request (depending on transfer mode)
  — 16-bit timer-pulse unit (TPU) compare match/input capture interrupts
  — Serial communication interface (SCI_0, SCI_1) transmission complete interrupt, reception complete interrupt
  — A/D conversion end interrupt
  — USB request
  — Auto-request
- Module stop mode can be set

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

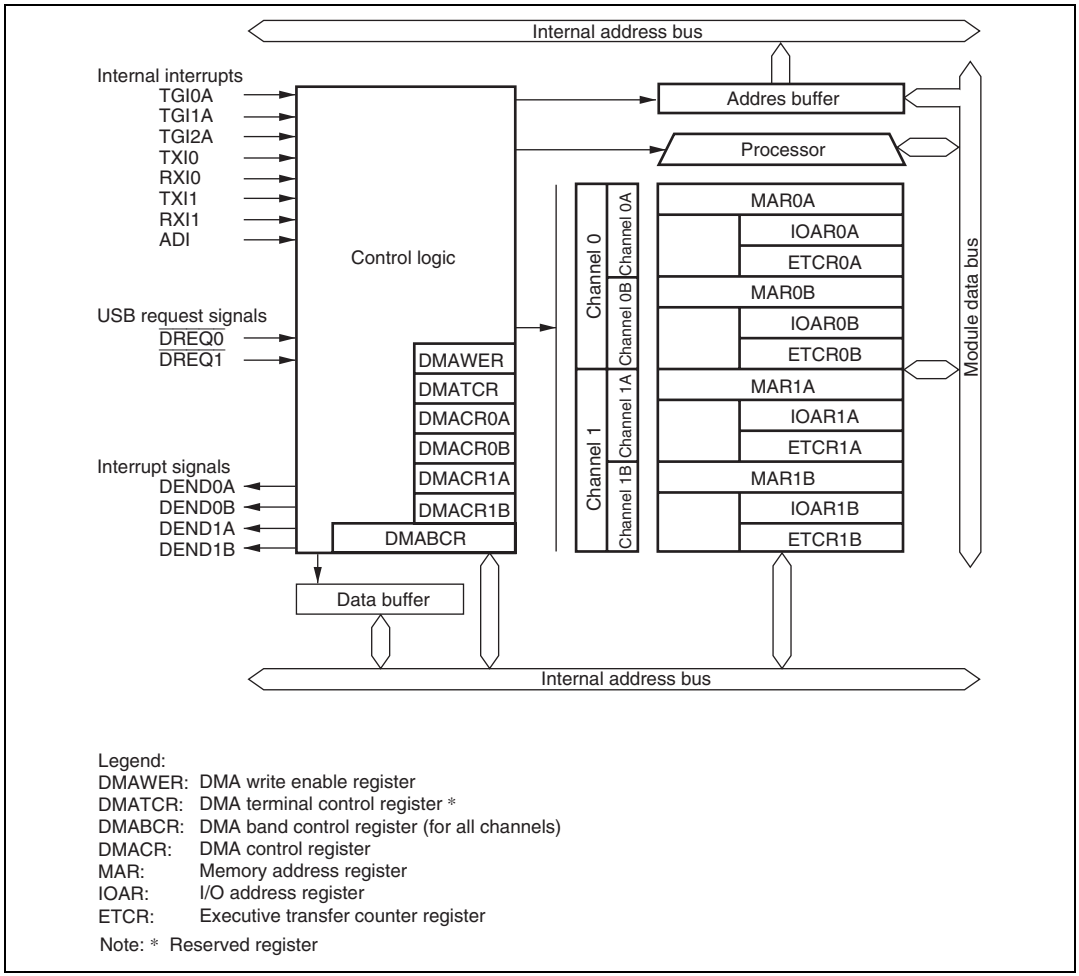


Figure 7.1 Block Diagram of DMAC

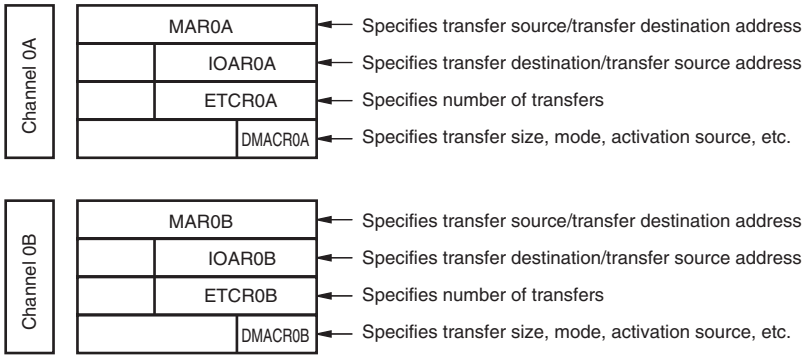
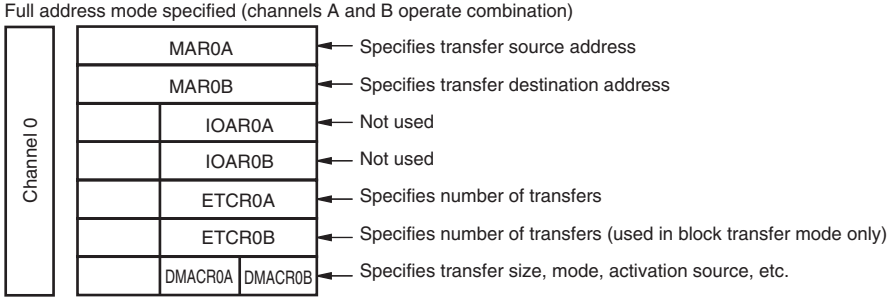
7.2 Register Configuration

The DMAC registers are listed below.

- Memory address register 0A (MAR0A)
- I/O address register 0A (IOAR0A)
- Transfer count register 0A (ETCR0A)
- Memory address register 0B (MAR0B)
- I/O address register 0B (IOAR0B)
- Transfer count register 0B (ETCR0B)
- Memory address register 1A (MAR1A)
- I/O address register 1A (IOAR1A)
- Transfer count register 1A (ETCR1A)
- Memory address register 1B (MAR1B)
- I/O address register 1B (IOAR1B)
- Transfer count register 1B (ETCR1B)
- DMA write enable register (DMAWER)
- DMA control register 0A (DMACR0A)
- DMA control register 0B (DMACR0B)
- DMA control register 1A (DMACR1A)
- DMA control register 1B (DMACR1B)
- DMA band control register (DMABCR)

The DMAC register functions differs depending on the address modes: short address mode and full address mode. The DMAC register functions are described in each address mode. Short address mode or full address mode can be selected for channels 1 and 0 independently by means of bits FAE1 and FAE0.

Table 7.1 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 combination) 

7.3 Register Descriptions

7.3.1 Memory Address Registers (MAR)

- Short Address Mode

  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 7.3.4, DMA Control Register (DMACR). MAR is not initialized by a reset or in standby mode.
- Full Address Mode

  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)

- Short Address Mode

  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 not incremented or decremented each time a transfer is executed, so that the address specified by IOAR is fixed. IOAR is not initialized by a reset or in standby mode.
- Full Address Mode

  IOAR is not used in full address mode transfer.

7.3.3 Execute Transfer Count Register (ETCR)

- Short Address Mode

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

  — Sequential Mode and Idle Mode

  In sequential mode and idle mode, ETCR functions as a 16-bit transfer counter (with a count range of 1 to 65,536). ETCR is decremented by 1 each time a transfer is performed, and when the count reaches H'0000, the DTE bit in DMABCR is cleared, and transfer ends.

  — Repeat Mode

  In repeat mode, ETCR functions as an 8-bit 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.

- Full Address Mode

  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

  (a) ETCRA

  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.

  (b) ETCRB

  ETCRB is not used in normal mode.

  — Block Transfer Mode

  (a) ETCRA

  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.

  (b) ETCRB

  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.

7.3.4 DMA Control Register (DMACR)

DMACR controls the operation of each DMAC channel.

- Short Address Mode (common to DMACRA and DMACRB)

Bit	Bit Name	Initial Value	R/W	Description
7	DTSZ	0	R/W	Data Transfer Size Selects the size of data to be transferred at one time. 0: Byte-size transfer 1: Word-size transfer
6	DTID	0	R/W	Data Transfer Increment/Decrement Selects incrementing or decrementing of MAR every data transfer in sequential mode or repeat mode. In idle mode, MAR is neither incremented nor decremented. 0: MAR is incremented after a data transfer • 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	Bit Name	Initial Value	R/W	Description
5	RPE	0	R/W	Repeat Enable Used in combination with the DTIE bit in DMABCR to select the mode (sequential, idle, or repeat) in which transfer is to be performed. RPE DTIE 0 0: Transfer in sequential mode (no transfer end interrupt) 0 1: Transfer in sequential mode (with transfer end interrupt) 1 0: Transfer in repeat mode (no transfer end interrupt) 1 1: Transfer in idle mode (with transfer end interrupt) Note: For details of operation in sequential, idle, and repeat mode, see section 7.4.2, Sequential Mode, section 7.4.3, Idle Mode, and section 7.4.4, Repeat Mode.
4	DTDIR	0	R/W	Data Transfer Direction Specifies the data transfer direction (source or destination). 0: Transfer with MAR as source address and IOAR as destination address 1: Transfer with IOAR as source address and MAR as destination address

Bit	Bit Name	Initial Value	R/W	Description
3	DTF3	0	R/W	Data Transfer Factor
2	DTF2	0	R/W	These bits select the data transfer factor (activation source).
1	DTF1	0	R/W	0000: —
0	DTF0	0	R/W	0001: Activated by A/D conversion end interrupt
				0010: —
				0011: —
				0100: Activated by SCI channel 0 transmission complete interrupt
				0101: Activated by SCI channel 0 reception complete interrupt
				0110: Activated by SCI channel 1 transmission complete interrupt
				0111: Activated by SCI channel 1 reception complete interrupt
				1000: Activated by TPU channel 0 compare match/input capture A interrupt
				1001: Activated by TPU channel 1 compare match/input capture A interrupt
				1010: Activated by TPU channel 2 compare match/input capture A interrupt
				1011: —
				1100: —
				1101: —
				1110: —
				1111: —

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 7.4.10, DMAC Multi-Channel Operation.

- Full Address Mode (DMACRA)

Bit	Bit Name	Initial Value	R/W	Description
15	DTSZ	0	R/W	Data Transfer Size Selects the size of data to be transferred at one time. 0: Byte-size transfer 1: Word-size transfer
14	SAID	0	R/W	Source Address Increment/Decrement
13	SAIDE	0	R/W	Source Address Increment/Decrement Enable These bits specify whether source address register MARA is to be incremented, decremented, or left unchanged, when data transfer is performed. 00: MARA is fixed 01: 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 10: MARA is fixed 11: 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	Bit Name	Initial Value	R/W	Description
12	BLKDIR	0	R/W	Block Direction
11	BLKE	0	R/W	Block Enable 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. 00: Transfer in normal mode 01: Transfer in block transfer mode, destination side is block area 10: Transfer in normal mode 11: Transfer in block transfer mode, source side is block area For operation in normal mode and block transfer mode, see section 7.4, Operation.
10 to 8	—	All 0	R/W	Reserved Although these bits are readable/writable, only 0 should be written here.

- Full Address Mode (DMACRB)

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R/W	Reserved Although this bit is readable/writable, only 0 should be written here.
6	DAID	0	R/W	Destination Address Increment/Decrement
5	DAIDE	0	R/W	Destination Address Increment/Decrement Enable These bits specify whether destination address register MARB is to be incremented, decremented, or left unchanged, when data transfer is performed. 00: MARB is fixed 01: 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 10: MARB is fixed 11: 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
4	—	0	R/W	Reserved Although this bit is readable/writable, only 0 should be written here.

Bit	Bit Name	Initial Value	R/W	Description
3	DTF3	0	R/W	Data Transfer Factor
2	DTF2	0	R/W	These bits select the data transfer factor (activation source).
1	DTF1	0	R/W	In normal mode
0	DTF0	0	R/W	0000: — 0001: — 0010: — 0011: — 010×: — 0110: Auto-request (cycle steal) 0111: Auto-request (burst) 1×××: — In block transfer mode 0000: — 0001: Activated by A/D conversion end interrupt 0010: — 0011: Activated by $\overline{\text{DREQ}}$ signal's low level input from USB (USB request) 0100: Activated by SCI channel 0 transmission complete interrupt 0101: Activated by SCI channel 0 reception complete interrupt 0110: Activated by SCI channel 1 transmission complete interrupt 0111: Activated by SCI channel 1 reception complete interrupt 1000: Activated by TPU channel 0 compare match/input capture A interrupt 1001: Activated by TPU channel 1 compare match/input capture A interrupt 1010: Activated by TPU channel 2 compare match/input capture A interrupt 1011: — 11××: — 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 7.4.10, DMAC Multi-Channel Operation. Legend: ×: Don't care

7.3.5 DMA Band Control Register (DMABCR)

DMABCR controls the operation of each DMAC channel.

- Short Address Mode

Bit	Bit Name	Initial Value	R/W	Description
15	FAE1	0	R/W	Full Address Enable 1 Specifies whether channel 1 is to be used in short address mode or full address mode. In short address mode, channels 1A and 1B are used as independent channels. 0: Short address mode 1: Full address mode
14	FAE0	0	R/W	Full Address Enable 0 Specifies whether channel 0 is to be used in short address mode or full address mode. In short address mode, channels 0A and 0B are used as independent channels. 0: Short address mode 1: Full address mode
13, 12	—	All 0	R/W	Reserved Only 0 should be written to these bits.

Bit	Bit Name	Initial Value	R/W	Description
11	DTA1B	0	R/W	Data Transfer Acknowledge
10	DTA1A	0	R/W	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. 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
9	DTA0B	0	R/W	
8	DTA0A	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
7	DTE1B	0	R/W	Data Transfer Enable
6	DTE1A	0	R/W	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 1when 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 0: Data transfer disabled 1: Data transfer enabled
5	DTE0B	0	R/W	
4	DTE0A	0	R/W	
3	DTIE1B	0	R/W	Data Transfer End Interrupt Enable
2	DTIE1A	0	R/W	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. 0: Transfer end interrupt disabled 1: Transfer end interrupt enabled
1	DTIE0B	0	R/W	
0	DTIE0A	0	R/W	

- Full Address Mode

Bit	Bit Name	Initial Value	R/W	Description
15	FAE1	0	R/W	Full Address Enable 1 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. 0: Short address mode 1: Full address mode
14	FAE0	0	R/W	Full Address Enable 0 Specifies whether channel 0 is to be used in short address mode or full address mode. In full address mode, channels 0A and 0B are used together as a single channel. 0: Short address mode 1: Full address mode
13, 12	—	All 0	R/W	Reserved Although these bits are readable/writable, only 0 should be written here.

Bit	Bit Name	Initial Value	R/W	Description
				Data Transfer Acknowledge Enables or disables 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.
11	DTA1	0	R/W	Data transfer acknowledge 1 Enables or disables clearing, when DMA transfer is performed, of the internal interrupt source selected by the channel 1 data transfer factor setting. 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
10	—	0	R/W	Reserved This bit can be read from or written to. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
9	DTA0	0	R/W	Data Transfer Acknowledge 0 Enables or disables clearing, when DMA transfer is performed, of the internal interrupt source selected by the channel 0 data transfer factor setting. 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
8	—	0	R/W	Reserved Although this bit is readable/writable, only 0 should be written here.
				Data Transfer Master Enable 1 Together with the DTE bit, this bit controls 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
7	DTME1	0	R/W	Data Transfer Master Enable 1 Enables or disables data transfer on channel 1 0: Data transfer disabled. In burst mode, cleared to 0 by an NMI interrupt 1: Data transfer enabled

Bit	Bit Name	Initial Value	R/W	Description
				Data Transfer Enable 1 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
6	DTE1	0	R/W	Data Transfer Enable 1 Enables or disables data transfer on channel 1. 0: Data transfer disabled 1: Data transfer enabled
5	DTME0	0	R/W	Data Transfer Master Enable 0 Enables or disables data transfer on channel 0. 0: Data transfer disabled. In burst mode, cleared to 0 by an NMI interrupt 1: Data transfer enabled
4	DTE0	0	R/W	Data Transfer Enable 0 Enables or disables data transfer on channel 0. 0: Data transfer disabled 1: Data transfer enabled

Bit	Bit Name	Initial Value	R/W	Description
				Data Transfer Interrupt Enable B Enables or disables 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.
3	DTIE1B	0	R/W	Data Transfer Interrupt Enable 1B Enables or disables the channel 1 transfer break interrupt. 0: Transfer break interrupt disabled 1: Transfer break interrupt enabled
				Data Transfer End Interrupt Enable A Enables or disables 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 DTE bit to 1.
2	DTIE1A	0	R/W	Data Transfer End Interrupt Enable 1A Enables or disables the channel 1 transfer end interrupt. 0: Transfer end interrupt disabled 1: Transfer end interrupt enabled
1	DTIE0B	0	R/W	Data Transfer Interrupt Enable 0B Enables or disables the channel 0 transfer break interrupt. 0: Transfer break interrupt disabled 1: Transfer break interrupt enabled
0	DTIE0A	0	R/W	Data Transfer End Interrupt Enable 0A Enables or disables the channel 0 transfer end interrupt. 0: Transfer end interrupt disabled 1: Transfer end interrupt enabled

7.3.6 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 only specific bits of DMACR for the specific channel and also DMABCR can be changed to prevent inadvertent changes being made to 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.

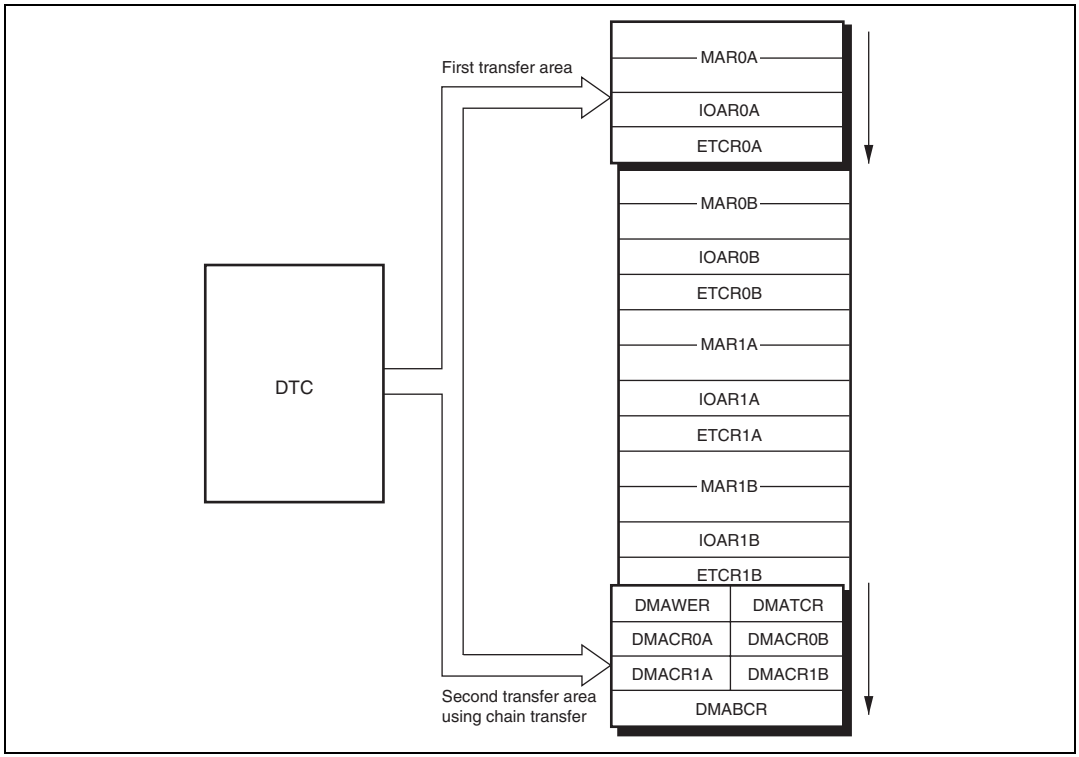


Figure 7.2 Areas for Register Re-Setting by DTC (Example: Channel 0A)

DMAWER controls enabling or disabling of writes to the DMACR and DMABCR by the DTC.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
3	WE1B	0	R/W	Write Enable 1B Enables or disables writes to all bits in DMACR1B, bits 11, 7, and 3 in DMABCR by the DTC. 0: Writes to all bits in DMACR1B, bits 11, 7, and 3 in DMABCR are disabled 1: Writes to all bits in DMACR1B, bits 11, 7, and 3 in DMABCR are enabled
2	WE1A	0	R/W	Write Enable 1A Enables or disables writes to all bits in DMACR1A, and bits 10, 6, and 2 in DMABCR by the DTC. 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
1	WE0B	0	R/W	Write Enable 0B Enables or disables writes to all bits in DMACR0B, bits 9, 5, and 1 in DMABCR by the DTC. 0: Writes to all bits in DMACR0B, bits 9, 5, and 1 in DMABCR, are disabled 1: Writes to all bits in DMACR0B, bits 9, 5, and 1 in DMABCR are enabled
0	WE0A	0	R/W	Write Enable 0A Enables or disables writes to all bits in DMACR0A, and bits 8, 4, and 0 in DMABCR by the DTC. 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

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.

7.4 Operation

7.4.1 Transfer Modes

Table 7.2 lists the DMAC modes.

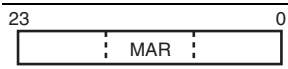
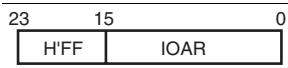
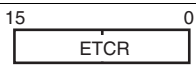
Table 7.2 DMAC Transfer Modes

Transfer Mode			Transfer Source	Remarks
Short address mode	Dual address mode	(1) Sequential mode (2) Idle mode (3) Repeat Mode	• TPU channel 0 to 2 compare match/input capture A interrupts • SCI transmission complete interrupt • SCI reception complete interrupt • A/D conversion end interrupt	• Up to 4 channels can operate independently
Full address mode		(4) Normal mode	• USB request • Auto-request	• Max. 2-channel operation, combining channels A and B • With auto-request, burst mode transfer or cycle steal transfer can be selected
		(5) Block transfer mode	• TPU channel 0 to 2 compare match/input capture A interrupts • SCI transmission complete interrupt • SCI reception complete interrupt • A/D conversion end interrupt	

7.4.2 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 7.3 summarizes register functions in sequential mode.

Table 7.3 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

MAR specifies the start address of the transfer source or transfer destination as 24 bits. MAR is incremented or decremented by 1 or 2 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 7.3 illustrates operation in sequential mode.

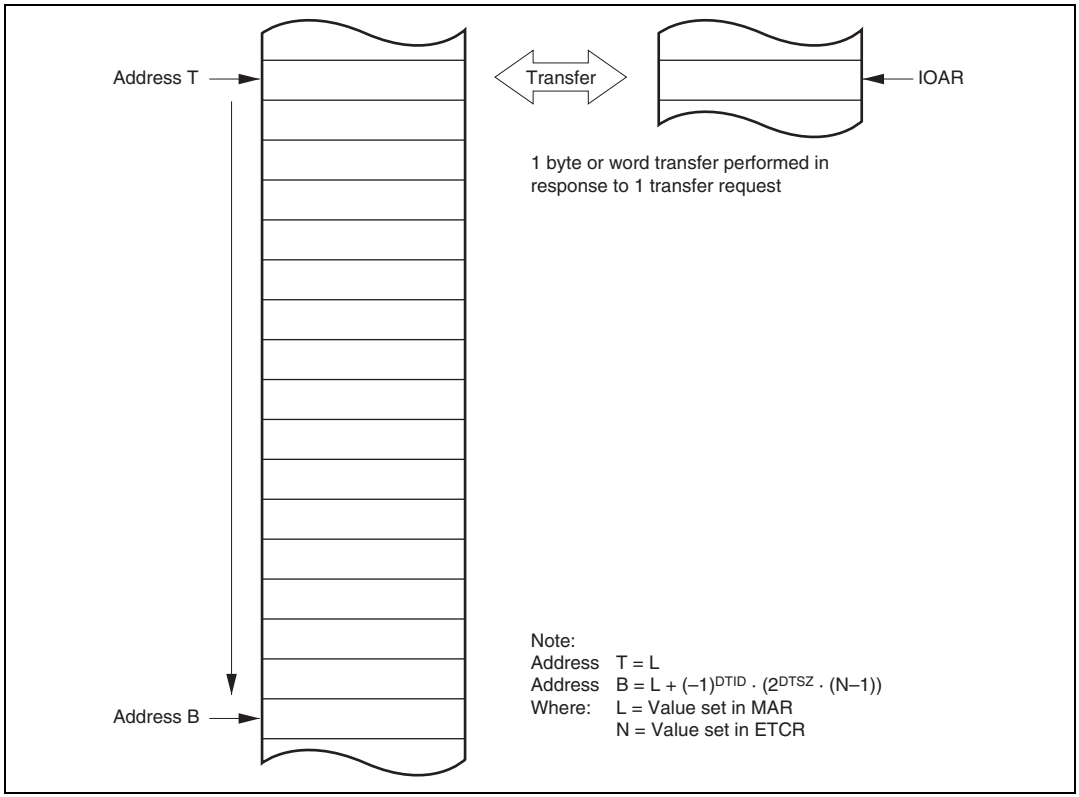
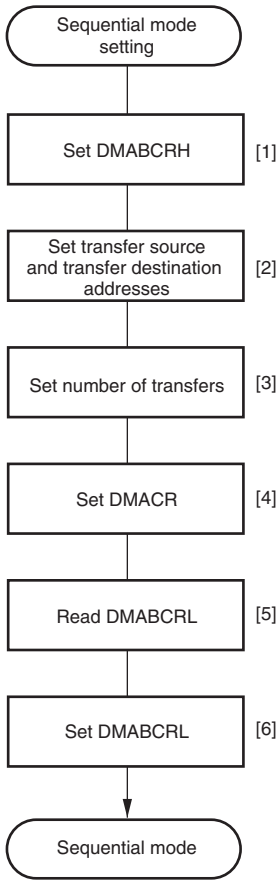


Figure 7.3 Operation in Sequential Mode

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. Transfer requests (activation sources) consist of A/D conversion end interrupt, SCI transmission complete and reception complete interrupts, and TPU channel 0 to 2 compare 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 andinterrupts with the DTIE bit.
- Set the DTE bit to 1 to enable transfer.

Figure 7.4 Example of Sequential Mode Setting Procedure

7.4.3 Idle Mode

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 7.4 summarizes register functions in idle mode.

Table 7.4 Register Functions in Idle Mode

Register	Function		Initial Setting	Operation
	DTDIR = 0	DTDIR = 1		
23 0 MAR	Source address register	Destination address register	Start address of transfer destination or transfer source	Fixed
23 15 0 H'FF IOAR	Destination address register	Source address register	Start address of transfer source or transfer destination	Fixed
15 0 ETCR	Transfer counter		Number of transfers	Decremented every transfer, transfer ends when count reaches H'0000

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 illustrates operation in idle mode.

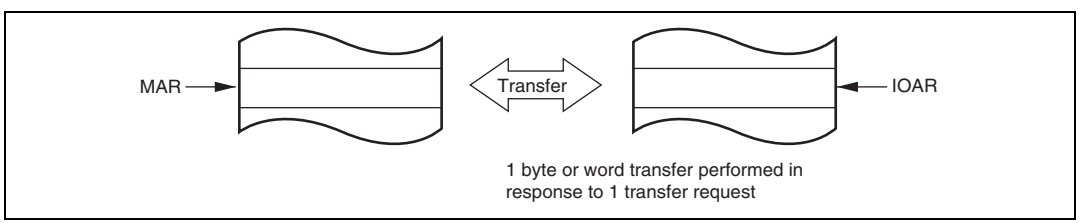


Figure 7.5 Operation in Idle Mode

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 conversion end interrupt, SCI transmission complete and reception complete interrupts, and TPU channel 0 to 2 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. Figure 7.6 shows an example of the setting procedure for idle mode.

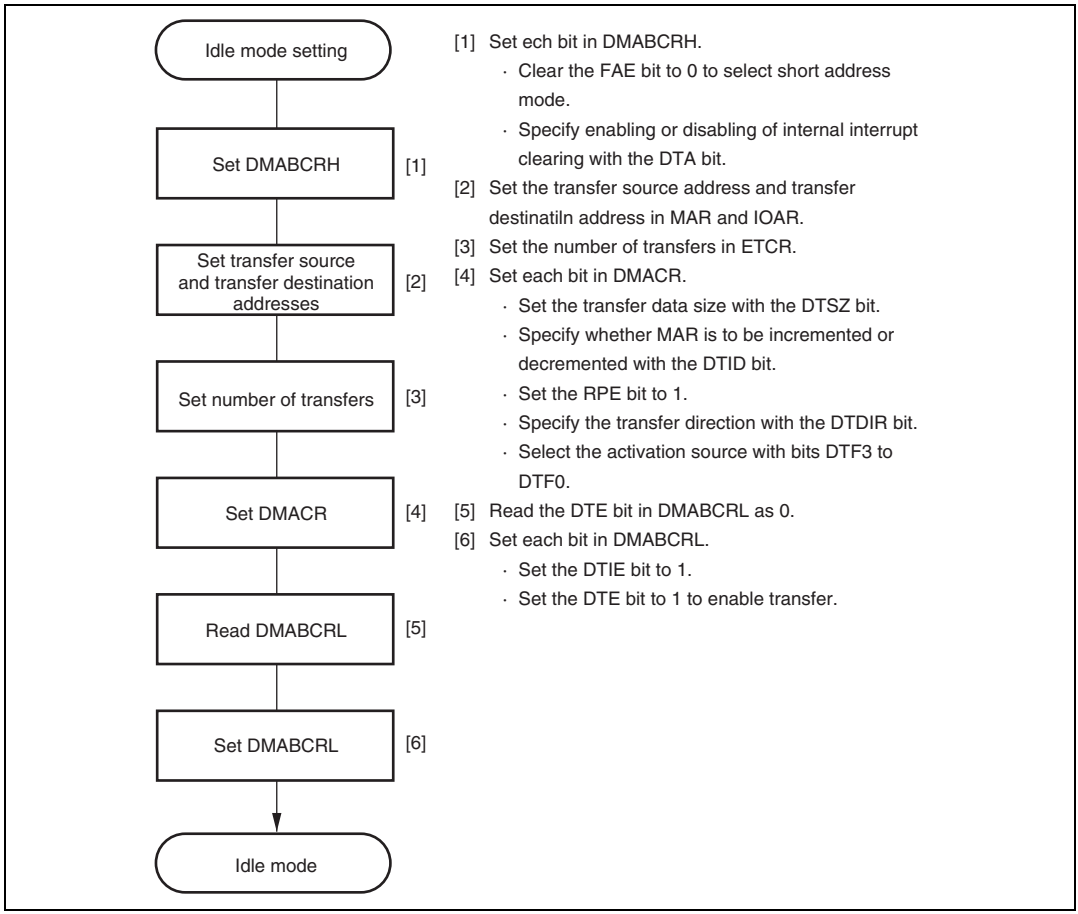
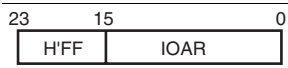
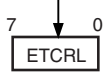


Figure 7.6 Example of Idle Mode Setting Procedure

7.4.4 Repeat Mode

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 ETCR. 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 7.5 summarizes register functions in repeat mode.

Table 7.5 Register Functions in Repeat Mode

	Function			
Register	**DTDIR = 0**	**DTDIR = 1**	**Initial Setting**	**Operation**
	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

MAR specifies the start address of the transfer source or transfer destination as 24 bits. MAR is incremented or decremented by 1 or 2 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. 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, you should clear the DTE bit 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 illustrates operation in repeat mode.

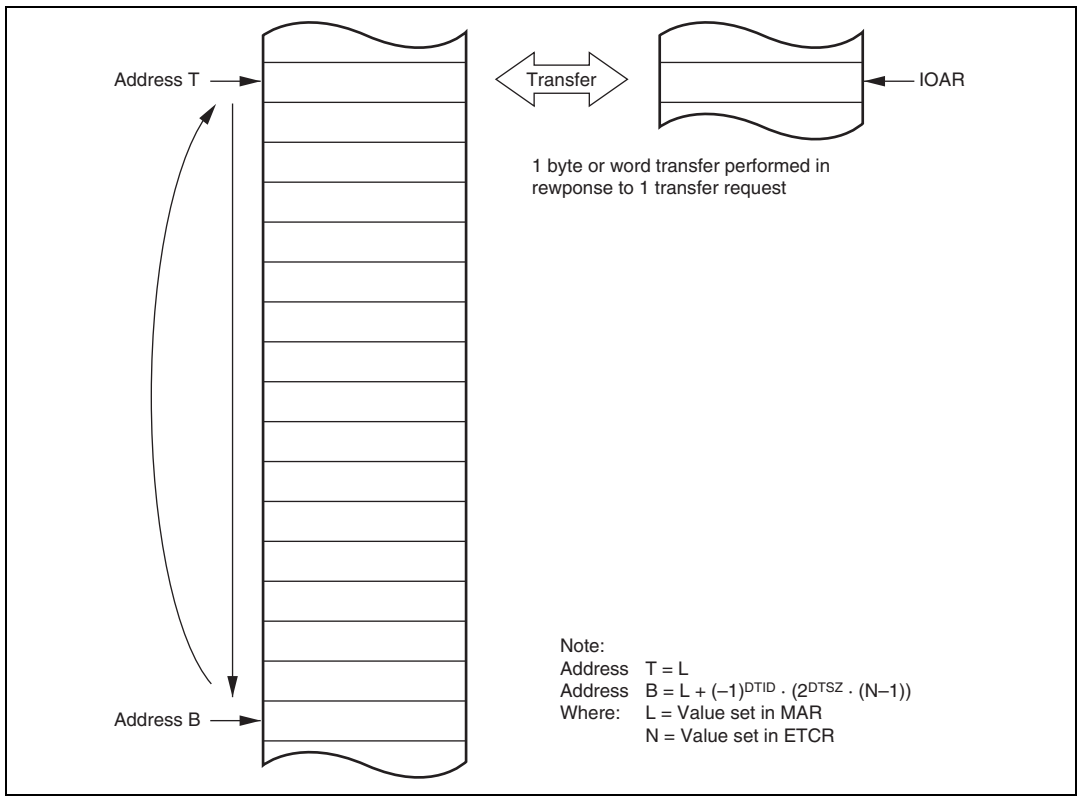


Figure 7.7 Operation in Repeat Mode

Transfer requests (activation sources) consist of A/D conversion end interrupt, SCI transmission complete and reception complete interrupts, and TPU channel 0 to 2 compare match/input capture A interrupts. External requests can be set for channel B only. Figure 7.8 shows an example of the setting procedure for repeat mode.

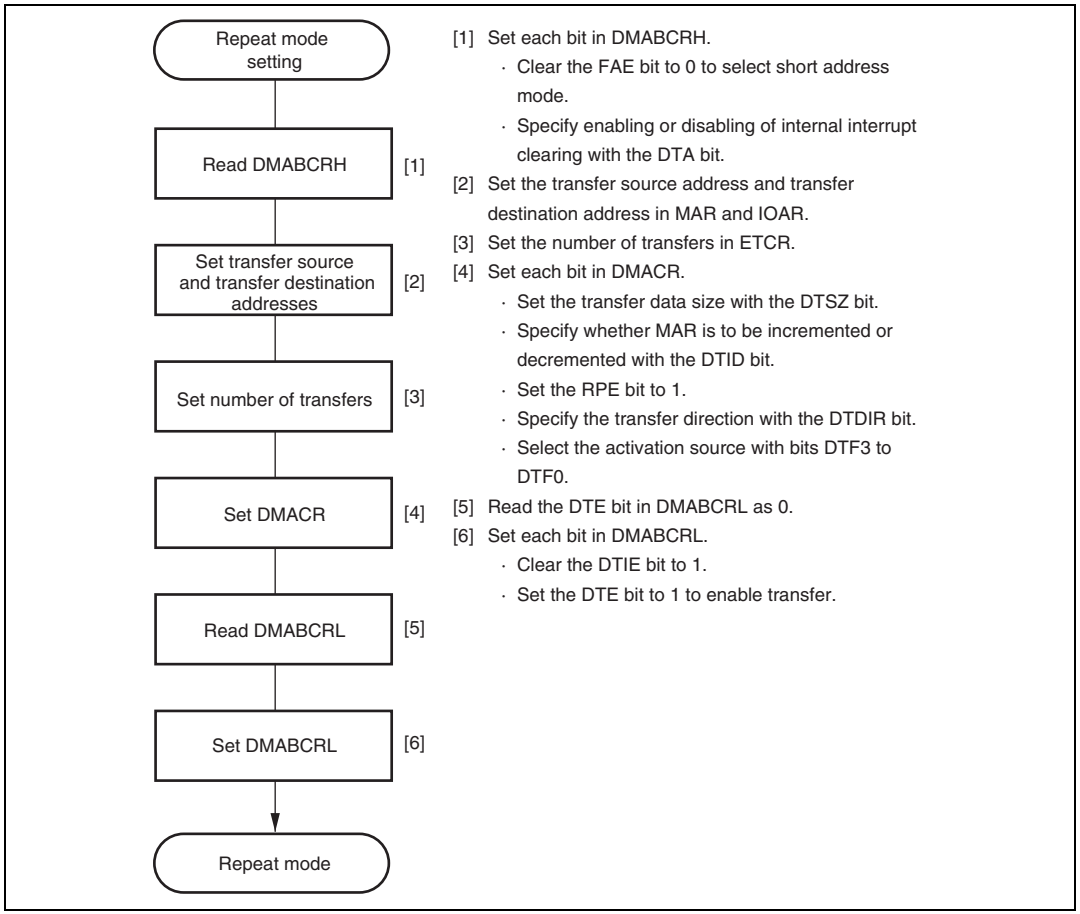
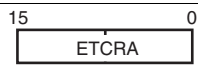


Figure 7.8 Example of Repeat Mode Setting Procedure

7.4.5 Normal Mode

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 7.6 summarizes register functions in normal mode.

Table 7.6 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

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 7.9 illustrates operation in normal mode.

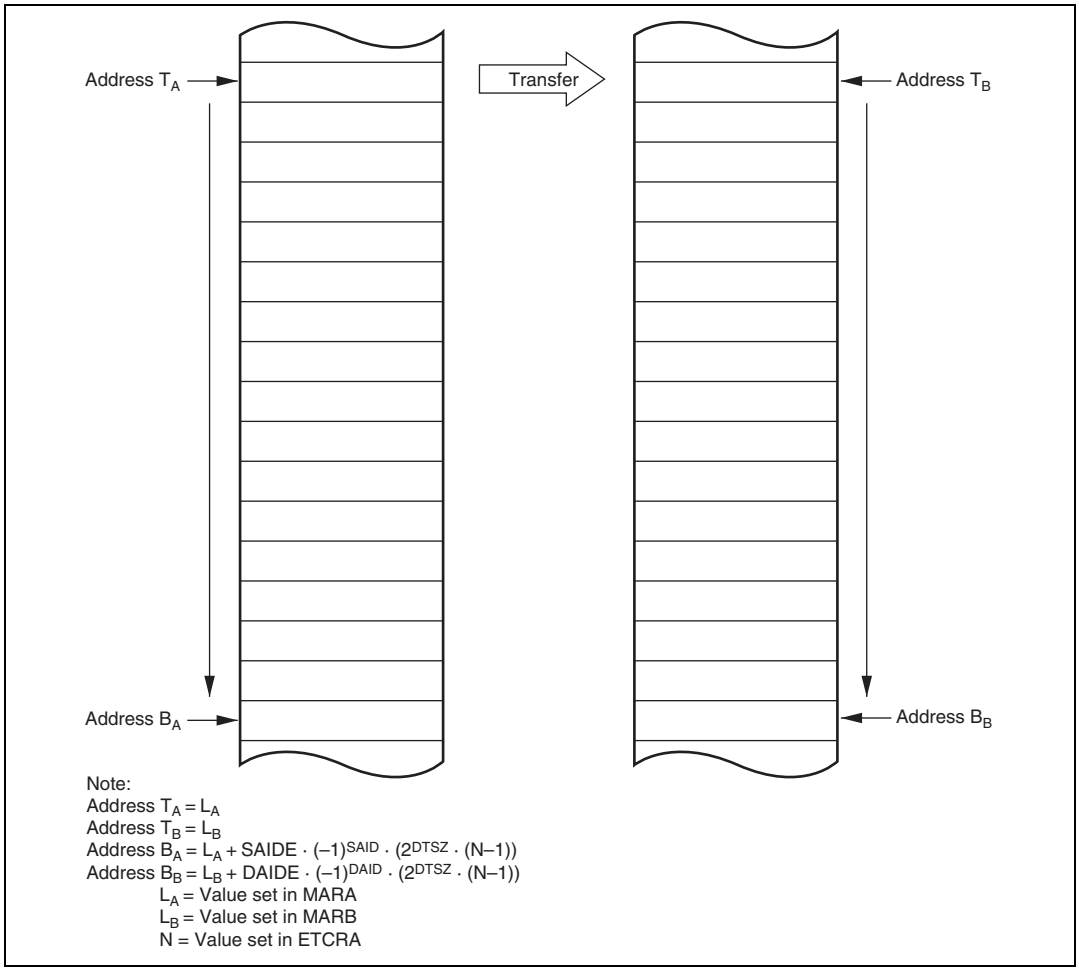
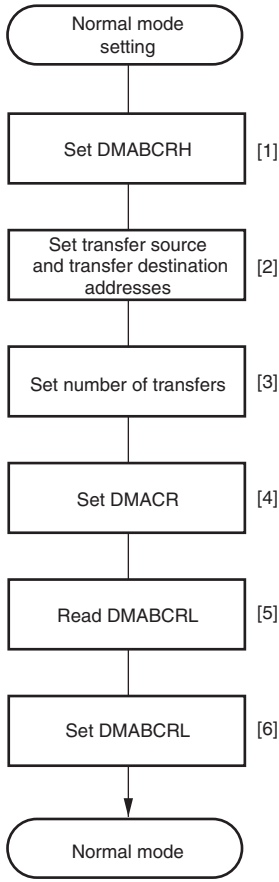


Figure 7.9 Operation in Normal Mode

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. For setting details, see section 7.3.4, DMA Controller Register (DMACR).

Figure 7.10 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 the DTE = 0 and DTME = 0 in DMABCRL.

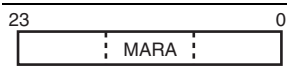
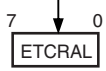
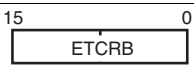
[6] Set each bit in DMABCRL.
- Specify enabling or desabling of transfer end interrupts with the DTIE bit.
- Set both the DTME bit and the DTE bit to 1 to enable transfer.

Figure 7.10 Example of Normal Mode Setting Procedure

7.4.6 Block Transfer Mode

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 7.7 summarizes register functions in block transfer mode.

Table 7.7 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
	Description 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

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. 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 7.11 illustrates operation in block transfer mode when MARB is designated as a block area.

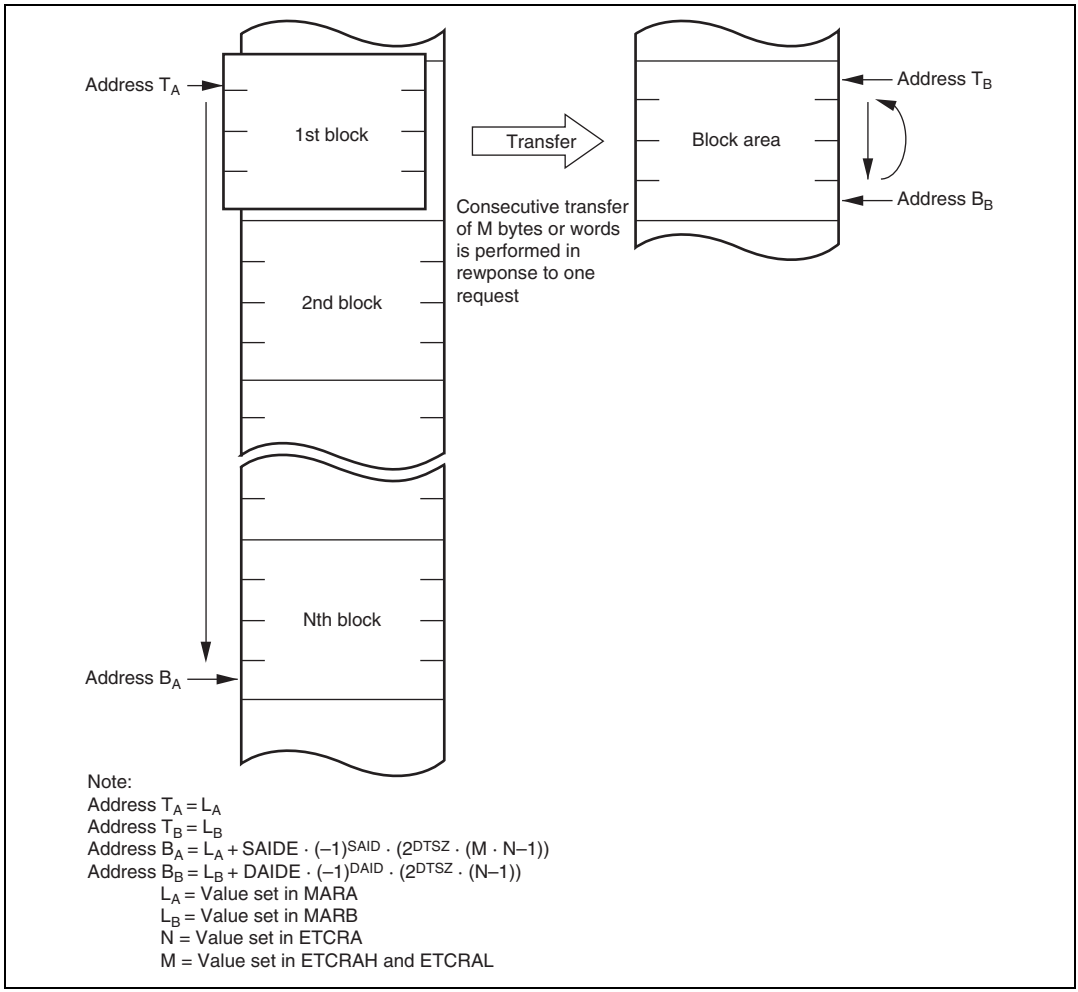


Figure 7.11 Operation in Block Transfer Mode (BLKDIR = 0)

Figure 7.12 illustrates operation in block transfer mode when MARA is designated as a block area.

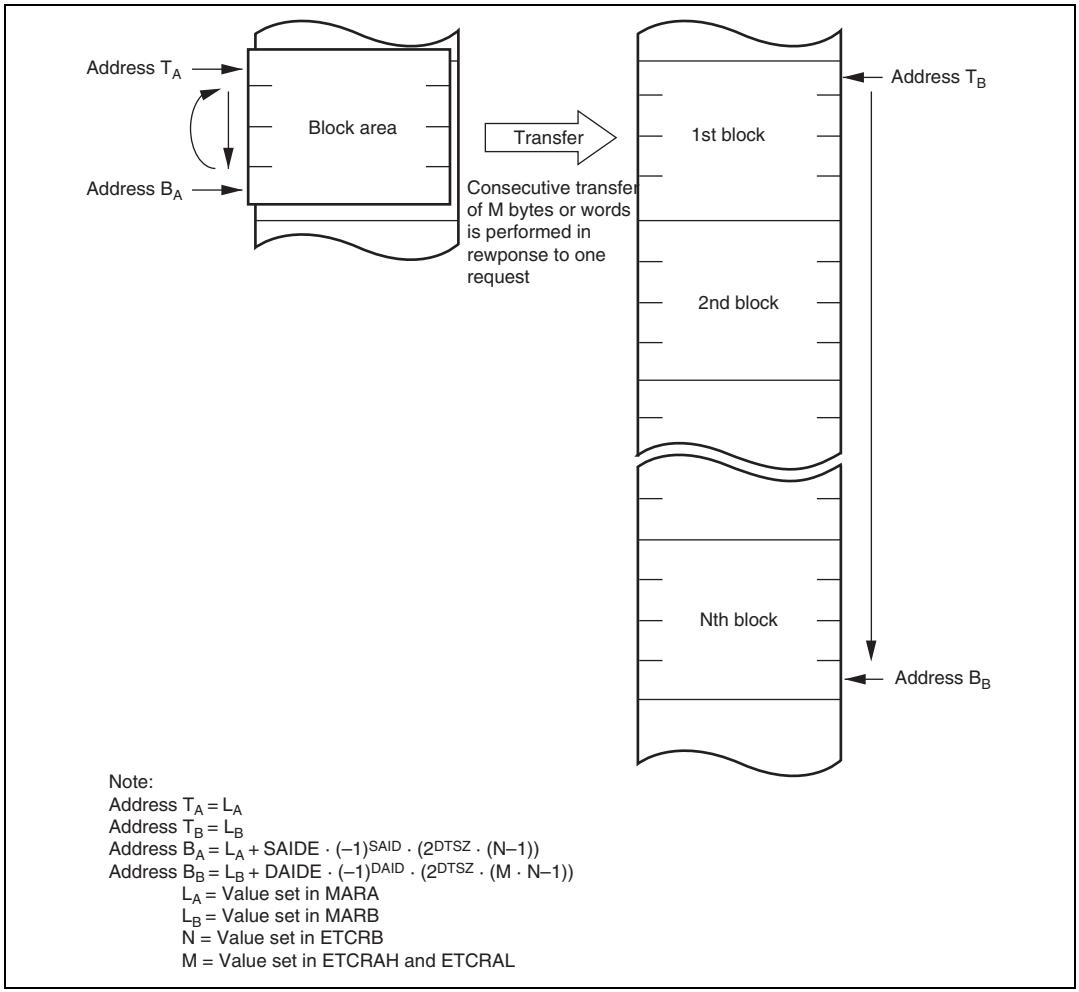


Figure 7.12 Operation in Block Transfer Mode (BLKDIR = 1)

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.

ETCRB is decremented by 1 every block transfer, and when the count reaches H'0000 the DTE bit is cleared and transfer ends. If the DTIE bit is set to 1 at this point, an interrupt request is sent to the CPU or DTC. Figure 7.13 shows the operation flow in block transfer mode.

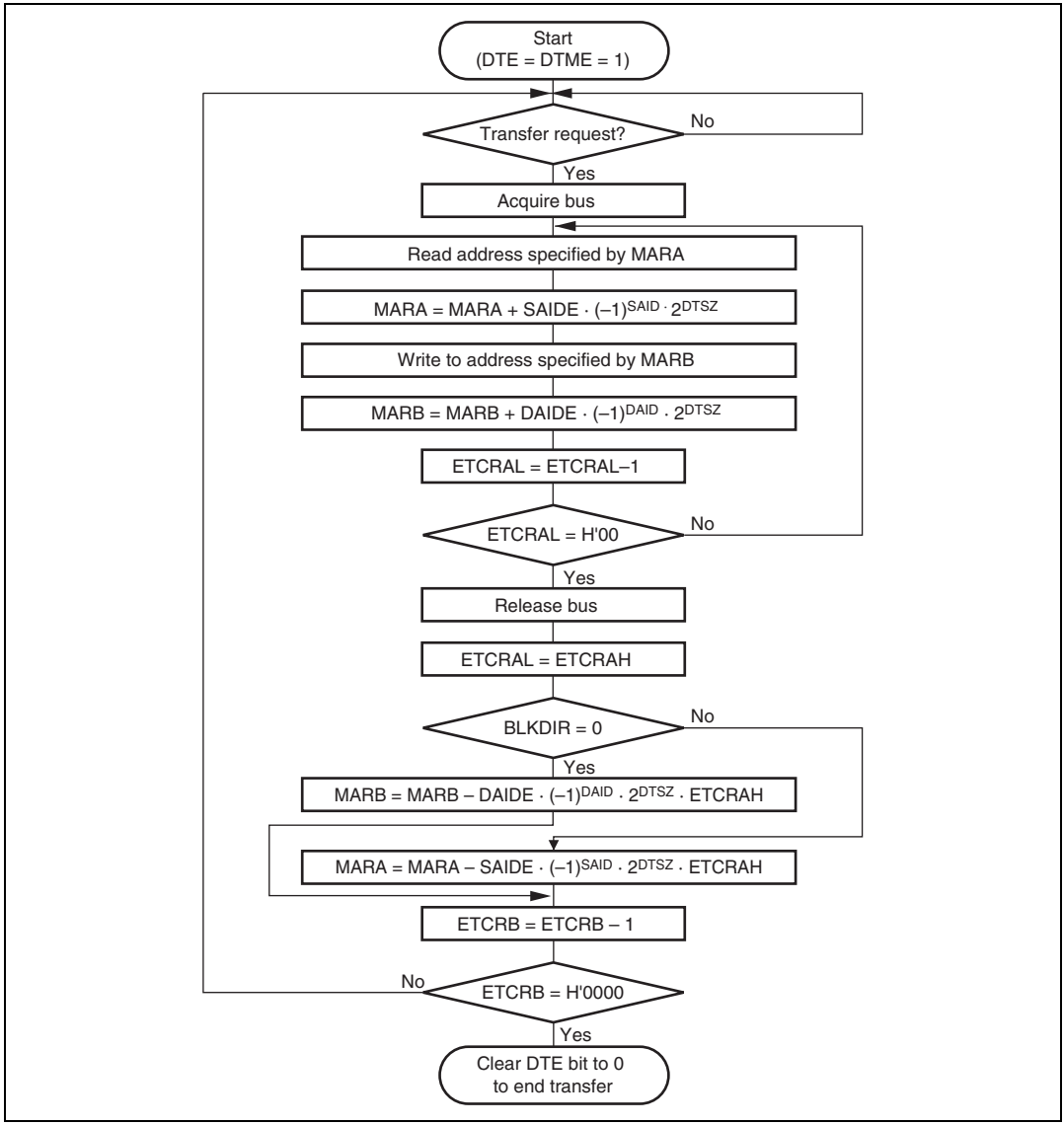


Figure 7.13 Operation Flow in Block Transfer Mode

Transfer requests (activation sources) consist of A/D conversion end interrupt, SCI transmission complete and reception complete interrupts, and TPU channel 0 to 2 compare match/input capture A interrupts. For details, see section 7.3.4, DMA Control Register (DMACR). Figure 7.14 shows an example of the setting procedure for block transfer mode.

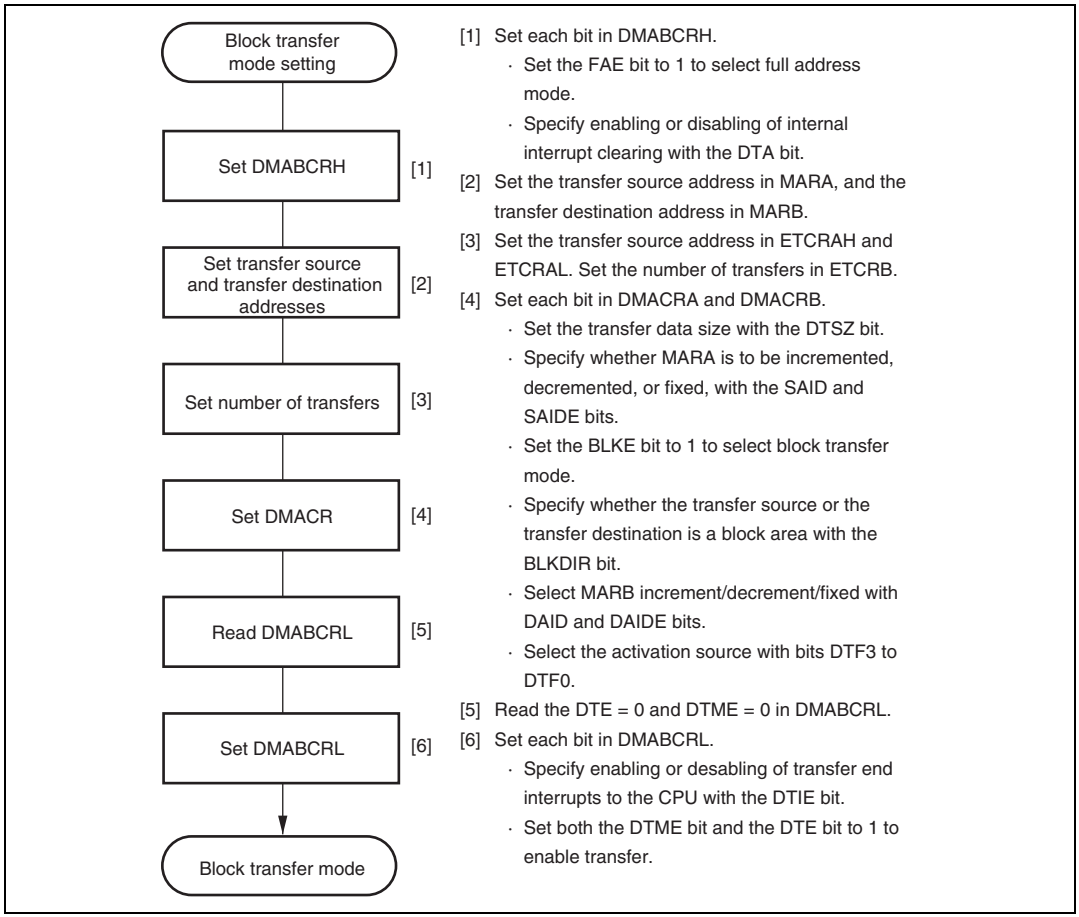


Figure 7.14 Example of Block Transfer Mode Setting Procedure

7.4.7 DMAC Activation Sources

DMAC activation sources consist of internal interrupts, external requests, and auto-requests. The activation sources that can be specified depend on the transfer mode, as shown in table 7.8.

Table 7.8 DMAC Activation Sources

Activation Source		Short Address Mode	Full Address Mode: Normal Mode	Full Address Mode: Block Transfer Mode
Internal interrupt	ADI	○	×	○
	TXI0	○	×	○
	RXI0	○	×	○
	TXI1	○	×	○
	RXI1	○	×	○
	TGI0A	○	×	○
	TGI1A	○	×	○
	TGI2A	○	×	○
USB request	Low level input of the $\overline{\text{DERQ}}$ signal	×	○	×
Auto-request		×	○	×

Legend:
○: Can be specified
×: 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 not accepted.

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 unless the prescribed register is accessed in a DMA transfer. If the same interrupt is used as an activation source for more than one channel, the interrupt request flag is cleared when the highest-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 USB Request: A USB request ($\overline{\text{DREQ}}$ signal) may be specified as the activation source. Level sensing is used for USB requests. In the normal mode of the full address mode, USB requests operate as follows.

Transfer request standby status continues while the $\overline{\text{DREQ}}$ signal is held high. If the $\overline{\text{DREQ}}$ signal is held low, the bus is released each time a single byte of data is transferred, causing continuous transfers to be split up into chunks. If the $\overline{\text{DREQ}}$ signal goes high while a transfer is in progress, the transfer is suspended and the status changes to transfer request standby.

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.

7.4.8 Basic DMAC Bus Cycles

An example of the basic DMAC bus cycle timing is shown in figure 7.15. 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.

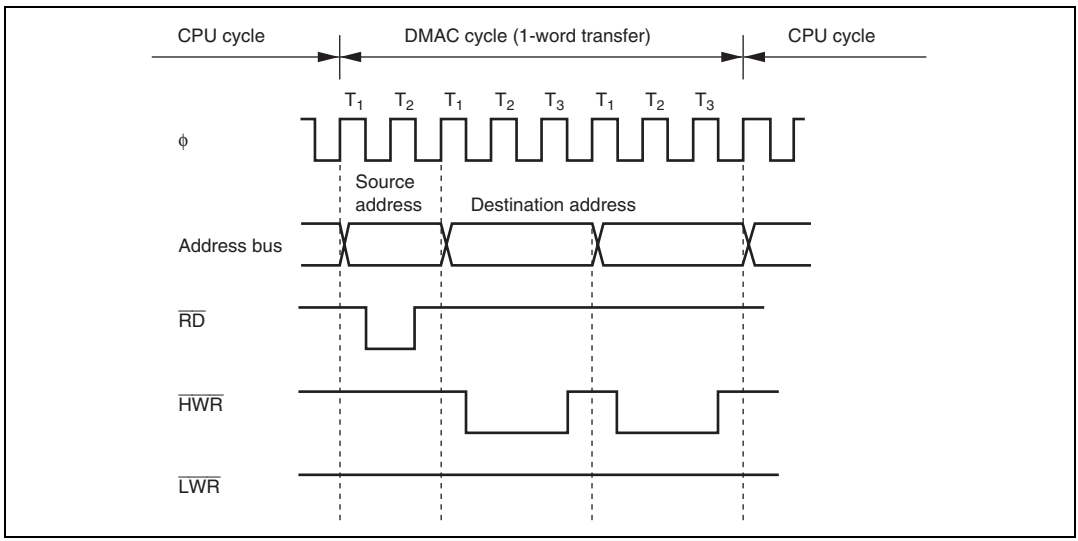
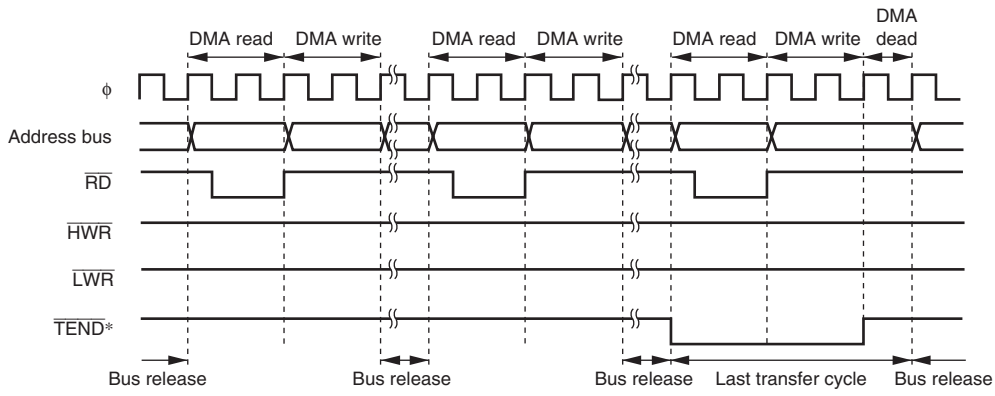


Figure 7.15 Example of DMA Transfer Bus Timing

The address is not output to the external address bus in an access to on-chip memory or an internal I/O register.

7.4.9 DMAC Bus Cycles (Dual Address Mode)

Short Address Mode: Figure 7.16 shows a transfer example in which $\overline{\text{TEND}}^*$ 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.



Note: * $\overline{\text{TEND}}$ output cannot be used with this LSI.

Figure 7.16 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 inserted 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.

Note: * $\overline{\text{TEND}}$ output cannot be used with this LSI.

Full Address Mode (Cycle Steal Mode): Figure 7.17 shows a transfer example in which $\overline{\text{TEND}}$* output is enabled and word-size full address mode transfer (cycle steal mode) is performed from external 16-bit, 2-state access space to external 16-bit, 2-state access space.

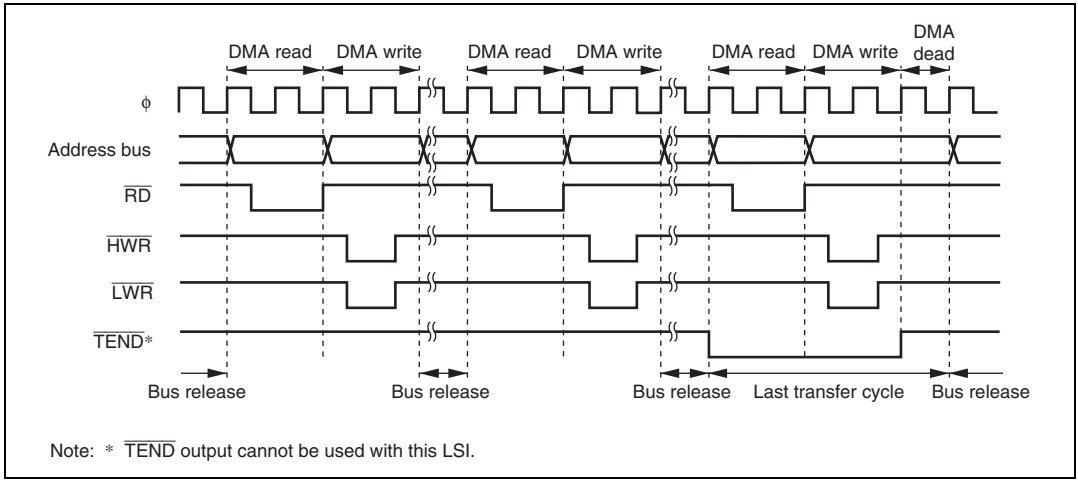


Figure 7.17 Example of Full Address Mode (Cycle Steal) Transfer

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 inserted 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.

Note: * $\overline{\text{TEND}}$ output cannot be used with this LSI.

Full Address Mode (Burst Mode): Figure 7.18 shows a transfer example in which $\overline{\text{TEND}}$* output is enabled and word-size full address mode transfer (burst mode) is performed from external 16-bit, 2-state access space to external 16-bit, 2-state access space.

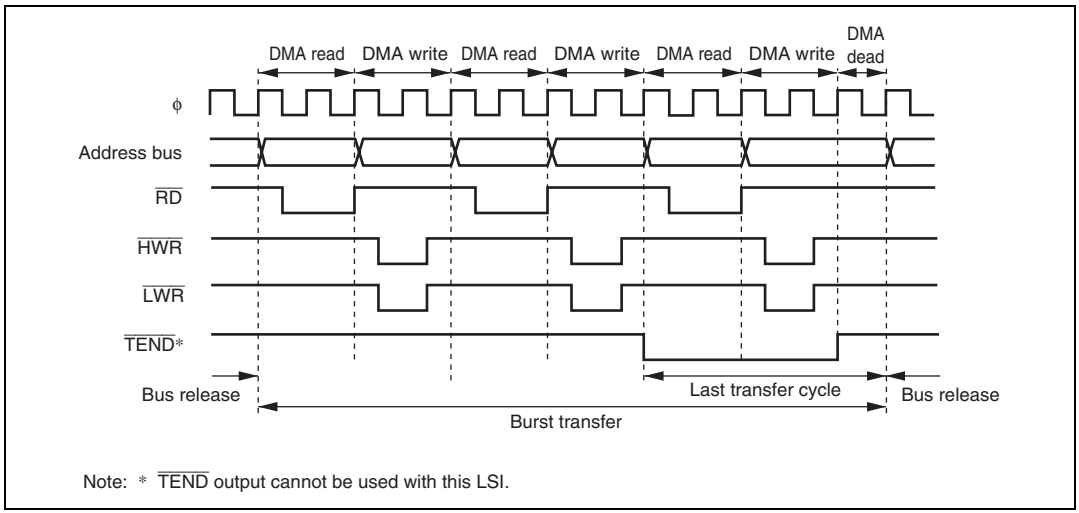


Figure 7.18 Example of Full Address Mode (Burst Mode) Transfer

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.

Note: * $\overline{\text{TEND}}$ output cannot be used with this LSI.

Full Address Mode (Block Transfer Mode): Figure 7.19 shows a transfer example in which $\overline{\text{TEND}}$* output is enabled and word-size full address mode transfer (block transfer mode) is performed from internal 16-bit, 1-state access space to external 16-bit, 2-state access space.

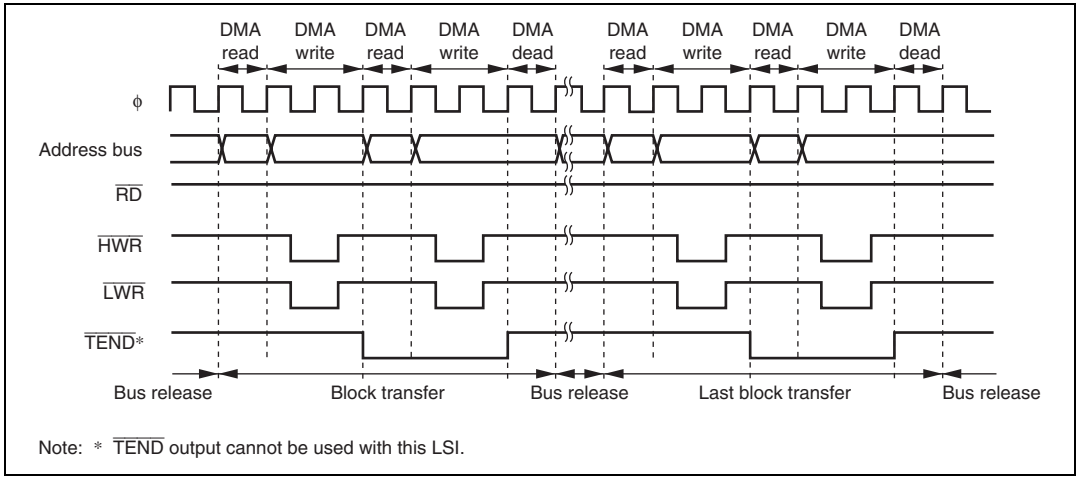


Figure 7.19 Example of Full Address Mode (Block Transfer Mode) Transfer

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 inserted by the CPU or DTC.

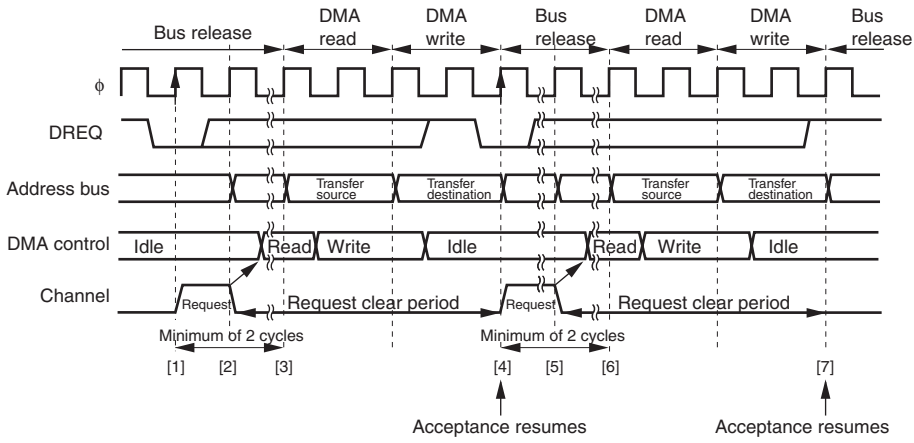
In the transfer end cycle of each block (the cycle in which the transfer counter reaches 0), a one-state DMA dead cycle is inserted after the DMA write cycle.

One block is transmitted without interruption. NMI generation does not affect block transfer operation.

Note: * $\overline{\text{TEND}}$ output cannot be used with this LSI.

$\overline{\text{DREQ}}$ Signal Level Activation Timing (Normal Mode): Set the DTA bit for the channel for which the $\overline{\text{DREQ}}$ signal is selected to 1.

Figure 7.20 shows an example of $\overline{\text{DREQ}}$ level activated normal mode transfer.



[1] Acceptance after transfer enabling; the $\overline{\text{DREQ}}$ signal low level is sampled on the rising edge of f, 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.
[4] [7] Acceptance is resumed after the write cycle is completed.
(As in [1], the $\overline{\text{DREQ}}$ signal low level is sampled on the rising edge of f, and the request is held.)

Figure 7.20 Example of $\overline{\text{DREQ}}$ Level Activated Normal Mode Transfer

$\overline{\text{DREQ}}$ signal 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 $\overline{\text{DREQ}}$ signal low level is sampled while acceptance by means of the $\overline{\text{DREQ}}$ signal is possible, the request is held in the DMAC. Then, when activation is initiated in the DMAC, the request is cleared. 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.

Note: The $\overline{\text{DREQ}}$ signal of this chip is an internal signal of chip, so it is not output from the pin.

7.4.10 DMAC Multi-Channel Operation

The DMAC channel priority order is: channel 0 > channel 1, and channel A > channel B. Table 7.9 summarizes the priority order for DMAC channels.

Table 7.9 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

If transfer requests are issued simultaneously for more than one channel, or if a transfer request for another channel is issued during a transfer, when the bus is released the DMAC selects the highest-priority channel from among those issuing a request according to the priority order shown in table 7.14. 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 7.21 shows a transfer example in which transfer requests are issued simultaneously for channels 0A, 0B, and 1.

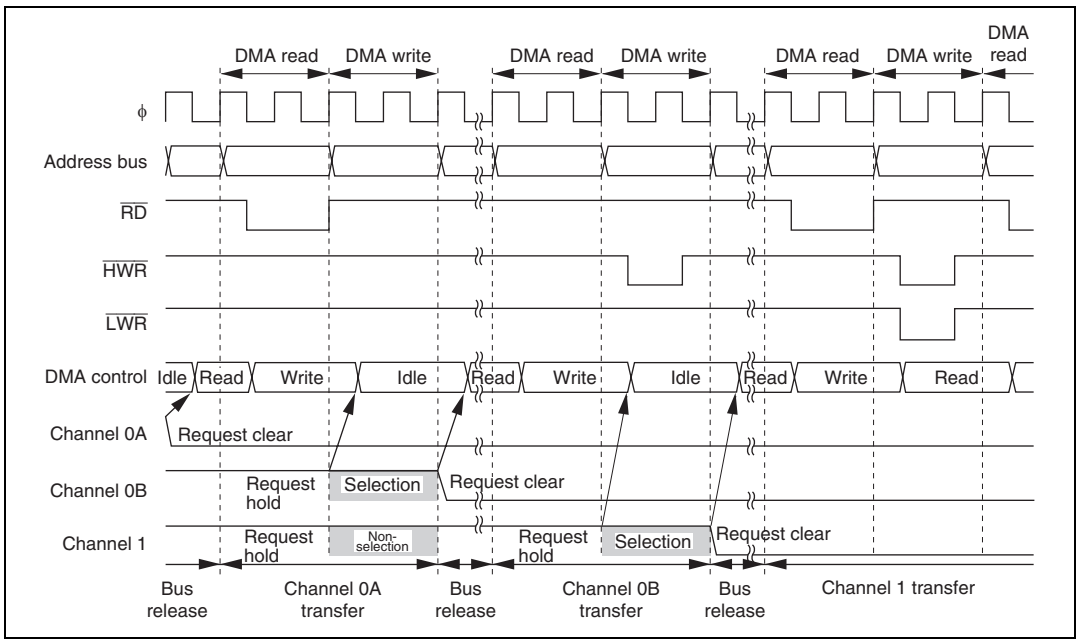


Figure 7.21 Example of Multi-Channel Transfer

7.4.11 Relation between the DMAC, External Bus Requests, and the DTC

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.

7.4.12 NMI Interrupts and DMAC

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 7.22 shows the procedure for continuing transfer when it has been interrupted by an NMI interrupt on a channel designated for burst mode transfer.

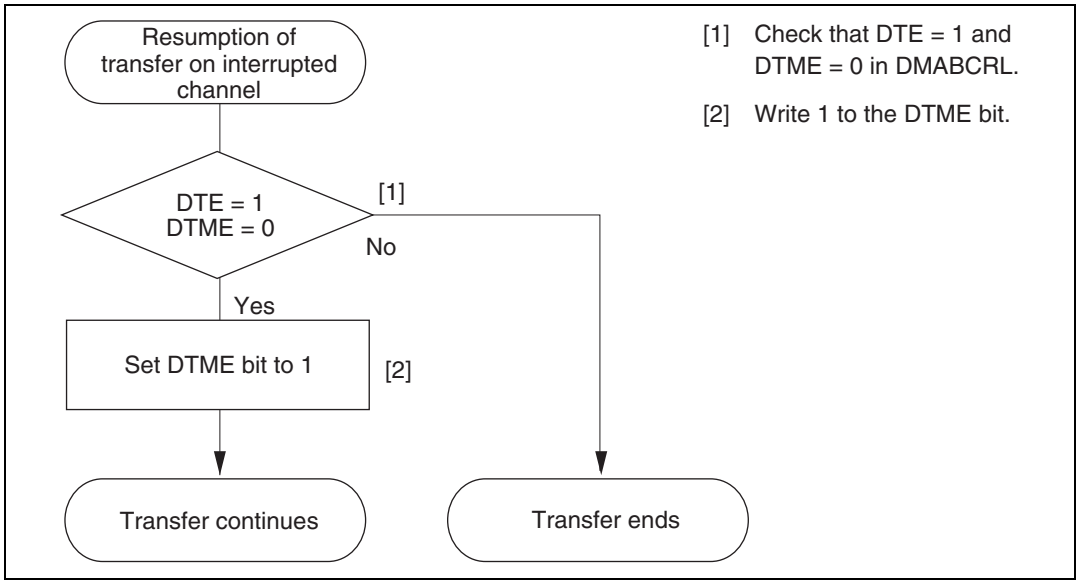


Figure 7.22 Example of Procedure for Continuing Transfer on Channel Interrupted by NMI Interrupt

7.4.13 Forced Termination of DMAC Operation

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 7.23 shows the procedure for forcibly terminating DMAC operation by software.

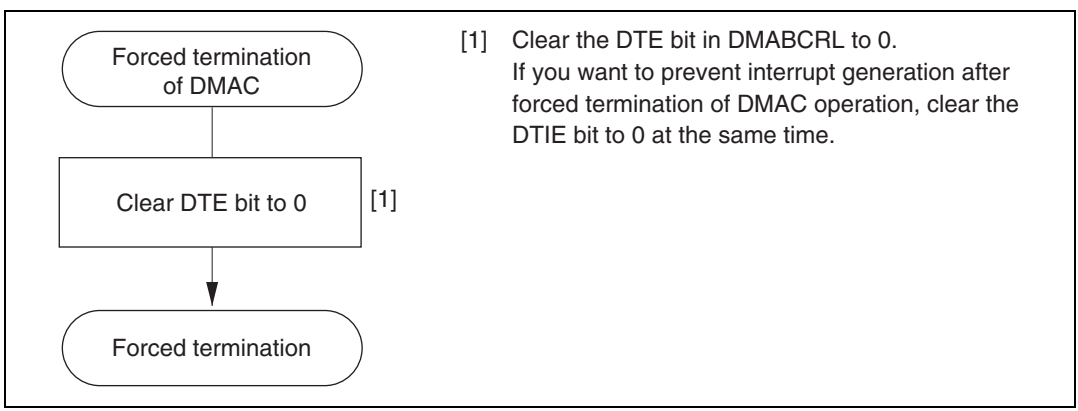


Figure 7.23 Example of Procedure for Forcibly Terminating DMAC Operation

7.4.14 Clearing Full Address Mode

Figure 7.24 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.

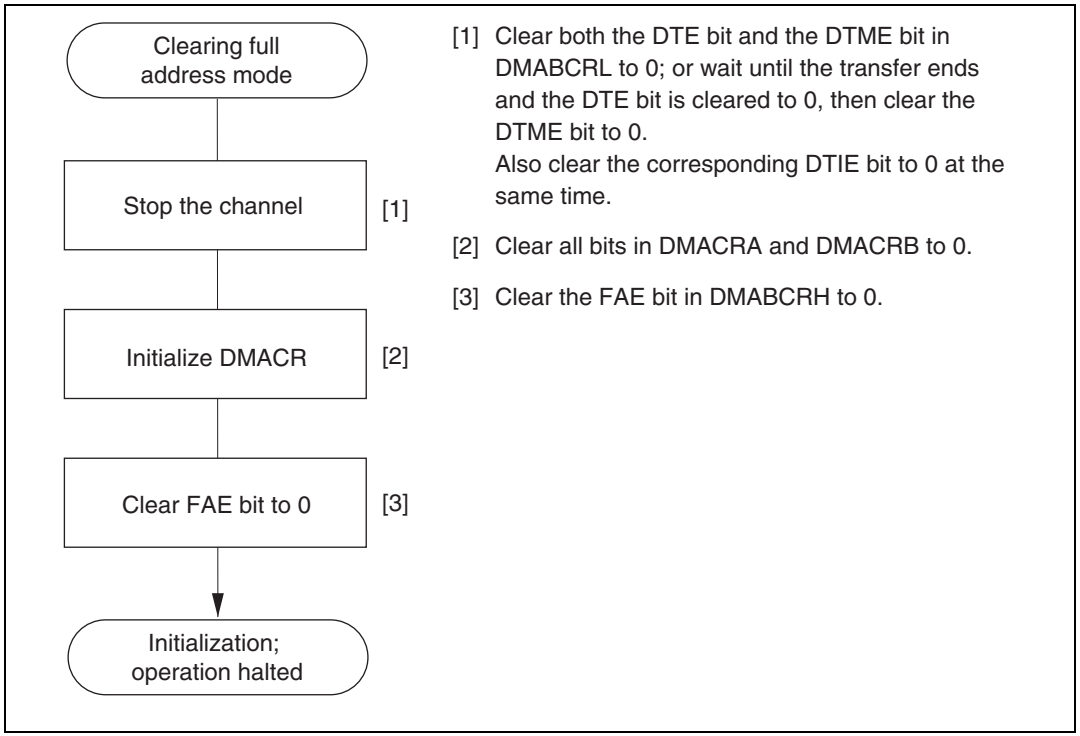


Figure 7.24 Example of Procedure for Clearing Full Address Mode

7.5 Interrupts

The sources of interrupts generated by the DMAC are transfer end and transfer break. Table 7.10 shows the interrupt sources and their priority order.

Table 7.10 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 7.10.

Figure 7.25 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 DTE bit is cleared to 0.

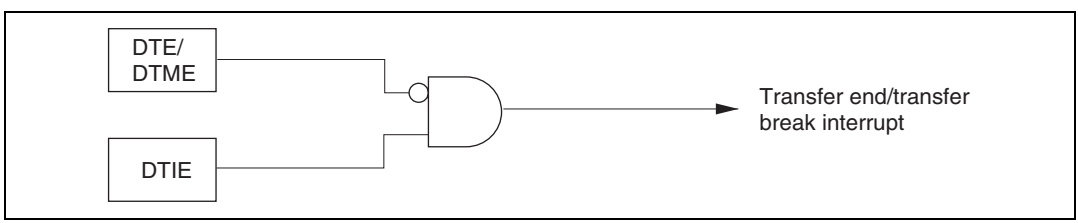


Figure 7.25 Block Diagram of Transfer End/Transfer Break Interrupt

In full address mode, a transfer break interrupt is generated when the DTME bit is cleared to 0 while 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.

7.6 Usage Notes

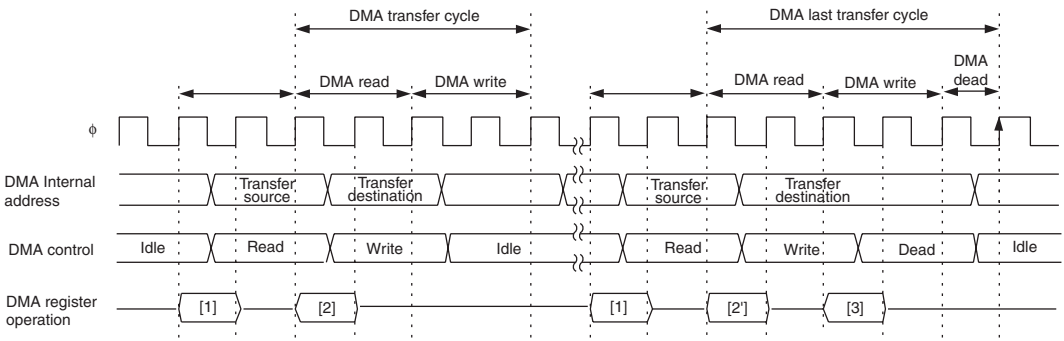
7.6.1 DMAC Register Access during Operation

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, the DMAC register should not be written to in a DMA transfer.

DMAC register reads during operation (including the transfer waiting state) are described below.

1. 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 7.26 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)

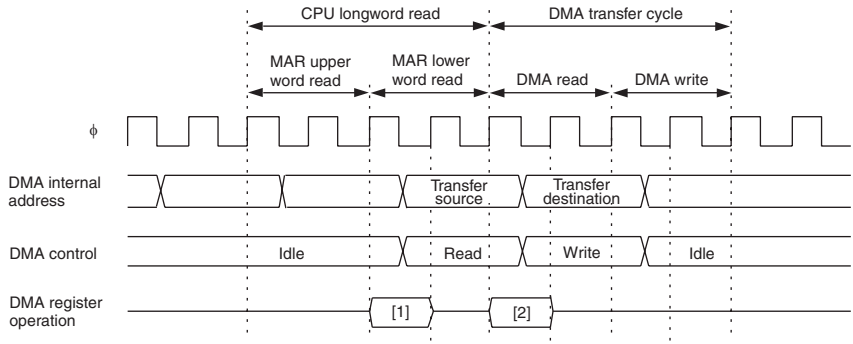
[2'] Transfer destination address register MAR operation (incremented/decremented/fixed)
Block transfer counter ETCR operation (decremented, in last transfer cycle of a block in block transfer mode)

[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)

Note: The MAR operation is post-incrementing/decrementing of the DMA internal address value.

Figure 7.26 DMAC Register Update Timing

2. If a DMAC transfer cycle occurs immediately after a DMAC register read cycle, the DMAC register is read as shown in figure 7.27.



Note: The lower word of MAR is the updated value after the operation in [1].

Figure 7.27 Contention between DMAC Register Update and CPU Read

7.6.2 Module Stop

When the MSTPA7 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 MSTPA7 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/suspend interrupt (DTE = 0 and DTIE = 1)

7.6.3 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 peripheral 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.

7.6.4 Activation Source Acceptance

At the start of activation source acceptance, a low level is detected in both $\overline{\text{DREQ}}$ signal 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 DREQ signal low level remaining from the end of the previous transfer, etc.

7.6.5 Internal Interrupt after End of Transfer

When the DTE bit is cleared to 0 by the end of transfer or an abort, the selected internal interrupt request will be sent to the CPU or DTC even if DTA is set to 1.

Also, if internal DMAC activation has already been initiated when operation is aborted, 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 an abort should be handled by the CPU as necessary.

7.6.6 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.

Section 8 Data Transfer Controller (DTC)

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

8.1 Features

- Transfer possible over any number of channels
  - — Transfer information is stored in memory
  - — One activation source can trigger a number of data transfer (chain transfer)
- Three transfer modes
  - — Normal, repeat, and block transfer modes available
- One activation source can trigger a number of data transfers (chain transfer)
- Direct specification of 16-Mbyte address space possible
- Activation by software is possible
- Transfer can be set in byte or word units
- A CPU interrupt can be requested for the interrupt that activated the DTC
- Module stop mode can be set

Figure 8.1 shows a block diagram of the DTC. The DTC's register information is stored in the on-chip RAM. When the DTC is used, the RAME bit in SYSCR must be set to 1. A 32-bit bus connects the DTC to the on-chip RAM (1 kbyte), enabling 32-bit/1-state reading and writing of the DTC register information.

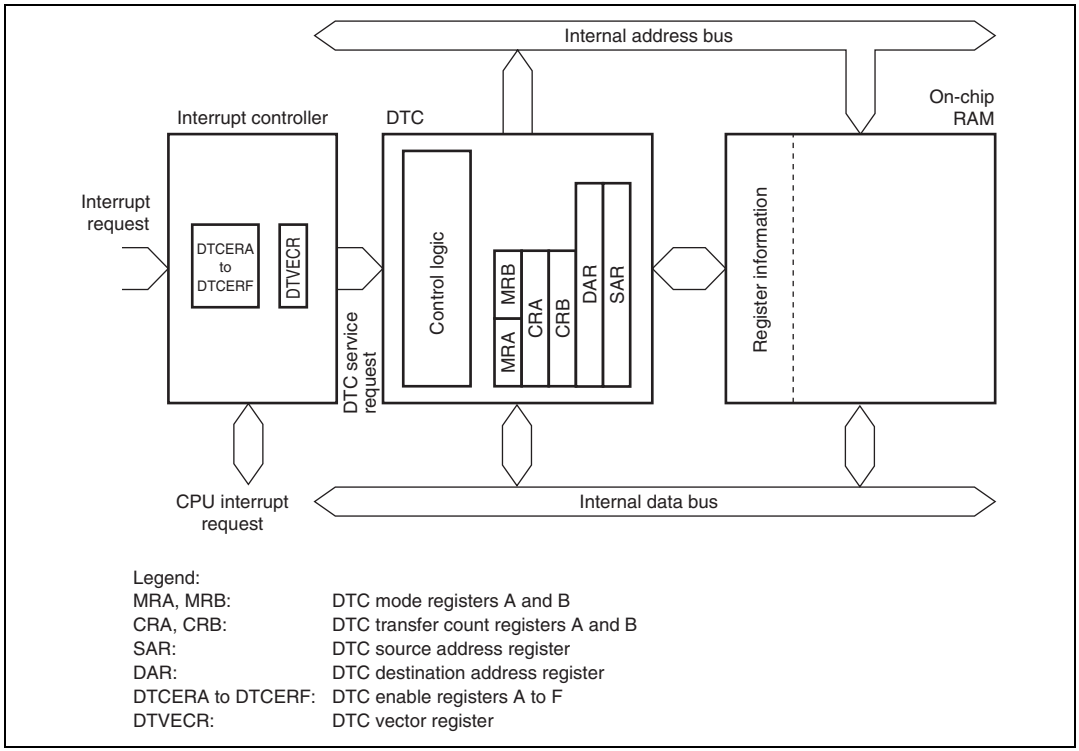


Figure 8.1 Block Diagram of DTC

8.2 Register Descriptions

DTC has the following registers.

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

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

- DTC enable registers (DTCERA to DTCERF)
- DTC vector register (DTVECR)

8.2.1 DTC Mode Register A (MRA)

MRA selects the DTC operating mode.

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

Legend:
×: Don't care

8.2.2 DTC Mode Register B (MRB)

MRB selects the DTC operating mode.

Bit	Bit Name	Initial Value	R/W	Description
7	CHNE	Undefined	—	DTC Chain Transfer Enable This bit specifies a chain transfer. For details, refer to section 8.5.4, Chain Transfer. 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. 0: DTC data transfer completed (waiting for start) 1: DTC chain transfer (reads new register information and transfers data)
6	DISEL	Undefined	—	DTC Interrupt Select This bit specifies whether CPU interrupt is disabled or enabled after a data transfer. 0: Interrupt request is issued to the CPU when the specified data transfer is completed. 1: DTC issues interrupt request to the CPU in every data transfer (DTC does not clear the interrupt request flag that is a cause of the activation).
5 to 0	—	Undefined	—	Reserved These bits have no effect on DTC operation, and the write value should always be 0.

8.2.3 DTC Source Address Register (SAR)

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

8.2.4 DTC Destination Address Register (DAR)

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

8.2.5 DTC Transfer Count Register A (CRA)

CRA is a 16-bit register that designates the number of times data is to be transferred by the DTC. In normal mode, the entire CRA functions as a 16-bit transfer counter (1 to 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 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). In block transfer mode, CRAH stores the block size while CRAL functions as an 8-bit block size 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.

8.2.7 DTC Enable Registers (DTCERA to DTCERF)

DTCER which is comprised of seven registers, DTCERA to DTCERF, is a register that specifies DTC activation interrupt sources. The correspondence between interrupt sources and DTCE bits is shown in table 8.2. For DTCE bit setting, use bit manipulation instructions such as BSET and BCLR for reading and writing. If all interrupts are masked, multiple activation sources can be set at one time (only at the initial setting) by writing data after executing a dummy read on the relevant register.

Bit	Bit Name	Initial Value	R/W	Description
7	DTCEn7	0	R/W	DTC Activation Enable 7 to 0 0: Prohibits DTC startup by an interrupt. 1: Selects a corresponding interrupt source as the DTC startup source. [Clearing conditions] • When the DISEL bit is 1 and the data transfer has ended • When the specified number of transfers have ended [Holding condition] • These bits are not cleared when the DISEL bit is 0 and the specified number of transfers have not ended
6	DTCEn6	0	R/W	
5	DTCEn5	0	R/W	
4	DTCEn4	0	R/W	
3	DTCEn3	0	R/W	
2	DTCEn2	0	R/W	
1	DTCEn1	0	R/W	
0	DTCEn0	0	R/W	

8.2.8 DTC Vector Register (DTVECR)

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

Bit	Bit Name	Initial Value	R/W	Description
7	SWDTE	0	R/W*	DTC Software Activation Enable This bit specifies whether DTC software startup is enabled or prohibited. 0: Prohibits DTC software startup. 1: Enables DTC software startup. [Clearing conditions] • When the DISEL bit is 0 and the specified number of transfers have not ended • When 0 s written to the DISEL bit after a software-activated data transfer end interrupt (SWDTEND) request has been sent to the CPU [Holding conditions] • The DISEL bit is set to 1 and data transfer has finished. • The specified number of data transfers have completed. • A software-triggered data transfer is in progress.
6	DTVEC6	0	R/W	DTC Software Activation Vector 6 to 0 These bits specify a vector number for DTC software activation. The vector address is expressed as H'0400 + (vector number × 2). For example, when DTVEC6 to DTVEC0 = H'10, the vector address is H'0420. When the bit SWDTE is 0, these bits can be written.
5	DTVEC5	0	R/W	
4	DTVEC4	0	R/W	
3	DTVEC3	0	R/W	
2	DTVEC2	0	R/W	
1	DTVEC1	0	R/W	
0	DTVEC0	0	R/W	

Note: * Only 1 may be written to the SWDTE bit.

8.3 Activation Sources

The DTC operates when activated by an interrupt or by a write to DTVECR by software. DTCER is used to select the activation interrupt source. An interrupt request can be directed to the CPU or DTC, as designated by the corresponding DTCER bit. 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 8.1 shows an activation source and DTCER clearance. The activation source flag, in the case of RXI0, for example, is the RDRF flag of SCI channel 0.

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. Figure 8.2 shows a block diagram of activation source control. For details see section 5, Interrupt Controller.

Table 8.1 Activation Source and DTCER Clearance

Activation Source	When the DISEL Bit Is 0 and the Specified Number of Transfer 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

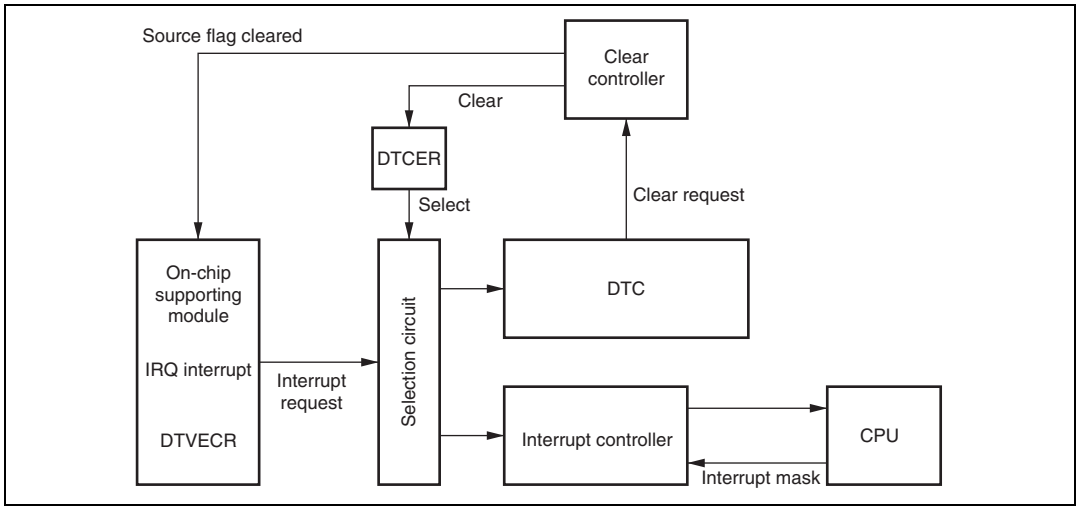


Figure 8.2 Block Diagram of DTC Activation Source Control

8.4 Location of Register Information and DTC Vector Table

Locate the register information in the on-chip RAM (addresses: H'FFEBC0 to H'FFEFBF). Register information should be located at the address that is multiple of four within the range. Locating the register information in address space is shown in figure 8.3. Locate the MRA, SAR, MRB, DAR, CRA, and CRB registers, in that order, from the start address of the register information. In the case of chain transfer, register information should be located in consecutive areas and the register information start address should be located at the corresponding vector address to the interrupt source. 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.

When the DTC is activated by software, the vector address is obtained from: H'0400 + (DTVECR[6:0] × 2). For example, if DTVECR is H'10, the vector address is H'0420. The configuration of the vector address is the same in both normal and advanced modes, a 2-byte unit being used in both cases. These two bytes specify the lower bits of the register information start address.

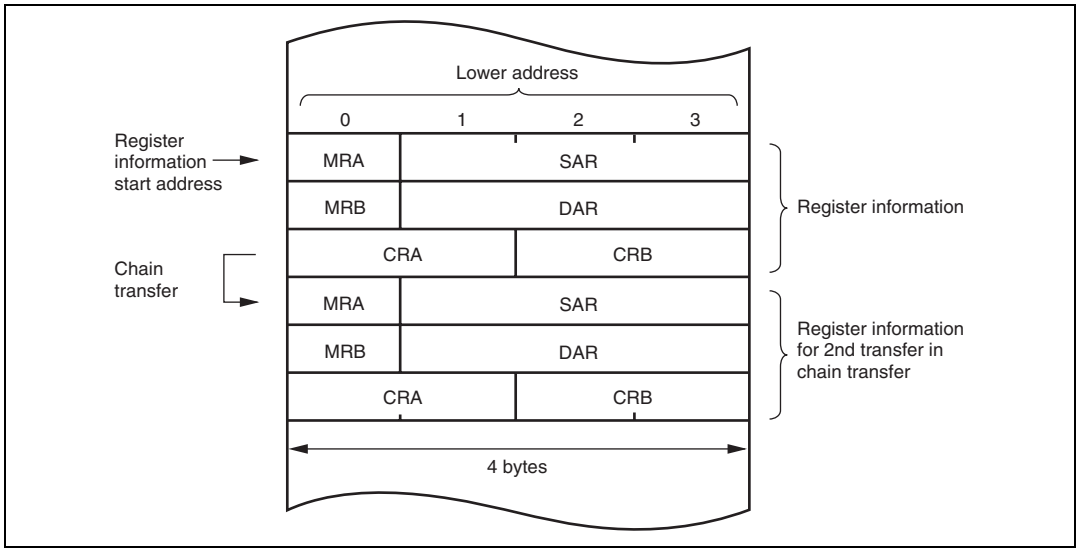


Figure 8.3 Correspondence between DTC Vector Address and Register Information

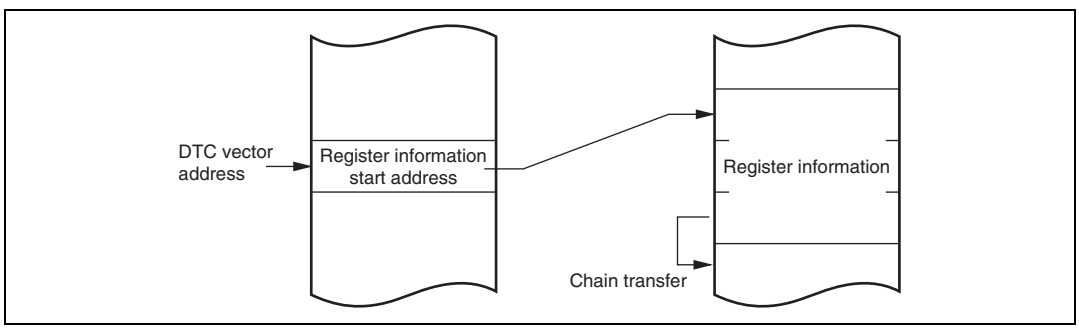


Figure 8.4 Correspondence between DTC Vector Address and Register Information

Table 8.2 Interrupt Sources, DTC Vector Addresses, and Corresponding DTCE

Interrupt Source	Origin of Interrupt Source	Vector Number	DTC Vector Address	DTCE*	Priority
Software	Write to DTVECR	DTVECR	H'0400 + DTVECR[6:0] ×2		High
External pins	IRQ0	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	
	IRQ7	23	H'042E	DTCEA0	
A/D	ADI	28	H'0438	DTCEB6	
TPU channel 0	TGIA0	32	H'0440	DTCEB5	
	TGIB0	33	H'0442	DTCEB4	
	TGIC0	34	H'0444	DTCEB3	
	TGID0	35	H'0446	DTCEB2	
TPU channel 1	TGI1A	40	H'0450	DTCEB1	
	TGI1B	41	H'0452	DTCEB0	
TPU channel 2	TGI2A	44	H'0458	DTCEC7	
	TGI2B	45	H'045A	DTCEC6	
8-bit timer channel 0	CMIA0	64	H'0480	DTCED3	
	CMIB0	65	H'0482	DTCED2	
8-bit timer channel 1	CMIA1	68	H'0488	DTCED1	
	CMIB1	69	H'048A	DTCED0	
DMAC	DEND0A	72	H'0490	DTCEE7	
	DEND0B	73	H'0492	DTCEE6	
	DEND1A	74	H'0494	DTCEE5	
	DEND1A	75	H'0496	DTCEE4	
SIC channel 0	RXI0	81	H'04A2	DTCEE3	
	TXI0	82	H'04A4	DTCEE2	
SIC channel 1	RXI1	85	H'04AA	DTCEE1	
	TXI1	86	H'04AC	DTCEE0	
SIC channel 2	RXI2	89	H'04B2	DTCEF7	
	TXI2	90	H'04B4	DTCEF6	Low

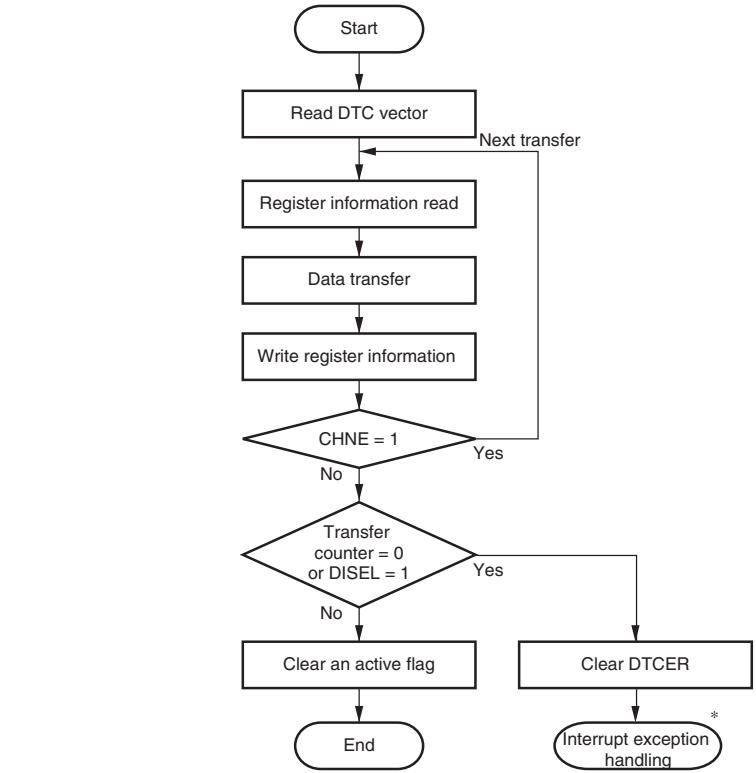
Note: * DTCE bits with no corresponding interrupt are reserved, and the write value should always be 0.

8.5 Operation

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

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

Figure 8.5 shows a flowchart of DTC operation.



Note: * For details on the processing that takes place, refer to the chapter on the peripheral module in question.

Figure 8.5 Flowchart of DTC Operation

Table 8.3 summarizes DTC functions.

Table 8.3 Overview of DTC Functions

Transfer Mode	Activation Source	Address Register	
		Source	Destination
Normal mode • One byte or one word data is transferred in response to a single transfer request. • The memory address is incremented by 1 or 2. • The number of times of data transfer is designated as 1 to 65,536. **Repeat mode** • One byte or one word data is transferred in response to a single transfer request. • The memory address is incremented by 1 or 2. • When the specified number of transfers (1 to 256) have ended, the initial state is restored, and transfer is repeated. **Block transfer mode** • The data of the specified block is transferred in response to a single transfer request. • The block size is designated as 1 to 256 bytes or words. • The number of times of data transfer is designated as 1 to 65,536. • Either the transfer source or destination is designated as a block area.	• IRQ • TGI for TPU • CMI for 8-bit timer • TXI and RXI for SCI • ADI for A/D converter • DEND for DMAC • Software	24 bits	24 bits

8.5.1 Normal Mode

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 8.4 shows the register information in normal mode, and figure 8.6 shows the memory mapping in normal mode.

Table 8.4 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

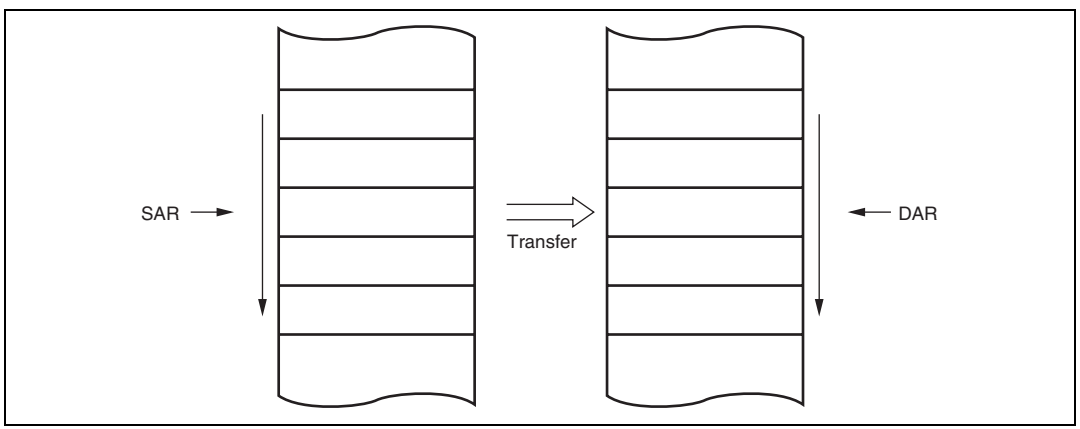


Figure 8.6 Memory Mapping in Normal Mode

8.5.2 Repeat 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 8.5 shows the register information in repeat mode, and figure 8.7 shows the memory mapping in repeat mode.

Table 8.5 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	Designates transfer count
DTC transfer count register B	CRB	Not used

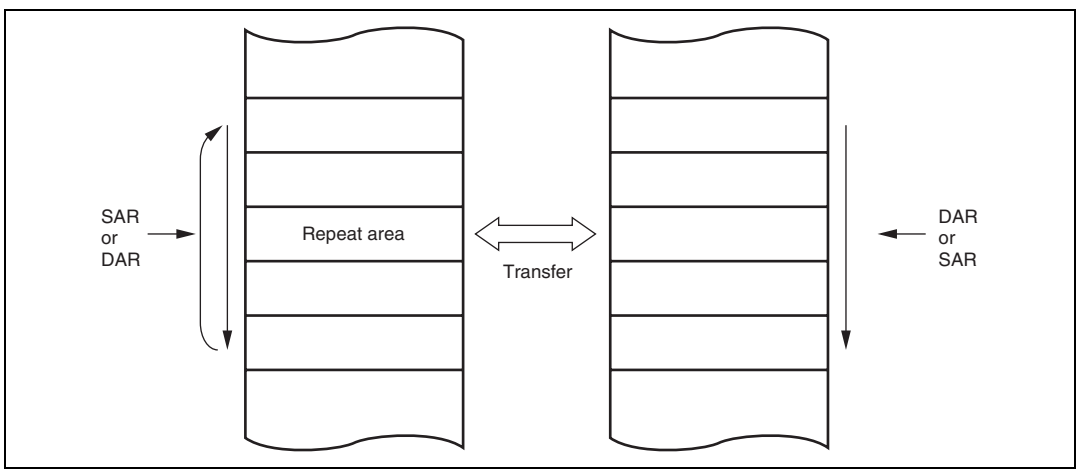


Figure 8.7 Memory Mapping in Repeat Mode

8.5.3 Block Transfer 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 8.6 shows the register information in block transfer mode, and figure 8.8 shows the memory mapping in block transfer mode.

Table 8.6 Register Information in Block Transfer 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 block size
DTC transfer count register AL	CRAL	Designates block size count
DTC transfer count register B	CRB	Transfer count

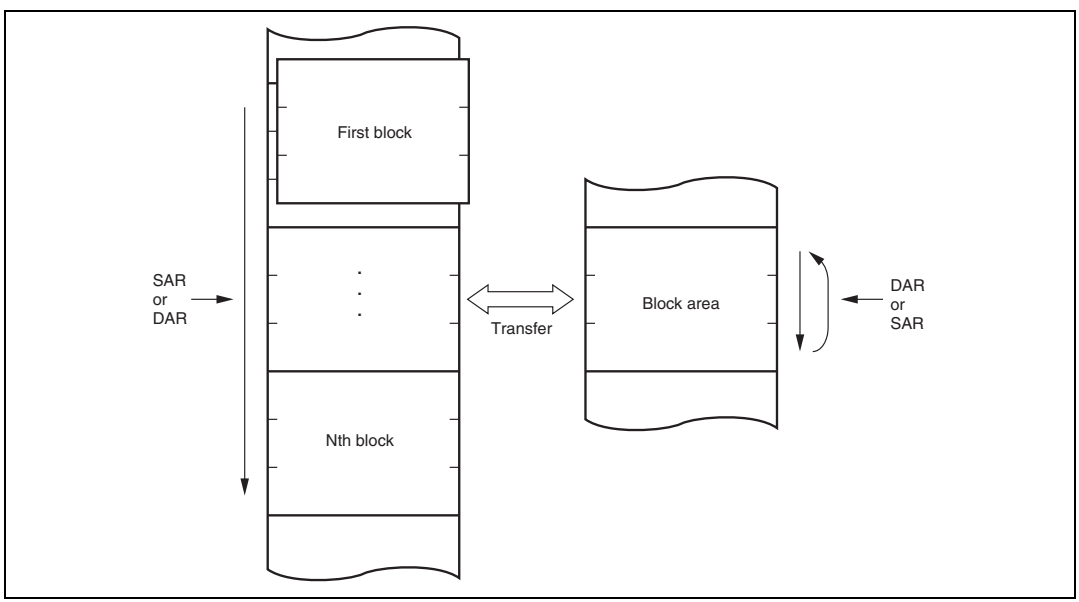


Figure 8.8 Memory Mapping in Block Transfer Mode

8.5.4 Chain Transfer

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

Figure 8.9 shows the memory map for chain transfer. When activated, the DTC reads the register information start address stored at the vector address, and then reads the first register information at that start address. After the data transfer, the CHNE bit will be tested. When it has been set to 1, DTC reads next register information located in a consecutive area and performs the data transfer. These sequences are repeated until the CHNE bit is cleared to 0.

In the case of transfer with 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.

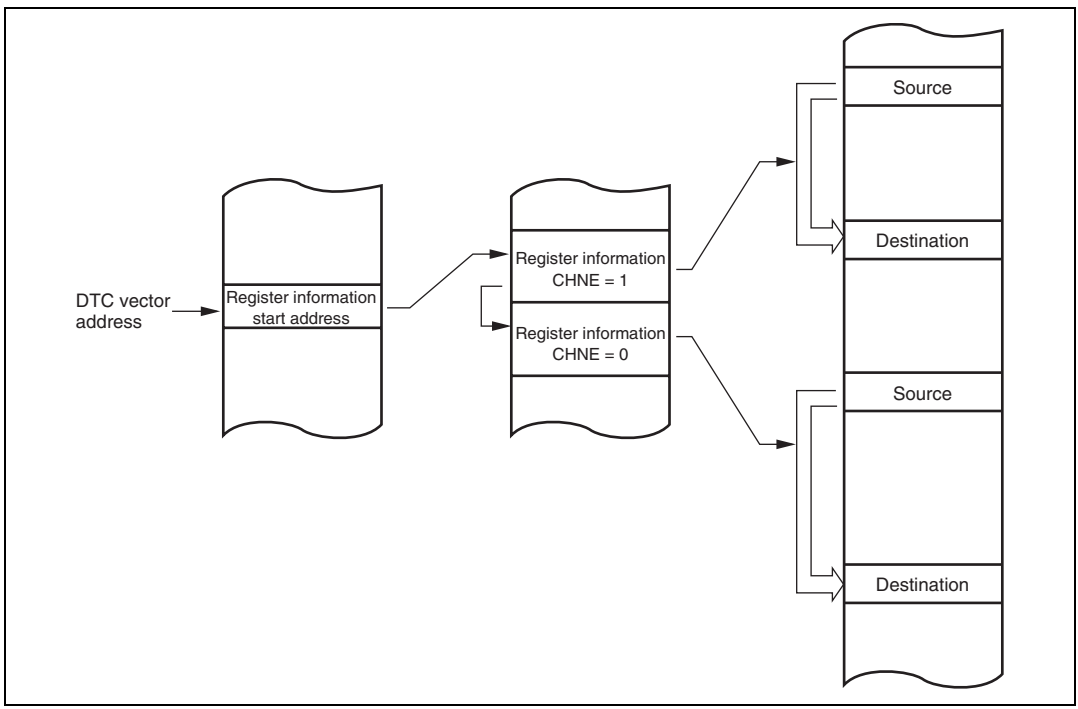


Figure 8.9 Chain Transfer Memory Map

8.5.5 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.

8.5.6 Operation Timing

Figures 8.10 to 8.12 show the DTC operation timing.

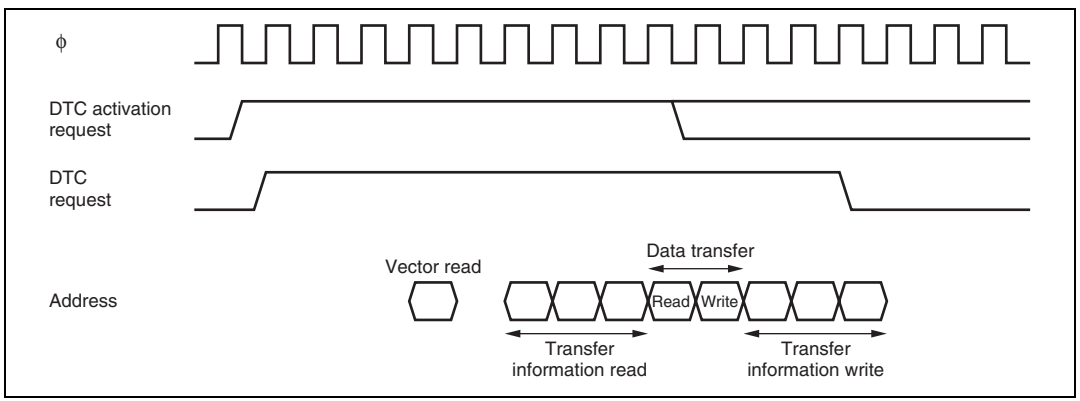


Figure 8.10 DTC Operation Timing (Example in Normal Mode or Repeat Mode)

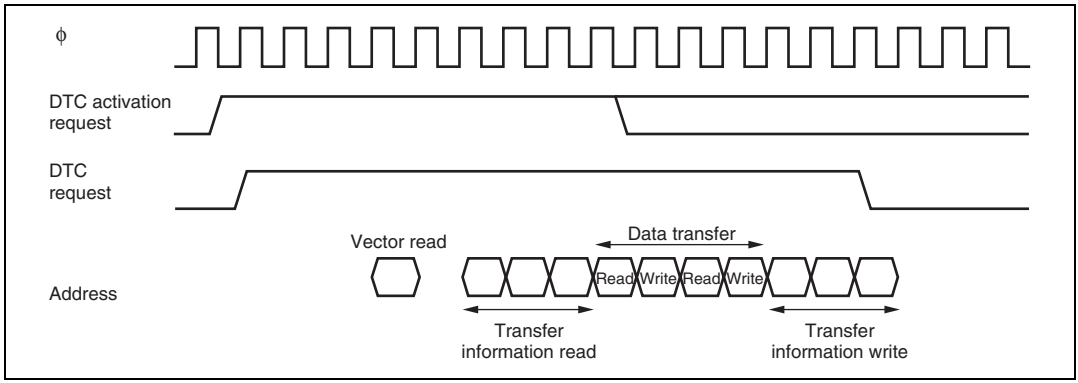


Figure 8.11 DTC Operation Timing (Example of Block Transfer Mode, with Block Size of 2)

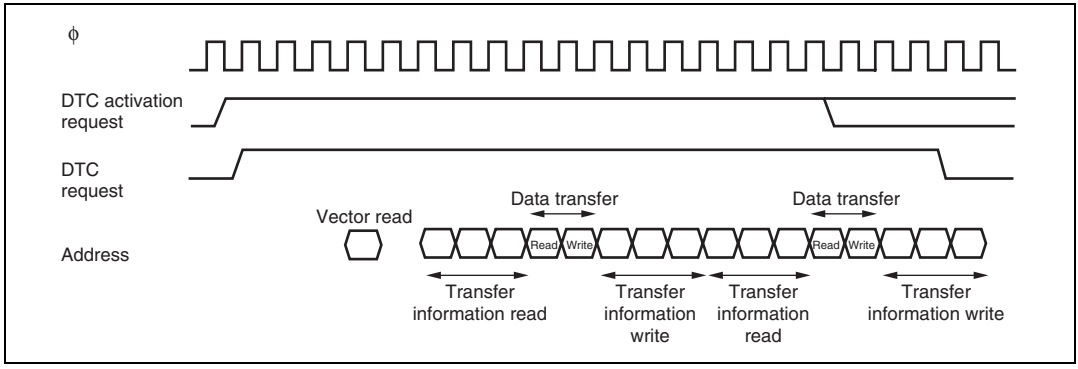


Figure 8.12 DTC Operation Timing (Example of Chain Transfer)

8.5.7 Number of DTC Execution States

Table 8.7 lists execution status for a single DTC data transfer, and table 8.8 shows the number of states required for each execution status.

Table 8.7 DTC Execution Status

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

Legend:

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

Table 8.8 Number of States Required for Each Execution Status

Object to be Accessed			On-Chip RAM	On-Chip ROM	On-Chip I/O Registers		External Devices			
Bus width			32	16	8	16	8		16	
Access states			1	1	2	2	2	3	2	3
Execution status	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							

Legend:

m: Number of wait states in an external device access

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.

8.6 Procedures for Using DTC

8.6.1 Activation by Interrupt

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

1. Set the MRA, MRB, SAR, DAR, CRA, and CRB register information in 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.

8.6.2 Activation by Software

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

1. Set the MRA, MRB, SAR, DAR, CRA, and CRB register information in 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 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.

8.7 Examples of Use of the DTC

8.7.1 Normal Mode

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

1. Set MRA to fixed source address (SM1 = SM0 = 0), incrementing destination address (DM1 = 1, DM0 = 0), normal mode (MD1 = MD0 = 0), and byte size (Sz = 0). The DTS bit can have any value. Set MRB for one data transfer by one interrupt (CHNE = 0, DISEL = 0). Set the SCI RDR address in SAR, the start address of the RAM area where the data will be received in DAR, and 128 (H'0080) in CRA. CRB can be set to any value.
2. Set the start address of the register information at the DTC vector address.
3. Set the corresponding bit in DTCER to 1.
4. Set the SCI to the appropriate receive mode. Set the RIE bit in SCR to 1 to enable the reception complete (RXI) interrupt. Since the generation of a receive error during the SCI reception operation will disable subsequent reception, the CPU should be enabled to accept receive error interrupts.
5. Each time 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.

8.7.2 Software Activation

An example is shown in which the DTC is used to transfer a block of 128 bytes of data by means of software activation. The transfer source address is H'1000 and the 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.

4. Write 1 to the SWDTE bit and the vector number (H'60) to DTVECR. The write data is H'E0.
5. Read DTVECR again and check that it is set to the vector number (H'60). If it is not, this indicates that the write failed. This is presumably because an interrupt occurred between steps 3 and 4 and led to a different software activation. To activate this transfer, go back to step 3.
6. If the write was successful, the DTC is activated and a block of 128 bytes of data is transferred.
7. After the transfer, an SWDTEND interrupt occurs. The interrupt handling routine should clear the SWDTE bit to 0 and perform other wrap-up processing.

8.8 Usage Notes

8.8.1 Module Stop

DTC operation can be prohibited or enabled using the module stop control register. Access to the register is prohibited in the module stop mode. However, the module stop mode cannot be specified while the DTC is operating. For details, see section 22, Power-Down Modes.

8.8.2 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.

8.8.3 DTCE Bit Setting

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

8.8.4 DMAC Transfer End Interrupt

When DTC transfer is activated by a DMAC transfer end interrupt, the DMAC's DTE bit is not subject to DTC control, regardless of the transfer counter and DISEL bit, and the write data has priority. Consequently, an interrupt request is not sent to the CPU when the DTC transfer counter reaches 0.

Section 9 I/O Ports

Table 9.1 summarizes the port functions. The pins of each port also have other functions such as input/output or external interrupt input pins of on-chip peripheral modules. Each I/O port includes a data direction register (DDR) that controls input/output, a data register (DR) that stores output data, and a port register (PORT) used to read the pin states. The input-only ports do not have a DR and a DDR.

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

Ports 3 and A include an open-drain control register (ODR) that controls the on/off state of the output buffer PMOS.

All the I/O ports can drive a single TTL load and 30 pF capacitive load.

Table 9.1 Port Functions (1)

Port	Description	Modes 4 and 5 Mode 6	Mode 7*	Input/Output Type
Port 1	General I/O port also functioning as TPU I/O pins, interrupt input pins, and external USB transceiver I/O	P17/TIOCB2/TCLKD/$\overline{OE}$	P17/TIOCB2/TCLKD	Schmitt triggered input ($\overline{IRQ1}$, $\overline{IRQ0}$)
		P16/TIOCA2/$\overline{IRQ1}$		
		P15/TIOCB1/TCLKC/FSE0	P15/TIOCB1/TCLKC	
		P14/TIOCA1/$\overline{IRQ0}$		
		P13/TIOCD0/TCLKB/A23/VPO	P13/TIOCD0/TCLKB	
		P12/TIOCC0/TCLKA/A22/RCV	P12/TIOCC0/TCLKA	
		P11/TIOCB0/A21/VP	P11/TIOCB0	
		P10/TIOCA0/A20/VM	P10/TIOCA0	
Port 3	General I/O port also functioning as SCI_0, SCI_1 pins and interrupt input pins	P36		Open-drain output Schmitt triggered input ($\overline{IRQ5}$, $\overline{IRQ4}$)
		P35/SCK1/$\overline{IRQ5}$		
		P34/RxD1		
		P33/TxD1		
		P32/SCK0/$\overline{IRQ4}$		
		P31/RxD0		
		P30/TxD0		
Port 4	General I/O port also functioning as A/D converter analog inputs	P43/AN3		
		P42/AN2		
		P41/AN1		
		P40/AN0		

Note: * The USB may be unusable in mode 7 in some cases. See section 3, MCU Operating Modes, for details.

Table 9.1 Port Functions (2)

Port	Description	Modes 4 and 5	Mode 6	Mode 7*1	Input/Output Type
Port 7	General I/O port also functioning as bus control output pins, manual reset input pin, and 8-bit timer I/O	P74/$\overline{\text{MRES}}$		P74/$\overline{\text{MRES}}$	
		P73/TMO1/$\overline{\text{CS7}}$		P73/TMO1	
		P72/TMO0/$\overline{\text{CS6}}$*2		P72/TMO0	
		P71/$\overline{\text{CS5}}$		P71	
		P70/TMRI01/TMCI01/$\overline{\text{CS4}}$		P70/TMRI01/TMCI01	
Port 9	General I/O port also functioning as D/A converter analog outputs and A/D converter analog input	P97/AN15/DA1			
		P96/AN14/DA0			
Port A	General I/O port also functioning as SCI_2 I/O pins, address output pins, and external USB transceiver output	PA3/A19/SCK2/SUSPND		PA3/SCK2	Built-in input pull-up MOS Open-drain output
		PA2/A18/RxD2		PA2/RxD2	
		PA1/A17/TxD2		PA1/TxD2	
		PA0/A16		PA0	
Port B	General I/O port also functioning as address output pins	PB7/A15		PB7	Built-in input pull-up MOS
		PB6/A14		PB6	
		PB5/A13		PB5	
		PB4/A12		PB4	
		PB3/A11		PB3	
		PB2/A10		PB2	
		PB1/A9		PB1	
		PB0/A8		PB0	
Port C	General I/O port also functioning as address output pins	A7	When DDR = 0: PC7	PC7	Built-in input pull-up MOS
			When DDR = 1: A7*2		
		A6	When DDR = 0: PC6	PC6	
			When DDR = 1: A6*2		
		A5	When DDR = 0: PC5	PC5	
			When DDR = 1: A5*2		

Notes: 1. The USB may be unusable in mode 7 in some cases. See section 3, MCU Operating Modes, for details.

2. $\overline{\text{CS6}}$ and A7 to A0 should be designated as an output when on-chip USB is used in mode 6.

Table 9.1 Port Functions (3)

Port	Description	Modes 4 and 5	Mode 6	Mode 7[*1]	Input/Output Type
Port C	General I/O port also functioning as address output pins	A4	When DDR = 0: PC4 When DDR = 1: A4[*2]	PC4	Built-in input pull-up MOS
		A3	When DDR = 0: PC3 When DDR = 1: A3[*2]	PC3	
		A2	When DDR = 0: PC2 When DDR = 1: A2[*2]	PC2	
		A1	When DDR = 0: PC1 When DDR = 1: A1[*2]	PC1	
		A0	When DDR = 0: PC0 When DDR = 1: A0[*2]	PC0	
Port D	General I/O port also functioning as data I/O pins	D15		PD7	Built-in input pull-up MOS
		D14		PD6	
		D13		PD5	
		D12		PD4	
		D11		PD3	
		D10		PD2	
		D9		PD1	
		D8		PD0	
Port E	General I/O port also functioning as address output pins	8-bit bus mode: PE7 16-bit bus mode: D7		PE7	Built-in input pull-up MOS
		8-bit bus mode: PE6 16-bit bus mode: D6		PE6	
		8-bit bus mode: PE5 16-bit bus mode: D5		PE5	
		8-bit bus mode: PE4 16-bit bus mode: D5		PE4	
		8-bit bus mode: PE3 16-bit bus mode: D3		PE3	
		8-bit bus mode: PE2 16-bit bus mode: D2		PE2	

Notes: 1. The USB may be unusable in mode 7 in some cases. See section 3, MCU Operating Modes, for details.

2. $\overline{CS6}$ and A7 to A0 should be designated as an output when on-chip USB is used in mode 6.

Table 9.1 Port Functions (4)

Port	Description	Modes 4 and 5	Mode 6	Mode 7*	Input/Output Type
Port E	General I/O port also functioning as address output pins	8-bit bus mode: PE1 16-bit bus mode: D1		PE1	Built-in input pull-up MOS
		8-bit bus mode: PE0 16-bit bus mode: D0		PE0	
Port F	General I/O port also functioning as interrupt input pins and bus control I/O pins	When DDR = 0: PF7 When DDR = 1 (after reset): ϕ		When DDR = 0 (after reset): PF7 When DDR = 1: ϕ	Schmitt triggered input ($\overline{IRQ3}$, $\overline{IRQ2}$)
		$\overline{AS}$		PF6	
		$\overline{RD}$		PF5	
		$\overline{HWR}$		PF4	
		8-bit bus mode: PF3/$\overline{ADTRG}$/$\overline{IRQ3}$ 16-bit bus mode: $\overline{LWR}$		PF3/$\overline{ADTRG}$/$\overline{IRQ3}$	
		When WAITE = 0 (after reset) : PF2 When WAITE = 1: $\overline{WAIT}$		PF2	
		When BRLE = 0 (after reset): PF1 When BRLE = 1: $\overline{BACK}$		PF1	
		When BRLE = 0 (after reset): PF0/$\overline{IRQ2}$ When BRLE = 1: $\overline{BREQ}$/$\overline{IRQ2}$		PF0/$\overline{IRQ2}$	
Port G	General I/O port also functioning as bus control input pins and interrupt Input pins	When DDR = 0 (after reset in mode 6): PG4 When DDR = 1 (after reset in modes 4, 5): $\overline{CS0}$		PG4	Schmitt triggered input ($\overline{IRQ7}$)
		When DDR = 0: PG3 When DDR = 1: $\overline{CS1}$		PG3	
		When DDR = 0: PG2 When DDR = 1: $\overline{CS2}$		PG2	
		When DDR = 0: PG1/$\overline{IRQ7}$ When DDR = 1: $\overline{CS3}$/$\overline{IRQ7}$		PG1/$\overline{IRQ7}$	
		PG0			

Note: * The USB may be unusable in mode 7 in some cases. See section 3, MCU Operating Modes, for details.

9.1 Port 1

Port 1 is an 8-bit I/O port. The port 1 has the following registers.

- Port 1 data direction register (P1DDR)
- Port 1 data register (P1DR)
- Port 1 register (PORT1)

9.1.1 Port 1 Data Direction Register (P1DDR)

P1DDR is an 8-bit write-only register, the individual bits of which specify input or output for the pins of port 1. Since this is a write-only register, bit manipulation instructions should not be used to write to it. For details, see section 2.9.4, Accessing Registers Containing Write-Only Bits.

Bit	Bit Name	Initial Value	R/W	Description
7	P17DDR	0	W	Modes 4 to 6 If address output is enabled by the setting of bits AE3 to AE0 in PFCR, pins P13 to P10 are address outputs. Pins P17 to P14, and pins P13 to P10 when address output is disabled, are output ports when the corresponding P1DDR bits are set to 1, and input ports when the corresponding P1DDR bits are cleared to 0. Mode 7 Setting a P1DDR bit to 1 makes the corresponding port 1 pin an output port, while clearing the bit to 0 makes the pin an input port.
6	P16DDR	0	W	
5	P15DDR	0	W	
4	P14DDR	0	W	
3	P13DDR	0	W	
2	P12DDR	0	W	
1	P11DDR	0	W	
0	P10DDR	0	W	

9.1.2 Port 1 Data Register (P1DR)

P1DR stores output data for the port 1 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P17DR	0	R/W	An output data for a pin is stored when the pin function is specified to a general purpose output port.
6	P16DR	0	R/W	
5	P15DR	0	R/W	
4	P14DR	0	R/W	
3	P13DR	0	R/W	
2	P12DR	0	R/W	
1	P11DR	0	R/W	
0	P10DR	0	R/W	

9.1.3 Port 1 Register (PORT1)

PORT1 shows the pin states.

Bit	Bit Name	Initial Value	R/W	Description
7	P17	—*	R	If a port 1 read is performed while P1DDR bits are set to 1, the P1DR value is read. If a port 1 read is performed while P1DDR bits are cleared to 0, the pin states are read.
6	P16	—*	R	
5	P15	—*	R	
4	P14	—*	R	
3	P13	—*	R	
2	P12	—*	R	
1	P11	—*	R	
0	P10	—*	R	

Note: * Determined by the states of pins P17 to P10.

9.1.4 Pin Functions

Port 1 pins also function as TPU I/O pins, external interrupt input pins ($\overline{IRQ1}$, $\overline{IRQ0}$), external USB transceiver input, and address bus (A23 to A20) output pins. The correspondence between the register specification and the pin functions is shown below.

Table 9.2 P17 Pin Function

FADSEL in UCTLR[*3]	0			1
TPU Channel 2 Setting[*1]	Output	Input or Initial Value		—
P17DDR	—	0	1	—
Pin function	TIOCB2 output	P17 input	P17 output	$\overline{OE}$ output[*3]
		TIOCB2 input		
	TCLKD input			

Table 9.3 P16 Pin Function

TPU Channel 2 Setting[*1]	Output	Input or Initial Value	
P16DDR	—	0	1
Pin function	TIOCA2 output	P16 input	P16 output
		TIOCA2 input	
	$\overline{IRQ1}$ input[*2]		

Notes: 1. For details on the TPU channel specification, refer to section 10, 16-Bit Timer Pulse Unit (TPU).
2. When used as an external interrupt pin, do not use for another functions.
3. The USB may be unusable in mode 7 in some cases. See section 3, MCU Operating Modes, for details.

Table 9.4 P15 Pin Function

FADSEL in UCTLR[*3]	0			1
TPU Channel 1 Setting[*1]	Output	Input or Initial Value		—
P15DDR	—	0	1	—
Pin function	TIOCB1 output	P15 input	P15 output	FSE0 output[*3]
		TIOCB1 input		
	TCLKC input			

Table 9.5 P14 Pin Function

TPU Channel 1 Setting[*1]	Output	Input or Initial Value	
P14DDR	—	0	1
Pin function	TIOCA1 output	P14 input	P14 output
		TIOCA1 input	
	$\overline{\text{IRQ0}}$ input[*2]		

Notes: 1. For details on the TPU channel specification, refer to section 10, 16-Bit Timer Pulse Unit (TPU).
2. When used as an external interrupt pin, do not use for another functions.
3. On-chip USB cannot be used in mode 7.

Table 9.6 P13 Pin Function

AE3 to AE0[*1]	Other than B'1111				B'1111
FADSEL in UCTLR[*3]	0			1	—
TPU Channel 0 Setting[*2]	Output	Input or Initial Value		—	—
P13DDR	—	0	1	—	—
Pin function	TIOCD0 output	P13 input	P13 output	VPO output[*3]	A23 output
		TIOCD0 input			
	TCLKB input				

Notes: 1. Valid in modes 4 to 6.
2. For details on the TPU channel specification, refer to section 10, 16-Bit Timer Pulse Unit (TPU).
3. The USB may be unusable in mode 7 in some cases. See section 3, MCU Operating Modes, for details.

Table 9.7 P12 Pin Function

AE3 to AE0[*1]	Other than B'1111				B'1111
FADSEL in UCTLR[*3]	0			1	—
TPU Channel 0 Setting[*2]	Output	Input or Initial Value		—	—
P12DDR	—	0	1	—	—
Pin function	TIOCC0 output	P12 input	P12 output	RCV input[*3]	A22 output
		TIOCC0 input			
	TCLKA input				

Table 9.8 P11 Pin Function

AE3 to AE0[*1]	Other than (B'1110 to B'1111)				B'1110 to B'1111
FADSEL in UCTLR[*3]	0			1	—
TPU Channel 0 Setting[*2]	Output	Input or Initial Value		—	—
P11DDR	—	0	1	—	—
Pin function	TIOCB0 output	P11 input	P11 output	VP input[*3]	A21 output
		TIOCB0 input			

Table 9.9 P10 Pin Function

AE3 to AE0[*1]	Other than (B'1101 to B'1111)				B'1101 to B'1111
FADSEL in UCTLR[*3]	0			1	—
TPU Channel 0 Setting[*2]	Output	Input or Initial Value		—	—
P10DDR	—	0	1	—	—
Pin function	TIOCA0 output	P10 input	P10 output	VM input	A20 output
		TIOCA0 input			

Notes: 1. Valid in modes 4 to 6.

2. For details on the TPU channel specification, refer to section 10, 16-Bit Timer Pulse Unit (TPU).
3. The USB may be unusable in mode 7 in some cases. See section 3, MCU Operating Modes, for details.

9.2 Port 3

Port 3 is a 7-bit I/O port also functioning as SCI I/O and external interrupt input ($\overline{\text{IRQ4}}$, $\overline{\text{IRQ5}}$).

- Port 3 data direction register (P3DDR)
- Port 3 data register (P3DR)
- Port 3 register (PORT3)
- Port 3 open-drain control register (P3ODR)

9.2.1 Port 3 Data Direction Register (P3DDR)

The individual bits of P3DDR specify input or output for the pins of port 3. Since this is a write-only register, bit manipulation instructions should not be used to write to it. For details, see section 2.9.4, Accessing Registers Containing Write-Only Bits.

Bit	Bit Name	Initial Value	R/W	Description
7	—	Undefined	—	Reserved This bit is undefined and cannot be modified.
6	P36DDR	0	W	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.
5	P35DDR	0	W	
4	P34DDR	0	W	
3	P33DDR	0	W	
2	P32DDR	0	W	
1	P31DDR	0	W	
0	P30DDR	0	W	

9.2.2 Port 3 Data Register (P3DR)

P3DR stores output data for the port 3 pins (P36 to P30).

Bit	Bit Name	Initial Value	R/W	Description
7	—	Undefined	—	Reserved This bit is undefined and cannot be modified.
6	P36DR	0	R/W	An output data for a pin is stored when the pin function is specified to a general purpose output port.
5	P35DR	0	R/W	
4	P34DR	0	R/W	
3	P33DR	0	R/W	
2	P32DR	0	R/W	
1	P31DR	0	R/W	
0	P30DR	0	R/W	

9.2.3 Port 3 Register (PORT3)

PORT3 shows the pin states.

Bit	Bit Name	Initial Value	R/W	Description
7	—	Undefined	—	Reserved This bit is undefined.
6	P36	—*	R	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.
5	P35	—*	R	
4	P34	—*	R	
3	P33	—*	R	
2	P32	—*	R	
1	P31	—*	R	
0	P30	—*	R	

Note: * Determined by the state of pins P36 to P30.

9.2.4 Port 3 Open-Drain Control Register (P3ODR)

P3ODR controls the PMOS on/off status for each port 3 pin (P36 to P30).

Bit	Bit Name	Initial Value	R/W	Description
7	—	Undefined	—	Reserved This bit is undefined and cannot be modified.
6	P36ODR	0	R/W	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.
5	P35ODR	0	R/W	
4	P34ODR	0	R/W	
3	P33ODR	0	R/W	
2	P32ODR	0	R/W	
1	P31ODR	0	R/W	
0	P30ODR	0	R/W	

9.2.5 Pin Functions

Port 3 pins also function as SCI I/O pins and external interrupt input pins ($\overline{\text{IRQ4}}$, $\overline{\text{IRQ5}}$). Port 3 pin functions are shown below.

Table 9.10 P36 Pin Function

P36DDR	0	1
Pin function	P36 input	P36 output (USB D+ pull-up control output in HD64F2215U, HD64F2215RU, HD64F2215TU)

Table 9.11 P35 Pin Function

CKE1 in SCR_1	0				1
C/$\overline{A}$ in SMR_1	0			1	—
CKE0 in SCR_1	0		1	—	—
P35DDR	0	1	—	—	—
Pin function	P35 input	P35 output*2	SCK1 output*2	SCK1 output*2	SCK1 input
	$\overline{IRQ5}$ input*1				

Notes: 1. When used as an external interrupt pin, do not use for another function.

2. Note on Development Using the E6000 Emulator

The H8S/2215 Group does not have an I^2C bus function and pins 35 and 34 are used for CMOS output (except when P35ODR and P34ODR are set to 1). The E6000 emulator expects pins 35 and 34 to be used for NMOS push-pull output, which differs from the pin output characteristics of the H8S/2215 Group. If it is necessary to use pins 35 and 34 for CMOS output, an appropriate resistance should be used for pull-up when using the H8S/2215 with the E6000.

Table 9.12 P34 Pin Function

RE in SCR_1	0		1
P34DDR	0	1	—
Pin function	P34 input	P34 output*	RxD1 input

Note: * Note on Development Using the E6000 Emulator

The H8S/2215 Group does not have an I^2C bus function and pins 35 and 34 are used for CMOS output (except when P35ODR and P34ODR are set to 1). The E6000 emulator expects pins 35 and 34 to be used for NMOS push-pull output, which differs from the pin output characteristics of the H8S/2215 Group. If it is necessary to use pins 35 and 34 for CMOS output, an appropriate resistance should be used for pull-up when using the H8S/2215 with the E6000.

Table 9.13 P33 Pin Function

SMIF in SCMR_1	0			1			
TE in SCR_1	0		1	0		1	
P33DDR	0	1	—	0	1	0	1
Pin function	P33 input	P33 output	TxD1 output	P33 input	Setting prohibited	TxD1 output	Setting prohibited

Table 9.14 P32 Pin Function

CKE1 in SCR_0	0				1
C/$\overline{A}$ in SMR_0	0			1	—
CKE0 in SCR_0	0		1	—	—
P32DDR	0	1	—	—	—
Pin function	P32 input	P32 output	SCK0 output	SCK0 output	SCK0 input
	$\overline{IRQ4}$ input*				

Note: * When used as an external interrupt pin, do not use for another function.

Table 9.15 P31 Pin Function

RE in SCR_0	0		1
P31DDR	0	1	—
Pin function	P31 input	P31 output	RxD0 input

Table 9.16 P30 Pin Function

SMIF in SCMR_0	0			1			
TE in SCR_0	0		1	0		1	
P30DDR	0	1	—	0	1	0	1
Pin function	P30 input	P30 output	TxD0 output	P30 input	Setting prohibited	TxD0 output	Setting prohibited

9.3 Port 4

Port 4 is a 4-bit I/O port also functioning as A/D converter analog input. Port 4 has the following register.

- Port 4 register (PORT4)

9.3.1 Port 4 Register (PORT4)

PORT4 shows port 4 pin states. PORT4 cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	Undefined	—	Reserved These bits are undefined.
3	P43	—*	R	The pin states are always read when a port 4 read is performed.
2	P41	—*	R	
1	P41	—*	R	
0	P40	—*	R	

Note: * Determined by the states of pins P43 to P40.

9.3.2 Pin Function

Port 4 also functions as A/D converter analog input (AN3 to AN0).

9.4 Port 7

Port 7 is a 5-bit I/O port also functioning as bus control output, manual reset input, and 8-bit timer I/O. Port 7 has the following registers.

- Port 7 data direction register (P7DDR)
- Port 7 data register (P7DR)
- Port 7 register (PORT7)

9.4.1 Port 7 Data Direction Register (P7DDR)

P7DDR is an 8-bit write-only register, the individual bits of which specify input or output for the pins of port 7. P7DDR cannot be read; if it is, an undefined value will be read. Since this is a write-only register, bit manipulation instructions should not be used to write to it. For details, see section 2.9.4, Accessing Registers Containing Write-Only Bits.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	Undefined	—	Reserved These bits are undefined and cannot be modified.
4	P74DDR	0	W	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 pin.
3	P73DDR	0	W	
2	P72DDR	0	W	
1	P71DDR	0	W	
0	P70DDR	0	W	

9.4.2 Port 7 Data Register (P7DR)

P7DR stores output data for the port 7 pins.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	Undefined	—	Reserved These bits are undefined and cannot be modified.
4	P74DR	0	R/W	Stores output data for the port 7 pins.
3	P73DR	0	R/W	
2	P72DR	0	R/W	
1	P71DR	0	R/W	
0	P70DR	0	R/W	

9.4.3 Port 7 Register (PORT7)

PORT7 shows the pin states.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	Undefined	—	Reserved These bits are undefined and cannot be modified.
4	P74	—*	R	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.
3	P73	—*	R	
2	P72	—*	R	
1	P71	—*	R	
0	P70	—*	R	

Note: * Determined by the state of pins P74 to P70.

9.4.4 Pin Functions

Port 7 pins also function as bus control output pins, manual reset input pin, and 8 bit timer input/output. Port 7 pin functions are shown below.

Table 9.17 P74 Pin Function

MRESE	0		1
P74DDR	0	1	—
Pin function	P74 input	P74 output	$\overline{\text{MRES}}$ input

Table 9.18 P73 Pin Function

Operating Mode	Modes 4 to 6			Mode 7		
OS3 to OS0 in TCSR_1	OS3 to OS0 are all 0		At least one of OS3 to OS0 is 1	OS3 to OS0 are all 0		At least one of OS3 to OS0 is 1
P73DDR	0	1	—	0	1	—
Pin function	P73 input	$\overline{\text{CS7}}$ output	TMO1 output	P73 input	P73 output	TMO1 output

Table 9.19 P72 Pin Function

Operating Mode	Modes 4 to 6*			Mode 7		
OS3 to OS0 in TCSR_0	OS3 to OS0 are all 0s		At least one of OS3 to OS0 is 1	OS3 to OS0 are all 0		At least one of OS3 to OS0 is 1
P72DDR	0	1	—	0	1	—
Pin function	P72 input	$\overline{CS6}$ output	TMO0 output	P72 input	P72output	TMO0 output

Note: * When on-chip USB is used in modes 4 to 6, bit P72DDR should be set to 1 so that the pin outputs $\overline{CS6}$.

Table 9.20 P71 Pin Function

Operating Mode	Modes 4 to 6		Mode 7	
P71DDR	0	1	0	1
Pin function	P71 input	$\overline{CS5}$ output	P71 input	P71 output

Table 9.21 P70 Pin Function

Operating Mode	Modes 4 to 6		Mode 7	
P70DDR	0	1	0	1
Pin function	P70 input	$\overline{CS4}$ output	P70 input	P70 output
	TMRI01, TMCI01 input			

9.5 Port 9

Port 9 pins also function as A/D converter analog input and D/A converter analog output pins. The port 9 has the following register.

- Port 9 register (PORT9)

9.5.1 Port 9 Register (PORT9)

PORT9 shows port 9 pin states.

Bit	Bit Name	Initial Value	R/W	Description
7	P97	—*	R	The pin states are always read when a port 9 read is performed.
6	P96	—*	R	
5 to 0	—	Undefined	—	Reserved These bits are undefined.

Note: * Determined by the states of pins P97 and P96.

9.5.2 Pin Function

Port 9 also functions as A/D converter analog input (AN15, AN14) and D/A converter analog output (DA1, DA0).

9.6 Port A

Port A is a 4-bit I/O port that also functions as address bus (A19 to A16) output, external USB transceiver output, and SCI_2 I/O, and interrupt input. The port A has the following registers.

- Port A data direction register (PADDR)
- Port A data register (PADR)
- Port A register (PORTA)
- Port A pull-up MOS control register (PAPCR)
- Port A open-drain control register (PAODR)

9.6.1 Port A Data Direction Register (PADDR)

The individual bits of PADDR specify input or output for the pins of port A. Since this is a write-only register, bit manipulation instructions should not be used to write to it. For details, see section 2.9.4, Accessing Registers Containing Write-Only Bits.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	Undefined	—	Reserved These bits are undefined and cannot be modified.
3	PA3DDR	0	W	Modes 4 to 6 If address output is enabled by the setting of bits AE3 to AE0 in PFCR, the corresponding port A pins are address outputs. When address output is disabled, setting a PADDR bit to 1 makes the corresponding port A pin an output port, while clearing the bit to 0 makes the pin an input port. Mode 7 Setting a PADDR bit to 1 makes the corresponding port A pin an output port, while clearing the bit to 0 makes the pin an input port.
2	PA2DDR	0	W	
1	PA1DDR	0	W	
0	PA0DDR	0	W	

9.6.2 Port A Data Register (PADR)

PADR stores output data for the port A pins.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	Undefined	—	Reserved These bits are undefined and cannot be modified.
3	PA3DR	0	R/W	An output data for a pin is stored when the pin function is specified to a general purpose output port.
2	PA2DR	0	R/W	
1	PA1DR	0	R/W	
0	PA0DR	0	R/W	

9.6.3 Port A Register (PORTA)

PORTA shows port A pin states.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	Undefined	—	Reserved These bits are undefined and cannot be modified.
3	PA3	—*	R	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.
2	PA2	—*	R	
1	PA1	—*	R	
0	PA0	—*	R	

Note: * Determined by the states of pins PA3 to PA0.

9.6.4 Port A MOS Pull-Up Control Register (PAPCR)

PAPCR controls the function of the port A input pull-up MOS. PAPCR is valid for port input and SCI input pins.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	Undefined	—	Reserved These bits are undefined and cannot be modified.
3	PA3PCR*	0	R/W	When a pin function is specified to an input port, setting the corresponding bit to 1 turns on the input pull-up MOS for that pin.
2	PA2PCR	0	R/W	
1	PA1PCR	0	R/W	
0	PA0PCR	0	R/W	

Note: * Set PA3PCR to 0 when FADSEL of USB is 1.

9.6.5 Port A Open Drain Control Register (PAODR)

PAODR specifies an output type of port A. PAODR is valid for port output and SCI output pins.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	Undefined	—	Reserved These bits are undefined and cannot be modified.
3	PA3ODR	0	R/W	Setting a PAODR bit to 1 makes the corresponding port A pin an NMOS open-drain output pin, while clearing the bit to 0 makes the pin a CMOS output pin.
2	PA2ODR	0	R/W	
1	PA1ODR	0	R/W	
0	PA0ODR	0	R/W	

9.6.6 Pin Functions

Port A pins also function as address bus (A19 to A16) output, external USB transceiver output, SCI_2 I/O, and interrupt input. The correspondence between the register specification and the pin functions is shown below.

Table 9.22 PA3 Pin Function

Operating mode	Modes 4 to 6						
AE3 to AE0	11xx	Other than 11xx					
FADSEL of UCTLR	—	0					1
CKE1 in SCR_2	—	0				1	—
C/$\overline{A}$ in SMR_2	—	0			1	—	—
CKE0 in SCR_2	—	0		1	—	—	—
PA3DDR	—	0	1	—	—	—	—
Pin function	A19 output	PA3 input	PA3 output	SCK2 output	SCK2 output	SCK2 input	SUSPND output

Operating mode	Mode 7					
AE3 to AE0	—					
FADSEL of UCTLR*	0					1*
CKE1 in SCR_2	0				1	—
C/$\overline{A}$ in SMR_2	0			1	—	—
CKE0 in SCR_2	0		1	—	—	—
PA3DDR	0	1	—	—	—	—
Pin function	PA3 input	PA3 output	SCK2 output	SCK2 output	SCK2 input	SUSPND output*

Note: * On-chip USB cannot be used in mode 7.

Table 9.23 PA2 Pin Function

Operating mode	Modes 4 to 6				Mode 7		
AE3 to AE0	1011 or 11xx	Other than 1011 or 11xx			—		
RE in SCR_2	—	0		1	0		1
PA2DDR	—	0	1	—	0	1	—
Pin function	A18 output	PA2 input	PA2 output	RxD2 input	PA2 input	PA2 output	RxD2 input

Table 9.24 PA1 Pin Function

Operating mode	Modes 4 to 6							
AE3 to AE0	101x or 11xx	Other than 101x or 11xx						
SMIF in SCMR_2	—	0			1			
TE in SCR_2	—	0		1	0		1	
PA1DDR	—	0	1	—	0	1	0	1
Pin function	A17 output	PA1 input	PA1 output	TxD2 output	PA1 input	Setting prohibited	TxD2 output	Setting prohibited

Operating mode	Mode 7						
SMIF in SCMR_2	0			1			
TE in SCR_2	0		1	0		1	
PA1DDR	0	1	—	0	1	0	1
Pin function	PA1 input	PA1 output	TxD2 output	PA1 input	Setting prohibited	TxD2 output	Setting prohibited

Table 9.25 PA0 Pin Function

Operating mode	Modes 4 to 6			Mode 7	
AE3 to AE0	Other than 0xxxx or 1000	0xxx or 1000		—	
PA0DDR	—	0	1	0	1
Pin function	PA16 output	PA0 input	PA0 output	PA0 input	PA0 output

9.6.7 Port A Input Pull-Up MOS Function

Port A has a built-in input pull-up MOS function that can be controlled by software. Input pull-up MOS can be specified as on or off for individual bits.

Table 9.26 summarizes the input pull-up MOS states.

Table 9.26 Input Pull-Up MOS States (Port A)

Pins	Power-On Reset	Hardware Standby Mode	Manual Reset	Software Standby Mode	In Other Operations
Address output, port output, SCI output	OFF		OFF		
Port input, SCI input			ON/OFF		

Legend:
OFF: Input pull-up MOS is always off.
ON/OFF: On when PADDR = 0 and PAPCR = 1; otherwise off.

9.7 Port B

Port B is an 8-bit I/O port that also has address bus (A15 to A8) output. The port B has the following registers. Internal I/O Register.

- Port B data direction register (PBDDR)
- Port B data register (PBDR)
- Port B register (PORTB)
- Port B MOS pull-up control register (PBPCR)

9.7.1 Port B Data Direction Register (PBDDR)

The individual bits of PBDDR specify input or output for the pins of port B. Since this is a write-only register, bit manipulation instructions should not be used to write to it. For details, see section 2.9.4, Accessing Registers Containing Write-Only Bits.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7DDR	0	W	Modes 4 to 6 If address output is enabled by the setting of bits AE3 to AE0 in PFCR, the corresponding port B pins are address outputs. When address output is disabled, 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. 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.
6	PB6DDR	0	W	
5	PB5DDR	0	W	
4	PB4DDR	0	W	
3	PB3DDR	0	W	
2	PB2DDR	0	W	
1	PB1DDR	0	W	
0	PB0DDR	0	W	

9.7.2 Port B Data Register (PBDR)

PBDR stores output data for the port B pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7DR	0	R/W	An output data for a pin is stored when the pin function is specified to a general purpose output port.
6	PB6DR	0	R/W	
5	PB5DR	0	R/W	
4	PB4DR	0	R/W	
3	PB3DR	0	R/W	
2	PB2DR	0	R/W	
1	PB1DR	0	R/W	
0	PB0DR	0	R/W	

9.7.3 Port B Register (PORTB)

PORTB shows port B pin states.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7	—*	R	If the 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.
6	PB6	—*	R	
5	PB5	—*	R	
4	PB4	—*	R	
3	PB3	—*	R	
2	PB2	—*	R	
1	PB1	—*	R	
0	PB0	—*	R	

Note: * Determined by the status of pins PB7 to PB0.

9.7.4 Port B MOS Pull-Up Control Register (PBPCR)

PBPCR controls the on/off state of input pull-up MOS of port B.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7PCR	0	R/W	When a pin functions specified to an input port, setting the corresponding bit to 1 turns on the input pull-up MOS for that pin.
6	PB6PCR	0	R/W	
5	PB5PCR	0	R/W	
4	PB4PCR	0	R/W	
3	PB3PCR	0	R/W	
2	PB2PCR	0	R/W	
1	PB1PCR	0	R/W	
0	PB0PCR	0	R/W	

9.7.5 Pin Functions

Port B pins also function as address bus (A15 to A9) output pins. The correspondence between the register specification and the pin functions is shown below.

Table 9.27 PB7 Pin Function

Operating mode	Modes 4 to 6			Mode 7	
AE3 to AE0	B'1xxx	Other than B'1xxx		—	
PB7DDR	—	0	1	0	1
Pin function	A15 output	PB7 input	PB7 output	PB7 input	PB7 output

Table 9.28 PB6 Pin Function

Operating mode	Modes 4 to 6			Mode 7	
AE3 to AE0	B'0111 or B'1xxx	Other than B'0111 or B'1xxx		—	
PB6DDR	—	0	1	0	1
Pin function	A14 output	PB6 input	PB6 output	PB6 input	PB6 output

Table 9.29 PB5 Pin Function

Operating mode	Modes 4 to 6			Mode 7	
AE3 to AE0	B'011x or B'1xxx	Other than B'011x or B'1xxx		—	
PB5DDR	—	0	1	0	1
Pin function	A13 output	PB5 input	PB5 output	PB5 input	PB5 output

Table 9.30 PB4 Pin Function

Operating mode	Modes 4 to 6			Mode 7	
AE3 to AE0	Other than B'0100 or B'00xx	B'0100 or B'00xx		—	
PB4DDR	—	0	1	0	1
Pin function	A12 output	PB4 input	PB4 output	PB4 input	PB4 output

Table 9.31 PB3 Pin Function

Operating mode	Modes 4 to 6			Mode 7	
AE3 to AE0	Other than B'00xx	B'00xx		—	
PB3DDR	—	0	1	0	1
Pin function	A11 output	PB3 input	PB3 output	PB3 input	PB3 output

Table 9.32 PB2 Pin Function

Operating mode	Modes 4 to 6			Mode 7	
AE3 to AE0	Other than B'0010 or B'000x	B'0010 or B'000x		—	
PB2DDR	—	0	1	0	1
Pin function	A10 output	PB2 input	PB2 output	PB2 input	PB2 output

Table 9.33 PB1 Pin Function

Operating mode	Modes 4 to 6			Mode 7	
AE3 to AE0	Other than B'000x	B'000x		—	
PB1DDR	—	0	1	0	1
Pin function	A9 output	PB1 input	PB1 output	PB1 input	PB1 output

Table 9.34 PB0 Pin Function

Operating mode	Modes 4 to 6			Mode 7	
AE3 to AE0	Other than B'0000	B'0000		—	
PB0DDR	0	0	1	0	1
Pin function	A8 output	PB0 input	PB0 output	PB0 input	PB0 output

9.7.6 Port B Input Pull-Up MOS Function

Port B has a built-in input pull-up MOS function that can be controlled by software. Input pull-up MOS can be specified as on or off for individual bits.

Table 9.35 summarizes the input pull-up MOS states.

Table 9.35 Input Pull-Up MOS States (Port B)

Pins	Power-On Reset	Hardware Standby Mode	Manual Reset	Software Standby Mode	In Other Operations
Address output, port output	OFF		OFF		
Port input			ON/OFF		

Legend:
OFF: Input pull-up MOS is always off.
ON/OFF: On when PBDDR = 0 and PBPCR = 1; otherwise off.

9.8 Port C

Port C is an 8-bit I/O port that also has address bus (A7 to A0) output pins. The port C has the following registers.

- Port C data direction register (PCDDR)
- Port C data register (PCDR)
- Port C register (PORTC)
- Port C pull-up MOS control register (PCPCR)

Note: When using the on-chip USB in mode 6, set PCDDR so that addresses A7 to A0 are output from PC7 to PC0.

9.8.1 Port C Data Direction Register (PCDDR)

The individual bits of PCDDR specify input or output for the pins of port C. Since this is a write-only register, bit manipulation instructions should not be used to write to it. For details, see section 2.9.4, Accessing Registers Containing Write-Only Bits.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7DDR	0	W	Modes 4 and 5 Port C pins are address outputs regardless of the PCDDR settings. 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.
6	PC6DDR	0	W	
5	PC5DDR	0	W	
4	PC4DDR	0	W	
3	PC3DDR	0	W	
2	PC2DDR	0	W	
1	PC1DDR	0	W	
0	PC0DDR	0	W	

9.8.2 Port C Data Register (PCDR)

PCDR stores output data for the port C pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7DR	0	R/W	An output data for a pin is stored when the pin function is specified to a general purpose output port.
6	PC6DR	0	R/W	
5	PC5DR	0	R/W	
4	PC4DR	0	R/W	
3	PC3DR	0	R/W	
2	PC2DR	0	R/W	
1	PC1DR	0	R/W	
0	PC0DR	0	R/W	

9.8.3 Port C Register (PORTC)

PORTC shows port C pin states.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7	—*	R	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.
6	PC6	—*	R	
5	PC5	—*	R	
4	PC4	—*	R	
3	PC3	—*	R	
2	PC2	—*	R	
1	PC1	—*	R	
0	PC0	—*	R	

Note: * Determined by the states of pins PC7 to PC0.

9.8.4 Port C Pull-Up MOS Control Register (PCPCR)

PCPCR controls the on/off state of input pull-up MOS of port C.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7PCR	0	R/W	When a pin function is specified to an input port, setting the corresponding bit to 1 turns on the input pull-up MOS for that pin.
6	PC6PCR	0	R/W	
5	PC5PCR	0	R/W	
4	PC4PCR	0	R/W	
3	PC3PCR	0	R/W	
2	PC2PCR	0	R/W	
1	PC1PCR	0	R/W	
0	PC0PCR	0	R/W	

9.8.5 Pin Functions

Port C pins also function as Address bus (A7 to A0) output. The correspondence between the register specification and the pin functions is shown below.

Table 9.36 PC7 Pin Function

Operating Mode	Modes 4 and 5	Mode 6*		Mode 7	
PC7DDR	—	0	1	0	1
Pin Function	A7 output	PC7 input	A7 output	PC7 input	PC7 output

Table 9.37 PC6 Pin Function

Operating Mode	Modes 4 and 5	Mode 6*		Mode 7	
PC6DDR	—	0	1	0	1
Pin Function	A6 output	PC6 input	A6 output	PC6 input	PC6 output

Table 9.38 PC5 Pin Function

Operating Mode	Modes 4 and 5	Mode 6*		Mode 7	
PC5DDR	—	0	1	0	1
Pin Function	A5 output	PC5 input	A5 output	PC5 input	PC5 output

Table 9.39 PC4 Pin Function

Operating Mode	Modes 4 and 5	Mode 6*		Mode 7	
PC4DDR	—	0	1	0	1
Pin Function	A4 output	PC4 input	A4 output	PC4 input	PC4 output

Table 9.40 PC3 Pin Function

Operating Mode	Modes 4 and 5	Mode 6*		Mode 7	
PC3DDR	—	0	1	0	1
Pin Function	A3 output	PC3 input	A3 output	PC3 input	PC3 output

Table 9.41 PC2 Pin Function

Operating Mode	Modes 4 and 5	Mode 6*		Mode 7	
PC2DDR	—	0	1	0	1
Pin Function	A2 output	PC2 input	A2 output	PC2 input	PC2 output

Table 9.42 PC1 Pin Function

Operating Mode	Modes 4 and 5	Mode 6*		Mode 7	
PC1DDR	—	0	1	0	1
Pin Function	A1 output	PC1 input	A1 output	PC1 input	PC1 output

Table 9.43 PC0 Pin Function

Operating Mode	Modes 4 and 5	Mode 6*		Mode 7	
PC0DDR	—	0	1	0	1
Pin Function	A0 output	PC0 input	A0 output	PC0 input	PC0 output

Note: * When on-chip USB is used in mode 6, bits PC7DDR to PC0DDR should be set to H'FF so that the pins output A7 to A0.

9.8.6 Port C Input Pull-Up MOS Function

Port C has a built-in input pull-up MOS function that can be controlled by software. Input pull-up MOS can be used in modes 6 and 7, and can be specified as on or off for individual bits.

Table 9.44 summarizes the input pull-up MOS states.

Table 9.44 Input Pull-Up MOS States (Port C)

Pins	Power-On Reset	Hardware Standby Mode	Manual Reset	Software Standby Mode	In Other Operations
Address output (modes 4 and 5), port output (modes 6 and 7)	OFF		OFF		
Port input (modes 6 and 7)			ON/OFF		

Legend:
OFF: Input pull-up MOS is always off.
ON/OFF: On when PCDDR = 0 and PCPCR = 1; otherwise off.

9.9 Port D

Port D is an 8-bit I/O port that also has data bus (D15 to D8) I/O. The port D has the following registers.

- Port D data direction register (PDDDR)
- Port D data register (PDDR)
- Port D register (PORTD)
- Port D Pull-up MOS control register (PDPCR)

9.9.1 Port D Data Direction Register (PDDDR)

The individual bits of PDDDR specify input or output for the pins of port D. Since this is a write-only register, bit manipulation instructions should not be used to write to it. For details, see section 2.9.4, Accessing Registers Containing Write-Only Bits.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7DDR	0	W	Modes 4 to 6 Port D pins automatically function as data input/output pins. 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.
6	PD6DDR	0	W	
5	PD5DDR	0	W	
4	PD4DDR	0	W	
3	PD3DDR	0	W	
2	PD2DDR	0	W	
1	PD1DDR	0	W	
0	PD0DDR	0	W	

9.9.2 Port D Data Register (PDDR)

PDDR stores output data for the port D pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7DR	0	R/W	An output data for a pin is stored when the pin function is specified to a general purpose I/O output port.
6	PD6DR	0	R/W	
5	PD5DR	0	R/W	
4	PD4DR	0	R/W	
3	PD3DR	0	R/W	
2	PD2DR	0	R/W	
1	PD1DR	0	R/W	
0	PD0DR	0	R/W	

9.9.3 Port D Register (PORTD)

PORTD shows port D pin states.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7	—*	R	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.
6	PD6	—*	R	
5	PD5	—*	R	
4	PD4	—*	R	
3	PD3	—*	R	
2	PD2	—*	R	
1	PD1	—*	R	
0	PD0	—*	R	

Note: * Determined by the states of pins PD7 to PD0.

9.9.4 Port D Pull-Up MOS Control Register (PDPCR)

PDPCR controls on/off states of the input pull-up MOS of port D.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7PCR	0	R/W	When the pin is in its input state, the input pull-up MOS of the input pin is on when the corresponding bit is set to 1.
6	PD6PCR	0	R/W	
5	PD5PCR	0	R/W	
4	PD4PCR	0	R/W	
3	PD3PCR	0	R/W	
2	PD2PCR	0	R/W	
1	PD1PCR	0	R/W	
0	PD0PCR	0	R/W	

9.9.5 Pin Functions

Port D pins also functions as data bus (D15 to D8) I/O. The correspondence between the register specification and the pin functions in shown below.

Table 9.45 PD7 Pin Function

Operating Mode	Modes 4 to 6	Mode 7	
PD7DDR	—	0	1
Pin Function	D15 input/output	PD7 input	PD7 output

Table 9.46 PD6 Pin Function

Operating Mode	Modes 4 to 6	Mode 7	
PD6DDR	—	0	1
Pin Function	D14 input/output	PD6 input	PD6 output

Table 9.47 PD5 Pin Function

Operating Mode	Modes 4 to 6	Mode 7	
PD5DDR	—	0	1
Pin Function	D13 input/output	PD5 input	PD5 output

Table 9.48 PD4 Pin Function

Operating Mode	Modes 4 to 6	Mode 7	
PD4DDR	—	0	1
Pin Function	D12 input/output	PD4 input	PD4 output

Table 9.49 PD3 Pin Function

Operating Mode	Modes 4 to 6	Mode 7	
PD3DDR	—	0	1
Pin Function	D11 input/output	PD3 input	PD3 output

Table 9.50 PD2 Pin Function

Operating Mode	Modes 4 to 6	Mode 7	
PD2DDR	—	0	1
Pin Function	D10 input/output	PD2 input	PD2 output

Table 9.51 PD1 Pin Function

Operating Mode	Modes 4 to 6	Mode 7	
PD1DDR	—	0	1
Pin Function	D9 input/output	PD1 input	PD1 output

Table 9.52 PD0 Pin Function

Operating Mode	Modes 4 to 6	Mode 7	
PD0DDR	—	0	1
Pin Function	D8 input/output	PD0 input	PD0 output

9.9.6 Port D Input Pull-Up MOS Function

Port D has a built-in input pull-up MOS function that can be controlled by software. Input pull-up MOS can be used in mode 7, and can be specified as on or off for individual bits.

Table 9.53 summarizes the input pull-up MOS states.

Table 9.53 Input Pull-Up MOS States (Port D)

Pins	Power-On Reset	Hardware Standby Mode	Manual Reset	Software Standby Mode	In Other Operations
Address output (modes 4 to 6), port output (mode 7)	OFF		OFF		
Port input (mode 7)			ON/OFF		

Legend:

OFF: Input pull-up MOS is always off.

ON/OFF: On when PDDDR = 0 and PDPCR = 1; otherwise off.

9.10 Port E

Port E is an 8-bit I/O port that also has data bus (D7 to D0) I/O. The port E has the following registers.

- Port E data direction register (PEDDR)
- Port E data register (PEDR)
- Port E register (PORTE)
- Port E Pull-up MOS control register (PEPCR)

9.10.1 Port E Data Direction Register (PEDDR)

The individual bits of PEDDR specify input or output for the pins of port E. Since this is a write-only register, bit manipulation instructions should not be used to write to it. For details, see section 2.9.4, Accessing Registers Containing Write-Only Bits.

Bit	Bit Name	Initial Value	R/W	Description
7	PE7DDR	0	W	Modes 4 to 6 When 8-bit bus mode is selected, port E functions as an I/O port. 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 settings in PEDDR are ignored, and port E pins automatically function as data input/output pins. See section 6, Bus Controller, on 8-/16-bit bus mode. 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.
6	PE6DDR	0	W	
5	PE5DDR	0	W	
4	PE4DDR	0	W	
3	PE3DDR	0	W	
2	PE2DDR	0	W	
1	PE1DDR	0	W	
0	PE0DDR	0	W	

9.10.2 Port E Data Register (PEDR)

PEDR stores output data for the port E pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PE7DR	0	R/W	An output data for a pin is stored when the pin function is specified to a general purpose output port.
6	PE6DR	0	R/W	
5	PE5DR	0	R/W	
4	PE4DR	0	R/W	
3	PE3DR	0	R/W	
2	PE2DR	0	R/W	
1	PE1DR	0	R/W	
0	PE0DR	0	R/W	

9.10.3 Port E Register (PORTE)

PORTE shows port E pin states.

Bit	Bit Name	Initial Value	R/W	Description
7	PE7	—*	R	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.
6	PE6	—*	R	
5	PE5	—*	R	
4	PE4	—*	R	
3	PE3	—*	R	
2	PE2	—*	R	
1	PE1	—*	R	
0	PE0	—*	R	

Note: * Determined by the states of pins PE7 to PE0.

9.10.4 Port E Pull-Up MOS Control Register (PEPCR)

PEPCR controls on/off states of the input pull-up MOS of port E.

Bit	Bit Name	Initial Value	R/W	Description
7	PE7PCR	0	R/W	When the pin is in the input state, the input pull-up MOS of the input pin is on when the corresponding bit is set to 1.
6	PE6PCR	0	R/W	
5	PE5PCR	0	R/W	
4	PE4PCR	0	R/W	
3	PE3PCR	0	R/W	
2	PE2PCR	0	R/W	
1	PE1PCR	0	R/W	
0	PE1PCR	0	R/W	

9.10.5 Pin Function

Port E pins also functions as data bus (D7 to D0) I/O. The correspondence between the register specification and the pin function in show below.

Table 9.54 PE7 Pin Function

Operating Mode	Modes 4 to 6			Mode 7	
Bus Mode	8-bit bus mode		16-bit bus mode	—	
PE7DDR	0	1	—	0	1
Pin Function	PE7 input	PE7 output	D7 input/output	PE7 input	PE7 output

Table 9.55 PE6 Pin Function

Operating Mode	Modes 4 to 6			Mode 7	
Bus Mode	8-bit bus mode		16-bit bus mode	—	
PE6DDR	0	1	—	0	1
Pin Function	PE6 input	PE6 output	D6 input/output	PE6 input	PE6 output

Table 9.56 PE5 Pin Function

Operating Mode	Modes 4 to 6			Mode 7	
Bus Mode	8-bit bus mode		16-bit bus mode	—	
PE5DDR	0	1	—	0	1
Pin Function	PE5 input	PE5 output	D5 input/output	PE5 input	PE5 output

Table 9.57 PE4 Pin Function

Operating Mode	Modes 4 to 6			Mode 7	
Bus Mode	8-bit bus mode		16-bit bus mode	—	
PE4DDR	0	1	—	0	1
Pin Function	PE4 input	PE4 output	D4 input/output	PE4 input	PE4 output

Table 9.58 PE3 Pin Function

Operating Mode	Modes 4 to 6			Mode 7	
Bus Mode	8-bit bus mode		16-bit bus mode	—	
PE3DDR	0	1	—	0	1
Pin Function	PE3 input	PE3 output	D3 input/output	PE3 input	PE3 output

Table 9.59 PE2 Pin Function

Operating Mode	Modes 4 to 6			Mode 7	
Bus Mode	8-bit bus mode		16-bit bus mode	—	
PE2DDR	0	1	—	0	1
Pin Function	PE2 input	PE2 output	D2 input/output	PE2 input	PE2 output

Table 9.60 PE1 Pin Function

Operating Mode	Modes 4 to 6			Mode 7	
Bus Mode	8-bit bus mode		16-bit bus mode	—	
PE1DDR	0	1	—	0	1
Pin Function	PE1 input	PE1 output	D1 input/output	PE1 input	PE1 output

Table 9.61 PE0 Pin Function

Operating Mode	Modes 4 to 6			Mode 7	
Bus Mode	8-bit bus mode		16-bit bus mode	—	
PE0DDR	0	1	—	0	1
Pin Function	PE0 input	PE0 output	D0 input/output	PE0 input	PE0 output

9.10.6 Port E Input Pull-Up MOS State

Port E has a built-in input pull-up MOS function that can be controlled by software. Input pull-up MOS can be used in 8-bit bus mode in modes 4 to 6 or in mode 7, and can be specified as on or off for individual bits.

Table 9.62 summarizes the input pull-up MOS states.

Table 9.62 Input Pull-Up MOS States (Port E)

Pins	Power-On Reset	Hardware Standby Mode	Manual Reset	Software Standby Mode	In Other Operations
Data output (16-bit bus mode in modes 4 to 6), port output (8-bit bus mode in modes 4 to 6 or mode 7)	OFF		OFF		
Port input (8-bit bus mode in modes 4 to 6 or mode 7)			ON/OFF		

Legend:

OFF: Input pull-up MOS is always off.

ON/OFF: On when PEDDR = 0 and PEPCR = 1; otherwise off.

9.11 Port F

Port F is an 8-bit I/O port that also has external interrupt input ($\overline{IRQ2}$, $\overline{IRQ3}$), bus control sign I/O, system clock output. The port F has the following registers.

- Port F data direction register (PFDDR)
- Port F data register (PFDR)
- Port F register (PORTF)

9.11.1 Port F Data Direction Register (PFDDR)

The individual bits of PFDDR specify input or output for the pins of port F. Since this is a write-only register, bit manipulation instructions should not be used to write to it. For details, see section 2.9.4, Accessing Registers Containing Write-Only Bits.

Bit	Bit Name	Initial Value	R/W	Description
7	PF7DDR	1/0*	W	Modes 4 to 6 Pin PF7 functions as the φ output pin when the corresponding PFDDR bit is set to 1, and as an input port when the bit is cleared to 0. The input/output direction specification in PFDDR is ignored for pins PF6 to PF3, which are automatically designated as bus control outputs. Pins PF2 to PF0 are made bus control input/output pins by bus controller settings. Otherwise, setting a PFDDR bit to 1 makes the corresponding pin an output port, while clearing the bit to 0 makes the pin an input port. Mode 7 Setting a PFDDR bit to 1 makes the corresponding port F pin PF6 to PF0 an output port, or in the case of pin PF7, the φ output pin. Clearing the bit to 0 makes the pin an input port.
6	PF6DDR	0	W	
5	PF5DDR	0	W	
4	PF4DDR	0	W	
3	PF3DDR	0	W	
2	PF2DDR	0	W	
1	PF1DDR	0	W	
0	PF0DDR	0	W	

Note: * In modes 4 to 6, set to 1; in mode 7 cleared to 0.

9.11.2 Port F Data Register (PFDR)

PFDR stores output data for the port F pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PF7DR	0	R/W	An output data for a pin is stored when the pin function is specified to a general purpose output port.
6	PF6DR	0	R/W	
5	PF5DR	0	R/W	
4	PF4DR	0	R/W	
3	PF3DR	0	R/W	
2	PF2DR	0	R/W	
1	PF1DR	0	R/W	
0	PF0DR	0	R/W	

9.11.3 Port F Register (PORTF)

PORTF shows port F pin states.

Bit	Bit Name	Initial Value	R/W	Description
7	PF7	—*	R	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.
6	PF6	—*	R	
5	PF5	—*	R	
4	PF4	—*	R	
3	PF3	—*	R	
2	PF2	—*	R	
1	PF1	—*	R	
0	PF0	—*	R	

Note: * Determined by the states of pins PF7 to PF0.

9.11.4 Pin Functions

Port F is an 8-bit I/O port. Port F pins also function as external interrupt input ($\overline{IRQ2}$ and $\overline{IRQ3}$), bus control signal, and system clock output (ϕ).

Table 9.63 PF7 Pin Function

PF7DDR	0	1
Pin function	PF7 input	ϕ output

Table 9.64 PF6 Pin Function

Operating Mode	Modes 4 to 6	Mode 7	
PF6DDR	—	0	1
Pin function	$\overline{AS}$ output	PF6 input	PF6 output

Table 9.65 PF5 Pin Function

Operating Mode	Modes 4 to 6	Mode 7	
PF5DDR	—	0	1
Pin function	$\overline{RD}$ output	PF5 input	PF5 output

Table 9.66 PF4 Pin Function

Operating Mode	Modes 4 to 6	Mode 7	
PF4DDR	—	0	1
Pin function	$\overline{HWR}$ output	PF4 input	PF4 output

Table 9.67 PF3 Pin Function

Operating Mode	Modes 4 to 6			Mode 7	
Bus Mode	16 bits	8 bits		—	
PF3DDR	—	0	1	0	1
Pin function	$\overline{\text{LWR}}$ output	PF3 input	PF3 output	PF3 input	PF3 output
		$\overline{\text{ADTRG}}$ input *1			
		$\overline{\text{IRQ3}}$ input *2			

Notes: 1. $\overline{\text{ADTRG}}$ input when TRGS0=TRGS1=1.

2. When used as an external interrupt input pin, do not use as an I/O pin for another function.

Table 9.68 PF2 Pin Function

Operating Mode	Modes 4 to 6			Mode 7	
WAITE	0		1	—	
PF2DDR	0	1	—	0	1
Pin function	PF2 input	PF2 output	$\overline{\text{WAIT}}$ input	PF2 input	PF2 output

Table 9.69 PF1 Pin Function

Operating Mode	Modes 4 to 6			Mode 7	
BRLE	0		1	—	
PF1DDR	0	1	—	0	1
Pin function	PF1 input	PF1 output	$\overline{\text{BACK}}$ output	PF1 input	PF1 output

Table 9.70 PF0 Pin Function

Operating Mode	Modes 4 to 6			Mode 7	
BRLE	0		1	—	
PF0DDR	0	1	—	0	1
Pin function	PF0 input	PF0 output	$\overline{\text{BREQ}}$ input	PF0 input	PF0 output
	$\overline{\text{IRQ2}}$ input*				

Note: * When used as an external interrupt input pin, do not use as an I/O pin for another function.

9.12 Port G

Port G is a 5-bit I/O port that also has functioning as external interrupt input ($\overline{\text{IRQ7}}$) and bus control output ($\overline{\text{CS0}}$ to $\overline{\text{CS3}}$). The port G has the following registers.

- Port G data direction register (PGDDR)
- Port G data register (PGDR)
- Port G register (PORTG)

9.12.1 Port G Data Direction Register (PGDDR)

The individual bits of PGDDR specify input or output for the pins of port G. If port G is, an undefined value will be read. Since this is a write-only register, bit manipulation instructions should not be used to write to it. For details, see section 2.9.4, Accessing Registers Containing Write-Only Bits.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	Undefined	—	Reserved These bits are undefined and cannot be modified.
4	PG4DDR	0/1*	W	Modes 4 to 6 Setting a PGDDR bit to 1 makes the PG4 to PG1 pins bus control signal outputs, while clearing the bit to 0 makes the pin input ports. Signal outputs, while clearing the bit to 0 makes the pin input ports. Setting a PGDDR bit to 1 makes the PG0 pin an output port, while clearing the bit to 0 makes the pin an input port. Mode 7 Setting a PGDDR bit to 1 makes the corresponding port G pin an output port, while clearing the bit to 0 makes the pin an input port.
3	PG3DDR	0	W	
2	PG2DDR	0	W	
1	PG1DDR	0	W	
0	PG0DDR	0	W	

Note: * In modes 4 and 5, set to 1; in modes 6 and 7 cleared to 0.

9.12.2 Port G Data Register (PGDR)

PGDR stores output data for the port G pins.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	Undefined	—	Reserved These bits are undefined and cannot be modified.
4	PG4DR	0	R/W	An output data for a pin is stored when the pin function is specified to a general purpose output port.
3	PG3DR	0	R/W	
2	PG2DR	0	R/W	
1	PG1DR	0	R/W	
0	PG0DR	0	R/W	

9.12.3 Port G Register (PORTG)

PORTG shows port G pin states.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	Undefined	—	Reserved These bits are undefined and cannot be modified.
4	PG4	—*	R	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.
3	PG3	—*	R	
2	PG2	—*	R	
1	PG1	—*	R	
0	PG0	—*	R	

Note: * Determined by the states of pins PG4 to PG0.

9.12.4 Pin Functions

Port G is an 8-bit I/O port. Port G pins also function as external interrupt inputs ($\overline{\text{IRQ7}}$) and bus control signals ($\overline{\text{CS0}}$ to $\overline{\text{CS3}}$).

Table 9.71 PG4 Pin Function

Operating Mode	Modes 4 to 6		Mode 7	
PG4DDR	0	1	0	1
Pin function	PG4 input	$\overline{\text{CS0}}$ output	PG4 input	PG4 output

Table 9.72 PG3 Pin Function

Operating Mode	Modes 4 to 6		Mode 7	
PG3DDR	0	1	0	1
Pin function	PG3 input	$\overline{\text{CS1}}$ output	PG3 input	PG3 output

Table 9.73 PG2 Pin Function

Operating Mode	Modes 4 to 6		Mode 7	
PG2DDR	0	1	0	1
Pin function	PG2 input	$\overline{\text{CS2}}$ output	PG2 input	PG2 output

Table 9.74 PG1 Pin Function

Operating Mode	Modes 4 to 6		Mode 7	
PG1DDR	0	1	0	1
Pin function	PG1 input	$\overline{\text{CS3}}$ output	PG1 input	PG1output
	$\overline{\text{IRQ7}}$ input*			

Note: * When used as an external interrupt input pin, do not use as an I/O pin for another function.

Table 9.75 PG0 Pin Function

PG0DDR	0	1
Pin function	PG0 input	PG0 output

9.13 Handling of Unused Pins

Unused input pins should be fixed high or low. Generally, the input pins of CMOS products are high-impedance. Leaving unused pins open can cause the generation of intermediate levels due to peripheral noise induction. This can result in shoot-through current inside the device and cause it to malfunction. Table 9.76 lists examples of ways to handle unused pins.

For the handling of dedicated boundary scan pins that are unused, see 14.2, Pin Configuration, and 14.5, Usage Notes. For the handling of dedicated USB pins that are unused, see 15.9.14, Pin Processing when USB Not Used.

Table 9.76 Examples of Ways to Handle Unused Input Pins

Pin Name	Pin Handling Example
Port 1	Connect each pin to Vcc (pull-up) or to Vss (pull-down) via a resistor.
Port 3	
Port 4	Connect each pin to AVcc (pull-up) or to AVss (pull-down) via a resistor.
Port 7	Connect each pin to Vcc (pull-up) or to Vss (pull-down) via a resistor.
Port 9	Connect each pin to AVcc (pull-up) or to AVss (pull-down) via a resistor.
Port A	Connect each pin to Vcc (pull-up) or to Vss (pull-down) via a resistor.
Port B	
Port C	
Port D	
Port E	
Port F	
Port G	

Section 10 16-Bit Timer Pulse Unit (TPU)

This LSI has an on-chip 16-bit timer pulse unit (TPU) that comprises three 16-bit timer channels. The function list of the 16-bit timer unit and its block diagram are shown in table 10.1 and figure 10.1, respectively.

10.1 Features

- Maximum 8-pulse input/output
- Selection of 8 counter input clocks for each channel
- The following operations can be set for each channel

  Waveform output at compare match, input capture function, counter clear operation, simultaneous writing to multiple timer counters (TCNT), simultaneous clearing using compare match or input capture, simultaneous input/output for individual registers using counter synchronous operation, PWM output using user-defined duty, up to 7-phase PWM output by combination with synchronous operation
- Buffer operation settable for channel 0
- Phase counting mode settable independently for each of channels 1 and 2
- Fast access via internal 16-bit bus
- 13 interrupt sources
- Automatic transfer of register data
- A/D converter conversion start trigger can be generated
- Module stop mode can be set
- Baud rate clock for the SCI0 can be generated by channels 1 and 2

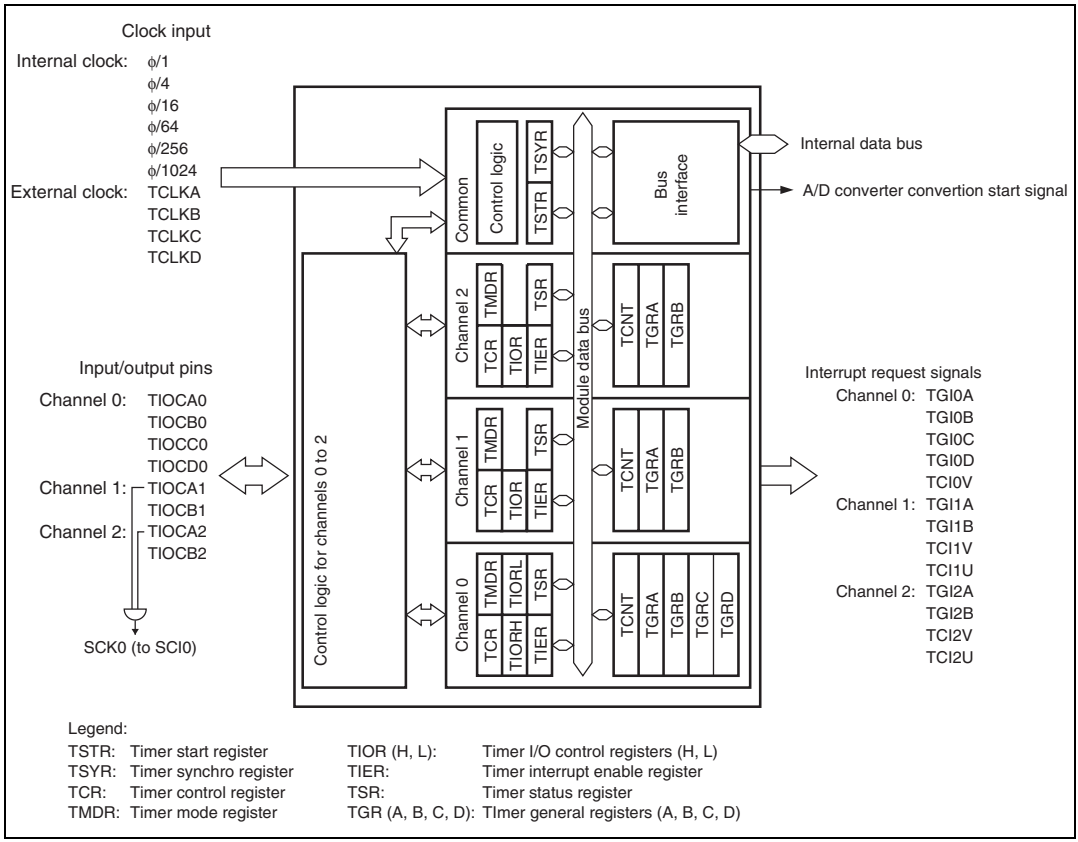


Figure 10.1 Block Diagram of TPU

Table 10.1 TPU Functions

Item		Channel 0	Channel 1	Channel 2
Count clock		φ/1	φ/1	φ/1
		φ/4	φ/4	φ/4
		φ/16	φ/16	φ/16
		φ/64	φ/64	φ/64
		TCLKA	φ/256	φ/1024
		TCLKB	TCLKA	TCLKA
		TCLKC	TCLKB	TCLKB
		TCLKD		TCLKC
General registers		TGRA_0	TGRA_1	TGRA_2
		TGRB_0	TGRB_1	TGRB_2
General registers/buffer registers		TGRC_0	not possible	not possible
		TGRD_0		
I/O pins		TIOCA0	TIOCA1	TIOCA2
		TIOCB0	TIOCB1	TIOCB2
		TIOCC0		
		TIOCD0		
Counter clear function		TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture
Compare match output	0 output	possible	possible	possible
	1 output	possible	possible	possible
	Toggle output	possible	possible	possible
Input capture function		possible	possible	possible
Synchronous operation		possible	possible	possible
PWM mode		possible	possible	possible
Phase counting mode		not possible	possible	possible
Buffer operation		possible	not possible	not possible

Item	Channel 0	Channel 1	Channel 2
DTC activation	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture
DMAC activation	TGRA_0 compare match or input capture	TGRA_1 compare match or input capture	TGRA_2 compare match or input capture
A/D converter trigger	TGRA_0 compare match or input capture	TGRA_1 compare match or input capture	TGRA_2 compare match or input capture
Interrupt sources	5 sources • Compare match or input capture 0A • Compare match or input capture 0B • Compare match or input capture 0C • Compare match or input capture 0D • Overflow	4 sources • Compare match or input capture 1A • Compare match or input capture 1B • Overflow • Underflow	4 sources • Compare match or input capture 2A • Compare match or input capture 2B • Overflow • Underflow

10.2 Input/Output Pins

Table 10.2 Pin Configuration

Channel	Symbol	I/O	Function
All	TCLKA	Input	External clock A input pin (Channel 1 phase counting mode A phase input)
	TCLKB	Input	External clock B input pin (Channel 1 phase counting mode B phase input)
	TCLKC	Input	External clock C input pin (Channel 2 phase counting mode A phase input)
	TCLKD	Input	External clock D input pin (Channel 2 phase counting mode B phase input)
0	TIOCA0	I/O	TGRA_0 input capture input/output compare output/PWM output pin
	TIOCB0	I/O	TGRB_0 input capture input/output compare output/PWM output pin
	TIOCC0	I/O	TGRC_0 input capture input/output compare output/PWM output pin
	TIOCD0	I/O	TGRD_0 input capture input/output compare output/PWM output pin
1	TIOCA1	I/O	TGRA_1 input capture input/output compare output/PWM output pin
	TIOCB1	I/O	TGRB_1 input capture input/output compare output/PWM output pin
2	TIOCA2	I/O	TGRA_2 input capture input/output compare output/PWM output pin
	TIOCB2	I/O	TGRB_2 input capture input/output compare output/PWM output pin

10.3 Register Descriptions

The TPU has the following registers.

- Timer control register_0 (TCR_0)
- Timer mode register_0 (TMDR_0)
- Timer I/O control register H_0 (TIORH_0)
- Timer I/O control register L_0 (TIORL_0)
- Timer interrupt enable register_0 (TIER_0)
- Timer status register_0 (TSR_0)
- Timer counter_0 (TCNT_0)
- Timer general register A_0 (TGRA_0)
- Timer general register B_0 (TGRB_0)
- Timer general register C_0 (TGRC_0)
- Timer general register D_0 (TGRD_0)
- Timer control register_1 (TCR_1)
- Timer mode register_1 (TMDR_1)
- Timer I/O control register _1 (TIOR_1)
- Timer interrupt enable register_1 (TIER_1)
- Timer status register_1 (TSR_1)
- Timer counter_1 (TCNT_1)
- Timer general register A_1 (TGRA_1)
- Timer general register B_1 (TGRB_1)
- Timer control register_2 (TCR_2)
- Timer mode register_2 (TMDR_2)
- Timer I/O control register_2 (TIOR_2)
- Timer interrupt enable register_2 (TIER_2)
- Timer status register_2 (TSR_2)
- Timer counter_2 (TCNT_2)
- Timer general register A_2 (TGRA_2)
- Timer general register B_2 (TGRB_2)

Common Registers

- Timer start register (TSTR)
- Timer synchro register (TSYR)

10.3.1 Timer Control Register (TCR)

The TCR registers control the TCNT operation for each channel. The TPU has a total of three TCR registers, one for each channel (channels 0 to 2). TCR register settings should be made only when TCNT operation is stopped.

Bit	Bit Name	Initial value	R/W	Description
7	CCLR2	0	R/W	Counter Clear 2 to 0
6	CCLR1	0	R/W	These bits select the TCNTcounter clearing source. See tables 10.3 and 10.4 for details.
5	CCLR0	0	R/W	
4	CKEG1	0	R/W	Clock Edge 1 and 0
3	CKEG0	0	R/W	These bits select the input clock edge. When the internal clock is counted using both edges, the input clock frequency is halved (e.g., ϕ/4 both edges = ϕ/2 rising edge). If phase counting mode is used on channels 1, 2, 4, and 5, this setting is ignored and the phase counting mode setting has priority. Internal clock edge selection is valid when the input clock is ϕ/4 or slower. If ϕ/1 is selected as the input clock, this setting is ignored and count at falling edge of ϕ is selected. 00: Count at rising edge 01: Count at falling edge 1×: Count at both edges Legend: ×: Don't care
2	TPSC2	0	R/W	Time Prescaler 2 to 0
1	TPSC1	0	R/W	These bits select the TCNT counter clock. The clock source can be selected independently for each channel. See tables 10.5 to 10.10 for details.
0	TPSC0	0	R/W	

Table 10.3 CCLR2 to CCLR0 (channel 0)

Channel	Bit 7 CCLR2	Bit 6 CCLR1	Bit 5 CCLR0	Description
0	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.

Table 10.4 CCLR2 to CCLR0 (channels 1 and 2)

Channel	Bit 7 Reserved*2	Bit 6 CCLR1	Bit 5 CCLR0	Description
1, 2	0	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRA compare match/input capture
		1	0	TCNT cleared by TGRB compare match/input capture
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/synchronous operation*1

Notes: 1. Synchronous operation setting is performed by setting the SYNC bit in TSYR to 1.
2. Bit 7 is reserved in channels 1 and 2. It is always read as 0 and cannot be modified.

Table 10.5 TPSC2 to TPSC0 (channel 0)

	Bit 2	Bit 1	Bit 0	
Channel	**TPSC2**	**TPSC1**	**TPSC0**	**Description**
0	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

Table 10.6 TPSC2 to TPSC0 (channel 1)

	Bit 2	Bit 1	Bit 0	
Channel	**TPSC2**	**TPSC1**	**TPSC0**	**Description**
1	0	0	0	Internal clock: counts on ϕ/1
			1	Internal clock: counts on ϕ/4
		1	0	Internal clock: counts on ϕ/16
			1	Internal clock: counts on ϕ/64
	1	0	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKB pin input
		1	0	Internal clock: counts on ϕ/256
			1	Setting prohibited

Note: This setting is ignored when channel 1 is in phase counting mode.

Table 10.7 TPSC2 to TPSC0 (channel 2)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
2	0	0	0	Internal clock: counts on ϕ/1
			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 1 is in phase counting mode.

10.3.2 Timer Mode Register (TMDR)

The TMDR registers are used to set the operating mode for each channel. The TPU has three TMDR registers, one for each channel. TMDR register settings should be made only when TCNT operation is stopped.

Bit	Bit Name	Initial value	R/W	Description
7, 6	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
5	BFB	0	R/W	Buffer Operation B Specifies whether TGRB is to operate in the normal way, or TGRB and TGRD are to be used together for buffer operation. When TGRD is used as a buffer register. TGRD input capture/output compare is not generation. In channels 1 and 2, which have no TGRD, bit 5 is reserved. It is always read as 0 and cannot be modified. 0: TGRB operates normally 1: TGRB and TGRD used together for buffer operation
4	BFA	0	R/W	Buffer Operation A Specifies whether TGRA is to operate in the normal way, or TGRA and TGRC are to be used together for buffer operation. When TGRC is used as a buffer register, TGRC input capture/output compare is not generated. In channels 1 and 2, which have no TGRC, bit 4 is reserved. It is always read as 0 and cannot be modified. 0: TGRA operates normally 1: TGRA and TGRC used together for buffer operation
3	MD3	0	R/W	Modes 3 to 0
2	MD2	0	R/W	These bits are used to set the timer operating mode.
1	MD1	0	R/W	MD3 is a reserved bit. In a write, the write value should always be 0. See table 10.8, for details.
0	MD0	0	R/W	

Table 10.8 MD3 to MD0

Bit 3	Bit2	Bit 1	Bit 0	
MD3[*1]	**MD2**[*2]	**MD1**	**MD0**	**Description**
0	0	0	0	Normal operation
			1	Reserved
		1	0	PWM mode 1
			1	PWM mode 2
	1	0	0	Phase counting mode 1
			1	Phase counting mode 2
		1	0	Phase counting mode 3
			1	Phase counting mode 4
1	×	×	×	—

Legend:

×: Don't care

Notes: 1. MD3 is reserved bit. In a write, it should be written with 0.

2. Phase counting mode cannot be set for channels 0 and 3. In this case, 0 should always be written to MD2.

10.3.3 Timer I/O Control Register (TIOR)

The TIOR registers control the TGR registers. The TPU has eight TIOR registers, two each for channel 0, and one each for channels 1 and 2. Care is required since TIOR is affected by the TMDR setting. The initial output specified by TIOR is valid when the counter is stopped (the CST bit in TSTR is cleared to 0). Note also that, in PWM mode 2, the output at the point at which the counter is cleared to 0 is specified. When TGRC or TGRD is designated for buffer operation, this setting is invalid and the register operates as a buffer register.

- **TIORH_0, TIOR_1, TIOR_2**

Bit	Bit Name	Initial value	R/W	Description
7	IOB3	0	R/W	I/O Control B3 to B0
6	IOB2	0	R/W	Specify the function of TGRB.
5	IOB1	0	R/W	
4	IOB0	0	R/W	
3	IOA3	0	R/W	I/O Control A3 to A0
2	IOA2	0	R/W	Specify the function of TGRA.
1	IOA1	0	R/W	
0	IOA0	0	R/W	

- **TIORL_0**

Bit	Bit Name	Initial value	R/W	Description
7	IOD3	0	R/W	I/O Control D3 to D0
6	IOD2	0	R/W	Specify the function of TGRD.
5	IOD1	0	R/W	
4	IOD0	0	R/W	
3	IOC3	0	R/W	I/O Control C3 to C0
2	IOC2	0	R/W	Specify the function of TGRC.
1	IOC1	0	R/W	
0	IOC0	0	R/W	

Table 10.9 TIORH_0 (channel 0)

				Description	
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**TGRB_0**	
IOB3	**IOB2**	**IOB1**	**IOB0**	**Function**	**TIOCB0 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCB0 pin Input capture at rising edge
			1		Capture input source is TIOCB0 pin Input capture at falling edge
		1	×		Capture input source is TIOCB0 pin Input capture at both edges
	1	×	×		Setting prohibited

Legend:
×: Don't care

Table 10.10 TIORH_0 (channel 0)

				Description	
Bit 3	**Bit 2**	**Bit 1**	**Bit 0**	**TGRA_0**	
IOA3	**IOA2**	**IOA1**	**IOA0**	**Function**	**TIOCA0 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA0 pin Input capture at rising edge
			1		Capture input source is TIOCA0 pin Input capture at falling edge
		1	×		Capture input source is TIOCA0 pin Input capture at both edges
	1	×	×		Setting prohibited

Legend:
×: Don't care

Table 10.11 TIORL_0 (channel 0)

				Description	
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**TGRD_0**	
IOD3	**IOD2**	**IOD1**	**IOD0**	**Function**	**TIOCD0 Pin Function**
0	0	0	0	Output Compare register*	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output
			1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register*	Capture input source is TIOCD0 pin Input capture at rising edge
			1		Capture input source is TIOCD0 pin Input capture at falling edge
		1	×		Capture input source is TIOCD0 pin Input capture at both edges
	1	×	×		Setting prohibited

Legend:

×: Don't care

Note: * When the BFB bit in TMDR_0 is set to 1 and TGRD_0 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 10.12 TIORL_0 (channel 0)

				Description	
Bit 3	**Bit 2**	**Bit 1**	**Bit 0**	**TGRC_0**	
IOC3	**IOC2**	**IOC1**	**IOC0**	**Function**	**TIOCC0 Pin Function**
0	0	0	0	Output compare register*	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register*	Capture input source is TIOCC0 pin Input capture at rising edge
			1		Capture input source is TIOCC0 pin Input capture at falling edge
		1	×		Capture input source is TIOCC0 pin Input capture at both edges
	1	×	×		Setting prohibited

Legend:

×: Don't care

Note: * When the BFA bit in TMDR_0 is set to 1and TGRC_0 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 10.13 TIOR_1 (channel 1)

				Description	
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**TGRB_1**	
IOB3	**IOB2**	**IOB1**	**IOB0**	**Function**	**TIOCB1 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCB1 pin Input capture at rising edge
			1		Capture input source is TIOCB1 pin Input capture at falling edge
		1	×		Capture input source is TIOCB1 pin Input capture at both edges
	1	×	×		Setting prohibited

Legend:

×: Don't care

Table 10.14 TIOR_1 (channel 1)

				Description	
Bit 3	Bit 2	Bit 1	Bit 0	TGRA_1	
IOA3	IOA2	IOA1	IOA0	Function	TIOCA1 Pin Function
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA1 pin Input capture at rising edge
			1		Capture input source is TIOCA1 pin Input capture at falling edge
		1	×		Capture input source is TIOCA1 pin Input capture at both edges
	1	×	×		Setting prohibited

Legend:
×: Don't care

Table 10.15 TIOR_2 (channel 2)

				Description	
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**TGRB_2**	
IOB3	**IOB2**	**IOB1**	**IOB0**	**Function**	**TIOCB2 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	×	0	0	Input capture register	Capture input source is TIOCB2 pin Input capture at rising edge
			1		Capture input source is TIOCB2 pin Input capture at falling edge
		1	×		Capture input source is TIOCB2 pin Input capture at both edges

Legend

×: Don't care

Table 10.16 TIOR_2 (channel 2)

				Description	
Bit 3	**Bit 2**	**Bit 1**	**Bit 0**	**TGRA_2**	
IOA3	**IOA2**	**IOA1**	**IOA0**	**Function**	**TIOCA2 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	×	0	0	Input capture register	Capture input source is TIOCA2 pin Input capture at rising edge
			1		Capture input source is TIOCA2 pin Input capture at falling edge
		1	×		Capture input source is TIOCA2 pin Input capture at both edges

Legend:
×: Don't care

10.3.4 Timer Interrupt Enable Register (TIER)

The TIER registers control enabling or disabling of interrupt requests for each channel. The TPU has three TIER registers, one for each channel.

Bit	Bit Name	Initial value	R/W	Description
7	TTGE	0	R/W	A/D Conversion Start Request Enable Enables or disables generation of A/D conversion start requests by TGRA input capture/compare match. 0: A/D conversion start request generation disabled 1: A/D conversion start request generation enabled
6	—	1	—	Reserved This bit is always read as 1 and cannot be modified.
5	TCIEU	0	R/W	Underflow Interrupt Enable Enables or disables interrupt requests (TCU) by the TCFU flag when the TCFU flag in TSR is set to 1 in channels 1 and 2. In channel 0, bit 5 is reserved. 0: Interrupt requests (TCIU) by TCFU disabled 1: Interrupt requests (TCIU) by TCFU enabled
4	TCIEV	0	R/W	Overflow Interrupt Enable Enables or disables interrupt requests (TCIV) by the TCFV flag when the TCFV flag in TSR is set to 1. 0: Interrupt requests (TCIV) by TCFV disabled 1: Interrupt requests (TCIV) by TCFV enabled
3	TGIED	0	R/W	TGR Interrupt Enable D Enables or disables interrupt requests (TGID) by the TGFD bit when the TGFD bit in TSR is set to 1 in channel 0. In channels 1 and 2, bit 3 is reserved. It is always read as 0 and cannot be modified. 0: Interrupt requests (TGID) by TGFD disabled 1: Interrupt requests (TGID) by TGFD enabled
2	TGIEC	0	R/W	TGR Interrupt Enable C Enables or disables interrupt requests (TGIC) by the TGFC bit when the TGFC bit in TSR is set to 1 in channel 0. In channels 1 and 2, bit 2 is reserved. It is always read as 0 and cannot be modified. 0: Interrupt requests (TGIC) by TGFC disabled 1: Interrupt requests (TGIC) by TGFC enabled

Bit	Bit Name	Initial value	R/W	Description
1	TGIEB	0	R/W	TGR Interrupt Enable B Enables or disables interrupt requests (TGIB) by the TGFB bit when the TGFB bit in TSR is set to 1. 0: Interrupt requests (TGIB) by TGFB disabled 1: Interrupt requests (TGIB) by TGFB enabled
0	TGIEA	0	R/W	TGR Interrupt Enable A Enables or disables interrupt requests (TGIA) by the TGFA bit when the TGFA bit in TSR is set to 1. 0: Interrupt requests (TGIA) by TGFA disabled 1: Interrupt requests (TGIA) by TGFA enabled

10.3.5 Timer Status Register (TSR)

The TSR registers indicate the status of each channel. The TPU has three TSR registers, one for each channel.

Bit	Bit Name	Initial value	R/W	Description
7	TCFD	1	R	Count Direction Flag Status flag that shows the direction in which TCNT counts in channels 1 and 2. In channel 0, bit 7 is reserved. It is always read as 0 and cannot be modified. 0: TCNT counts down 1: TCNT counts up
6	—	1	—	Reserved This bit is always read as 1 and cannot be modified.
5	TCFU	0	R/(W)*	Underflow Flag Status flag that indicates that TCNT underflow has occurred when channels 1 and 2 are set to phase counting mode. The write value should always be 0 to clear this flag. In channel 0, bit 5 is reserved. [Setting condition] • When the TCNT value underflows (change from H'0000 to H'FFFF) [Clearing condition] • When 0 is written to TCFU after reading TCFU = 1
4	TCFV	0	R/(W)*	Overflow Flag Status flag that indicates that TCNT overflow has occurred. The write value should always be 0 to clear this flag. [Setting condition] • When the TCNT value overflows (change from H'FFFF to H'0000) [Clearing condition] • When 0 is written to TCFV after reading TCFV = 1

Bit	Bit Name	Initial value	R/W	Description
3	TGFD	0	R/(W)*	Input Capture/Output Compare Flag D Status flag that indicates the occurrence of TGRD input capture or compare match in channel 0. The write value should always be 0 to clear this flag. In channels 1 and 2, bit 3 is reserved. It is always read as 0 and cannot be modified. [Setting conditions] • When TCNT = TGRD while TGRD is functioning as output compare register • When TCNT value is transferred to TGRD by input capture signal while TGRD is functioning as input capture register [Clearing conditions] • When DTC is activated by TGID interrupt, DISEL bit in MRB of DTC is cleared to 0, and transfer counter value is not 0 • When 0 is written to TGFD after reading TGFD = 1
2	TGFC	0	R/(W)*	Input Capture/Output Compare Flag C Status flag that indicates the occurrence of TGRC input capture or compare match in channel 0. The write value should always be 0 to clear this flag. In channels 1 and 2, bit 2 is reserved. It is always read as 0 and cannot be modified. [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 [Clearing conditions] • When DTC is activated by TGIC interrupt, DISEL bit in MRB of DTC is cleared to 0, and transfer counter value is not 0 • When 0 is written to TGFC after reading TGFC = 1

Bit	Bit Name	Initial value	R/W	Description
1	TGFB	0	R/(W)*	Input Capture/Output Compare Flag B Status flag that indicates the occurrence of TGRB input capture or compare match. The write value should always be 0 to clear this flag. [Setting conditions] • When TCNT = TGRB while TGRB is functioning as output compare register • When TCNT value is transferred to TGRB by input capture signal while TGRB is functioning as input capture register [Clearing conditions] • When DTC is activated by TGIB interrupt, DISEL bit in MRB of DTC is cleared to 0, and transfer counter value is not 0 • When 0 is written to TGFB after reading TGFB = 1
0	TGFA	0	R/(W)*	Input Capture/Output Compare Flag A Status flag that indicates the occurrence of TGRA input capture or compare match. The write value should always be 0 to clear this flag. [Setting conditions] • When TCNT = TGRA while TGRA is functioning as output compare register • When TCNT value is transferred to TGRA by input capture signal while TGRA is functioning as input capture register [Clearing conditions] • When DTC is activated by TGIA interrupt, DISEL bit in MRB of DTC is cleared to 0, and transfer counter value is not 0 • When 0 is written to TGFA after reading TGFA = 1

Note: * The write value should always be 0 to clear the flag.

10.3.6 Timer Counter (TCNT)

The TCNT registers are 16-bit counters. The TPU has three TCNT counters, one for each channel. The TCNT counters are initialized to H'0000 by a reset, and in hardware standby mode. The TCNT counters cannot be accessed in 8-bit units; they must always be accessed as a 16-bit unit.

10.3.7 Timer General Register (TGR)

The TGR registers are 16-bit registers with a dual function as output compare and input capture registers. The TPU has 16 TGR registers, four each for channel 0 and two each for channels 1 and 2. TGRC and TGRD for channel 0 can also be designated for operation as buffer registers. The TGR registers cannot be accessed in 8-bit units; they must always be accessed as a 16-bit unit. TGR buffer register combinations are TGRA—TGRC and TGRB—TGRD.

10.3.8 Timer Start Register (TSTR)

TSTR is an 8-bit readable/writable register that selects operation/stoppage for channels 0 to 2. When setting the operating mode in TMDR or setting the count clock in TCR, first stop the TCNT counter.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	—	Reserved The write value should always be 0.
2	CST2	0	R/W	Counter Start 2 to 0 (CST2 to CST0)
1	CST1	0	R/W	These bits select operation or stoppage for TCNT.
0	CST0	0	R/W	If 0 is written to the CST bit during operation with the TIOC pin designated for output, the counter stops but the TIOC pin output compare output level is retained. If TIOR is written to when the CST bit is cleared to 0, the pin output level will be changed to the set initial output value. 0: TCNT_2 to TCNT_0 count operation is stopped 1: TCNT_2 to TCNT_0 performs count operation

10.3.9 Timer Synchro Register (TSYR)

TSYR selects independent operation or synchronous operation for the channel 0 to 2 TCNT counters. A channel performs synchronous operation when the corresponding bit in TSYR is set to 1.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	—	Reserved The write value should always be 0.
2	SYNC2	0	R/W	Timer Synchro 2 to 0
1	SYNC1	0	R/W	These bits select whether operation is independent of or synchronized with other channels. When synchronous operation is selected, synchronous presetting of multiple channels, and synchronous clearing through counter clearing on another channel are possible. To set synchronous operation, the SYNC bits for at least two channels must be set to 1. To set synchronous clearing, in addition to the SYNC bit, the TCNT clearing source must also be set by means of bits CCLR2 to CCLR0 in TCR. 0: TCNT_2 to TCNT_0 operates independently (TCNT presetting /clearing is unrelated to other channels) 1: TCNT_2 to TCNT_0 performs synchronous operation TCNT synchronous presetting/synchronous clearing is possible
0	SYNC0	0	R/W	

10.4 Interface to Bus Master

10.4.1 16-Bit Registers

TCNT and TGR are 16-bit registers. As the data bus to the bus master is 16 bits wide, these registers can be read and written to in 16-bit units.

These registers cannot be read from or written to in 8-bit units; 16-bit access must always be used.

An example of 16-bit register access operation is shown in figure 10.2.

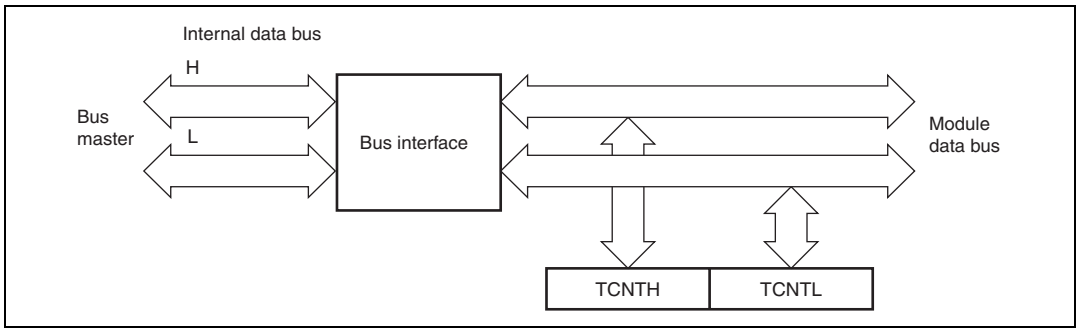


Figure 10.2 16-Bit Register Access Operation [Bus Master ↔ TCNT (16 Bits)]

10.4.2 8-Bit Registers

Registers other than TCNT and TGR are 8-bit. As the data bus to the CPU is 16 bits wide, these registers can be read and written to in 16-bit units. They can also be read and written to in 8-bit units.

Examples of 8-bit register access operation are shown in figures 10.3 to 10.5.

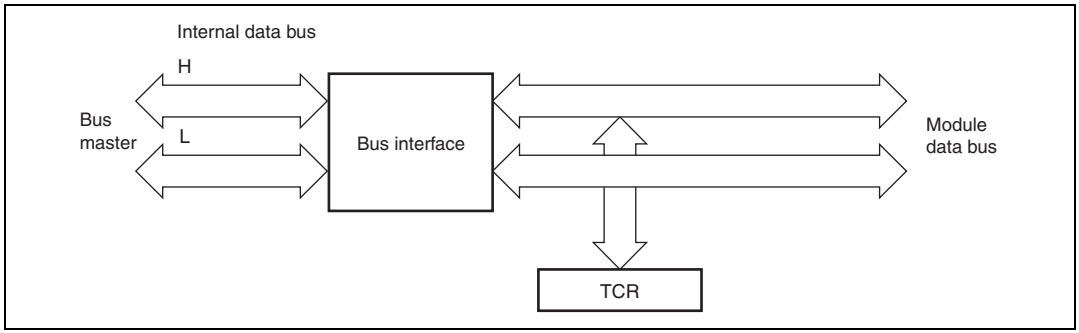


Figure 10.3 8-Bit Register Access Operation [Bus Master ↔ TCR (Upper 8 Bits)]

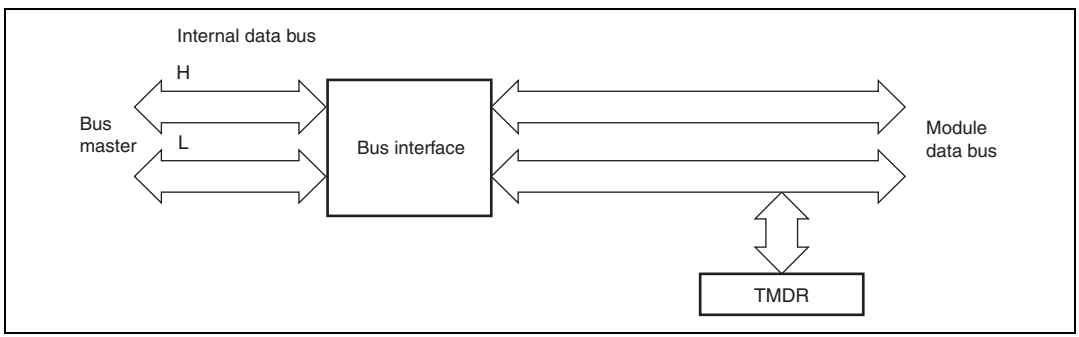


Figure 10.4 8-Bit Register Access Operation [Bus Master ↔ TMDR (Lower 8 Bits)]

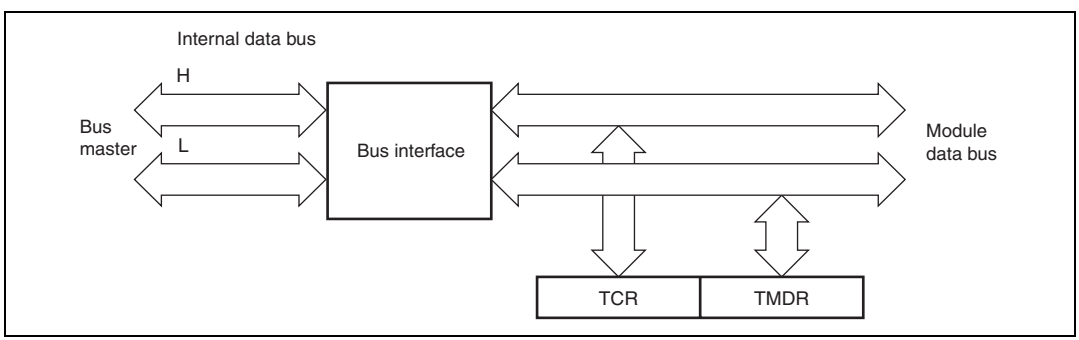


Figure 10.5 8-Bit Register Access Operation [Bus Master ↔ TCR and TMDR (16 Bits)]

10.5 Operation

10.5.1 Basic Functions

Each channel has a TCNT and TGR. TCNT performs up-counting, and is also capable of free-running operation, synchronous counting, and external event counting. Each TGR can be used as an input capture register or output compare register.

Counter Operation: When one of bits CST0 to CST2 is set to 1 in TSTR, the TCNT counter for the corresponding channel starts counting. TCNT can operate as a free-running counter, periodic counter, and so on.

1. Example of count operation setting procedure
   Figure 10.6 shows an example of the count operation setting procedure.

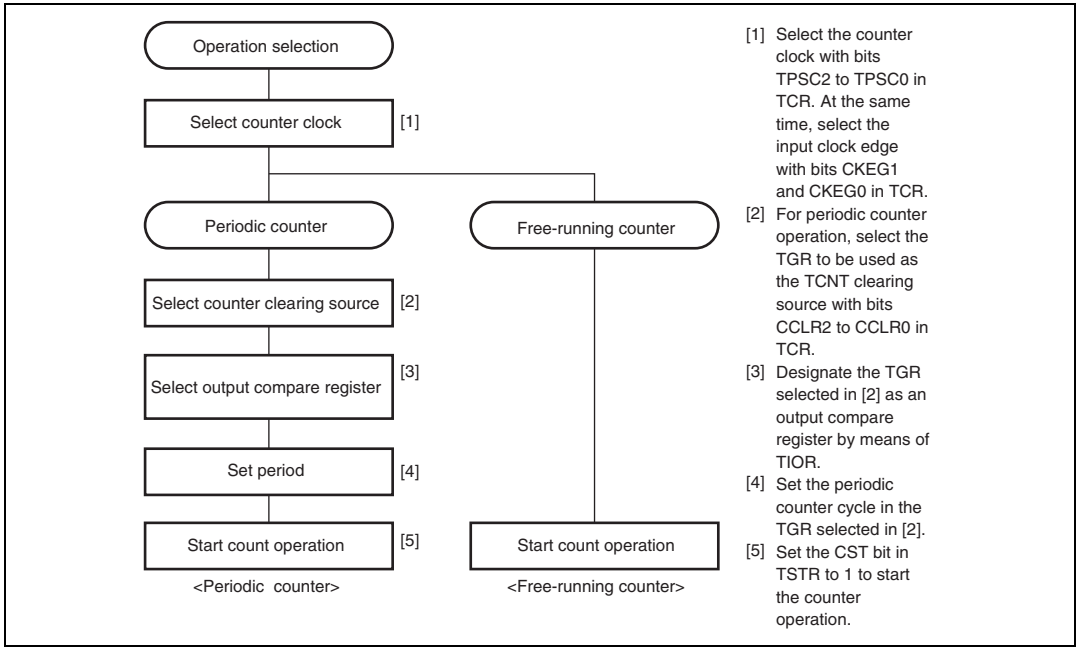


Figure 10.6 Example of Counter Operation Setting Procedure

2. Free-running count operation and periodic count operation

   Immediately after a reset, the TPU's TCNT counters are all designated as free-running counters. When the relevant bit in TSTR is set to 1 the corresponding TCNT counter starts up-count operation as a free-running counter. When TCNT overflows (from H'FFFF to H'0000), the TCFV bit in TSR is set to 1. If the value of the corresponding TCIEV bit in TIER is 1 at this point, the TPU requests an interrupt. After overflow, TCNT starts counting up again from H'0000. Figure 10.7 illustrates free-running counter operation.

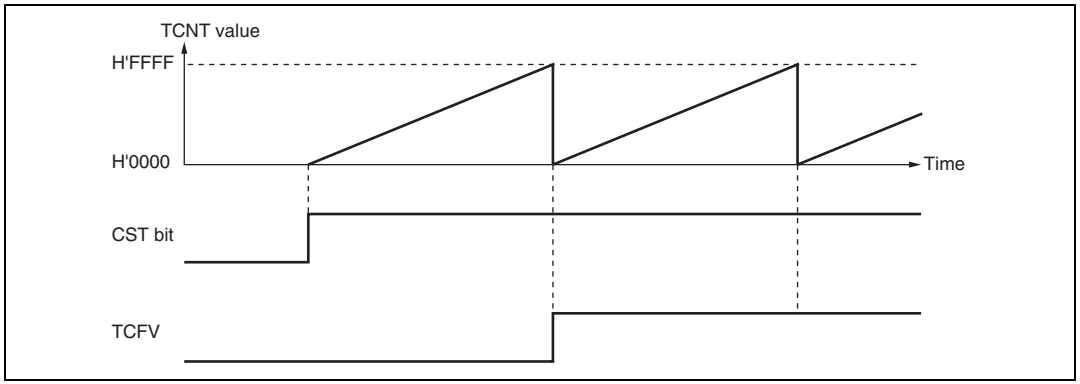


Figure 10.7 Free-Running Counter Operation

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 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.

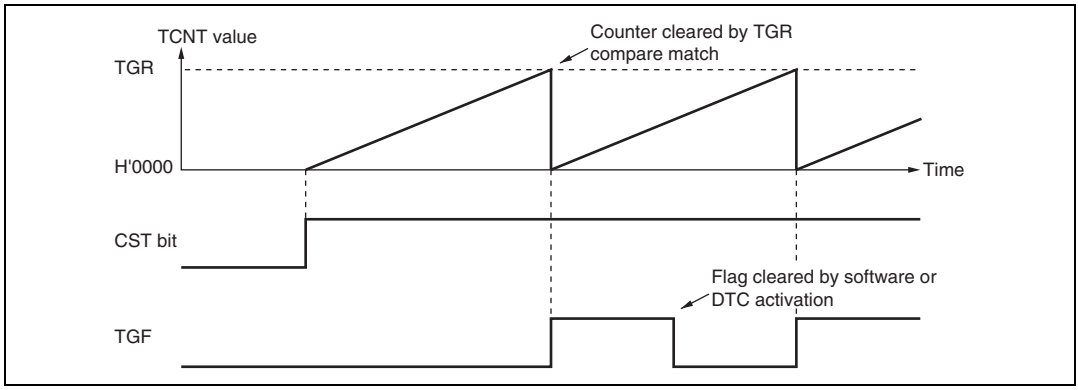


Figure 10.8 Periodic Counter Operation

Waveform Output by Compare Match: The TPU can perform 0, 1, or toggle output from the corresponding output pin using compare match.

1. 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.

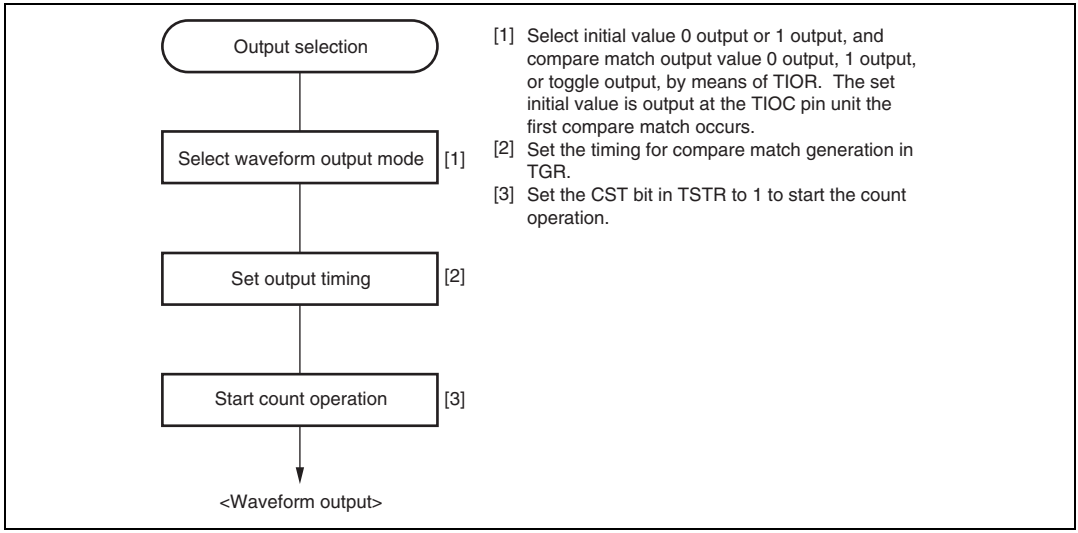


Figure 10.9 Example of Setting Procedure for Waveform Output by Compare Match

2. Examples of waveform output operation

Figure 10.10 shows an example of 0 output/1 output. In this example TCNT has been designated as a free-running counter, and settings have been made so that 1 is output by compare match A, and 0 is output by compare match B. When the set level and the pin level coincide, the pin level does not change.

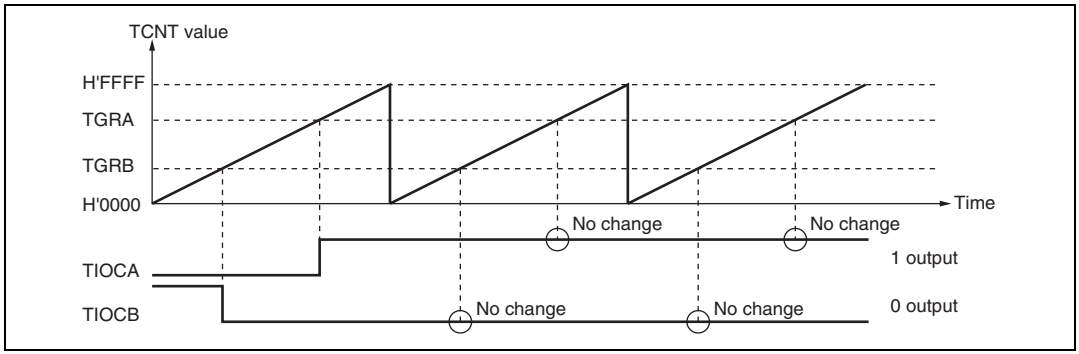


Figure 10.10 Example of 0 Output/1 Output Operation

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.

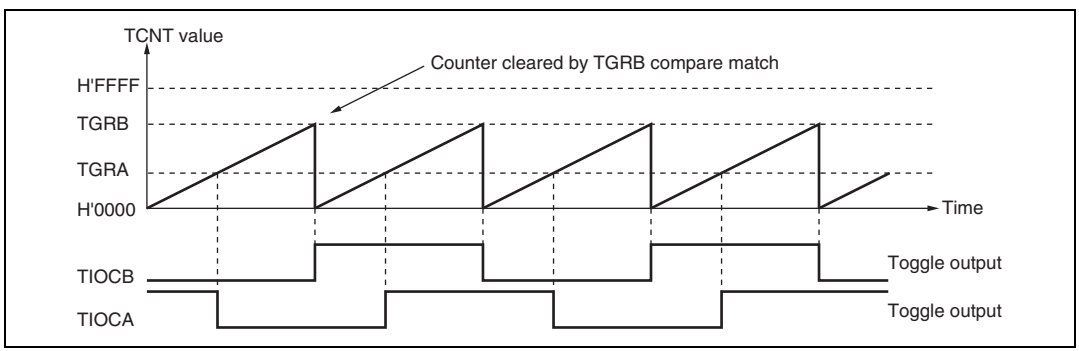


Figure 10.11 Example of Toggle Output Operation

Input Capture Function: The TCNT value can be transferred to TGR on detection of the TIOC pin input edge. Rising edge, falling edge, or both edges can be selected as the detected edge.

1. Example of input capture operation setting procedure
   Figure 10.12 shows an example of the input capture operation setting procedure.

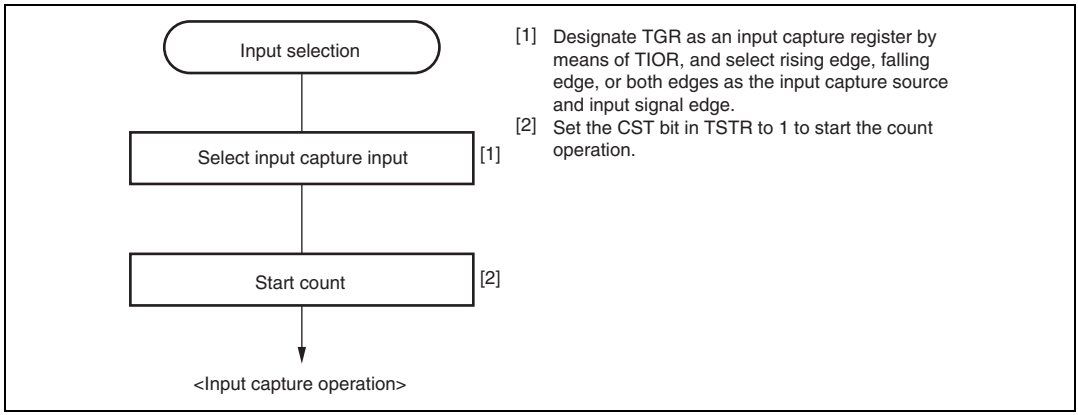


Figure 10.12 Example of Input Capture Operation Setting Procedure

2. Example of input capture operation

Figure 10.13 shows an example of input capture operation. In this example both rising and falling edges have been selected as the TIOCA pin input capture input edge, falling edge has been selected as the TIOCB pin input capture input edge, and counter clearing by TGRB input capture has been designated for TCNT.

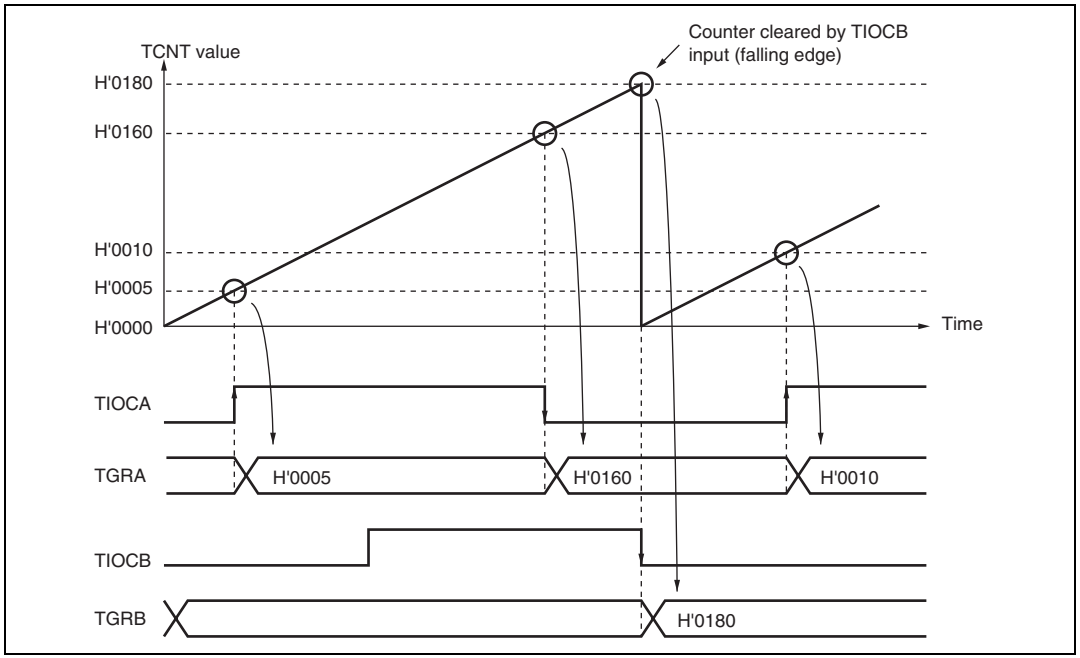
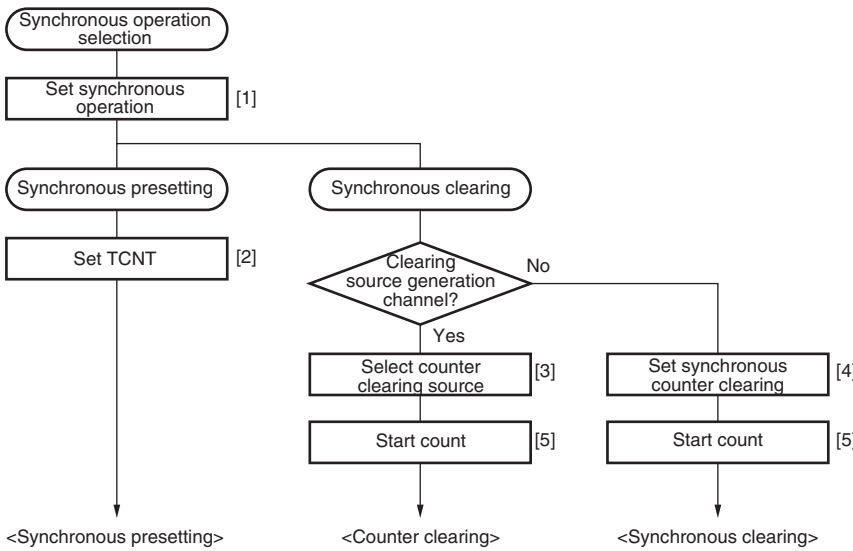


Figure 10.13 Example of Input Capture Operation

10.5.2 Synchronous Operation

In synchronous operation, the values in a number of TCNT counters can be rewritten simultaneously (synchronous presetting). Also, a number of TCNT counters can be cleared simultaneously by making the appropriate setting in TCR (synchronous clearing). Synchronous operation enables TGR to be incremented with respect to a single time base. Channels 0 to 2 can all be designated for synchronous operation.

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

Example of Synchronous Operation: Figure 10.15 shows an example of synchronous operation. In this example, synchronous operation and PWM mode 1 have been designated for channels 0 to 2, TGRB_0 compare match has been set as the channel 0 counter clearing source, and synchronous clearing has been set for the channel 1 and 2 counter clearing sources. Three-phase PWM waveforms are output from pins TIOC0A, TIOC1A, and TIOC2A. At this time, synchronous presetting, and synchronous clearing by TGRB_0 compare match, is performed for channel 0 to 2 TCNT counters, and the data set in TGRB_0 is used as the PWM cycle. For details of PWM modes, see section 10.5.4, PWM Modes.

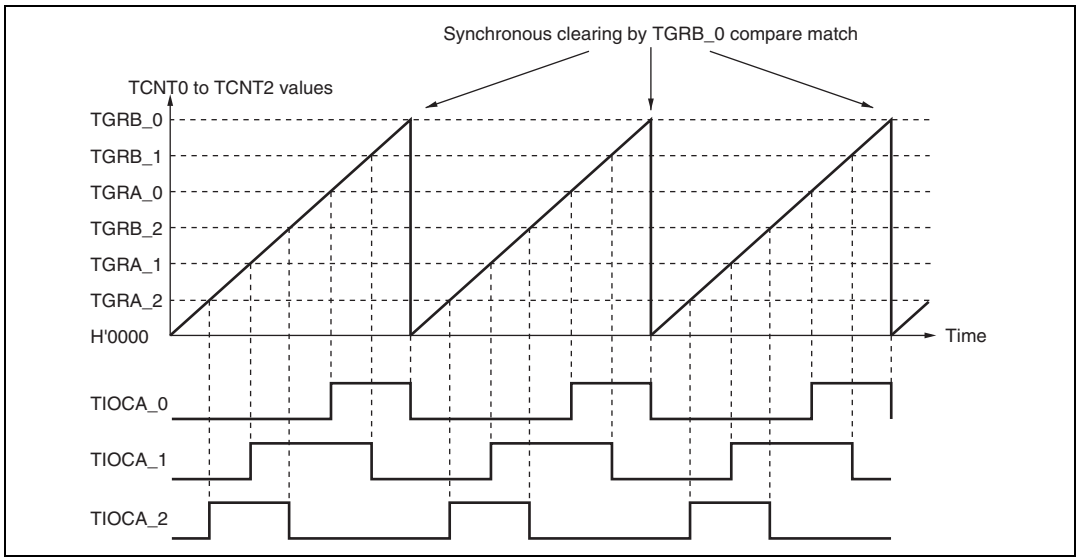


Figure 10.15 Example of Synchronous Operation

10.5.3 Buffer Operation

Buffer operation, provided for channel 0, 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 10.17 shows the register combinations used in buffer operation.

Table 10.17 Register Combinations in Buffer Operation

Channel	Timer General Register	Buffer Register
0	TGRA_0	TGRC_0
	TGRB_0	TGRD_0

- When TGR is an output compare register

  When a compare match occurs, the value in the buffer register for the corresponding channel is transferred to the timer general register. This operation is illustrated in figure 10.16.

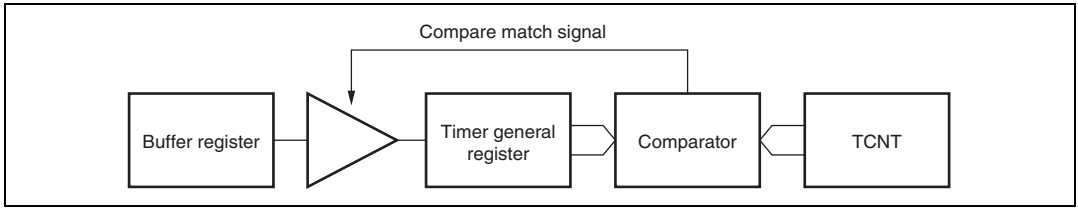


Figure 10.16 Compare Match Buffer Operation

- When TGR is an input capture register

  When input capture occurs, the value in TCNT is transferred to TGR and the value previously held in the timer general register is transferred to the buffer register. This operation is illustrated in figure 10.17.

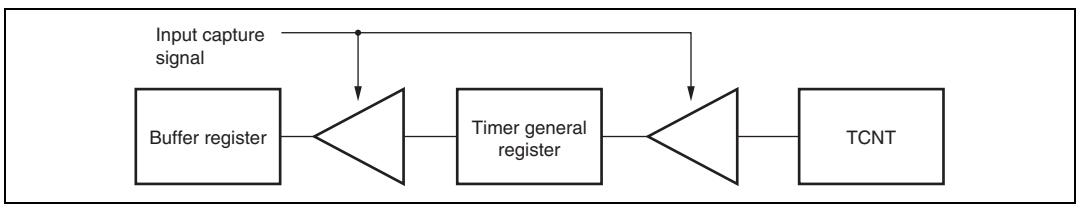


Figure 10.17 Input Capture Buffer Operation

Example of Buffer Operation Setting Procedure: Figure 10.18 shows an example of the buffer operation setting procedure.

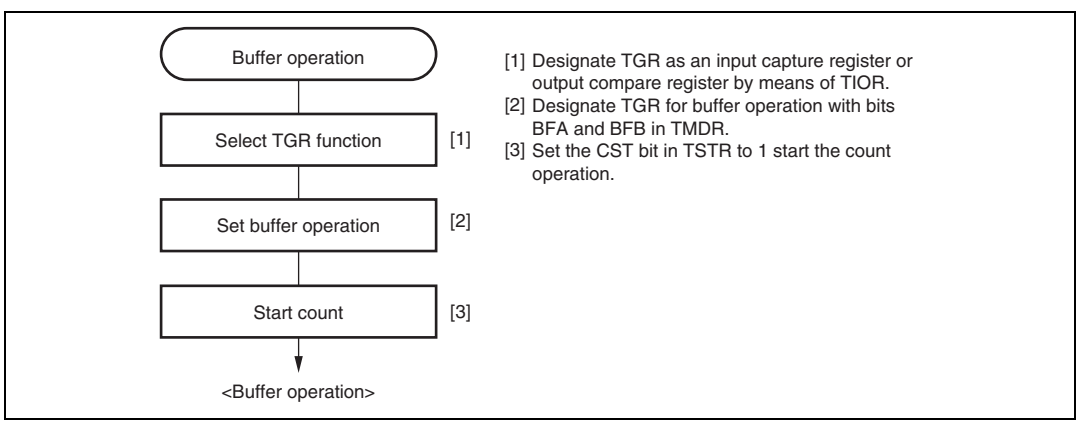


Figure 10.18 Example of Buffer Operation Setting Procedure

Examples of Buffer Operation

1. 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 10.5.4, PWM Modes.

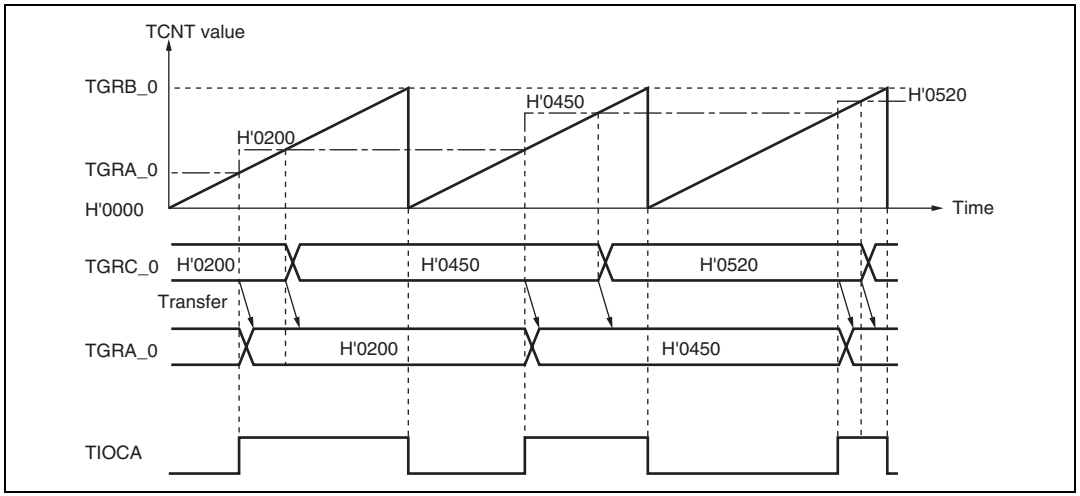


Figure 10.19 Example of Buffer Operation (1)

2. When TGR is an input capture register

   Figure 10.20 shows an operation example in which TGRA has been designated as an input capture register, and buffer operation has been designated for TGRA and TGRC. Counter clearing by TGRA input capture has been set for TCNT, and both rising and falling edges have been selected as the TIOCA pin input capture input edge. As buffer operation has been set, when the TCNT value is stored in TGRA upon occurrence of input capture A, the value previously stored in TGRA is simultaneously transferred to TGRC.

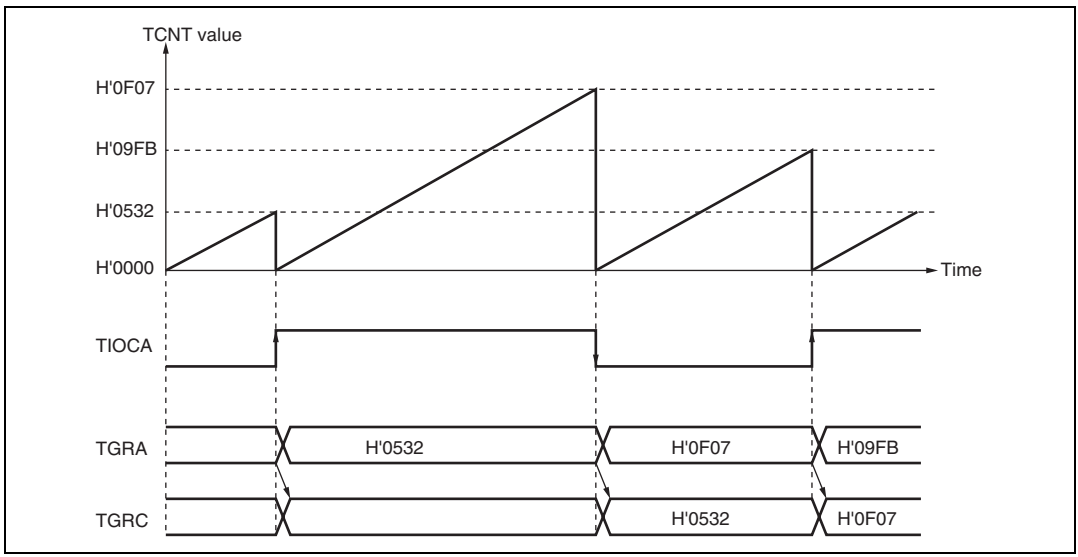


Figure 10.20 Example of Buffer Operation (2)

10.5.4 PWM Modes

In PWM mode, PWM waveforms are output from the output pins. 0, 1, or toggle output can be selected as the output level in response to compare match of each TGR. Settings of TGR registers can output a PWM waveform in the range of 0 % to 100 % duty. Designating TGR compare match as the counter clearing source enables the period to be set in that register. All channels can be designated for PWM mode independently. Synchronous operation is also possible. There are two PWM modes, as described below.

- PWM mode 1

  PWM output is generated from the TIOCA and TIOCC pins by pairing TGRA with TGRB and TGRC with TGRD. The output specified by bits IOA3 to IOA0 and IOC3 to IOC0 in TIOR is output from the TIOCA and TIOCC pins at compare matches A and C, and the output specified by bits IOB3 to IOB0 and IOD3 to IOD0 in TIOR is output at compare matches B and D. The initial output value is the value set in TGRA or TGRC. If the set values of paired TGRs are identical, the output value does not change when a compare match occurs. In PWM mode 1, a maximum 4-phase PWM output is possible.
- PWM mode 2

  PWM output is generated using one TGR as the cycle register and the others as duty registers. The output specified in TIOR is performed by means of compare matches. Upon counter clearing by a synchronization register compare match, the output value of each pin is the initial value set in TIOR. If the set values of the cycle and duty registers are identical, the output value does not change when a compare match occurs. In PWM mode 2, a maximum 7-phase PWM output is possible by combined use with synchronous operation. The correspondence between PWM output pins and registers is shown in table 10.18.

Table 10.18 PWM Output Registers and Output Pins

		Output Pins	
Channel	**Registers**	**PWM Mode 1**	**PWM Mode 2**
0	TGRA_0	TIOCA0	TIOCA0
	TGRB_0		TIOCB0
	TGRC_0	TIOCC0	TIOCC0
	TGRD_0		TIOCD0
1	TGRA_1	TIOCA1	TIOCA1
	TGRB_1		TIOCB1
2	TGRA_2	TIOCA2	TIOCA2
	TGRB_2		TIOCB2

Note: In PWM mode 2, PWM output is not possible for the TGR register in which the period is set.

Example of PWM Mode Setting Procedure: Figure 10.21 shows an example of the PWM mode setting procedure.

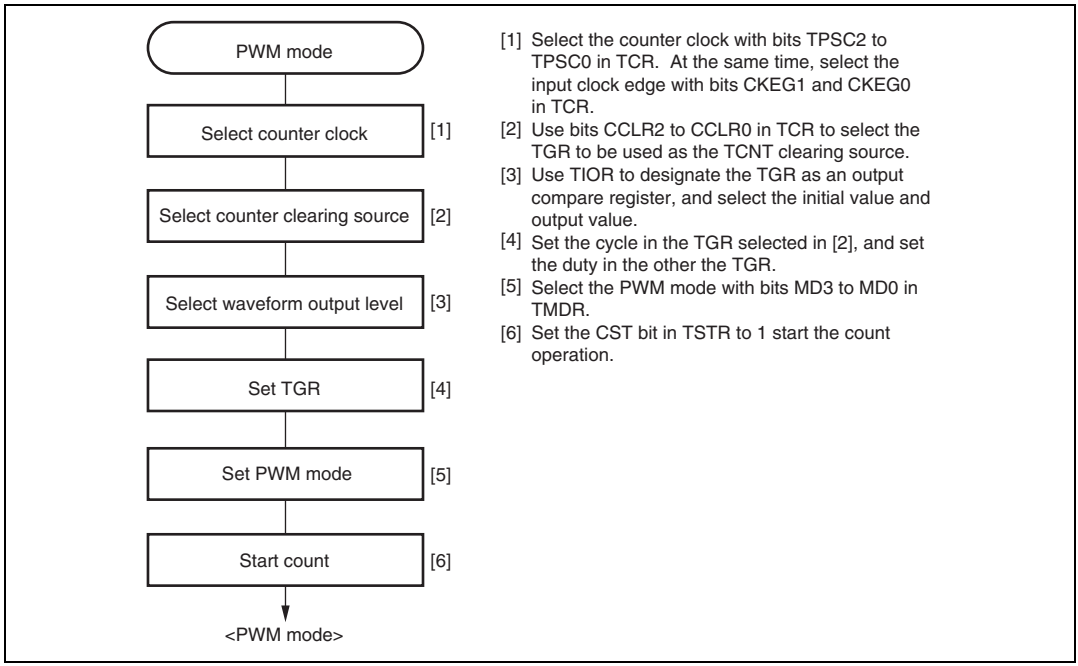


Figure 10.21 Example of PWM Mode Setting Procedure

Examples of PWM Mode Operation: Figure 10.22 shows an example of PWM mode 1 operation.

In this example, TGRA compare match is set as the TCNT clearing source, 0 is set for the TGRA initial output value and output value, and 1 is set as the TGRB output value. In this case, the value set in TGRA is used as the period, and the values set in TGRB registers as the duty.

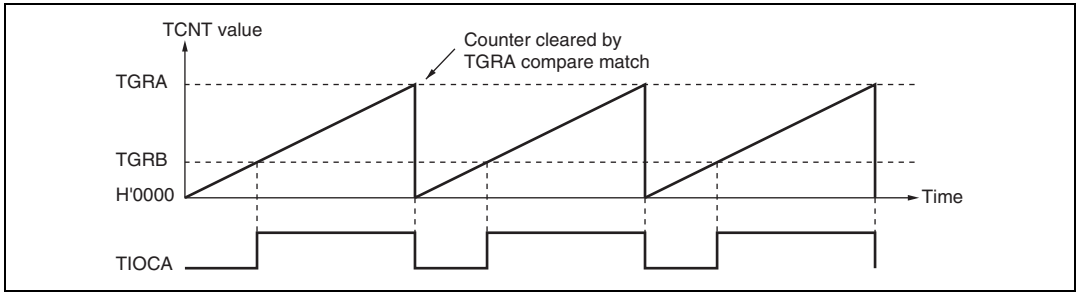


Figure 10.22 Example of PWM Mode Operation (1)

Figure 10.23 shows an example of PWM mode 2 operation. In this example, synchronous operation is designated for channels 0 and 1, TGRB_1 compare match is set as the TCNT clearing source, and 0 is set for the initial output value and 1 for the output value of the other TGR registers (TGRA_0 to TGRD_0, TGRA_1), to output a 5-phase PWM waveform. In this case, the value set in TGRB_1 is used as the cycle, and the values set in the other TGRs as the duty.

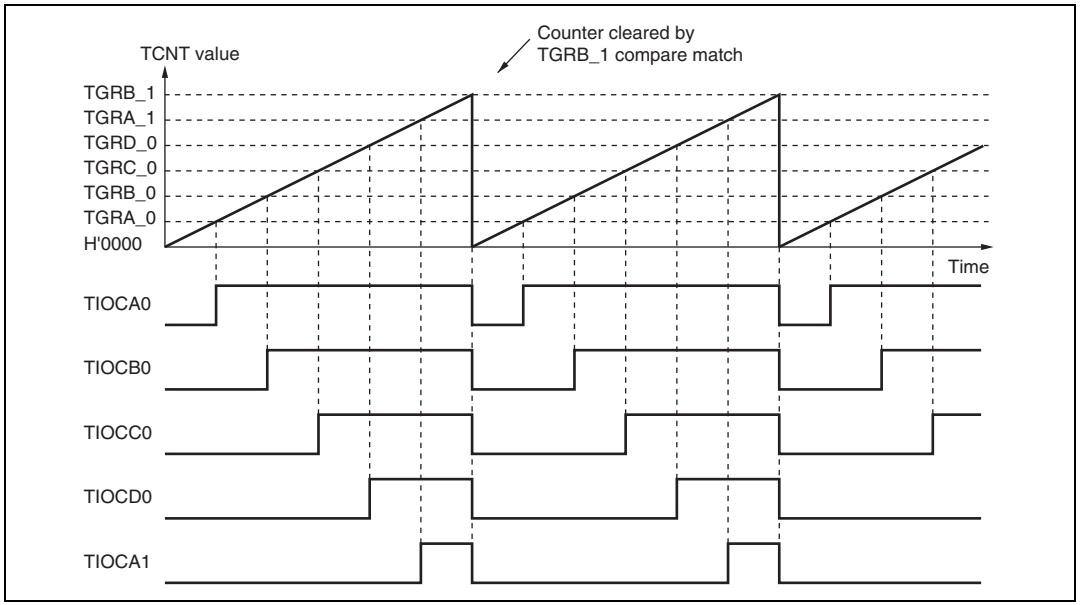


Figure 10.23 Example of PWM Mode Operation (2)

Figure 10.24 shows examples of PWM waveform output with 0% duty and 100% duty in PWM mode.

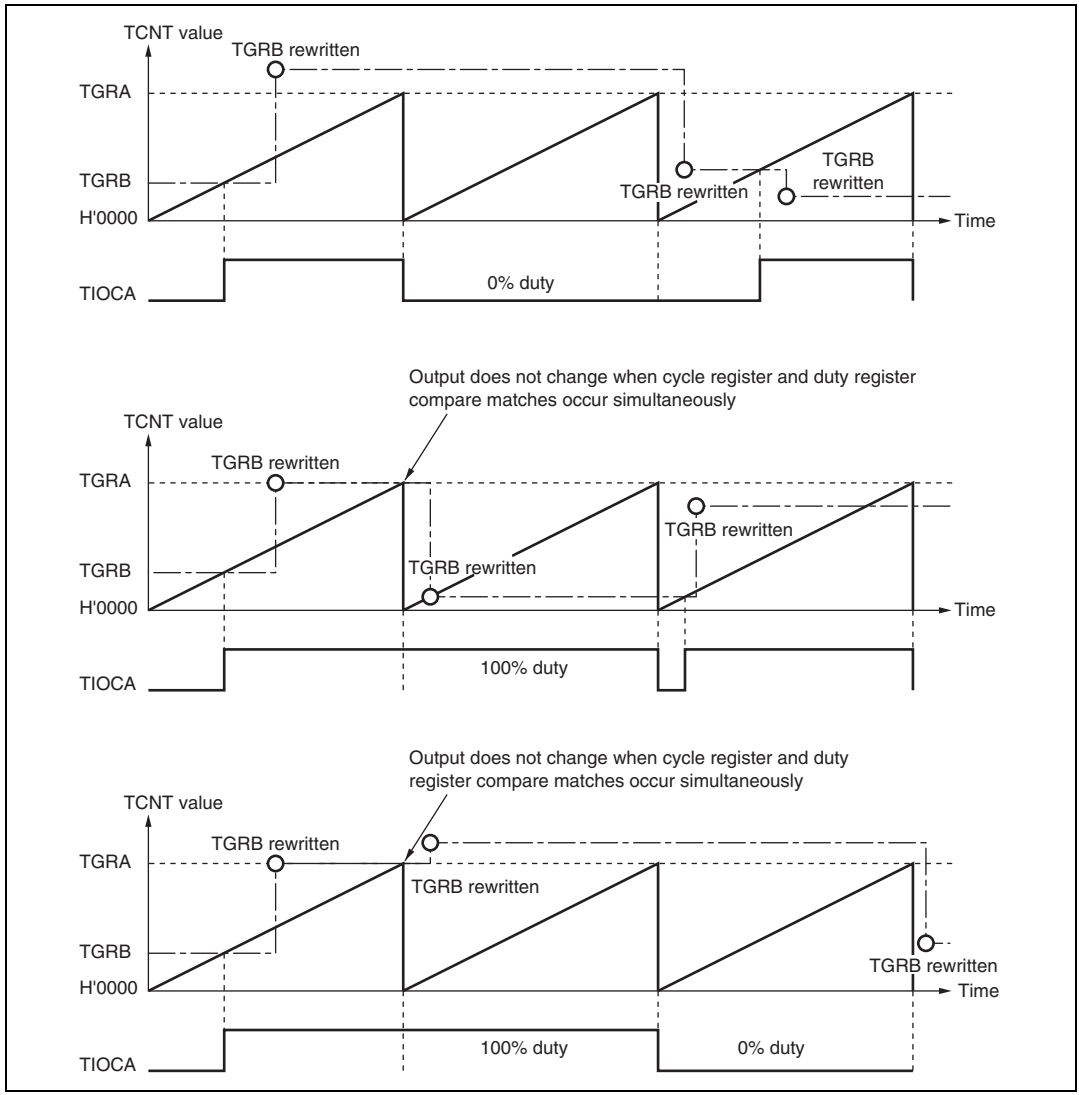


Figure 10.24 Example of PWM Mode Operation (3)

10.5.5 Phase Counting Mode

In phase counting mode, the phase difference between two external clock inputs is detected and TCNT is incremented/decremented accordingly. This mode can be set for channels 1 and 2. When phase counting mode is set, an external clock is selected as the counter input clock and TCNT operates as an up/down-counter regardless of the setting of bits TPSC2 to TPSC0 and bits CKEG1 and CKEG0 in TCR. However, the functions of bits CCLR1 and CCLR0 in TCR, and of TIOR, TIER, and TGR are valid, and input capture/compare match and interrupt functions can be used. This can be used for two-phase encoder pulse input. When overflow occurs while TCNT is counting up, the TCFV flag in TSR is set; when underflow occurs while TCNT is counting down, the TCFU flag is set. The TCFD bit in TSR is the count direction flag. Reading the TCFD flag provides an indication of whether TCNT is counting up or down. Table 10.19 shows the correspondence between external clock pins and channels.

Table 10.19 Phase Counting Mode Clock Input Pins

	External Clock Pins	
Channels	**A-Phase**	**B-Phase**
When channel 1 is set to phase counting mode	TCLKA	TCLKB
When channel 2 is set to phase counting mode	TCLKC	TCLKD

Example of Phase Counting Mode Setting Procedure: Figure 10.25 shows an example of the phase counting mode setting procedure.

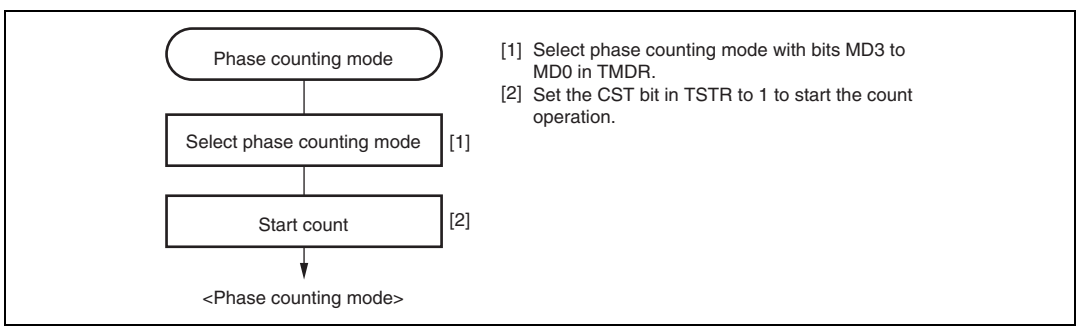


Figure 10.25 Example of Phase Counting Mode Setting Procedure

Examples of Phase Counting Mode Operation: In phase counting mode, TCNT counts up or down according to the phase difference between two external clocks. There are four modes, according to the count conditions.

1. Phase counting mode 1

   Figure 10.26 shows an example of phase counting mode 1 operation, and table 10.20 summarizes the TCNT up/down-count conditions.

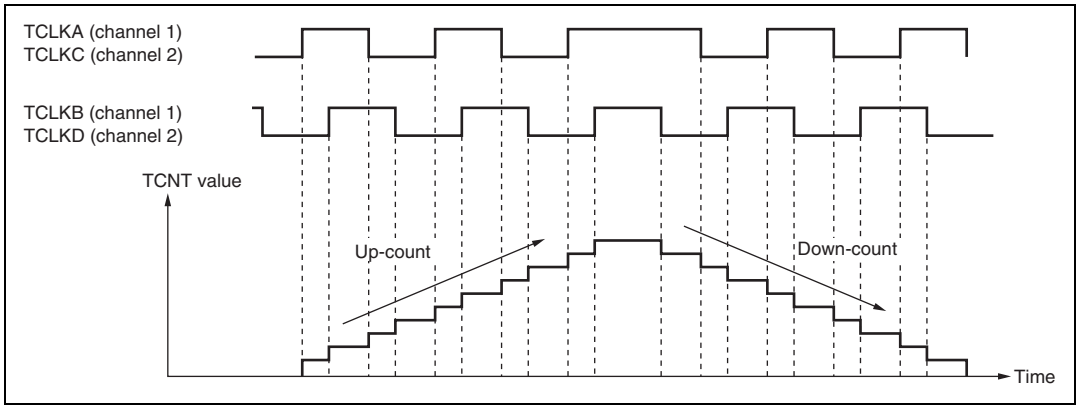


Figure 10.26 Example of Phase Counting Mode 1 Operation

Table 10.20 Up/Down-Count Conditions in Phase Counting Mode 1

TCLKA (Channel 1) / TCLKC (Channel 2)	TCLKB (Channel 1) / TCLKD (Channel 2)	Operation
High level	↑	Up-count
Low level	↓	
↑	Low level	
↓	High level	
High level	↓	Down-count
Low level	↑	
↑	High level	
↓	Low level	

Legend:
↑ : Rising edge
↓ : Falling edge

2. Phase counting mode 2

Figure 10.27 shows an example of phase counting mode 2 operation, and table 10.21 summarizes the TCNT up/down-count conditions.

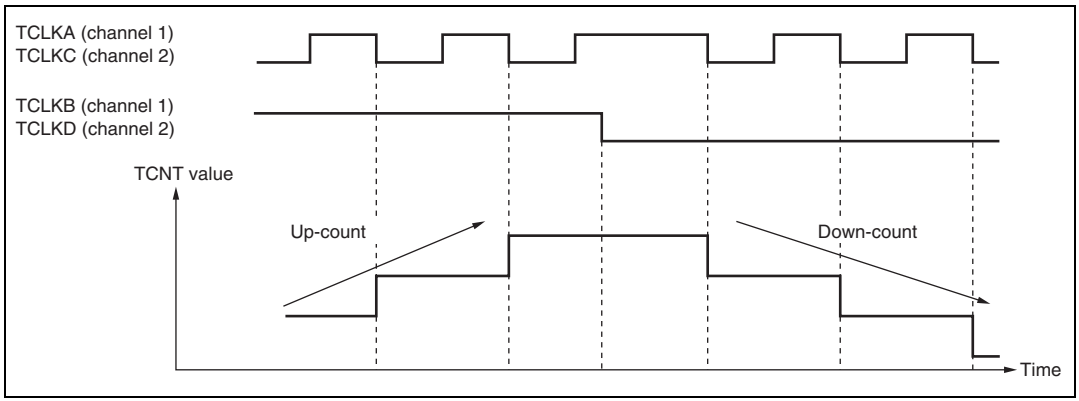


Figure 10.27 Example of Phase Counting Mode 2 Operation

Table 10.21 Up/Down-Count Conditions in Phase Counting Mode 2

TCLKA (Channel 1) TCLKC (Channel 2)	TCLKB (Channel 1) TCLKD (Channel 2)	Operation
High level	↑	Don't care
Low level	↓	
↑	Low level	
↓	High level	Up-count
High level	↓	Don't care
Low level	↑	
↑	High level	
↓	Low level	Down-count

Legend:
↑ : Rising edge
↓ : Falling edge

3. Phase counting mode 3

Figure 10.28 shows an example of phase counting mode 3 operation, and table 10.22 summarizes the TCNT up/down-count conditions.

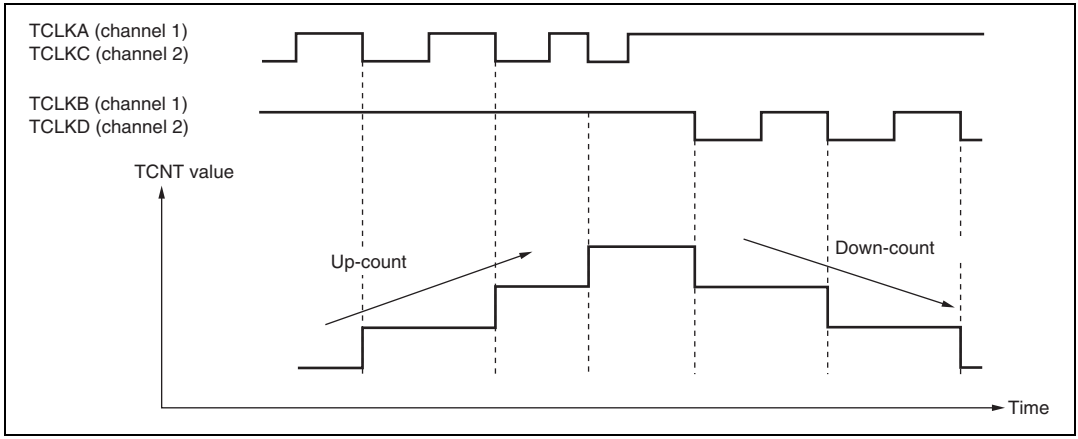


Figure 10.28 Example of Phase Counting Mode 3 Operation

Table 10.22 Up/Down-Count Conditions in Phase Counting Mode 3

TCLKA (Channel 1) / TCLKC (Channel 2)	TCLKB (Channel 1) / TCLKD (Channel 2)	Operation
High level	↑	Don't care
Low level	↓	
↑	Low level	
↓	High level	Up-count
High level	↓	Down-count
Low level	↑	Don't care
↑	High level	
↓	Low level	

Legend:
↑ : Rising edge
↓ : Falling edge

4. Phase counting mode 4

   Figure 10.29 shows an example of phase counting mode 4 operation, and table 10.23 summarizes the TCNT up/down-count conditions.

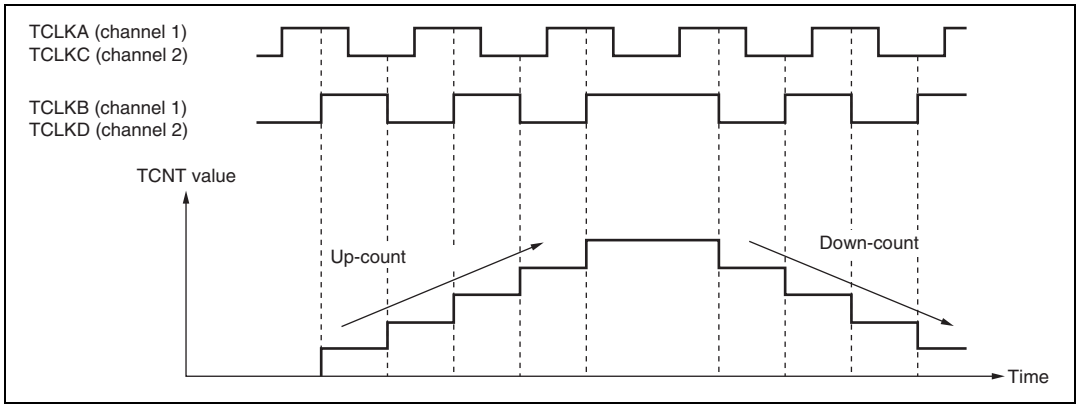


Figure 10.29 Example of Phase Counting Mode 4 Operation

Table 10.23 Up/Down-Count Conditions in Phase Counting Mode 4

TCLKA (Channel 1) TCLKC (Channel 2)	TCLKB (Channel 1) TCLKD (Channel 2)	Operation
High level	↑	Up-count
Low level	↓	
↑	Low level	Don't care
↓	High level	
High level	↓	Down-count
Low level	↑	
↑	High level	Don't care
↓	Low level	

Legend:
↑ : Rising edge
↓ : Falling edge

10.6 Interrupts

10.6.1 Interrupt Source and Priority

There are three kinds of TPU interrupt source: TGR input capture/compare match, TCNT overflow, and TCNT underflow. Each interrupt source has its own status flag and enable/disabled bit, allowing generation of interrupt request signals to be enabled or disabled individually. When an interrupt request is generated, the corresponding status flag in TSR is set to 1. If the corresponding enable/disable bit in TIER is set to 1 at this time, an interrupt is requested. The interrupt request is cleared by clearing the status flag to 0. Relative channel priorities can be changed by the interrupt controller, but the priority order within a channel is fixed. For details, see section 5, Interrupt Controller. Table 10.24 lists the TPU interrupt sources.

Table 10.24 TPU Interrupts

Channel	Name	Interrupt Source	Interrupt Flag	DTC Activation	DMAC Activation	Priority*
0	TGI0A	TGRA_0 input capture/compare match	TGFA	Possible	Possible	High
	TGI0B	TGRB_0 input capture/compare match	TGFB	Possible	Not possible	
	TGI0C	TGRC_0 input capture/compare match	TGFC	Possible	Not possible	
	TGI0D	TGRD_0 input capture/compare match	TGFD	Possible	Not possible	
	TCI0V	TCNT_0 overflow	TCFV	Not possible	Not possible	
1	TGI1A	TGRA_1 input capture/compare match	TGFA	Possible	Possible	
	TGI1B	TGRB_1 input capture/compare match	TGFB	Possible	Not possible	
	TCI1V	TCNT_1 overflow	TCFV	Not possible	Not possible	
	TCI1U	TCNT_1 underflow	TCFU	Not possible	Not possible	
2	TGI2A	TGRA_2 input capture/compare match	TGFA	Possible	Possible	
	TGI2B	TGRB_2 input capture/compare match	TGFB	Possible	Not possible	
	TCI2V	TCNT_2 overflow	TCFV	Not possible	Not possible	
	TCI2U	TCNT_2 underflow	TCFU	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.

Input Capture/Compare Match Interrupt: An interrupt is requested if the TGIE bit in TIER is set to 1 when the TGF flag in TSR is set to 1 by the occurrence of a TGR input capture/compare match on a particular channel. The interrupt request is cleared by clearing the TGF flag to 0. The TPU has 8 input capture/compare match interrupts, four each for channel 0, and two each for channels 1 and 2.

Overflow Interrupt: An interrupt is requested if the TCIEV bit in TIER is set to 1 when the TCFV flag in TSR is set to 1 by the occurrence of TCNT overflow on a channel. The interrupt request is cleared by clearing the TCFV flag to 0. The TPU has three overflow interrupts, one for each channel.

Underflow Interrupt: An interrupt is requested if the TCIEU bit in TIER is set to 1 when the TCFU flag in TSR is set to 1 by the occurrence of TCNT underflow on a channel. The interrupt request is cleared by clearing the TCFU flag to 0. The TPU has two underflow interrupts, one each for channels 1 and 2.

10.6.2 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 (DTC). A total of 8 TPU input capture/compare match interrupts can be used as DTC activation sources, four each for channel 0, and two each for channels 1 and 2.

10.6.3 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 (DMAC). With the TPU, a total of three TGRA input capture/compare match interrupts can be used as DMAC activation sources, one for each channel.

10.6.4 A/D Converter Activation

The A/D converter can be activated by the TGRA input capture/compare match for a channel. If the TTGE bit in TIER is set to 1 when the TGFA flag in TSR is set to 1 by the occurrence of a TGRA input capture/compare match on a particular channel, a request to start A/D conversion is sent to the A/D converter. If the TPU conversion start trigger has been selected on the A/D converter side at this time, A/D conversion is started. In the TPU, a total of three TGRA input capture/compare match interrupts can be used as A/D converter conversion start sources, one for each channel.

10.7 Operation Timing

10.7.1 Input/Output Timing

TCNT Count Timing: Figure 10.30 shows TCNT count timing in internal clock operation, and figure 10.31 shows TCNT count timing in external clock operation.

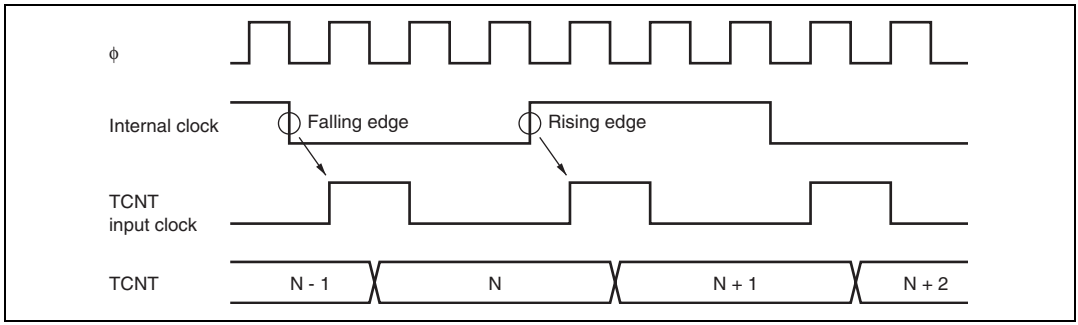


Figure 10.30 Count Timing in Internal Clock Operation

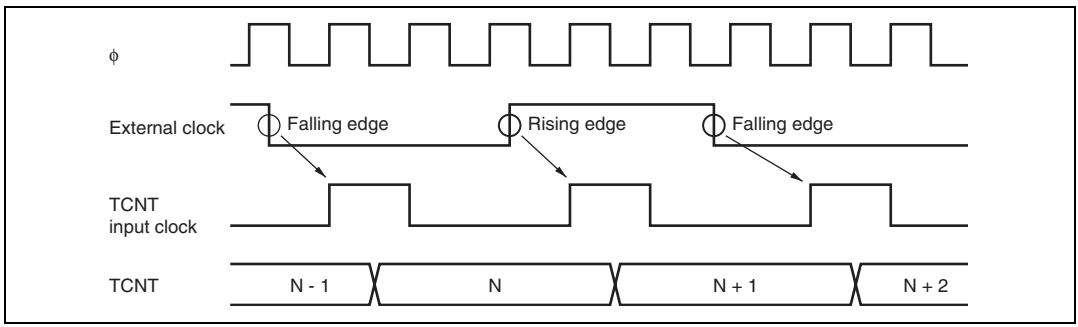


Figure 10.31 Count Timing in External Clock Operation

Output Compare Output Timing: A compare match signal is generated in the final state in which TCNT and TGR match (the point at which the count value matched by TCNT is updated). When a compare match signal is generated, the output value set in TIOR is output at the output compare output pin. After a match between TCNT and TGR, the compare match signal is not generated until the TCNT input clock is generated. Figure 10.32 shows output compare output timing.

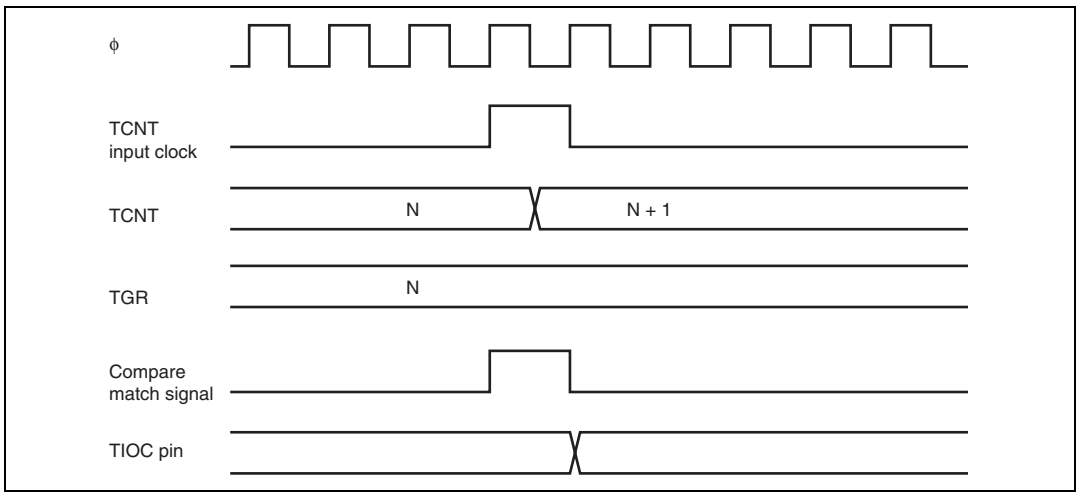


Figure 10.32 Output Compare Output Timing

Input Capture Signal Timing: Figure 10.33 shows input capture signal timing.

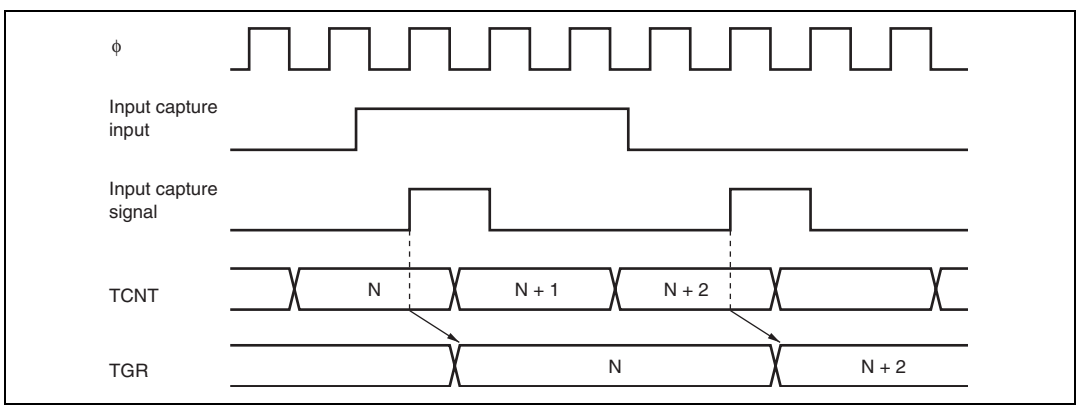


Figure 10.33 Input Capture Input Signal Timing

Timing for Counter Clearing by Compare Match/Input Capture: Figure 10.34 shows the timing when counter clearing by compare match occurrence is specified, and figure 10.35 shows the timing when counter clearing by input capture occurrence is specified.

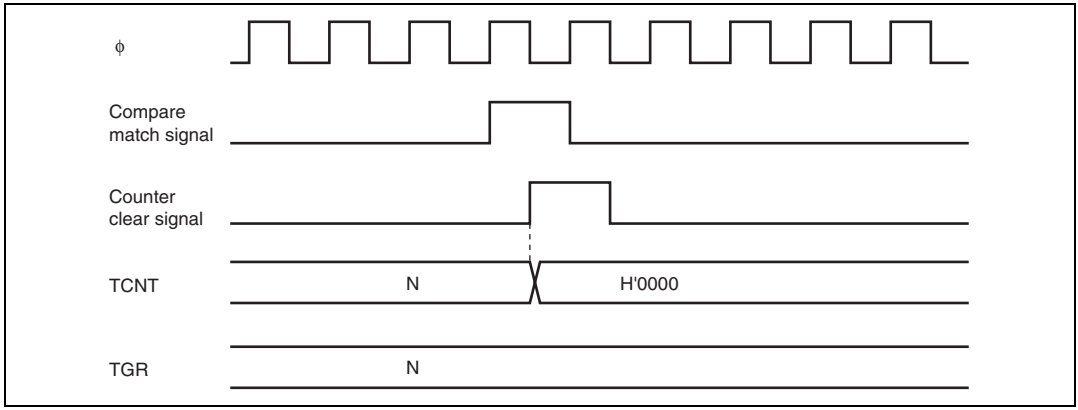


Figure 10.34 Counter Clear Timing (Compare Match)

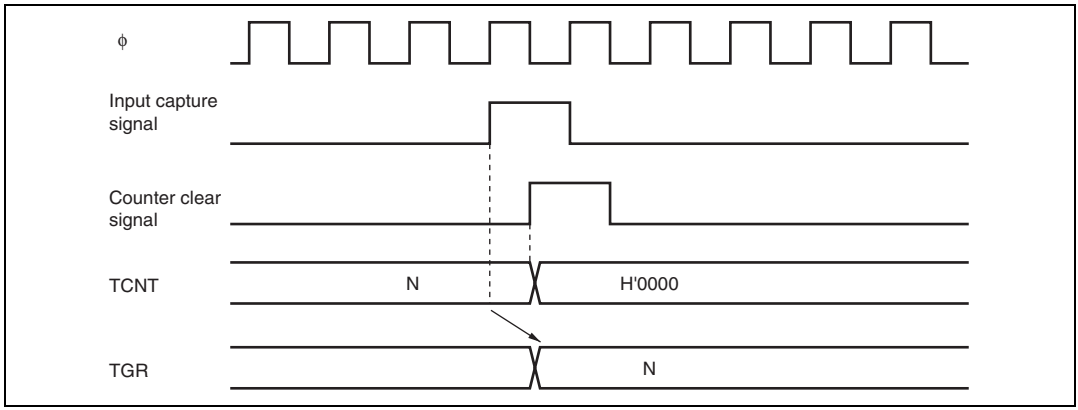


Figure 10.35 Counter Clear Timing (Input Capture)

Buffer Operation Timing: Figures 10.36 and 10.37 show the timing in buffer operation.

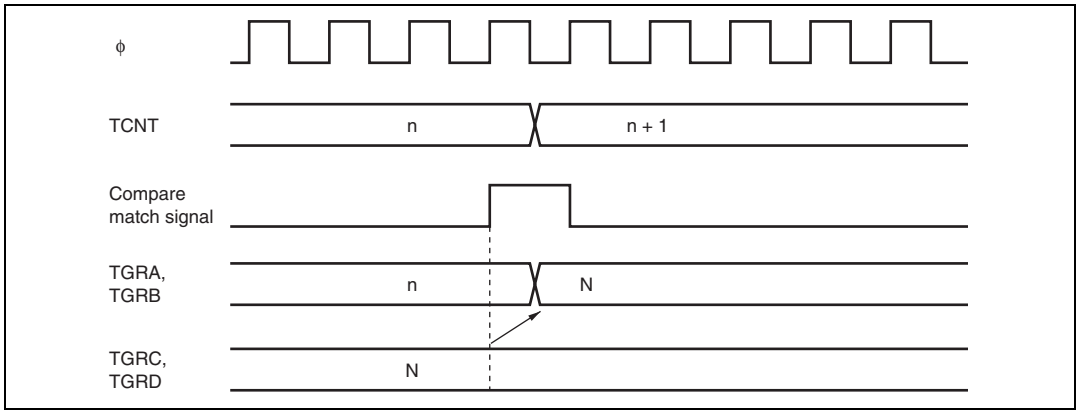


Figure 10.36 Buffer Operation Timing (Compare Match)

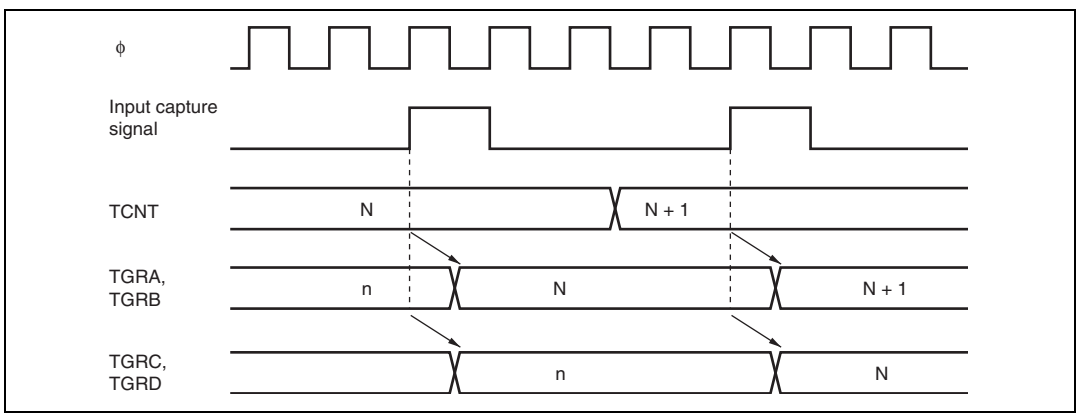


Figure 10.37 Buffer Operation Timing (Input Capture)

10.7.2 Interrupt Signal Timing

TGF Flag Setting Timing in Case of Compare Match: Figure 10.38 shows the timing for setting of the TGF flag in TSR by compare match occurrence, and TGI interrupt request signal timing.

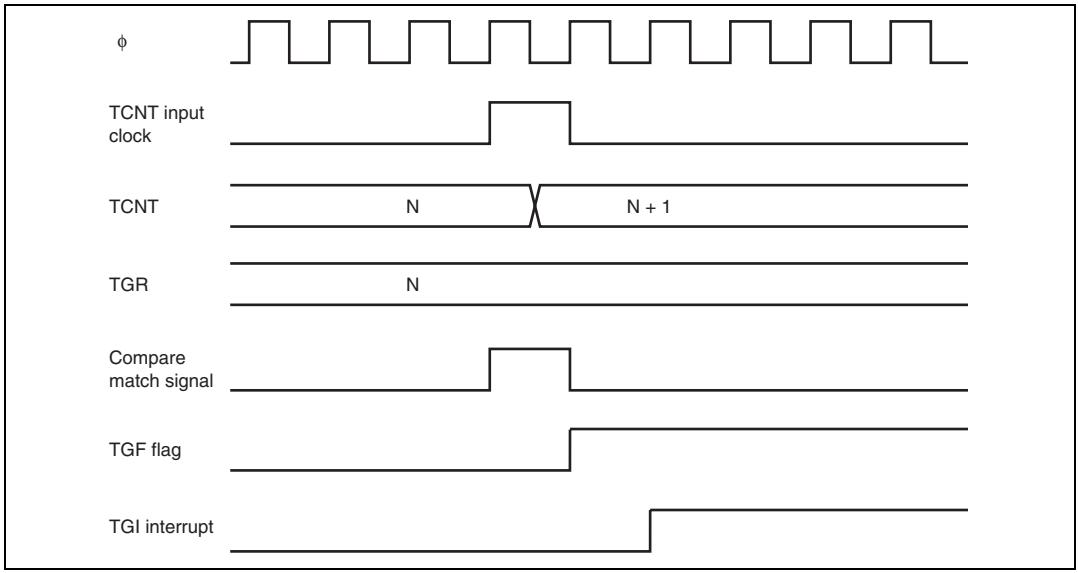


Figure 10.38 TGI Interrupt Timing (Compare Match)

TGF Flag Setting Timing in Case of Input Capture: Figure 10.39 shows the timing for setting of the TGF flag in TSR by input capture occurrence, and TGI interrupt request signal timing.

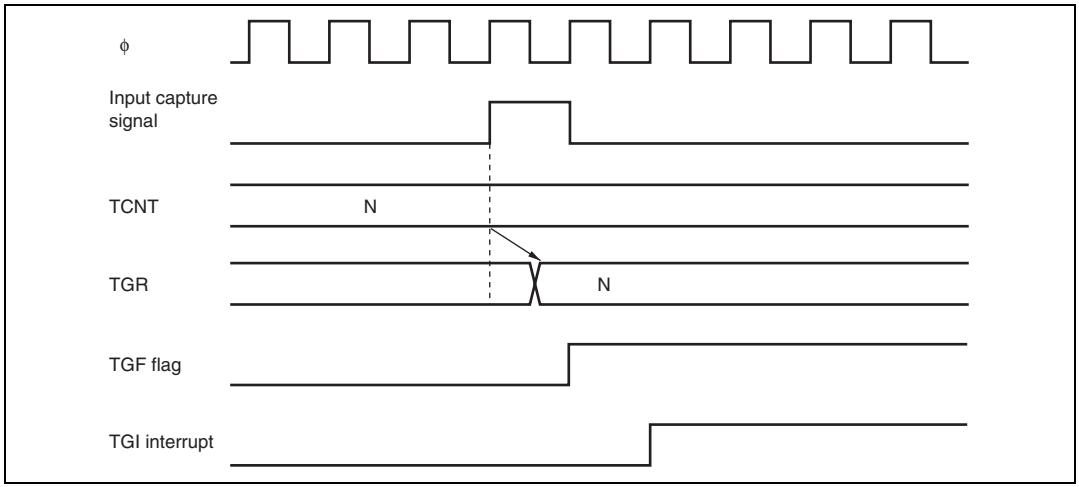


Figure 10.39 TGI Interrupt Timing (Input Capture)

TCFV Flag/TCFU Flag Setting Timing: Figure 10.40 shows the timing for setting of the TCFV flag in TSR by overflow occurrence, and TCIV interrupt request signal timing. Figure 10.41 shows the timing for setting of the TCFU flag in TSR by underflow occurrence, and TCIU interrupt request signal timing.

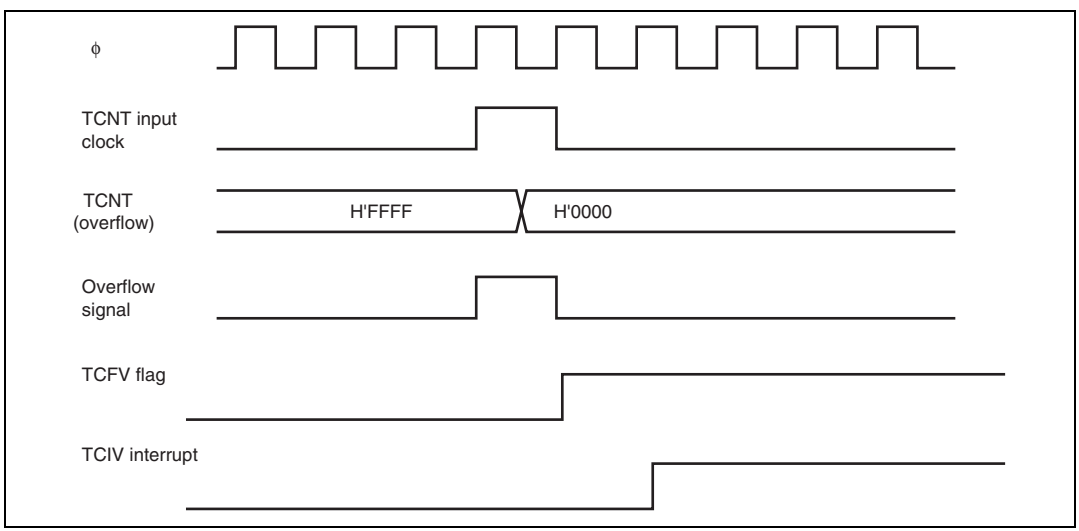


Figure 10.40 TCIV Interrupt Setting Timing

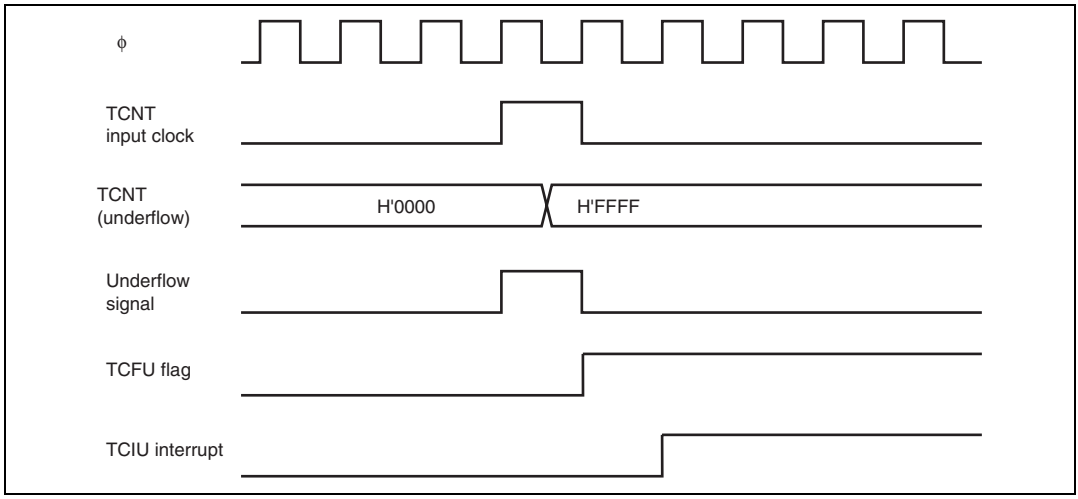


Figure 10.41 TCIU Interrupt Setting Timing

Status Flag Clearing Timing: After a status flag is read as 1 by the CPU, it is cleared by writing 0 to it. When the DTC or DMAC is activated, the flag is cleared automatically. Figure 10.42 shows the timing for status flag clearing by the CPU, and figure 10.43 shows the timing for status flag clearing by the DTC or DMAC.

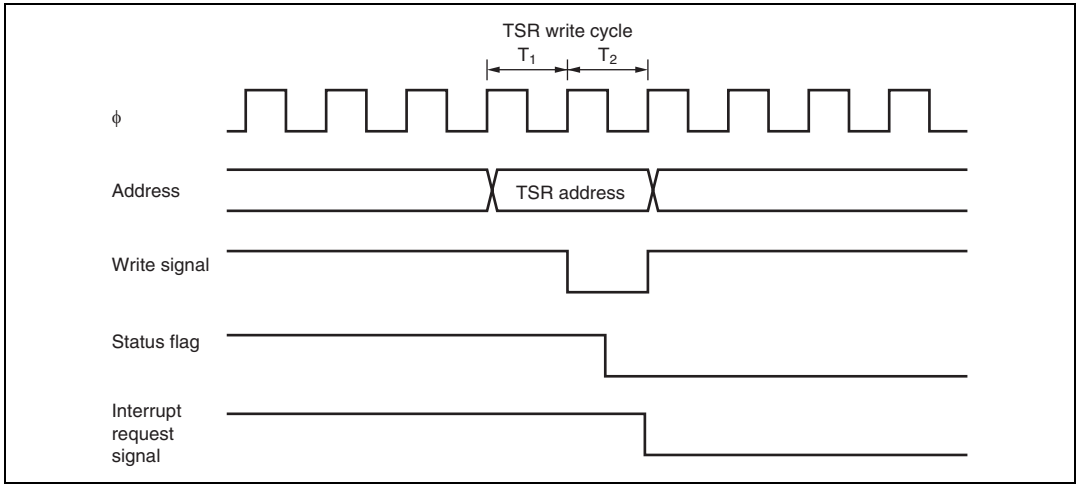


Figure 10.42 Timing for Status Flag Clearing by CPU

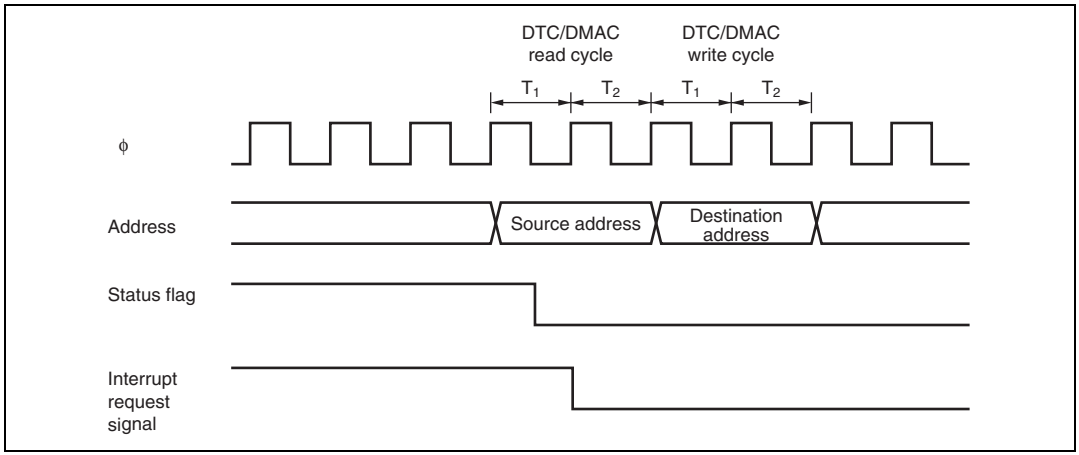


Figure 10.43 Timing for Status Flag Clearing by DTC or DMAC Activation

10.8 Usage Notes

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 10.44 shows the input clock conditions in phase counting mode.

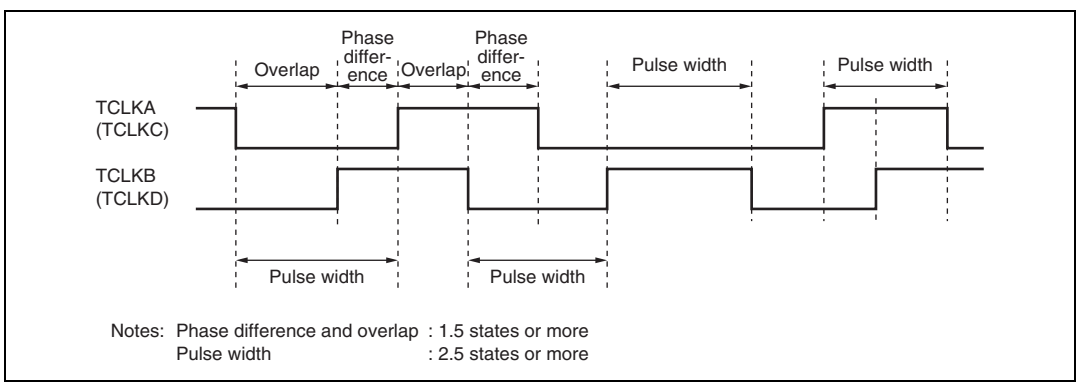


Figure 10.44 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

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

Contention between TCNT Write and Clear Operations: If the counter clear signal is generated in the T_2 state of a TCNT write cycle, TCNT clearing takes precedence and the TCNT write is not performed. Figure 10.45 shows the timing in this case.

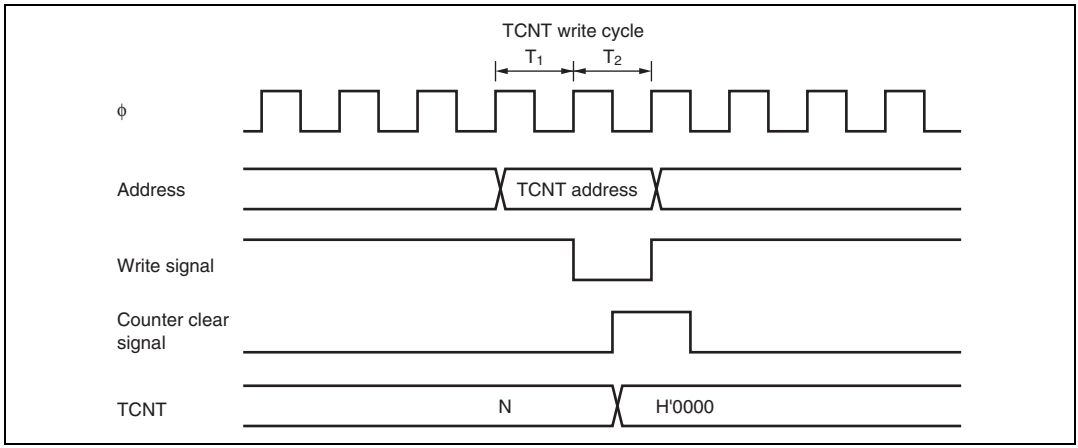


Figure 10.45 Contention between TCNT Write and Clear Operations

Contention between TCNT Write and Increment Operations: If incrementing occurs in the T_2 state of a TCNT write cycle, the TCNT write takes precedence and TCNT is not incremented. Figure 10.46 shows the timing in this case.

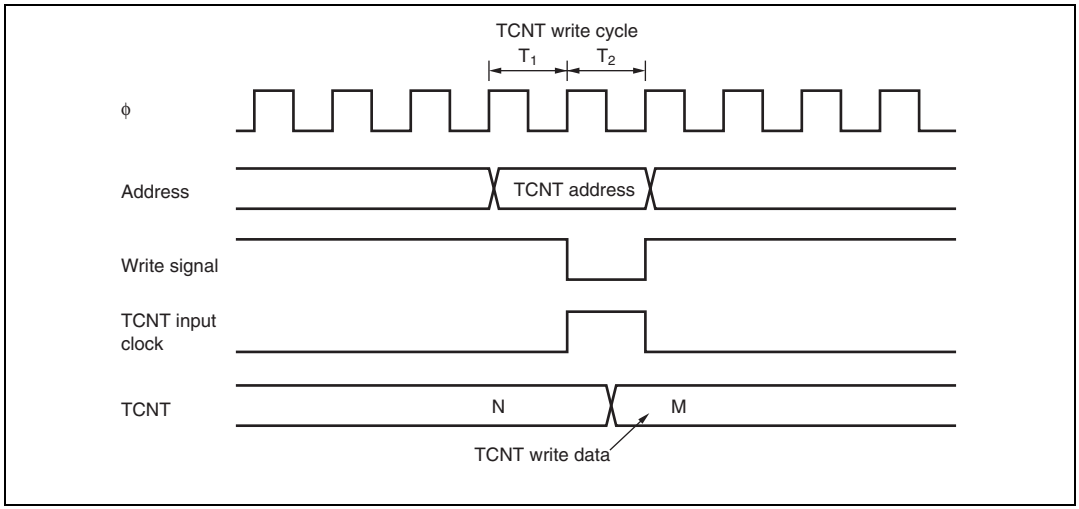


Figure 10.46 Contention between TCNT Write and Increment Operations

Contention between TGR Write and Compare Match: If a compare match occurs in the T_2 state of a TGR write cycle, the TGR write takes precedence and the compare match signal is inhibited. A compare match does not occur even if the same value as before is written. Figure 10.47 shows the timing in this case.

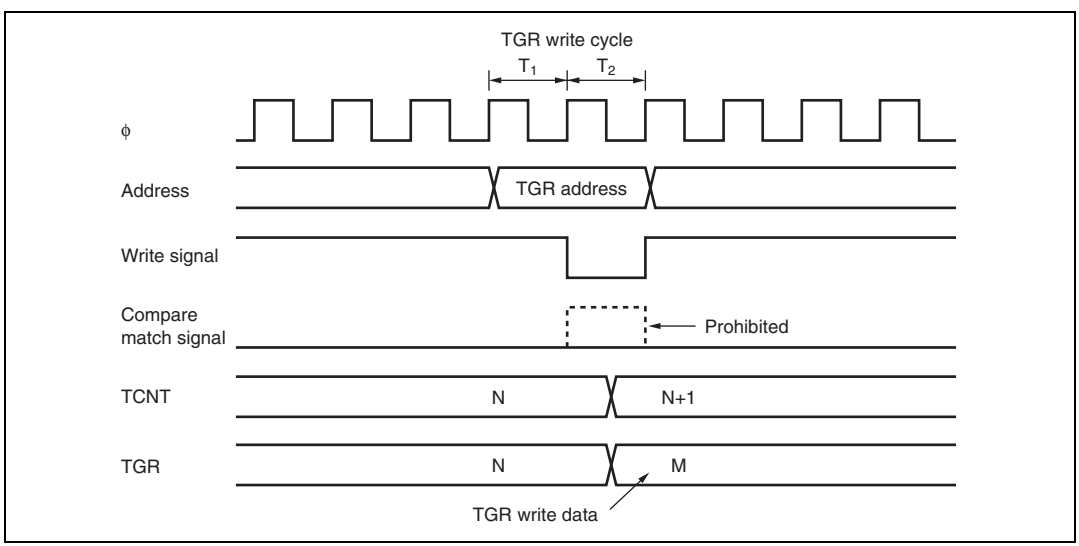


Figure 10.47 Contention between TGR Write and Compare Match

Contention between Buffer Register Write and Compare Match: If a compare match occurs in the T_2 state of a TGR write cycle, the data transferred to TGR by the buffer operation will be the data prior to the write. Figure 10.48 shows the timing in this case.

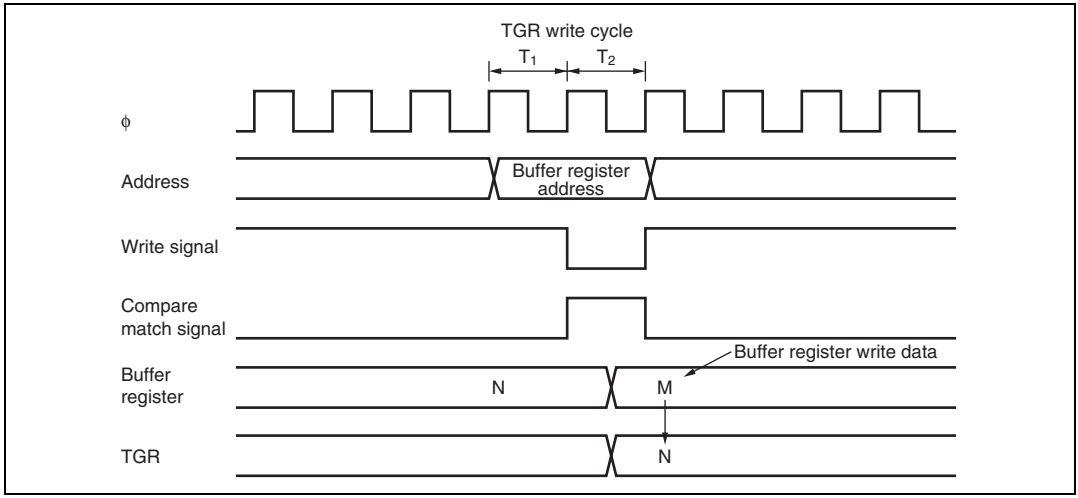


Figure 10.48 Contention between Buffer Register Write and Compare Match

Contention between TGR Read and Input Capture: If the input capture signal is generated in the T_1 state of a TGR read cycle, the data that is read will be the data after input capture transfer. Figure 10.49 shows the timing in this case.

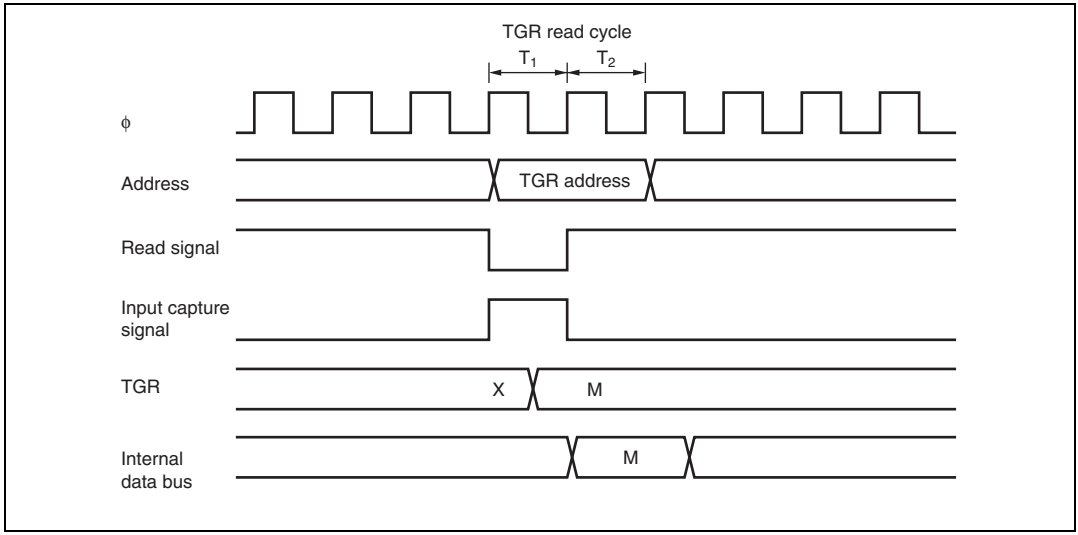


Figure 10.49 Contention between TGR Read and Input Capture

Contention between TGR Write and Input Capture: If the input capture signal is generated in the T_2 state of a TGR write cycle, the input capture operation takes precedence and the write to TGR is not performed. Figure 10.50 shows the timing in this case.

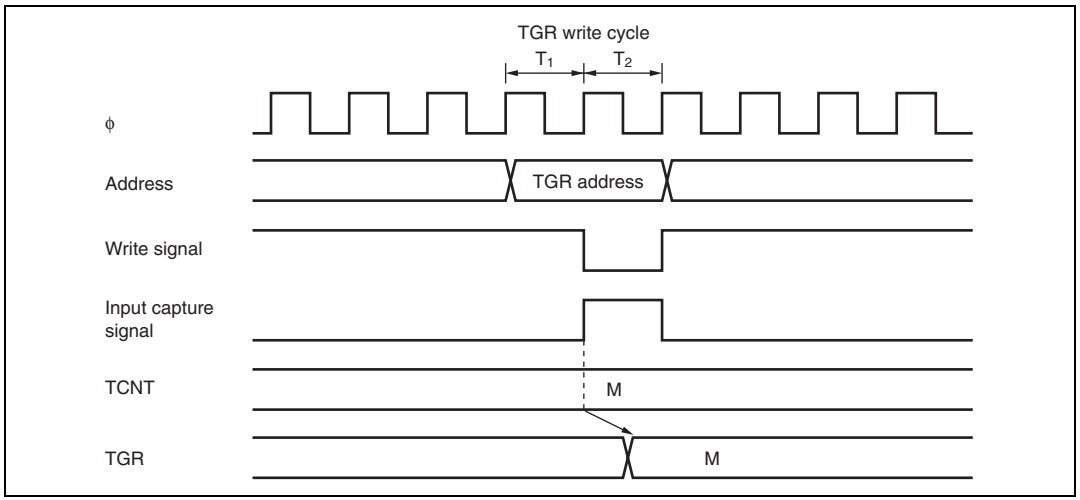


Figure 10.50 Contention between TGR Write and Input Capture

Contention between Buffer Register Write and Input Capture: If the input capture signal is generated in the T_2 state of a buffer register write cycle, the buffer operation takes precedence and the write to the buffer register is not performed. Figure 10.51 shows the timing in this case.

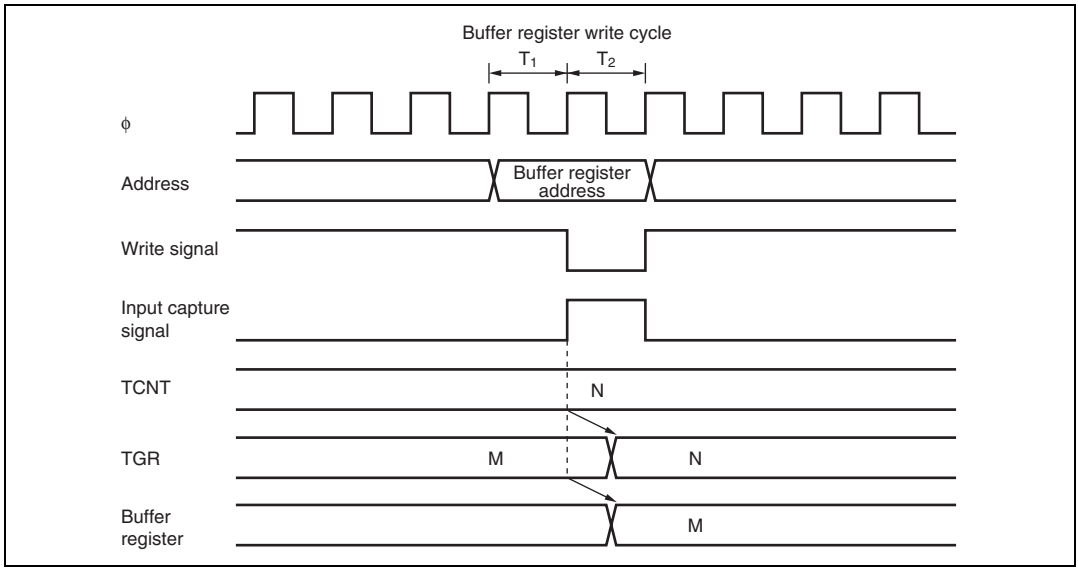


Figure 10.51 Contention between Buffer Register Write and Input Capture

Contention between Overflow/Underflow and Counter Clearing: If overflow/underflow and counter clearing occur simultaneously, the TCFV/TCFU flag in TSR is not set and TCNT clearing takes precedence. Figure 10.52 shows the operation timing when a TGR compare match is specified as the clearing source, and H'FFFF is set in TGR.

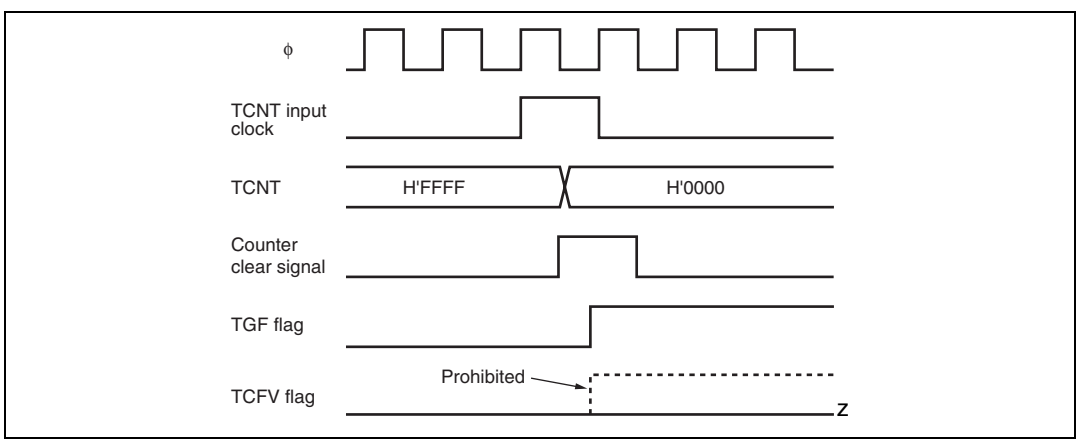


Figure 10.52 Contention between Overflow and Counter Clearing

Contention between TCNT Write and Overflow/Underflow: If there is an up-count or down-count in the T_2 state of a TCNT write cycle, and overflow/underflow occurs, the TCNT write takes precedence and the TCFV/TCFU flag in TSR is not set. Figure 10.53 shows the operation timing when there is contention between TCNT write and overflow.

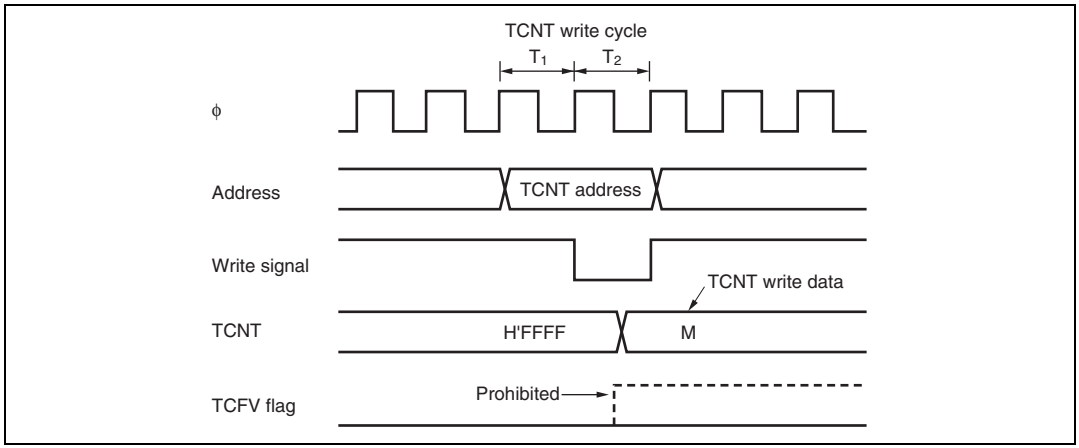


Figure 10.53 Contention between TCNT Write and Overflow

Multiplexing of I/O Pins: In this LSI, the TCLKA input pin is multiplexed with the TIOCC0 I/O pin, the TCLKB input pin with the TIOCD0 I/O pin, the TCLKC input pin with the TIOCB1 I/O pin, and the TCLKD input pin with the TIOCB2 I/O pin. When an external clock is input, compare match output should not be performed from a multiplexed pin.

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 DTC activation source. Interrupts should therefore be disabled before entering module stop mode.

Module Stop Mode Setting: TPU operation can be disabled or enabled using the module stop control register. The initial setting is for TPU operation to be halted. Register access is enabled by clearing module stop mode. For details, refer to section 22, Power-Down Modes.

Section 11 8-Bit Timers (TMR)

This LIS includes an 8-bit timer module with two channels. Each channel has an 8-bit counter and two registers that are constantly compared with the TCNT value to detect compare match events.

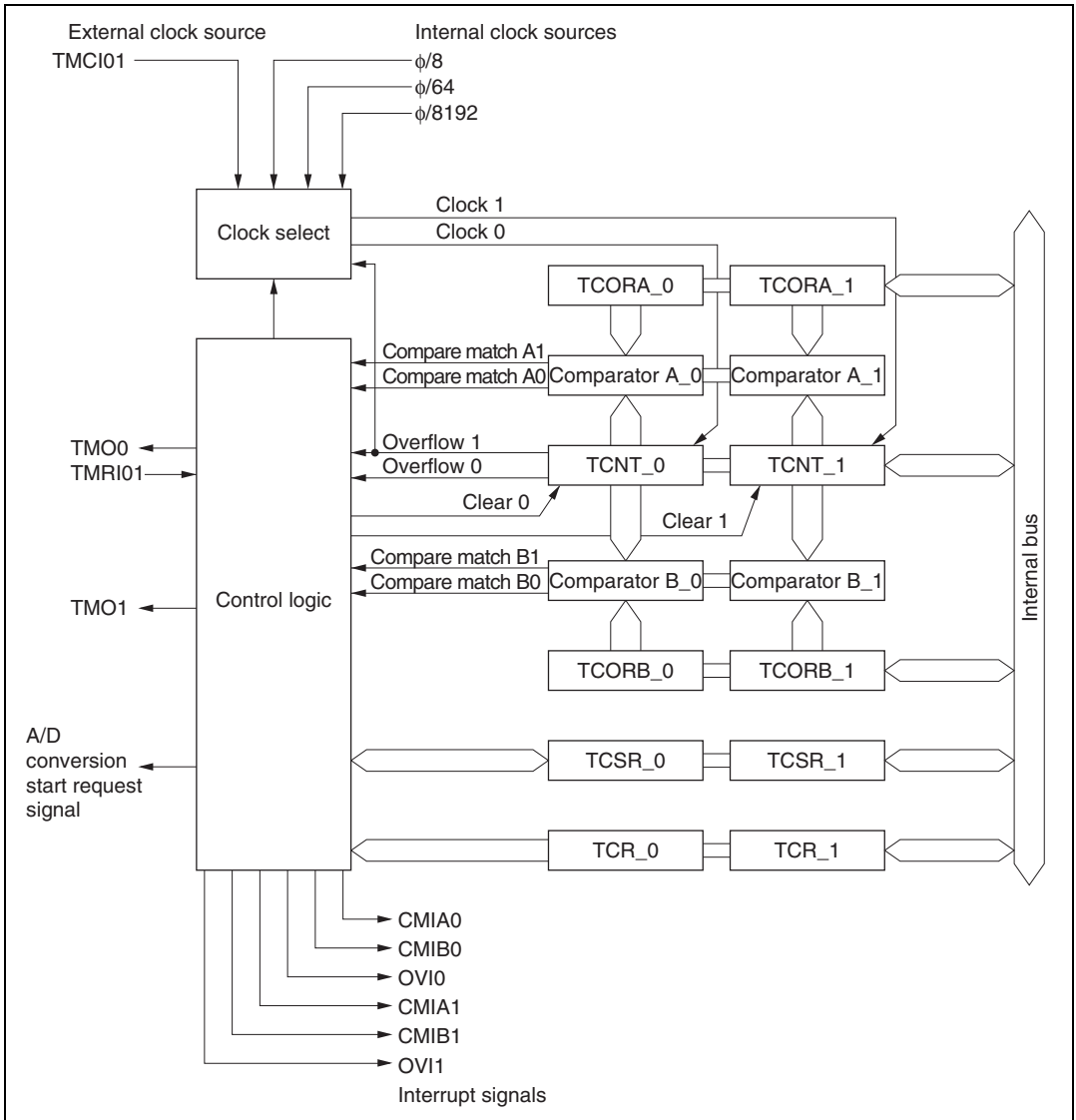
The 8-bit timer module can thus be used for a variety of functions, including pulse output with an arbitrary duty cycle.

11.1 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 (TMR_0) for the upper 8 bits and channel 1 (TMR_1) for the lower 8 bits (16-bit count mode).
  - Channel 1 (TMR_1) can be used to count channel 0 (TMR_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

Figure 11.1 shows a block diagram of the 8-bit timer module (TMR_0 and TMR_1).



Legend:

TCORA_0:	Time constant register A_0	TCORA_1:	Time constant register A_1
TCORB_0:	Time constant register B_0	TCORB_1:	Time constant register B_1
TCNT_0:	Timer counter_0	TCNT_1:	Timer counter_1
TCSR_0:	Timer control/status register_0	TCSR_1:	Timer control/status register_1
TCR_0:	Timer control register_0	TCR_1:	Timer control register_1

Figure 11.1 Block Diagram of 8-Bit Timer

11.2 Input/Output Pins

Table 11.1 summarizes the input and output pins of the TMR.

Table 11.1 Pin Configuration

Channel	Name	Symbol	I/O	Function
0	Timer output pin 0	TMO0	Output	Outputs at compare match
1	Timer output pin 1	TMO1	Output	Outputs at compare match
All	Timer clock input pin 01	TMCI01	Input	Inputs external clock for counter
	Timer reset input pin 01	TMRI01	Input	Inputs external reset to counter

11.3 Register Descriptions

The TMR registers are listed below. For details on the module stop control register, refer to section 22.1.2, Module Stop Registers A to C (MSTPCRA to MSTPCRC).

- Timer counter (TCNT)
- Time constant register A (TCORA)
- Time constant register B (TCORB)
- Timer control register (TCR)
- Timer control/status register (TCSR)

11.3.1 Timer Counters (TCNT)

The TCNT registers are 8-bit up-counters. TCNT_0 and TCNT_1 comprise a single 16-bit register so they can be accessed together by a word transfer instruction. Bits CKS2 to CKS0 in TCR are used to select a clock. The TCNT counters can be cleared by an external reset input or by a compare match signal A or B. Which signal is to be used for clearing is selected by bits CCLR1 and CCLR0 in TCR. When a TCNT counter overflows from H'FF to H'00, OVF in TCSR is set to 1. The TCNT counters are each initialized to H'00.

11.3.2 Time Constant Registers A (TCORA)

The TCORA_0 and TCORA_1 registers are 8-bit readable/writable registers. TCORA_0 and TCORA_1 comprise a single 16-bit register so they can be accessed together by a word transfer instruction. The value in TCORA is continually compared with the value in TCNT. When a match is detected, the corresponding CMFA flag in TCSR is set to 1. Note, however, that comparison is disabled during the T_2 state of a TCOR write cycle. The timer output (TMO) can be freely controlled by these compare match signals and the settings of bits OS1 and OS0 in TCSR. TCORA_0 and TCORA_1 are each initialized to H'FF.

11.3.3 Time Constant Registers B (TCORB)

The TCORB_0 registers are 8-bit readable/writable registers. TCORB_0 and TCORB_1 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 to 1. 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. TCORB_0 and TCORB_1 are each initialized to H'FF.

11.3.4 Time Control Registers (TCR)

The TCR registers select the clock source and the time at which TCNT is cleared, and enable interrupts.

Bit	Bit Name	Initial Value	R/W	Description
7	CMIEB	0	R/W	Compare Match Interrupt Enable B Selects whether CMFB interrupt requests (CMIB) are enabled or disabled when the CMFB flag in TCSR is set to 1. 0: CMFB interrupt requests (CMIB) are disabled 1: CMFB interrupt requests (CMIB) are enabled
6	CMIEA	0	R/W	Compare Match Interrupt Enable A Selects whether CMFA interrupt requests (CMIA) are enabled or disabled when the CMFA flag in TCSR is set to 1. 0: CMFA interrupt requests (CMIA) are disabled 1: CMFA interrupt requests (CMIA) are enabled
5	OVIE	0	R/W	Timer Overflow Interrupt Enable Selects whether OVF interrupt requests (OVI) are enabled or disabled when the OVF flag in TCSR is set to 1. 0: OVF interrupt requests (OVI) are disabled 1: OVF interrupt requests (OVI) are enabled
4 3	CCLR1 CCLR0	0 0	R/W R/W	Counter Clear 1 and 0 These bits select the method by which TCNT is cleared. 00: Clear is disabled 01: Clear by compare match A 10: Clear by compare match B 11: Clear by rising edge of external reset input
2 1 0	CKS2 CKS1 CKS0	0 0 0	R/W R/W R/W	Clock Select 2 to 0 These bits select the clock input to TCNT and count condition. See table 11.2.

Table 11.2 Clock Input to TCNT and Count Condition

	TCR			
	Bit 2	Bit 1	Bit 0	
Channel	CKS2	CKS1	CKS0	Description
TMR_0	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	Count at TCNT1 overflow signal*
TMR_1	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	Count at TCNT0 compare match A*
All	1	0	1	External clock, counted at rising edge
		1	0	External clock, counted at falling edge
		1	1	External clock, counted at both rising and falling edges

Note: * If the count input of TMR_0 is the TCNT_1 overflow signal and that of TMR_1 is the TCNT_0 compare match signal, no incrementing clock is generated. This setting is prohibited.

11.3.5 Timer Control/Status Registers (TCSR)

The TCSR registers display status flags, and control compare match output.

Bit	Bit Name	Initial Value	R/W	Description
7	CMFB	0	R/(W)*	Compare Match Flag B [Setting condition] • Set when TCNT matches TCORB [Clearing conditions] • Cleared by reading CMFB when CMFB = 1, then writing 0 to CMFB • When DTC is activated by CMIB interrupt, while DISEL bit is 0, and transfer counter value is not 0
6	CMFA	0	R/(W)*	Compare Match Flag A [Setting condition] • Set when TCNT matches TCORA [Clearing conditions] • Cleared by reading CMFA when CMFA = 1, then writing 0 to CMFA • When DTC is activated by CMIA interrupt, while DISEL bit is 0, and transfer counter value is not 0
5	OVF	0	R/(W)*	Timer Overflow Flag [Setting condition] • Set when TCNT overflows from H'FF to H'00 [Clearing condition] • Cleared by reading OVF when OVF = 1, then writing 0 to OVF
4	ADTE	0	R/W	A/D Trigger Enable (only in channel 0) Selects enabling or disabling of A/D converter start requests by compare match A. This bit is reserved in channel 1. Always read as 1, and cannot be modified. 0: A/D converter start requests by compare match A are disabled 1: A/D converter start requests by compare match A are enabled
3	OS3	0	R/W	Output Select 3 and 2 These bits select a method of TMO pin output when compare match B of TCOR and TCNT occurs. 00: No change when compare match B occurs 01: 0 is output when compare match B occurs 10: 1 is output when compare match B occurs 11: Output is inverted when compare match B occurs (toggle output)
2	OS2	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
1	OS1	0	R/W	Output Select 1 and 0
0	OS0	0	R/W	These bits select a method of TMO pin output when compare match A of TCOR and TCNT occurs. 00: No change when compare match A occurs 01: 0 is output when compare match A occurs 10: 1 is output when compare match A occurs 11: Output is inverted when compare match A occurs (toggle output)

Note: * The write value should always be 0 to clear these flags.

11.4 Operation

11.4.1 Pulse Output

Figure 11.2 shows an example that the 8-bit timer is used to generate a pulse output with a selected duty cycle. 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 at a TCORA compare match.
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.

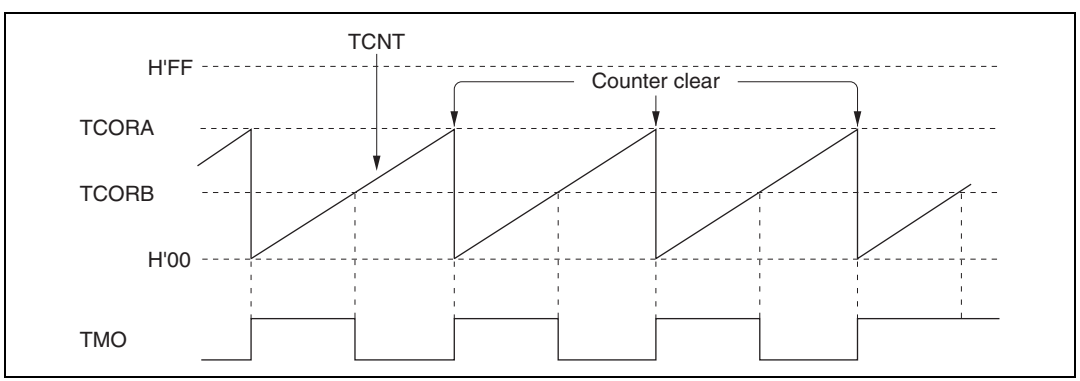


Figure 11.2 Example of Pulse Output

11.5 Operation Timing

11.5.1 TCNT Incrementation Timing

Figure 11.3 shows the count timing for internal clock input. Figure 11.4 shows the count timing for external clock signal. 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.

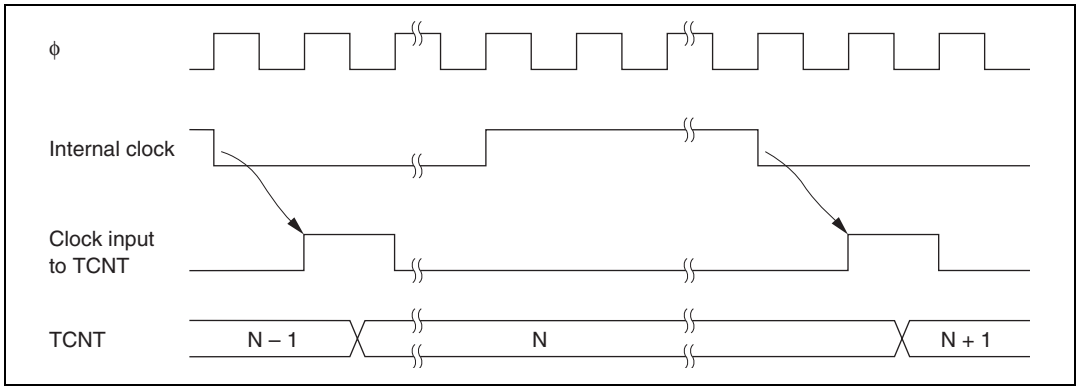


Figure 11.3 Count Timing for Internal Clock Input

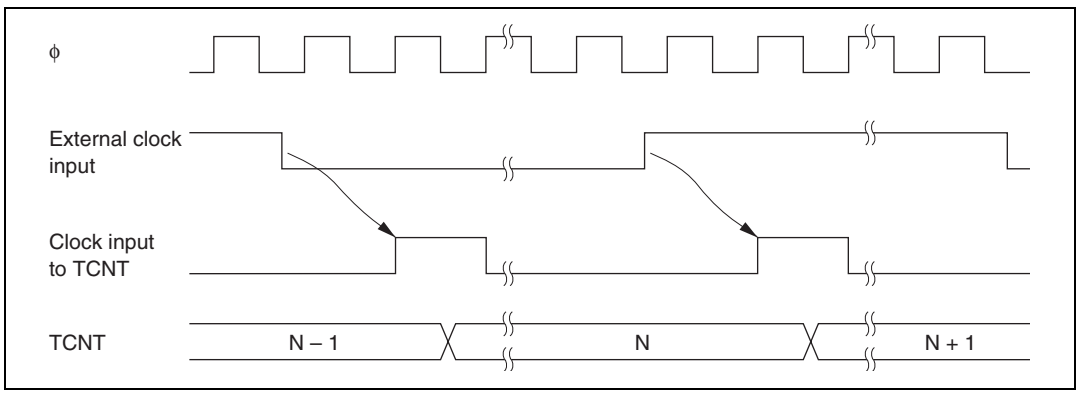


Figure 11.4 Count Timing for External Clock Input

11.5.2 Setting of Compare Match Flags CMFA and 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 11.5 shows this timing.

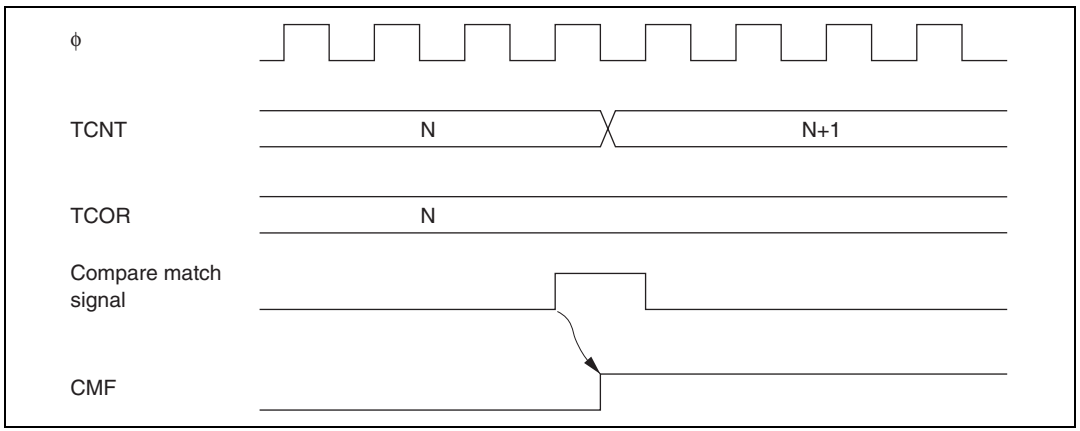


Figure 11.5 Timing of CMF Setting

11.5.3 Timer Output Timing

When compare match A or B occurs, the timer output changes as specified by bits OS3 to OS0 in TCSR. Figure 11.6 shows the timing when the output is set to toggle at compare match A.

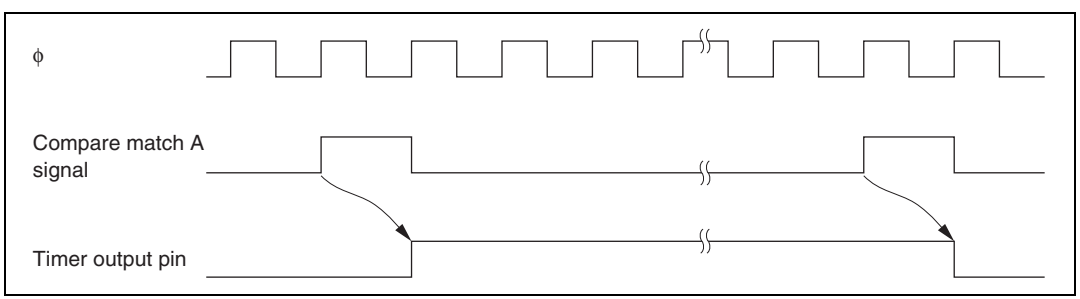


Figure 11.6 Timing of Timer Output

11.5.4 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 11.7 shows the timing of this operation.

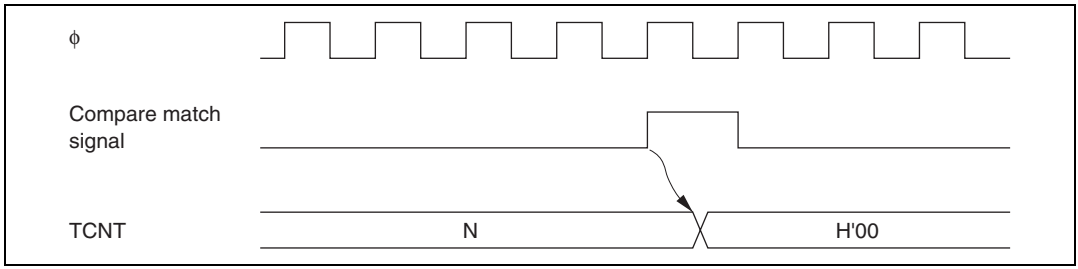


Figure 11.7 Timing of Compare Match Clear

11.5.5 Timing of TCNT External Reset

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 11.8 shows the timing of this operation.

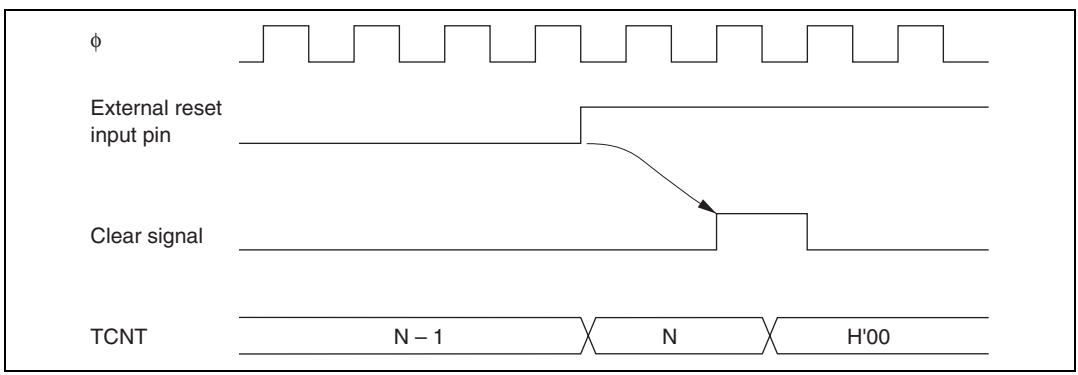


Figure 11.8 Timing of Clearance by External Reset

11.5.6 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 11.9 shows the timing of this operation.

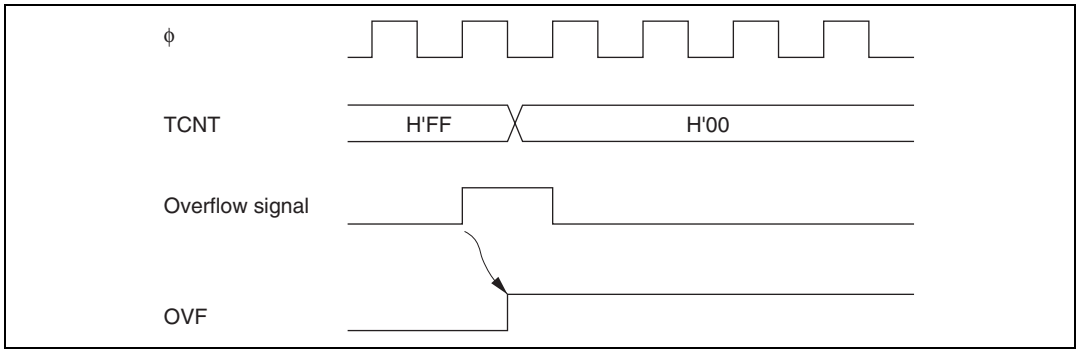


Figure 11.9 Timing of OVF Setting

11.6 Operation with Cascaded Connection

If bits CKS2 to CKS0 in either TCR_0 or TCR_1 are set to B'100, the 8-bit timers of the two channels are cascaded. With this configuration, 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 count mode). In this case, the timer operates as below.

11.6.1 16-Bit Counter Mode

When bits CKS2 to CKS0 in TCR_0 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 TCSR_0 is set to 1 when a 16-bit compare match event occurs.
  - The CMF flag in TCSR_1 is set to 1 when a lower 8-bit compare match event occurs.
- Counter clear specification
  - If the CCLR1 and CCLR0 bits in TCR_0 have been set for counter clear at compare match, the 16-bit counters (TCNT_0 and TCNT_1 together) are cleared when a 16-bit compare match event occurs. The 16-bit counters (TCNT0 and TCNT1 together) are cleared even if counter clear by the TMRI0 pin has also been set.
  - The settings of the CCLR1 and CCLR0 bits in TCR_1 are ignored. The lower 8 bits cannot be cleared independently.
- Pin output
  - Control of output from the TMO0 pin by bits OS3 to OS0 in TCSR_0 is in accordance with the 16-bit compare match conditions.
  - Control of output from the TMO1 pin by bits OS3 to OS0 in TCSR_1 is in accordance with the lower 8-bit compare match conditions.

11.6.2 Compare Match Count Mode

When bits CKS2 to CKS0 in TCR_1 are B'100, TCNT_1 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.

11.7 Interrupts

11.7.1 Interrupt Sources and DTC Activation

There are three 8-bit timer interrupt sources: CMIA, CMIB, and OVI. Their relative priorities are shown in table 11.3. Each interrupt source is set as enabled or disabled by the corresponding interrupt enable bit in TCR or TCSR, 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 11.3 8-Bit Timer Interrupt Sources

Channel	Name	Interrupt Source	Interrupt Flag	DTC Activation	Priority*
0	CMIA0	TCORA_0 compare match	CMFA	Possible	High
	CMIB0	TCORB_0 compare match	CMFB	Possible	
	OVI0	TCNT_0 overflow	OVF	Not possible	
1	CMIA1	TCORA_1 compare match	CMFA	Possible	
	CMIB1	TCORB_1 compare match	CMFB	Possible	
	OVI1	TCNT_1 overflow	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.

11.7.2 A/D Converter Activation

The A/D converter can be activated only by TMR_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 TMR_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.

11.8 Usage Notes

11.8.1 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 11.10 shows this operation.

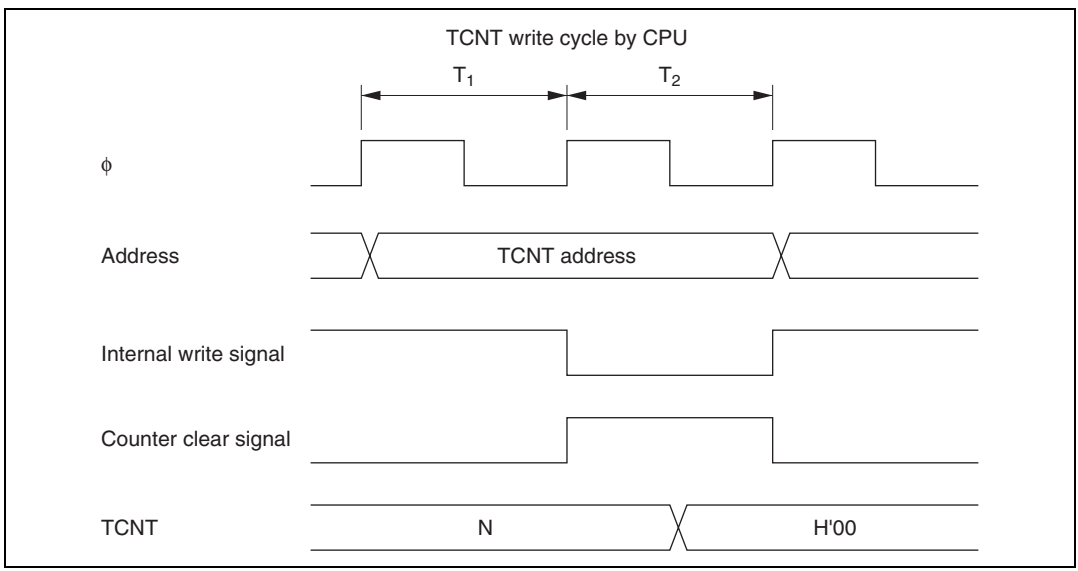


Figure 11.10 Contention between TCNT Write and Clear

11.8.2 Contention between 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 counter is not incremented. Figure 11.11 shows this operation.

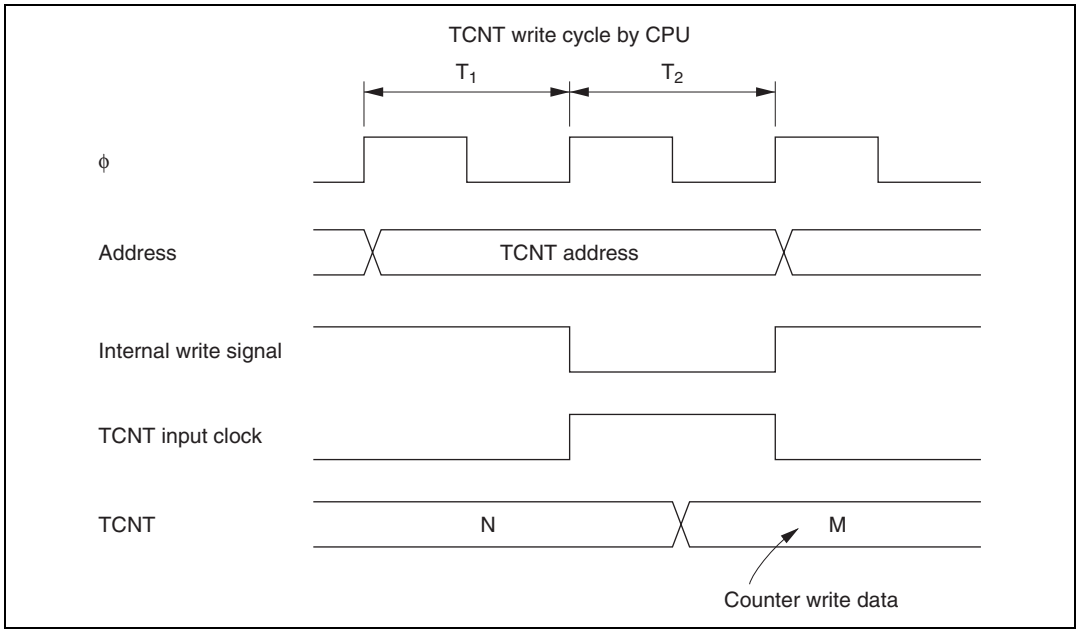


Figure 11.11 Contention between TCNT Write and Increment

11.8.3 Contention between TCOR Write and Compare Match

During the T_2 state of a TCOR write cycle, the TCOR write has priority and the compare match signal is prohibited even if a compare match event occurs. Figure 11.12 shows this operation.

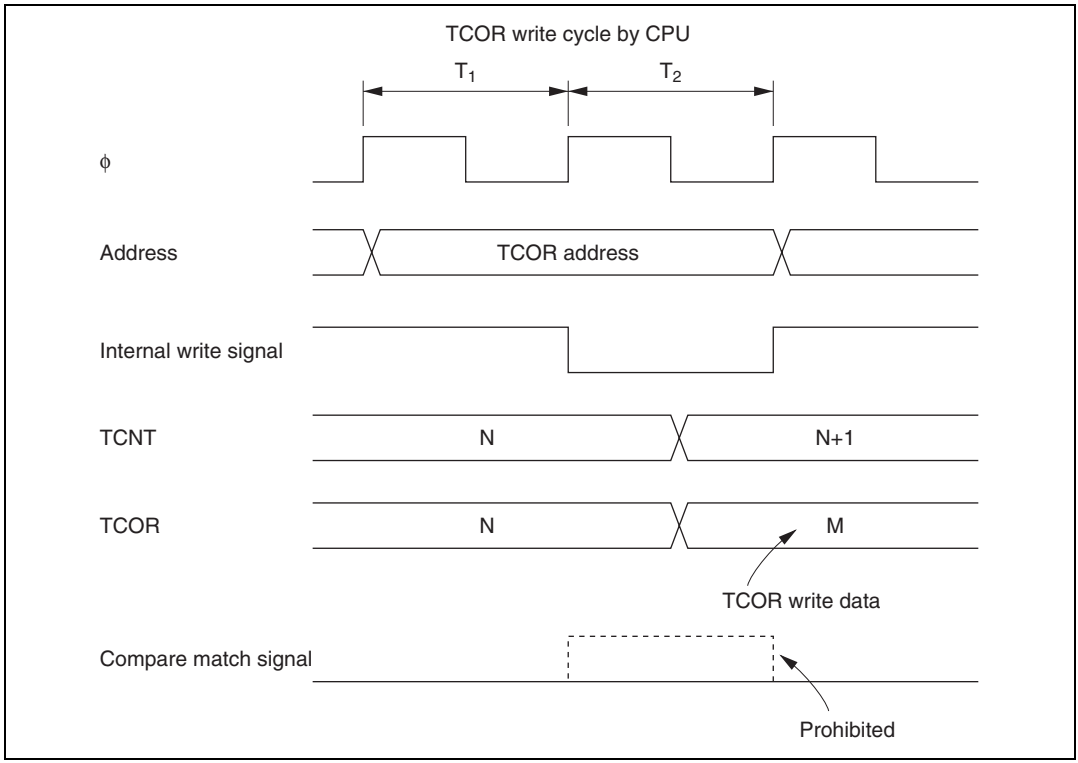


Figure 11.12 Contention between TCOR Write and Compare Match

11.8.4 Contention between Compare Matches A and B

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 11.4.

Table 11.4 Timer Output Priorities

Output Setting	Priority
Toggle output	High
1 output	↑
0 output	
No change	Low

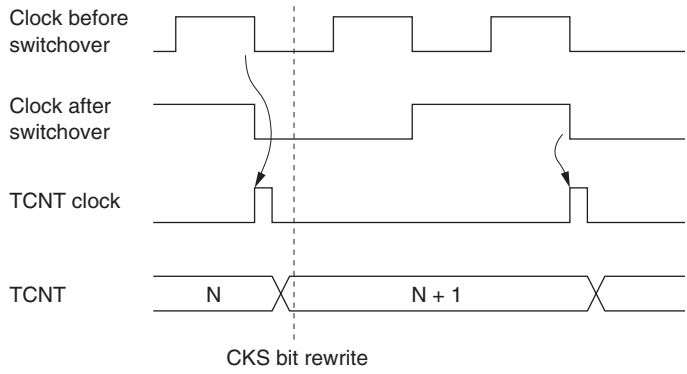
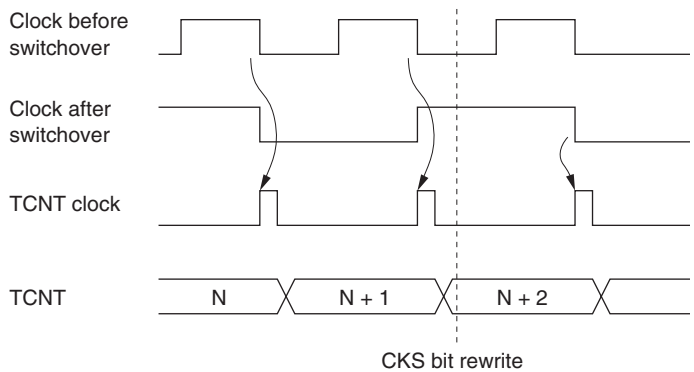
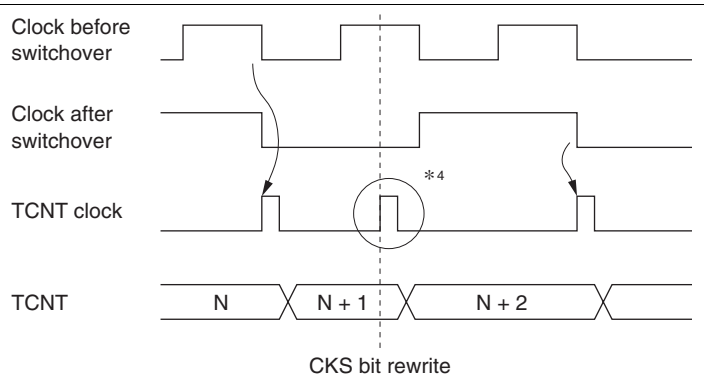
11.8.5 Switching of Internal Clocks and TCNT Operation

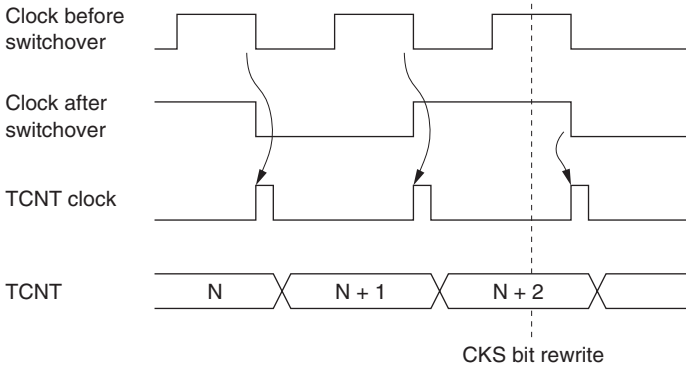
TCNT may increment erroneously when the internal clock is switched over. Table 11.5 shows the relationship between the timing at which the internal clock is switched (by writing to the CKS1 and CKS0 bits) and the TCNT operation.

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 11.5, 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.

Table 11.5 Switching of Internal Clock and TCNT Operation

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.	Timing of Switchover by Means of CKS1 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.

11.8.6 Mode Setting with Cascaded Connection

If 16-bit counter mode and compare match count mode are specified at the same time, input clocks for TCNT_0 and TCNT_1 are not generated, and the counter stops. Do not specify 16-bit counter and compare match count modes simultaneously.

11.8.7 Module Stop Mode Setting

Operation of the TMR can be disabled or enabled using the module stop control register. The initial setting is for operation of the TMR to be halted. Register access is enabled by clearing module stop mode. For details, refer to section 22, Power-Down Modes.

Section 12 Watchdog Timer (WDT)

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

The block diagram of the WDT is shown in figure 12.1.

12.1 Features

- Selectable from eight counter input clocks
- Switchable between watchdog timer mode and interval timer mode

In watchdog timer mode

- If the counter overflows, it is possible to select whether this LSI is internally reset or not.

In interval timer mode

- If the counter overflows, the WDT generates an interval timer interrupt (WOVI).

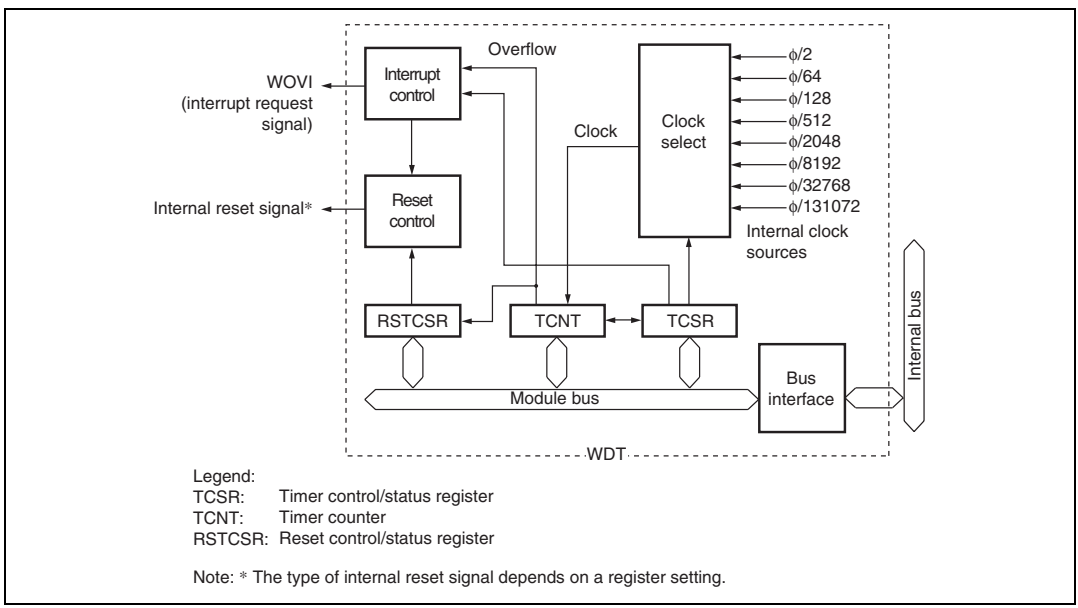


Figure 12.1 Block Diagram of WDT

12.2 Register Descriptions

The WDT has the following three registers. For details, refer to section 23, List of Registers. To prevent accidental overwriting, TCSR, TCNT, and RSTCSR have to be written to by a different method to normal registers. For details, refer to section 12.5.1, Notes on Register Access.

- Timer counter (TCNT)
- Timer control/status register (TCSR)
- Reset control/status register (RSTCSR)

12.2.1 Timer Counter (TCNT)

TCNT is an 8-bit readable/writable up-counter. TCNT is initialized to H'00 by a reset, when the TME bit in TCSR is cleared to 0.

12.2.2 Timer Control/Status Register (TCSR)

TCSR is an 8-bit readable/writable register. Its functions include selecting the clock source to be input to TCNT, and selecting the timer mode.

Bit	Bit Name	Initial Value	R/W	Description
7	OVF	0	R/(W)*	Overflow Flag Indicates that TCNT has overflowed. Only a write of 0 is permitted, to clear the flag. [Setting condition] • When TCNT overflows (changes from H'FF to H'00) When internal reset request generation is selected in watchdog timer mode, OVF is cleared automatically by the internal reset. [Clearing condition] • Cleared by reading TCSR when OVF = 1, then writing 0 to OVF When polling CVF when the interval timer interrupt has been prohibited, OVF = 1 status should be read two or more times.

Bit	Bit Name	Initial Value	R/W	Description
6	WT/$\overline{IT}$	0	R/W	Timer Mode Select Selects whether the WDT is used as a watchdog timer or interval timer. 0: Interval timer mode 1: Watchdog timer mode
5	TME	0	R/W	Timer Enable When this bit is set to 1, TCNT starts counting. When this bit is cleared, TCNT stops counting and is initialized to H'00.
4, 3	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
2	CKS2	0	R/W	Clock Select 2 to 0 Selects the clock source to be input to TCNT. The overflow frequency for ϕ = 16 MHz is enclosed in parentheses. 000: Clock ϕ/2 (frequency: 32.0 μs) 001: Clock ϕ/64 (frequency: 1.0 ms) 010: Clock ϕ/128 (frequency: 2.0 ms) 011: Clock ϕ/512 (frequency: 8.2 ms) 100: Clock ϕ/2048 (frequency: 32.8 ms) 101: Clock ϕ/8192 (frequency: 131.1 ms) 110: Clock ϕ/32768 (frequency: 524.3 ms) 111: Clock ϕ/131072 (frequency: 2.1 s)
1	CKS1	0	R/W	
0	CKS0	0	R/W	

Note: * The write value should always be 0 to clear this flag.

12.2.3 Reset Control/Status Register (RSTCSR)

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, and not by the WDT internal reset signal caused by overflows.

Bit	Bit Name	Initial Value	R/W	Description
7	WOVF	0	R/(W)*	Watchdog Overflow Flag This bit is set when TCNT overflows in watchdog timer mode. This bit cannot be set in interval timer mode, and the write value should always be 0. [Setting condition] • Set when TCNT overflows (changed from H'FF to H'00) in watchdog timer mode [Clearing condition] • Cleared by reading RSTCSR when WOVF = 1, and then writing 0 to WOVF
6	RSTE	0	R/W	Reset Enable Specifies whether or not a reset signal is generated in the chip if TCNT overflows during watchdog timer operation. 0: Reset signal is not generated even if TCNT overflows (Though this LSI is not reset, TCNT and TCSR in WDT are reset) 1: Reset signal is generated if TCNT overflows
5	RSTS	0	R/W	Reset Select Selects the type of internal reset generated if TCNT overflows during watchdog timer operation. 0: Power-on reset 1: Setting prohibited
4 to 0	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.

Note: * The write value should always be 0 to clear this flag.

12.3 Operation

12.3.1 Watchdog Timer Mode

To use the WDT as a watchdog timer, set the WT/$\overline{IT}$ bit in TCSR and the TME bit to 1.

TCNT does not overflow while the system is operating normally. Software must prevent TCNT overflows by rewriting the TCNT value (normally be writing H'00) before overflows occurs.

When the RSTE bit of the RSTCSR is set to 1, and if the TCNT overflows, an internal reset signal for this LSI is issued. In this case, select power-on reset or manual reset by setting the RSTS bit of the RSTCSR to 0.

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. The internal reset signal is output for 518 states.

When the TCNT overflows in watchdog timer mode, the WOVF bit of the RSTCSR is set to 1. If the RSTE bit of the RSTCSR has been set to 1, an internal reset signal for the entire LSI is generated at TCNT overflow.

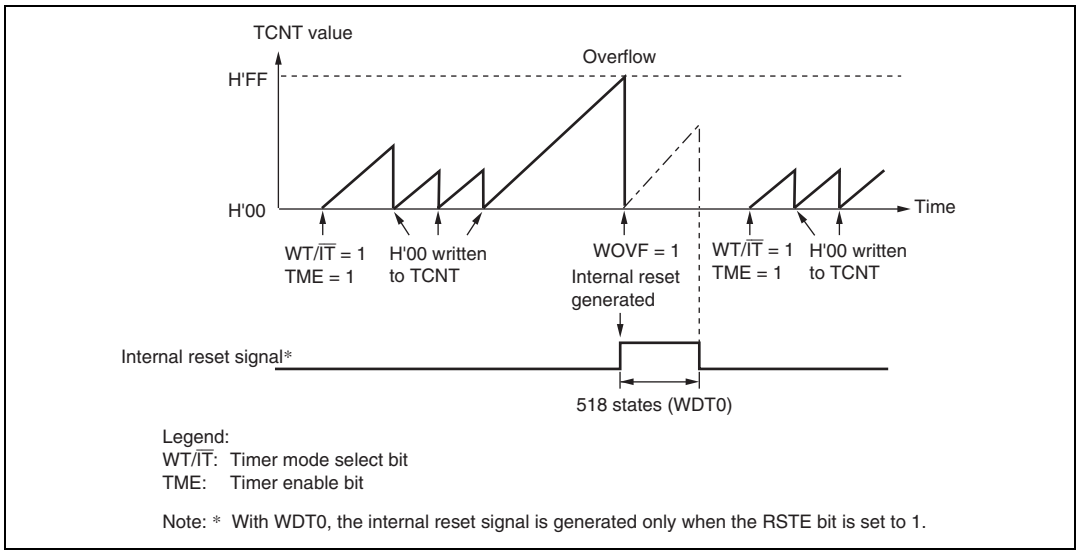


Figure 12.2 Operation in Watchdog Timer Mode

12.3.2 Timing of Setting of Watchdog Timer Overflow Flag (WOVF)

With WDT0, the WOVF bit in RSTCSR is set to 1 if TCNT overflows in watchdog timer mode. If TCNT overflows while the RSTE bit in RSTCSR is set to 1, an internal reset signal is generated for the entire chip. This timing is illustrated in figure 12.3.

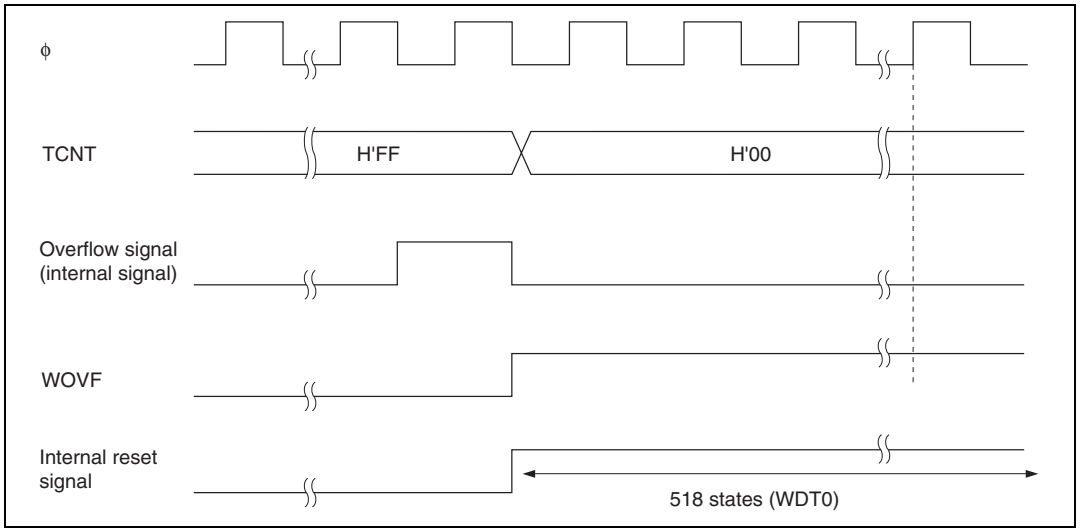


Figure 12.3 Timing of WOVF Setting

12.3.3 Interval Timer Mode

To use the WDT as an interval timer, clear bit WT/$\overline{IT}$ in TCSR to 0 and set bit TME to 1. When the interval timer is operating, an interval timer interrupt (WOVI) is generated each time the TCNT overflows. Therefore, an interrupt can be generated at intervals.

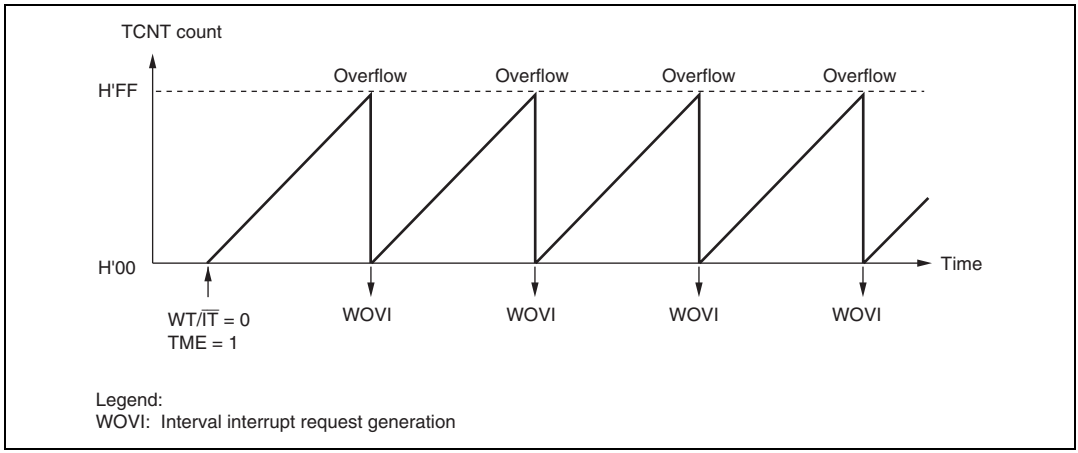


Figure 12.4 Operation in Interval Timer Mode

12.3.4 Timing of Setting of Overflow Flag (OVF)

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 12.5.

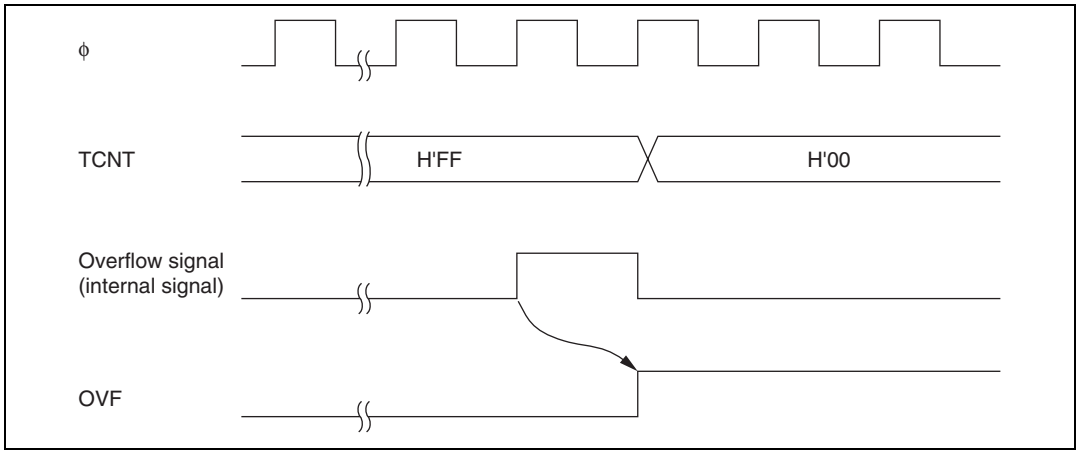


Figure 12.5 Timing of OVF Setting

12.4 Interrupts

During interval timer mode operation, an overflow generates an interval timer interrupt (WOVI). The interval timer interrupt is requested whenever the OVF flag is set to 1 in TCSR. OVF must be cleared to 0 in the interrupt handling routine.

Table 12.1 WDT Interrupt Source

Name	Interrupt Source	Interrupt Flag	DTC Activation
WOVI	TCNT overflow	WOVF	Impossible

12.5 Usage Notes

12.5.1 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 transfer instructions. Figure 12.6 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.

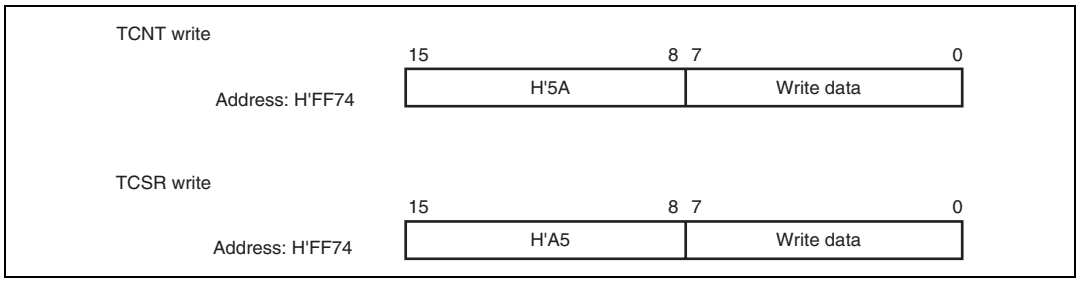


Figure 12.6 Format of Data Written to TCNT and TCSR

Writing to RSTCSR: RSTCSR must be written to by a word transfer to address H'FF76. It cannot be written to with byte instructions. Figure 12.7 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 and RSTS bits.

To write 0 to the WOVF bit, the upper byte of the written word must contain H'A5 and the lower byte must contain H'00. This clears the WOVF bit to 0, but has no effect on the RSTE and RSTS bits. To write to the RSTE and RSTS bits, the upper byte must contain H'5A and the lower byte must contain the write data. This writes the values in bits 6 and 5 of the lower byte into the RSTE and RSTS bits, but has no effect on the WOVF bit.

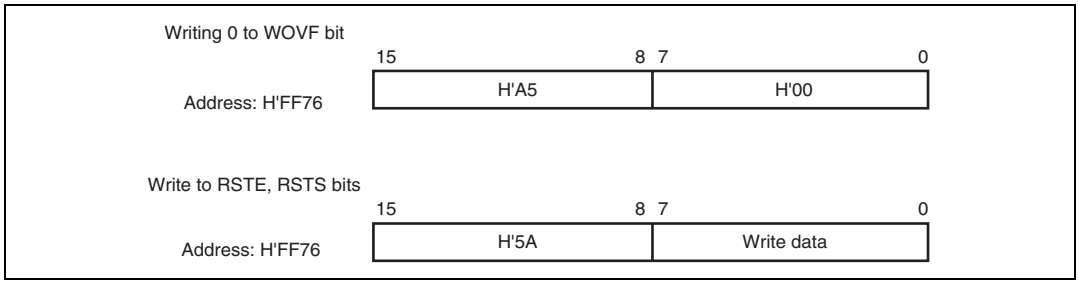


Figure 12.7 Format of Data Written to RSTCSR (Example of WDT0)

Reading from TCNT, TCSR, and RSTCSR: TCNT, TCSR, and RSTCSR are read by using the same method as for the general registers. TCSR, TCNT, and RSTCSR are allocated in addresses H'FF74, H'FF75, and H'FF77 respectively.

12.5.2 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 12.8 shows this operation.

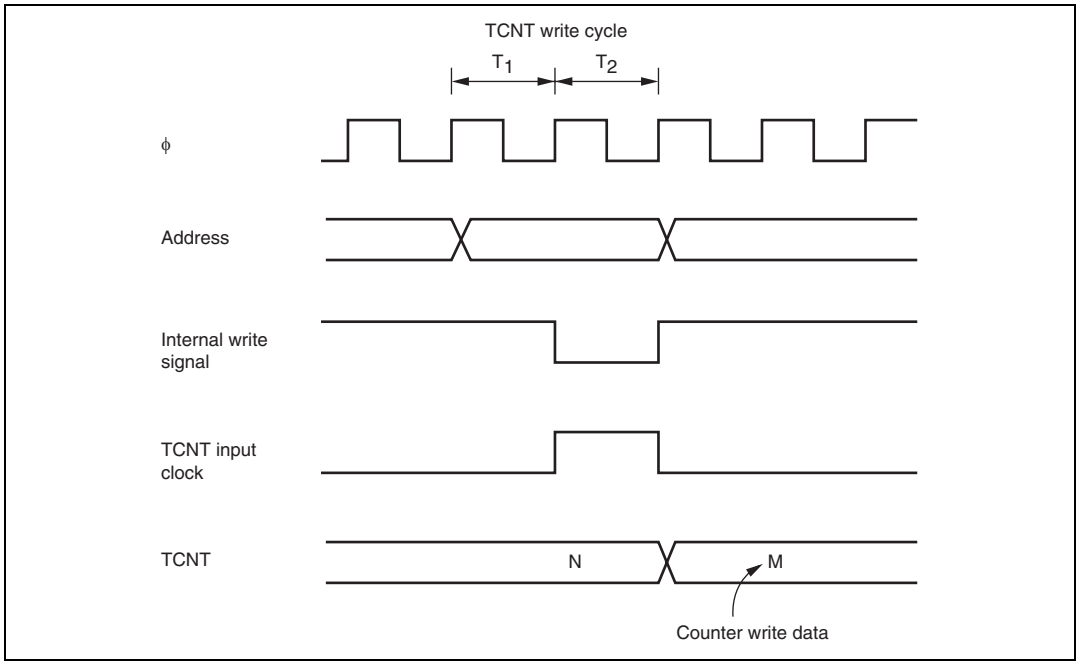


Figure 12.8 Contention between TCNT Write and Increment

12.5.3 Changing Value of CKS2 to CKS0

If bits CKS0 to CKS2 in TCSR are written to while the WDT is operating, errors could occur in the incrementation. Software must be used to stop the watchdog timer (by clearing the TME bit to 0) before changing the value of bits CKS0 to CKS2.

12.5.4 Switching between Watchdog Timer Mode and Interval Timer Mode

If the mode is switched from watchdog timer to interval timer while the WDT is operating, errors could occur in the incrementation. Software must be used to stop the watchdog timer (by clearing the TME bit to 0) before switching the mode.

12.5.5 Internal Reset in Watchdog Timer Mode

This LSI is not reset internally if TCNT overflows while the RSTE bit is cleared to 0 during watchdog timer operation, however TCNT and TCSR of the WDT are reset.

TCNT, TCSR, or RSTCR cannot be written to for 132 states following an overflow. During this period, any attempt to read the WOVF flag is not acknowledged. Accordingly, wait 132 states after overflow to write 0 to the WOVF flag for clearing.

12.5.6 OVF Flag Clearing in Interval Timer Mode

When the OVF flag setting conflicts with the OVF flag reading in interval timer mode, writing 0 to the OVF flag may not clear the flag even though the OVF flag has been read while it is 1. If there is a possibility that the OVF flag setting and reading will conflict, such as when the OVF flag is polled with the interval timer interrupt disabled, read the OVF flag while it is 1 at least twice before writing 0 to the OVF flag to clear the flag.

Section 13 Serial Communication Interface

This LSI has three independent serial communication interface (SCI) channels. The SCI can handle both asynchronous and clocked synchronous serial communication. Asynchronous serial data communication can be carried out using standard asynchronous communication chips such as a Universal Asynchronous Receiver/Transmitter (UART) or an Asynchronous Communication Interface Adapter (ACIA). The SCI also supports the smart card (IC card) interface based on ISO/IEC 7816-3 (Identification Card) as an enhanced asynchronous communication function.

13.1 Features

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

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

  Double-buffering is used in both the transmitter and the receiver, enabling continuous transmission and continuous reception of serial data.
    - On-chip baud rate generator allows any bit rate to be selected

  External clock can be selected as a transfer clock source (except for in Smart Card interface mode).
- Choice of LSB-first or MSB-first transfer (except in the case of asynchronous mode 7-bit data)
- Four interrupt sources

  Transmit-end, transmit-data-empty, receive-data-full, and receive error — that can issue requests.

  The transmit-data-empty interrupt and receive data full interrupts can be used to activate the DMA controller (DMAC) or the Data Transfer Controller (DTC).
- Module stop mode can be set

Asynchronous Mode

- Data length: 7 or 8 bits
- Stop bit length: 1 or 2 bits
- Parity: Even, odd, or none
- Receive error detection: Parity, overrun, and framing errors
- Break detection: Break can be detected by reading the RxD pin level directly in the case of a framing error

- Average transfer rate generator (SCI_0):

  In H8S/2215 720 kbps, 460.784 kbps, or 115.196 kbps can be selected at 16 MHz.

  In H8S/2215R and H8S/2215T

  921.569 kbps, 720 kbps, 460.784 kbps, or 115.196 kbps can be selected at 16 MHz.

  921.053 kbps, 720 kbps, 460.526 kbps, or 115.132 kbps can be selected at 24 MHz.
- A transfer rate clock can be input from the TPU (SCI_0)
- A multiprocessor communication function is provided that enables serial data communication with a number of processors

Clocked Synchronous Mode

- Data length: 8 bits
- Receive error detection: Overrun errors detected
- SCI select function (SCI_0): TxD0 = high-impedance and SCK0 = fixed high-level input can selected when $\overline{\text{IRQ7}}$ = 1)
- Serial data communication can be carried out with other chips that have a synchronous communication function

Smart Card Interface

- An error signal can be automatically transmitted on detection of a parity error during reception
- Data can be automatically re-transmitted on detection of a error signal during transmission
- Both direct convention and inverse convention are supported

13.1.1 Block Diagram

Figure 13.1 shows the block diagram of the SCI_0 for H8S/2215, figure 13.2 shows the block diagram of the SCI_0 for H8S/2215R and H8S/2215T. Figure 13.2 shows the block diagram of the SCI_1 and SCI_2.

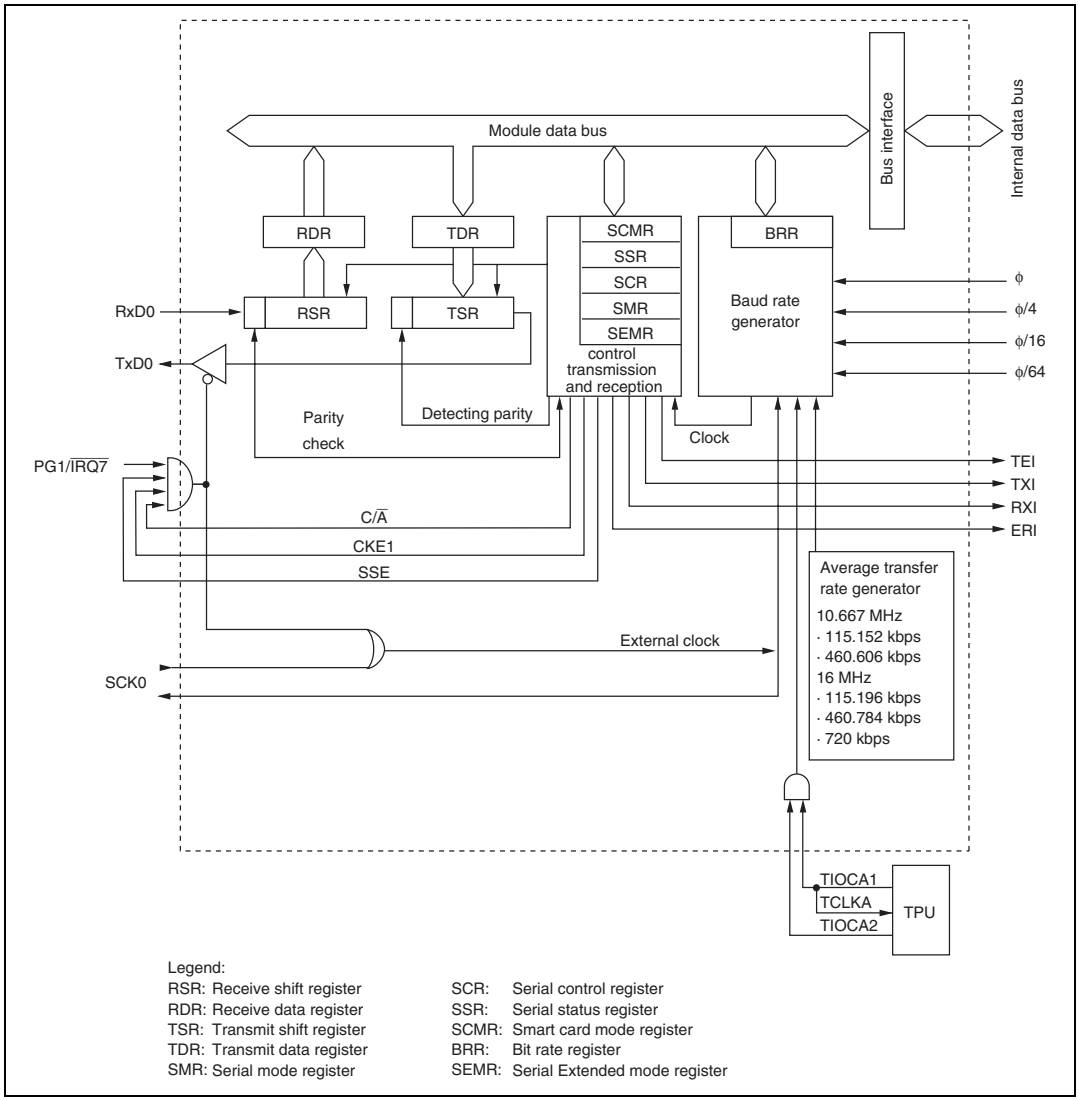


Figure 13.1 Block Diagram of SCI_0 (H8S/2215)

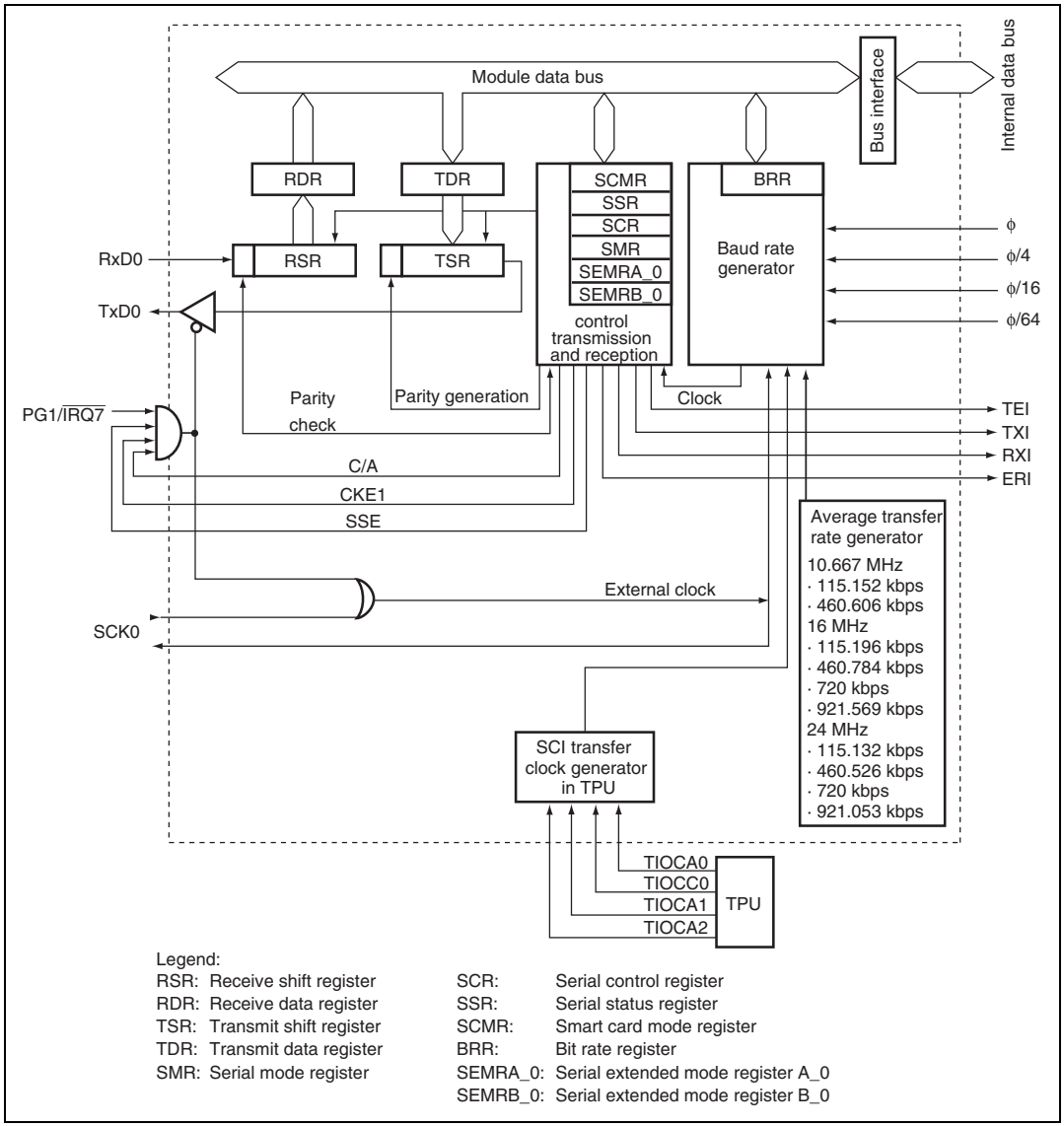


Figure 13.2 Block Diagram of SCI_0 (H8S/2215R and H8S/2215T)

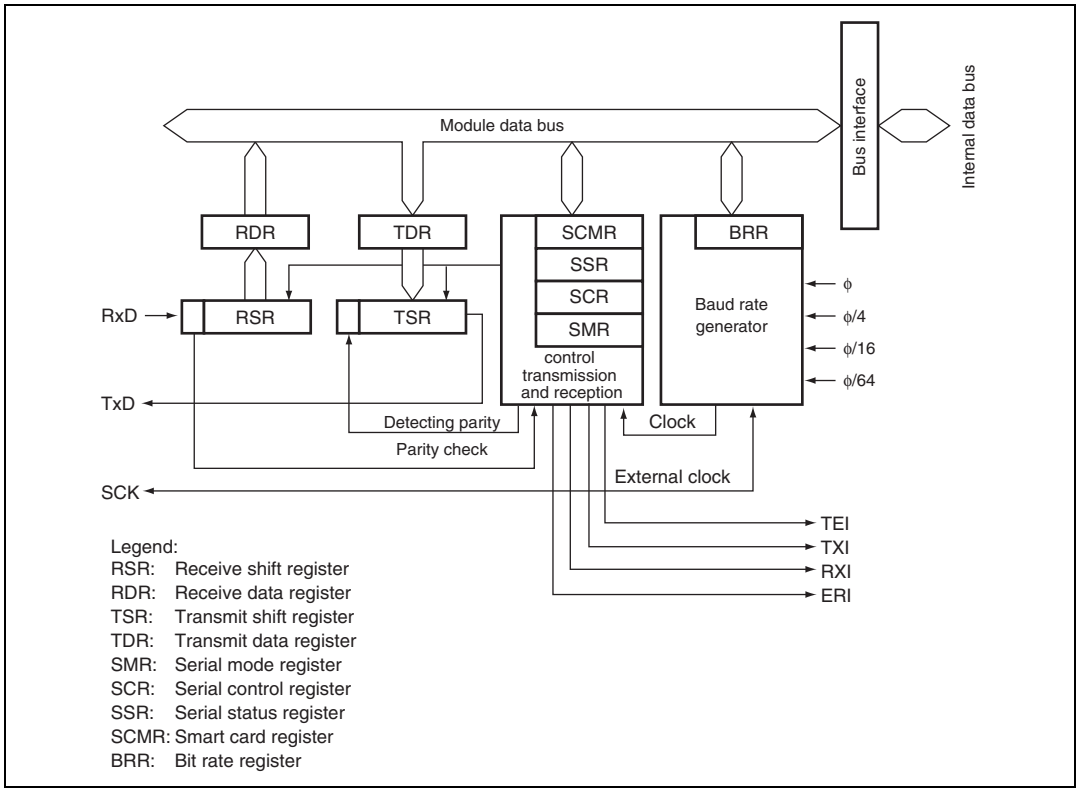


Figure 13.3 Block Diagram of SCI_1 and SCI_2

13.2 Input/Output Pins

Table 13.1 shows the serial pins for each SCI channel.

Table 13.1 Pin Configuration

Channel	Pin Name*	I/O	Function
0	SCK0	I/O	SCI_0 clock input/output
	RxD0	Input	SCI_0 receive data input
	TxD0	Output	SCI_0 transmit data output
1	SCK1	I/O	SCI_1 clock input/output
	RxD1	Input	SCI_1 receive data input
	TxD1	Output	SCI_1 transmit data output
2	SCK2	I/O	SCI_2 clock input/output
	RxD2	Input	SCI_2 receive data input
	TxD2	Output	SCI_2 transmit data output

Note: * Pin names SCK, RxD, and TxD are used in the text for all channels, omitting the channel designation.

13.3 Register Descriptions

The SCI has the following registers for each channel. Some bits in the serial mode register (SMR), serial status register (SSR), and serial control register (SCR) have different functions in different modes—normal serial communication interface mode and smart card interface mode; therefore, the bits are described separately for each mode in the corresponding register sections.

- Receive shift register (RSR)
- Receive data register (RDR)
- Transmit data register (TDR)
- Transmit shift register (TSR)
- Serial mode register (SMR)
- Serial control register (SCR)
- Serial status register (SSR)
- Smart card mode register (SCMR)
- Serial extended mode register (SEMR) (only for channel 0 in H8S/2215)
- Serial extended mode register A_0 (SEMRA_0) (only for channel 0 in H8S/2215R and H8S/2215T)

- Serial extended mode register B_0 (SEMRB_0) (only for channel 0 in H8S/2215R and H8S/2215T)
- Bit rate register (BRR)

13.3.1 Receive Shift Register (RSR)

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

13.3.2 Receive Data Register (RDR)

RDR is an 8-bit register that stores received 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. After this, RSR is receive-enabled. As RSR and RDR function as a double buffer in this way, continuous receive operations are possible. After confirming that the RDRF bit in SSR is set to 1, read RDR only once. RDR cannot be written to by the CPU.

13.3.3 Transmit Data Register (TDR)

TDR is an 8-bit register that stores data for transmission. When the SCI detects that TSR is empty, it transfers the transmit data written in TDR to TSR and starts transmission. The double-buffered structure of TDR and TSR enables continuous serial transmission. If the next transmit data has already been written to TDR during serial transmission, the SCI transfers the written data to TSR to continue transmission. Although TDR can be read from or written to by the CPU at all times, to achieve reliable serial transmission, write transmit data to TDR only once after confirming that the TDRE bit in SSR is set to 1.

13.3.4 Transmit Shift Register (TSR)

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

13.3.5 Serial Mode Register (SMR)

SMR is used to set the SCI's serial transfer format and select the baud rate generator clock source. Some bits in SMR have different functions in normal mode and smart card interface mode.

- Normal Serial Communication Interface Mode (When SMIF in SCMR is 0)

Bit	Bit Name	Initial Value	R/W	Description
7	C/$\overline{A}$	0	R/W	Communication Mode 0: Asynchronous mode 1: Clocked synchronous mode
6	CHR	0	R/W	Character Length (enabled only in asynchronous mode) 0: Selects 8 bits as the data length 1: Selects 7 bits as the data length. LSB-first is fixed and the MSB of TDR is not transmitted in transmission In clocked synchronous mode, a fixed data length of 8 bits is used.
5	PE	0	R/W	Parity Enable (enabled only in asynchronous mode) When this bit is set to 1, the parity bit is added to transmit data before transmission, and the parity bit is checked in reception. For a multiprocessor format, parity bit addition and checking are not performed regardless of the PE bit setting.
4	O/$\overline{E}$	0	R/W	Parity Mode (enabled only when the PE bit is 1 in asynchronous mode) 0: Selects even parity 1: Selects odd parity
3	STOP	0	R/W	Stop Bit Length (enabled only in asynchronous mode) Selects the stop bit length in transmission. 0: 1 stop bit 1: 2 stop bits In reception, only the first stop bit is checked. If the second stop bit is 0, it is treated as the start bit of the next transmit character.

Bit	Bit Name	Initial Value	R/W	Description
2	MP	0	R/W	Multiprocessor Mode (enabled only in asynchronous mode) When this bit is set to 1, the multiprocessor communication function is enabled. The PE bit and O/$\overline{E}$ bit settings are invalid in multiprocessor mode. For details, see section 13.5, Multiprocessor Communication Function.
1	CKS1	0	R/W	Clock Select 0 and 1:
0	CKS0	0	R/W	These bits select the clock source for the baud rate generator. 00: ϕ clock (n = 0) 01: ϕ/4 clock (n = 1) 10: ϕ/16 clock (n = 2) 11: ϕ/64 clock (n = 3) For the relationship between the bit rate register setting and the baud rate, see section 13.3.12, Bit Rate Register (BRR). n is the decimal representation of the value of n in BRR (see section 13.3.12, Bit Rate Register (BRR)).

- Smart Card Interface Mode (When SMIF in SCMR is 1)

Bit	Bit Name	Initial Value	R/W	Description
7	GM	0	R/W	GSM Mode Setting this bit to 1 allows GSM mode operation. In GSM mode, the TEND set timing is put forward to 11.0 etu from the start and the clock output control function is appended. For details, see section 13.7.9, Clock Output Control. 0: Normal smart card interface mode operation (initial value) (1) The TEND flag is generated 12.5 etu (11.5 etu in the block transfer mode) after the beginning of the start bit. (2) Clock output on/off control only. 1: GSM mode operation in smart card interface mode (1) The TEND flag is generated 11.0 etu after the beginning of the start bit. (2) In addition to clock output on/off control, high/how fixed control is supported (set using SCR).
6	BLK	0	R/W	Setting this bit to 1 allows block transfer mode operation. For details, see section 13.7.4, Block Transfer Mode. 0: Normal smart card interface mode operation (initial value) (1) Error signal transmission, detection, and automatic data retransmission are performed. (2) The TXI interrupt is generated by the TEND flag. (3) The TEND flag is set 12.5 etu (11.0 etu in the GSM mode) after transmission starts. 1: Operation in block transfer mode (1) Error signal transmission, detection, and automatic data retransmission are not performed. (2) The TXI interrupt is generated by the TDRE flag. (3) The TEND flag is set 11.5 etu (11.0 etu in the GSM mode) after transmission starts.

Bit	Bit Name	Initial Value	R/W	Description
5	PE	0	R/W	Parity Enable When this bit is set to 1, the parity bit is added to transmit data before transmission, and the parity bit is checked in reception. Set this bit to 1 in smart card interface mode.
4	O/$\overline{E}$	0	R/W	Parity Mode (valid only when the PE bit is 1) 0: Selects even parity 1: Selects odd parity For details on the usage of this bit in smart card interface mode, see section 13.7.2, Data Format (Except for Block Transfer Mode).
3	BCP1	0	R/W	Basic Clock Pulse 1,0
2	BCP0	0	R/W	These bits select the number of basic clock cycles in a 1-bit data transfer time in smart card interface mode. 00: 32 clock cycles (S = 32) 01: 64 clock cycles (S = 64) 10: 372 clock cycles (S = 372) 11: 256 clock cycles (S = 256) For details, see section 13.7.5, Receive Data Sampling Timing and Reception Margin. S is described in section 13.3.12, Bit Rate Register (BRR).
1	CKS1	0	R/W	Clock Select 1,0
0	CKS0	0	R/W	These bits select the clock source for the baud rate generator. 00: ϕ clock (n = 0) 01: ϕ/4 clock (n = 1) 10: ϕ/16 clock (n = 2) 11: ϕ/64 clock (n = 3) For the relation between the bit rate register setting and the baud rate, see section 13.3.12, Bit Rate Register (BRR). n is the decimal display of the value of n in BRR (see section 13.3.12, Bit Rate Register (BRR)).

13.3.6 Serial Control Register (SCR)

SCR is a register that enables or disables SCI transfer operations and interrupt requests, and is also used to selection of the transfer clock source. For details on interrupt requests, refer to section 13.9, Interrupts. Some bits in SCR have different functions in normal mode and smart card interface mode.

- Normal Serial Communication Interface Mode (When SMIF in SCMR is 0)

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable When this bit is set to 1, the TXI interrupt request is enabled. TXI interrupt request cancellation can be performed by reading 1 from the TDRE flag, then clearing it to 0, or clearing the TIE bit to 0.
6	RIE	0	R/W	Receive Interrupt Enable When this bit is set to 1, RXI and ERI interrupt requests are enabled. 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 clearing the RIE bit to 0.
5	TE	0	R/W	Transmit Enable When this bit s set to 1, transmission is enabled. 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. The TDRE flag in SSR is fixed at 1 if transmission is disabled by clearing this bit to 0.
4	RE	0	R/W	Receive Enable When this bit is set to 1, reception is enabled. Serial reception is started in this state when a start bit is detected in asynchronous mode or serial clock input is detected in clocked synchronous mode. SMR setting must be performed to decide the transfer format before setting the RE bit to 1. Clearing the RE bit to 0 does not affect the RDRF, FER, PER, and ORER flags, which retain their states.

Bit	Bit Name	Initial Value	R/W	Description
3	MPIE	0	R/W	Multiprocessor Interrupt Enable (enabled only when the MP bit in SMR is 1 in asynchronous mode) When this bit is set to 1, receive data in which the multiprocessor bit is 0 is skipped, and setting of the RDRF, FER, and ORER status flags in SSR is prohibited. On receiving data in which the multiprocessor bit is 1, this bit is automatically cleared and normal reception is resumed. For details, refer to section 13.5, Multiprocessor Communication Function. 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.
2	TEIE	0	R/W	Transmit End Interrupt Enable This bit is set to 1, TEI interrupt request is enabled. 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 clearing the TEIE bit to 0.
1	CKE1	0	R/W	Clock Enable 0 and 1
0	CKE0	0	R/W	Selects the clock source and SCK pin function. Asynchronous mode 00: Internal baud rate generator SCK pin functions as I/O port 01: Internal baud rate generator Outputs a clock of the same frequency as the bit rate from the SCK pin. 1X: External clock Inputs a clock with a frequency 16 times the bit rate from the SCK pin. Clocked synchronous mode 0X: Internal clock (SCK pin functions as clock output) 1X: External clock (SCK pin functions as clock input)

Legend:
X: Don't care

- Smart Card Interface Mode (When SMIF in SCMR is 1)

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable When this bit is set to 1, TXI interrupt request is enabled. TXI interrupt request cancellation can be performed by reading 1 from the TDRE flag in SSR, then clearing it to 0, or clearing the TIE bit to 0.
6	RIE	0	R/W	Receive Interrupt Enable When this bit is set to 1, RXI and ERI interrupt requests are enabled. RXI and ERI interrupt request cancellation can be performed by reading 1 from the RDRF, FER, PER, or ORER flag in SSR, then clearing the flag to 0, or clearing the RIE bit to 0.
5	TE	0	R/W	Transmit Enable When this bit s set to 1, transmission is enabled. 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. When this bit is cleared to 0, the transmission operation is disabled, and the TDRE flag is fixed at 1.
4	RE	0	R/W	Receive Enable When this bit is set to 1, reception is enabled. Serial reception is started in this state when a start bit is detected in asynchronous mode or serial clock input is detected in clocked synchronous mode. SMR setting must be performed to decide the reception format before setting the RE bit to 1. Clearing the RE bit to 0 does not affect the RDRF, FER, PER, and ORER flags, which retain their states.

Bit	Bit Name	Initial Value	R/W	Description
3	MPIE	0	R/W	Multiprocessor Interrupt Enable (enabled only when the MP bit in SMR is 1 in asynchronous mode) Write 0 to this bit in Smart Card interface mode. When receive data including MPB = 0 is received, receive data transfer from RSR to RDR, receive error detection, and setting of the RERF, FER, and ORER flags in SSR, are 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 are enabled.
2	TEIE	0	R/W	Transmit End Interrupt Enable Write 0 to this bit in Smart Card interface mode. 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 clearing the TEIE bit to 0.
1	CKE1	0	R/W	Clock Enable 0 and 1
0	CKE0	0		Enables or disables clock output from the SCK pin. The clock output can be dynamically switched in GSM mode. For details, refer to section 13.7.9, Clock Output Control. When the GM bit in SMR is 0: 00: Output disabled (SCK pin can be used as an I/O port pin) 01: Clock output 1X: Reserved When the GM bit in SMR is 1: 00: Output fixed low 01: Clock output 10: Output fixed high 11: Clock output

Legend:

X: Don't care

13.3.7 Serial Status Register (SSR)

SSR is a register containing status flags of the SCI and multiprocessor bits for transfer. 1 cannot be written to flags TDRE, RDRF, ORER, PER, and FER; they can only be cleared. Some bits in SSR have different functions in normal mode and smart card interface mode.

- Normal Serial Communication Interface Mode (When SMIF in SCMR is 0)

Bit	Bit Name	Initial Value	R/W	Description
7	TDRE	1	R/(W)*1	Transmit Data Register Empty Displays whether TDR contains transmit data. [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 [Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DMAC or the DTC*2 is activated by a TXI interrupt request and writes data to TDR
6	RDRF	0	R/(W)*1	Receive Data Register Full Indicates that the received data is stored in RDR. [Setting condition] • When serial reception ends normally and receive data is transferred from RSR to RDR [Clearing conditions] • When 0 is written to RDRF after reading RDRF = 1 • When the DMAC or the DTC*2 is activated by an RXI interrupt and transferred data from RDR RDR and the RDRF flag are not affected and retain their previous values when the RE bit in SCR is cleared to 0. The RDRF flag is not affected and retains their previous values when the RE bit in SCR is cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
5	ORER	0	R/(W)*1	Overrun Error [Setting condition] • When the next serial reception is completed while RDRF = 1 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 clocked synchronous mode, serial transmission cannot be continued, either. [Clearing condition] • When 0 is written to ORER after reading ORER = 1 The ORER flag is not affected and retains its previous state when the RE bit in SCR is cleared to 0.
4	FER	0	R/(W)*1	Framing Error [Setting condition] • When the stop bit is 0 In 2-stop-bit mode, only the first stop bit is checked for a value of 0; the second stop bits 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 clocked synchronous mode, serial transmission cannot be continued, either. [Clearing condition] • When 0 is written to FER after reading FER = 1 The FER flag is not affected and retains its previous state when the RE bit in SCR is cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
3	PER	0	R/(W)*1	Parity Error [Setting condition] • When a parity error is detected during reception 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 clocked synchronous mode, serial transmission cannot be continued, either. [Clearing condition] • When 0 is written to PER after reading PER = 1 The PER flag is not affected and retains its previous state when the RE bit in SCR is cleared to 0.
2	TEND	1	R	Transmit End [Setting conditions] • When the TE bit in SCR is 0 • When TDRE = 1 at transmission of the last bit of a 1-byte serial transmit character [Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DMAC or the DTC*2 is activated by a TXI interrupt and writes data to TDR
1	MPB	0	R	Multiprocessor Bit MPB stores the multiprocessor bit in the receive data. When the RE bit in SCR is cleared to 0 its previous state is retained. This bit retains its previous state when the RE bit in SCR is cleared to 0.
0	MPBT	0	R/W	Multiprocessor Bit Transfer MPBT stores the multiprocessor bit to be added to the transmit data.

Notes: 1. The write value should always be 0 to clear the flag.
2. The clearing conditions using the DTC are that DISEL bit be cleared to 0 and the transfer counter value be other than 0.

- Smart Card Interface Mode (When SMIF in SCMR is 1)

Bit	Bit Name	Initial Value	R/W	Description
7	TDRE	1	R/(W)*1	Transmit Data Register Empty Indicates whether TDR contains transmit data. [Setting conditions] • When the TE bit in SCR is 0 • When data is transferred from TDR to TSR and data can be written to TDR [Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DMAC or the DTC*2 is activated by a TXI interrupt request and writes data to TDR
6	RDRF	0	R/(W)*1	Receive Data Register Full Indicates that the received data is stored in RDR. [Setting condition] • When serial reception ends normally and receive data is transferred from RSR to RDR [Clearing conditions] • When 0 is written to RDRF after reading RDRF = 1 • When the DMAC or the DTC*2 is activated by an RXI interrupt and transferred data from RDR The RDRF flag is not affected and retains their previous values 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	Bit Name	Initial Value	R/W	Description
5	ORER	0	R/(W)*1	Overrun Error Indicates that an overrun error occurred during reception, causing abnormal termination. [Setting condition] • When the next serial reception is completed while RDRF = 1 The receive data prior to the overrun error is retained in RDR, and the data received subsequently is lost. Also, subsequent serial cannot be continued while the ORER flag is set to 1. In clocked synchronous mode, serial transmission cannot be continued, either. [Clearing condition] • When 0 is written to ORER after reading ORER = 1 The ORER flag is not affected and retains its previous state when the RE bit in SCR is cleared to 0.
4	ERS	0	R/(W)*1	Error Signal Status Indicates that the status of an error, signal 1 returned from the reception side at reception [Setting condition] • When the low level of the error signal is sampled [Clearing condition] • When 0 is written to ERS after reading ERS = 1 The ERS flag is not affected and retains its previous state when the RE bit in SCR is cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
3	PER	0	R/(W)*1	Parity Error Indicates that a parity error occurred during reception using parity addition in asynchronous mode, causing abnormal termination. [Setting condition] • When a parity error is detected during reception 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 clocked synchronous mode, serial transmission cannot be continued, either. [Clearing condition] • When 0 is written to PER after reading PER = 1 The PER flag is not affected and retains its previous state when the RE bit in SCR is cleared to 0.
2	TEND	1	R	Transmit End This bit is set to 1 when no error signal has been sent back from the receiving end and the next transmit data is ready to be transferred to TDR. [Setting conditions] • When the TE bit in SCR is 0 and the ERS bit is also 0 • When the ESR bit is 0 and the TDRE bit is 1 after the specified interval following transmission of 1-byte data. The timing of bit setting differs according to the register setting as follows: When GM = 0 and BLK = 0, 2.5 etu after transmission starts When GM = 0 and BLK = 1, 1.0 etu after transmission starts When GM = 1 and BLK = 0, 1.5 etu after transmission starts When GM = 1 and BLK = 1, 1.0 etu after transmission starts [Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DMAC or the DTC*2 is activated by a TXI interrupt and transfers transmission data to TDR

Bit	Bit Name	Initial Value	R/W	Description
1	MPB	0	R	Multiprocessor Bit This bit is not used in Smart Card interface mode.
0	MPBT	0	R/W	Multiprocessor Bit Transfer Write 0 to this bit in Smart Card interface mode.

Notes: 1. The write value should always be 0 to clear the flag.
2. The clearing conditions using the DTC are that DISEL bit be cleared to 0 and the transfer counter value be other than 0.

13.3.8 Smart Card Mode Register (SCMR)

SCMR selects the operation in smart card interface or the data Transfer formats.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	—	Reserved These bits are always read as 1.
3	DIR	0	R/W	Smart Card Data Transfer Direction Selects the serial/parallel conversion format. 0: LSB-first in transfer 1: MSB-first in transfer The bit setting is valid only when the transfer data format is 8 bits.
2	INV	0	R/W	Smart Card Data Invert Specifies inversion of the data logic level. The SINV bit does not affect the logic level of the parity bit. To invert the parity bit, invert the O/$\overline{E}$ bit in SMR. 0: TDR contents are transmitted as they are. Receive data is stored as it is in RDR 1: TDR contents are inverted before being transmitted. Receive data is stored in inverted form in RDR
1	—	1	—	Reserved This bit is always read as 1.
0	SMIF	0	R/W	Smart Card Interface Mode Select When this bit is set to 1, smart card interface mode is selected. 0: Normal asynchronous or clocked synchronous mode 1: Smart card interface mode

13.3.9 Serial Extended Mode Register (SEMR) (Only for Channel 0 in H8S/2215)

SEMR extends the functions of SCI_0. SEMR0 enables selection of the SCI_0 select function in synchronous mode, base clock setting in asynchronous mode, and also clock source selection and automatic transfer rate setting. Figure 13.3 shows an example of the internal base clock when an average transfer rate is selected and figure 13.4 shows as example of the setting when the TPU clock input is selected.

Bit	Bit Name	Initial Value	R/W	Description
7	SSE	0	R/W	SCI_0 Select Enable Allows selection of the SCI0 select function when an external clock is input in synchronous mode. The SSE setting is valid when external clock input is used (CKE1 = 1 in SCR) in synchronous mode (C/$\overline{A}$ = 1 in SMR). 0: SCI_0 select function disabled 1: SCI_0 select function enabled When the SCI_0 select function is enabled, if 1 is input to the PG1/$\overline{IRQ7}$ pin, TxD0 output goes to the high-impedance state, SCK0 input is fixed high.
6 to 4	—	Undefined	—	Reserved The write value should always be 0.
3	ABCS	0	R/W	Asynchronous Base Clock Select Selects the 1-bit-interval base clock in asynchronous mode. The ABCS setting is valid in asynchronous mode (C/$\overline{A}$ = 0 in SMR). 0: SCI_0 operates on base clock with frequency of 16 times transfer rate 1: SCI_0 operates on base clock with frequency of 8 times transfer rate

Bit	Bit Name	Initial Value	R/W	Description
2	ACS2	0	R/W	Asynchronous Clock Source Select 2 to 0
1	ACS1	0	R/W	These bits select the clock source in asynchronous mode. When an average transfer rate is selected, the base clock is set automatically regardless of the ABCS value. Note that average transfer rates are not supported for operating frequencies other than 10.667 MHz and 16 MHz. The setting in bits ACS2 to ACS0 is valid when external clock input is used (CKE1 = 1 in SCR) in asynchronous mode (C/A = 0 in SMR). Figures 13.3 and 13.4 show setting examples. 000: External clock input 001: 115.152 kbps average transfer rate (for ϕ = 10.667 MHz only) is selected* (SCI_0 operates on base clock with frequency of 16 times transfer rate) 010: 460.606 kbps average transfer rate (for ϕ = 10.667 MHz only) is selected* (SCI_0 operates on base clock with frequency of 8 times transfer rate) 011: Reserved 100: TPU clock input (AND of TIOCA1 and TIOCA2) The signal generated by TIOCA1 and TIOCA2, which are the compare match outputs for TPU_1 and TPU_2 or PWM outputs, is used as a base clock. Note that IRQ0 and IRQ1 cannot be used since TIOCA1 and TIOCA2 are used as outputs. The high pulse width for TIOCA1 should be its low pulse width or less. 101: 115.196 kbps average transfer rate (for ϕ = 16 MHz only) is selected (SCI_0 operates on base clock with frequency of 16 times transfer rate) 110: 460.784 kbps average transfer rate (for ϕ = 16 MHz only) is selected (SCI_0 operates on base clock with frequency of 16 times transfer rate) 111: 720 kbps average transfer rate (for ϕ = 16 MHz only) is selected (SCI_0 operates on base clock with frequency of 8 times transfer rate)
0	ACS0	0	R/W	

Note: * Cannot be used in this LSI because the operating frequency ϕ in this LSI is 13 MHz or greater.

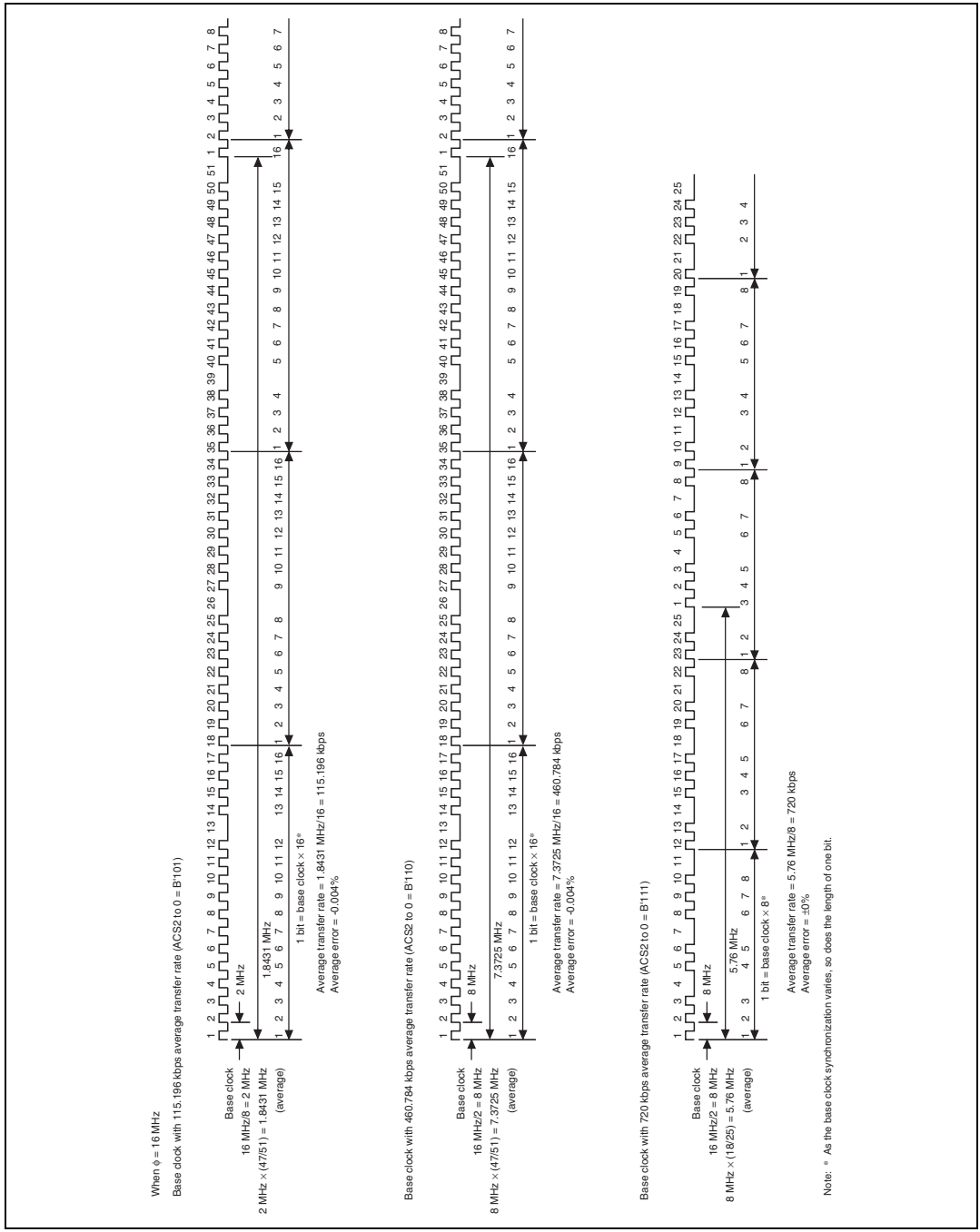


Figure 13.4 Examples of Base Clock when Average Transfer Rate Is Selected (1)

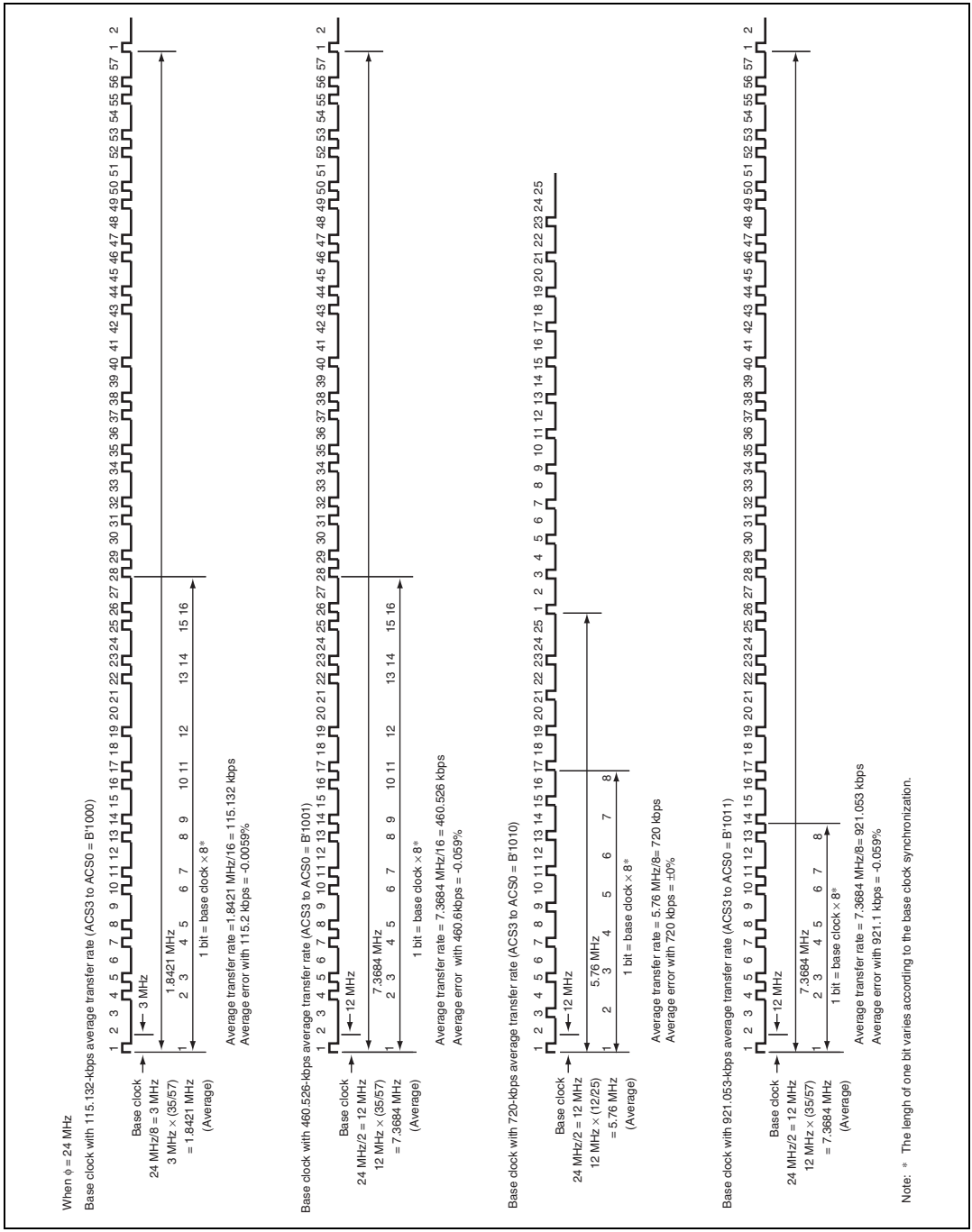


Figure 13.4 Examples of Base Clock when Average Transfer Rate Is Selected (2)

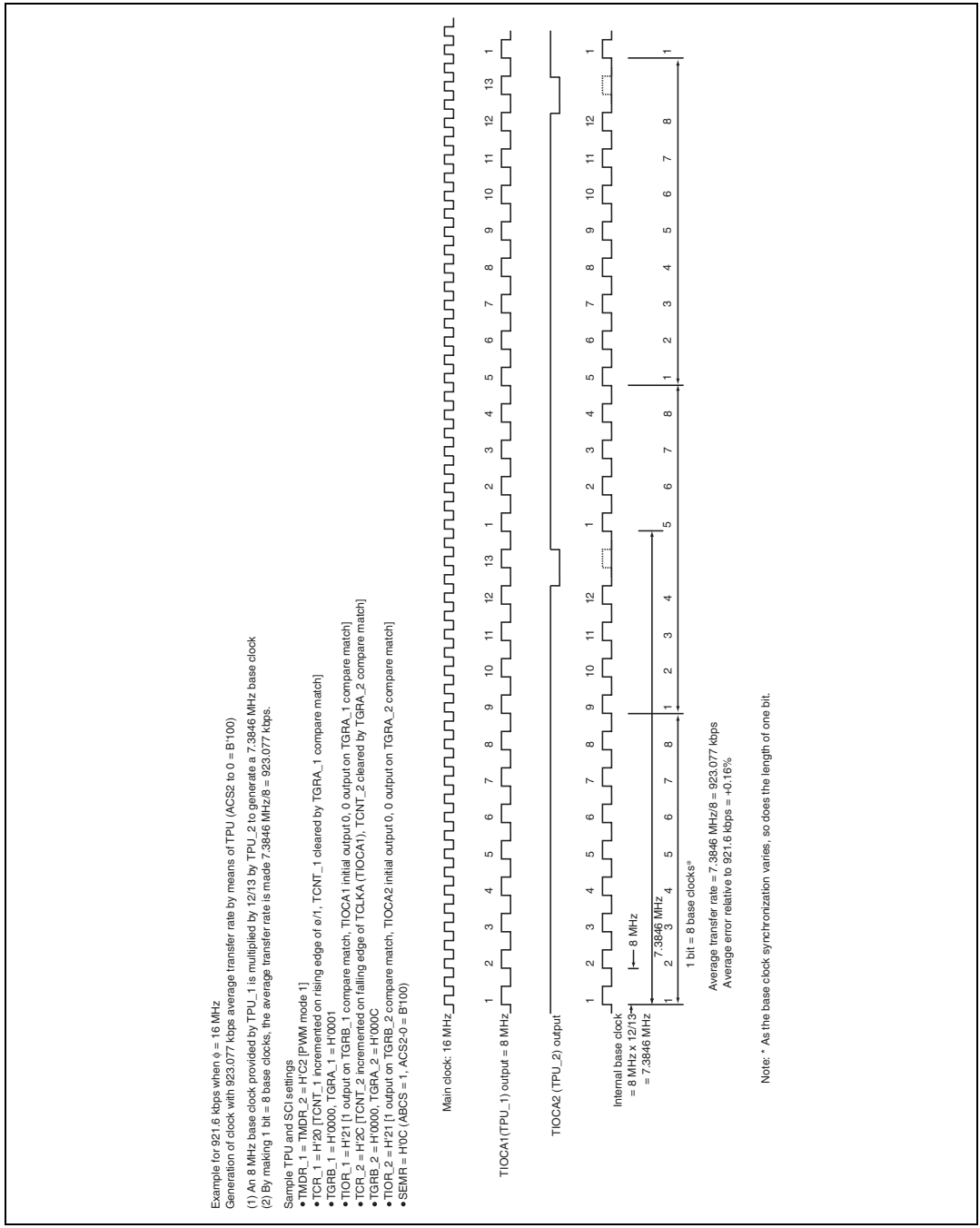


Figure 13.5 Example of Average Transfer Rate Setting when TPU Clock Is Input (1)

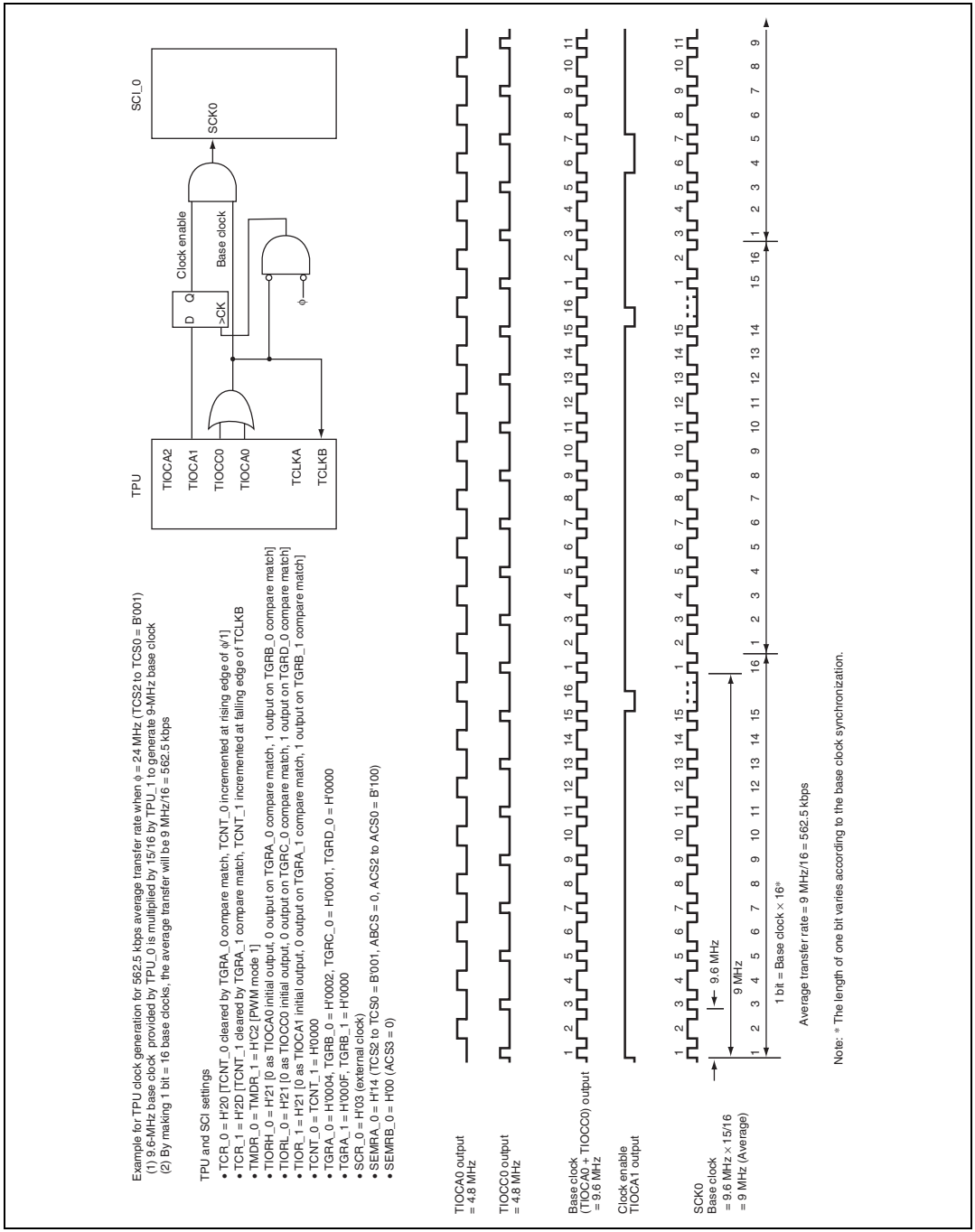


Figure 13.5 Example of Average Transfer Rate Setting when TPU Clock Is Input (2)

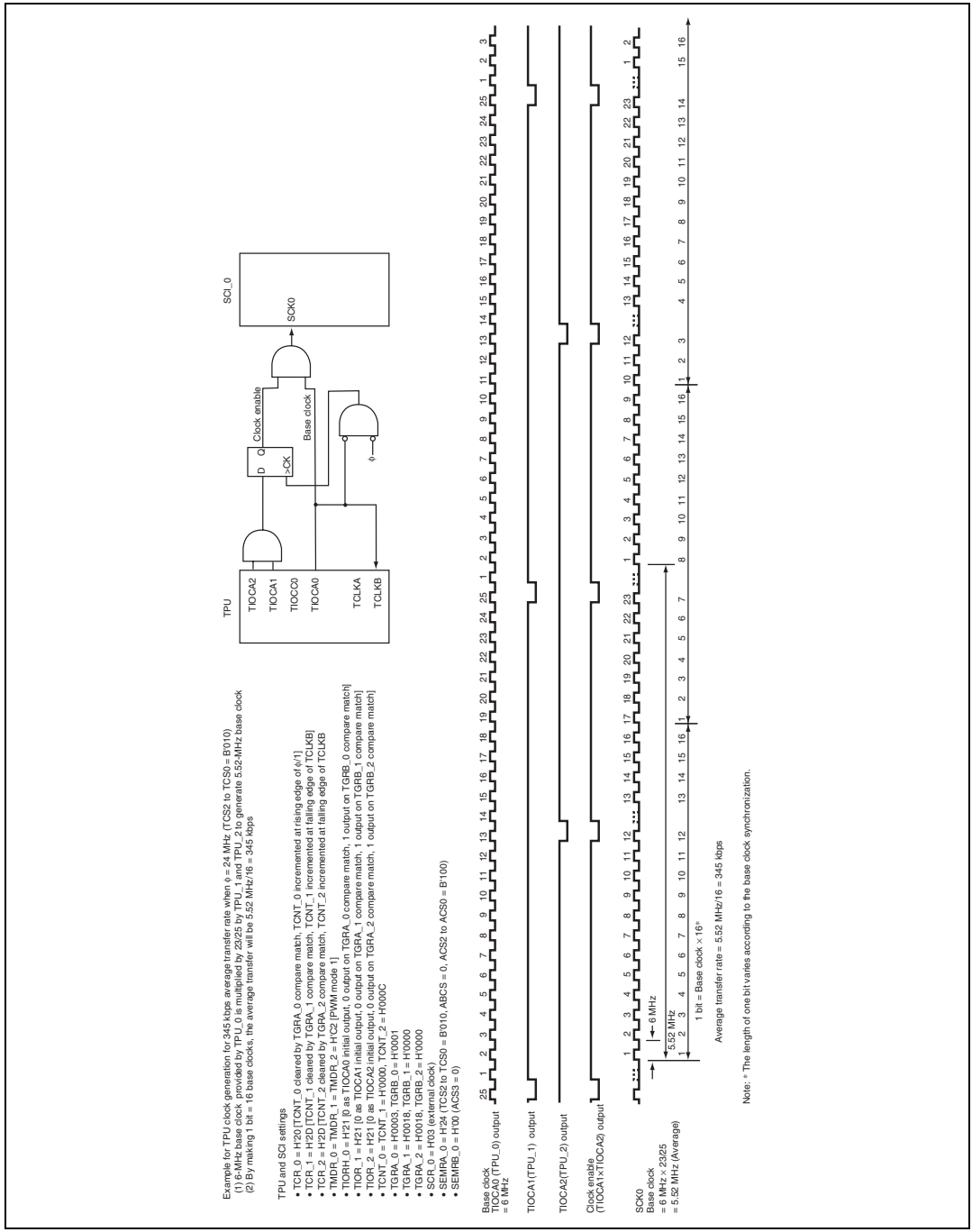


Figure 13.5 Example of Average Transfer Rate Setting when TPU Clock Is Input (3)

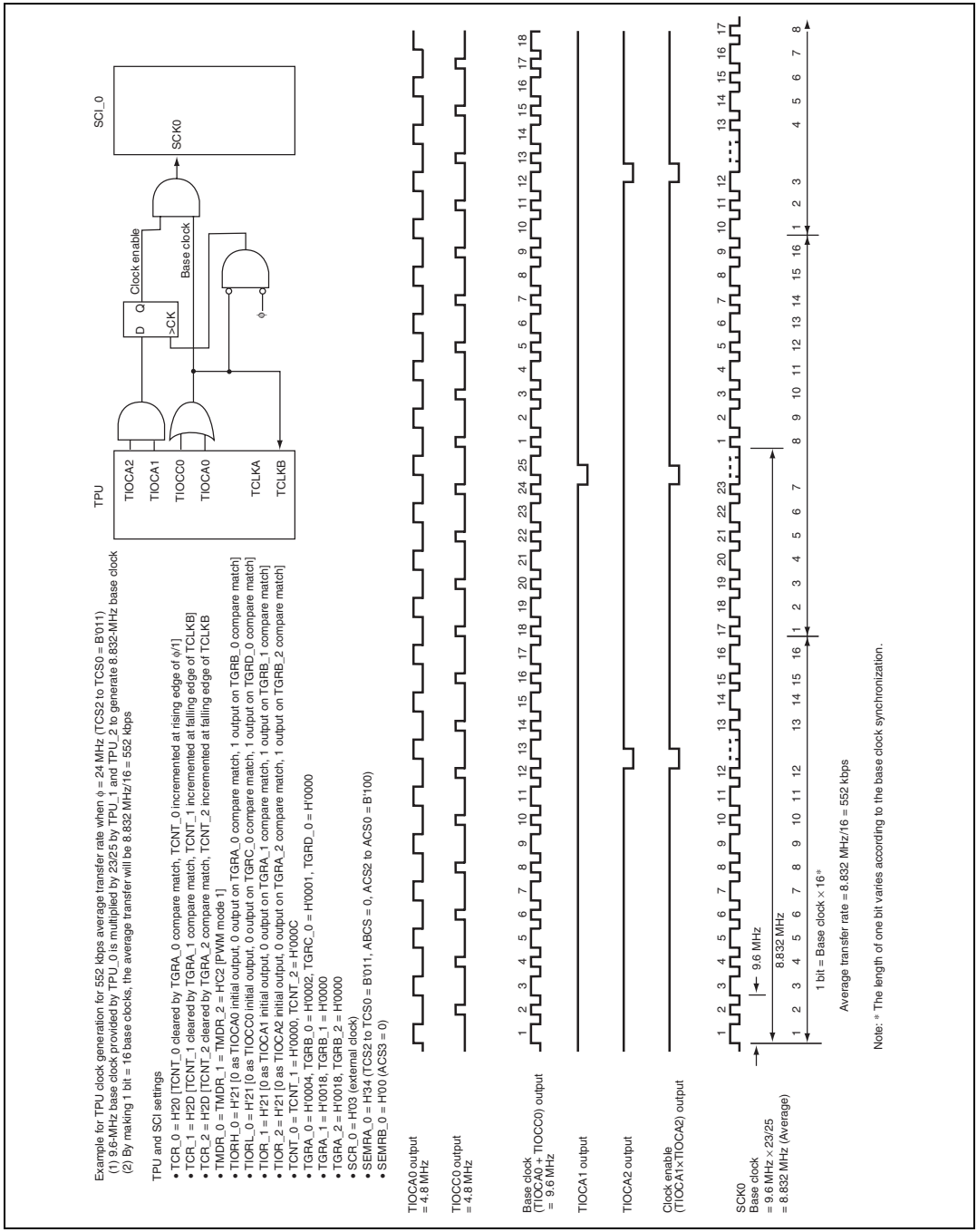


Figure 13.5 Example of Average Transfer Rate Setting when TPU Clock Is Input (4)

13.3.10 Serial Extended Mode Register A_0 (SEMRA_0) (Only for Channel 0 in H8S/2215R and H8S/2215T)

SEMRA_0 extends the functions of SCI_0. SEMR0 enables selection of the SCI_0 select function in synchronous mode, base clock setting in asynchronous mode, and also clock source selection and automatic transfer rate setting. Figure 13.4 shows an example of the internal base clock when an average transfer rate is selected and figure 13.5 shows as example of the setting when the TPU clock input is selected.

Bit	Bit Name	Initial Value	R/W	Description
7	SSE	0	R/W	SCI_0 Select Enable Allows selection of the SCI0 select function when an external clock is input in synchronous mode. The SSE setting is valid when external clock input is used (CKE1 = 1 in SCR) in synchronous mode (C/$\overline{A}$ = 1 in SMR). 0: SCI_0 select function disabled 1: SCI_0 select function enabled When the SCI_0 select function is enabled, if 1 is input to the PG1/$\overline{IRQ7}$ pin, TxD0 output goes to the high-impedance state, SCK0 input is fixed high.
6	TCS2	0	R/W	TPU Clock Select
5	TCS1	0	R/W	When the TPU clock is input (ACS3 to ACS0 = B'0100) as the clock source in asynchronous mode, serial transfer clock is generated depending on the combination of the TPU clock.
4	TCS0	0	R/W	

	Base Clock	Clock Enable	TCLKA	TCLKB	TCLKC
000	TIOCA1	TIOCA2	Base clock written in the left column	Pin input	Pin input
001	TIOCA0 | TIOCC0	TIOCA1	Pin input	Base clock written in the left column	Pin input
010	TIOCA0	TIOCA1 & TIOCA2	Pin input	Base clock written in the left column	Pin input
011	TIOCA0 | TIOCC0	TIOCA1 & TIOCA2	Pin input	Base clock written in the left column	Pin input
1××	Reserved (Setting prohibited)				

Legend:

&: AND (logical multiplication)

I : OR (logical addition)

Note: The functions of bits 6 to 4 are not supported by the E6000 emulator. Figure 13.5 shows the setting examples.

Bit	Bit Name	Initial Value	R/W	Description
3	ABCS	0	R/W	Asynchronous Base Clock Select Selects the 1-bit-interval base clock in asynchronous mode. The ABCS setting is valid in asynchronous mode (C/$\overline{A}$ = 0 in SMR). 0: SCI_0 operates on base clock with frequency of 16 times transfer rate 1: SCI_0 operates on base clock with frequency of 8 times transfer rate
2	ACS2	0	R/W	Asynchronous Clock Source Select 2 to 0 These bits select the clock source in asynchronous mode depending on the combination with the bit 7 (ACS3) in SEMRB_0 (serial extended mode register B_0). When an average transfer rate is selected, the base clock is set automatically regardless of the ABCS value. Note that average transfer rates support only 10.667 MHz, 16 MHz, and 24 MHz, and not support other operating frequencies. Set ACS3 to ACS0 when inputting the external clock (the CKE1 bit in the SCR register is 1) in asynchronous mode (the C/$\overline{A}$ bit in the SMR register is 0). Figures 13.4 and 13.5 show the setting examples. ACS 3210 0000: External clock input 0001: 115.152 kbps average transfer rate (for ϕ = 10.667 MHz only) is selected (SCI_0 operates on base clock with frequency of 16 times transfer rate) 0010: 460.606 kbps average transfer rate (for ϕ = 10.667 MHz only) is selected (SCI_0 operates on base clock with frequency of eight times transfer rate) 0011: 921.569 kbps average transfer rate (for ϕ = 16 MHz only) is selected (SCI_0 operates on base clock with frequency of eight times transfer rate) 0100: TPU clock input The signal generated by TIOCA0, TIOCC0, TIOCA1, and TIOCA2, which are the compare match outputs for TPU_0 to TPU_2 or PWM outputs, is used as a base clock. Note that $\overline{IRQ0}$ and $\overline{IRQ1}$ cannot be used since TIOCA1 and TIOCA2 are used as outputs.
1	ACS1	0	R/W	
0	ACS0	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
2	ACS2	0	R/W	0101: 115.196 kbps average transfer rate (for ϕ = 16 MHz only) is selected (SCI_0 operates on base clock with frequency of 16 times transfer rate)
1	ACS1	0	R/W	
0	ACS0	0	R/W	0110: 460.784 kbps average transfer rate (for ϕ = 16 MHz only) is selected (SCI_0 operates on base clock with frequency of eight times transfer rate)
				0111: 720 kbps average transfer rate (for ϕ = 16 MHz only) is selected (SCI_0 operates on base clock with frequency of eight times transfer rate)
				1000: 115.132 kbps average transfer rate (for ϕ = 24 MHz only) is selected* (SCI_0 operates on base clock with frequency of 16 times transfer rate)
				1001: 460.526 kbps average transfer rate (for ϕ = 24 MHz only) is selected* (SCI_0 operates on base clock with frequency of 16 times transfer rate)
				1010: 720 kbps average transfer rate (for ϕ = 24 MHz only) is selected* (SCI_0 operates on base clock with frequency of eight times transfer rate)
				1011: 921.053 kbps average transfer rate (for ϕ = 24 MHz only) is selected* (SCI_0 operates on base clock with frequency of eight times transfer rate)
				11××: Reserved (Setting prohibited)

Note: * The average transfer rate select functions for 24 MHz only (ACS3 to ACS0 = B'10XX) are not supported by the E6000 emulator.

13.3.11 Serial Extended Mode Register B_0 (SEMRB_0) (Only for Channel 0 in H8S/2215R and H8S/2215T)

SEMRB_0 enables clock source selection with the combination of SEMRA_0, automatic transfer rate setting, and control of port 1 pins (P16, P14, P12, and P10) at the transfer clock generation by TPU.

Note: SEMRB_0 is not supported by the E6000 emulator.

Bit	Bit Name	Initial Value	R/W	Description
7	ACS3	0	R/W	Asynchronous Clock Source Select Selects the clock source in asynchronous mode depending on the combination with the ACS2 to ACS0 (bits 2 to 0 in SEMRA_0). For details, see section 13.3.9, Serial Extended Mode Register (SEMR) (Only for channel 0 in H8S/2215).
6 to 4	—	Undefined	—	Reserved The write value should always be 0.
3	TIOCA2E	1	R/W	TIOCA2 Output Enable Controls the TIOCA2 output on the P16 pin. When the TIOCA2 in TPU is output to generate the transfer clock, P16 is used as other function pin by setting this bit to 0. 0: Disables output of TIOCA2 in TPU 1: Enables output of TIOCA2 in TPU
2	TIOCA1E	1	R/W	TIOCA1 Output Enable Controls the TIOCA1 output on the P14 pin. When the TIOCA1 in TPU is output to generate the transfer clock, P14 is used as other function pin by setting this bit to 0. 0: Disables output of TIOCA1 in TPU 1: Enables output TIOCA1 in TPU
1	TIOCC0E	1	R/W	TIOCC0 Output Enable Controls the TIOCC0 output on the P12 pin. When the TIOCC0 in TPU is output to generate the transfer clock, P12 is used as other function pin by setting this bit to 0. 0: Disables output of TIOCC0 in TPU 1: Enables output of TIOCC0 in TPU
0	TIOCA0E	1	R/W	TIOCA0 Output Enable Controls the TIOCA0 output on the P10 pin. When the TIOCA0 in TPU is output to generate the transfer clock, P10 is used as other function pin by setting this bit to 0. 0: Disables output of TIOCA0 in TPU 1: Enables output of TIOCA0 in TPU

13.3.12 Bit Rate Register (BRR)

BRR is an 8-bit register that adjusts the bit rate. As the SCI performs baud rate generator control independently for each channel, different bit rates can be set for each channel. Table 13.2 shows the relationships between the N setting in BRR and bit rate B for normal asynchronous mode, clocked synchronous mode, and Smart Card interface mode. The initial value of BRR is H'FF, and it can be read from or written to by the CPU at all times.

Table 13.2 Relationships between the N Setting in BRR and Bit Rate B

Mode	ABCS	Bit Rate	Error
Asynchronous mode	0	$B = \frac{\phi \times 10^6}{64 \times 2^{2n-1} \times (N + 1)}$	$\text{Error (\%)} = \left\| \frac{\phi \times 10^6}{B \times 64 \times 2^{2n-1} \times (N + 1)} - 1 \right\| \times 100$
	1	$B = \frac{\phi \times 10^6}{32 \times 2^{2n-1} \times (N + 1)}$	$\text{Error (\%)} = \left\| \frac{\phi \times 10^6}{B \times 32 \times 2^{2n-1} \times (N + 1)} - 1 \right\| \times 100$
Clocked synchronous mode	x	$B = \frac{\phi \times 10^6}{8 \times 2^{2n-1} \times (N + 1)}$	—
Smart Card interface mode	x	$B = \frac{\phi \times 10^6}{S \times 2^{2n+1} \times (N + 1)}$	$\text{Error (\%)} = \left\| \frac{\phi \times 10^6}{B \times S \times 2^{2n+1} \times (N + 1)} - 1 \right\| \times 100$

Legend:

B: Bit rate (bps)

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

ϕ: Operating frequency (MHz)

n, S: Determined by the SMR settings shown in the following tables.

x: Don't care

SMR Setting				SMR Setting		
CKS1	CKS0	Clock Source	n	BCP1	BCP0	S
0	0	ϕ	0	0	0	32
0	1	$\phi/4$	1	0	1	64
1	0	$\phi/16$	2	1	0	372
1	1	$\phi/64$	3	1	1	256

Table 13.3 shows sample N settings in BRR in normal asynchronous mode. Table 13.4 shows the maximum bit rate for each frequency in normal asynchronous mode. Table 13.6 shows sample N settings in BRR in clocked synchronous mode. Table 13.8 shows sample N settings in BRR in Smart Card interface mode. In Smart Card interface mode, S (the number of basic clock periods in a 1-bit transfer interval) can be selected. For details, see section 13.7.5, Receive Data Sampling

Timing and Reception Margin. Tables 13.5 and 13.7 show the maximum bit rates with external clock input.

When the ABCS bit in SCI_0's serial extended mode register (SEMR) is set to 1 in asynchronous mode, the maximum bit rates are twice those shown in table 13.3.

Table 13.3 BRR Settings for Various Bit Rates (Asynchronous Mode)

Bit Rate	Operating Frequency ϕ (MHz)											
	2			2.097152			2.4576			3		
(bit/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	—	—	—	—	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	—	—	—

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

Bit Rate (bit/s)	Operating Frequency φ (MHz)											
	6			6.144			7.3728			8		
	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	—	—	—

Bit Rate (bit/s)	Operating Frequency ϕ (MHz)											
	9.8304			10			12			12.288		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	174	–0.26	2	177	–0.25	2	212	0.03	2	217	0.08
150	2	127	0.00	2	129	0.16	2	155	0.16	2	159	0.00
300	1	255	0.00	2	64	0.16	2	77	0.16	2	79	0.00
600	1	127	0.00	1	129	0.16	1	155	0.16	1	159	0.00
1200	0	255	0.00	1	64	0.16	1	77	0.16	1	79	0.00
2400	0	127	0.00	0	129	0.16	0	155	0.16	0	159	0.00
4800	0	63	0.00	0	64	0.16	0	77	0.16	0	79	0.00
9600	0	31	0.00	0	32	–1.36	0	38	0.16	0	39	0.00
19200	0	15	0.00	0	15	1.73	0	19	–2.34	0	19	0.00
31250	0	9	–1.70	0	9	0.00	0	11	0.00	0	11	2.40
38400	0	7	0.00	0	7	1.73	0	9	–2.34	0	9	0.00

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

Bit Rate (bps)	Operating Frequency ϕ (MHz)											
	17.2032			18			19.6608			20		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	75	0.48	3	79	–0.12	3	86	0.31	3	88	–0.25
150	2	223	0.00	2	233	0.16	2	255	0.00	3	64	0.16
300	2	111	0.00	2	116	0.16	2	127	0.00	2	129	0.16
600	1	223	0.00	1	233	0.16	1	255	0.00	2	64	0.16
1200	1	111	0.00	1	116	0.16	1	127	0.00	1	129	0.16
2400	0	223	0.00	0	233	0.16	0	255	0.00	1	64	0.16
4800	0	111	0.00	0	116	0.16	0	127	0.00	0	129	0.16
9600	0	55	0.00	0	58	–0.69	0	63	0.00	0	64	0.16
19200	0	27	0.00	0	28	1.02	0	31	0.00	0	32	–1.36
31250	0	16	1.20	0	17	0.00	0	19	–1.17	0	19	0.00
38400	0	13	0.00	0	14	–2.34	0	15	0.00	0	15	1.73

Bit Rate (bps)	Operating Frequency ϕ (MHz)		
	24		
	n	N	Error (%)
110	3	106	–0.44
150	3	77	0.16
300	2	155	0.16
600	2	77	0.16
1200	1	155	0.16
2400	1	77	0.16
4800	0	155	0.16
9600	0	77	0.16
19200	0	38	0.16
31250	0	23	0.00
38400	0	19	–2.34

Note: This table shows bit rates when the ABCS bit in SEMRA_0 is cleared to 0.
When the ABCS bit in SEMRA_0 is set to 1, the bit rates are twice those shown in this table.
In this LSI, operating frequency ϕ must be 13 MHz or greater.

Table 13.4 Maximum Bit Rate for Each Frequency (Asynchronous Mode)

	Maximum Bit Rate (kbps)					Maximum Bit Rate (kbps)			
φ (MHz)	ABCS = 0	ABCS = 1	n	N	φ (MHz)	ABCS = 0	ABCS = 1	n	N
2	62.5	125.0	0	0	9.8304	307.2	614.4	0	0
2.097152	65.536	131.027	0	0	10	312.5	625.0	0	0
2.4576	76.8	153.6	0	0	12	375.0	750.0	0	0
3	93.75	187.5	0	0	12.288	384.0	768.0	0	0
3.6864	115.2	230.4	0	0	14	437.5	875.0	0	0
4	125.0	250.0	0	0	14.7456	460.8	921.6	0	0
4.9152	153.6	307.2	0	0	16	500.0	1000.0	0	0
5	156.25	312.5	0	0	17.2032	537.6	1075.2	0	0
6	187.5	375.0	0	0	18	562.5	1125.0	0	0
6.144	192.0	384.0	0	0	19.6608	614.4	1228.8	0	0
7.3728	230.4	460.8	0	0	20	625.0	1250.0	0	0
8	250.0	500.0	0	0	24	750.0	1500.0	0	0

Table 13.5 Maximum Bit Rate with External Clock Input (Asynchronous Mode)

	External Input Clock	Maximum Bit Rate (kbps)			External Input Clock	Maximum Bit Rate (kbps)	
φ (MHz)	(MHz)	ABCS = 0	ABCS = 1	φ (MHz)	(MHz)	ABCS = 0	ABCS = 1
2	0.5000	31.25	62.5	9.8304	2.4576	153.6	307.2
2.097152	0.5243	327.68	65.536	10	2.5000	156.25	312.5
2.4576	0.6144	38.4	76.8	12	3.0000	187.5	375.0
3	0.7500	46.875	93.75	12.288	3.0720	192.0	384.0
3.6864	0.9216	57.6	115.2	14	3.5000	218.75	437.0
4	1.0000	62.5	125.0	14.7456	3.6864	230.4	460.8
4.9152	1.2288	76.8	153.6	16	4.0000	250.0	500.0
5	1.2500	78.125	156.25	17.2032	4.3008	268.8	537.6
6	1.5000	93.75	187.5	18	4.5000	281.25	562.5
6.144	1.5360	96.0	192.0	19.6608	4.9152	307.2	614.4
7.3728	1.8432	115.2	230.4	20	5.0000	312.5	625.0
8	2.0000	125.0	250.0	24	6.0000	375.0	750.0

Note: In this LSI, operating frequency φ must be 13 MHz or greater.

Table 13.6 BRR Settings for Various Bit Rates (Clocked Synchronous Mode)

Bit Rate (bps)	Operating Frequency φ (MHz)															
	2		4		6		8		10		16		20		24	
	n	N	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	—	—	—	—
2.5 k	0	199	1	99	1	149	1	199	1	249	2	99	2	124	2	149
5 k	0	99	0	199	1	74	1	99	1	124	1	199	1	249	2	74
10 k	0	49	0	99	0	149	0	199	0	249	1	99	1	124	1	149
25 k	0	19	0	39	0	59	0	79	0	99	0	159	0	199	0	239
50 k	0	9	0	19	0	29	0	39	0	49	0	79	0	99	0	119
100 k	0	4	0	9	0	14	0	19	0	24	0	39	0	49	0	59
250 k	0	1	0	3	0	5	0	7	0	9	0	15	0	19	0	23
500 k	0	0*	0	1	0	2	0	3	0	4	0	7	0	9	0	11
1 M			0	0*			0	1			0	3	0	4	0	5
2 M							0	0*			0	1			0	2
2.5 M									0	0*			0	1	—	—
4 M											0	0*				
5 M													0	0*	—	—
6 M															0	0*

Legend:

Blank: Cannot be set.

—: Can be set, but there will be a degree of error.

*: Continuous transfer is not possible.

Table 13.7 Maximum Bit Rate with External Clock Input (Clocked Synchronous Mode)

φ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (Mbps)	φ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (Mbps)
2	0.333	0.333	14	2.333	2.333
4	0.667	0.667	16	2.667	2.667
6	1.000	1.000	18	3.000	3.000
8	1.333	1.333	20	3.333	3.333
10	1.667	1.667	24	4.000	4.000
12	2.000	2.000			

Note: In this LSI, operating frequency φ must be 13 MHz or greater.

Table 13.8 BRR Settings for Various Bit Rates (Smart Card Interface Mode, when n = 0 and S = 372)

	Operating Frequency φ (MHz)											
	5.00		7.00		7.1424		10.00		10.7136		13.00	
Bit Rate (bps)	N	Error (%)	N	Error (%)	N	Error (%)	N	Error (%)	N	Error (%)	N	Error (%)
6720	0	0.01	1	30.00	1	28.57	1	0.01	1	7.14	2	13.33
9600	0	30.00	0	1.99	0	0.00	1	30.00	1	25.00	1	8.99

	Operating Frequency φ (MHz)									
	14.2848		16.00		18.00		20.00		24.00	
Bit Rate (bps)	N	Error (%)	N	Error (%)	N	Error (%)	N	Error (%)	N	Error (%)
6720	2	4.76	2	6.67	3	0.01	3	0.01	4	3.99
9600	1	0.00	1	12.01	2	15.99	2	6.66	2	12.01

Table 13.9 Maximum Bit Rate at Various Frequencies (Smart Card Interface Mode)

	Maximum Bit Rate (bps)					
φ (MHz)	S = 32	S = 64	S = 256	S = 372	n	N
5.00	78125	39063	9766	6720	0	0
6.00	93750	46875	11719	8065		
7.00	109375	54688	13672	9409	0	0
7.1424	111600	55800	13950	9600	0	0
10.00	156250	78125	19531	13441	0	0
10.7136	167400	83700	20925	14400	0	0
13.00	203125	101563	25391	17473	0	0
14.2848	223200	111600	27900	19200	0	0
16.00	250000	125000	31250	21505	0	0
18.00	281250	140625	35156	24194	0	0
20.00	312500	156250	39063	26882	0	0
24.00	375000	187500	46875	32258	0	0

Note: In this LSI, operating frequency φ must be 13 MHz or greater.

13.4 Operation in Asynchronous Mode

Figure 13.6 shows the general format for asynchronous serial communication. One frame consists of a start bit (low level), followed by data (in LSB-first order), a parity bit (high or low level), and finally stop bits (high level). In asynchronous serial communication, the transmission line is usually held in the mark state (high level). The SCI monitors the transmission line. When the transmission line goes to the space state (low level), the SCI recognizes a start bit and starts serial communication. Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex. Both the transmitter and the receiver also have a double-buffered structure, so data can be read from or written during transmission or reception, enabling continuous data transfer.

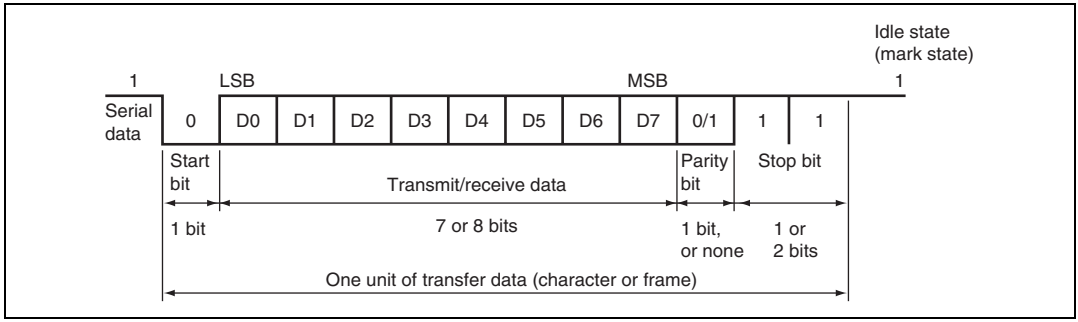


Figure 13.6 Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, Two Stop Bits)

13.4.1 Data Transfer Format

Table 13.10 shows the data transfer formats that can be used in asynchronous mode. Any of 12 transfer formats can be selected according to the SMR setting. For details on the multiprocessor bit, refer to section 13.5, Multiprocessor Communication Function.

Table 13.10 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

13.4.2 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode

In asynchronous mode, the SCI operates on a basic clock with a frequency of 16 times the transfer rate. In reception, the SCI samples the falling edge of the start bit using the basic clock, and performs internal synchronization. Receive data is latched internally at the rising edge of the 8th pulse of the basic clock as shown in Figure 13.7. Thus, the reception margin in asynchronous mode is given by formula (1) below.

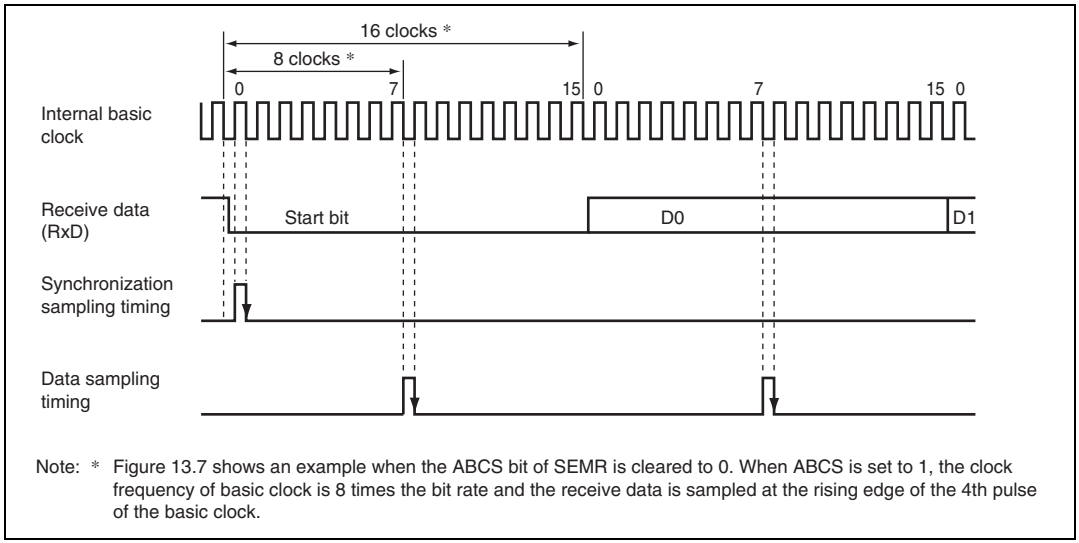
$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5)\,F - \frac{|\,D - 0.5\,|}{N}\,(1 + F) \right| \times 100\ [\%] \qquad \text{... Formula (1)}$$

Where M: Reception margin
N: Ratio of bit rate to clock (N = 16 if ABCS = 0, N = 8 if ABCS = 1)
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 (absolute value of clock rate deviation) = 0, D (clock duty) = 0.5, and N (ratio of bit rate to clock) = 16 in formula (1), the reception margin can be given by the formula.

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

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



Note: * Figure 13.7 shows an example when the ABCS bit of SEMR is cleared to 0. When ABCS is set to 1, the clock frequency of basic clock is 8 times the bit rate and the receive data is sampled at the rising edge of the 4th pulse of the basic clock.

Figure 13.7 Receive Data Sampling Timing in Asynchronous Mode

13.4.3 Clock

Either an internal clock generated by the on-chip baud rate generator or an external clock input at the SCK pin can be selected as the SCI's serial clock, according to the setting of the C/$\overline{A}$ bit in SMR and the CKE0 and CKE1 bits in SCR. When an external clock is input at the SCK pin, the clock frequency should be 16 times the bit rate used. When an external clock is selected, the basic clock of average transfer rate can be selected according to the ACS2 to ACS0 bit setting of SEMR.

When the SCI is operated on an internal clock, the clock can be output from the SCK pin by setting CKE1 = 0 and CKE0 = 1. The frequency of the clock output in this case is equal to the bit rate, and the phase is such that the rising edge of the clock is in the middle of the transmit data, as shown in figure 13.8.

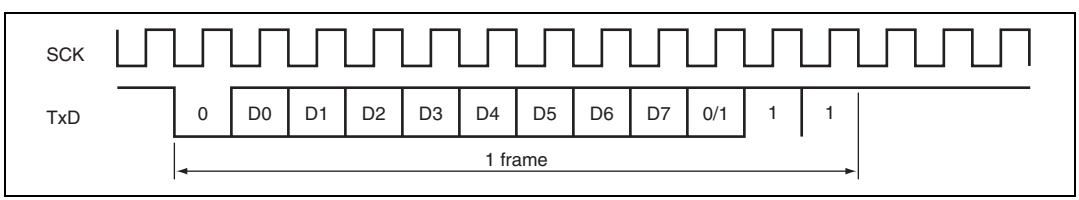


Figure 13.8 Relationship between Output Clock and Transfer Data Phase (Asynchronous Mode)

13.4.4 SCI Initialization (Asynchronous Mode)

Before transmitting and receiving data, you should first clear the TE and RE bits in SCR to 0, then initialize the SCI as described in a sample flowchart in figure 13.9. When the operating mode, or transfer format, is changed for example, 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. Note that clearing the RE bit to 0 does not initialize the contents of the RDRF, PER, FER, and ORER flags, or the contents of RDR. When the external clock is used in asynchronous mode, the clock must be supplied even during initialization.

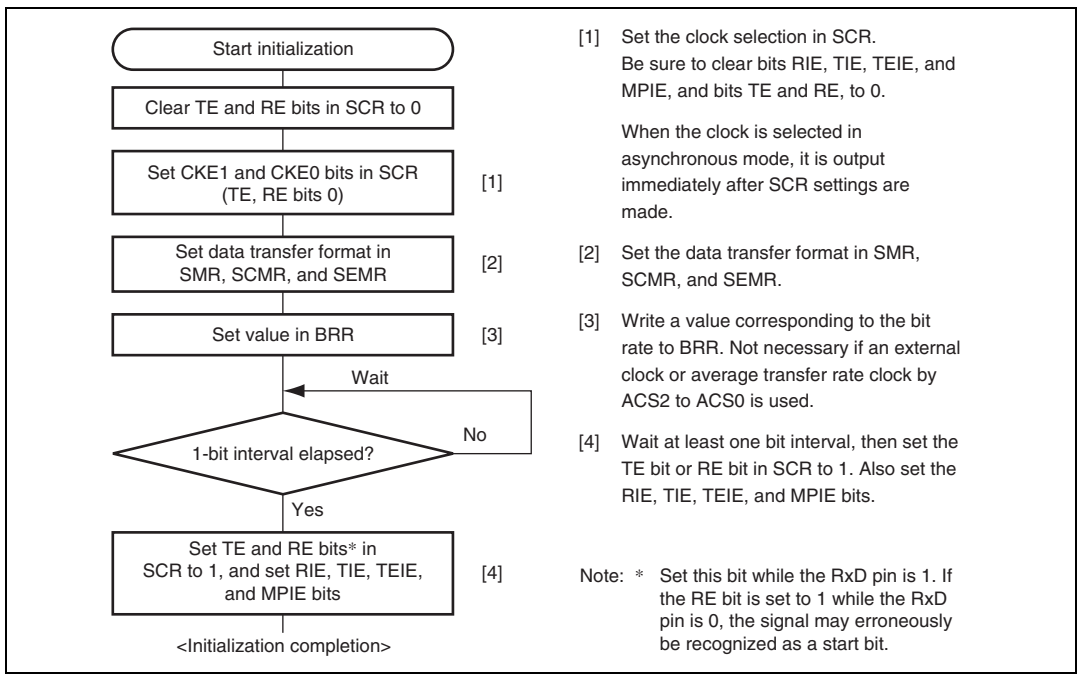


Figure 13.9 Sample SCI Initialization Flowchart

13.4.5 Data Transmission (Asynchronous Mode)

Figure 13.10 shows an example of operation for transmission in asynchronous mode. In transmission, the SCI operates as described below.

1. The SCI monitors the TDRE flag in SSR. If the flag is cleared to 0, the SCI 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 request (TXI) is generated. Continuous transmission is possible because the TXI interrupt routine writes next transmit data to TDR before transmission of the current transmit data has been completed.
3. Data is sent from the TxD pin in the following order: start bit, transmit data, parity bit or multiprocessor bit (may be omitted depending on the format), and stop bit.
4. The SCI checks the TDRE flag at the timing for sending the stop bit.
5. If the TDRE flag is 0, the data is transferred from TDR to TSR, the stop bit is sent, and then serial transmission of the next frame is started.
6. If the TDRE flag is 1, the TEND flag in SSR is set to 1, the stop bit is sent, and then the "mark state" is entered, in which 1 is output. If the TEIE bit in SCR is set to 1 at this time, a TEI interrupt request is generated.

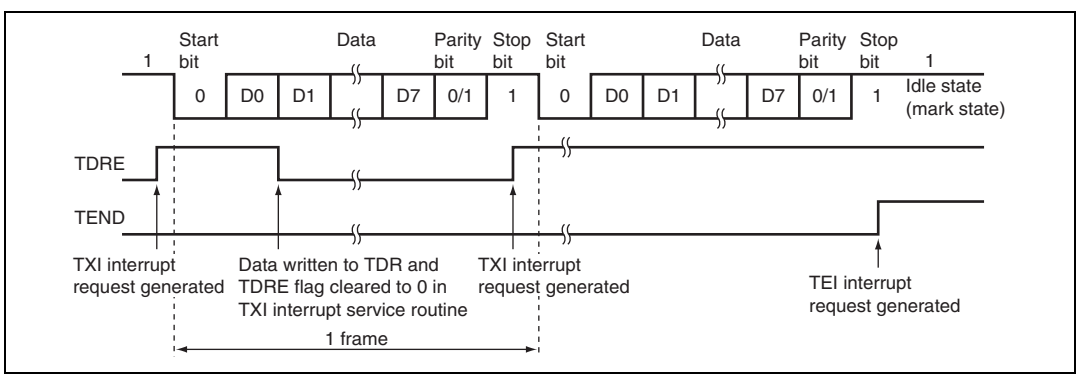
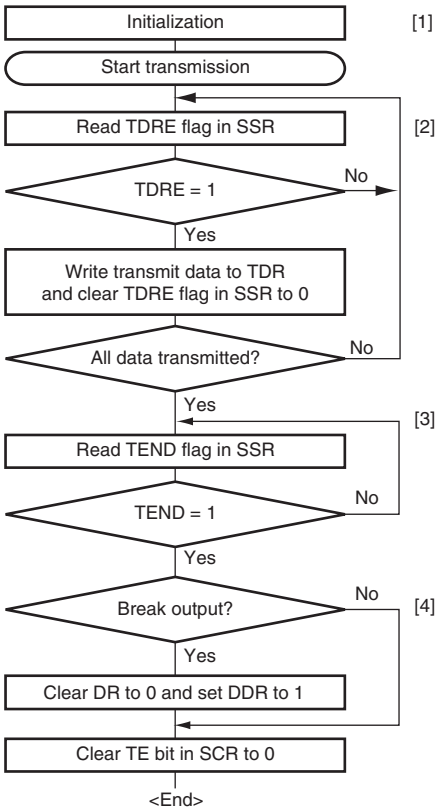


Figure 13.10 Example of Operation in Transmission in Asynchronous Mode (Example with 8-Bit Data, Parity, One Stop Bit)

Figure 13.11 shows a sample flowchart for transmission in asynchronous mode.



[1] SCI initialization:
The TxD pin is automatically designated as the transmit data output pin.
After the TE bit is set to 1, a frame of 1s is output, and transmission is enabled.

[2] SCI status check and transmit data write:
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR 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 the 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.

Note: * Checking and clearing of the TDRE flag are performed automatically by the DTC when the DTC's DISEL bit is cleared to 0 and the transfer counter value is not 0. Consequently, it is necessary to use the CPU to clear the TDRE flag if DISEL is set to 1 or if the transfer counter value is 0.

Figure 13.11 Sample Serial Transmission Data Flowchart

13.4.6 Serial Data Reception (Asynchronous Mode)

Figure 13.12 shows an example of operation for reception in asynchronous mode. In serial reception, the SCI operates as described below.

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

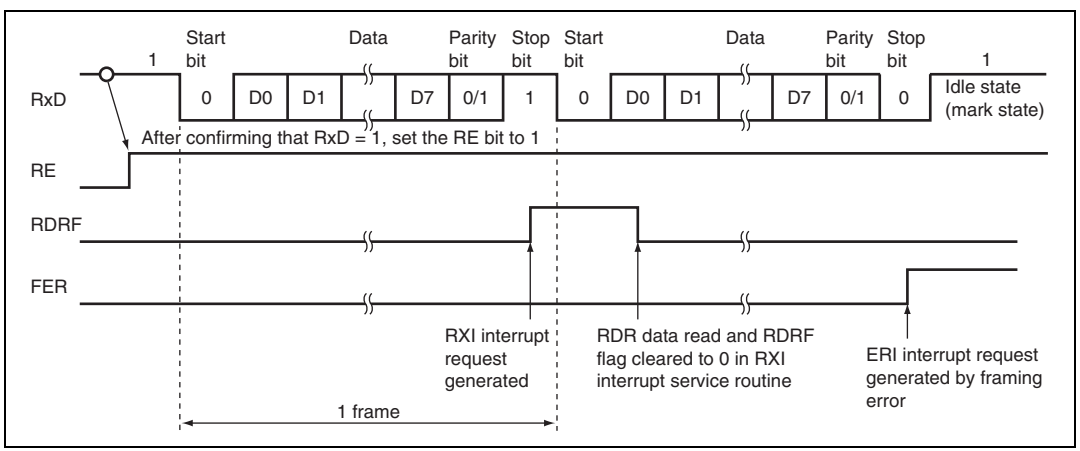


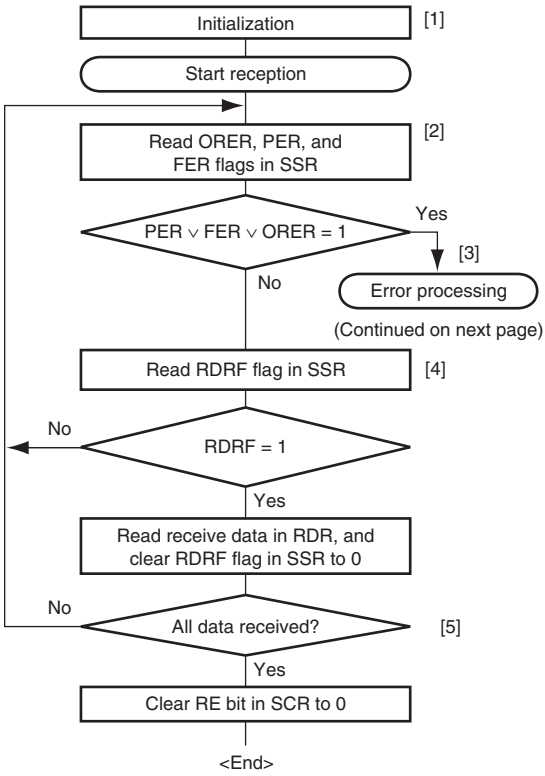
Figure 13.12 Example of SCI Operation in Reception (Example with 8-Bit Data, Parity, One Stop Bit)

Table 13.11 shows the states of the SSR status flags and receive data handling when a receive error is detected. If a receive error is detected, the RDRF flag retains its state before receiving data. Reception cannot be resumed while a receive error flag is set to 1. Accordingly, clear the ORER, FER, PER, and RDRF bits to 0 before resuming reception. Figure 13.13 shows a sample flow chart for serial data reception.

Table 13.11 SSR Status Flags and Receive Data Handling

SSR Status Flag					
RDRF*	**ORER**	**FER**	**PER**	**Receive Data**	**Receive Error Type**
1	1	0	0	Lost	Overrun error
0	0	1	0	Transferred to RDR	Framing error
0	0	0	1	Transferred to RDR	Parity error
1	1	1	0	Lost	Overrun error + framing error
1	1	0	1	Lost	Overrun error + parity error
0	0	1	1	Transferred to RDR	Framing error + parity error
1	1	1	1	Lost	Overrun error + framing error + parity error

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



[1] SCI initialization:
The RxD pin is automatically designated as the receive data input pin.

[2] [3] Receive error processing and break detection:
If a receive error occurs, read the ORER, PER, and FER flags in SSR to identify the error. After performing the appropriate error processing, ensure that the ORER, PER, and FER flags are all cleared to 0. Reception cannot be resumed if any of these flags are set to 1. In the case of a framing error, a break can be detected by reading the value of the input port corresponding to the RxD pin.

[4] SCI status check and receive data read:
Read SSR and check that RDRF = 1, then read the receive data in RDR and clear the RDRF flag to 0. Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial reception continuation procedure:
To continue serial reception, before the end bit for the current frame is received, reading the RDRF flag and RDR, and clearing the RDRF flag to 0 should be finished. The RDRF flag is cleared automatically when DMAC or the DTC* is activated by a reception complete interrupt (RXI) and the RDR value is read.

Note: * Clearing of the RDRF flag are performed automatically by the DTC when the DTC's DISEL bit is cleared to 0 and the transfer counter value is not 0. Consequently, it is necessary to use the CPU to clear the RDRF flag if DISEL is set to 1 or if the transfer counter value is 0.

Figure 13.13 Sample Serial Reception Data Flowchart (1)

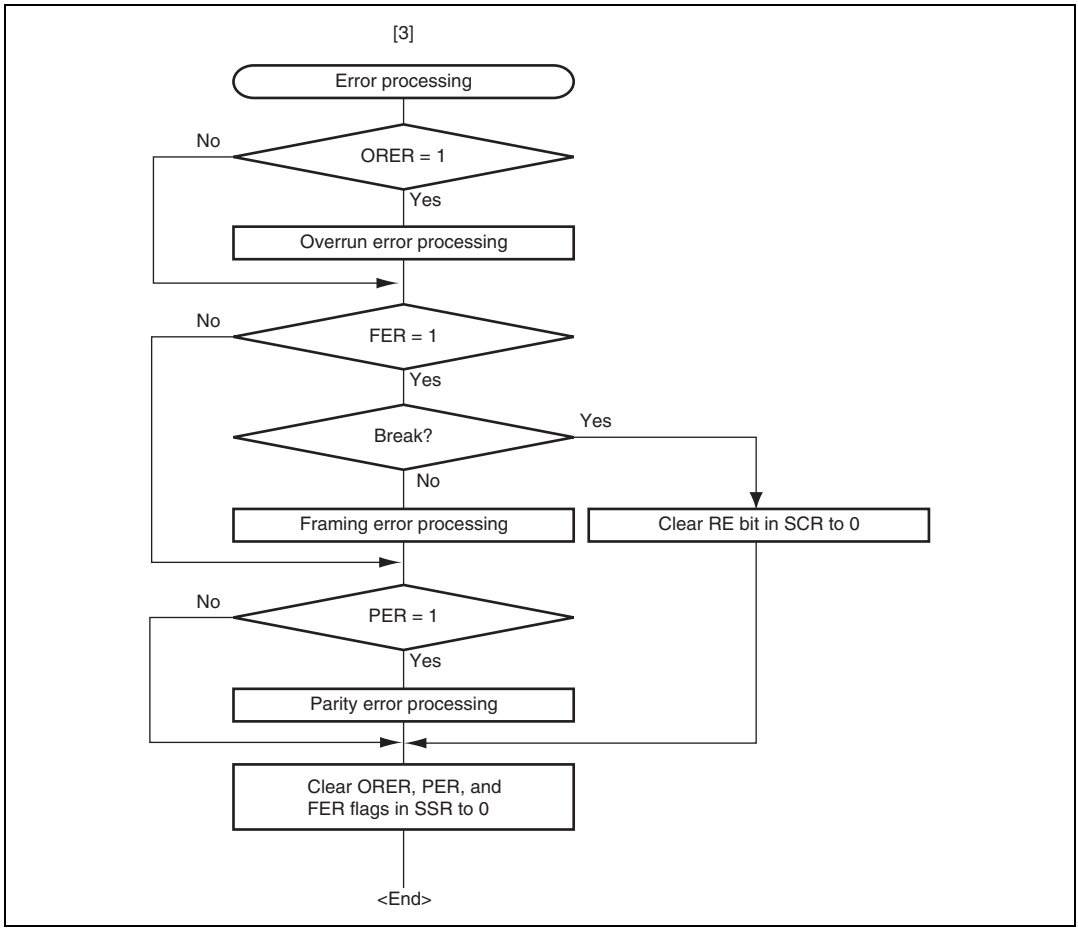


Figure 13.13 Sample Serial Reception Data Flowchart (2)

13.5 Multiprocessor Communication Function

Use of the multiprocessor communication function enables data transfer between a number of processors sharing communication lines by asynchronous serial communication using the multiprocessor format, in which a multiprocessor bit is added to the transfer data. When multiprocessor communication is performed, each receiving station is addressed by a unique ID code. The serial communication cycle consists of two component cycles; an ID transmission cycle that specifies the receiving station, and a data transmission cycle. The multiprocessor bit is used to differentiate between the ID transmission cycle and the data transmission cycle. If the multiprocessor bit is 1, the cycle is an ID transmission cycle; if the multiprocessor bit is 0, the cycle is a data transmission cycle. Figure 13.14 shows an example of inter-processor communication using the multiprocessor format. The transmitting station first sends the ID code of the receiving station with which it wants to perform serial communication as data with a 1 multiprocessor bit added. It then sends transmit data as data with a 0 multiprocessor bit added. When data with a 1 multiprocessor bit is received, the receiving station compares that data with its own ID. The station whose ID matches then receives the data sent next. Stations whose ID do not match continue to skip data until data with a 1 multiprocessor bit is again received.

The SCI uses the MPIE bit in SCR to implement this function. When the MPIE bit is set to 1, transfer of receive data from RSR to RDR, error flag detection, and setting the SSR status flags, RDRF, FER, and ORER to 1, are inhibited until data with a 1 multiprocessor bit is received. On reception of a receive character with a 1 multiprocessor bit, the MPB bit in SSR is set to 1 and the MPIE bit is automatically cleared, thus normal reception is resumed. If the RIE bit in SCR is set to 1 at this time, an RXI interrupt is generated.

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

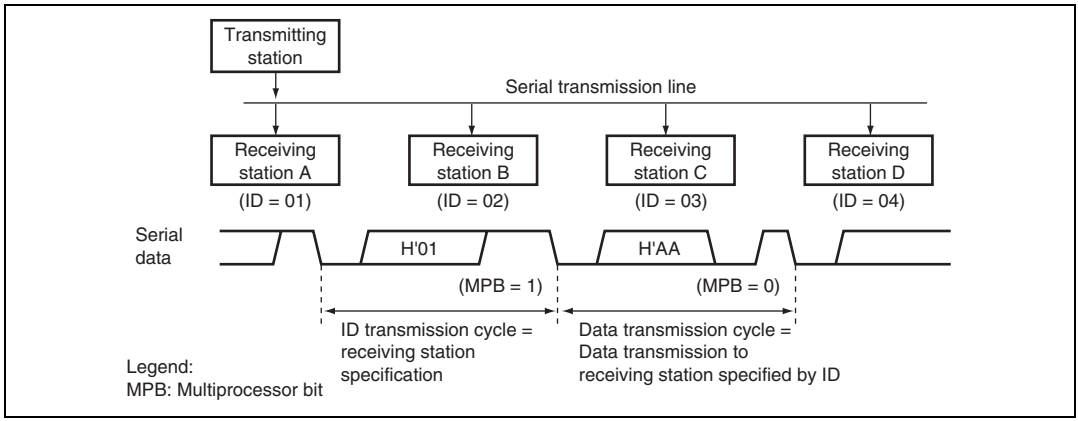
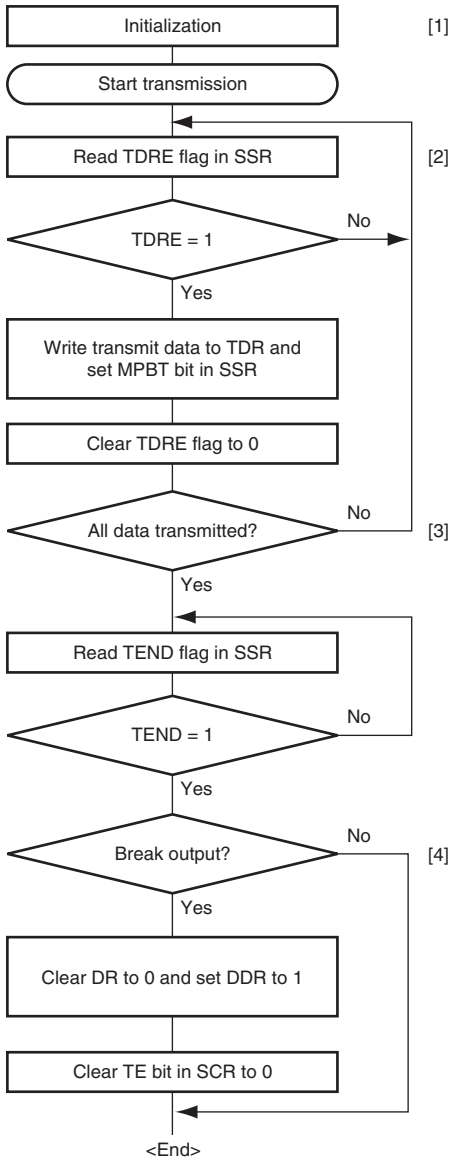


Figure 13.14 Example of Communication Using Multiprocessor Format (Transmission of Data H'AA to Receiving Station A)

13.5.1 Multiprocessor Serial Data Transmission

Figure 13.15 shows a sample flowchart for multiprocessor serial data transmission. For an ID transmission cycle, set the MPBT bit in SSR to 1 before transmission. For a data transmission cycle, clear the MPBT bit in SSR to 0 before transmission. All other SCI operations are the same as those in asynchronous mode.



[1] SCI initialization:
The TxD pin is automatically designated as the transmit data output pin.
After the TE bit is set to 1, a frame of 1s is output, and transmission is enabled.

[2] SCI status check and transmit data write:
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR. Set the MPBT bit in SSR to 0 or 1. Finally, clear the TDRE flag to 0.

[3] Serial transmission continuation procedure:
To continue serial transmission, be sure to read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and then clear the TDRE flag to 0. Checking and clearing of the TDRE flag is automatic when the DMAC or the 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.

Note: * Checking and clearing of the TDRE flag are performed automatically by the DTC when the DTC's DISEL bit is cleared to 0 and the transfer counter value is not 0. Consequently, it is necessary to use the CPU to clear the TDRE flag if DISEL is set to 1 or if the transfer counter value is 0.

Figure 13.15 Sample Multiprocessor Serial Transmission Flowchart

13.5.2 Multiprocessor Serial Data Reception

Figure 13.17 shows a sample flowchart for multiprocessor serial data reception. If the MPIE bit in SCR is set to 1, data is skipped until data with a 1 multiprocessor bit is sent. On receiving data with a 1 multiprocessor bit, the receive data is transferred to RDR. An RXI interrupt request is generated at this time. All other SCI operations are the same as in asynchronous mode. Figure 13.16 shows an example of SCI operation for multiprocessor format reception.

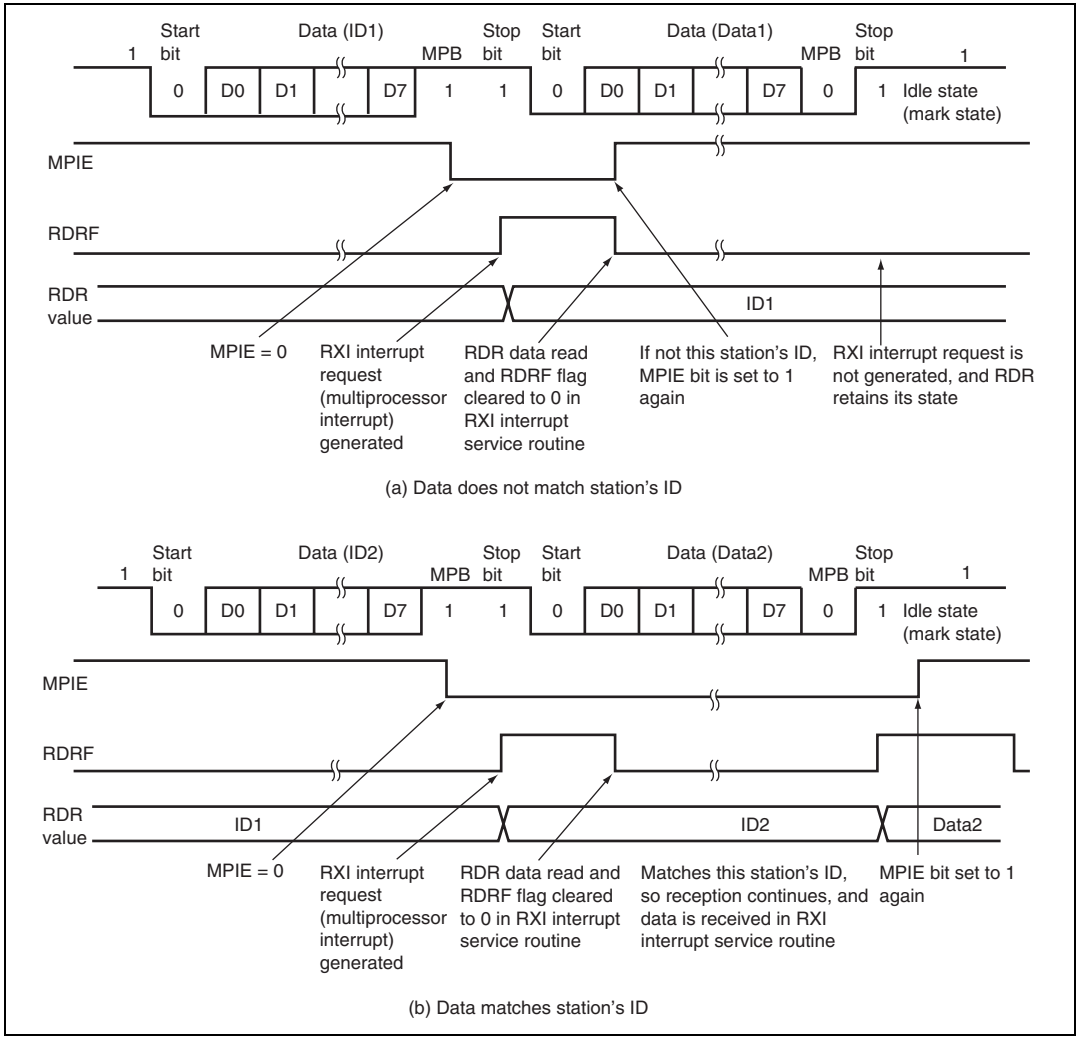


Figure 13.16 Example of SCI Operation in Reception (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)

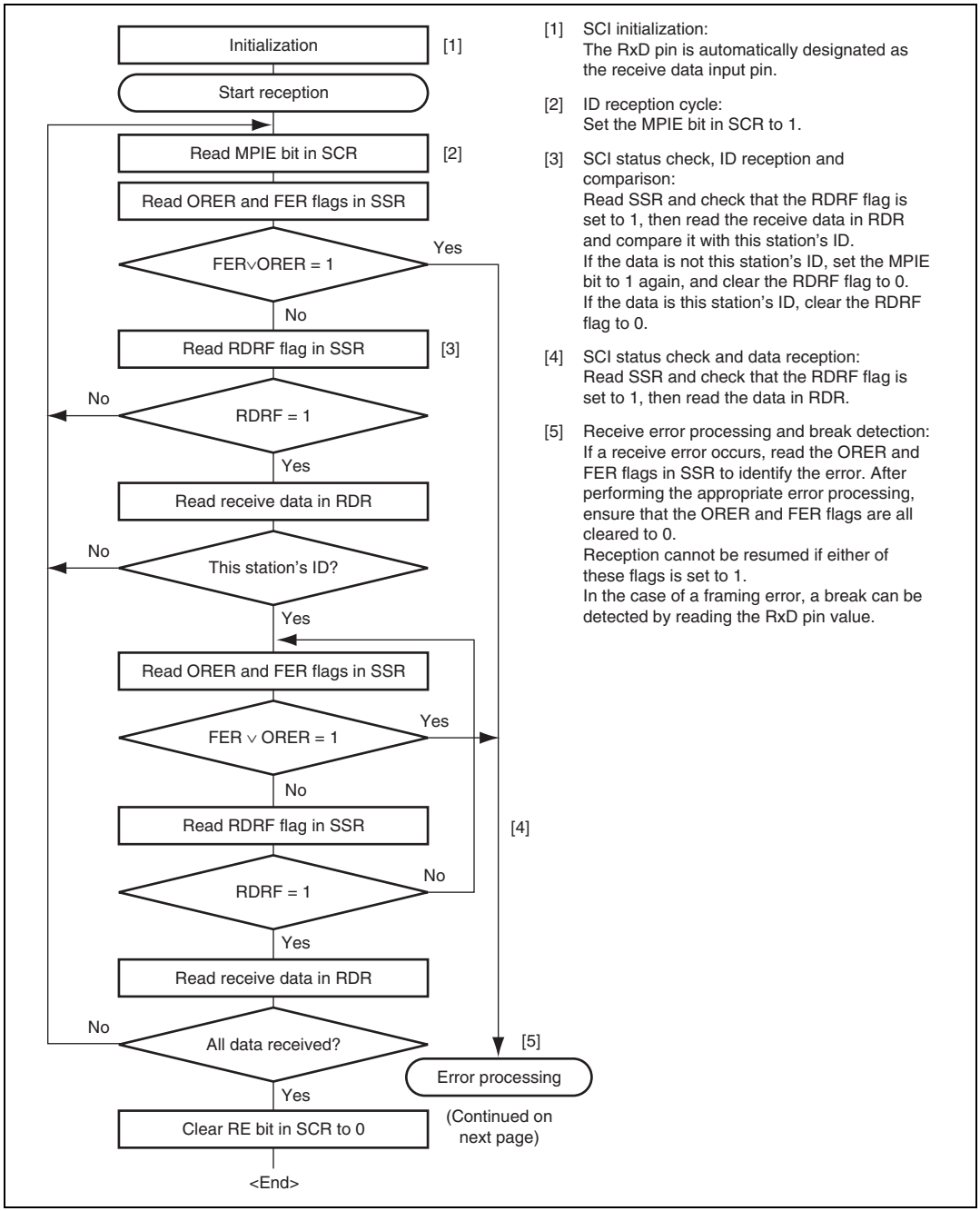


Figure 13.17 Sample Multiprocessor Serial Reception Flowchart (1)

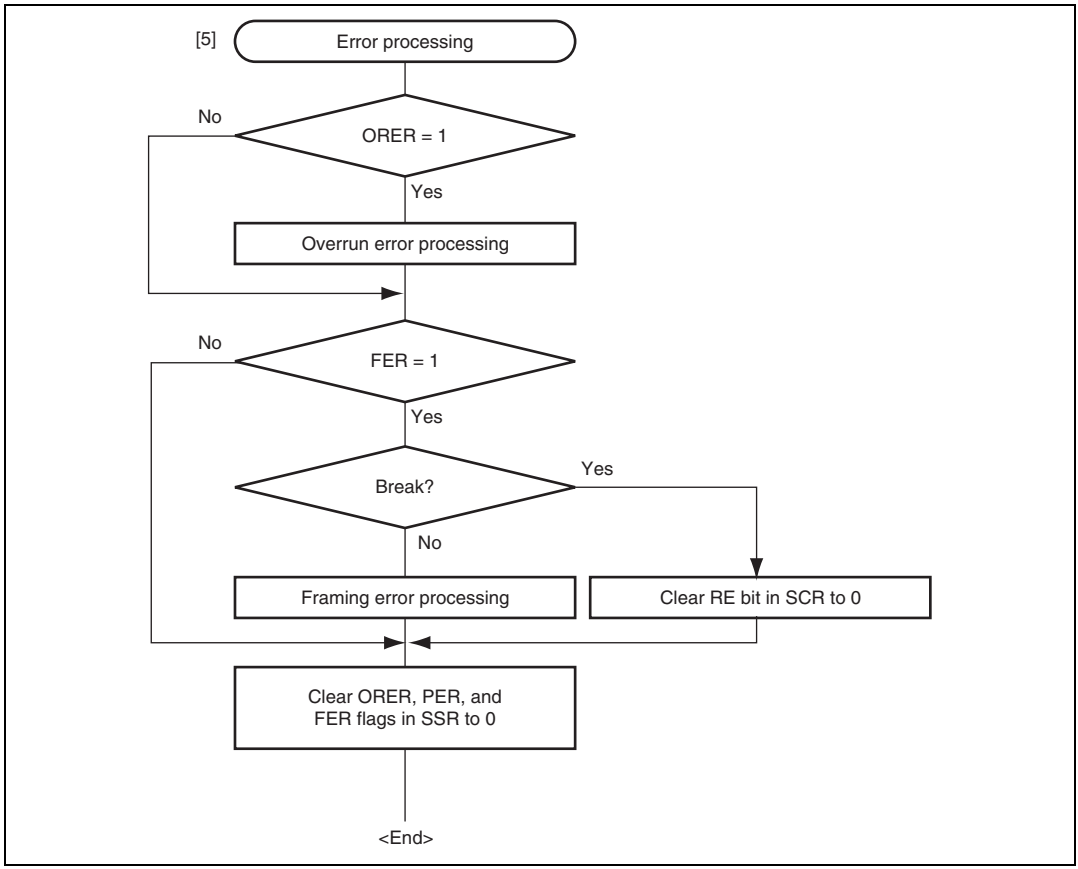


Figure 13.17 Sample Multiprocessor Serial Reception Flowchart (2)

13.6 Operation in Clocked Synchronous Mode

Figure 13.18 shows the general format for clocked synchronous communication. In clocked synchronous mode, data is transmitted or received synchronous with clock pulses. In clocked synchronous serial communication, data on the transmission line is output from one falling edge of the serial clock to the next. In clocked synchronous mode, the SCI receives data in synchronous with the rising edge of the serial clock. After 8-bit data is output, the transmission line holds the MSB state. In clocked synchronous mode, no parity or multiprocessor bit is added. Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communication through the use of a common clock. Both the transmitter and the receiver also have a double-buffered structure, so data can be read from or written during transmission or reception, enabling continuous data transfer.

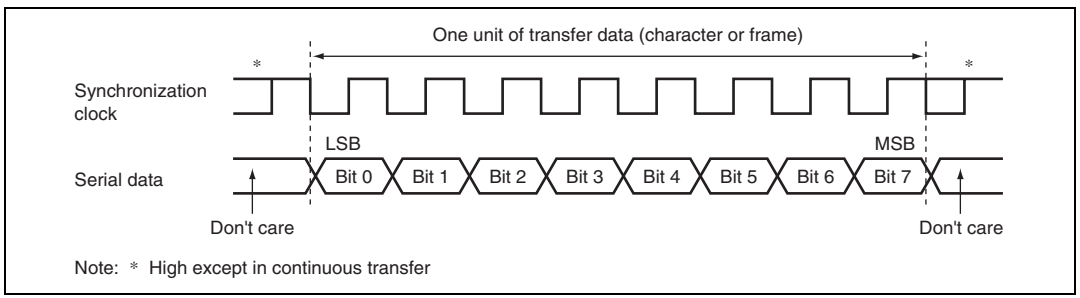


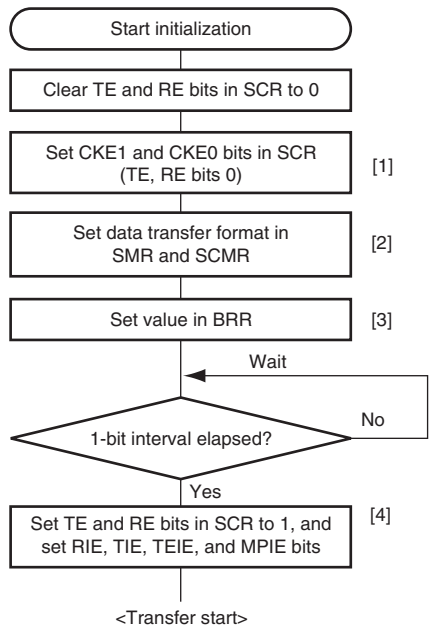
Figure 13.18 Data Format in Synchronous Communication (For LSB-First)

13.6.1 Clock

Either an internal clock generated by the on-chip baud rate generator or an external synchronization clock input at the SCK pin can be selected, according to the setting of CKE0 and CKE1 bits in SCR. 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.

13.6.2 SCI Initialization (Clocked Synchronous Mode)

Before transmitting and receiving data, the TE and RE bits in SCR should be cleared to 0, then the SCI should be initialized as described in a sample flowchart in figure 13.19. When the operating mode, or transfer format, is changed for example, 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. 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.



[1] Set the clock selection in SCR. Be sure to clear bits RIE, TIE, TEIE, MPIE, TE, and RE, to 0.

[2] Set the data transfer format in SMR and SCMR.

[3] Write a value corresponding to the bit rate to BRR. Not necessary if an external clock is used.

[4] Wait at least one bit interval, then set the TE bit or RE bit in SCR to 1. Also set the RIE, TIE TEIE, and MPIE bits. Setting the TE and RE bits enables the TxD and RxD pins to be used.

Note: In simultaneous transmit and receive operations, the TE and RE bits should both be cleared to 0 or set to 1 simultaneously.

Figure 13.19 Sample SCI Initialization Flowchart

13.6.3 Serial Data Transmission (Clocked Synchronous Mode)

Figure 13.20 shows an example of SCI operation for transmission in clocked synchronous mode. In serial transmission, the SCI operates as described below.

1. The SCI monitors the TDRE flag in SSR, and if the flag is 0, the SCI recognizes that data has been written to TDR, and transfers the data from TDR to TSR.

2. After transferring data from TDR to TSR, the SCI sets the TDRE flag to 1 and starts transmission. If the TIE bit in SCR is set to 1 at this time, a transmit data empty interrupt (TXI) is generated. Continuous transmission is possible because the TXI interrupt routine writes the next transmit data to TDR before transmission of the current transmit data has been completed.
3. 8-bit data is sent from the TxD pin synchronized with the output clock when output clock mode has been specified, and synchronized with the input clock when use of an external clock has been specified.
4. The SCI checks the TDRE flag at the timing for sending the MSB (bit 7).
5. If the TDRE flag is cleared to 0, data is transferred from TDR to TSR, and serial transmission of the next frame is started.
6. If the TDRE flag is set to 1, the TEND flag in SSR is set to 1, and the TDRE flag maintains the output state of the last bit. If the TEIE bit in SCR is set to 1 at this time, a TEI interrupt request is generated. The SCK pin is fixed high.

Figure 13.21 shows a sample flow chart for serial data transmission. Even if the TDRE flag is cleared to 0, transmission will not start while a receive error flag (ORER, FER, or PER) is set to 1. Make sure that the receive error flags are cleared to 0 before starting transmission. Note that clearing the RE bit to 0 does not clear the receive error flags.

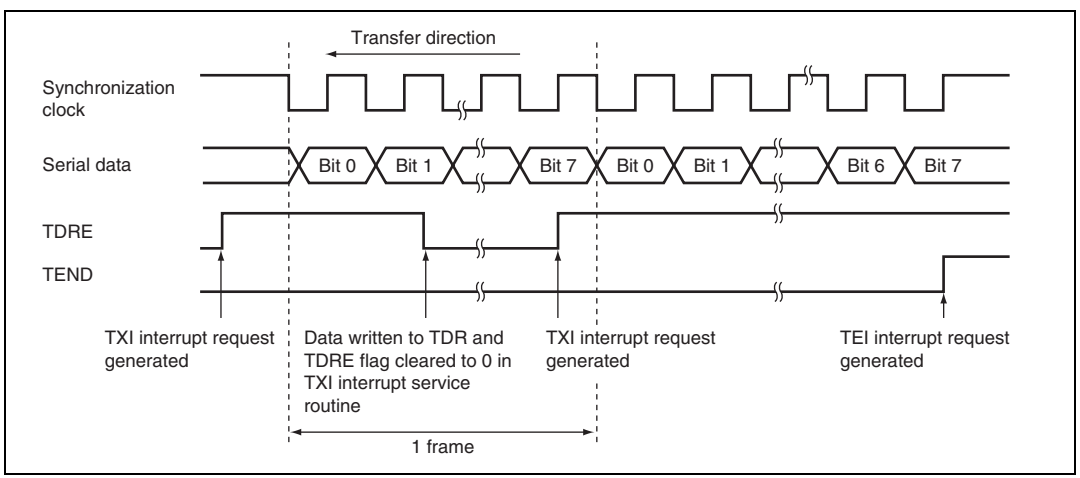
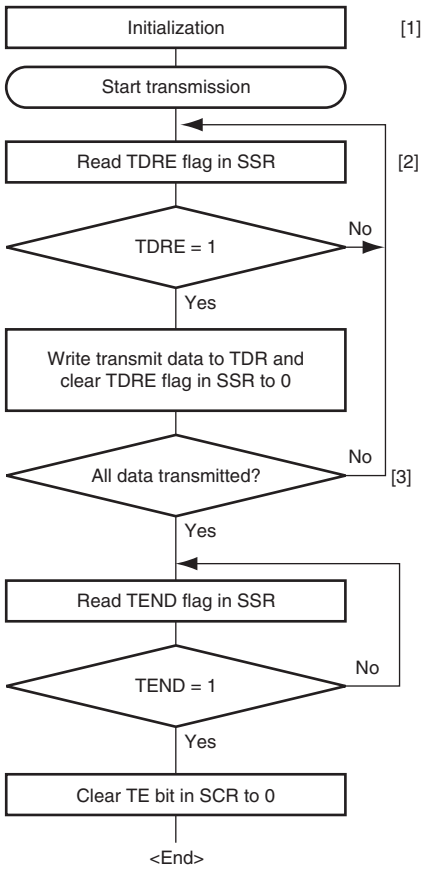


Figure 13.20 Sample SCI Transmission Operation in Clocked Synchronous Mode



[1] SCI initialization:
The TxD pin is automatically designated as the transmit data output pin.

[2] SCI status check and transmit data write:
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR and clear the TDRE flag to 0.

[3] Serial transmission continuation procedure:
To continue serial transmission, be sure to read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and then clear the TDRE flag to 0.
Checking and clearing of the TDRE flag is automatic when the DMAC or the DTC* is activated by a transmit data empty interrupt (TXI) request and data is written to TDR.

Note: * Checking and clearing of the TDRE flag are performed automatically by the DTC when the DTC's DISEL bit is cleared to 0 and the transfer counter value is not 0. Consequently, it is necessary to use the CPU to clear the TDRE flag if DISEL is set to 1 or if the transfer counter value is 0.

Figure 13.21 Sample Serial Transmission Data Flowchart

13.6.4 Serial Data Reception (Clocked Synchronous Mode)

Figure 13.22 shows an example of SCI operation for reception in clocked synchronous mode. In serial reception, the SCI operates as described below.

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

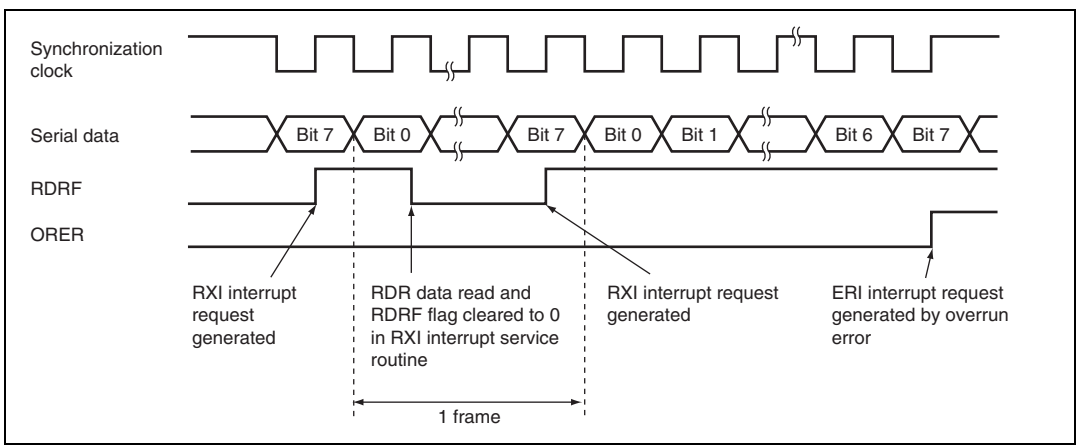


Figure 13.22 Example of SCI Operation in Reception

Reception cannot be resumed while a receive error flag is set to 1. Accordingly, clear the ORER, FER, PER, and RDRF bits to 0 before resuming reception. Figure 13.23 shows a sample flow chart for serial data reception.

When the internal clock is selected during reception, the synchronization clock will be output until an overrun error occurs or the RE bit is cleared. To receive data in frame units, a dummy data of one frame must be transmitted simultaneously.

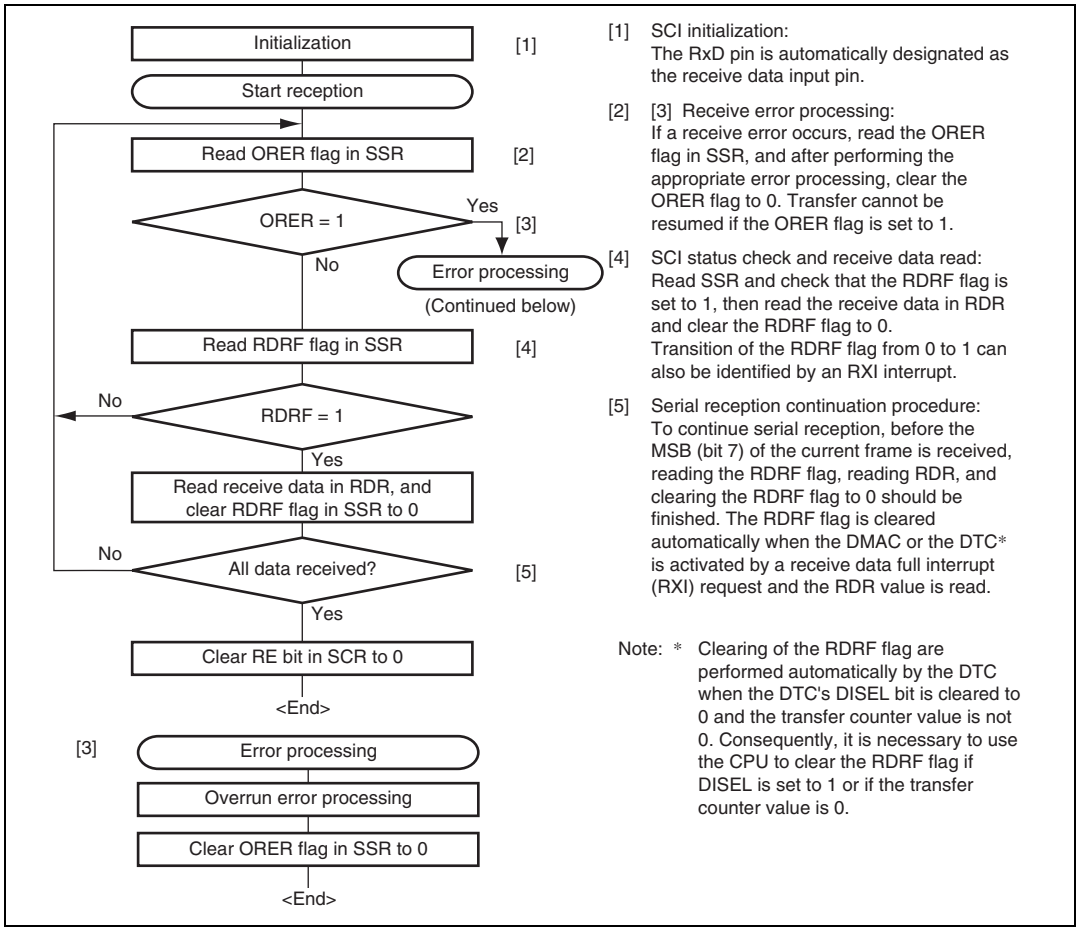
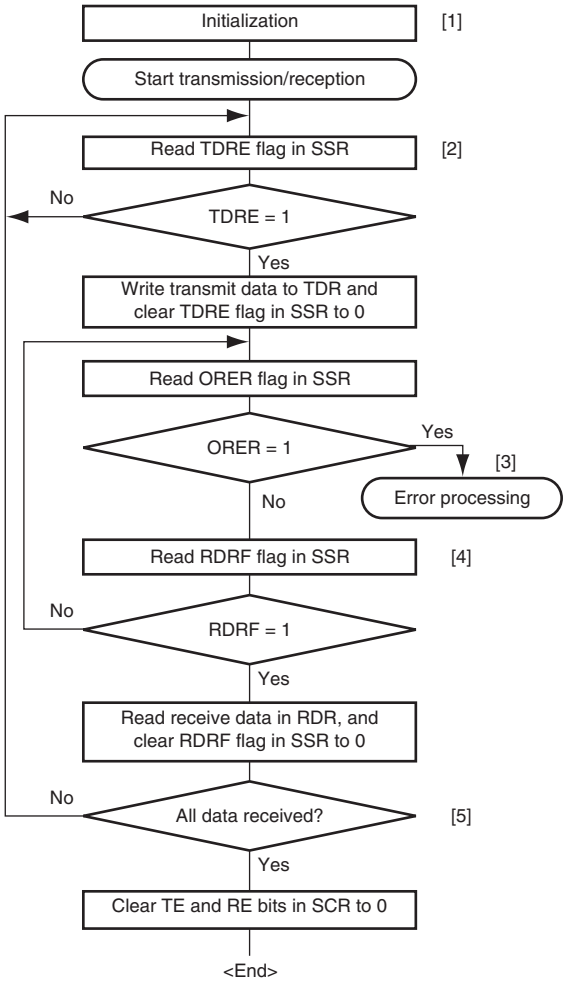


Figure 13.23 Sample Serial Reception Flowchart

13.6.5 Simultaneous Serial Data Transmission and Reception (Clocked Synchronous Mode)

Figure 13.24 shows a sample flowchart for simultaneous serial transmit and receive operations. The following procedure should be used for simultaneous serial data transmit and receive operations. To switch from transmit mode to simultaneous transmit and receive mode, after checking that the SCI has finished transmission and the TDRE and TEND flags are set to 1, clear TE to 0. Then simultaneously set TE and RE to 1 with a single instruction. To switch from receive mode to simultaneous transmit and receive mode, after checking that the SCI has finished reception, clear RE to 0. Then after checking that the RDRF and receive error flags (ORER, FER, and PER) are cleared to 0, simultaneously set TE and RE to 1 with a single instruction.



Notes: When switching from transmit or receive operation to simultaneous transmit and receive operations, first clear the TE bit and RE bit to 0, then set both these bits to 1 simultaneously.

* The TDRE and RDRF flags are automatically cleared by the DTC when the DTC's DISEL bit is cleared to 0 and the transfer counter value is not 0. Consequently, it is necessary to use the CPU to clear the TDRE and RDRF flags if DISEL is set to 1 or if the transfer counter value is 0.

[1] SCI initialization:
The TxD pin is designated as the transmit data output pin, and the RxD pin is designated as the receive data input pin, enabling simultaneous transmit and receive operations.

[2] SCI status check and transmit data write:
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR and clear the TDRE flag to 0. Transition of the TDRE flag from 0 to 1 can also be identified by a TXI interrupt.

[3] Receive error processing:
If a receive error occurs, read the ORER flag in SSR, and after performing the appropriate error processing, clear the ORER flag to 0. Transmission/reception cannot be resumed if the ORER flag is set to 1.

[4] SCI status check and receive data read:
Read SSR and check that the RDRF flag is set to 1, then read the receive data in RDR and clear the RDRF flag to 0. Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial transmission/reception continuation procedure:
To continue serial transmission/reception, before the MSB (bit 7) of the current frame is received, finish reading the RDRF flag, reading RDR, and clearing the RDRF flag to 0. Also, before the MSB (bit 7) of the current frame is transmitted, read 1 from the TDRE flag to confirm that writing is possible. Then write data to TDR and clear the TDRE flag to 0.
Checking and clearing of the TDRE flag is automatic when the DMAC or the 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 the DTC* is activated by a receive data full interrupt (RXI) request and the RDR value is read.

Figure 13.24 Sample Flowchart of Simultaneous Serial Transmit and Receive Operations

13.7 Operation in Smart Card Interface

The SCI supports an IC card (Smart Card) interface that conforms 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 mode is carried out by means of a register setting.

13.7.1 Pin Connection Example

Figure 13.25 shows an example of connection with the Smart Card. In communication with an IC card, as both transmission and reception are carried out on a single data transmission line, the TxD pin and RxD pin should be connected to the LSI pin. The data transmission line should be pulled up to the V_{CC} power supply with a resistor. 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. 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. This LSI port output is used as the reset signal.

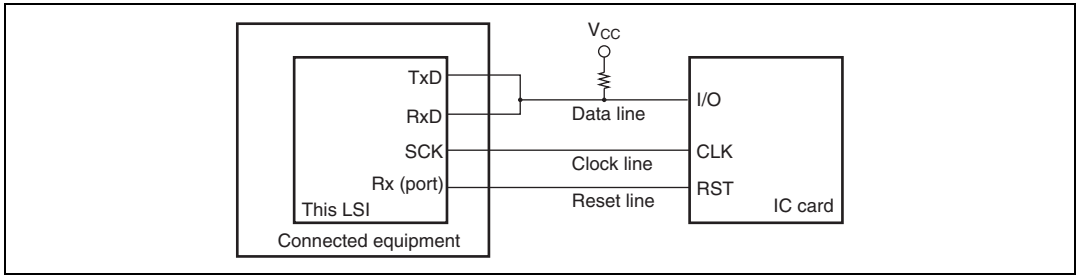


Figure 13.25 Schematic Diagram of Smart Card Interface Pin Connections

13.7.2 Data Format (Except for Block Transfer Mode)

Figure 13.26 shows the transfer data format in Smart Card interface mode.

- One frame consists of 8-bit data plus a parity bit in asynchronous mode.
- In transmission, a guard time of at least 2 etu (Elementary Time Unit: the time for transfer of one 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.
- If an error signal is sampled during transmission, the same data is retransmitted automatically after a delay of 2 etu or longer.

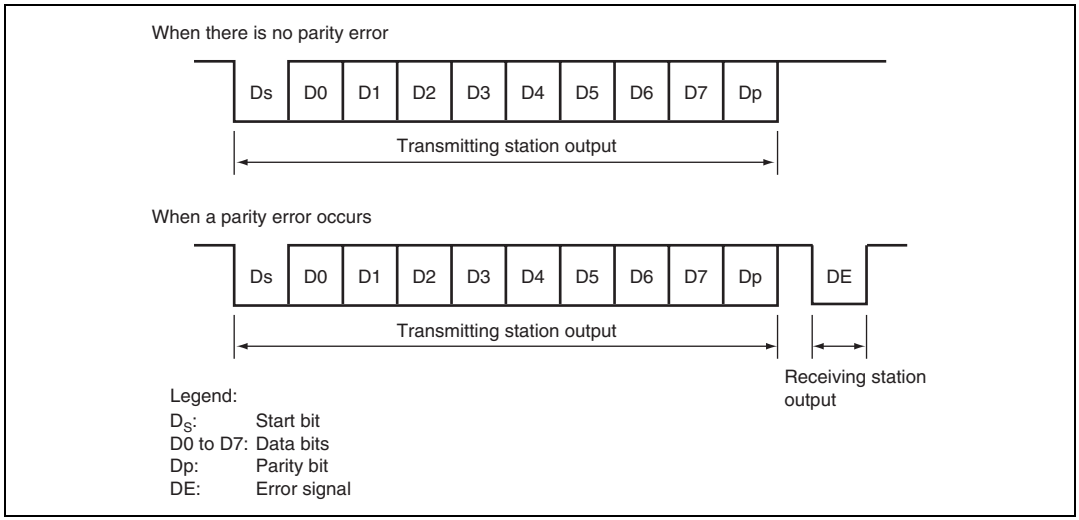


Figure 13.26 Normal Smart Card Interface Data Format

Data transfer with other types of IC cards (direct convention and inverse convention) are performed as described in the following.

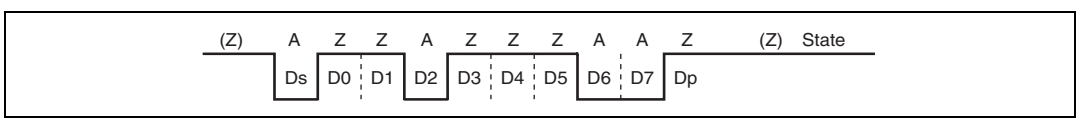


Figure 13.27 Direct Convention (SDIR = SINV = O/$\overline{E}$ = 0)

With the direction convention type IC and the above sample start character, 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. For the direct convention type, clear the SDIR and SINV

bits in SCMR to 0. According to Smart Card regulations, clear the O/$\overline{E}$ bit in SMR to 0 to select even parity mode.

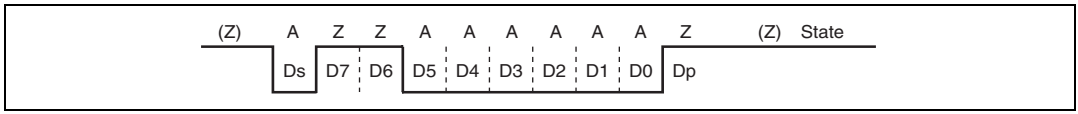


Figure 13.28 Inverse Convention (SDIR = SINV = O/$\overline{E}$ = 1)

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 for the above is H'3F. For the inverse convention type, set the SDIR and SINV bits in SCMR to 1. According to Smart Card regulations, even parity mode is the logic 0 level of the parity bit, and corresponds to state Z. In this LSI, the SINV bit inverts only data bits D0 to D7. Therefore, set the O/$\overline{E}$ bit in SMR to 1 to invert the parity bit for both transmission and reception.

13.7.3 Clock

Only an internal clock which is generated by the on-chip baud rate generator is used as a transmit/receive clock. When an output clock is selected by setting CKE0 to 1, a clock with a frequency S* times the bit rate is output from the SCK pin.

Note: * S is the value shown in section 13.3.12, Bit Rate Register (BRR).

13.7.4 Block Transfer Mode

Operation in block transfer mode is the same as that in the normal Smart Card interface mode, except for the following points.

- In reception, though the parity check is performed, no error signal is output even if an error is detected. However, the PER bit in SSR is set to 1 and must be cleared before receiving the parity bit of the next frame.
- In transmission, a guard time of at least 1 etu is left between the end of the parity bit and the start of the next frame.
- In transmission, because retransmission is not performed, the TEND flag is set to 1, 11.5 etu after transmission start.
- As with the normal Smart Card interface, the ERS flag indicates the error signal status, but since error signal transfer is not performed, this flag is always cleared to 0.

13.7.5 Receive Data Sampling Timing and Reception Margin

In Smart Card interface mode an internal clock generated by the on-chip baud rate generator can only be used as a transmission/reception clock. In this mode, the SCI operates on a basic clock with a frequency of 32, 64, 372, or 256 times the transfer rate (fixed to 16 times in normal asynchronous mode) as determined by bits BCP1 and BCP0. In reception, the SCI samples the falling edge of the start bit using the basic clock, and performs internal synchronization. As shown in figure 13.29, by sampling receive data at the rising-edge of the 16th, 32nd, 186th, or 128th pulse of the basic clock, data can be latched at the middle of the bit. The reception margin is given by the following formula.

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5)\,F - \frac{|D - 0.5|}{N}\,(1 + F) \right| \quad 100\ [\%]$$

Where M: Reception margin (%)
N: Ratio of bit rate to clock (N = 32, 64, 372, and 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 reception margin formula is as follows.

$$M = (0.5 - 1/2 \times 372) \times 100\%$$
$$= 49.866\%$$

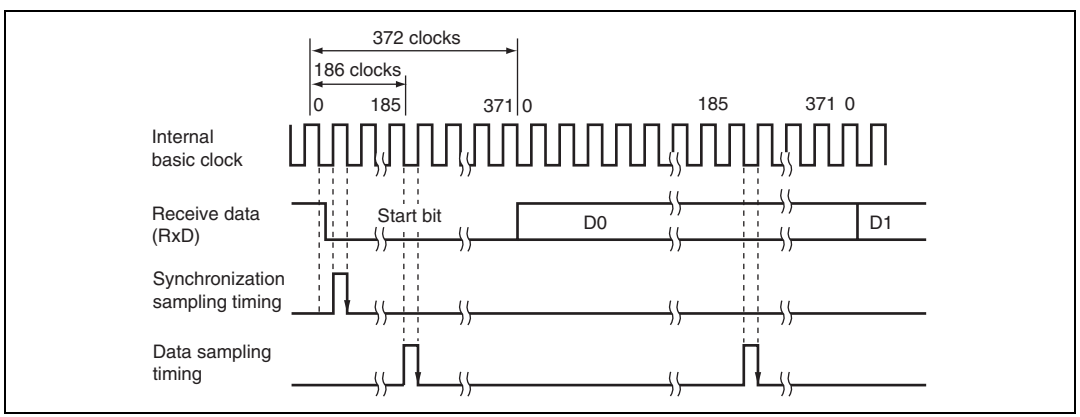


Figure 13.29 Receive Data Sampling Timing in Smart Card Mode (Using Clock of 372 Times the Transfer Rate)

13.7.6 Initialization

Before transmitting and 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/$\overline{E}$, BCP0, BCP1, CKS0, CKS1 bits in SMR. 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 CKE0 and CKE1 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.

To switch from receive mode to transmit mode, after checking that the SCI has finished reception, initialize the SCI, and set RE to 0 and TE to 1. Whether SCI has finished reception or not can be checked with the RDRF, PER, or ORER flags. To switch from transmit mode to receive mode, after checking that the SCI has finished transmission, initialize the SCI, and set TE to 0 and RE to 1. Whether SCI has finished transmission or not can be checked with the TEND flag.

13.7.7 Serial Data Transmission (Except for Block Transfer Mode)

As data transmission in Smart Card interface mode involves error signal sampling and retransmission processing, the operations are different from those in normal serial communication interface mode (except for block transfer mode). Figure 13.30 illustrates the retransfer operation when the SCI is in transmit mode.

1. If an error signal is sent back from the receiving end after transmission of one frame is complete, 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 cleared to 0 by the time the next parity bit is sampled.
2. The TEND bit in SSR is not set for a frame in which an error signal indicating an abnormality is received. Data is retransferred from TDR to TSR, and retransmitted automatically.
3. If an error signal is not sent back from the receiving end, the ERS bit in SSR is not set. 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. Writing transmit data to TDR transfers the next transmit data.

Figure 13.32 shows a flowchart for transmission. A sequence of transmit operations can be performed automatically by specifying the DTC or the DMAC to be activated with a TXI interrupt source. In a transmit operation, the TDRE flag is set to 1 at the same time as the TEND flag in SSR is set, and a TXI interrupt will be generated if the TIE bit in SCR has been set to 1. If the TXI request is designated beforehand as a DTC* or the DMAC activation source, the DTC* or the DMAC 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 is transferred by the DTC* or the DMAC. In the event of an error, the SCI retransmits the same data automatically. During this period, the TEND flag remains cleared to 0 and the DTC* or the DMAC is not activated. Therefore, the SCI and DTC* or the DMAC will automatically transmit the specified number of bytes in the event of an error, including retransmission. 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 the DTC, it is essential to set and enable the DMAC or the DTC* before carrying out SCI setting. For details of the DMAC or the DTC* setting procedures, refer to section 8, Data Transfer Controller (DTC) or section 7, DMA controller (DMAC).

Note: * The Flags are automatically cleared by the DTC when the DTC's DISEL bit is cleared to 0 and the transfer counter value is not 0.

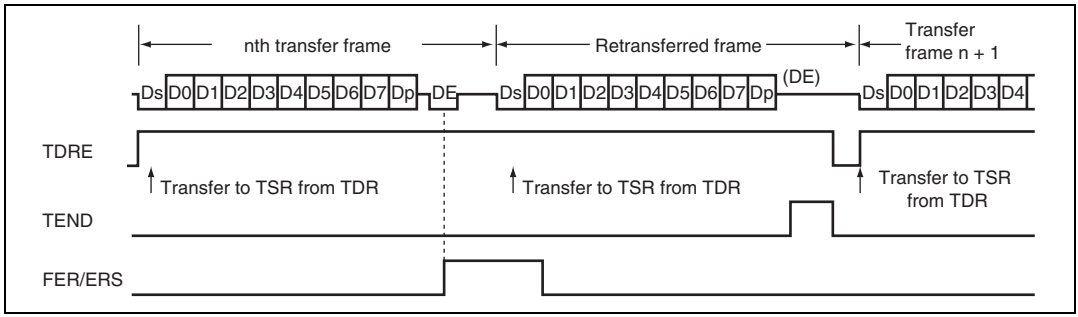


Figure 13.30 Retransfer Operation in SCI Transmit Mode

The timing for setting the TEND flag depends on the value of the GM bit in SMR. The TEND flag set timing is shown in figure 13.31.

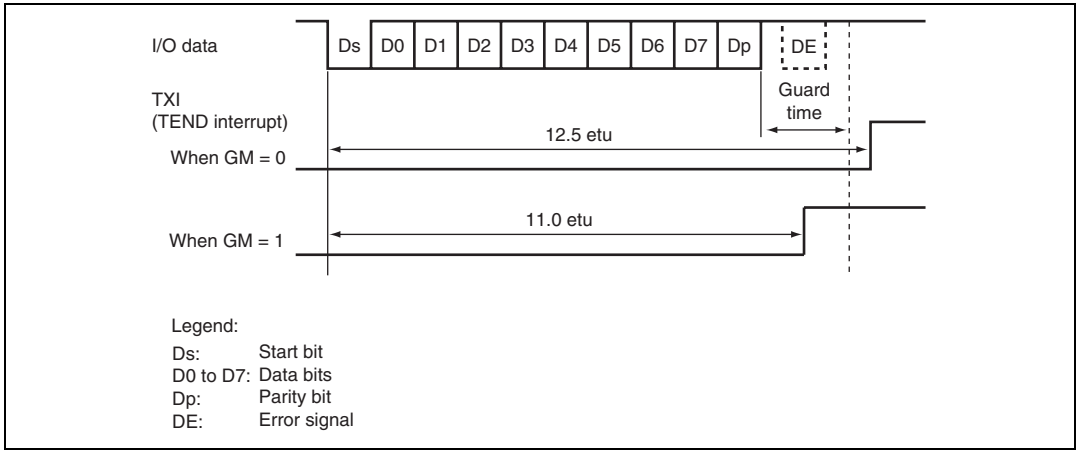


Figure 13.31 TEND Flag Generation Timing in Transmission Operation

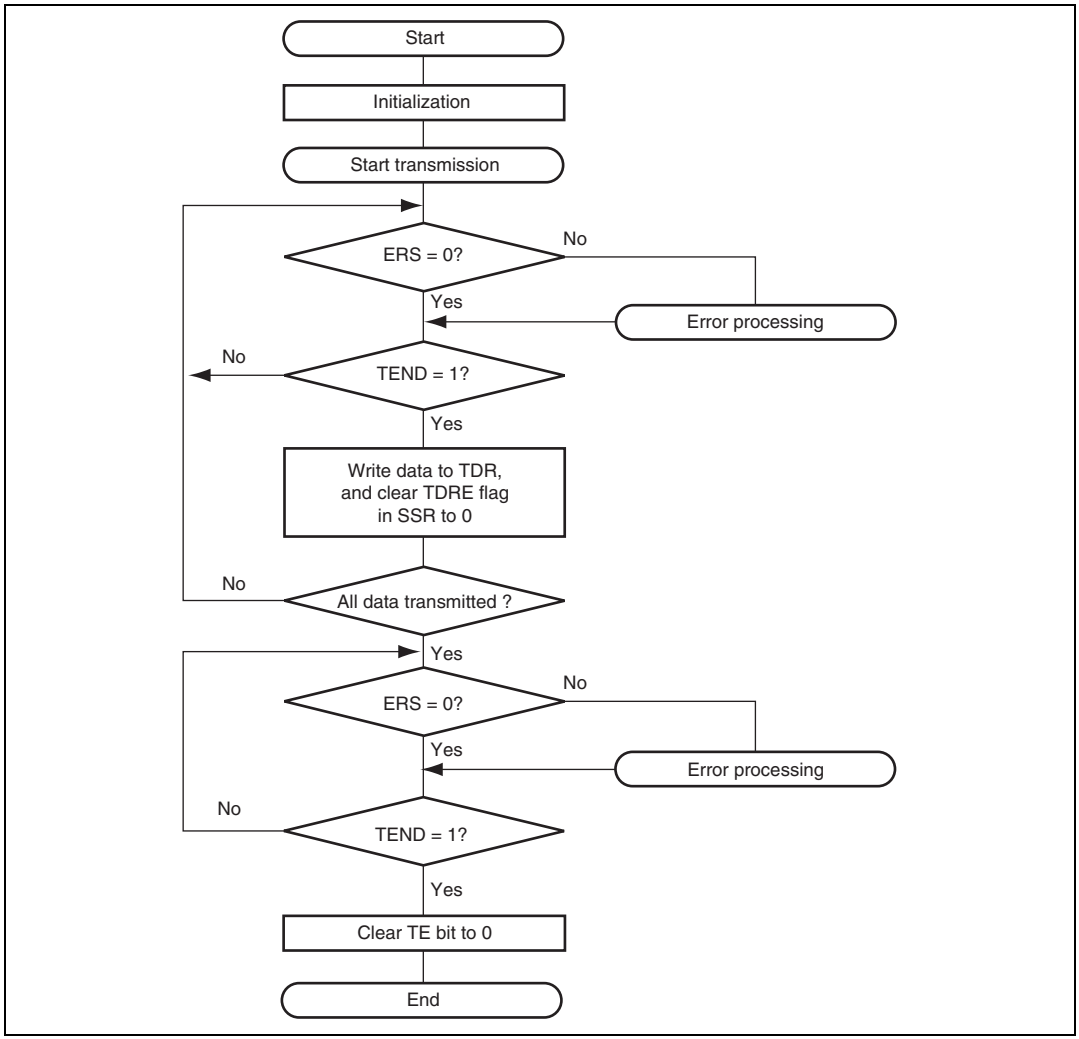


Figure 13.32 Example of Transmission Processing Flow

13.7.8 Serial Data Reception (Except for Block Transfer Mode)

Data reception in Smart Card interface mode uses the same operation procedure as for normal serial communication interface mode. Figure 13.33 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 set 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 to 1, 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.

Figure 13.34 shows a flowchart for reception. A sequence of receive operations can be performed automatically by specifying the DTC* or the DMAC to be activated using an RXI interrupt source. 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 DTC* or the DMAC activation source, the DTC* or the DMAC will be activated by the RXI request, and the receive data will be transferred. The RDRF flag is cleared to 0 automatically when data is transferred by the DTC* or the DMAC. If an error occurs in receive mode and the ORER or PER flag is set to 1, a transfer error interrupt (ERI) request will be generated. Hence, so the error flag must be cleared to 0. In the event of an error, the DTC* or the DMAC is not activated and receive data is skipped. Therefore, receive data is transferred for only the specified number of bytes in the event of an error. Even when a parity error occurs in receive mode and the PER flag is set to 1, the data that has been received is transferred to RDR and can be read from there.

Notes: For details on receive operations in block transfer mode, refer to section 13.4, Operation in Asynchronous Mode.

* The Flags are automatically cleared by the DTC when the DTC's DISEL bit is cleared to 0 and the transfer counter value is not 0.

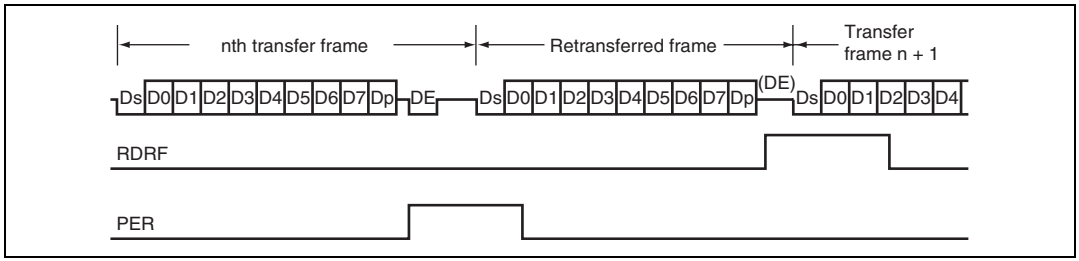


Figure 13.33 Retransfer Operation in SCI Receive Mode

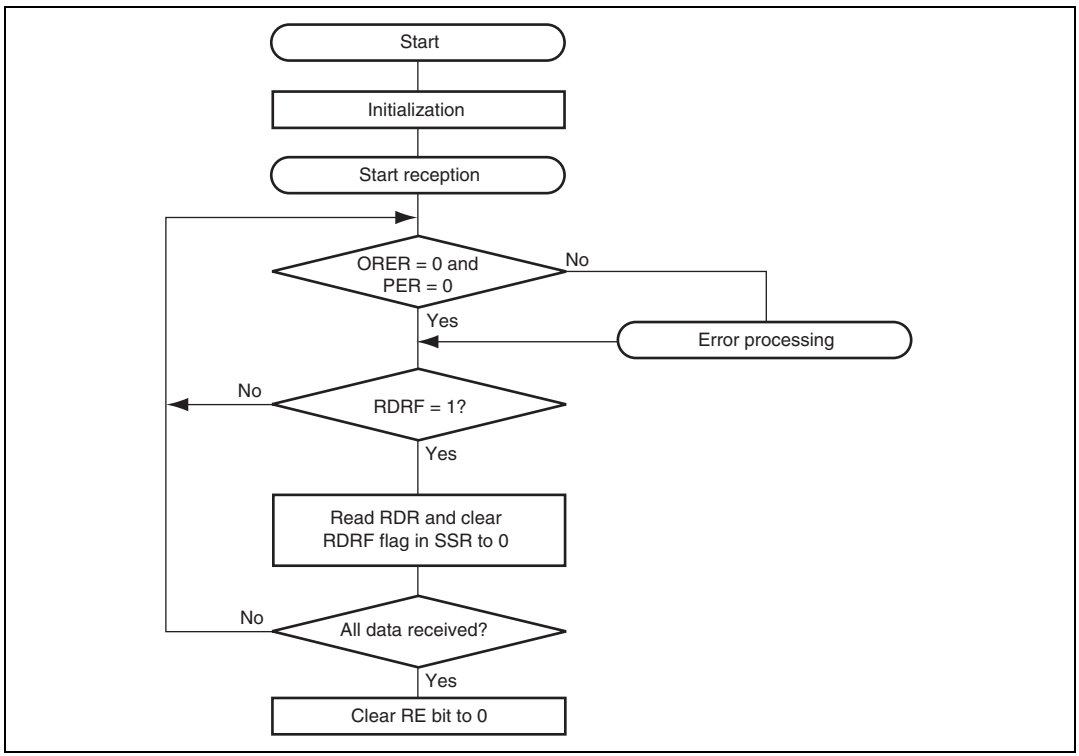


Figure 13.34 Example of Reception Processing Flow

13.7.9 Clock Output Control

When the GM bit in SMR is set to 1, the clock output level can be fixed with bits CKE0 and CKE1 in SCR. At this time, the minimum clock pulse width can be made the specified width. Figure 13.35 shows the timing for fixing the clock output level. In this example, GM is set to 1, CKE1 is cleared to 0, and the CKE0 bit is controlled.

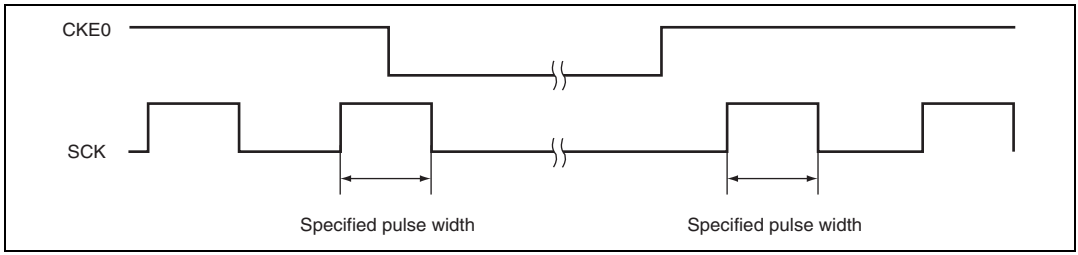


Figure 13.35 Timing for Fixing Clock Output Level

When turning on the power or switching between Smart Card interface mode and software standby mode, the following procedures should be followed in order to maintain the clock duty.

Powering On: To secure clock duty from power-on, the following switching procedure should be followed.

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.

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 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. Make the transition to the software standby state.

When returning to smart card interface mode from software standby mode

1. Exit the software standby state.
2. Write 1 to the CKE0 bit in SCR and output the clock. Signal generation is started with the normal duty.

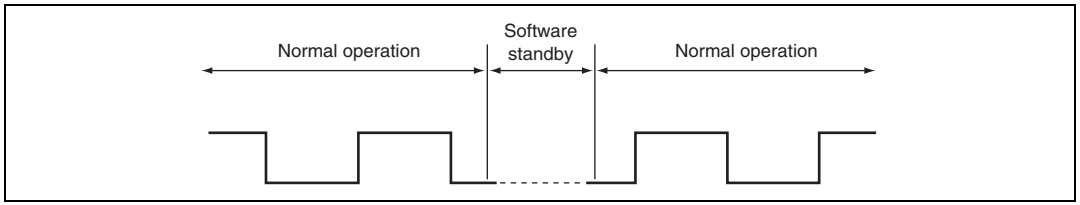


Figure 13.36 Clock Halt and Restart Procedure

13.8 SCI Select Function

The SCI_0 supports the SCI select function which allows clock synchronous communication between master LSI and one of multiple slave LSI. Figure 13.37 shows an example of communication using the SCI select function. Figure 13.38 shows the operation.

The master LSI can communicate with slave LSI_A by bringing $\overline{\text{SEL\_A}}$ and $\overline{\text{SEL\_B}}$ signals low and high, respectively. In this case, the TxD0_B pin of the slave LSI_B is brought high-impedance state and the internal SCK0_A signal is fixed high. This halts the communication operation of slave LSI_B. The master LSI can communicate with slave LSI_B by bringing the $\overline{\text{SEL\_A}}$ and $\overline{\text{SEL\_B}}$ signals high and low, respectively.

The slave LSI detects the selection by receiving the low level input from the $\overline{\text{IRQ7}}$ pin and immediately executes data transmission/reception processing.

Note: The selection signals ($\overline{\text{SEL\_A}}$ and $\overline{\text{SEL\_B}}$) of the LSI must be switched while the serial clock (M_SCK) is high after the end bit of the transmit data has been send. Note that one selection signal can be brought low at the same time.

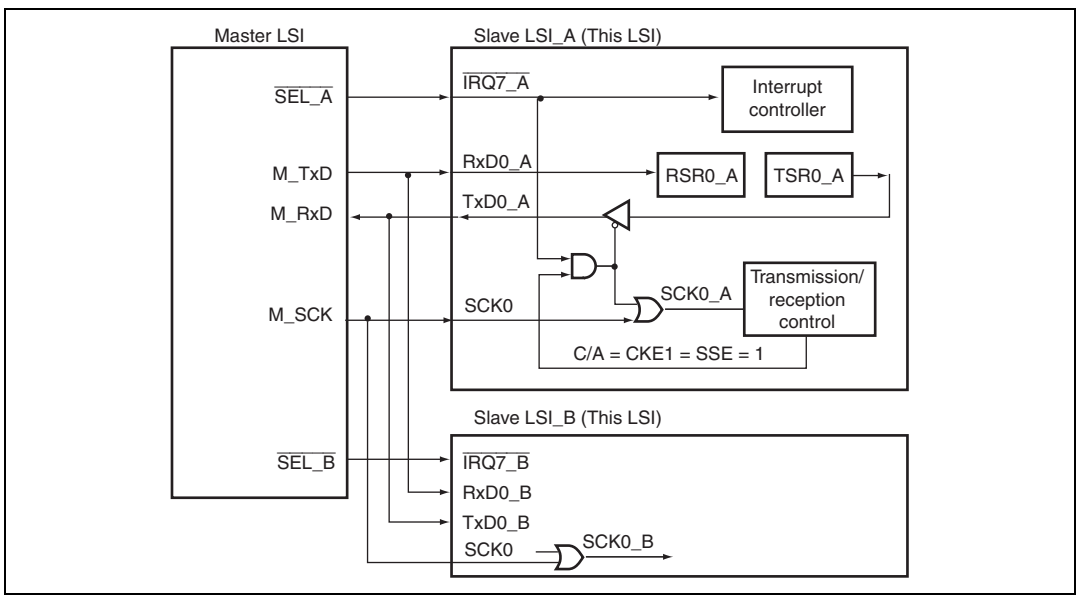


Figure 13.37 Example of Communication Using the SCI Select Function

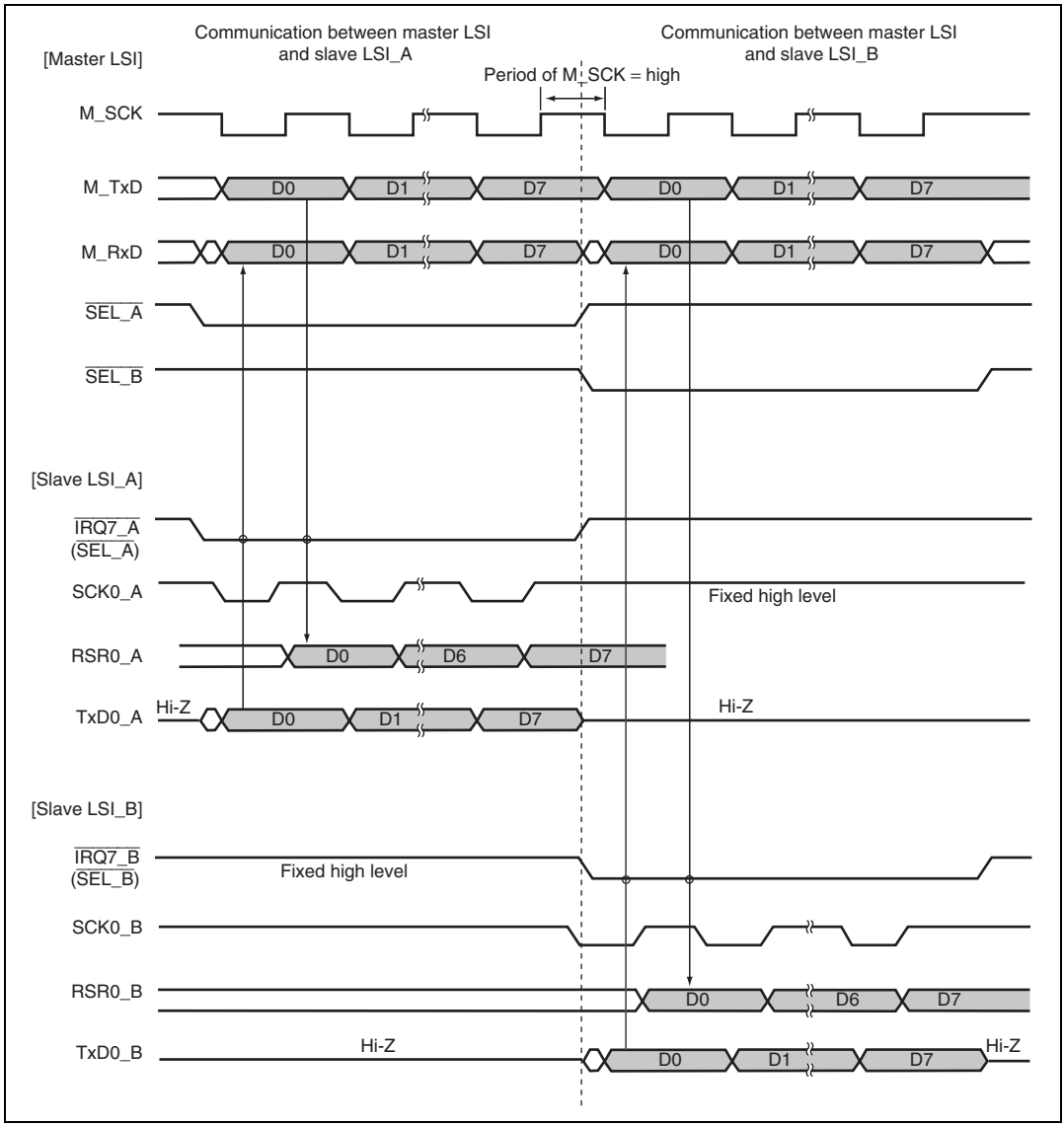


Figure 13.38 Example of Communication Using the SCI Select Function

13.9 Interrupts

13.9.1 Interrupts in Normal Serial Communication Interface Mode

Table 13.12 shows the interrupt sources in normal serial communication interface mode. A different interrupt vector is assigned to each interrupt source, and individual interrupt sources can be enabled or disabled using the enable bits in SCR.

When the TDRE flag in SSR is set to 1, a TXI interrupt request is generated. When the TEND flag in SSR is set to 1, a TEI interrupt request is generated. A TXI interrupt can activate the DMAC or the DTC to perform data transfer. The TDRE flag is cleared to 0 automatically when data is transferred by the DMAC or the DTC*.

When the RDRF flag in SSR is set to 1, an RXI interrupt request is generated. When the ORER, PER, or FER flag in SSR is set to 1, an ERI interrupt request is generated. An RXI interrupt request can activate the DMAC or the DTC to transfer data. The RDRF flag is cleared to 0 automatically when data is transferred by the DMAC or the DTC*.

A TEI interrupt is requested when the TEND flag is set to 1 and the TEIE bit is set to 1. If a TEI interrupt and a TXI interrupt are requested simultaneously, the TXI interrupt has priority for acceptance. However, if the TDRE and TEND flags are cleared simultaneously by the TXI interrupt routine, the SCI cannot branch to the TEI interrupt routine later.

Note: * The Flags are automatically cleared by the DTC when the DTC's DISEL bit is cleared to 0 and the transfer counter value is not 0.

Table 13.12 SCI Interrupt Sources

Channel	Name	Interrupt Source	Interrupt Flag	DTC Activation	DMAC Activation	Priority*
0	ERI0	Receive Error	ORER, FER, PER	Not possible	Not possible	High
	RXI0	Receive Data Full	RDRF	Possible	Possible	
	TXI0	Transmit Data Empty	TDRE	Possible	Possible	
	TEI0	Transmission End	TEND	Not possible	Not possible	
1	ERI1	Receive Error	ORER, FER, PER	Not possible	Not possible	
	RXI1	Receive Data Full	RDRF	Possible	Possible	
	TXI1	Transmit Data Empty	TDRE	Possible	Possible	
	TEI1	Transmission End	TEND	Not possible	Not possible	
2	ERI2	Receive Error	ORER, FER, PER	Not possible	Not possible	
	RXI2	Receive Data Full	RDRF	Possible	Not possible	
	TXI2	Transmit Data Empty	TDRE	Possible	Not possible	
	TEI2	Transmission End	TEND	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.

13.9.2 Interrupts in Smart Card Interface Mode

Table 13.13 shows the interrupt sources in Smart Card interface mode. The transmit end interrupt (TEI) request cannot be used in this mode.

Note: In case of block transfer mode, see section 13.9.1, Interrupts in Normal Serial Communication Interface Mode.

Table 13.13 Interrupt Sources in Smart Card Interface Mode

Channel	Name	Interrupt Source	Interrupt Flag	DTC Activation	DMAC Activation	Priority*
0	ERI0	Receive Error, detection	ORER, PER, ERS	Not possible	Not possible	High
	RXI0	Receive Data Full	RDRF	Possible	Possible	
	TXI0	Transmit Data Empty	TEND	Possible	Possible	
1	ERI1	Receive Error, detection	ORER, PER, ERS	Not possible	Not possible	
	RXI1	Receive Data Full	RDRF	Possible	Possible	
	TXI1	Transmit Data Empty	TEND	Possible	Possible	
2	ERI2	Receive Error, detection	ORER, PER, ERS	Not possible	Not possible	
	RXI2	Receive Data Full	RDRF	Possible	Not possible	
	TXI2	Transmit Data Empty	TEND	Possible	Not possible	Low

Notes: * Indicates the initial state immediately after a reset. Priorities in channels can be changed by the interrupt controller.

13.10 Usage Notes

13.10.1 Break Detection and Processing (Asynchronous Mode Only)

When framing error detection is performed, a break can be detected by reading the RxD pin value directly. In a break, the input from the RxD pin becomes all 0s, setting the FER flag, and possibly the PER flag. Note that as 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.

13.10.2 Mark State and Break Detection (Asynchronous Mode Only)

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

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

Transmission cannot be started when a receive error flag (ORER, PER, or FER) is set 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.

13.10.4 Restrictions on Use of DMAC or DTC

- 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 the DTC. Misoperation may occur if the transmit clock is input within 4 ϕ clocks after TDR is updated. (figure 13.39)
- When RDR is read by the DMAC or the DTC, be sure to set the activation source to the relevant SCI reception end interrupt (RXI).
- During data transfer, the TDRE and RDRF flags are automatically cleared by the DTC when the DTC's DISEL bit is cleared to 0 and the transfer counter value is not 0. Consequently, it is necessary to use the CPU to clear the TDRE and RDRF flags if DISEL is set to 1 or if the transfer counter value is 0.

In particular, data transmission cannot be completed correctly unless the TDRE flag is cleared using the CPU.

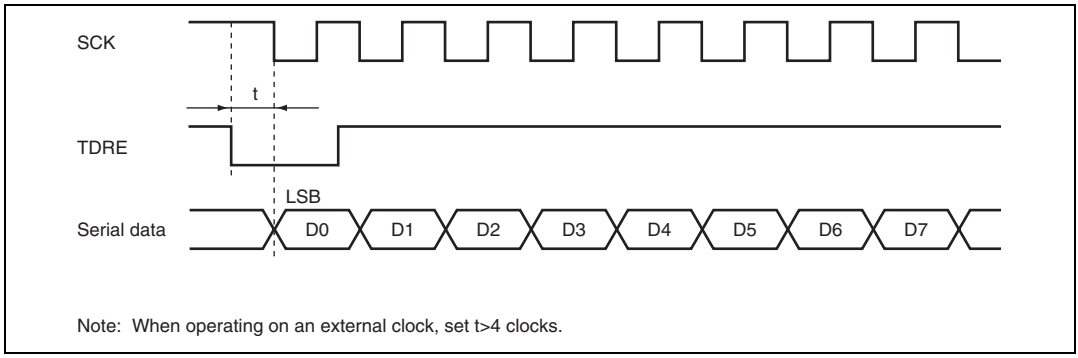


Figure 13.39 Example of Clocked Synchronous Transmission by DMAC or DTC

13.10.5 Operation in Case of Mode Transition

- Transmission

  Operation should be stopped (by clearing TE, TIE, and TEIE to 0) before making a module stop mode, software standby mode, or subsleep mode transition. TSR, TDR, and SSR are reset. The output pin states in module stop mode, software standby mode, or subsleep 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 13.40 shows a sample flowchart for mode transition during transmission. Port pin states are shown in figures 13.41 and 13.42.

  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.

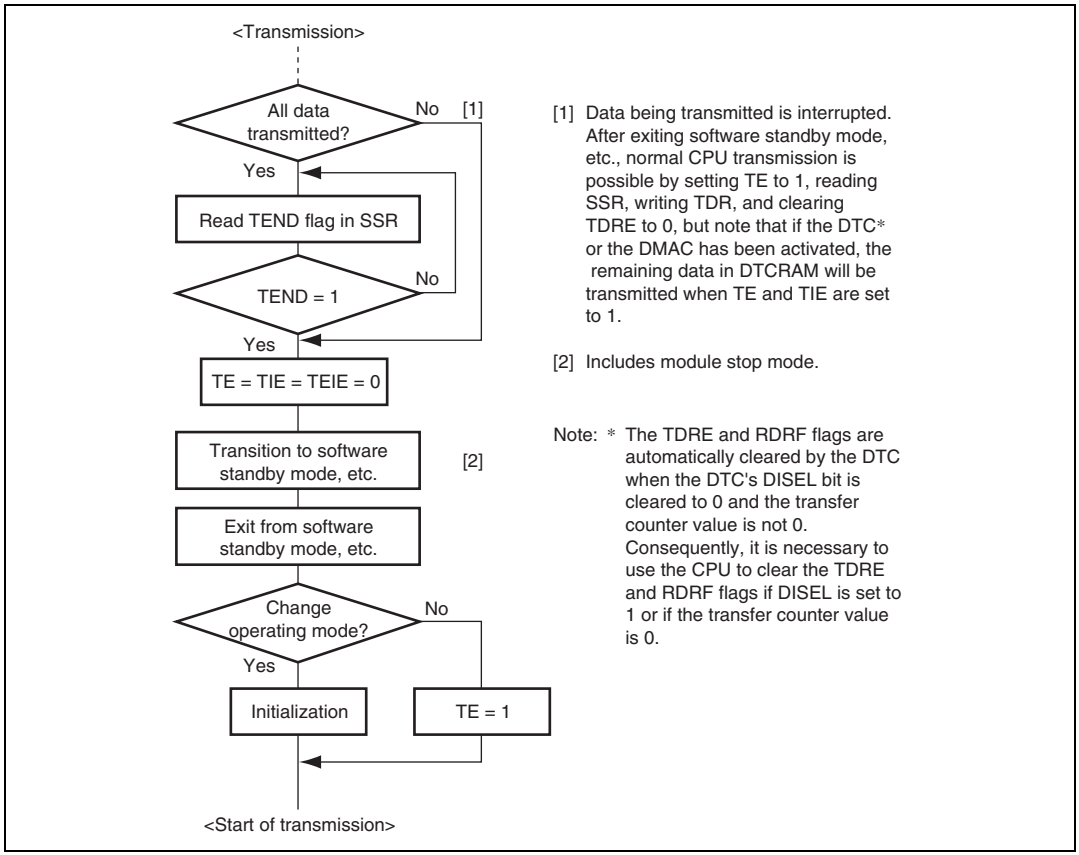


Figure 13.40 Sample Flowchart for Mode Transition during Transmission

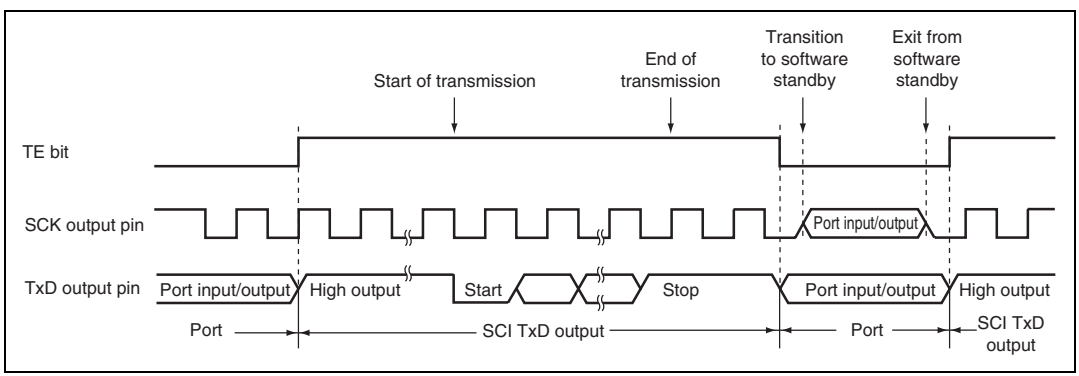


Figure 13.41 Port Pin State of Asynchronous Transmission Using Internal Clock

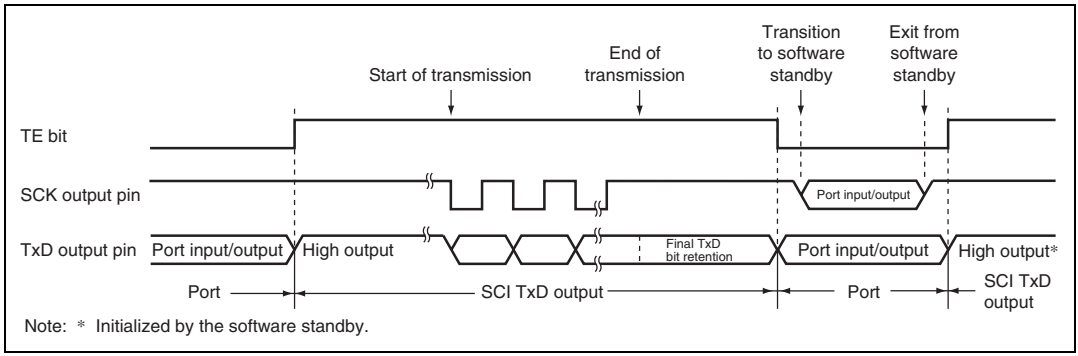


Figure 13.42 Port Pin State of Synchronous Transmission Using Internal Clock

- Reception

  Receive operation should be stopped (by clearing RE to 0) before making a module stop mode, software standby mode, watch mode, subactive mode, or subsleep 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 13.43 shows a sample flowchart for mode transition during reception.

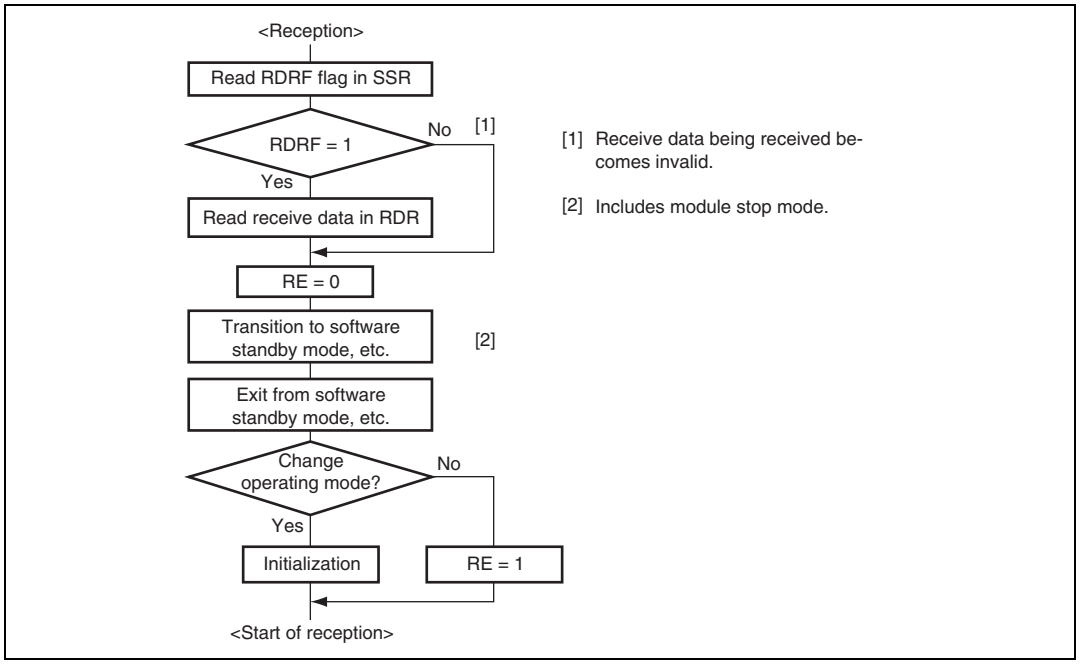


Figure 13.43 Sample Flowchart for Mode Transition during Reception

13.10.6 Switching from SCK Pin Function to Port Pin Function

When switching the SCK pin function to the output port function (high-level output) by making the following settings while DDR = 1, DR = 1, C/$\overline{A}$ = 1, CKE1 = 0, CKE0 = 0, and TE = 1 (synchronous mode), low-level output occurs for one half-cycle.

1. End of serial data transmission
2. TE bit = 0
3. C/$\overline{A}$ bit = 0 ... switchover to port output
4. Occurrence of low-level output (see figure 13.44)

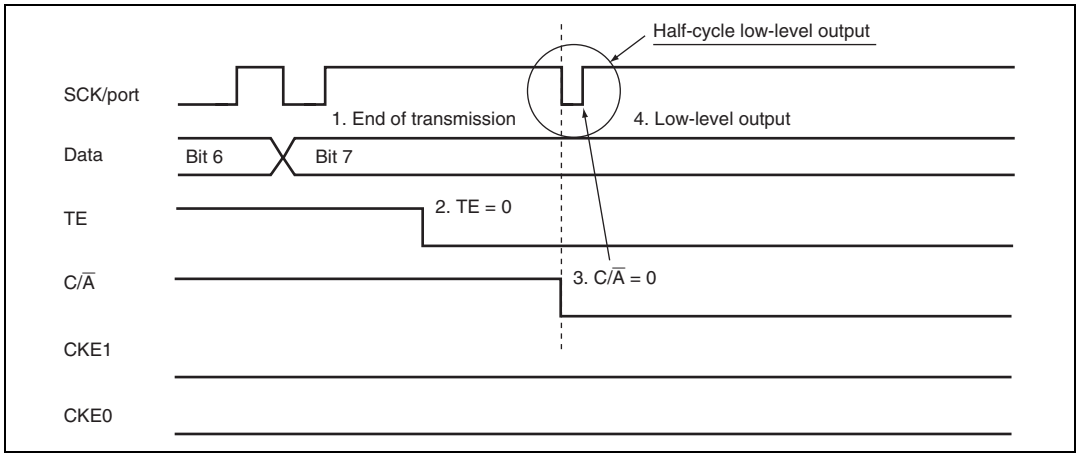


Figure 13.44 Operation when Switching from SCK Pin Function to Port Pin Function

Sample Procedure for Avoiding Low-Level Output: As this sample procedure temporarily places the SCK pin in the input state, the SCK/port pin should be pulled up beforehand with an external circuit.
With DDR = 1, DR = 1, C/$\overline{A}$ = 1, CKE1 = 0, CKE0 = 0, and TE = 1, make the following settings in the order shown.

1. End of serial data transmission
2. TE bit = 0
3. CKE1 bit = 1
4. C/$\overline{A}$ bit = 0 ... switchover to port output
5. CKE1 bit = 0

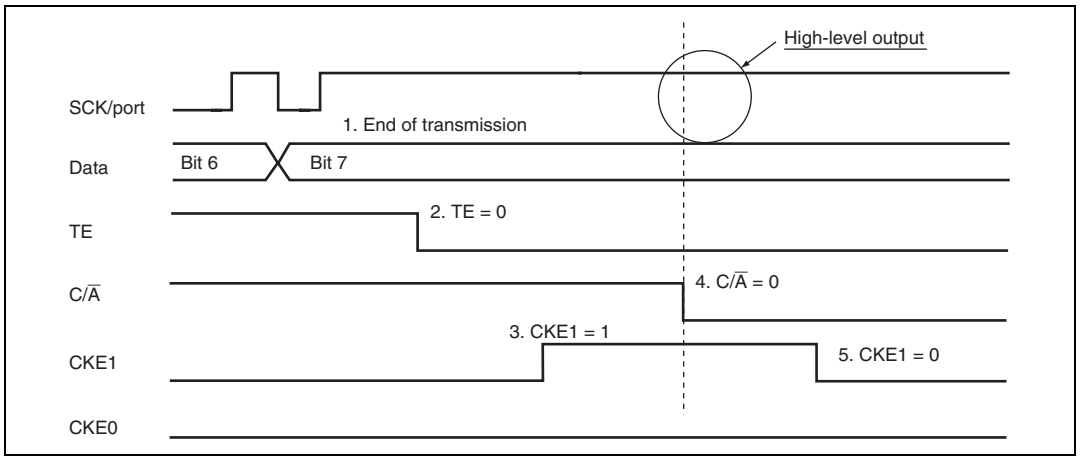


Figure 13.45 Operation when Switching from SCK Pin Function to Port Pin Function (Example of Preventing Low-Level Output)

13.10.7 Module Stop Mode Setting

Operation of the SCI can be disabled or enabled using the module stop control register. The initial setting is for operation of the SCI to be halted. Register access is enabled by clearing module stop mode. For details, refer to section 22, Power-Down Modes.

Section 14 Boundary Scan Function

This LSI incorporates a boundary scan function, which is a serial I/O interface based on the JTAG (Joint Test Action Group, IEEEStd.1149.1 and IEEE Standard Test Access Port and Boundary Scan Architecture). Figure 14.1 shows the block diagram of the boundary scan function.

14.1 Features

- Five test signals
  - TCK, TDI, TDO, TMS, $\overline{\text{TRST}}$
- Six test modes supported
  - BYAPASS, SAMPLE/PRELOAD, EXTEST, CLAMP, HIGHZ, IDCODE
- Boundary scan function cannot be performed on the following pins.
  - Power supply pins: VCC, VSS, Vref, AVCC, AVSS, PLLVCC, PLLVSS, PLLCAP, DrVCC, DrVSS
  - Clock signals: EXTAL, XTAL, EXTAL48, XTAL48
  - Analog signals: P40 to P43, P96, P97, USD+, USD-
  - Boundary scan signals: TCK, TDI, TDO, TMS, $\overline{\text{TRST}}$
  - E10A signal ($\overline{\text{EMLE}}$)

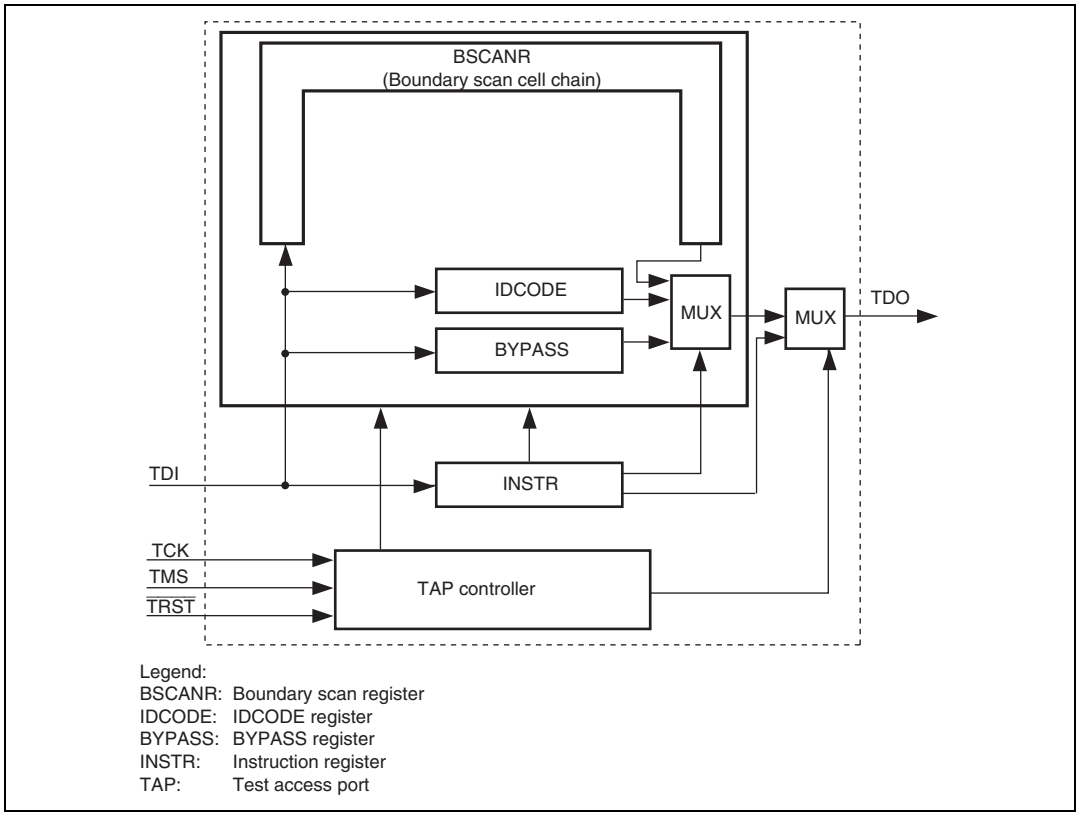


Figure 14.1 Block Diagram of Boundary Scan Function

14.2 Pin Configuration

Table 14.1 shows the I/O pins used in the boundary scan function.

Table 14.1 Pin Configuration

Pin Name	I/O	Function
TMS	Input	Test Mode Select Controls the TAP controller which is a 16-state Finite State Machine. The TMS input value at the rising edge of TCK determines the status transition direction on the TAP controller. The TMS is fixed high when the boundary scan function is not used. The protocol is based on JTAG standard (IEEE Std.1149.1). This pin has a pull-up resistor.
TCK	Input	Test Clock A clock signal for the boundary scan function. When the boundary scan function is used, input a clock of 50% duty to this pin. This pin has a pull-up resistor.
TDI	Input	Test Data Input A data input signal for the boundary scan function. Data input from the TDI is latched at the rising edge of TCK. TDI is fixed high when the boundary scan function is not used. This pin has a pull-up resistor.
TDO	Output	Test Data Output A data output signal for the boundary scan function. Data output from the TDO changes at the falling edge of TCK. The output driver of the TDO is driven only when it is necessary only in Shift-IR or Shift-DR states, and is brought to the high-impedance state when not necessary.
$\overline{\text{TRST}}$	Input	Test Reset Asynchronously resets the TAP controller when $\overline{\text{TRST}}$ is brought low. The user must apply power-on reset signal specific to the boundary scan function when the power is supplied (For details on signal design, see section 14.5, Usage Notes). This pin has a pull-up resistor.

14.3 Register Descriptions

The boundary scan function has the following registers. These registers cannot be accessed by the CPU.

- Instruction register (INSTR)
- IDCODE register (IDCODE)
- BYPASS register (BYPASS)
- Boundary scan register (BSCANR)

14.3.1 Instruction Register (INSTR)

INSTR is a 3-bit register. At initialization, this register is specified to IDCODE mode. When $\overline{\text{TRST}}$ is pulled low, or when the TAP controller is in the Test-Logic-Reset state, INSTR is initialized. INSTR can be written by the serial data input from the TDI. If more than three bits of instruction is input from the TDI, INSTR stores the last three bits of serial data.

If a command reserved in INSTR is used, the correct operation cannot be guaranteed.

Bit	Bit Name	Initial Value	R/W	Description
2	TI2	1	—	Test Instruction Bits
1	TI1	0	—	Instruction configuration is shown in table 14.2.
0	TI0	1	—	

Table 14.2 Instruction configuration

Bit 2	Bit1	Bit 0		
TI2	TI1	TI0	Instruction	
0	0	0	EXTEST	
0	0	1	SAMPLE/PRELOAD	
0	1	0	CLAMP	
0	1	1	HIGHZ	
1	0	0	Reserved	
1	0	1	IDCODE	(initial value)
1	1	0	Reserved	
1	1	1	BYPASS	

EXTEST: The EXTEST instruction is used to test external circuits when this LSI is installed on the print circuit board. If this instruction is executed, output pins are used to output test data (specified by the SAMPLE/PRELOAD instruction) from the boundary scan register to the print circuit board, and input pins are used to input test results.

SAMPLE/PRELOAD: The SAMPLE/PRELOAD instruction is used to input data from the LSI internal circuits to the boundary scan register, output data from scan path, and reload the data to the scan path. While this instruction is executed, input signals are directly input to the LSI and output signals are also directly output to the external circuits. The LSI system circuit is not affected by this instruction.

In SAMPLE operation, the boundary scan register latches the snap shot of data transferred from input pins to internal circuit or data transferred from internal circuit to output pins. The latched data is read from the scan path. The scan register latches the snap data at the rising edge of the TCK in Capture-DR state. The scan register latches snap shot without affecting the LSI normal operation.

In PRELOAD operation, initial value is written from the scan path to the parallel output latch of the boundary scan register prior to the EXTEST instruction execution. If the EXTEST is executed without executing this RELOAD operation, undefined values are output from the beginning to the end (transfer to the output latch) of the EXTEST sequence. (In EXTEST instruction, output parallel latches are always output to the output pins.)

CLAMP: When the CLAMP instruction is selected output pins output the boundary scan register value which was specified by the SAMPLE/PRELOAD instruction in advance. While the CLAMP instruction is selected, the status of boundary scan register is maintained regardless of the TAP controller state. BYPASS is connected between TDI and TDO, the same operation as BYPASS instruction can be achieved.

HIGHZ: When the HIGHZ instruction is selected, all outputs enter high-impedance state. While this instruction is selected, the status of boundary scan register is maintained regardless of the TAP controller state. BYPASS resistor is connected between TDI and TDO, the same operation as BYPASS instruction can be achieved.

IDCODE : When the IDCODE instruction is selected, IDCODE register value is output to the TDO in Shift-DR state of TAP controller. In this case, IDCODE register value is output from the LSB. During this instruction execution, test circuit does not affect the system circuit. INSTR is initialized by the IDCODE instruction in Test-Logic-Reset state of TAP controller.

BYPASS: The BYPASS instruction is a standard instruction necessary to operate bypass register. The BYPASS instruction improves the serial data transfer speed by bypassing the scan path. During this instruction execution, test circuit does not affect the system circuit.

14.3.2 IDCODE Register (IDCODE)

IDCODE is a 32-bit register. If INSTR is set to IDCODE mode, IDCODE is connected between TDI and TDO. The HD64F2215 and H8S/2215U output fixed code H'0002200F, HD6432215A output fixed code H'0003200F, HD6432215B output fixed code H'001B200F, HD6432215C output fixed code H'001C200F, HD64F2215R and HD64F2215RU output fixed code H'08030447, and HD64F2215T and HD64F2215TU output fixed code H'08031447, respectively, from the TDO. Serial data cannot be written to IDCODE through TDI. Table 14.3 shows the IDCODE configuration.

Table 14.3 IDCODE Register Configuration

Bits	31 to 28	27 to 12	11 to 1	0
HD64F2215 code HD64F2215U code	0000	0000 0000 0010 0010	0000 0000 111	1
HD6432215A code	0000	0000 0000 0011 0010	0000 0000 111	1
HD6432215B code	0000	0000 0001 1011 0010	0000 0000 111	1
HD6432215C code	0000	0000 0001 1100 0010	0000 0000 111	1
HD64F2215R code	0000	1000 0000 0011 0001	0100 0100 011	1
HD64F2215RU code	0000	1000 0000 0011 0000	0100 0100 011	1
HD64F2215T and HD64F2215TU code	0000	1000 0000 0011 0001	0100 0100 011	1
Contents	Version (4 bits)	Part No. (16 bits)	Product No. (11 bits)	Fixed code (1 bit)

14.3.3 BYPASS Register (BYPASS)

BYPASS is a 1-bit register. If INSTR is specified to BYPASS mode, CLAMP mode, or HIGHZ mode, BYPASS is connected between TDI and TDO.

14.3.4 Boundary Scan Register (BSCANR)

BSCAN is a 217-bit shift register assigned on the pins to control input/output pins.

The I/O pins consists of three bits (IN, Control, OUT), input pins 1 bit (IN), and output pins 1 bit (OUT) of shift registers. The boundary scan test based on the JTAG standard can be performed by using instructions listed in table 14.2. Table 14.4 shows the correspondence between the LSI pins and boundary scan registers. (In table 14.4, Control indicates the high active pin. By specifying Control to high, the pin is driven by OUT.) Figure 14.2 shows the boundary scan register configuration example.

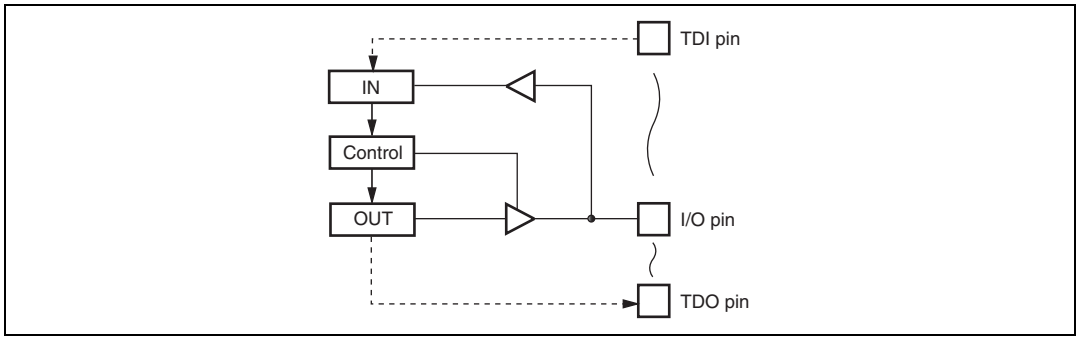


Figure 14.2 Boundary Scan Register Configuration

Table 14.4 Correspondence between LSI Pins and Boundary Scan Register

TFP-120, TFP-120V Pin No.	BP-112, BP-112V Pin No.	Pin Name	I/O	Bit Name
		From TDI		
111	A4	PE0/D0	IN	216
			Control	215
			OUT	214
113	D5	PE1/D1	IN	213
			Control	212
			OUT	211
115	B4	PE2/D2	IN	210
			Control	209
			OUT	208
116	A3	PE3/D3	IN	207
			Control	206
			OUT	205
117	C4	PE4/D4	IN	204
			Control	203
			OUT	202
118	B3	PE5/D5	IN	201
			Control	200
			OUT	199
119	A2	PE6/D6	IN	198
			Control	197
			OUT	196
120	C3	PE7/D7	IN	195
			Control	194
			OUT	193
2	B2	PD0/D8	IN	192
			Control	191
			OUT	190
3	B1	PD1/D9	IN	189
			Control	188
			OUT	187

TFP-120, TFP-120V Pin No.	BP-112, BP-112V Pin No.	Pin Name	I/O	Bit Name
4	D4	PD2/D10	IN	186
			Control	185
			OUT	184
5	C2	PD3/D11	IN	183
			Control	182
			OUT	181
6	C1	PD4/D12	IN	180
			Control	179
			OUT	178
7	D3	PD5/D13	IN	177
			Control	176
			OUT	175
8	D2	PD6/D14	IN	174
			Control	173
			OUT	172
9	D1	PD7/D15	IN	171
			Control	170
			OUT	169
11	E3	PC0/A0	IN	168
			Control	167
			OUT	166
13	E2	PC1/A1	IN	165
			Control	164
			OUT	163
14	F3	PC2/A2	IN	162
			Control	161
			OUT	160
15	F1	PC3/A3	IN	159
			Control	158
			OUT	157
16	F2	PC4/A4	IN	156
			Control	155
			OUT	154

TFP-120, TFP-120V Pin No.	BP-112, BP-112V Pin No.	Pin Name	I/O	Bit Name
17	F4	PC5/A5	IN	153
			Control	152
			OUT	151
18	G1	PC6/A6	IN	150
			Control	149
			OUT	148
19	G2	PC7/A7	IN	147
			Control	146
			OUT	145
20	G3	PB0/A8	IN	144
			Control	143
			OUT	142
21	H1	PB1/A9	IN	141
			Control	140
			OUT	139
23	G4	PB2/A10	IN	138
			Control	137
			OUT	136
25	H2	PB3/A11	IN	135
			Control	134
			OUT	133
26	J1	PB4/A12	IN	132
			Control	131
			OUT	130
27	H3	PB5/A13	IN	129
			Control	128
			OUT	127
28	J2	PB6/A14	IN	126
			Control	125
			OUT	124
29	K1	PB7/A15	IN	123
			Control	122
			OUT	121

TFP-120, TFP-120V Pin No.	BP-112, BP-112V Pin No.	Pin Name	I/O	Bit Name
30	J3	PA0/A16	IN	120
			Control	119
			OUT	118
31	K2	PA1/A17/TxD2	IN	117
			Control	116
			OUT	115
32	L2	PA2/A18/RxD2	IN	114
			Control	113
			OUT	112
33	H4	PA3/A19/SCK2/SUSPND	IN	111
			Control	110
			OUT	109
35	K3	P10/TIOCA0/A20/VM	IN	108
			Control	107
			OUT	106
36	L3	P11/TIOCB0/A21/VP	IN	105
			Control	104
			OUT	103
37	J4	P12/TIOCC0/TCLKA/A22/RCV	IN	102
			Control	101
			OUT	100
38	K4	P13/TIOCD0/TCLKB/A23/VPO	IN	99
			Control	98
			OUT	97
39	L4	P14/TIOCA1/$\overline{\text{IRQ0}}$	IN	96
			Control	95
			OUT	94
40	H5	P15/TIOCB1/TCLKC/FSE0	IN	93
			Control	92
			OUT	91
41	J5	P16/TIOCA2/$\overline{\text{IRQ1}}$	IN	90
			Control	89
			OUT	88

TFP-120, TFP-120V Pin No.	BP-112, BP-112V Pin No.	Pin Name	I/O	Bit Name
42	L5	P17/TIOCB2/TCLKD/$\overline{OE}$	IN	87
			Control	86
			OUT	85
53	H7	USPND	OUT	84
55	K8	VBUS	IN	83
56	L9	$\overline{UBPM}$	IN	82
67	H9	MD0	IN	81
68	H10	MD1	IN	80
69	H11	FWE	IN	79
70	G8	NMI	IN	78
71	G9	$\overline{STBY}$	IN	77
72	G11	RES	IN	76
77	F8	MD2	IN	75
78	E11	PF7/ϕ	IN	74
			Control	73
			OUT	72
79	E10	PF6/$\overline{AS}$	IN	71
			Control	70
			OUT	69
80	E9	PF5/$\overline{RD}$	IN	68
			Control	67
			OUT	66
81	D11	PF4/$\overline{HWR}$	IN	65
			Control	64
			OUT	63
83	E8	PF3/$\overline{LWR}$/$\overline{ADTRG}$/$\overline{IRQ3}$	IN	62
			Control	61
			OUT	60
85	D10	PF2/$\overline{WAIT}$	IN	59
			Control	58
			OUT	57
86	C11	PF1/$\overline{BACK}$	IN	56
			Control	55
			OUT	54

TFP-120, TFP-120V Pin No.	BP-112, BP-112V Pin No.	Pin Name	I/O	Bit Name
87	D9	PF0/$\overline{\text{BREQ}}$/$\overline{\text{IRQ2}}$	IN	53
			Control	52
			OUT	51
88	C10	P30/TxD0	IN	50
			Control	49
			OUT	48
89	B11	P31/RxD0	IN	47
			Control	46
			OUT	45
90	C9	P32/SCK0/$\overline{\text{IRQ4}}$	IN	44
			Control	43
			OUT	42
91	B10	P33/TxD1	IN	41
			Control	40
			OUT	39
92	A10	P34/RxD1	IN	38
			Control	37
			OUT	36
93	D8	P35/SCK1/$\overline{\text{IRQ5}}$	IN	35
			Control	34
			OUT	33
94	B9	P36	IN	32
			Control	31
			OUT	30
96	A9	P74/$\overline{\text{MRES}}$	IN	29
			Control	28
			OUT	27
97	C8	P73/TMO1/$\overline{\text{CS7}}$	IN	26
			Control	25
			OUT	24
98	B8	P72/TMO0/$\overline{\text{CS6}}$	IN	23
			Control	22
			OUT	21

TFP-120, TFP-120V Pin No.	BP-112, BP-112V Pin No.	Pin Name	I/O	Bit Name
99	A8	P71/$\overline{CS5}$	IN	20
			Control	19
			OUT	18
100	D7	P70/TMRI01/TMCI01/$\overline{CS4}$	IN	17
			Control	16
			OUT	15
101	C7	PG0	IN	14
			Control	13
			OUT	12
102	A7	PG1/$\overline{CS3}$/$\overline{IRQ7}$	IN	11
			Control	10
			OUT	9
103	B7	PG2/$\overline{CS2}$	IN	8
			Control	7
			OUT	6
104	C6	PG3/$\overline{CS1}$	IN	5
			Control	4
			OUT	3
105	A6	PG4/$\overline{CS0}$	IN	2
			Control	1
			OUT	0
		to TDO		

14.4 Boundary Scan Function Operation

14.4.1 TAP Controller

Figure 14.3 shows the TAP controller status transition diagram, based on the JTAG standard.

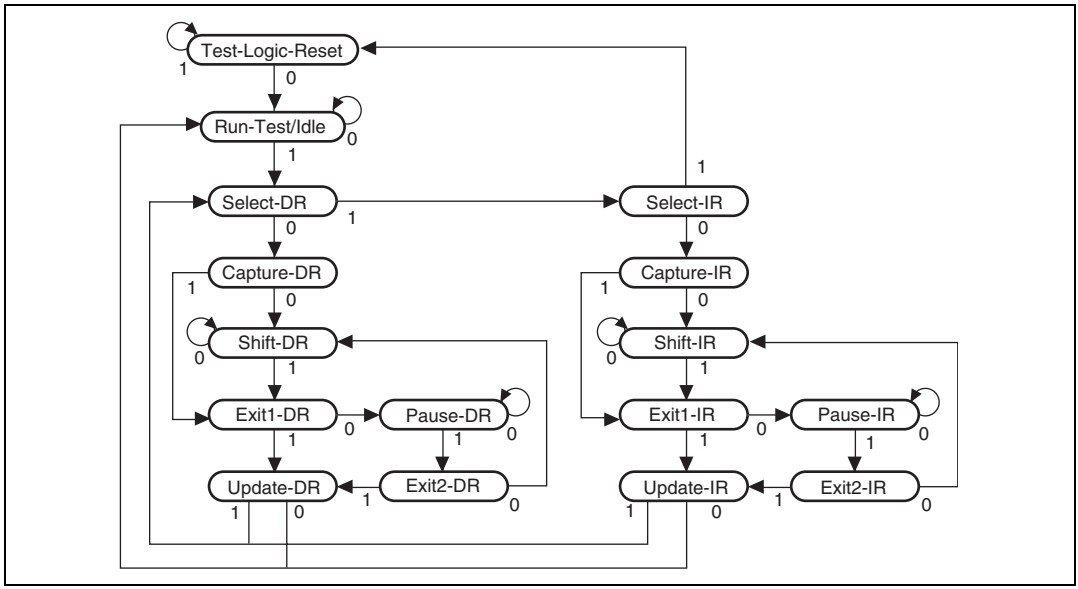


Figure 14.3 TAP Controller Status Transition

Note: The transition condition is the TMS value at the rising edge of TCK. The TDI value is sampled at the rising edge of the TCK and shifted at the falling edge of the TCK. The TDO value changes at the falling edge of the TCK. In addition, TDO is high-impedance state in a state other than Shift-DR or Shift-IR state. If $\overline{\text{TRST}}$ is 0, Test-Logic-Reset state is entered asynchronously with the TCK.

14.5 Usage Notes

1. When using the boundary scan function, clear $\overline{\text{TRST}}$ to 0 at power-on and after the t_{RESW} time has elapsed set $\overline{\text{TRST}}$ to 1 and set TCK, TMS, and TDI appropriately. During normal operation when the boundary scan function is not used, set TCK, TMS, and TDI to Hi-Z, clear $\overline{\text{TRST}}$ to 0 at power-on, and after the t_{RESW} time has elapsed set TRST to 1 or to Hi-Z. These pins are pulled up internally, so care must be taken in standby mode because breakthrough current flow can occur if there is a potential difference between the pin input voltage value when set to 1 and the power supply voltage Vcc.
2. The following must be noted on the power-on reset signal applied to the $\overline{\text{TRST}}$ pin.
   - Reset signal must be applied at power-on.
   - $\overline{\text{TRST}}$ must be separated in order not to affect the system operation.
   - $\overline{\text{TRST}}$ must be separated from the system circuitry in order not to affect the system operation.
   - System circuitry must also be separated from the $\overline{\text{TRST}}$ in order not to affect $\overline{\text{TRST}}$ operation as shown in figure 14.4.

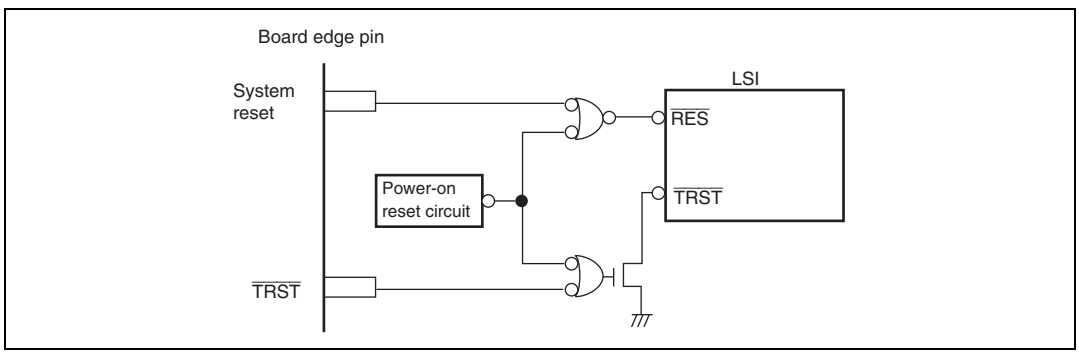


Figure 14.4 Recommended Reset Signal Design

3. TCK clock speed should be slower than system clock frequency.
4. In serial communication, data is input or output from the LSB as shown in figure 14.5.

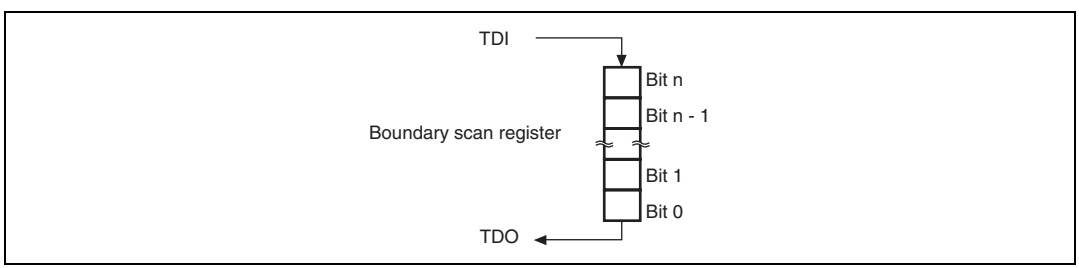


Figure 14.5 Serial Data Input/Output

5. If a pin with pull-up function is SAMPLEed with pull-up function enabled, the corresponding IN register is set to 1. In this case, the corresponding Control register must be cleared to 0.
6. If a pin with open-drain function is SAMPLEed while its open-drain function is enabled and while the corresponding OUT register is set to 1, the corresponding Control register is cleared to 0 (the pin status is Hi-Z). If the pin is SAMPLEed while the corresponding OUT register is cleared to 0, the corresponding Control register is set to 1 (the pin status is 0).
7. If EXTEST, CLAMP, or HIGHZ state is entered, this LSI enters guarded mode such as hardware standby mode ($\overline{RES}$ = $\overline{STBY}$ = 0). Before entering normal operating mode from EXTEST, CLAMP, or HIGHZ state, specify $\overline{RES}$, $\overline{STBY}$, FWE, and MD2 to MD0 pin to the designated mode.
8. When using the boundary scan function, leave the $\overline{EMLE}$ pin open.

Section 15 Universal Serial Bus Interface (USB)

This LSI incorporates a USB function module complying with USB standard version 1.1. Figure 15.1 shows the block diagram of the USB.

15.1 Features

- USB standard version 2.0 full speed mode (12 Mbps) support
- Bus-powered mode or self-powered mode is selectable via the USB specific pin ($\overline{\text{UBPM}}$)
- On-chip 48-MHz clock generator and PLL circuit (16 MHz × 3 = 48 MHz, 24 MHz* × 2 = 48 MHz)

Note: * Available only in H8S/2215R and H8S/2215T.

- On-chip bus transceiver
- Standard commands are processed automatically by hardware
  - Only Set_Descriptor, Get_Descriptor, Class/VendorCommand, and SynchFrame commands should be processed by software
- Configuration value, InterfaceNumber value, and AlternateSetting value can be checked by Set_Configuration and Set_Interface interrupts
- Four transfer mode supported (Control, Interrupt, Bulk, Isochronous)
- Endpoint configuration selectable

  Maximum of 9 endpoints can be specified (including endpoint 0)

  The size of the FIFO buffer used by each endpoint can be specified via firmware

  The FIFO buffer for bulk transfer and isochronous transfer has a double-buffer configuration

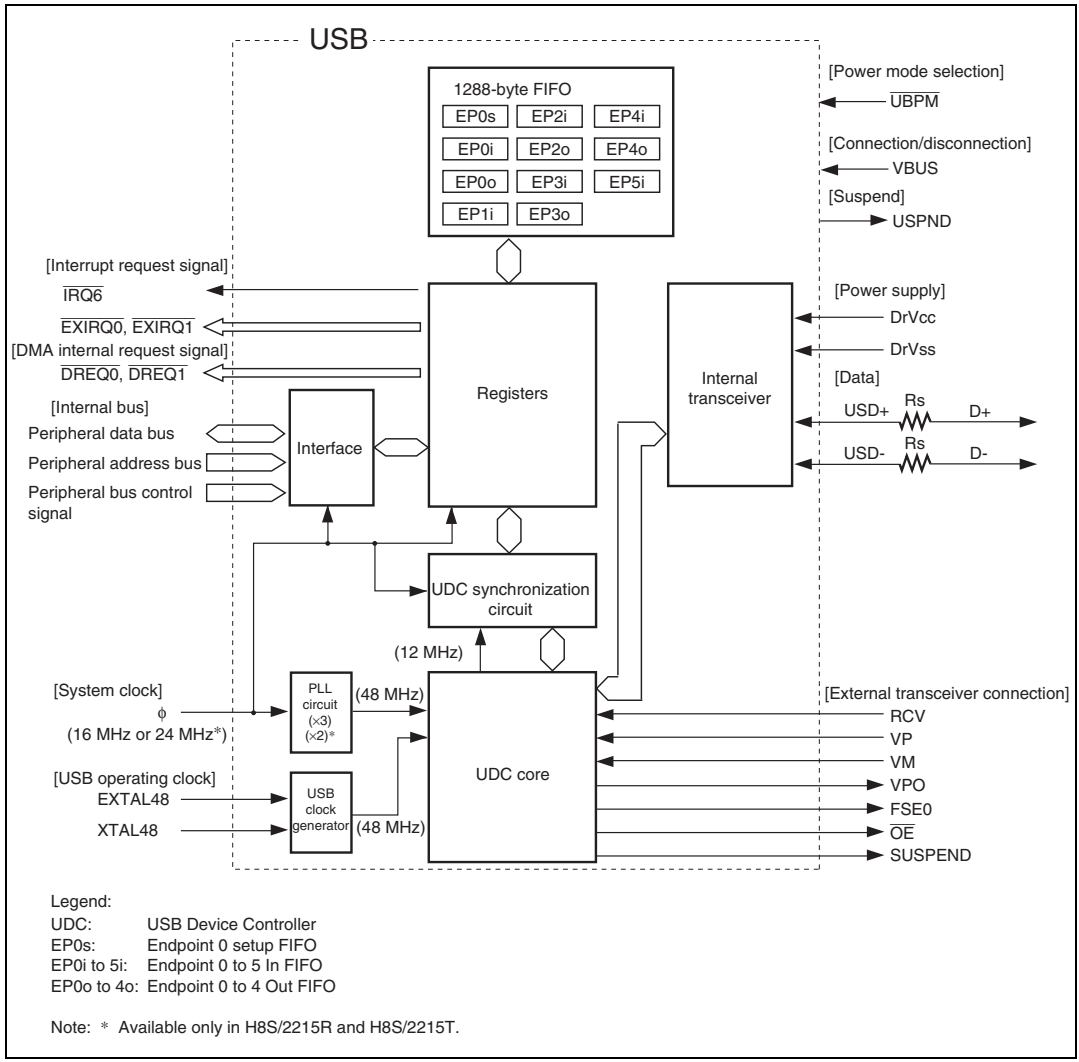
  Total 1288-byte FIFO
  - —EP0s fixed: Control_setup FIFO, 8 bytes
  - —EP0i fixed: Control_in FIFO, 64 bytes
  - —EP0o fixed: Control_out FIFO, 64 bytes
  - —EPn selectable: Interrupt_in FIFO, variable 0 to 64 bytes
  - —EPn selectable: Bulk_in FIFO, 64 bytes × 2 (double-buffer configuration)
  - —EPn selectable: Bulk_out FIFO, 64 bytes × 2 (double-buffer configuration)
  - —EPn selectable: Isochronous_in FIFO, variable 0 to 128 bytes × 2 (double-buffer configuration)
  - —EPn selectable: Isochronous_out FIFO, variable 0 to 128 bytes × 2 (double-buffer configuration)
  - —EPn selectable: Bulk_in FIFO, 64 bytes × 2 (double-buffer configuration)
  - —EPn selectable: Bulk_out FIFO, 64 bytes × 2 (double-buffer configuration)
  - —EPn selectable: Interrupt_in FIFO, variable 0 to 64 bytes

- Maximum Configuration, InterfaceNumber, and AlternateSetting configuration specifications of this LSI

  H8S/2215: EP0
  Configuration 1 ----- InterfaceNumber 0 to 2 ----- AlternateSetting 0 to 7 ----- EP1 to EP8

  H8S/2215R and H8S/2215T: EP0
  Configuration 1 ----- InterfaceNumber 0 to 3 ----- AlternateSetting 0 to 7 ----- EP1 to EP8

- Start of frame (SOF) marker function
  - SOF interrupt occurs every 1 ms even though broken SOF received by error
- 23 kinds of interrupts (H8S/2215)
  25 kinds of interrupts (H8S/2215R and H8S/2215T)
  - Suspend/resume interrupt source can be assigned for $\overline{\text{IRQ6}}$
  - Each interrupt source can be assigned for $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$ via registers
- DMA transfer interface
  - Two DMA requests are selectable from four Bulk transfer requests
- 8-bit bus (3 cycle bus access timing) connected to the external bus interface
  - Internal registers are addressed to a part of area 6 of external address (H'C00000 to H'DFFFFF)
  - The area of H'C00100 to H'DFFFFF is reserved for USB and should not be accessed



Legend:
UDC: USB Device Controller
EP0s: Endpoint 0 setup FIFO
EP0i to 5i: Endpoint 0 to 5 In FIFO
EP0o to 4o: Endpoint 0 to 4 Out FIFO

Note: * Available only in H8S/2215R and H8S/2215T.

Figure 15.1 Block Diagram of USB

15.2 Input/Output Pins

Table 15.1 shows the USB pin configuration.

Table 15.1 Pin Configuration

Pin Name	I/O	Function
USD+	I/O	I/O pin for USB data
USD-		
DrVCC	Input	USB internal transceiver power supply pin
DrVSS	Input	USB internal transceiver ground pin
VBUS	Input	USB cable connection/disconnection detection signal pin
$\overline{\text{UBPM}}$	Input	USB bus-power/self-power mode selection pin When USB is used in bus-power mode, $\overline{\text{UBPM}}$ must be fixed low. When USB is used in self-power mode, $\overline{\text{UBPM}}$ must be fixed high.
XTAL48, EXTAL48	Input	USB operating clock input pin 48-MHz clock for USB communication is input. When the internal PLL is used, EXTAL48 and XTAL48 must be fixed low and open, respectively.
USPND	Output	USB suspend output pin Set to high level when the system enter the suspend state.
RCV	Input	External transceiver connection signals Signals used to connect with the transceiver (ISP1104) manufactured by NXP.
VP	Input	
VM	Input	
VPO	Output	
FSE0	Output	
$\overline{\text{OE}}$	Output	
SUSPND	Output	

15.3 Register Descriptions

The USB has the following registers for each channel.

- USB endpoint information register 00_0 to 22_4 (UEPIR00_0 to UEPIR22_4)
- USB control register (UCTLR)
- USB DMAC transfer request register (UDMAR)*
- USB device resume register (UDRR)
- USB trigger register 0 (UTRG0)*
- USB trigger register 1 (UTRG1)*
- USB FIFO clear register 0 (UFCLR0)*
- USB FIFO clear register 1 (UFCLR1)*
- USB endpoint stall register 0 (UESTL0)*
- USB endpoint stall register 1 (UESTL1)*
- USB endpoint data register 0s (UEDR0s) [for Setup data reception]
- USB endpoint data register 0i (UEDR0i) [for Control_in data transmission]
- USB endpoint data register 0o (UEDR0o) [for Control_out data reception]
- USB endpoint data register 1i (UEDR1i)* [for Interrupt_in data transmission]
- USB endpoint data register 2i (UEDR2i)* [for Bulk_in data transmission]
- USB endpoint data register 2o (UEDR2o)* [for Bulk_out data reception]
- USB endpoint data register 3i (UEDR3i)* [for Isochronous_in data transmission]
- USB endpoint data register 3o (UEDR3o)* [for Isochronous_out data reception]
- USB endpoint data register 4i (UEDR4i)* [for Bulk_in data transmission]
- USB endpoint data register 4o (UEDR4o)* [for Bulk_out data reception]
- USB endpoint data register 5i (UEDR5i)* [for Interrupt_in data transmission]
- USB endpoint receive data size register 0o (UESZ0o) [for Control _out data reception]
- USB endpoint receive data size register 2o (UESZ2o)* [for Bulk_out data reception]
- USB endpoint receive data size register 3o (UESZ3o)* [for Isochronous_out data reception]
- USB endpoint receive data size register 4o (UESZ4o)* [for Bulk _out data reception]
- USB interrupt flag register 0 (UIFR0)*
- USB interrupt flag register 1 (UIFR1)*
- USB interrupt flag register 2 (UIFR2)*
- USB interrupt flag register 3 (UIFR3)
- USB interrupt enable register 0 (UIER0)*
- USB interrupt enable register 1 (UIER1)*
- USB interrupt enable register 2 (UIER2)*

- USB interrupt enable register 3 (UIER3)
- USB interrupt selection register 0 (UISR0)*
- USB interrupt selection register 1 (UISR1)*
- USB interrupt selection register 2 (UISR2)*
- USB interrupt selection register 3 (UISR3)
- USB data status register (UDSR)*
- USB configuration value register (UCVR)
- USB time stamp register H, L (UTSRH, L)
- USB test register 0 (UTSTR0)
- USB test register 1 (UTSTR1)
- USB test register 2 (UTSTR2)
- USB test register A (UTSTRA)
- USB test register B (UTSTRB)
- USB test register C (UTSTRC)
- USB test register D (UTSTRD)
- USB test register E (UTSTRE)
- USB test register F (UTSTRF)
- Module stop control register B (MSTPCRB)

Note: * Indicates the register name or bit name when each endpoint formation is specified based on the Bluetooth standard. Register names and bit names must be modified according to the endpoint configuration selected. For details, refer to section 15.7, Endpoint Configuration Example.
The area of H'C00100 to H'DFFFFF is reserved for USB and should not be accessed.

15.3.1 USB Endpoint Information Registers 00_0 to 22_4 (UEPIR00_0 to UEPIR22_4)

UEPIR is used to set 23 kinds of endpoint (EPINFO data). EPINFO data for each endpoint consists of 40 bits (five bytes). 115 bytes of endpoint data for all UEPIR00_0 to UEPIR22_4 registers must be written after the UDC interface software reset has been cancelled (the UIFST bit of the UCTLR register is cleared to 0). The endpoint data is automatically loaded and stored in the buffers in the UDC core after the UDC core software reset has been cancelled (the UDCRST bit of the UCTLR register is cleared to 0). For details on EPINFO data setting procedure, refer to section 15.5, Communication Operation.

The USB module in this LSI is designed to automatically load EPINFO data after UDC software reset. Accordingly, EPINFO data must be specified correctly. Otherwise, USB communication cannot be performed correctly.

EPINFO data written to UEPIR is maintained in the register. This EPINFO data is automatically re-loaded after each UDC core reset. Accordingly, EPINFO data need to be written only once.

- **UEPIRnn_0**

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	D39 to D36	—	R/W	Endpoint number (4-bit configuration, settable values: 0 to 8) 0000: Control transfer (EP0) 0001 to 1000: Other than Control transfer (EP1 to EP8) There are restrictions on settable endpoint numbers according to the Interface number and Alternate number to which the endpoint belongs. Restriction 1: Set different endpoint numbers under one Alternate. However, there is no problem with use of the same endpoint number if the transfer directions (IN/OUT) are different. (Ex: Alt0 -- EP1, EP2i, EP2o) Restriction 2: Do not set the same endpoint number under different Interface numbers. (Ex: Int0 -- Alt0 -- EP1, EP2, Int1 -- Alt0 -- EP3)
3	D35	—	R/W	Configuration number to which endpoint belongs (2-bit configuration, settable values: 0, 1) 00: Control transfer 01: Other than Control transfer
2	D34	—	R/W	
1	D33	—	R/W	H8S/2215
0	D32	—	R/W	Interface number to which endpoint belongs (2-bit configuration, settable values: 0 to 2) 00: Control transfer 00 to 10: Other than Control transfer H8S/2215R and H8S/2215T Interface number to which endpoint belongs (2-bit configuration, settable values: 0 to 3) 00: Control transfer 00 to 11: Other than Control transfer

- **UEPIRnn_1**

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	D31 to D29	—	R/W	Alternate number to which endpoint belongs (3-bit configuration, settable values: 0 to 7) 000: Control transfer 000 to 111: Other than Control transfer
4	D28	—	R/W	Endpoint transfer type (2-bit configuration)
3	D27	—	R/W	00: Control (UEPIR00) 01: Isochronous (UEPIR04 to UEPIR19) 10: Bulk (UEPIR02, UEPIR03, UEPIR20, UEPIR21) 11: Interrupt (UEPIR01, UEPIR22)
2	D26	—	R/W	Endpoint transfer direction (1-bit configuration) 0: out (UEPIR00, 03, 05, 07, 09, 11, 13, 15, 17, 19, 21) 1: in (UEPIR01, 02, 04, 06, 08, 10, 12, 14, 16, 18, 20, 22)
1	D25	—	R/W	Endpoint maximum packet size (D25 to D16 10-bit configuration) Control transfer = 64 only (UEPIR00) Interrupt transfer = 0 to 64 (UEPIR01, UEPIR22) Bulk transfer = 0 or 64 (UEPIR02, UEPIR03, UEPIR20, UEPIR21) Isochronous transfer = 0 to 128 (UEPIR04 to UEPIR19)
0	D24	—	R/W	

- **UEPIRnn_2**

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D23 to D16	—	R/W	Endpoint maximum packet size (D25 to D16 10-bit configuration) Control transfer = 64 only (UEPIR00) Interrupt transfer = 0 to 64 (UEPIR01, UEPIR22) Bulk transfer = 0 or 64 (UEPIR02, UEPIR03, UEPIR20, UEPIR21) Isochronous transfer = 0 to 128 (UEPIR04 to UEPIR19)

- **UEPIRnn_3**

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D15 to D8	—	R/W	Endpoint internal address (D15 to D0 16-bit configuration) Set UEPIR00_3, UEPIR00_4 = H'0000 Set UEPIR01_3, UEPIR01_4 = H'0001 : Set UEPIR21_3, UEPIR21_4 = H'0015 Set UEPIR22_3, UEPIR22_4 = H'0016

- **UEPIRnn_4**

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	—	R/W	Endpoint internal address (D15 to D0 16-bit configuration) Set UEPIR00_3, UEPIR00_4 = H'0000 Set UEPIR01_3, UEPIR01_4 = H'0001 : Set UEPIR21_3, UEPIR21_4 = H'0015 Set UEPIR22_3, UEPIR22_4 = H'0016

This manual assumes that endpoint information (EPINFO data) is configured based on the Bluetooth standard shown in figure 15.2. If endpoint data is configured in a configuration other than that shown in figure 15.2, care must be taken for the correspondence between endpoint number, Configuration/Interface/Alternate number and maximum packet size, and register name and bit name. For details, refer to section 15.7, Endpoint Configuration Example.

Endpoint data configured based on the Bluetooth standard can be specified as shown in table 15.2. Endpoint data shown in table 15.2 includes unused endpoints (EP4i, EP4o, and EP5i). To load all EPINFO data items from UEPIR00_0 to UEPIR22_4 correctly, unused end pints must also be dummy written as shown in table 15.2.

In addition, to prevent unused endpoints from being accessed from the host, descriptor information for the unused endpoints must not be returned in the enumeration phase at connection. This correctly informs the host of usable endpoint information and enables access control for unused endpoints. If descriptor information for the unused endpoints is returned to the host, the USB cannot operate correctly when the host accesses the unused endpoint.

Note that endpoint data information must match the corresponding descriptor information to be returned to the host . Otherwise, the USB cannot operate correctly. For example, if the descriptor information is returned as 16 bytes while the maximum packet size of the EPINFO data is eight bytes, the host attempts to access the EPINFO data in 16 byte units and cannot operate correctly.

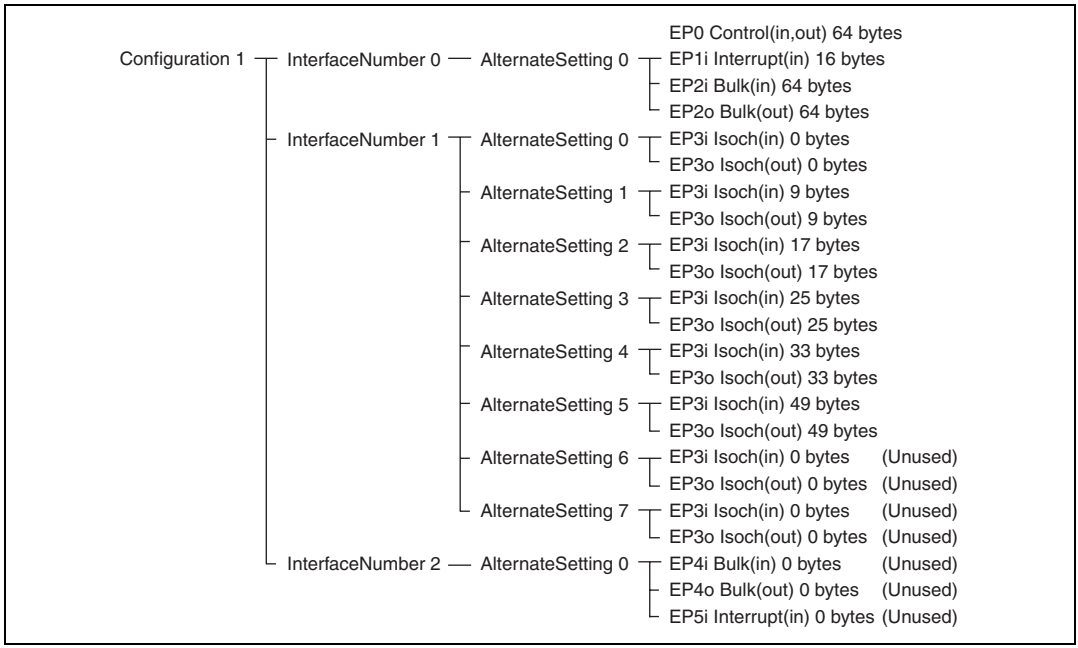


Figure 15.2 Example of Endpoint Configuration based on Bluetooth Standard

Table 15.2 shows the example of EPINFO data setting for endpoint configuration based on the Bluetooth standard.

The USB module of this LSI is optimized by the hardware specific to the transfer type. Accordingly, endpoints cannot be configured completely freely. Endpoints can be modified within the restrictions (only data within parentheses [ ]) in table 15.2. Data other than that within parentheses [ ] must be specified according to table 15.2. For details on other endpoint configuration, refer to section 15.7, Endpoint Configuration Example.

Table 15.2 EPINFO Data Settings

			EPINFO Data Settings Based on Bluetooth Standard						
No.	Register Name	Address	Corresponding Transfer Mode[*1]	UEPIRn_0 to UEPIRn_4 Settings	UEPI Rn_0	UEPI Rn_1	UEPI Rn_2	UEPI Rn_3	UEPI Rn_3
1	UEPIR00_0 to UEPIR00_4	H'C00000 to H'C0004	Specific to Control transfer	B'0000_00_00_000_00_0_0001000000_0000000000000000	H'00	H'00	H'40	H'00	H'00
2	UEPIR01_0 to UEPIR01_4	H'C00005 to H'C0009	Specific to Interrupt in transfer	B'[0001]_01_[00]_[000]_11_1_[0000010000]_0000000000000001[*3]	H'14	H'1C	H'10	H'00	H'01
3	UEPIR02_0 to UEPIR02_4	H'C0000A to H'C000E	Specific to Bulk in transfer	B'[0010]_01_[00]_[000]_10_1_[0001000000]_0000000000000010[*4]	H'24	H'14	H'40	H'00	H'02
4	UEPIR03_0 to UEPIR03_4	H'C0000F to H'C0013	Specific to Bulk out transfer	B'[0010]_01_[00]_[000]_10_0[0001000000]_0000000000000011[*4]	H'24	H'10	H'40	H'00	H'03
5	UEPIR04_0 to UEPIR04_4	H'C00014 to H'C0018	Specific to Isoch in transfer	B'[0011]_01_[01]_[000]_01_1_[0000000000]_0000000000000100[*5]	H'35	H'0C	H'00	H'00	H'04
6	UEPIR05_0 to UEPIR05_4	H'C00019 to H'C001D	Specific to Isoch out transfer	B'[0011]_01_[01]_[000]_01_0_[0000000000]_0000000000000101[*5]	H'35	H'08	H'00	H'00	H'05
7	UEPIR06_0 to UEPIR06_4	H'C0001E to H'C0022	Specific to Isoch in transfer	B'[0011]_01_[01]_[001]_01_1_[0000001001]_0000000000000110[*5]	H'35	H'2C	H'09	H'00	H'06
8	UEPIR07_0 to UEPIR07_4	H'C00023 to H'C0027	Specific to Isoch out transfer	B'[0011]_01_[01]_[001]_01_0_[0000001001]_0000000000000111[*5]	H'35	H'28	H'09	H'00	H'07
9	UEPIR08_0 to UEPIR08_4	H'C00028 to H'C002C	Specific to Isoch in transfer	B'[0011]_01_[01]_[010]_01_1_[0000010001]_0000000000001000[*5]	H'35	H'4C	H'11	H'00	H'08
10	UEPIR09_0 to UEPIR09_4	H'C0002D to H'C0031	Specific to Isoch out transfer	B'[0011]_01_[01]_[010]_01_0_[0000010001]_0000000000001001[*5]	H'35	H'48	H'11	H'00	H'09
11	UEPIR10_0 to UEPIR10_4	H'C00032 to H'C0036	Specific to Isoch in transfer	B'[0011]_01_[01]_[011]_01_1_[0000011001]_0000000000001010[*5]	H'35	H'6C	H'19	H'00	H'0A
12	UEPIR11_0 to UEPIR11_4	H'C00037 to H'C003B	Specific to Isoch out transfer	B'[0011]_01_[01]_[011]_01_0_[0000011001]_0000000000001011[*5]	H'35	H'68	H'19	H'00	H'0B
13	UEPIR12_0 to UEPIR12_4	H'C0003C to H'C0040	Specific to Isoch in transfer	B'[0011]_01_[01]_[100]_01_1_[0000100001]_0000000000001100[*5]	H'35	H'8C	H'21	H'00	H'0C
14	UEPIR13_0 to UEPIR13_4	H'C00041 to H'C0045	Specific to Isoch out transfer	B'[0011]_01_[01]_[100]_01_0_[0000100001]_0000000000001101[*5]	H'35	H'88	H'21	H'00	H'0D
15	UEPIR14_0 to UEPIR14_4	H'C00046 to H'C004A	Specific to Isoch in transfer	B'[0011]_01_[01]_[101]_01_1_[0000110001]_0000000000001110[*5]	H'35	H'AC	H'31	H'00	H'0E
16	UEPIR15_0 to UEPIR15_4	H'C0004B to H'C004F	Specific to Isoch out transfer	B'[0011]_01_[01]_[101]_01_0_[0000110001]_0000000000001111[*5]	H'35	H'A8	H'31	H'00	H'0F
17	UEPIR16_0 to UEPIR16_4	H'C00050 to H'C0054	Specific to Isoch in transfer	B'[0011]_01_[01]_[110]_01_1_[0000000000]_0000000000010000[*5][*6]	H'35	H'CC	H'00	H'00	H'10

EPINFO Data Settings Based on Bluetooth Standard

No.	Register Name	Address	Corresponding Transfer Mode*1	UEPIRn_0 to UEPIRn_4 Settings	UEPIRn_0	UEPIRn_1	UEPIRn_2	UEPIRn_3	UEPIRn_3
18	UEPIR17_0 to UEPIR17_4	H'C00055 to H'C0059	Specific to Isoch out transfer	B'[0011]_01_[01]_[110]_01_0_ [0000000000]_0000000000010001*5*6	H'35	H'C8	H'00	H'00	H'11
19	UEPIR18_0 to UEPIR18_4	H'C0005A to H'C005E	Specific to Isoch in transfer	B'[0011]_01_[01]_[111]_01_1_ [0000000000]_0000000000010010*5*6	H'35	H'EC	H'00	H'00	H'12
20	UEPIR19_0 to UEPIR19_4	H'C0005F to H'C0063	Specific to Isoch out transfer	B'[0011]_01_[01]_[111]_01_0_ [0000000000]_0000000000010011*5*6	H'35	H'E8	H'00	H'00	H'13
21	UEPIR20_0 to UEPIR20_4	H'C00064 to H'C0068	Specific to Bulk in transfer	B'[0100]_01_[10]_[000]_10_1_ [0000000000]_0000000000010100*4*6	H'46	H'14	H'00	H'00	H'14
22	UEPIR21_0 to UEPIR21_4	H'C00069 to H'C006D	Specific to Bulk out transfer	B'[0100]_01_[10]_[000]_10_0_ [0000000000]_0000000000010101*4*6	H'46	H'10	H'00	H'00	H'15
23	UEPIR22_0 to UEPIR22_4	H'C0006E to H'C0072	Specific to Interrupt in transfer	B'[0101]_01_[10]_[000]_11_1_ [0000000000]_0000000000010110*3*6	H'56	H'1C	H'00	H'00	H'16

Notes: 1. Each endpoint is optimized by the hardware specific for the transfer mode. The transfer mode shown in table 15.2 must be specified. (D28 and D27 for all EPINFO data items must e specified as shown in table 15.2.)
2. Data indicated within parentheses [ ] can be modified. Data other than that within parentheses [ ] must be specified as shown in table 15.2.
3. Maximum packet size of Interrupt transfer must be from 0 to 64.
4. Maximum packet size of Bulk transfer must be 64 when used or 0 when unused.
5. Maximum packet size of Isochronous transfer must be from 0 to 128.
6. Maximum packet size of endpoint must be 0 when unused.

15.3.2 USB Control Register (UCTLR)

UCTLR is used to select USB data input/output pin and USB operating clock, specify SOF marker function, and controls the USB module reset. UCTLR can be read from or written to even in USB module stop mode. For details on UCTLR setting procedure, refer to section 15.5, Communication Operation.

Bit	Bit Name	Initial Value	R/W	Description
7	FADSEL	0	R/W	I/O Analog or Digital Selection Selects USB function data I/O pins 0: USD+ and USD- are used as data I/O pins 1: Control I/O ports 1 and A compatible with Philips Corp. transceiver are connected to data I/O pins. P17 (output) → $\overline{OE}$: Output enable P15 (output) → FSE0: SE0 setting P13 (output) → VPO: Data+ output P12 (input) ← PCV: Differential input P11 (input) ← VP: Data+ input P10 (input) ← VM: Data– input PA3 (output) → SUSPND: Suspend enable Ports 1 and A are prioritized to address outputs. Accordingly, before setting FADSEL to 1, disable A23 to A19 output via PFCR. In addition, FADSEL must be set during USB module stop mode.
6	SFME	0	R/W	Start Of Frame (SOF) Marker Function Enable Controls the SOF marker function. If SFME is set to 1, the SOF interrupt flag can be set to 1 every 1ms even if the SOF packet has been broken. Note, however, that UTSR stores a time stamp when the correct SOF packet is received. The USB does not support UTSR automatic update function when the SOF packet is broken. To set SFME the first time, SFME must be set after SOF flag detection. SFME must be cleared to 0 when the suspension is detected. To set SFME after resume detection, SFME must also be set after SOF flag detection. 0: Disables the SOF marker function 1: Enables the SOF marker function

Bit	Bit Name	Initial Value	R/W	Description
5	UCKS3	0	R/W	USB Operating Clock Selection 3 to 0
4	UCKS2	0	R/W	Select the USB operating clock (48 MHz). When UCKS0 to UCKS3 are 0000, both the 48-MHz oscillator and internal PLL circuit stop and USB operating clock must be selected according to the clock source.
3	UCKS1	0	R/W	
2	UCKS0	0	R/W	
				The internal PLL circuit and 48-MHz oscillator start operating after USB module stop mode has been cancelled. In addition, the USB operating clock is supplied to the UDC core after 48-MHz clock stabilization time has been passed. The USB clock stabilization wait time completion can be detected by the CK48READY flag of UIFR3.
				UCKS0 to UCKS3 muse be written during USB module stop mode.
				0000: USB operating clock stops (Both 48-MHz oscillator and PLL stop)
				0001: Reserved
				0010 (H8S/2215): Reserved
				0010 (H8S/2215R and H8S/2215T): Uses a clock (48 MHz) generated by doubling the 24-MHz system clock by the PLL circuit. The 48-MHz oscillator stops. The USB operating clock stabilization time is 2 ms.
				0011: Uses a clock (48 MHz) generated by tripling the 16-MHz external clock (EXTAL pin input) by the PLL circuit.
				0100: Reserved
				0101: Reserved
				0110 (H8S/2215): Reserved
				0110 (H8S/2215R and H8S/2215T): Uses a clock (48 MHz) generated by doubling the 24-MHz system clock by the PLL circuit. The 48-MHz oscillator stops. The USB operating clock stabilization time is 8 ms.
				0111: Uses a clock (48 MHz) generated by tripling the 16-MHz crystal oscillator (system clock pulse generator) by the PLL circuit.

Bit	Bit Name	Initial Value	R/W	Description
5	UCKS3	0	R/W	1000: Uses a clock supplied by the 48-MHz external clock (EXTAL48 pin input) directly. The PLL stops. The USB operating clock stabilization time is 246 to 200 µs.
4	UCKS2	0	R/W	
3	UCKS1	0	R/W	
2	UCKS0	0	R/W	1001 (H8S/2215): Reserved 1001 (H8S/2215R and H8S/2215T): Uses the clock supplied by the 48-MHz external clock (EXTAL48 pin input) directly. The PLL stops. The USB operating clock stabilization time is 300 to 200 µs (when using a 16-MHz to 24-MHz system clock). 1010: Reserved 1011: Reserved 1100: Uses the USB operating clock (48 MHz) directly. The PLL stops. The USB operating clock stabilization time is 9.9 to 8 ms. 1101 (H8S/2215): Reserved 1101 (H8S/2215R and H8S/2215T): Uses the USB operating clock (48 MHz) directly. The PLL stops. The USB operating clock stabilization time is 12 to 8 ms (when using a 16-MHz to 24-MHz system clock). 1110: Reserved 1111: Reserved Note that the USB operating clock stabilization time differs according to the selected clock source and is automatically counted by the system clock. The USB operating clock stabilization time shown above is for the case when the 13- to 24- MHz system clock is used.

Bit	Bit Name	Initial Value	R/W	Description
1	UIFRST	1	R/W	USB Interface Software Reset Controls USB module internal reset. When the UIFRST bit is set to 1, the USB internal modules other than UCTLR, UIER3, and the CK48 READY bit of UIFR3 are all reset. At initialization, the UIFRST bit must be cleared to 0 after the USB operating clock stabilization time has passed following USB module stop mode cancellation. 0: Sets the USB internal modules to the operating state (at initialization, this bit must be cleared after the USB operating clock stabilization time has passed). 1: Sets the USB internal modules other than UCTLR, UIER3, and the CK48 READY bit of UIFR3 reset state. If the UIFRST bit is set to 1 after it is cleared to 0, the UDCRST bit is also set to 1 simultaneously.
0	UDCRST	1	R/W	UDC Core Software Reset Controls reset of the UDC core in the USB module. When the UDCRST bit is set to 1, the UDC core is reset and USB bus synchronization operation stops. At initialization, UDCRST must be cleared to 0 after D+ pull-up following UIFRST clearing to 0. In the suspend state, to maintain the internal state of the UDC core, enter software standby mode after setting USB module stop mode with the UDCRST bit to be maintained. After VBUS disconnection detection, UDCRST must be set to 1. 0: Sets the UDC core in the USB module to operating state (at initialization, UDCRST must be cleared after D+ pull-up following UIFRST clearing to 0). 1: Sets the UDC core in the USB module to reset state (in the suspend state, UDCRST must not be set to 1; after VBUS disconnection detection, UDCRST must be set to 1).

15.3.3 USB DMAC Transfer Request Register (UDMAR)

UDMAR is set when data transfer by means of a USB request of the on-chip DMAC is performed for data registers UEDR2i, UEDR2o, UEDR4i, and UEDR4o corresponding to EP2i, EP2o, EP4i, and EP4o used for Bulk transfer, respectively. DMAC transfer request sources specified in UDMAR must be two or less. If two DMAC transfer request sources are specified, a source must use $\overline{\text{DREQ0}}$ and another source must use $\overline{\text{DREQ1}}$. If three or more DMAC transfer requests are specified or if $\overline{\text{DREQ0}}$ and $\overline{\text{DREQ1}}$ usage overlaps, the USB cannot operate correctly. For details on DMAC transfer, refer to section 15.6, DMA Transfer Specifications.

Note: As the $\overline{\text{DREQ}}$ signal is not used in the data transfer by auto request of the on-chip DMAC, set UDMAR to H'00.

Bit	Bit Name	Initial Value	R/W	Description
7	EP4oT1	0	R/W	EP4o DMAC Transfer Request Selection 1, 0
6	EP4oT0	0	R/W	00: Does not request EP4o DMAC transfer
				01: Reserved
				10: Requests EP4o DMAC transfer by $\overline{\text{DREQ0}}$
				11: Requests EP4o DMAC transfer by $\overline{\text{DREQ1}}$
5	EP4iT1	0	R/W	EP4i DMAC Transfer Request Selection 1, 0
4	EP4iT0	0	R/W	00: Does not request EP4i DMAC transfer
				01: Reserved
				10: Requests EP4i DMAC transfer by $\overline{\text{DREQ0}}$
				11: Requests EP4i DMAC transfer by $\overline{\text{DREQ1}}$
3	EP2oT1	0	R/W	EP2o DMAC Transfer Request Selection 1, 0
2	EP2oT0	0	R/W	00: Does not request EP2o DMAC transfer
				01: Reserved
				10: Requests EP2o DMAC transfer by $\overline{\text{DREQ0}}$
				11: Requests EP2o DMAC transfer by $\overline{\text{DREQ1}}$
1	EP2iT1	0	R/W	EP2i DMAC Transfer Request Selection 1, 0
0	EP2iT0	0	R/W	00: Does not request EP2i DMAC transfer
				01: Reserved
				10: Requests EP2i DMAC transfer by $\overline{\text{DREQ0}}$
				11: Requests EP2i DMAC transfer by $\overline{\text{DREQ1}}$

15.3.4 USB Device Resume Register (UDRR)

UDRR indicates remote wakeup according to the host enable/disable state and enables or disables remote wakeup of the USB modules in the suspend state.

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
1	RWUPs	0	R	Remote Wakeup Status Indicates the enabled or disabled state of remote wakeup by the host. This bit is a status bit and cannot be written to. If the remote wakeup from the host is disabled by Device_Remote_Wakeup through the Set_Feature/Clear_Feature request, this bit is cleared to 0. If the remote wakeup is enabled, this bit is set to 1. 0: Remote wakeup disabled state 1: Remote wakeup enabled state
0	DVR	0	W	Device Resume Cancels suspend state (remote wakeup execution). This bit can be written to 1 and is always read as 0. Before executing remote wakeup, software standby mode or USB module stop mode must be cancelled to provide a clock for the USB module. 0: Performs no operation 1: Cancels suspend state (executes remote wakeup)

15.3.5 USB Trigger Register 0 (UTRG0)

UTRG0 generates one-shot triggers to the FIFO for each endpoint EP0 to EP2.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
5	EP2oRDFN	0	W	EP2o Read Completion 0: Performs no operation 1: Writes 1 to this bit after reading data for EP2o OUT FIFO. EP2o FIFO has a dual FIFO configuration. This trigger is generated to the currently effective FIFO.
4	EP2iPKTE	0	W	EP2i Packet Enable 0: Performs no operation 1: Generates a trigger to enable data transfer to the EP2i IN FIFO. EP2i FIFO has a dual FIFO configuration. This trigger is generated for the currently effective FIFO.
3	EP1iPKTE	0	W	EP1i Packet Enable 0: Performs no operation 1: Generates a trigger to enable data transfer to the EP1i IN FIFO.
2	EP0oRDFN	0	W	EP0o Read Completion 0: Performs no operation 1: Writes 1 to this bit after reading data for EP0o OUT FIFO. This trigger enables the next packet to be received.
1	EP0iPKTE	0	W	EP0i Packet Enable 0: Performs no operation 1: Generates a trigger to enable data transfer to the EP0i IN FIFO.

Bit	Bit Name	Initial Value	R/W	Description
0	EP0sRDFN	0	W	EP0s Read Completion 0: Performs no operation. A NAK handshake is returned in response to transmit/receive requests from the host in the data stage until 1 is written to this bit. 1: Writes 1 to this bit after reading data for EP0s OUT FIFO. After receiving the setup command, this trigger enables the next packet to be received by the EP0i and EP0o in the data stage. EP0s can always be overwritten and receive data regardless of this trigger.

Note: As triggers to EP3i and EP3o for Isochronous transfer are automatically generated each time the SOF packet is received from the host, the user need not generate triggers to EP3i and EP3o. Accordingly, data write to UEDR3i and data read from UEDR3o must be completed before the next packet has been received.

15.3.6 USB Trigger Register 1 (UTRG1)

UTRG1 generates one-shot triggers to the FIFO for each endpoint EP4 and EP5.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
2	EP5iPKTE	0	W	EP5i Packet Enable 0: Performs no operation 1: Generates a trigger to enable data transfer to the EP5i IN FIFO.
1	EP4oRDFN	0	W	EP4o Read Completion 0: Performs no operation 1: Writes 1 to this bit after reading data for EP4o OUT FIFO. EP4o FIFO has a dual FIFO configuration. This trigger is generated to the currently effective FIFO.
0	EP4iPKTE	0	W	EP4i Packet Enable 0: Performs no operation 1: Generates a trigger to enable data transfer to the EP4i IN FIFO. EP4i FIFO has a dual FIFO configuration. This trigger is generated for the currently effective FIFO.

15.3.7 USBFIFO Clear Register 0 (UFCLR0)

UFCLR0 is a one-shot register used to clear the FIFO for each end point from EP0 to EP3. Writing 1 to a bit clears the data in the corresponding FIFO. For IN FIFO, writing 1 to a bit in UFCLR0 clears the data for which the corresponding PKTE bit in UTRG0 is cleared to 0 after data write, or data that is validated by setting the corresponding PKTE bit in UTRG0. For OUT FIFO, writing 1 to a bit in UFCLR0 clears data that has not been fixed during reception or received data for which the corresponding RDFN bit is not set to 1. Accordingly, care must be taken not to clear data that is currently being received or transmitted. EP2i, EP2o, EP3i, and EP3o FIFOs, having a dual FIFO configuration, are cleared by entire FIFOs. Note that this trigger does not clear the corresponding interrupt flag.

Bit	Bit Name	Initial Value	R/W	Description
7	EP3oCLR	0	W	EP3o clear 0: Performs no operation 1: Clears EP3o OUT FIFO
6	EP3iCLR	0	W	EP3i clear 0: Performs no operation 1: Clears EP3i IN FIFO
5	EP2oCLR	0	W	EP2o clear 0: Performs no operation 1: Clears EP2o OUT FIFO
4	EP2iCLR	0	W	EP2i clear 0: Performs no operation 1: Clears EP2i IN FIFO
3	EP1iCLR	0	W	EP1i clear 0: Performs no operation 1: Clears EP1i IN FIFO
2	EP0oCLR	0	W	EP0o clear 0: Performs no operation 1: Clears EP0o OUT FIFO
1	EP0iCLR	0	W	EP0i clear 0: Performs no operation 1: Clears EP0i IN FIFO
0	—	0	R	Reserved This bit is always read as 0 and cannot be modified.

15.3.8 USBFIFO Clear Register 1 (UFCLR1)

UFCLR1 is a one-shot register used to clear the FIFO for each endpoint from EP4 to EP5. Writing 1 to a bit clears the data in the corresponding FIFO. For IN FIFO, writing 1 to a bit in UFCLR1 clears the data for which the corresponding PKTE bit in UTRG1 is cleared to 0 after data write, or data that is validated by setting the corresponding PKTE bit in UTRG1. For OUT FIFO, writing 1 to a bit in UFCLR1 clears data that has not been fixed during reception or received data for which the corresponding read completion bit is not set to 1. Accordingly, care must be taken not to clear data that is currently being received or transmitted. EP4i and EP4o FIFOs, having a dual FIFO configuration, are cleared by entire FIFOs. Note that this trigger does not clear the corresponding interrupt flag.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
2	EP5iCLR	0	W	EP5i clear 0: Performs no operation 1: Clears EP5i IN FIFO
1	EP4oCLR	0	W	EP4o clear 0: Performs no operation 1: Clears EP4o OUT FIFO
0	EP4iCLR	0	W	EP4i clear 0: Performs no operation 1: Clears EP4i IN FIFO

15.3.9 USB Endpoint Stall Register 0 (UESTL0)

UESTL0 is used to forcibly stall the endpoints for EP0 to EP3. While the bit is set to 1, the corresponding endpoint returns a stall handshake to the host. However, note that EP3 (Isochronous transfer) does not return a stall handshake.

The stall bit for endpoint 0 (EP0STL) is cleared automatically on reception of 8-bit command data for which decoding is performed by the function. When the SetupTS flag in UIFR0 is set, a write of 1 to the EP0STL bit is ignored. For details, refer to section 15.5.11, Stall Operations.

Bit	Bit Name	Initial Value	R/W	Description
7	EP3oSTL	0	R/W	EP3o stall 0: Cancels the EP3o stall state 1: Places the EP3o stall state
6	EP3iSTL	0	R/W	EP3i stall 0: Cancels the EP3i stall state 1: Places the EP3i stall state When the EP3i is placed in the stall state, a 0-length packet is returned for the first IN token. For the following IN token, nothing is returned.
5	EP2oSTL	0	R/W	EP2o stall 0: Cancels the EP2o stall state 1: Places the EP2o stall state
4	EP2iSTL	0	R/W	EP2i stall 0: Cancels the EP2i stall state 1: Places the EP2i stall state
3	EP1iSTL	0	R/W	EP1i stall 0: Cancels the EP1i stall state 1: Places the EP1i stall state
2 ,1	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
0	EP0STL	0	R/W	EP0 stall 0: Cancels the EP0 stall state 1: Places the EP0 stall state

15.3.10 USB Endpoint Stall Register 1 (UESTL1)

UESTL1 is used to forcibly stall the endpoints for EP4 and EP5. In addition, UESTL1 can cancel all endpoint stall states. While the bit is set to 1, the corresponding endpoint returns a stall handshake to the host. For details, refer to section 15.5.11, Stall Operations.

Bit	Bit Name	Initial Value	R/W	Description
7	SCME	0	R/W	Reserved The write value should always be 0.
6 to 3	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
2	EP5iSTL	0	R/W	EP5i stall 0: Cancels the EP5i stall state 1: Places the EP5i stall state
1	EP4oSTL	0	R/W	EP4o stall 0: Cancels the EP4o stall state 1: Places the EP4o stall state
0	EP4iSTL	0	R/W	EP4i stall 0: Cancels the EP4i stall state 1: Places the EP4i stall state

15.3.11 USB Endpoint Data Register 0s (UEDR0s)

UEDR0s stores the setup command for endpoint 0s (for Control_out transfer). UEDR0s stores 8-byte command data sent from the host in setup stage.

For details on USB operation when the data for the next setup stage is received while data in UEDR0s is being read, refer to section 15.9, Usage Notes.

UEDR0s is a byte register to which 4-byte address area is assigned. Accordingly, UEDR0s allows the user to read 2-byte or 4-byte data by word transfer or longword transfer.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	—	R	These bits store setup command for Control_out transfer

15.3.12 USB Endpoint Data Register 0i (UEDR0i)

UEDR0i is a data register for endpoint 0i (for Control_in transfer). UEDR0i stores data to be sent to the host. The number of data items to be written continuously must be the maximum packet size or less.

UEDR0i is a byte register to which 4-byte address area is assigned. Accordingly, UEDR0i allows the user to write 2-byte or 4-byte data by word transfer or longword transfer.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	All 0	W	These bits store data for Control_in transfer

15.3.13 USB Endpoint Data Register 0o (UEDR0o)

UEDR0o is a data register for endpoint 0o (for Control_out transfer). UEDR0o stores data received from the host. The number of data items to be read must be specified by UESZ0o.

When 1 byte is read from UEDR0o, UESZ0o is decremented by 1.

UEDR0o is a 1-byte register to which a 4-byte address area is assigned. Accordingly, UEDR0o allows the user to read 2-byte or 4-byte data by word transfer or longword transfer.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	—	R	These bits store data for Control_out transfer

15.3.14 USB Endpoint Data Register 1i (UEDR1i)

UEDR1i is a data register for endpoint 1i (for Interrupt_in transfer). UEDR1i stores data to be sent to the host. The number of data items to be written continuously must be the maximum packet size or less.

UEDR1i is a byte register to which 4-byte address area is assigned. Accordingly, UEDR1i allows the user to write 2-byte or 4-byte data by word transfer or longword transfer.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	All 0	W	These bits store data for Interrupt_in transfer

15.3.15 USB Endpoint Data Register 2i (UEDR2i)

UEDR2i is a data register for endpoint 2i (for Bulk_in transfer). UEDR2i stores data to be sent to the host. The number of data items to be written continuously must be the maximum packet size or less.

UEDR2i is a byte register to which 4-byte address area is assigned. Accordingly, UEDR2i allows the user to write 2-byte or 4-byte data by word transfer or longword transfer.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	All 0	W	These bits store data for Bulk_in transfer

15.3.16 USB Endpoint Data Register 2o (UEDR2o)

UEDR2o is a data register for endpoint 2o (for Bulk_out transfer). UEDR2o stores data received from the host. The number of data items to be read must be specified by UESZ2o.

When 1 byte is read from UEDR2o, UESZ2o is decremented by 1.

UEDR2o is a byte register to which 4-byte address area is assigned. Accordingly, UEDR2o allows the user to read 2-byte or 4-byte data by word transfer or longword transfer.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	—	R	These bits store data for Bulk_out transfer

15.3.17 USB Endpoint Data Register 3i (UEDR3i)

UEDR3i is a data register for endpoint 3i (for Isochronous_in transfer). UEDR3i stores data to be sent to the host. The number of data items to be written continuously must be the maximum packet size or less.

All data items must be written to before the next SOF packet is received.

UEDR3i is a byte register to which 4-byte address area is assigned. Accordingly, UEDR3i allows the user to write 2-byte or 4-byte data by word transfer or longword transfer.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	All 0	W	These bits store data for Isochronous_in transfer

15.3.18 USB Endpoint Data Register 3o (UEDR3o)

UEDR3o is a data register for endpoint 3o (for Isochronous_out transfer). UEDR3o stores data received from the host. The number of data items to be read must be specified by UESZ3o.

When 1 byte is read from UEDR3o, UESZ3o is decremented by 1.

All data items must be read before the next SOF packet is received.

UEDR3o is a byte register to which 4-byte address area is assigned. Accordingly, UEDR3o allows the user to read 2-byte or 4-byte data by word transfer or longword transfer.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	—	R	These bits store data for Isochronous_out transfer

15.3.19 USB Endpoint Data Register 4i (UEDR4i)

UEDR4i is a data register for endpoint 4i (for Bulk_in transfer). UEDR4i stores data to be sent to the host. The number of data items to be written continuously must be the maximum packet size or less.

UEDR4i is a byte register to which 4-byte address area is assigned. Accordingly, UEDR4i allows the user to write 2-byte or 4-byte data by word transfer or longword transfer.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	All 0	W	These bits store data for Bulk_in transfer

15.3.20 USB Endpoint Data Register 4o (UEDR4o)

UEDR4o is a data register for endpoint 4o (for Bulk_out transfer). UEDR4o stores data received from the host. The number of data items to be read must be specified by UESZ4o.

When 1 byte is read from UEDR4o, UESZ4o is decremented by 1.

UEDR4o is a byte register to which 4-byte address area is assigned. Accordingly, UEDR4o allows the user to read 2-byte or 4-byte data by word transfer or longword transfer.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	—	R	These bits store data for Bulk_out transfer

15.3.21 USB Endpoint Data Register 5i (UEDR5i)

UEDR5i is a data register for endpoint 5i (for Interrupt_in transfer). UEDR5i stores data to be sent to the host. The number of data items to be written continuously must be the maximum packet size or less.

UEDR5i is a byte register to which 4-byte address area is assigned. Accordingly, UEDR5i allows the user to write 2-byte or 4-byte data by word transfer or longword transfer.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	All 0	W	These bits store data for Interrupt_in transfer

15.3.22 USB Endpoint Receive Data Size Register 0o (UESZ0o)

UESZ0o is the receive data size register for endpoint 0o (for Control_out transfer). UESZ0o indicates the number of bytes of data to be received from the host.

Note that UESZ0o is decremented by 1 every time when 1 byte is read from UEDR0o.

Bit	Bit Name	Initial Value	R/W	Description
7	—	—	R	Reserved
6 to 0	D6 to D0	—	R	These bits indicate the size of data to be received in Control_out transfer

15.3.23 USB Endpoint Receive Data Size Register 2o (UESZ2o)

UESZ2o is the receive data size register for endpoint 2o (for Bulk_out transfer). UESZ2o indicates the number of bytes of data to be received from the host.

Note that UESZ2o is decremented by 1 every time when 1 byte is read from UEDR2o.

The FIFO for endpoint 2o (for Bulk_out transfer) has a dual-FIFO configuration. The data size indicated by this register refers to the currently selected FIFO.

Bit	Bit Name	Initial Value	R/W	Description
7	—	—	R	Reserved
6 to 0	D6 to D0	—	R	These bits indicate the size of data to be received in Bulk_out transfer

15.3.24 USB Endpoint Receive Data Size Register 3o (UESZ3o)

UESZ3o is the receive data size register for endpoint 3o (for Isochronous_out transfer). UESZ3o indicates the number of bytes of data to be received from the host.

Note that UESZ3o is decremented by 1 every time when 1 byte is read from UEDR3o.

The FIFO for endpoint 3o (for Isochronous_out transfer) has a dual-FIFO configuration. The data size indicated by this register refers to the currently selected FIFO.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	—	R	These bits indicate the size of data to be received in Isochronous_out transfer

15.3.25 USB Endpoint Receive Data Size Register 4o (UESZ4o)

UESZ4o is the receive data size register for endpoint 4o (for Bulk_out transfer). UESZ4o indicates the number of bytes of data to be received from the host.

Note that UESZ4o is decremented by 1 every time when 1 byte is read from UEDR4o.

The FIFO for endpoint 4o (for Bulk_out transfer) has a dual-FIFO configuration. The data size indicated by this register refers to the currently selected FIFO.

Bit	Bit Name	Initial Value	R/W	Description
7	—	—	R	Reserved
6 to 0	D6 toD0	—	R	These bits indicate the size of data to be received in Bulk_out transfer

15.3.26 USB Interrupt Flag Register 0 (UIFR0)

UIFR0 is an interrupt flag register indicating the setup command reception, EP0 and EP1 transmission/reception, and bus reset states. If the corresponding bit is set to 1, the corresponding $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$ interrupt is requested to the CPU. A bit in this register can be cleared by writing 0 to it. Writing 1 to a bit is invalid and causes no operation.

Consequently, to clear only a specific flag it is necessary to write 0 to the bit corresponding to the flag to be cleared and 1 to all the other bits. (To clear bit 5 only, write H'DF.) The bit-clear instruction is a read/modify/write instruction. There is a danger that the wrong bits may be cleared if a new flag is set between the read and write. Therefore, the bit-clear instruction should not be used to clear bits in this interrupt flag register.

Bit	Bit Name	Initial Value	R/W	Description
7	BRST	0	R/(W)*	Bus Reset Set to 1 when the bus reset signal is detected on the USB bus. The corresponding interrupt output is $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$. Note that BRST is also set to 1if D+ is not pulled-up during USB cable connection.
6	—	0	R	Reserved This bit is always read as 0 and cannot be modified.
5	EP1iTR	0	R/(W)*	EP1i Transfer Request Set to 1 if there is no valid transmit data in the FIFO when an IN token is sent from the host to EP1i. The corresponding interrupt output is $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$.
4	EP1iTS	0	R/(W)*	EP1i Transfer Complete Set to 1 if the transmit data written in EP1i is transferred to the host normally and the ACK handshake is returned. The corresponding interrupt output is $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$.
3	EP0oTS	0	R/(W)*	EP0o Receive Complete Set to 1 if the EP0o receives data from the host normally and returns the ACK handshake to the host. The corresponding interrupt output is $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$.
2	EP0iTR	0	R/(W)*	EP0i Transmit Request Set to 1 if there is no valid transmit data in the FIFO when an IN token is sent from the host to EP0i. The corresponding interrupt output is $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$.
1	EP0iTS	0	R/(W)*	EP0i Transmit Complete Set to 1 if the transmit data written in EP0i is transferred to the host normally and the ACK handshake is returned. The corresponding interrupt output is $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$.
0	SetupTS	0	R/(W)*	Setup Command Receive Complete Set to 1 if the EP0s normally receives 8-byte data to be decoded by the function from the host and returns the ACK handshake to the host. The corresponding interrupt output is $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$.

Note: * The write value should always be 0 to clear this flag.

15.3.27 USB Interrupt Flag Register 1 (UIFR1) (Only in H8S/2215)

UIFR1 is an interrupt flag register indicating the EP2i, EP2o, EP3i, and EP3o. If the corresponding bit is set to 1, the corresponding $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$ interrupt is requested from the CPU. EP2iTR and EP3iTR flags can cleared by writing 0 to them. Writing 1 to them is invalid and causes no operation. However, EP2iEMPTY, EP2oREDY, EP3oTS and EP3oTF are status bits, and cannot be cleared.

Bit	Bit Name	Initial Value	R/W	Description
7	EP3oTF	0	R	EP3o Abnormal Receive Indicates the status of EP3o FIFO, which can be read after the next SOF packet has been received following the data transmission from the host. This flag is set to 1 if a PID error, CRC error, bit staff error, data size error, or Bad EOP occurs when the data is transferred from the host to the EP3o. This is a status bit and cannot be cleared. In addition, an interrupt cannot be requested by this flag.
6	EP3oTS	0	R	EP3o Normal Receive Indicates the status of EP3o FIFO, which can be read after the next SOF packet has been received following the data transmission from the host. This flag is set to 1 if data is normally transferred from the host to the EP3o. This is a status bit and cannot be cleared. In addition, an interrupt cannot be requested by this flag.
5	EP3iTF	0	R/(W)*	EP3i Abnormal Transfer Set to 1 if data to be written to the EP3i FIFO is lost because no IN token has been returned. This flag is set when the SOF packet that is two packets after the data write is received. The corresponding interrupt output is $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$.
4	EP3iTR	0	R/(W)*	EP3i Transmit Request Set to 1 if there is no valid transmit data in the FIFO to be accessed by the UDC when an IN token is sent from the host to EP3i. The corresponding interrupt output is $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$.

Bit	Bit Name	Initial Value	R/W	Description
3	—	0	R	Reserved This bit is always read as 0 and cannot be modified.
2	EP2oREADY	0	R	EP2o Data Ready EP2o FIFO has a dual-FIFO configuration. This flag is set if there is a valid data in at least one EP2o FIFO. This flag is cleared to 0 if there is no valid data in EP2o FIFO. This flag is a status flag and cannot be cleared. The corresponding interrupt output is $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$.
1	EP2iTR	0	R/(W)*	EP2i Transmit Request Set to 1 if the EP2i FIFO is empty when an IN token is sent from the host to EP2i. The corresponding interrupt output is $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$.
0	EP2iEMPTY	1	R	EP2i FIFO Empty EP2i FIFO has a dual-FIFO configuration. This flag is set if at least one EP2i FIFO is empty. This flag is cleared to 0 if EP2i FIFO is full. This flag is a status flag and cannot be cleared. The corresponding interrupt output is $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$.

Note: * The write value should always be 0 to clear this flag.

15.3.28 USB Interrupt Flag Register 1 (UIFR1) (Only in H8S/2215R and H8S/2215T)

UIFR1 is an interrupt flag register indicating the EP2i, EP2o, EP3i, and EP3o. If the corresponding bit is set to 1, the corresponding $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$ interrupt is requested from the CPU. EP2iTR and EP3iTR flags can cleared by writing 0 to them. Writing 1 to them is invalid and causes no operation. Consequently, to clear only a specific flag it is necessary to write 0 to the bit corresponding to the flag to be cleared and 1 to all the other bits. (To clear bit 5 only, write H'DF.) The bit-clear instruction is a read/modify/write instruction. There is a danger that the wrong bits may be cleared if a new flag is set between the read and write. Therefore, the bit-clear instruction should not be used to clear bits in this interrupt flag register. However, EP2iEMPTY, EP2oREDY, EP3oTS and EP3oTF are status bits, and cannot be cleared.

Bit	Bit Name	Initial Value	R/W	Description
7	EP3oTF	0	R	EP3o Abnormal Receive Indicates the status of EP3o FIFO, which can be read after the next SOF packet has been received following the data transmission from the host. This flag is set to 1 if a PID error, CRC error, bit staff error, data size error, or Bad EOP occurs when the data is transferred from the host to the EP3o. This is a status bit and cannot be cleared. In addition, an interrupt cannot be requested by this flag.
6	EP3oTS	0	R	EP3o Normal Receive Indicates the status of EP3o FIFO, which can be read after the next SOF packet has been received following the data transmission from the host. This flag is set to 1 if data is normally transferred from the host to the EP3o. This is a status bit and cannot be cleared. In addition, an interrupt cannot be requested by this flag.
5	EP3iTF	0	R/(W)*	EP3i Abnormal Transfer Set to 1 if data to be written to the EP3i FIFO is lost because no IN token has been returned. This flag is set when the SOF packet that is two packets after the data write is received. The corresponding interrupt output is $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$.
4	EP3iTR	0	R/(W)*	EP3i Transmit Request Set to 1 if there is no valid transmit data in the FIFO to be accessed by the UDC when an IN token is sent from the host to EP3i. The corresponding interrupt output is $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$.

Bit	Bit Name	Initial Value	R/W	Description
3	EP2iALL EMPTYS	1	R	EP2i FIFO All Empty Status EP2i FIFO has a dual FIFO configuration. This flag is set to 1 if both FIFOs are empty. (Corresponds to a UDSR/EP2iDE negative-polarity signal.)
2	EP2oREADY	0	R	EP2o Data Ready EP2o FIFO has a dual-FIFO configuration. This flag is set if there is a valid data in at least one EP2o FIFO. This flag is cleared to 0 if there is no valid data in EP2o FIFO. This flag is a status flag and cannot be cleared. The corresponding interrupt output is $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$.
1	EP2iTR	0	R/(W)*	EP2i Transmit Request Set to 1 if the EP2i FIFO is empty when an IN token is sent from the host to EP2i. The corresponding interrupt output is $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$.
0	EP2iEMPTY	1	R	EP2i FIFO Empty EP2i FIFO has a dual-FIFO configuration. This flag is set if at least one EP2i FIFO is empty. This flag is cleared to 0 if EP2i FIFO is full. This flag is a status flag and cannot be cleared. The corresponding interrupt output is $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$.

Note: * The write value should always be 0 to clear this flag.

15.3.29 USB Interrupt Flag Register 2 (UIFR2) (Only in H8S/2215)

UIFR2 is an interrupt flag register indicating the state of EP4i, EP4o, and EP5i. If the corresponding bit is set to 1, the corresponding $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$ interrupt is requested to the CPU. EP4iTR EP5iTS and EP4iTR flags can cleared by writing 0 to them. Writing 1 to them is invalid and causes no operation. However, EP4iEMPTY and EP4oREADY are status bits indicating the EP4i and EP4o FIFO status, and cannot be cleared.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
5	EP5iTR	0	R/(W)*	EP5i Transfer Request Set to 1 if there is no valid transmit data in the FIFO when an IN token is sent from the host to EP5i. The corresponding interrupt output is $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$.
4	EP5iTS	0	R/(W)*	EP5i Transfer Complete Set to 1 if the transmit data written in EP5i is transferred to the host normally and the ACK handshake is returned. The corresponding interrupt output is $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$.
3	—	0	R	Reserved This bit is always read as 0 and cannot be modified.
2	EP4oREADY	0	R	EP4o Data Ready EP4o FIFO has a dual-FIFO configuration. This flag is set if there is a valid data in at least one EP4o FIFO. This flag is cleared to 0 if there is no valid data in EP4o FIFO. This flag is a status flag and cannot be cleared. The corresponding interrupt output is $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$.
1	EP4iTR	0	R/(W)*	EP4i Transfer Request Set to 1 if the EP4i FIFO is empty when an IN token is sent form the host to EPi4. The corresponding interrupt output is $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$.
0	EP4iEMPTY	1	R	EP4i FIFO Empty EP4i FIFO has a dual-FIFO configuration. This flag is set if at least one EP4i FIFO is empty. This flag is cleared to 0 if EP4i FIFO is full. This flag is a status flag and cannot be cleared. The corresponding interrupt output is $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$.

Note: * The write value should always be 0 to clear this flag.

15.3.30 USB Interrupt Flag Register 2 (UIFR2) (Only in H8S/2215R and H8S/2215T)

UIFR2 is an interrupt flag register indicating the state of EP4i, EP4o, and EP5i. If the corresponding bit is set to 1, the corresponding $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$ interrupt is requested to the CPU. EP4iTR EP5iTS and EP4iTR flags can cleared by writing 0 to them. Writing 1 to them is invalid and causes no operation. Consequently, to clear only a specific flag it is necessary to write 0 to the bit corresponding to the flag to be cleared and 1 to all the other bits. (To clear bit 5 only, write H'DF.) The bit-clear instruction is a read/modify/write instruction. There is a danger that the wrong bits may be cleared if a new flag is set between the read and write. Therefore, the bit-clear instruction should not be used to clear bits in this interrupt flag register. However, EP4iEMPTY and EP4oREADY are status bits indicating the EP4i and EP4o FIFO status, and cannot be cleared.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
5	EP5iTR	0	R/(W)*	EP5i Transfer Request Set to 1 if there is no valid transmit data in the FIFO when an IN token is sent from the host to EP5i. The corresponding interrupt output is $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$.
4	EP5iTS	0	R/(W)*	EP5i Transfer Complete Set to 1 if the transmit data written in EP5i is transferred to the host normally and the ACK handshake is returned. The corresponding interrupt output is $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$.
3	EP4iALL EMPTYS	1	R	EP4i FIFO All Empty Status EP4i FIFO has a dual FIFO configuration. This flag is set to 1 if both FIFOs are empty. (Corresponds to a UDSR/EP4iDE negative-polarity signal.)
2	EP4oREADY	0	R	EP4o Data Ready EP4o FIFO has a dual-FIFO configuration. This flag is set if there is a valid data in at least one EP4o FIFO. This flag is cleared to 0 if there is no valid data in EP4o FIFO. This flag is a status flag and cannot be cleared. The corresponding interrupt output is $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$.

Note: * The write value should always be 0 to clear this flag.

Bit	Bit Name	Initial Value	R/W	Description
1	EP4iTR	0	R/(W)*	EP4i Transfer Request Set to 1 if the EP4i FIFO is empty when an IN token is sent form the host to EPi4. The corresponding interrupt output is $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$.
0	EP4iEMPTY	1	R	EP4i FIFO Empty EP4i FIFO has a dual-FIFO configuration. This flag is set if at least one EP4i FIFO is empty. This flag is cleared to 0 if EP4i FIFO is full. This flag is a status flag and cannot be cleared. The corresponding interrupt output is $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$.

Note: * The write value should always be 0 to clear this flag.

15.3.31 USB Interrupt Flag Register 3 (UIFR3)

UIFR3 is an interrupt flag register indicating the USB status. If the corresponding bit is set to 1, the corresponding $\overline{\text{EXIRQ0}}$, $\overline{\text{EXIRQ1}}$, or $\overline{\text{IRQ6}}$ interrupt is requested from the CPU. VBUSi, SPRSi, SETI, SETC, SOF, and CK48READY flags can be cleared by writing 0. Writing 1 to them is invalid and causes no operation. Consequently, to clear only a specific flag it is necessary to write 0 to the bit corresponding to the flag to be cleared and 1 to all the other bits. (To clear bit 5 only, write H'DF.) The bit-clear instruction is a read/modify/write instruction. There is a danger that the wrong bits may be cleared if a new flag is set between the read and write. Therefore, the bit-clear instruction should not be used to clear bits in this interrupt flag register. VBUSs and SPRSs are status flags and cannot be cleared.

Bit	Bit Name	Initial Value	R/W	Description
7	CK48READY	0	R/(W)*	USB Operating Clock (48 MHz) Stabilization Detection Set to 1 when the 48-MHz USB operating clock stabilization time has been automatically counted after USB module rest mode cancellation. The corresponding interrupt output is $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$. CK48READY can also operate in USB interface software reset state (the UIFRST bit of UCTLR is set to 1). Note that USB operating clock stabilization time differs according to the clock source, refer to the UCKS3 to UCKS0 bits of the UCTLR.
6	SOF	0	R/(W)*	Start of Frame Packet Detection Set to 1 if the SOF packet is detected. This flag can be used to start time stamp check, EP3i transmit data write, or EP3o receive data read timing in EP3 isochronous transfer. The corresponding interrupt output is $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$.
5	SETC	0	R/(W)*	Set_Configuration Command Detection Set to 1 if the Set_Configuration command is detected. The corresponding interrupt output is $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$.
4	SETI	0	R/(W)*	Set_Inferface Command Detection Set to 1 if the Set_Interface command is detected. The corresponding interrupt output is $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$.

Bit	Bit Name	Initial Value	R/W	Description
3	SPRSs	0	R	Suspend/Resume Status Indicates the suspend/resume status and cannot request an interrupt. 0: Indicates that the bus is in the normal state. 1: Indicates that the bus is in the suspend state.
2	SPRSi	0	R/(W)*	Suspend/Resume Interrupt Set to 1 if a transition from normal state to suspend state or suspend state to normal state has occurred. The corresponding interrupt output is $\overline{\text{IRQ6}}$. This bit can be used to cancel software standby state at resume.
1	VBUSs	0	R	VBUS Status Indicates the VBUS state by the USB cable connection and disconnection. An interrupt cannot be requested by the VBUSs. 0: Indicates that the VBUS (USB cable) bus is disconnected. 1: Indicates that the VBUS (USB cable) bus is connected.
0	VBUSi	0	R/(W)*	VBUS Interrupt Set to 1 if a VBUS state changes by USB cable connection or disconnection. The corresponding interrupt output is $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$.

Note: * The write value should always be 0 to clear this flag.

15.3.32 USB Interrupt Enable Register 0 (UIER0)

UIER0 enables the interrupt request indicated in the interrupt flag register 0 (UIFR0). When an interrupt flag is set while the corresponding bit in UIER0 is set to 1, an interrupt is requested by asserting the corresponding $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$ pin. Either $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$ must be selected by the interrupt select register 0 (UISR0).

Bit	Bit Name	Initial Value	R/W	Description
7	BRSTE	0	R/W	Enables the BRST interrupt
6	—	0	R	Reserved This bit is always read as 0.
5	EP1iTRE	0	R/W	Enables the EP1iTR interrupt
4	EP1iTSE	0	R/W	Enables the EP1iTS interrupt
3	EP0oTSE	0	R/W	Enables the EP0oTS interrupt
2	EP0iTRE	0	R/W	Enables the EP0iTR interrupt
1	EP0iTSE	0	R/W	Enables the EP0iTS interrupt
0	SetupTSE	0	R/W	Enables the SetupTS interrupt

15.3.33 USB Interrupt Enable Register 1 (UIER1) (Only in H8S/2215)

UIER1 enables the interrupt request indicated in the interrupt flag register 1 (UIFR1). When an interrupt flag is set while the corresponding bit in UIER1 is set to 1, an interrupt is requested by asserting the corresponding $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$ pin. Either $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$ must be selected by the interrupt select register 1 (UISR1).

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0.
5	EP3iTFE	0	R/W	Enables the EP3iTF interrupt
4	EP3iTRE	0	R/W	Enables the EP3iTR interrupt
3	—	0	R	Reserved This bit is always read as 0.
2	EP2oREADYE	0	R/W	Enables the EP2oREADY interrupt
1	EP2iTRE	0	R/W	Enables the EP2iTR interrupt
0	EP2iEMPTYE	0	R/W	Enables the EP2iEMPTYE interrupt

15.3.34 USB Interrupt Enable Register 1 (UIER1) (Only in H8S/2215R and H8S/2215T)

UIER1 enables the interrupt request indicated in the interrupt flag register 1 (UIFR1). When an interrupt flag is set while the corresponding bit in UIER1 is set to 1, an interrupt is requested by asserting the corresponding $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$ pin. Either $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$ must be selected by the interrupt select register 1 (UISR1).

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0.
5	EP3iTFE	0	R/W	Enables the EP3iTF interrupt
4	EP3iTRE	0	R/W	Enables the EP3iTR interrupt
3	EP2iALL EMPTYE	0	R/W	Enables EP2iALLEMPTYE interrupt
2	EP2oREADYE	0	R/W	Enables the EP2oREADY interrupt
1	EP2iTRE	0	R/W	Enables the EP2iTR interrupt
0	EP2iEMPTYE	0	R/W	Enables the EP2iEMPTYE interrupt

15.3.35 USB Interrupt Enable Register 2 (UIER2)

UIER2 enables the interrupt request indicated in the interrupt flag register 2 (UIFR2). When an interrupt flag is set while the corresponding bit in UIER2 is set to 1, an interrupt is requested by asserting the corresponding $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$ pin. Either $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$ must be selected by the interrupt select register 2 (UISR2).

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0.
5	EP5iTRE	0	R/W	Enables the EP5iTR interrupt
4	EP5iTSE	0	R/W	Enables the EP5iTS interrupt
3	—	0	R	Reserved This bit is always read as 0.
2	EP4oREADYE	0	R/W	Enables the EP4oREADY interrupt
1	EP4iTRE	0	R/W	Enables the EP4iTR interrupt
0	EP4iEMPTYE	0	R/W	Enables the EP4iEMPTY interrupt

15.3.36 USB Interrupt Enable Register 2 (UIER2) (Only in H8S/2215R and H8S/2215T)

UIER2 enables the interrupt request indicated in the interrupt flag register 2 (UIFR2). When an interrupt flag is set while the corresponding bit in UIER2 is set to 1, an interrupt is requested by asserting the corresponding $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$ pin. Either $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$ must be selected by the interrupt select register 2 (UISR2).

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0.
5	EP5iTRE	0	R/W	Enables the EP5iTR interrupt
4	EP5iTSE	0	R/W	Enables the EP5iTS interrupt
3	EP4iALL EMPTYE	0	R/W	Enables EP4iALLEMPTYE interrupt
2	EP4oREADYE	0	R/W	Enables the EP4oREADY interrupt
1	EP4iTRE	0	R/W	Enables the EP4iTR interrupt
0	EP4iEMPTYE	0	R/W	Enables the EP4iEMPTY interrupt

15.3.37 USB Interrupt Enable Register 3 (UIER3)

UIER3 enables the interrupt request indicated in the interrupt flag register 3 (UIFR3). This register is readable/writable even though in USB module stop mode. When an interrupt flag is set while the corresponding bit in UIER3 is set to 1, an interrupt is requested by asserting the corresponding $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$ pin. Either $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$ must be selected by the interrupt select register 3 (UISR3). Note, however, that the SPRSiE bit is an interrupt enable bit specific to the $\overline{\text{IRQ6}}$ pin and cannot be selected by UISR3.

Bit	Bit Name	Initial Value	R/W	Description
7	CK48READYE	1	R/W	Enables the CK48READY interrupt
6	SOFE	0	R/W	Enables the SOF interrupt
5	SETCE	0	R/W	Enables the SETC interrupt
4	SETIE	0	R/W	Enables the SETI interrupt
3	—	0	R	Reserved This bit is always read as 0.
2	SPRSiE	0	R/W	Enables the SPRSi interrupt (only for $\overline{\text{IRQ6}}$)
1	—	0	R	Reserved This bit is always read as 0.
0	VBUSiE	0	R/W	Enables the VBUSi interrupt

15.3.38 USB Interrupt Select Register 0 (UISR0)

UISR0 selects the $\overline{\text{EXIRQ}}$ pin to output interrupt request indicated in the interrupt flag register 0 (UIFR0). When a bit in UIER0 corresponding to the UISR0 bit is cleared to 0, an interrupt request is output to $\overline{\text{EXIRQ0}}$. When a bit in UIER0 corresponding to the UISR0 bit is set to 1, an interrupt request is output to $\overline{\text{EXIRQ1}}$.

Bit	Bit Name	Initial Value	R/W	Description
7	BRSTS	0	R/W	Selects the BRST interrupt
6	—	0	R	Reserved This bit is always read as 0.
5	EP1iTRS	0	R/W	Selects the EP1iTR interrupt
4	EP1iTSS	0	R/W	Selects the EP1iTS interrupt
3	EP0oTSS	0	R/W	Selects the EP0oTS interrupt
2	EP0iTRS	0	R/W	Selects the EP0iTR interrupt
1	EP0iTSS	0	R/W	Selects the EP0iTS interrupt
0	SetupTSS	0	R/W	Selects the SetupTS interrupt

15.3.39 USB Interrupt Select Register 1 (UISR1) (Only in H8S/2215)

UISR1 selects the $\overline{\text{EXIRQ}}$ pin to output interrupt request indicated in the interrupt flag register 1 (UIFR1). When a bit in UIER1 corresponding to the UISR1 bit is cleared to 0, an interrupt request is output to $\overline{\text{EXIRQ0}}$. When a bit in UIER1 corresponding to the UISR1 bit is set to 1, an interrupt request is output to $\overline{\text{EXIRQ1}}$.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0.
5	EP3iTFS	0	R/W	Selects the EP3iTF interrupt
4	EP3iTRS	0	R/W	Selects the EP3iTR interrupt
3	—	0	R	Reserved This bit is always read as 0.
2	EP2oREADYS	0	R/W	Selects the EP2oREADY interrupt
1	EP2iTRS	0	R/W	Selects the EP2iTR interrupt
0	EP2iEMPTYS	0	R/W	Selects the EP2iEMPTY interrupt

15.3.40 USB Interrupt Select Register 1 (UISR1) (Only in H8S/2215R and H8S/2215T)

UISR1 selects the $\overline{\text{EXIRQ}}$ pin to output interrupt request indicated in the interrupt flag register 1 (UIFR1). When a bit in UIER1 corresponding to the UISR1 bit is cleared to 0, an interrupt request is output to $\overline{\text{EXIRQ0}}$. When a bit in UIER1 corresponding to the UISR1 bit is set to 1, an interrupt request is output to $\overline{\text{EXIRQ1}}$.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0.
5	EP3iTFS	0	R/W	Selects the EP3iTF interrupt
4	EP3iTRS	0	R/W	Selects the EP3iTR interrupt
3	EP2iALL EMPTYS	0	R/W	Selects EP2iALLEMPTY interrupt
2	EP2oREADYS	0	R/W	Selects the EP2oREADY interrupt
1	EP2iTRS	0	R/W	Selects the EP2iTR interrupt
0	EP2iEMPTYS	0	R/W	Selects the EP2iEMPTY interrupt

15.3.41 USB Interrupt Select Register 2 (UISR2) (Only in H8S/2215)

UISR2 selects the $\overline{\text{EXIRQ}}$ pin to output interrupt request indicated in the interrupt flag register 2 (UIFR2). When a bit in UIER2 corresponding to the UISR2 bit is cleared to 0, an interrupt request is output to $\overline{\text{EXIRQ0}}$. When a bit in UIER2 corresponding to the UISR2 bit is set to 1, an interrupt request is output to $\overline{\text{EXIRQ1}}$.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0.
5	EP5iTRS	0	R/W	Selects the EP5iTR interrupt
4	EP5iTSS	0	R/W	Selects the EP5iTS interrupt
3	—	0	R	Reserved This bit is always read as 0.
2	EP4oREADYS	0	R/W	Selects the EP4oREADY interrupt
1	EP4iTRS	0	R/W	Selects the EP4iTR interrupt
0	EP4iEMPTYS	0	R/W	Selects the EP4iEMPTY interrupt

15.3.42 USB Interrupt Select Register 2 (UISR2) (Only in H8S/2215R and H8S/2215T)

UISR2 selects the $\overline{\text{EXIRQ}}$ pin to output interrupt request indicated in the interrupt flag register 2 (UIFR2). When a bit in UIER2 corresponding to the UISR2 bit is cleared to 0, an interrupt request is output to $\overline{\text{EXIRQ0}}$. When a bit in UIER2 corresponding to the UISR2 bit is set to 1, an interrupt request is output to $\overline{\text{EXIRQ1}}$.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0.
5	EP5iTRS	0	R/W	Selects the EP5iTR interrupt
4	EP5iTSS	0	R/W	Selects the EP5iTS interrupt
3	EP4iALL EMPTYS	0	R/W	Selects the EP4iALLEMPTY interrupt
2	EP4oREADYS	0	R/W	Selects EP4oREADY interrupt
1	EP4iTRS	0	R/W	Selects the EP4iTR interrupt
0	EP4iEMPTYS	0	R/W	Selects the EP4iEMPTY interrupt

15.3.43 USB Interrupt Select Register 3 (UISR3)

UISR3 selects the $\overline{\text{EXIRQ}}$ pin to output interrupt request indicated in the interrupt flag register 3 (UIFR3). When a bit in UIER3 corresponding to the UISR3 bit is cleared to 0, an interrupt request is output to $\overline{\text{EXIRQ0}}$. When a bit in UIER3 corresponding to the UISR3 bit is set to 1, an interrupt request is output to $\overline{\text{EXIRQ1}}$.

Bit	Bit Name	Initial Value	R/W	Description
7	CK48READYS	1	R/W	Selects the CK48READY interrupt
6	SOFS	0	R/W	Selects the SOF interrupt
5	SETCS	0	R/W	Selects the SETC interrupt
4	SETIS	0	R/W	Selects the SETI interrupt
3 to 1	—	All 0	R	Reserved These bits are always read as 0.
0	VBUSiS	0	R/W	Selects the VBUSi interrupt

15.3.44 USB Data Status Register (UDSR)

UDSR indicates whether the IN FIFO data registers (EP0i, EP1i, EP2i, EP4i, and EP5i) contain valid data or not. A bit in USDR is set when data written to the corresponding IN FIFO becomes valid after the corresponding PKTE bit in UTRG is set to 1. A bit in USDR is cleared when all valid data is sent to the host. For EP2i and EP4i, having a dual-FIFO configuration, the corresponding bit in USDR is cleared to 0 and FIFO becomes empty.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
5	EP5iDE	0	R	EP5i Data Enable 0: Indicates that the EP5i contains no valid data 1: Indicates that the EP5i contains valid data
4	EP4iDE	0	R	EP4i Data Enable 0: Indicates that the EP4i contains no valid data 1: Indicates that the EP4i contains valid data
3	—	0	R	Reserved This bit is always read as 0 and cannot be modified.
2	EP2iDE	0	R	EP2i Data Enable 0: Indicates that the EP2i contains no valid data 1: Indicates that the EP2i contains valid data
1	EP1iDE	0	R	EP1i Data Enable 0: Indicates that the EP1i contains no valid data 1: Indicates that the EP1i contains valid data
0	EP0iDE	0	R	EP0i Data Enable 0: Indicates that the EP0i contains no valid data 1: Indicates that the EP0i contains valid data

15.3.45 USB Configuration Value Register (UCVR)

UCVR stores the Configuration value, Interface Number value and Alternate Setting value when the Set_Configuration and Set_Interface commands are received from the host.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
5	CNFV0	0	R	Configuration Value 0 Stores the Configuration value when the Set_Configuration command is received. CNFV0 is modified when the SETC bit in UIFR3 is set to 1.
4	INTV1	0	R	Interface Number Value 1, 0
3	INTV0	0	R	Store the Interface number value when the Set_Interface command is received. INTV1 and INTV0 are modified when the SETI bit in UIFR3 is set to 1.
2	ATLV2	0	R	Alternate Setting Value 2 to 0
1	ATLV1	0	R	Store the Alternate Setting value when the Set_Interface command is received. ATLV2 to ATLV0 are modified when the SETI bit in UIFR3 is set to 1.
0	ATLV0	0	R	

15.3.46 USB Time Stamp Registers H, L (UTSRH, UTSRL)

UTSRH and UTSRL store the current time stamp values. The time stamp values in UTSRH and UTSRL are modified when the SOF flag in UIFR3 is set to 1.

UTSRH combined with UTSRL can also be handled as a 16-bit register. The USB module has an 8-bit bus. The upper byte of UTSRH can be read directly, while the lower byte of UTSRL is read through an 8-bit temporary register. Accordingly, UTSRH and UTSRL must be read in this order. If only UTSRL is read, the read data cannot be guaranteed. In addition, note that the time stamp automatic update function is not supported if the SOF packed has been broken even if the SOF marker function is enabled by setting the SFME bit of UCTLR.

- **UTSRH**

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0.
2 to 0	D10 to D8	All 0	R	Stores time stamp D10 to D8.

- **UTSRL**

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	All 0	R	Stores time stamp D7 to D0.

15.3.47 USB Test Register 0 (UTSTR0)

UTSTR0 controls internal or external transceiver output signals. After clearing UCTLR/UIFRST and UDCRST to 0, setting the PTSTE bit to 1 enable user setting of transceiver output. Table 15.3 shows the relationship between UTSTR0 settings and pin outputs.

Bit	Bit Name	Initial Value	R/W	Description
7	PTSTE	0	R/W	Pin Test Enable Enables the test control of the internal/external transceiver output signals. When FADSEL in UCTLR is 0, the test control for the internal transceiver output pins (USD+ and USD-) and USPND pin are enabled. When FADSEL in UCTLR is 1, the test control for the external transceiver output pins (P17/$\overline{OE}$, P15/FSE0, P13/VPO, and PA3/SUSPND) and USPND pin are enabled.
6 to 4	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
3	SUSPEND	0	R/W	Internal/External Transceiver Output Signal
2	$\overline{OE}$	1	R/W	Setting Bits
1	FSE0	0	R/W	SUSPEND: Specifies USPND and PA3/SUSPND pin.
0	VPO	0	R/W	$\overline{OE}$: Specifies internal transceiver $\overline{OE}$ signal and P17/$\overline{OE}$ pin. FSE0: Specifies internal transceiver FSE0 signal and P15/FSE0 pin. VPO: Specifies internal transceiver VPO signal and P13/VPO pin.

Table 15.3 Relationship between the UTSTR0 Setting and Pin Outputs

Pin Input	Register Setting						Pin Outputs						
VBUS	UCTLR/ FADSEL	PTSTE	SUSPEND	$\overline{OE}$	FSE0	VPO	USD+	USD-	USPND	PA3/ SUSPND	P17/ $\overline{OE}$	P15/ FSE0	P13/ VPO
0	×	0	×	×	×	×	—	—	—	—	—	—	—
0	0	1	0/1	×	×	×	Hi-Z	Hi-Z	0/1	—	—	—	—
0	1	1	0/1	×	×	×	Hi-Z	Hi-Z	0/1	1	1	—	—
0	1	1	×	×	0/1	×	Hi-Z	Hi-Z	—	1	1	0/1	—
0	1	1	×	×	×	0/1	Hi-Z	Hi-Z	—	1	1	—	0/1
1	×	0	×	×	×	×	—	—	—	—	—	—	—
1	0	1	0	0	0	0	0	1	0	—	—	—	—
1	0	1	0	0	0	1	1	0	0	—	—	—	—
1	0	1	0	0	1	×	0	0	0	—	—	—	—
1	0	1	0	1	×	×	Hi-Z	Hi-Z	0	—	—	—	—
1	0	1	1	0	0	0	0	1	1	—	—	—	—
1	0	1	1	0	0	1	1	0	1	—	—	—	—
1	0	1	1	0	1	×	0	0	1	—	—	—	—
1	0	1	1	1	×	×	Hi-Z	Hi-Z	1	—	—	—	—
1	1	1	0/1	×	×	×	Hi-Z	Hi-Z	0/1	0/1	—	—	—
1	1	1	×	0/1	×	×	Hi-Z	Hi-Z	—	—	0/1	—	—
1	1	1	×	×	0/1	×	Hi-Z	Hi-Z	—	—	—	0/1	—
1	1	1	×	×	×	0/1	Hi-Z	Hi-Z	—	—	—	—	0/1

Legend:

×: Don't care

0/1: Register setting equals pin output

—: Cannot be controlled. Indicates state in normal operation according to the USB operation and port settings.

15.3.48 USB Test Register 1 (UTSTR1)

UTSTR1 allows internal or external transceiver input signals to be monitored. When the FADSEL bit of UCTLR is set to 0, internal transceiver input signals can be monitored. When the FADSEL bit is FADSEL = 1, external transceiver input signals can be monitored. Table 15.4 shows the relationship between UTSTR1 settings and pin inputs.

Bit	Bit Name	Initial Value	R/W	Description
7	VBUS	—*	R	Internal/External Transceiver Input Signal Monitor Bits
6	$\overline{\text{UBPM}}$	—*	R	VBUS: Monitors VBUS pin $\overline{\text{UBPM}}$: Monitors $\overline{\text{UBPM}}$ pin
5 to 3	—	Al 0	R	Reserved These bits are always read as 0 and cannot be modified.
2	RCV	—*	R	Internal/External Transceiver Input Signal Monitor Bits RCV: Monitors the RCV signal of the internal/external transceiver VP: Monitors the VP signal of the internal/external transceiver VM: Monitors the VM signal of the internal/external transceiver
1	VP	—*	R	
0	VM	—*	R	

Note: * Determined by the status of the VBUS, $\overline{\text{UBPM}}$, USD+, USD-, RCV, VP, and VM pins.

Table 15.4 Relationship between the UTSTR1 Settings and Pin Inputs

Pin Input		UTSTR1 Monitor	
VBUS	**$\overline{\text{UBPM}}$**	**VBUS**	**$\overline{\text{UBPM}}$**
0/1	×	0/1	×
×	0/1	×	0/1

Register Settings			Pin Input						UTSTR1 Monitor		
UCTLR/ FADSEL	**UTSTR0/ PTSTE**	**UTSTR0/ SUSPEND**	**VBUS**	**USD+**	**USD-**	**P12/ RCV**	**P11/ VP**	**P10/ VM**	**RCV**	**VP**	**VM**
0	×	×	0	×	×	×	×	×	0	0	0
1	×	×	0	×	×	0/1	×	×	0/1	0	0
0	0	×	1	0	0	×	×	×	×	0	0
0	0	×	1	0	1	×	×	×	0	0	1
0	0	×	1	1	0	×	×	×	1	1	0
0	0	×	1	1	1	×	×	×	×	1	1
0	1	0	1	0	0	×	×	×	×	0	0
0	1	0	1	0	1	×	×	×	0	0	1
0	1	0	1	1	0	×	×	×	1	1	0
0	1	0	1	1	1	×	×	×	×	1	1
0	1	1	1	0/1	×	×	×	×	0	0/1	×
0	1	1	1	×	0/1	×	×	×	0	×	0/1
1	×	×	1	×	×	0/1	×	×	0/1	×	×
1	×	×	1	×	×	×	0/1	×	×	0/1	×
1	×	×	1	×	×	×	×	0/1	×	×	0/1

Legend:

×: Don't care

0/1: Register setting equals pin output

15.3.49 USB Test Registers 2 and A to F (UTSTR2, UTSRA to UTSRF)

UTSTR2 and UTSRTA to UTSRTF are test registers and cannot be written to.

15.3.50 Module Stop Control Register B (MSTPCRB)

Bit	Bit Name	Initial Value	R/W	Description
7	MSTPB7	1	R/W	Module Stop Bits
6	MSTPB6	1	R/W	For details, refer to section 22.1.2, Module Stop Control Registers A to C (MSTPCRA to MSTPCRC).
5	MSTPB5	1	R/W	
4	MSTPB4	1	R/W	
3	MSTPB3	1	R/W	
2	MSTPB2	1	R/W	
1	MSTPB1	1	R/W	
0	MSTPB0	1	R/W	Module Stop USB 0: Cancels USB module stop mode. A clock is provided for the USB module. After this bit has been cleared, the USB operating clock (48 MHz) oscillator or internal PLL circuit starts operation. Registers in the USB module must be accessed after the USB operating clock stabilization time (CK48READY bit of UIFR3 is set) has passed. 1: Places the USB module in stop mode. Both the USB operating clock (48 MHz) oscillator and internal PLL circuit stop operation. In this mode, the USB module register contents are maintained.

Note: For details on USB module stop mode cancellation procedure, refer to section 15.5, Communication Operation.

15.4 Interrupt Sources

This module has three interrupt signals. Table 15.5 shows the interrupt sources and their corresponding interrupt request signals. $\overline{\text{EXIRQ}}$ interrupt signals are activated at low level. The $\overline{\text{EXIRQ}}$ interrupt requests can only be detected at low level (specified as level sensitive). The suspend/resume interrupt request $\overline{\text{IRQ6}}$ must be specified to be detected at the falling edge (falling-edge sensitive) by the interrupt controller register.

Table 15.5 SCI Interrupt Sources

Register	Bit	Transfer Mode	Interrupt Source	Description	Interrupt Request Signal	DMAC Activation by USB Request*9
UIFR0	0	Control transfer (EP0)	SetupTS*1	Setup command receive completion	$\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$	×
	1		EP0iTS*1	EP0i transfer completion	$\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$	×
	2		EP0iTR*1	EP0i transfer request	$\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$	×
	3		EP0oTS *1	EP0o receive request	$\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$	×
	4	Interrupt_in transfer (EP1i)	EP1iTS	EP1i transfer completion	$\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$	×
	5		EP1iTR	EP1i transfer request	$\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$	×
	6	—	Reserved	—	—	—
	7	(Status)	BRST	Bus reset	$\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$	×
UIFR1	0	Bulk_in transfer (EP2i)	EP2iEMPTY	EP2i FIFO empty	$\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$	$\overline{\text{DREQ0}}$ or $\overline{\text{DREQ1}}$*2
	1		EP2iTR	EP2i transfer request	$\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$	×
	2	Bulk_out transfer (EP2o)	EP2oREADY	EP2o data ready	$\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$	$\overline{\text{DREQ0}}$ or $\overline{\text{DREQ1}}$*3
	3	Bulk_in transfer*7 (EP2i)	EP2iALLEMPTYS*8	EP2i all empty states*7	$\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$*7	×*7
	4	Isochronous_in Transfer (EP3i)	EP3iTR	EP3i transfer request	$\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$	×
	5		EP3iTF	EP3i abnormal transfer	$\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$	×
	6	Isochronous_out Transfer (EP3o)	EP3oTS	EP3o normal receive	×	×
	7		EP3oTF	EP3o abnormal receive	×	×

Register	Bit	Transfer Mode	Interrupt Source	Description	Interrupt Request Signal	DMAC Activation by USB Request*9
UIFR2	0	Bulk_in transfer (EP4i)	EP4iEMPTY	EP4i FIFO empty	$\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$	$\overline{\text{DREQ0}}$ or $\overline{\text{DREQ1}}$*4
	1		EP4iTR	EP4i transfer request	$\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$	×
	2	Bulk_out transfer (EP4o)	EP4oREADY	EP4o data ready	$\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$	$\overline{\text{DREQ0}}$ or $\overline{\text{DREQ1}}$*5
	3	Bulk_in transfer*7 (EP4i)	EP4iALLEMPTYS*8	EP4i all empty states*7	$\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$*7	×*7
	4	Interrupt_in transfer (EP5i)	EP5iTS	EP5i transfer completion	$\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$	×
	5		EP5iTR	EP5i transfer request	$\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$	×
	6	—	Reserved	—	—	×
	7	—	Reserved	—	—	×
UIFR3	0	— (Status)	VBUSi	VBUS interrupt	$\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$	×
	1		VBUSs	VBUS status	×	×
	2		SPRSi	Suspend/resume interrupt	$\overline{\text{IRQ6}}$ *6	×
	3		SPRSs	Suspend/resume status	×	×
	4		SETI	Set_Interface detection	$\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$	×
	5		SETC	Set_Configuration detection	$\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$	×
	6		SOF	Start of Frame packet detection	$\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$	×
	7		CK48READY	USB bus clock stabilization detection	$\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$	×

Notes:
1. EP0 interrupts must be assigned to the same interrupt request signal.
2. An EP2i DMA transfer by a USB request is specified by the EP2iT1 and EP2iT0 bits of UDMAR.
3. An EP2o DMA transfer by a USB request is specified by the EP2oT1 and EP2oT0 bits of UDMAR.
4. An EP4i DMA transfer by a USB request is specified by the EP4iT1 and EP4iT0 bits of UDMAR.
5. An EP4oDMA transfer by a USB request is specified by the EP4oT1 and EP4oT0 bits of UDMAR.
6. The suspend/resume interrupt request $\overline{\text{IRQ6}}$ must be specified to be detected at the falling edge (IRQ6SCB, A = 01 in ISCRH) by the interrupt controller register.
7. Available only in H8S/2215R and H8S/2215T. "—" in H8S/2215.
8. Available only in H8S/2215R and H8S/2215T. Reserved in H8S/2215.
9. The $\overline{\text{DREQ}}$ signal is not used for auto-request. The CPU can activate the DMAC using any flags and interrupts.

- $\overline{\text{EXIRQ0}}$ signal

  The $\overline{\text{EXIRQ0}}$ signal requests interrupt sources for which the corresponding bits in interrupt select registers 0 to 3 (UISR0 to UISR3) are cleared to 0. The $\overline{\text{EXIRQ0}}$ is driven low if a corresponding bit in the interrupt flag register is set to 1.

- $\overline{\text{EXIRQ1}}$ signal

  The $\overline{\text{EXIRQ1}}$ signal requests interrupt sources for which the corresponding bits in interrupt select registers 0 to 3 (UISR0 to UISR3) are cleared to 0. The $\overline{\text{EXIRQ1}}$ is driven low if a corresponding bit in the interrupt flag register is set to 1.

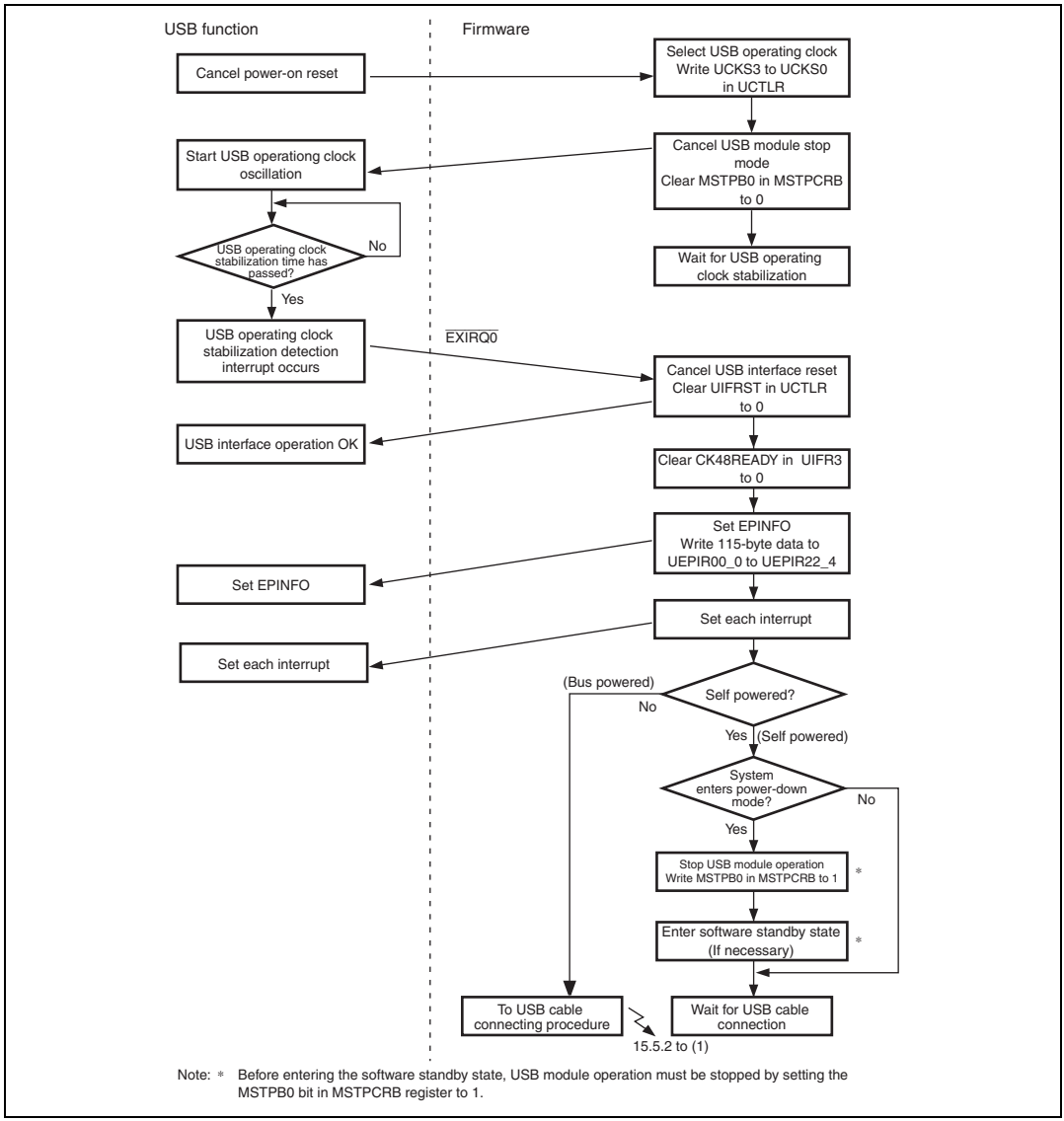
- $\overline{\text{IRQ6}}$ signal

  The $\overline{\text{IRQ6}}$ signal is specific to the suspend/resume interrupt request. The rising edge of the $\overline{\text{IRQ6}}$ signal is output at the transition from the suspend state or from the resume state.

15.5 Communication Operation

15.5.1 Initialization

The USB must be initialized as described in the flowchart in figure 15.3.



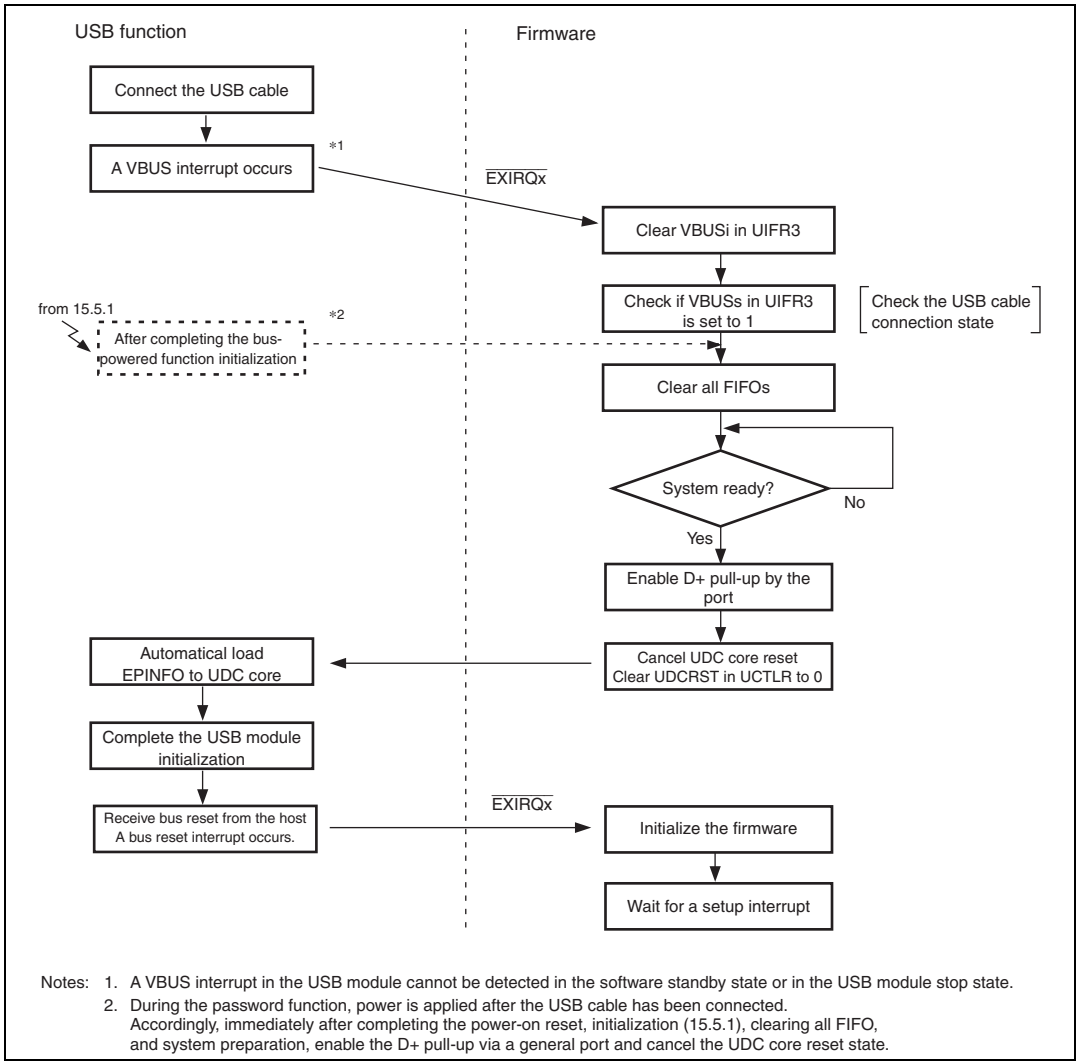
Note: * Before entering the software standby state, USB module operation must be stopped by setting the MSTPB0 bit in MSTPCRB register to 1.

Figure 15.3 USB Initialization

15.5.2 USB Cable Connection/Disconnection

(1) USB Cable Connection (When USB Module Stop or Software Standby Is Not Used)

If the USB cable enters the connection state from the disconnection state in an application (self powered) where USB module stop or software standby mode is not used, perform the operation shown in figure 15.4. In bus-powered mode, perform the operation described in note 2.

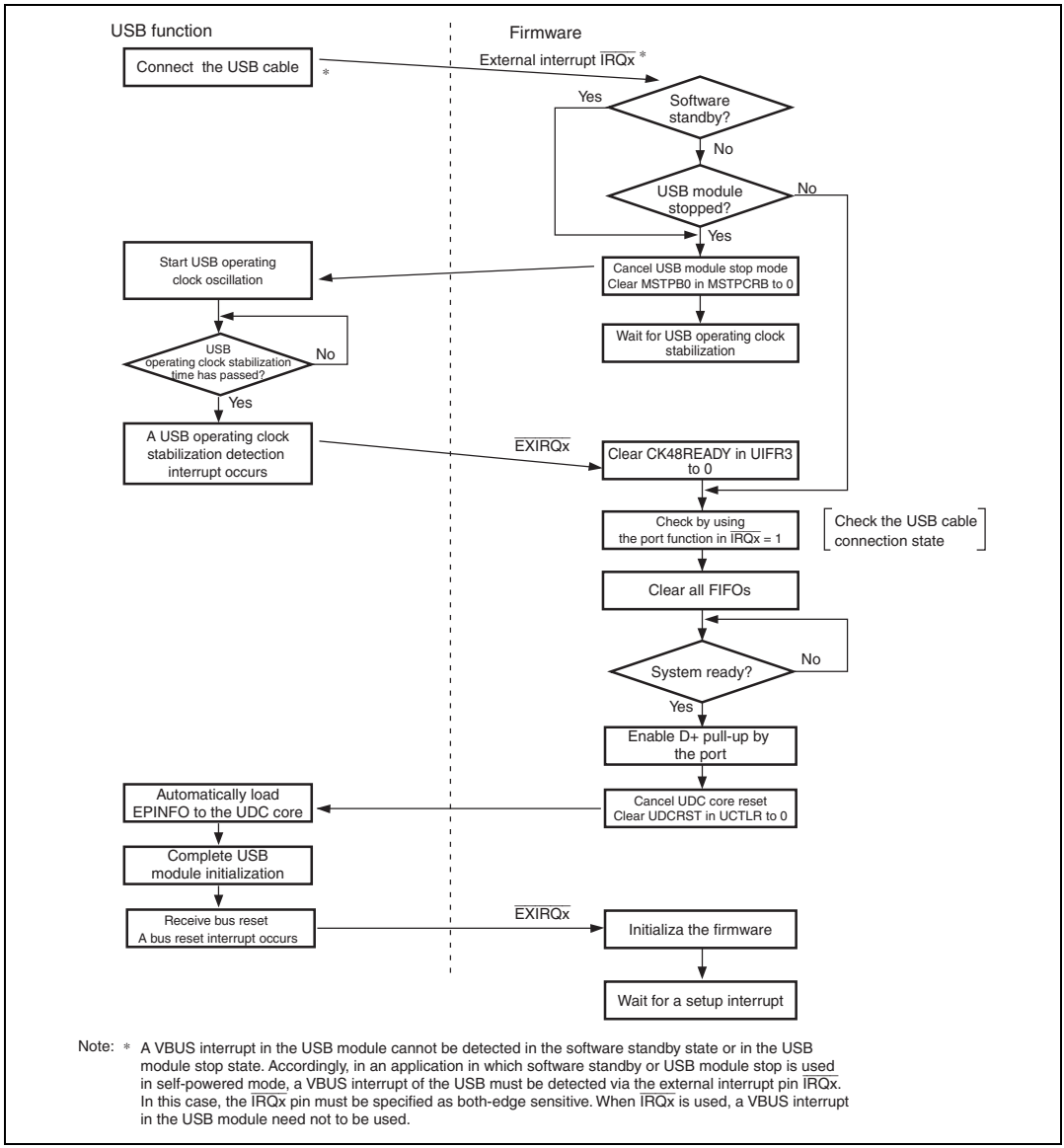


Notes: 1. A VBUS interrupt in the USB module cannot be detected in the software standby state or in the USB module stop state.
2. During the password function, power is applied after the USB cable has been connected. Accordingly, immediately after completing the power-on reset, initialization (15.5.1), clearing all FIFO, and system preparation, enable the D+ pull-up via a general port and cancel the UDC core reset state.

Figure 15.4 USB Cable Connection (When USB Module Stop or Software Standby Is Not Used)

(2) USB Cable Connection (When USB Module Stop or Software Standby Is Used)

If the USB cable enters the connection state from disconnection state an application (self powered) where USB module stop or software standby mode is used, perform the operation as shown in figure 15.5.



Note: * A VBUS interrupt in the USB module cannot be detected in the software standby state or in the USB module stop state. Accordingly, in an application in which software standby or USB module stop is used in self-powered mode, a VBUS interrupt of the USB must be detected via the external interrupt pin IRQx. In this case, the IRQx pin must be specified as both-edge sensitive. When IRQx is used, a VBUS interrupt in the USB module need not to be used.

Figure 15.5 USB Cable Connection (When USB Module Stop or Software Standby Is Used)

(3) USB Cable Disconnection (When USB Module Stop or Software Standby Is Not Used)

If the USB cable enters the disconnection state from the connection state in an application (self powered) where USB module stop or software standby mode is not used, perform the operation shown in figure 15.6. In bus-powered mode, the power is automatically turned off when the USB cable is disconnected and the following processing is not required.

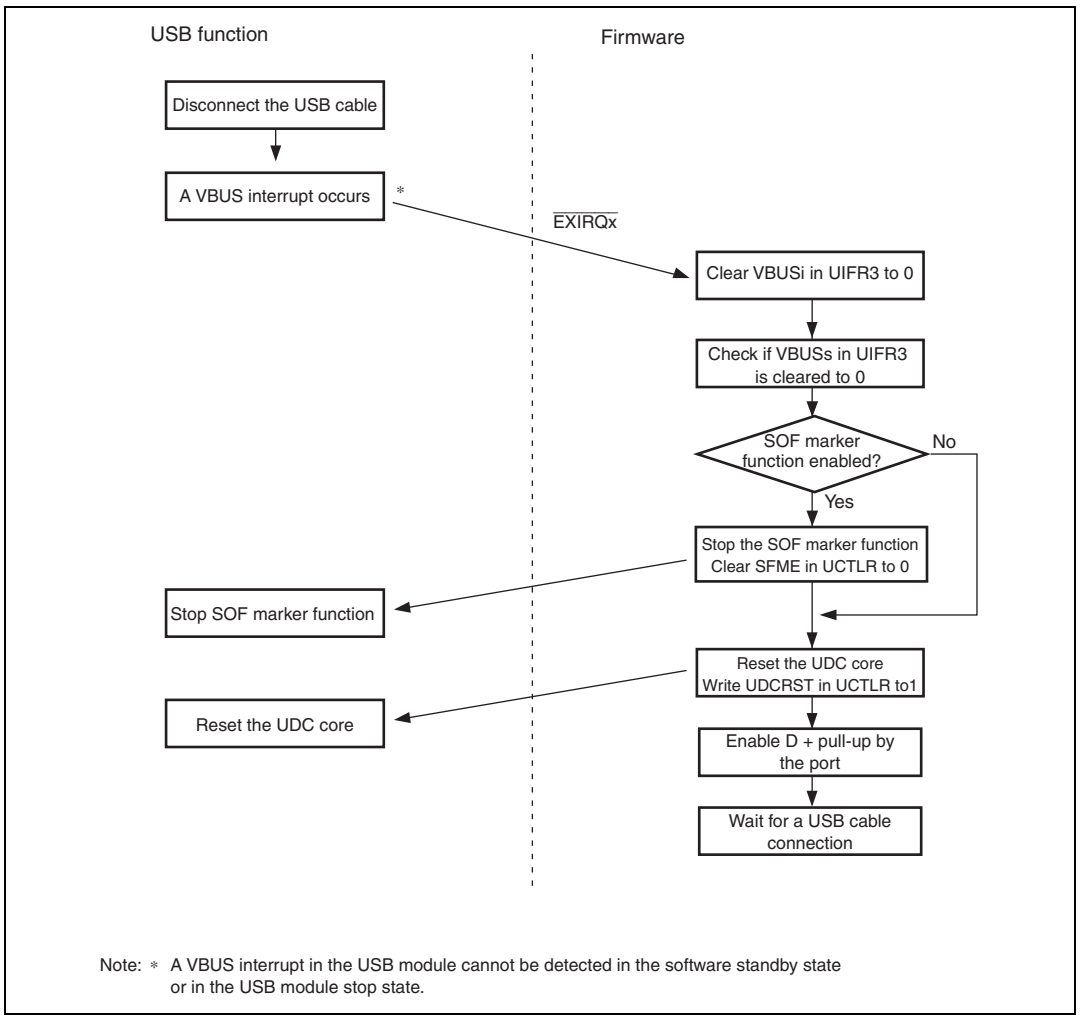
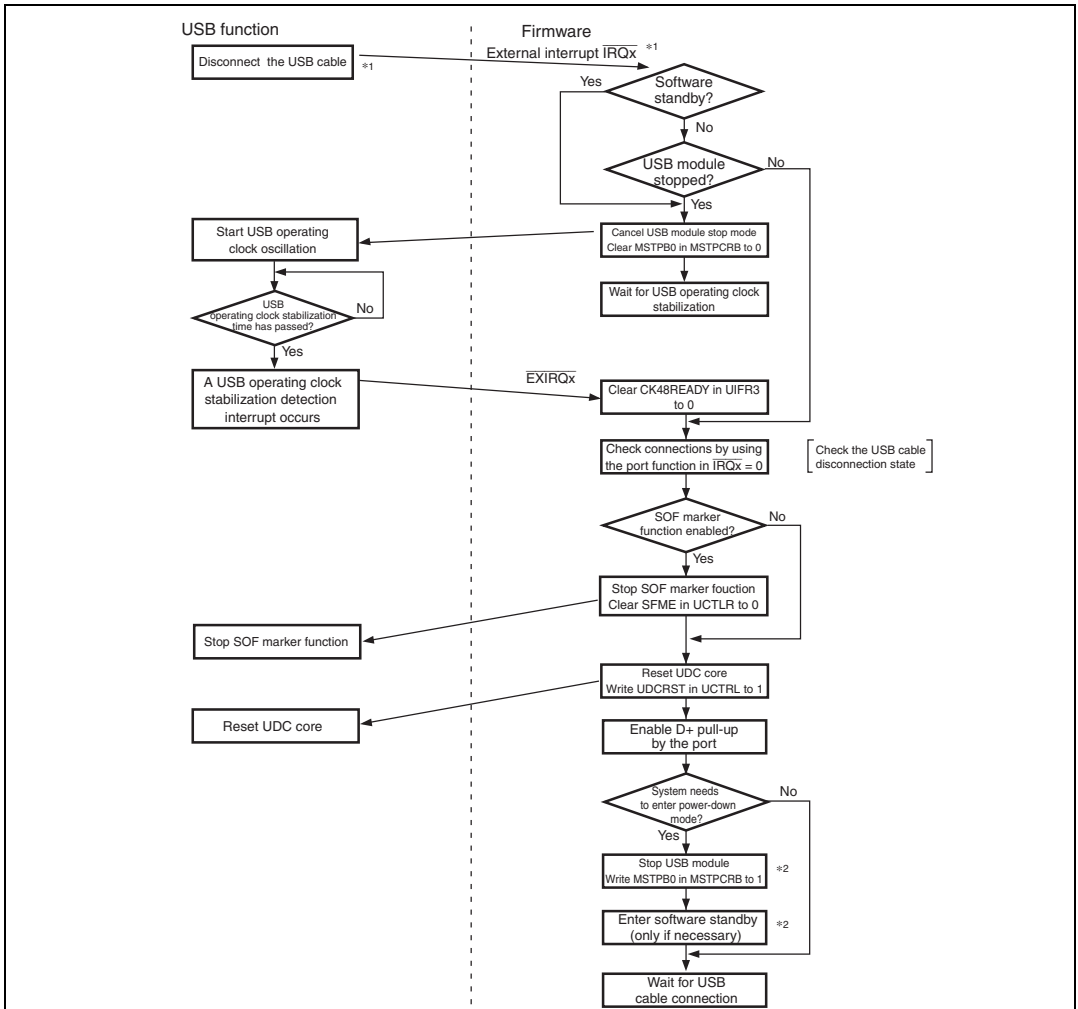


Figure 15.6 USB Cable Disconnection (When USB Module Stop or Software Standby Is Not Used)

(4) USB Cable Disconnection (When USB Module Stop or Software Standby Is Used)

If the USB cable enters the disconnection state from the connection state in an application (self powered) where USB module stop or software standby mode is used, perform the operation shown in figure 15.7.



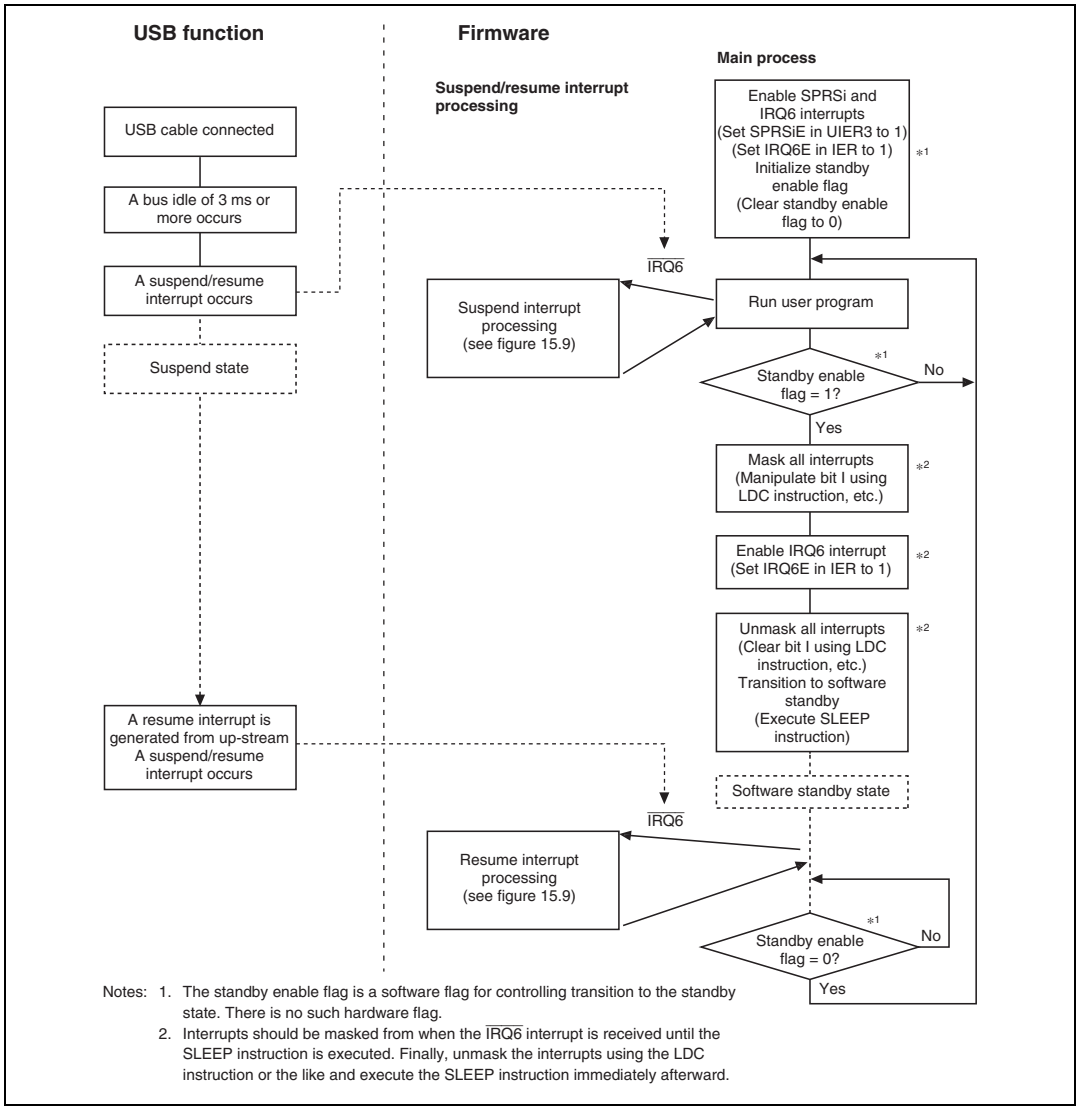
Notes: 1. A VBUS interrupt in the USB module cannot be detected in the software standby state or in the USB module stop state. Accordingly, in an application in which software standby or USB module stop is used in self-powered mode, a VBUS interrupt of the USB must be detected via the external interrupt pin IRQx. In this case, the IRQx pin must be specified as both edge sensitive. When IRQx is used, a VBUS interrupt in the USB module need not to be used.
2. Before entering the software standby state, USB module operation must be stopped by setting the MSTPB0 bit in MSTPCRB register to 1.

Figure 15.7 USB Cable Disconnection (When USB Module Stop or Software Standby Is Used)

15.5.3 Suspend and Resume Operations

(1) Suspend and Resume Operations

Figures 15.8 and 15.9 are flowcharts of the suspend and resume operations. If the USB bus enters the suspend state from a non-suspend state, or if it enters a non-suspend state from the suspend state due to a resume signal from up-stream, perform the operations shown below.

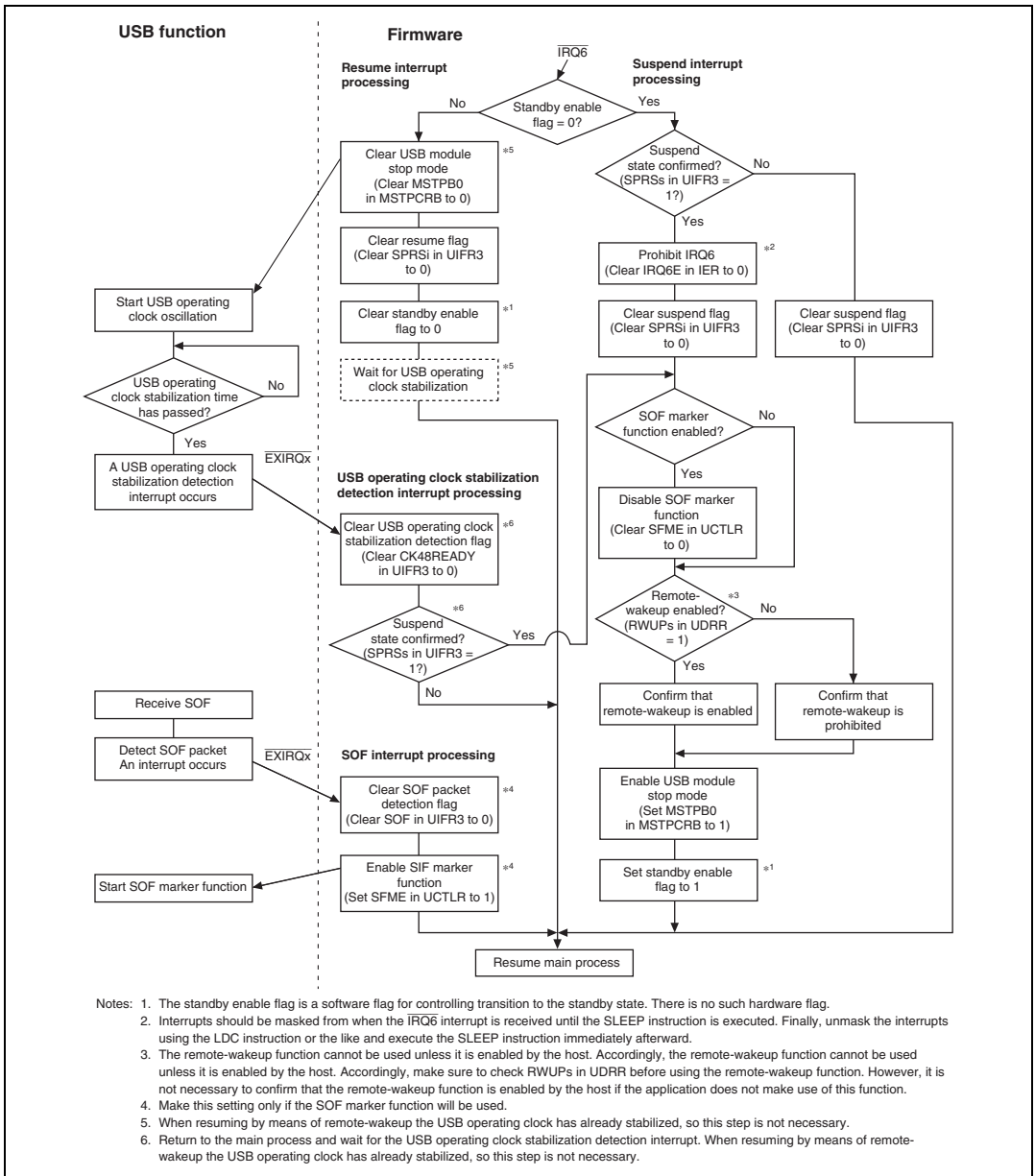


Notes: 1. The standby enable flag is a software flag for controlling transition to the standby state. There is no such hardware flag.
2. Interrupts should be masked from when the $\overline{\text{IRQ6}}$ interrupt is received until the SLEEP instruction is executed. Finally, unmask the interrupts using the LDC instruction or the like and execute the SLEEP instruction immediately afterward.

Figure 15.8 Example Flowchart of Suspend and Resume Operations

(2) Suspend and Resume Interrupt Processing

Figure 15.9 is a flowchart of suspend and resume interrupt processing.

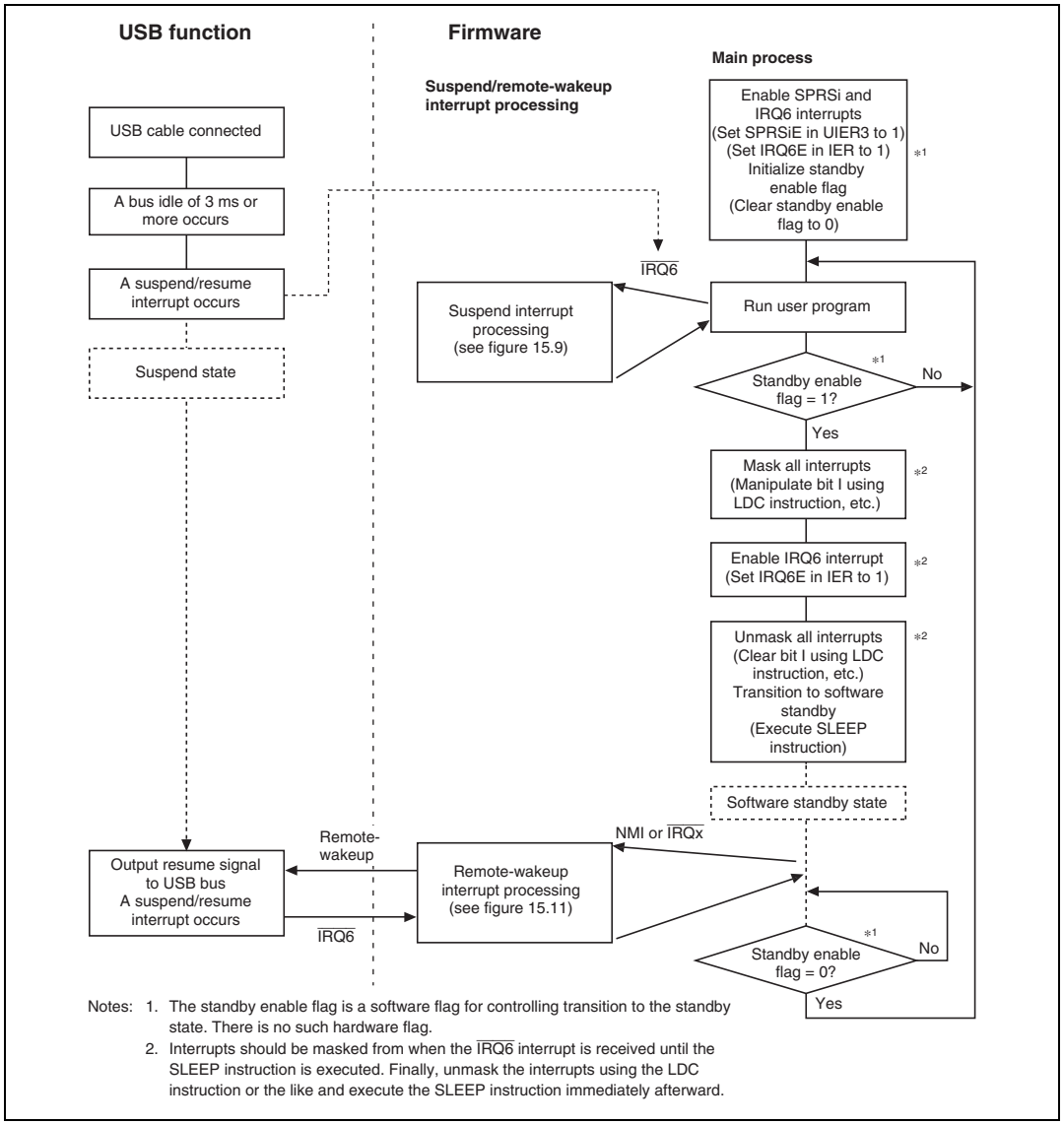


Notes: 1. The standby enable flag is a software flag for controlling transition to the standby state. There is no such hardware flag.
2. Interrupts should be masked from when the IRQ6 interrupt is received until the SLEEP instruction is executed. Finally, unmask the interrupts using the LDC instruction or the like and execute the SLEEP instruction immediately afterward.
3. The remote-wakeup function cannot be used unless it is enabled by the host. Accordingly, the remote-wakeup function cannot be used unless it is enabled by the host. Accordingly, make sure to check RWUPs in UDRR before using the remote-wakeup function. However, it is not necessary to confirm that the remote-wakeup function is enabled by the host if the application does not make use of this function.
4. Make this setting only if the SOF marker function will be used.
5. When resuming by means of remote-wakeup the USB operating clock has already stabilized, so this step is not necessary.
6. Return to the main process and wait for the USB operating clock stabilization detection interrupt. When resuming by means of remote-wakeup the USB operating clock has already stabilized, so this step is not necessary.

Figure 15.9 Example Flowchart of Suspend and Resume Interrupt Processing

(3) Suspend and Remote-Wakeup Operations

Figures 15.10 and 15.11 are flowcharts of the suspend and remote-wakeup operations. If the USB bus enters a non-suspend state from the suspend state due to a remote-wakeup signal from this function, perform the operations shown below.



Notes: 1. The standby enable flag is a software flag for controlling transition to the standby state. There is no such hardware flag.
2. Interrupts should be masked from when the IRQ6 interrupt is received until the SLEEP instruction is executed. Finally, unmask the interrupts using the LDC instruction or the like and execute the SLEEP instruction immediately afterward.

Figure 15.10 Example Flowchart of Suspend and Remote-Wakeup Operations

(4) Remote-Wakeup Interrupt Processing

Figure 15.11 is a flowchart of remote-wakeup interrupt processing.

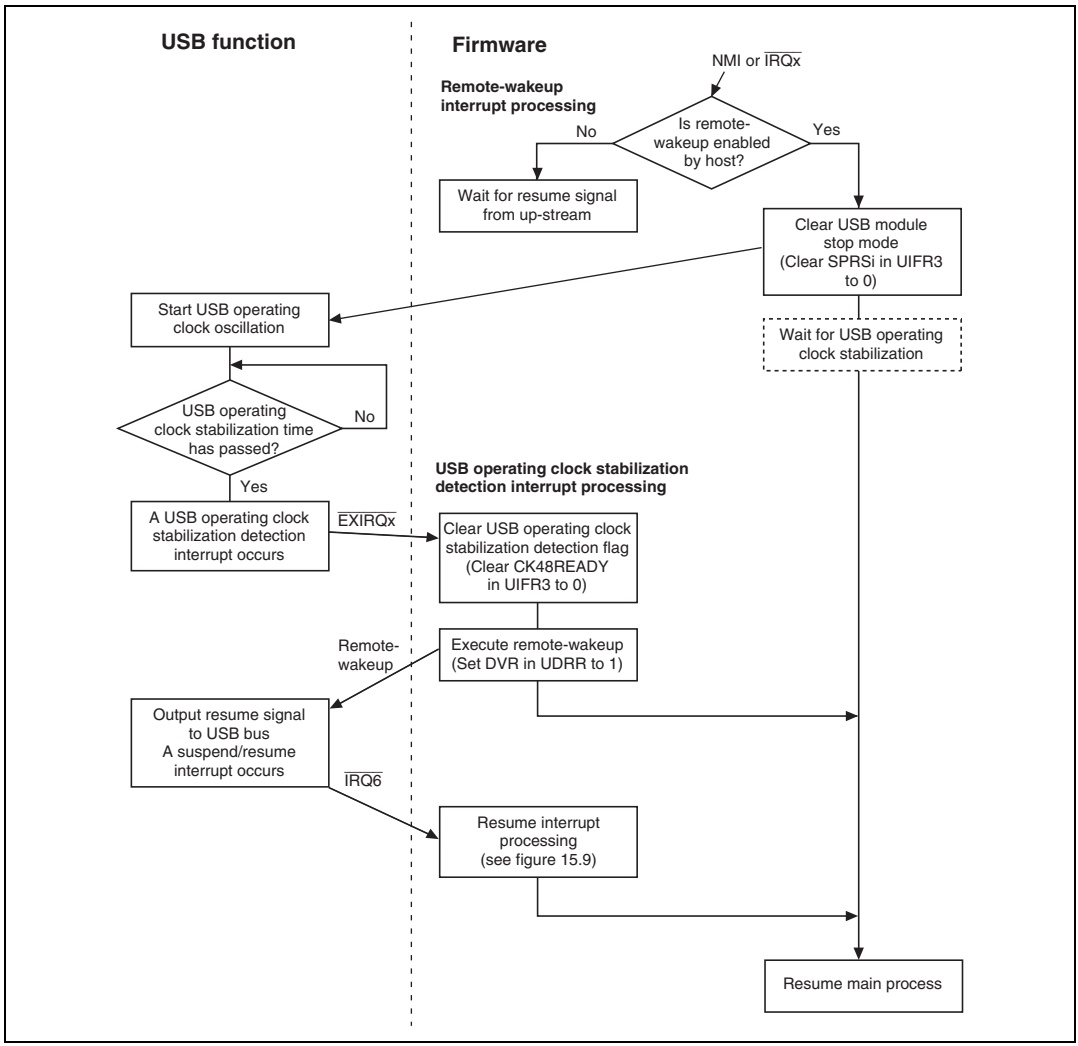


Figure 15.11 Example Flowchart of Remote-Wakeup Interrupt Processing

15.5.4 Control Transfer

The control transfer consists of three stages; setup, data (sometimes omitted), and status, as shown in figure 15.12. The data stage consists of multiple bus transactions. Figures 15.13 to 15.17 show operation flows in each stage.

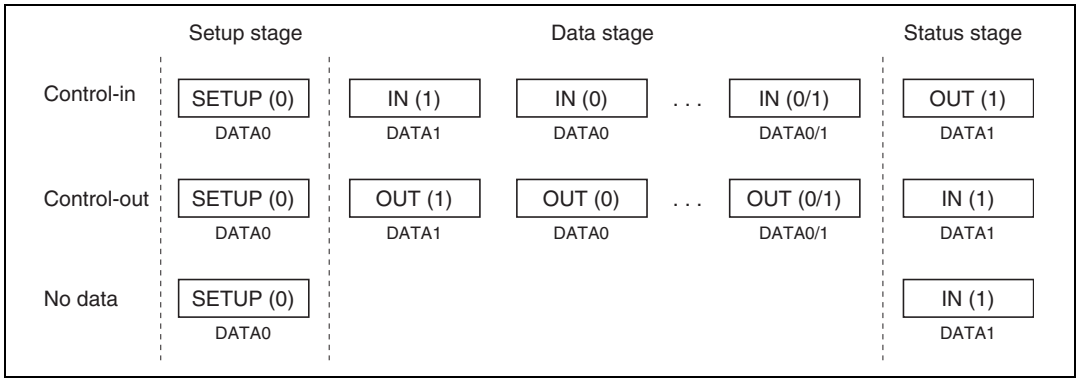
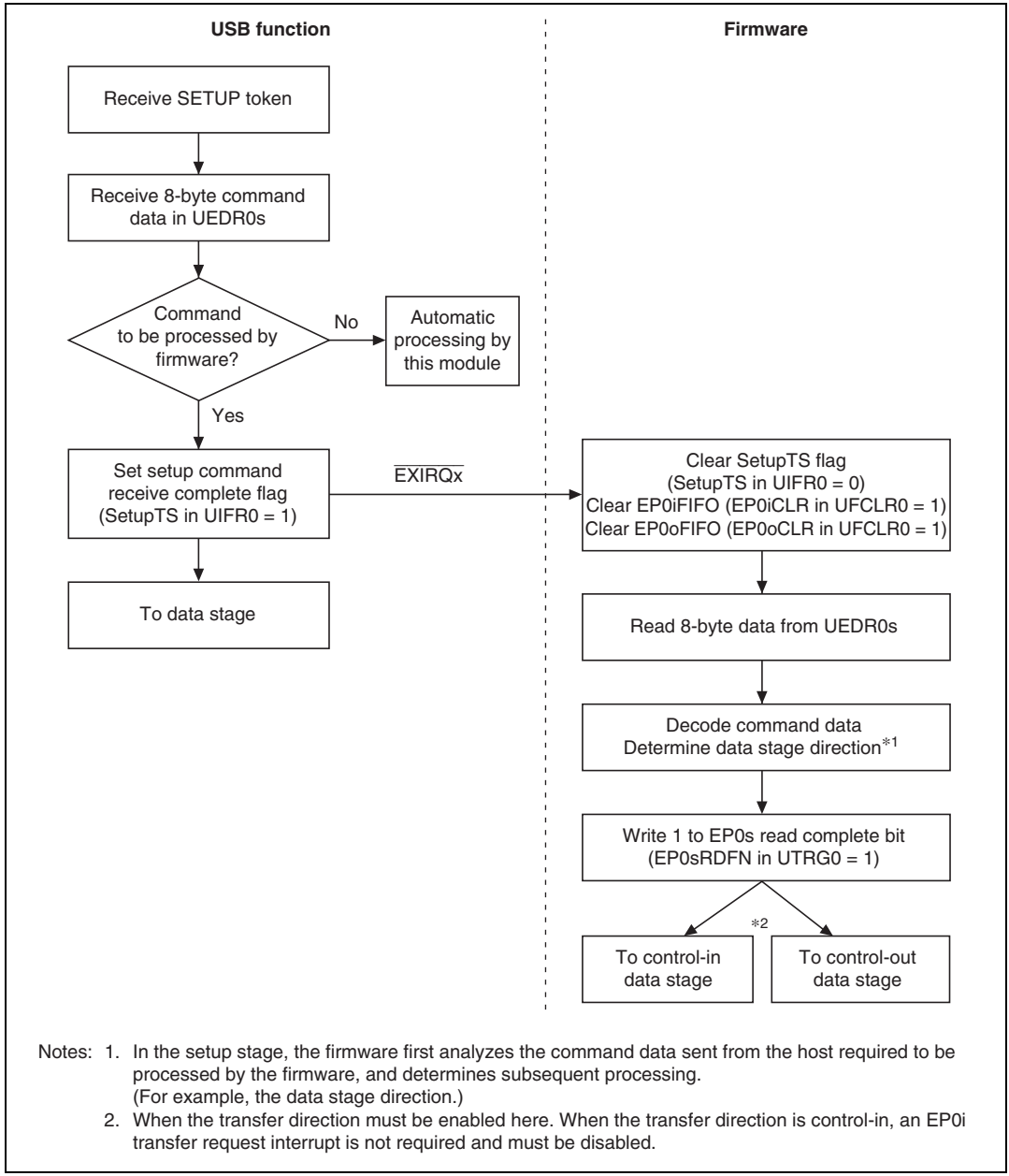


Figure 15.12 Control Transfer Stage Configuration

(1) Setup Stage



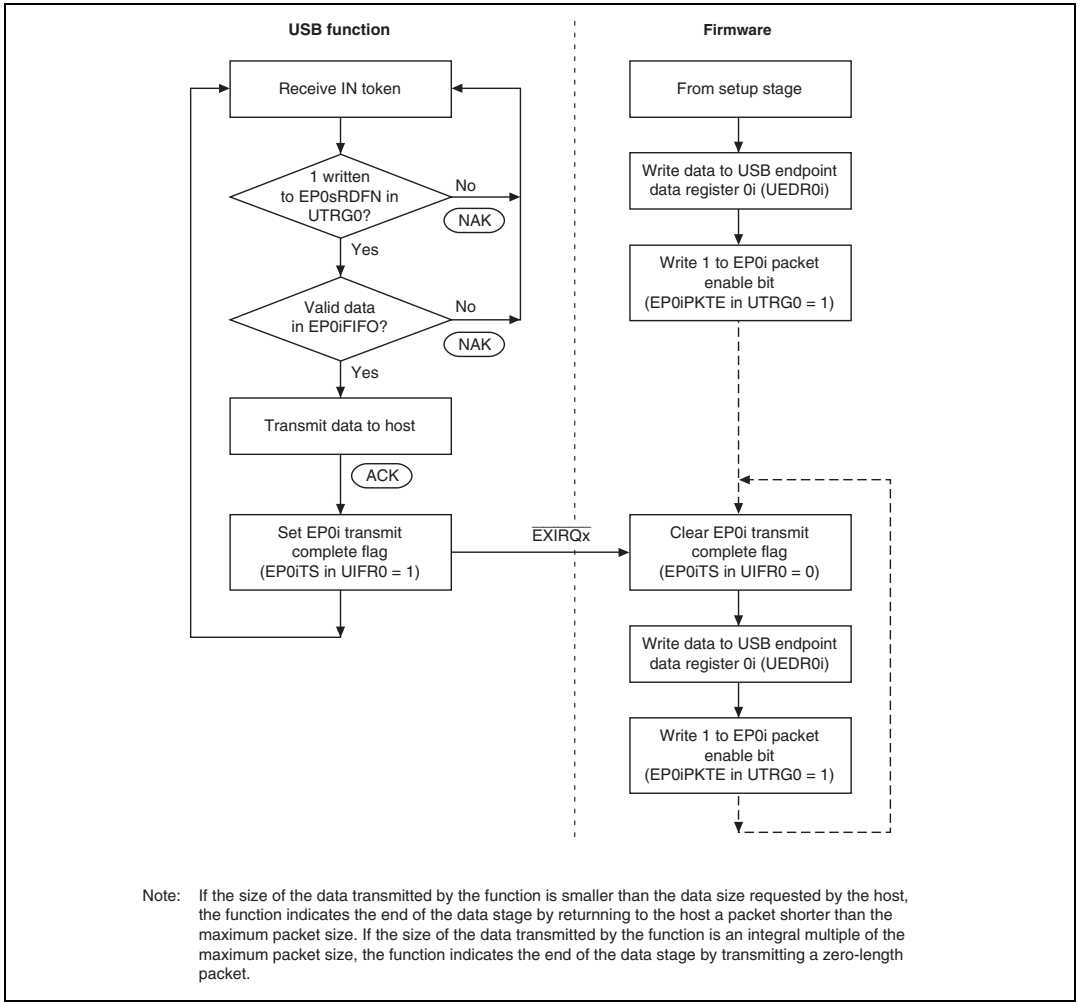
Notes: 1. In the setup stage, the firmware first analyzes the command data sent from the host required to be processed by the firmware, and determines subsequent processing. (For example, the data stage direction.)
2. When the transfer direction must be enabled here. When the transfer direction is control-in, an EP0i transfer request interrupt is not required and must be disabled.

Figure 15.13 Setup Stage Operation

(2) Data Stage (Control-In)

The firmware first analyzes command data from the host in the setup stage, and determines the subsequent data stage direction. If the result of command data analysis is that the data stage is in-transfer, one packet of data to be sent to the host is written to the FIFO. If there is more data to be sent, this data is written to the FIFO after the data written first has been sent to the host (EP0iTS of UIFR0 is set to 1).

The end of the data stage is identified when the host transmits an OUT token and the status stage is entered.



Note: If the size of the data transmitted by the function is smaller than the data size requested by the host, the function indicates the end of the data stage by returnning to the host a packet shorter than the maximum packet size. If the size of the data transmitted by the function is an integral multiple of the maximum packet size, the function indicates the end of the data stage by transmitting a zero-length packet.

Figure 15.14 Data Stage Operation (Control-In)

(3) Data Stage (Control-Out)

The firmware first analyzes command data from the host in the setup stage, and determines the subsequent data stage direction. If the result of command data analysis is that the data stage is out-transfer, the application waits for data from the host, and after data is received (EP0oTS of UIFR0 is set to 1), reads data from the FIFO. Next, the firmware writes 1 to the EP0o read complete bit, empties the receive FIFO, and waits for reception of the next data.

The end of the data stage is identified when the host transmits an IN token and the status stage is entered.

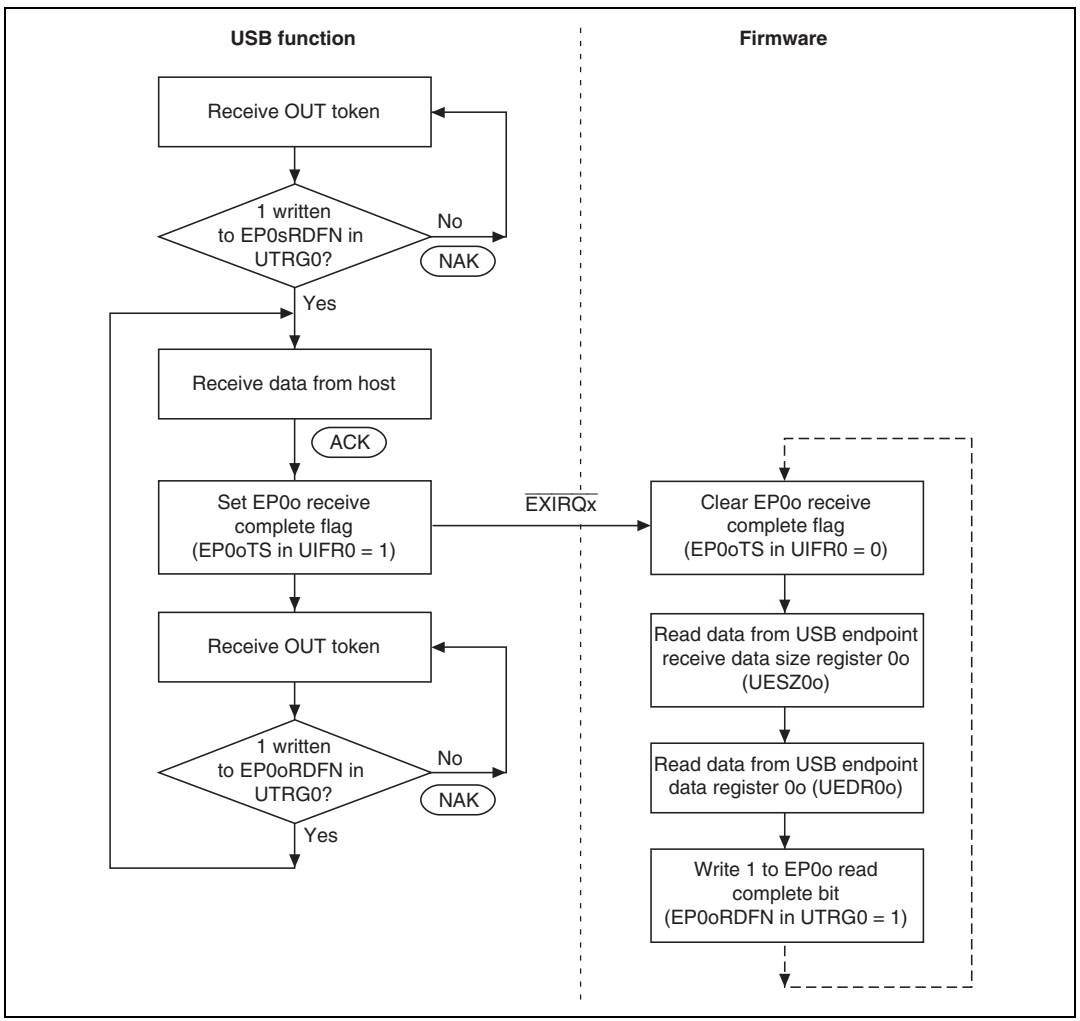


Figure 15.15 Data Stage Operation (Control-Out)

(4) Status Stage (Control-In)

The control-in status stage starts with an OUT token from the host. The firmware receives 0-byte data from the host, and ends control transfer.

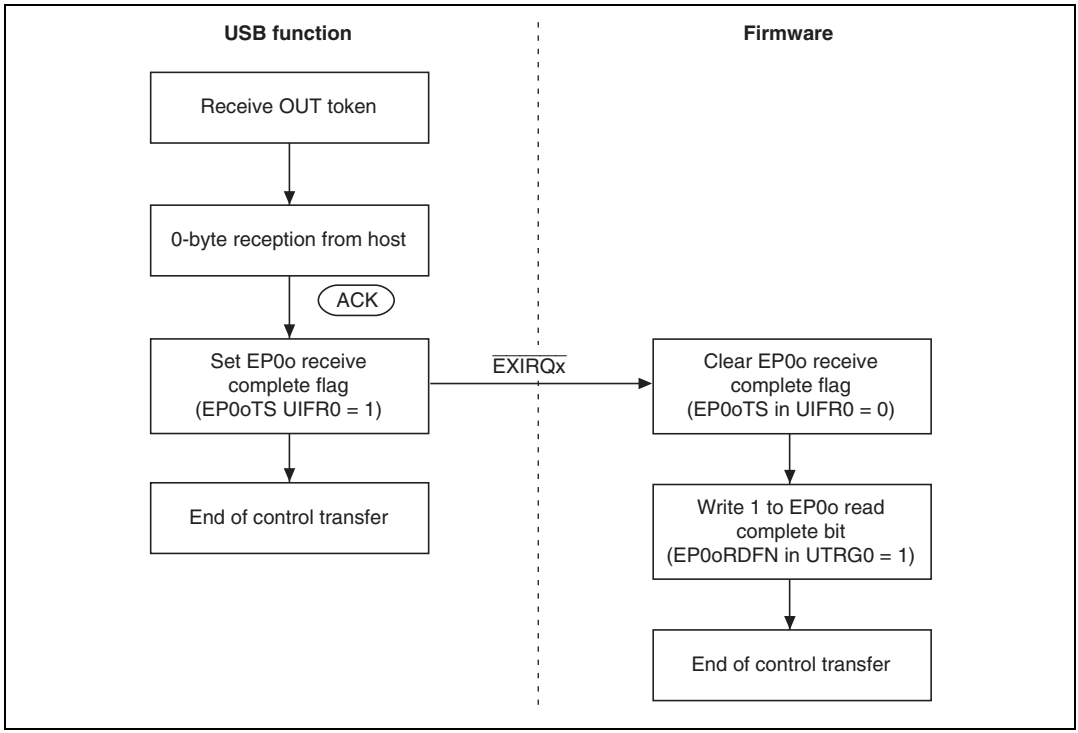


Figure 15.16 Status Stage Operation (Control-In)

(5) Status Stage (Control-Out)

The control-out status stage starts with an IN token from the host. When an IN-token is received at the start of the status stage, there is not yet any data in the EP0iFIFO, and so an EP0i transfer request interrupt is generated. The application recognizes from this interrupt that the status stage has started. Next, in order to transmit 0-byte data to the host, 1 is written to the EP0i packet enable bit but no data is written to the EP0iFIFO. As a result, the next IN token causes 0-byte data to be transmitted to the host, and control transfer ends.

After the application has finished all processing relating to the data stage, 1 should be written to the EP0i packet enable bit.

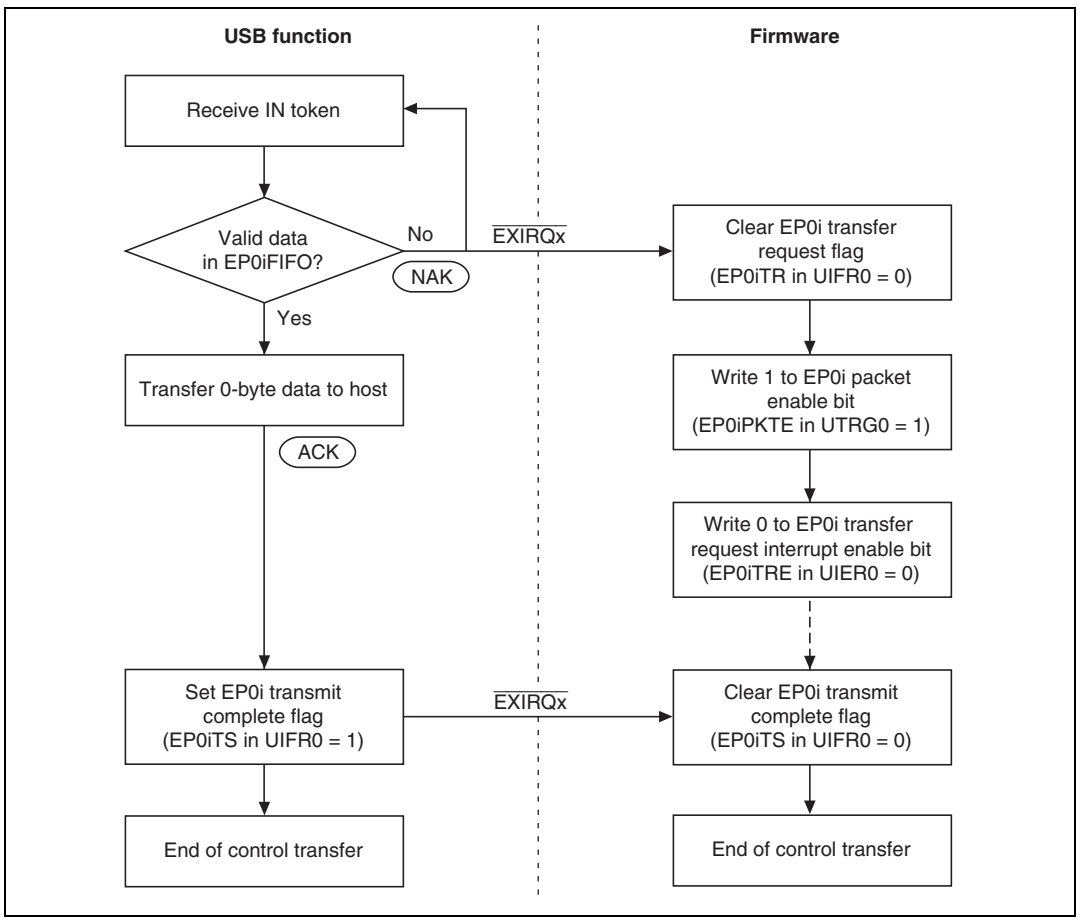
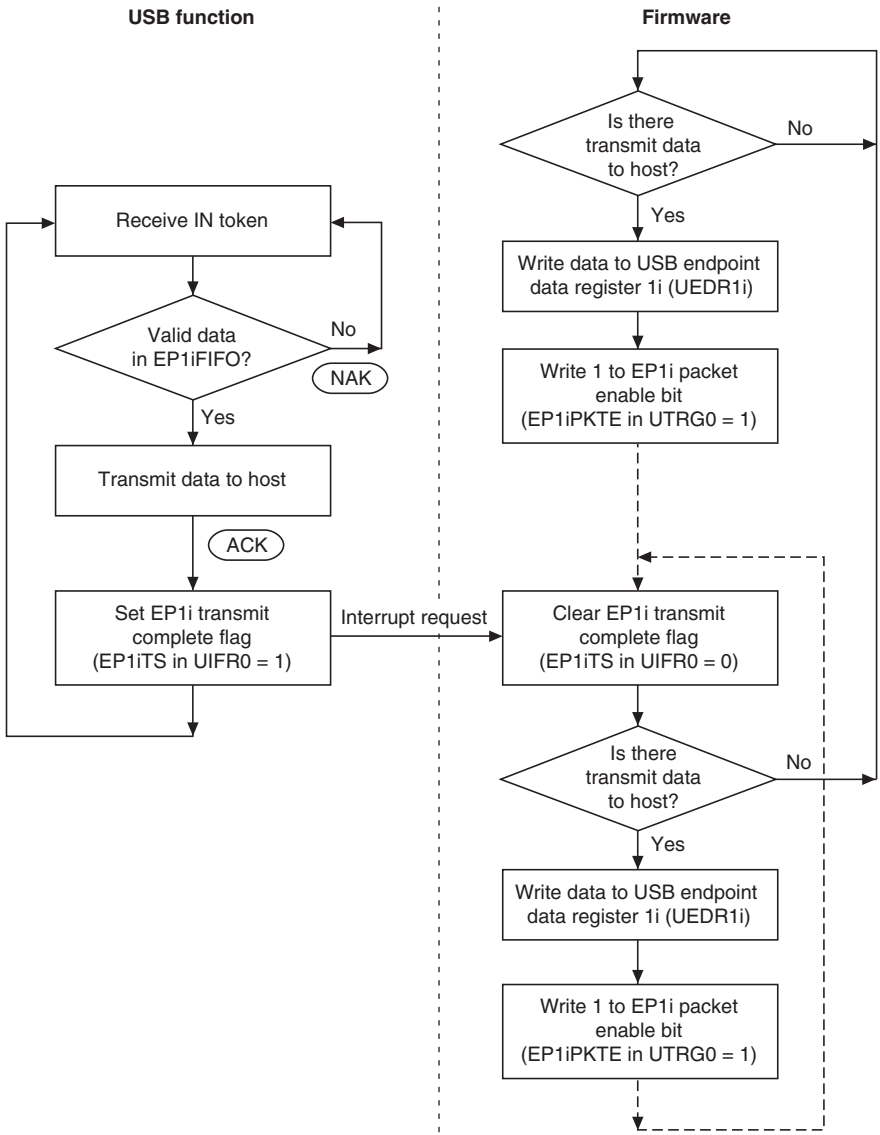


Figure 15.17 Status Stage Operation (Control-Out)

15.5.5 Interrupt-In Transfer (EP1i Is specified as Endpoint)



Note: This flowchart shows just one example of interrupt transfer processing. Other possibilities include an operation flow in which, if there is data to be transferred, the EP1i data enable bit in the USB data status register is referenced to confirm that the FIFO is empty, and then data is written to the FIFO.

Figure 15.18 EP1i Interrupt-In Transfer Operation

15.5.6 Bulk-In Transfer (Dual FIFOs) (EP2i Is specified as Endpoint)

EP2i has two 64-byte FIFOs, but the user can perform data transmission and transmit data writes without being aware of this dual-FIFO configuration. However, one data write is performed for one FIFO. For example, even if both FIFOs are empty, it is not possible to perform EP2iPKTE at one time after consecutively writing 128 bytes of data. EP2iPKTE must be performed for each 64-byte write.

When transmitting data to the host using a bulk-in transfer, the EP2iFIFO empty interrupt must first be enabled. 1 is written to the UIER1/EP2iEMPTYE bit, and the EP2iFIFO empty interrupt is enabled. At first, both EP2iFIFOs are empty, and so an EP2iFIFO empty interrupt is generated immediately.

The data to be transmitted is written to the data register using this interrupt. After the first transmit data write for one FIFO, the other FIFO is empty, and so the next transmit data can be written to the other FIFO immediately. When both FIFOs are full, EP2iEMPTY is cleared to 0. If at least one FIFO is empty, UIFR1/EP2iEMPTY is set to 1. When ACK is returned from the host after data transmission is completed, the FIFO used in the data transmission becomes empty. If the other FIFO contains valid transmit data at this time, transmission can be continued.

When transmission of all data has been completed, write 0 to UIER1/EP2iEMPTYE and disable $\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$ interrupt requests.

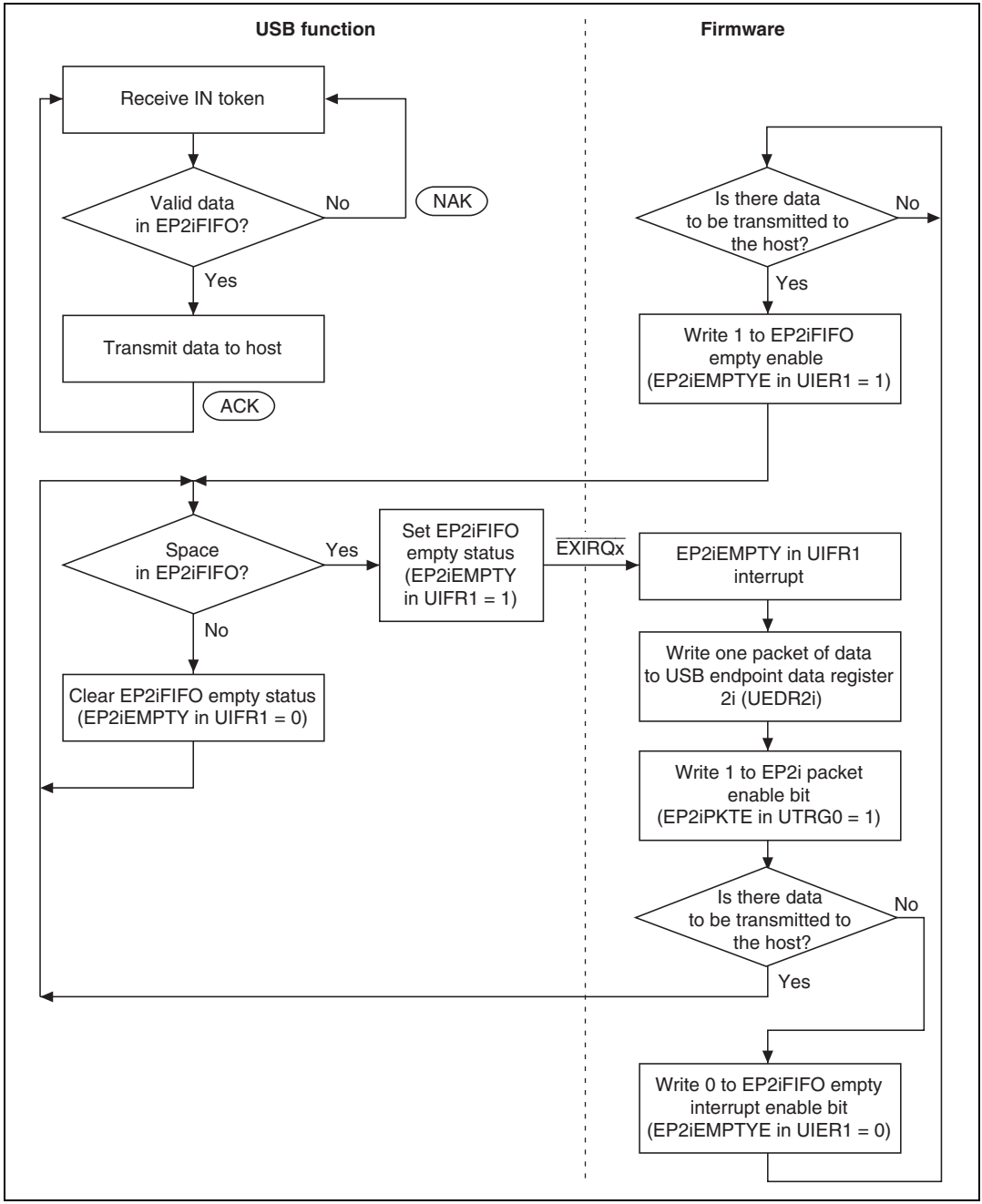


Figure 15.19 EP2i Bulk-In Transfer Operation

15.5.7 Bulk-Out Transfer (Dual FIFOs) (EP2o Is specified as Endpoint)

EP2o has two 64-byte FIFOs, but the user can perform data reception and receive data reads without being aware of this dual-FIFO configuration.
When one FIFO is full after reception is completed, the UIFR1/EP2oREADY bit is set. After the first receive operation into one of the FIFOs when both FIFOs are empty, the other FIFO is empty, and so the next packet can be received immediately. When both FIFOs are full, NAK is returned to the host automatically. When reading of the receive data is completed following data reception, 1 is written to the UTRG0/EP2oRDFN bit. This operation empties the FIFO that has just been read, and makes it ready to receive the next packet.

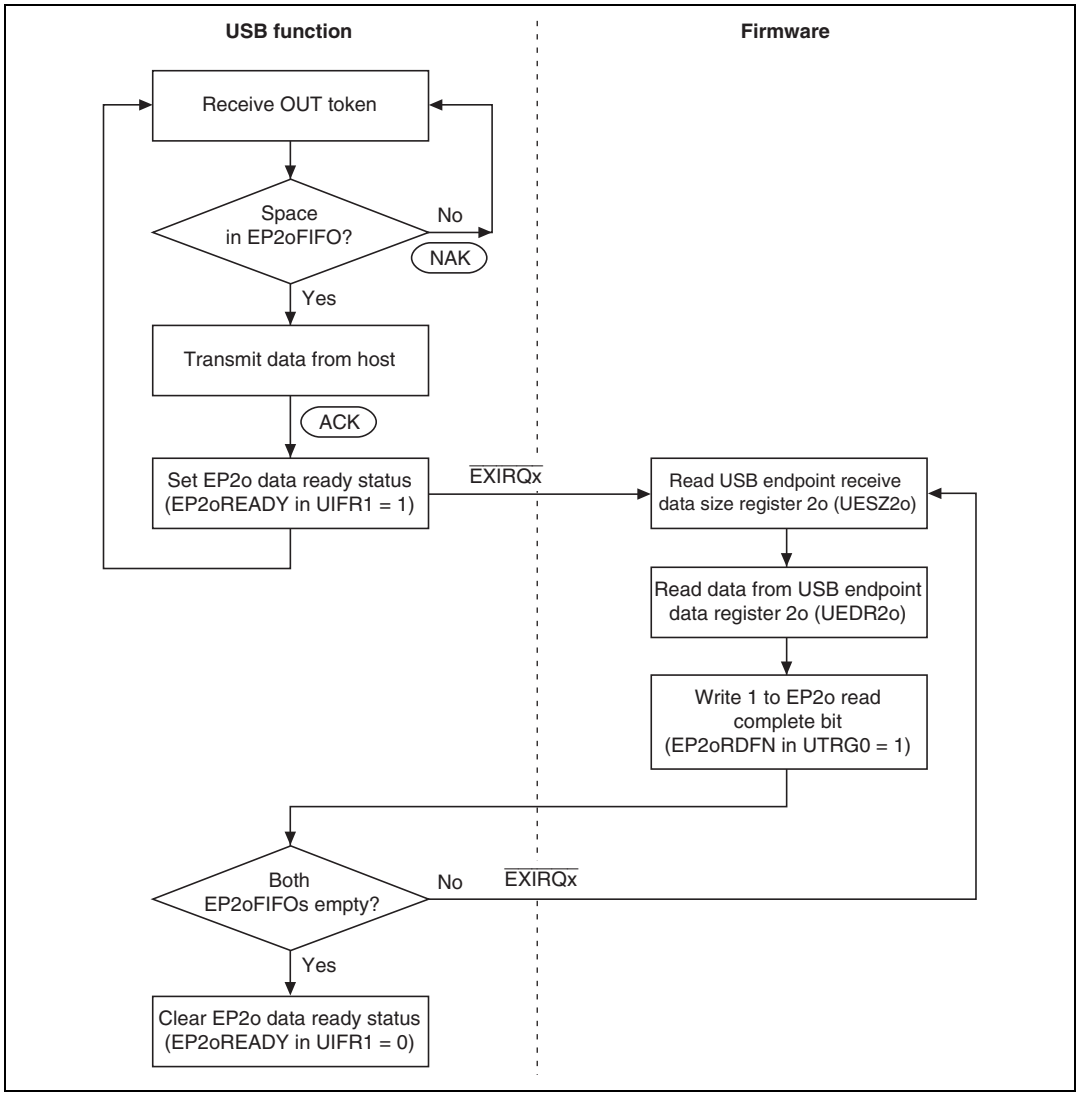


Figure 15.20 EP2o Bulk-Out Transfer Operation

15.5.8 Isochronous–In Transfer (Dual-FIFO) (When EP3i Is Specified as Endpoint)

EP3i has two 128-byte (maximum) FIFOs, however the user can perform data transmission and transmit data writes without being aware of this dual-FIFO configuration.

In isochronous transfer, as a transmission is performed once a frame (1 ms), the hardware automatically switches FIFOs when the hardware receives the SOF. Even when SOF cannot be received by an error, enabling the SOF marker function allows the hardware to automatically switch the FIFOs every 1 ms. In addition, the USB function checks if the valid data of the previous frame was transferred from the FIFO to the host after SOF has been received. As a result, if the valid data in the FIFO is not transferred to the host (if the host does not return an IN token or if an IN token error has occurred), the USB regards it as EP3i IN token not received and sets the EP3iTF bit of UIFR1 to 1.

Two FIFOs are switched when the SOF is received, the FIFO used to transfer data to the host differs from the FIFO to which the firmware writes transmit data. Accordingly, no contention occurs between one FIFO read and the other FIFO write. The data to be written by the firmware is transferred to the host in the next frame. As two FIFOs are automatically switched when the SOF is received, data must be written within a single frame.

The USB function transfers data to the host if the FIFO contains data to be sent to the host after an IN token has been received. If the FIFO contains no data, the USB function sets the TR flag to 1 and sends 0-byte data to the host.

The firmware first calls the isochronous transfer process routine by the SOF interrupt and checks the time stamp. The firmware then writes 1-packet data to the FIFO and this 1-packed data is sent to the host in the next frame.

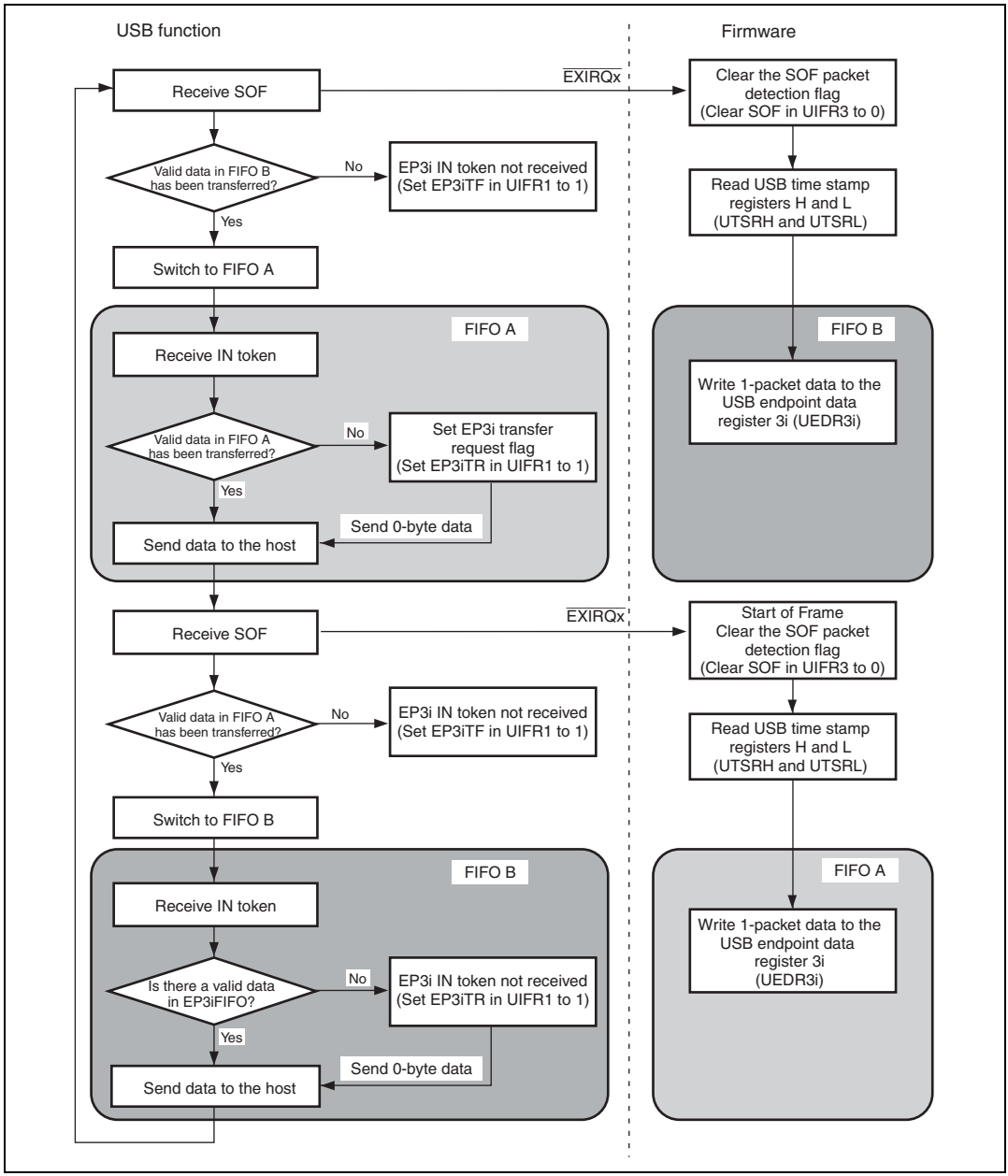


Figure 15.21 EP3i Isochronous-In Transfer Operation

15.5.9 Isochronous–Out Transfer (Dual-FIFO) (When EP3o Is Specified as Endpoint)

EP3o has two 128-byte (maximum) FIFOs, however the user can perform data transmission and transmit data writes without being aware of this dual-FIFO configuration.

In isochronous transfer, as a transmission is performed once a frame (1 ms), the hardware automatically switches FIFOs when the hardware receives the SOF. (Even when SOF cannot be received by an error, enabling the SOF marker function allows the hardware to automatically switch the FIFOs every 1 ms.)

Two FIFOs are switched when the SOF is received, the FIFO used to transfer data from the host to the firmware differs from the FIFO from which the firmware reads transmit data. Accordingly, no contention occurs between one FIFO read and the other FIFO write. The firmware read the data in the previous frame. As two FIFOs are automatically switched when the SOF is received, data must be read within a single frame.

The USB function receives data from the host after an OUT token has been received. If a data error occurs on data reception, the USB function sets the TF flag to 1; if no data error occurs, the USB function sets the TS flag to 1.

The firmware first calls the isochronous transfer process routine via the SOF interrupt, checks the time stamp, and then reads from the FIFO. Accordingly, the firmware checks whether a data error occurs or not via status information indicated by the TF and TS flags. These TF and TS flags indicate the status of the FIFO currently being read.

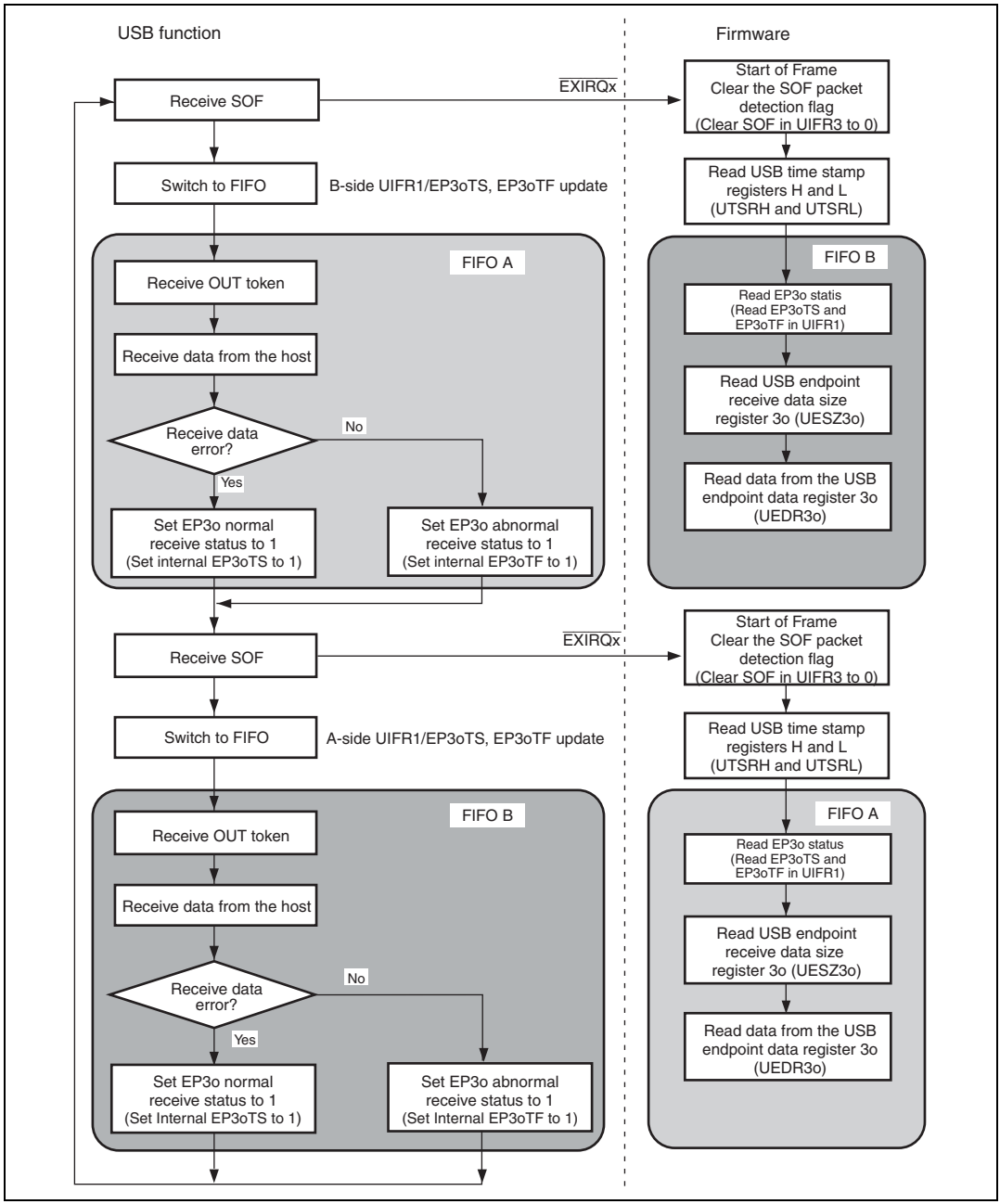


Figure 15.22 EP3o Isochronous-Out Transfer Operation

15.5.10 Processing of USB Standard Commands and Class/Vendor Commands

(1) Processing of Commands Transmitted by Control Transfer

A command transmitted from the host by control transfer may require decoding and execution of command processing by the firmware. Whether or not command decoding is required by the firmware is indicated in table 15.6 below.

Table 15.6 Command Decoding on Firmware

Decoding not Necessary on Firmware	Decoding Necessary on Firmware
Clear Feature	Get Descriptor
Get Configuration	Synch Frame
Get Interface	Set Descriptor
Get Status	Class/Vendor command
Set Address	
Set Configuration	
Set Feature	
Set Interface	

If decoding is not necessary on the firmware, command decoding and data stage and status stage processing are performed automatically. No processing is necessary by the user. An interrupt is not generated in this case.

If decoding is necessary on the firmware, the USB function module stores the command in the EP0sFIFO. After normal reception is completed, the SetupTS flag of UIER0 is set and an interrupt request is generated from the $\overline{\text{EXIRQx}}$. In the interrupt routine, eight bytes of data must be read from the EP0s data register (UEDR0s) and decoded by firmware. The necessary data stage and status stage processing should then be carried out according to the result of the decoding operation.

15.5.11 Stall Operations

(1) Overview

This section describes stall operations in the USB function module. There are two cases in which the USB function module stall function is used:

1. When the firmware forcibly stalls an endpoint for some reason
2. When a stall is performed automatically within the USB function module due to a USB specification violation

The USB function module has internal status bits that hold the status (stall or non-stall) of each endpoint. When a transaction is sent from the host, the module references these internal status bits and determines whether to return a stall to the host. These bits cannot be cleared by the application; they must be cleared with a Clear Feature command from the host.

(2) Forcible Stall by Firmware

The firmware uses UESTL to issue a stall request for the USB function module. When the firmware wishes to stall a specific endpoint, it sets the corresponding EPnSTL bit (1-1 in figure 15.23). The internal status bits are not changed.

When a transaction is sent from the host for the endpoint for which the EPnSTL bit was set, the USB function module references the internal status bit, and if this is not set, references the corresponding EPnSTL bit (1-2 in figure 15.23). If the corresponding EPnSTL bit is not set, the internal status bit is not changed and the transaction is accepted. If the corresponding EPnSTL bit is set, the USB function module sets the internal status bit and returns a stall handshake to the host (1-3 in figure 15.23).

Once an internal status bit is set, it remains set until cleared by a Clear Feature command from the host, without regard to EPnSTL. Even after a bit is cleared by the Clear Feature command (3-1 in figure 15.23), the USB function module continues to return a stall handshake while the EPnSTL bit is set, since the internal status bit is set each time a transaction is executed for the corresponding endpoint (1-2 in figure 15.23). To clear a stall, therefore, it is necessary for the corresponding EPnSTL bit to be cleared by the firmware, and also for the internal status bit to be cleared with a Clear Feature command (2-1 to 2-3 in figure 15.23).

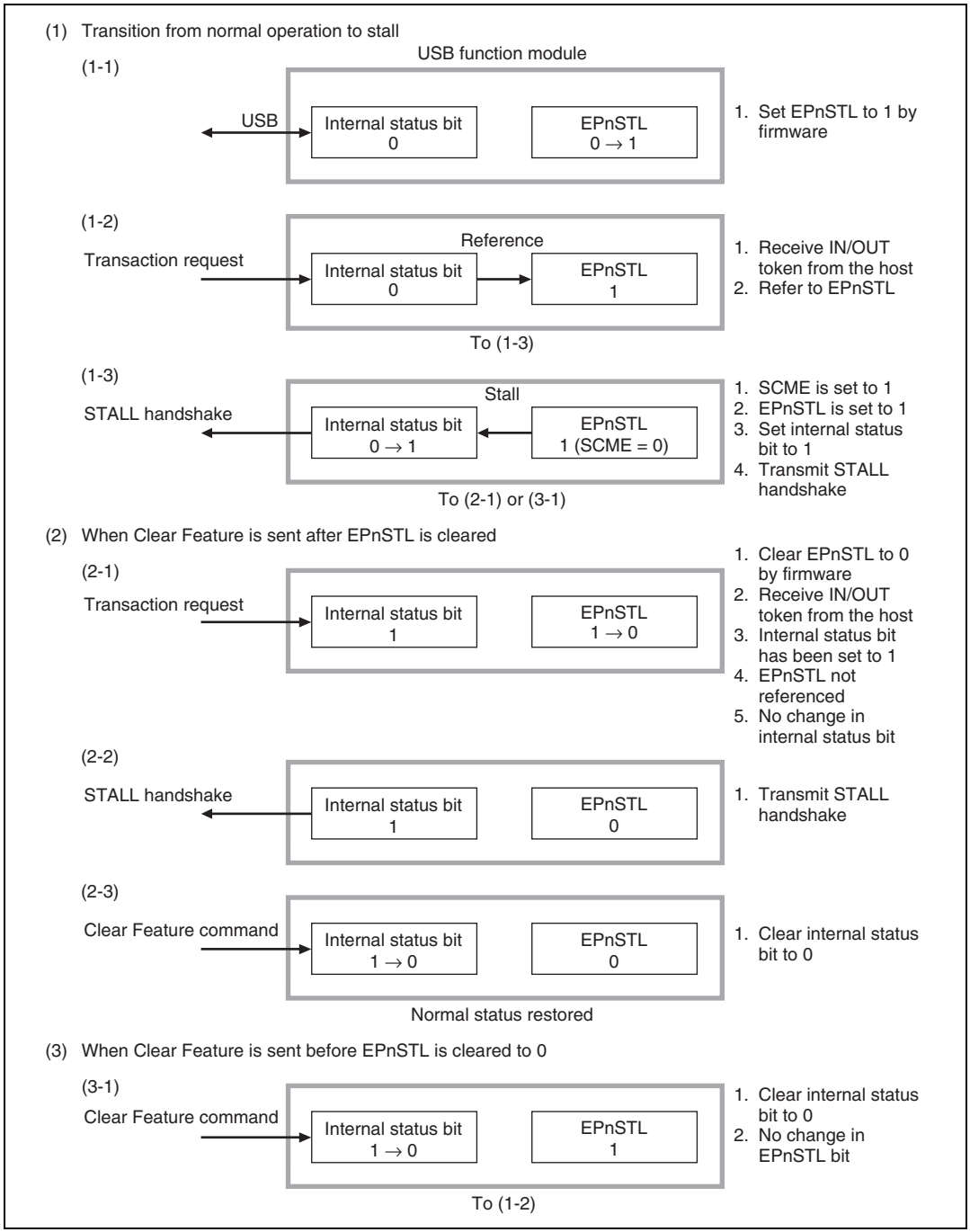


Figure 15.23 Forcible Stall by Firmware

(3) Automatic Stall by USB Function Module

When a stall setting is made with the Set Feature command, when the information of this module differs from that returned to the host by the Get Descriptor, or in the event of a USB specification violation, the USB function module automatically sets the internal status bit for the relevant endpoint without regard to EPnSTL, and returns a stall handshake (1-1 in figure 15.24).

Once an internal status bit is set, it remains set until cleared by a Clear Feature command from the host, without regard to EPnSTL. After a bit is cleared by the Clear Feature command, EPnSTL is referenced (3-1 in figure 15.24). The USB function module continues to return a stall handshake while the internal status bit is set, since the internal status bit is set even if a transaction is executed for the corresponding endpoint (2-1 and 2-2 in figure 15.24). To clear a stall, therefore, the internal status bit must be cleared with a Clear Feature command (3-1 in figure 15.24). If set by the firmware, EPnSTL should also be cleared (2-1 in figure 15.24).

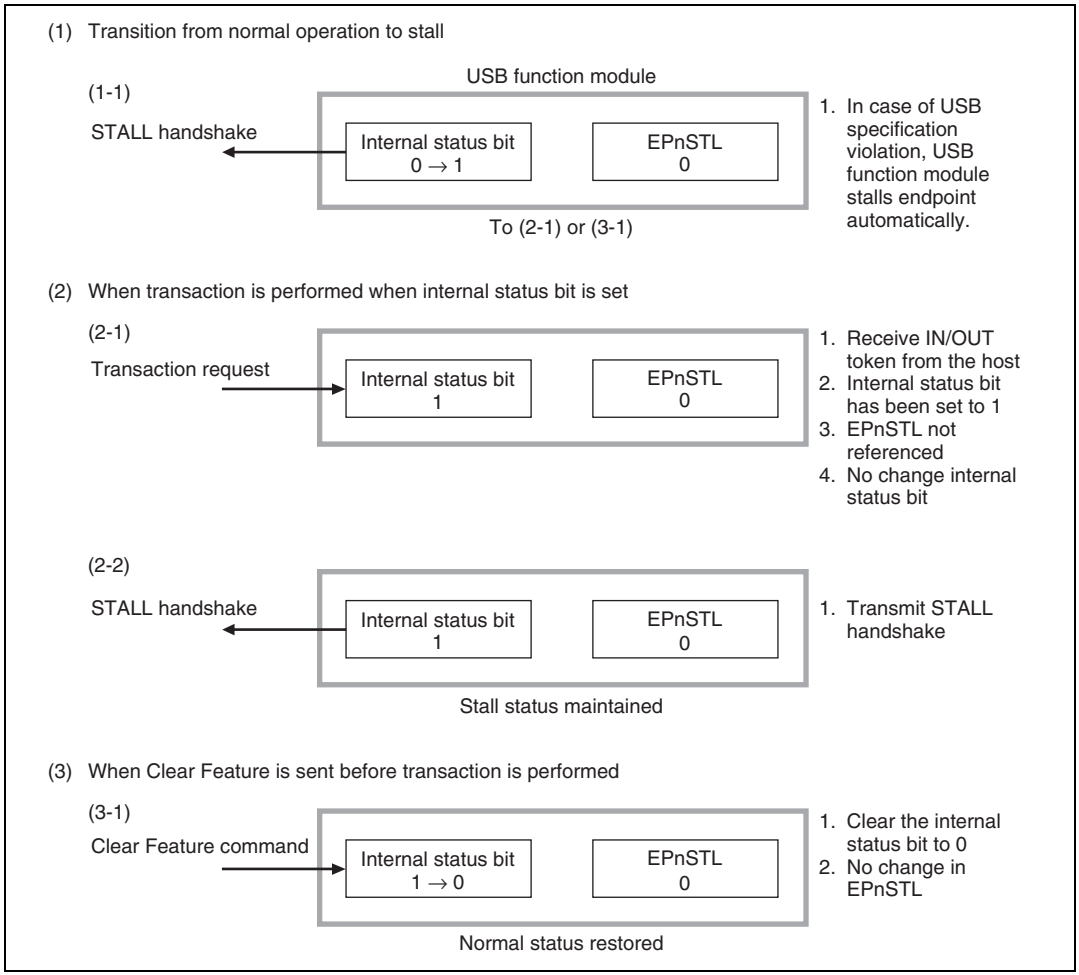


Figure 15.24 Automatic Stall by USB Function Module

15.6 DMA Transfer Specifications

Two methods of USB request and auto request are available for the DMA transfer of USB data.

15.6.1 DMA Transfer by USB Request

(1) Overview

Only normal mode in full address mode (cycle steal mode) supports the transfer by a USB request of the on-chip DMAC. Endpoints that can be transferred by the on-chip DMAC are EP2 and EP4 in Bulk transfer (corresponding registers are UEDR2i, UEDR2o, UEDR4i, and UEDR4o). In DMA transfer, the USB module must be accessed as an external device in area 6. The USB module cannot be accessed as a device with external ACK (single-address transfer cannot be performed). 0-byte data transfer to EP2o or EP4o is ignored even if the DMA transfer is enabled by setting the EP2oT1 or EP4oT1 bit of UDMAR to 1.

(2) On-Chip DMAC Settings

The on-chip DMAC must be specified as follows: A USB request ($\overline{\text{DREQ}}$ signal), activated by low-level input, byte size, full-address mode transfer, and the DTA bit of DMABCR = 1. After completing the DMA transfers of specified time, the DMAC automatically stops. Note, however, that the USB module keeps the $\overline{\text{DREQ}}$ signal low while data to be transferred by the on-chip DMAC remains regardless of the DMAC status.

(3) EP2i and EP4i DMA Transfer

The EP2iT1 and EP4iT1 bits of UDMAR enable DMA transfer. The EP2iT0 and EP4iT0 bits of the UDMAR specify the $\overline{\text{DREQ}}$ signal to be used by the DMA transfer. When the EP2iT1 or EP4iT1 is set to 1, the $\overline{\text{DREQ}}$ signal is driven low if at least one of EP2i and EP4i data FIFOs are empty; the $\overline{\text{DREQ}}$ signal is driven high if both EP2i and EP4i data FIFOs are full.

(a) EP2iPKTE and EP4iPKTE Bits of UTRG

When DMA transfer is performed on EP2i and EP4i transmit data, the USB module automatically performs the same processing as writing 1 to EP2iPKTE and EP4iPKTE if one data FIFO (64 bytes) becomes full. Accordingly, to transfer data of integral multiples of 64 bytes, the user need not write EP2iPKTE and EP4iPKTE to 1. To transfer data of less than 64 bytes, the user must write EP2iPKTE and EP4iPKTE to 1 using the DMA transfer end interrupt of the on-chip DMAC. If the user writes 1 to EP2iPKTE and EP4iPKTE in cases other than the case when data of less than 64 bytes is transferred, excess transfer occurs and correct operation cannot be guaranteed.

Figure 15.25 shows an example for transmitting 150 bytes of data from EP2i to the host. In this case, internal processing the same as writing 1 to EP2iPKTE is automatically performed twice.

This kind of internal processing is performed when the currently selected data FIFO becomes full. Accordingly, this processing is automatically performed only when 64-byte data is sent. This processing is not performed automatically when data less than 64 bytes is sent.

(b) EP2i DMA Transfer Procedure

1. Set bits EP2iT1 and EP2iT0 in UDMAR.
2. DMAC settings (in DMAC specify number of transfers for 150 bytes).
3. Start DMAC.
4. DMA transfer.
5. Write 1 to EP2iPKTE in UTRG0 using DMA transfer end interrupt.

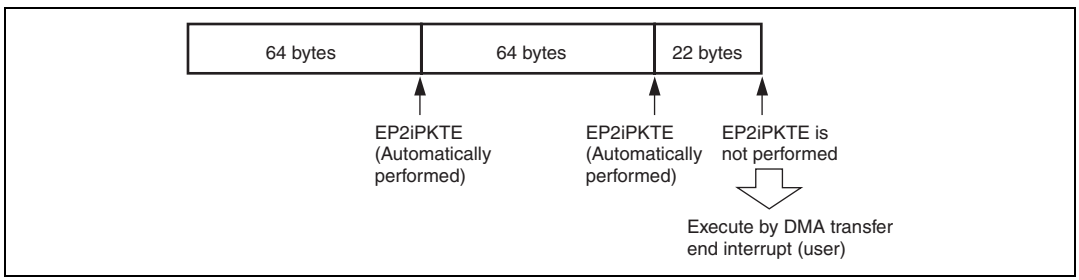


Figure 15.25 EP2iPKTE Operation in UTRG0

(4) EP2o and EP4o DMA Transfer

The EP2oT1 and EP4oT1 bits of UDMAR enable DMA transfer. The EP2oT0 and EP4oT0 bits of the UDMAR specify the $\overline{\text{DREQ}}$ signal to be used by the DMA transfer. When the EP2oT1 or EP4oT1 is set to 1, the $\overline{\text{DREQ}}$ signal is driven low if at least one of EP2o and EP4o data FIFOs are full (ready state); the $\overline{\text{DREQ}}$ signal is driven high if both EP2o and EP4o data FIFOs are empty when all receive data items are read.

(a) EP2oRDFN and EP4oRDFN Bits of UTRG

When DMA transfer is performed on EP2o and EP4o receive data, do not write 1 to EP2oRDFN or EP4oRDFN after one data FIFO (64 bytes) has been read. In data transfer other than DMA transfer, the next data cannot be read after one data FIFO (64 bytes) has been read unless EP2oRDFN and EP4oRDFN are set to 1. While in DMA transfer, the USB module automatically performs the same processing as writing 1 to EP2oRDFN and EP4oRDFN if the currently selected FIFO becomes empty. Accordingly, in DMA transfer, the user need not write EP2oRDFN and EP4oRDFN to 1. If the user writes EP2oRDFN and EP4oRDFN to 1 in DMA transfer, excess transfer occurs and correct operation cannot be guaranteed.

Figure 15.26 shows an example of EP2o receiving 150 bytes of data from the host. In this case, internal processing the same as writing 1 to EP2oRDFN is automatically performed three times. This kind of internal processing is performed when the currently selected data FIFO becomes

empty. Accordingly, this processing is automatically performed both when 64-byte data is sent and when data less than 64 bytes is sent.

(b) EP2o DMA Transfer Procedure

The DMAC transfer unit should be one packet. Therefore, after the EP2oREADY flag is set, check the size of the data received from the host and make DMAC settings to match the number of transfers required.

1. Set bits EP2oT1 and EP2oT0 in UDMAR.
2. Wait for EP2oREADY flag to be set.
3. DMAC settings.

   Read value of UESZ2o and specify number of transfers to match size of received data (64 bytes or less).
4. Start DMAC.
5. DMA transfer (transfer of 64 bytes or less).
6. Wait for end of DMA transfer.
7. Repeat steps 2 to 6 above.

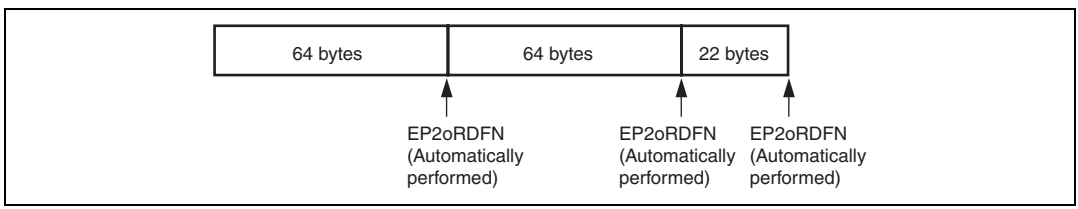


Figure 15.26 EP2oRDFN Operation in UTRG0

15.6.2 DMA Transfer by Auto-Request

(1) Overview

Burst mode transfer or ycle steal transfer can be selected for the on-chip DMAC auto-request transfer. Endpoints that can be transferred by the on-chip DMAC are all registers (UEDR0s, UEDR0i, UEDR0o, UEDR1i, UEDR2i, UEDR2o, UEDR3i, UEDR3o, UEDR4i, UEDR4o, and UEDR5i). Confirm flags and interrupts corresponding to each data register before activating the DMA. As UDMAR is not used in auto-request mode, set UDMAR to H'00.

(2) On-Chip DMAC Settings

The on-chip DMAC must be specified as follows: Auto-request, byte size, full-address mode transfer, and number of transfers equal to or less than the maximum packet size of the data register. After completing the DMAC transfers of specified time, the DMAC automatically stops.

(3) EPni DMA Transfer (n = 0 to 5)

(a) EPniPKTE Bits of UTRG (n = 0 to 5)
Note that 1 is not automatically written to EPniPKTE in case of auto-request transfer. Always write 1 to EPniPKTE by the CPU. The following example shows when 150-byte data is transmitted from EP2i to the host. In this case, 1 should be written to EP2iPKTE three times as shown in figure 15.27.

(b) EP2i DMA Transfer Procedure
The DMAC transfer unit should be one packet. Therefore, set the number of transfers so that it is equal to or less than the maximum packet size of each endpoint.

1. Confirm that UIFR1/EP2iEMPTY flag is 1.
2. DMAC settings for EP2i data transfer (such as auto-request and address setting).
3. Set the number of transfers for 64 bytes (the maximum packet size or less) in the DMAC.
4. Activate the DMAC (write 1 to DTE after reading DTE as 0).
5. DMA transfer.
6. Write 1 to the UTRG0/EP2iPKTE bit after the DMA transfer is completed.
7. Repeat steps 1 to 6 above.
8. Confirm that UIFR1/EP2iEMPTY flag is 1.
9. Set the number of transfer for 22 bytes in the DMAC.
10. Activate the DMAC (write 1 to DTE after reading DTE as 0).
11. DMA transfer.
12. Write 1 to the UTRG0/EP2iPKTE bit after the DMA transfer is completed.

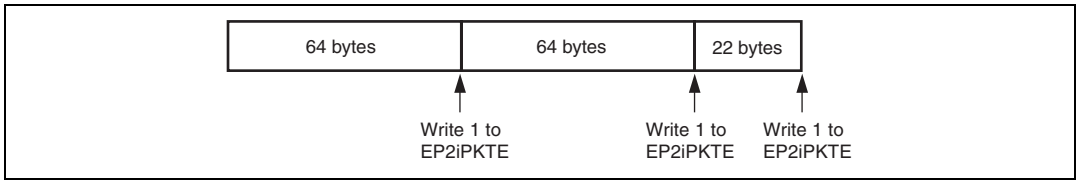


Figure 15.27 EP2iPKTE Operation in UTRG0 (Auto-Request)

(4) EPno DMA Transfer (n = 0, 2, 4)

(a) EPnoRDFN Bits of UTRG (n = 0, 2, 4)

Note that 1 is not automatically written to EPnoRDFN in case of auto-request transfer. Always write 1 to EPnoRDFN by the CPU. The following example shows when EP2o receives 150-byte data from the host. In this case, 1 should be written to EP2oRDFN three times as shown in figure 15.28.

(b) EP2o DMA Transfer Procedure

The DMAC transfer unit should be one packet. Therefore, set the number of transfers so that it is equal to or less than the maximum packet size of each endpoint.

1. Wait for the UIFR1/EP2oREADY flag to be set.
2. DMAC settings for EP2o data transfer (such as auto-request and address setting). Read value of UESZ2o and specify number of transfers to match size of received data (64 bytes or less).
3. Activate the DMAC (write 1 to DTE after reading DTE as 0).
4. DMA transfer (transfer of 64 bytes or less).
5. Write 1 to the UTRG0/EP2oRDFN bit after the DMA transfer is completed.
6. Repeat steps 1 to 5 above.

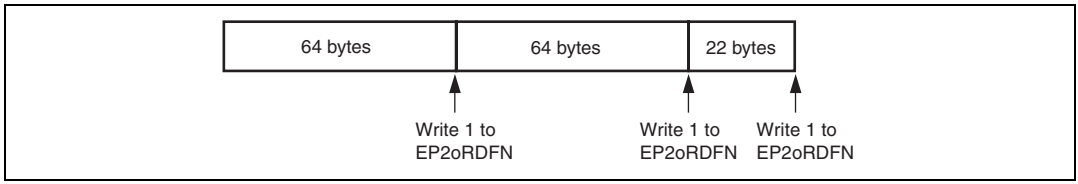


Figure 15.28 EP2oRDFN Operation in UTRG0 (Auto-Request)

15.7 Endpoint Configuration Example

Figure 15.29 shows an example of endpoint configuration. EPINFO data for the endpoint configuration shown in figure 15.29 is shown in table 15.9. In this example, two endpoints are not used. However, note that to load all EPINFO data from UEP1R00_0 to UEPIR22_4, dummy data must be written to the unused endpoints. An example of dummy data is also shown in table 15.9.

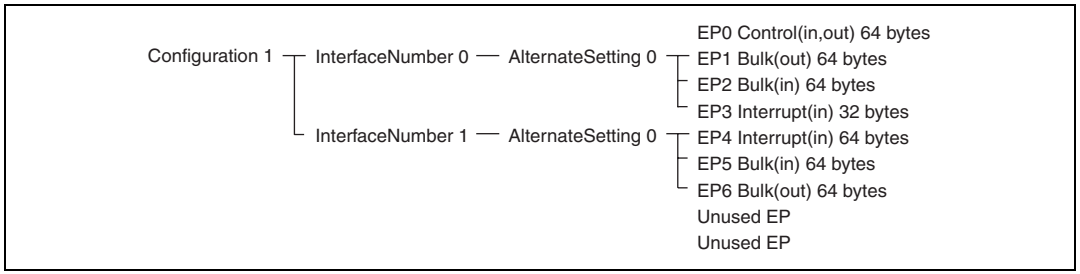


Figure 15.29 Endpoint Configuration Example

If endpoints are configured as shown in figure 15.27, some register names change as shown in table 15.7. In addition, some register bit names also change as shown in table 15.8. In the example shown in figure 15.27, register or bit names are modified for those determined based on the Bluetooth standard as follows: EP1i→EP3, EP2i→EP2, EP2o→EP1, EP4i→EP5, EP4o→EP6, and EP5i→EP4.

Table 15.7 Register Name Modification List

Register Name Based on Bluetooth Standard	Modified Register Name	Abbreviation	R/W	Initial Value	Address	Access Width
UEDR1i	USB endpoint data register 3 (For Interrupt_in data transfer)	UEDR3	W	H'00	H'C0009C to H'C0009F	8
UEDR2i	USB endpoint data register 2 (For Bulk_in data transfer)	UEDR2	W	H'00	H'C000A0 to H'C000A3	8
UEDR2o	USB endpoint data register 1 (For Bulk_out data transfer)	UEDR1	R	Undefined	H'C000A4 to H'C000A7	8
UEDR3i	Reserved register (For Isochronous_in data transfer)*	(UEDRn)*	W	H'00	H'C000A8 to H'C000AB	8
UEDR3o	Reserved register (For Isochronous_out data transfer)*	(UEDRn)*	R	Undefined	H'C000AC to H'C000AF	8
UEDR4i	USB endpoint data register 5 (For Bulk_in data transfer)	UEDR5	W	H'00	H'C000B0 to H'C000B3	8
UEDR4o	USB endpoint data register 6 (For Bulk_out data transfer)	UEDR6	R	Undefined	H'C000B4 to H'C000B7	8
UEDR5i	USB endpoint data register 4 (For Interrupt_in data transfer)	UEDR4	W	H'00	H'C000B8 to H'C000BB	8
USEZ2o	USB endpoint receive data size register 1 (For Bulk_out data transfer)	UESZ1	R	Undefined	H'C000BD	8
USEZ3o	Reserved register (For Isochronous_out data transfer)*	(UESZn)*	R	Undefined	H'C000BE	8
USEZ4o	USB endpoint receive data size register 6 (For Bulk_out data transfer)	UESZ6	R	Undefined	H'C000BF	8

Note: * Registers related to unused endpoints are handled as reserved registers.

Table 15.8 Bit Name Modification List

Abbreviation	R/W	Initial Value	Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
UDMAR	R/W	H'00	H'C00082	EP6T1	EP6T0	EP5T1	EP5T0	EP1T1	EP1T0	EP2T1	EP2T0
UTRG0	W	H'00	H'C00084	—	—	EP1RDFN	EP2PKTE	EP3PKTE	EP0o RDFN	EP0iPKTE	EP0s RDFN
UTRG1	W	H'00	H'C00085	—	—	—	—	—	EP4PKTE	EP6RDFN	EP5PKTE
UFCLR0	W	H'00	H'C00086	(EPnCLR)	(EPnCLR)	EP1CLR	EP2CLR	EP3CLR	EP0oCLR	EP0iCLR	—
UFCLR1	W	H'00	H'C00087	—	—	—	—	—	EP4CLR	EP6CLR	EP5CLR
UESTL0	R/W	H'00	H'C00088	(EPnSTL)	(EPnSTL)	EP1STL	EP2STL	EP3STL	—	—	EP0STL
UESTL1	R/W	H'00	H'C00089	SCME	—	—	—	—	EP4STL	EP6STL	EP5STL
UIFR0	R/W	H'00	H'C000C0	BRST	—	EP3TR	EP3TS	EP0oTS	EP0iTR	EP0iTS	SetupTS
UIFR1	R/W	H'01*1 or H'09	H'C000C1	(EPnTF)	(EPnTS)	(EPnTF)	(EPnTR)	*2 EP2ALL EMPTY	EP1 READY	EP2TR	EP2 EMPTY
UIFR2	R/W	H'01*1 or H'09	H'C000C2	—	—	EP4TR	EP4TS	*2 EP5ALL EMPTY	EP6 READY	EP5TR	EP5 EMPTY
UIER0	R/W	H'00	H'C000C4	BRSTE	—	EP3TRE	EP3TSE	EP0oTSE	EP0iTRE	EP0iTSE	SetupTSE
UIER1	R/W	H'00	H'C000C5	(EPnTFE)	(EPnTSE)	(EPnTFE)	(EPnTRE)	*2 EP2ALL EMPTYE	EP1 READYE	EP2TRE	EP2 EMPTYE
UIER2	R/W	H'00	H'C000C6	—	—	EP4TRE	EP4TSE	*2 EP5ALL EMPTYE	EP6 READYE	EP5TRE	EP5 EMPTYE
UISR0	R/W	H'00	H'C000C8	BRSTS	—	EP3TRS	EP3TSS	EP0oTSS	EP0iTRS	EP0iTSS	SetupTSS
UISR1	R/W	H'00	H'C000C9	(EPnTFS)	(EPnTSS)	(EPnTFS)	(EPnTRS)	*2 EP2ALL EMPTYS	EP1 READYS	EP2TRS	EP2 EMPTYS
UISR2	R/W	H'00	H'C000CA	—	—	EP4TRS	EP4TSS	*2 EP5ALL EMPTYS	EP6 READYS	EP5TRS	EP5 EMPTYS
UDSR	R	H'00	H'C000CC	—	—	EP4DE	EP5DE	—	EP2DE	EP3DE	EP0iDE

Notes: 1. H'01 in H8S/2215. H'09 in H8S/2215R and H8S/2215T.
2. Available only in H8S/2215R and H8S/2215T. "—" in H8S/2215.

Table 15.9 shows the EPINFO data for the endpoint configuration shown in figure 15.27.

This USB module is optimized by the hardware specific to the transfer type. Accordingly, endpoints cannot be configured completely freely. Endpoint configuration can be modified within the restriction as shown in table 15.9 (data indicated within parentheses [ ]), data other than that within parentheses [ ] must be specified the value shown in table 15.9. For unused endpoints, dummy data (0) must be written.

Table 15.9 EPINFO Data Settings

			EPINFO Data Settings Based on Bluetooth Standard						
No.	Register Name	Address	Corresponding Transfer Mode*1	UEPIRn_0 to UEPIRn_4 Settings*2	UEPI Rn_0	UEPI Rn_1	UEPI Rn_2	UEPI Rn_3	UEPI Rn_4
1	UEPIR00_0 to UEPIR00_4	H'C00000 to H'C0004	Specific to Control transfer	B'0000_00_00_000_00_0_ 0001000000_0000000000000000	H'00	H'00	H'40	H'00	H'00
2	UEPIR01_0 to UEPIR01_4	H'C00005 to H'C0009	Specific to Interrupt in transfer	B'[0011]_01_[00]_[000]_11_1_ [0000100000]_0000000000000001*3	H'34	H'1C	H'20	H'00	H'01
3	UEPIR02_0 to UEPIR02_4	H'C0000A to H'C000E	Specific to Bulk in transfer	B'[0010]_01_[00]_[000]_10_1_ [0001000000]_0000000000000010*4	H'24	H'14	H'40	H'00	H'02
4	UEPIR03_0 to UEPIR03_4	H'C0000F to H'C0013	Specific to Bulk out transfer	B'[0001]_01_[00]_[000]_10_0 [0001000000]_0000000000000011*4	H'14	H'10	H'40	H'00	H'03
5	UEPIR04_0 to UEPIR04_4	H'C00014 to H'C0018	Specific to Isoch in transfer	B'[0000]_01_[00]_[000]_01_1_ [0000000000]_0000000000000100*5*6	H'04	H'0C	H'00	H'00	H'04
6	UEPIR05_0 to UEPIR05_4	H'C00019 to H'C001D	Specific to Isoch out transfer	B'[0000]_01_[00]_[000]_01_0_ [0000000000]_0000000000000101*5*6	H'04	H'08	H'00	H'00	H'05
7	UEPIR06_0 to UEPIR06_4	H'C0001E to H'C0022	Specific to Isoch in transfer	B'[0000]_01_[00]_[000]_01_1_ [0000000000]_0000000000000110*5*6	H'04	H'0C	H'00	H'00	H'06
8	UEPIR07_0 to UEPIR07_4	H'C00023 to H'C0027	Specific to Isoch out transfer	B'[0000]_01_[00]_[000]_01_0_ [0000000000]_0000000000000111*5*6	H'04	H'08	H'00	H'00	H'07
9	UEPIR08_0 to UEPIR08_4	H'C00028 to H'C002C	Specific to Isoch in transfer	B'[0000]_01_[00]_[000]_01_1_ [0000000000]_0000000000001000*5*6	H'04	H'0C	H'00	H'00	H'08
10	UEPIR09_0 to UEPIR09_4	H'C0002D to H'C0031	Specific to Isoch out transfer	B'[0000]_01_[00]_[000]_01_0_ [0000000000]_0000000000001001*5*6	H'04	H'08	H'00	H'00	H'09
11	UEPIR10_0 to UEPIR10_4	H'C00032 to H'C0036	Specific to Isoch in transfer	B'[0000]_01_[00]_[000]_01_1_ [0000000000]_0000000000001010*5*6	H'04	H'0C	H'00	H'00	H'0A
12	UEPIR11_0 to UEPIR11_4	H'C00037 to H'C003B	Specific to Isoch out transfer	B'[0000]_01_[00]_[000]_01_0_ [0000000000]_0000000000001011*5*6	H'04	H'08	H'00	H'00	H'0B
13	UEPIR12_0 to UEPIR12_4	H'C0003C to H'C0040	Specific to Isoch in transfer	B'[0000]_01_[00]_[000]_01_1_ [0000000000]_0000000000001100*5*6	H'04	H'0C	H'00	H'00	H'0C
14	UEPIR13_0 to UEPIR13_4	H'C00041 to H'C0045	Specific to Isoch out transfer	B'[0000]_01_[00]_[000]_01_0_ [0000000000]_0000000000001101*5*6	H'04	H'08	H'00	H'00	H'0D
15	UEPIR14_0 to UEPIR14_4	H'C00046 to H'C004A	Specific to Isoch in transfer	B'[0000]_01_[00]_[000]_01_1_ [0000000000]_0000000000001110*5*6	H'04	H'0C	H'00	H'00	H'0E
16	UEPIR15_0 to UEPIR15_4	H'C0004B to H'C004F	Specific to Isoch out transfer	B'[0000]_01_[00]_[000]_01_0_ [0000000000]_0000000000001111*5*6	H'04	H'08	H'00	H'00	H'0F
17	UEPIR16_0 to UEPIR16_4	H'C00050 to H'C0054	Specific to Isoch in transfer	B'[0000]_01_[00]_[000]_01_1_ [0000000000]_0000000000010000*5*6	H'04	H'0C	H'00	H'00	H'10

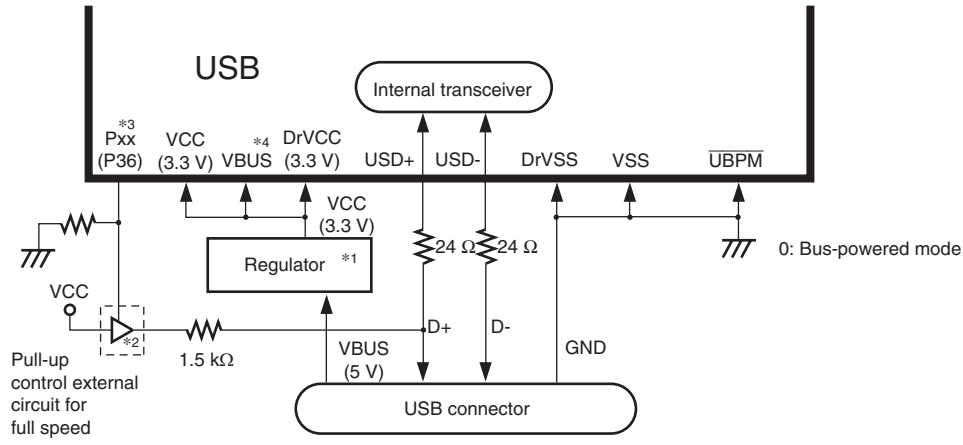
EPINFO Data Settings Based on Bluetooth Standard

No.	Register Name	Address	Corresponding Transfer Mode*1	UEPIRn_0 to UEPIRn_4 Settings*2	UEPI Rn_0	UEPI Rn_1	UEPI Rn_2	UEPI Rn_3	UEPI Rn_4
18	UEPIR17_0 to UEPIR17_4	H'C00055 to H'C0059	Specific to Isoch out transfer	B'[0000]_01_[00]_[000]_01_0_ [0000000000]_0000000000010001*5*6	H'04	H'08	H'00	H'00	H'11
19	UEPIR18_0 to UEPIR18_4	H'C0005A to H'C005E	Specific to Isoch in transfer	B'[0000]_01_[00]_[000]_01_1_ [0000000000]_0000000000010010*5*6	H'04	H'0C	H'00	H'00	H'12
20	UEPIR19_0 to UEPIR19_4	H'C0005F to H'C0063	Specific to Isoch out transfer	B'[0000]_01_[00]_[000]_01_0_ [0000000000]_0000000000010011*5*6	H'04	H'08	H'00	H'00	H'13
21	UEPIR20_0 to UEPIR20_4	H'C00064 to H'C0068	Specific to Bulk in transfer	B'[0101]_01_[01]_[000]_10_1_ [0001000000]_0000000000010100*4	H'55	H'14	H'40	H'00	H'14
22	UEPIR21_0 to UEPIR21_4	H'C00069 to H'C006D	Specific to Bulk out transfer	B'[0110]_01_[01]_[000]_10_0_ [0001000000]_0000000000010101*4	H'65	H'10	H'40	H'00	H'15
23	UEPIR22_0 to UEPIR22_4	H'C0006E to H'C0072	Specific to Interrupt in transfer	B'[0100]_01_[01]_[000]_11_1_ [0001000000]_0000000000010110*3	H'45	H'1C	H'40	H'00	H'16

Notes:
1. Each endpoint is optimized by the hardware specific for the transfer mode. The transfer mode shown in table 15.8 must be specified. (D28 and D27 for all EPINFO data items must e specified as shown in table 15.8.)
2. Data indicated within parentheses [ ] can be modified. Data other than that within parentheses [ ] must be specified as shown in table 15.8.
3. Maximum packet size of Interrupt transfer must be from 0 to 64.
4. Maximum packet size of Bulk transfer must be 64 when used or 0 when unused.
5. Maximum packet size of Isochronous transfer must be from 0 to 128. Endpoint number of Isochronous_in can differ from that of Isochronous_out. However, note that endpoint numbers of all Isochronous_in must be the same. Endpoint numbers of all Isochronous_out must also be the same.
6. Maximum packet size of the unused endpoint must be 0.

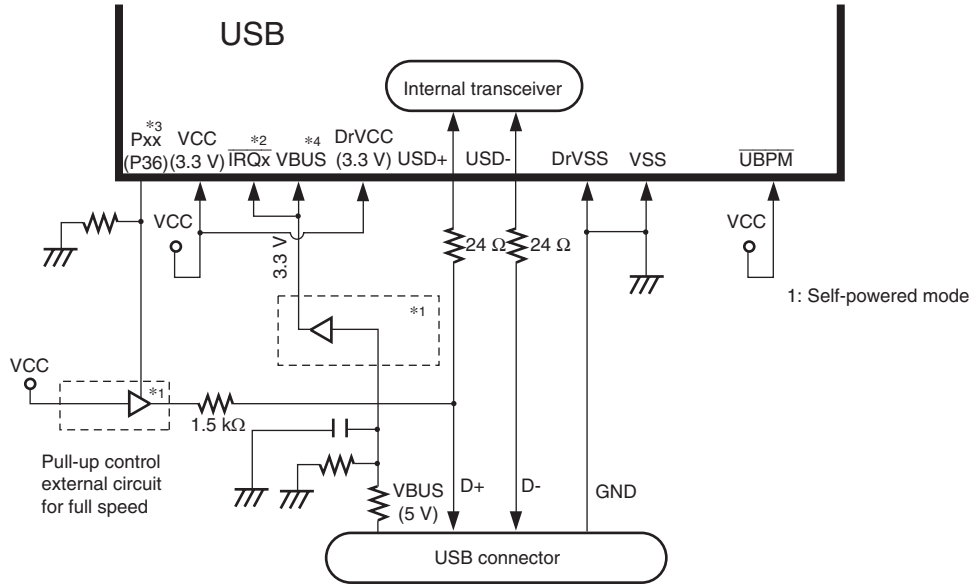
15.8 USB External Circuit Example

Figures 15.30 and 15.31 show the USB external circuit examples when the on-chip transceiver is used. Figures 15.32 and 15.33 show the USB external circuit examples when an external transceiver is used.



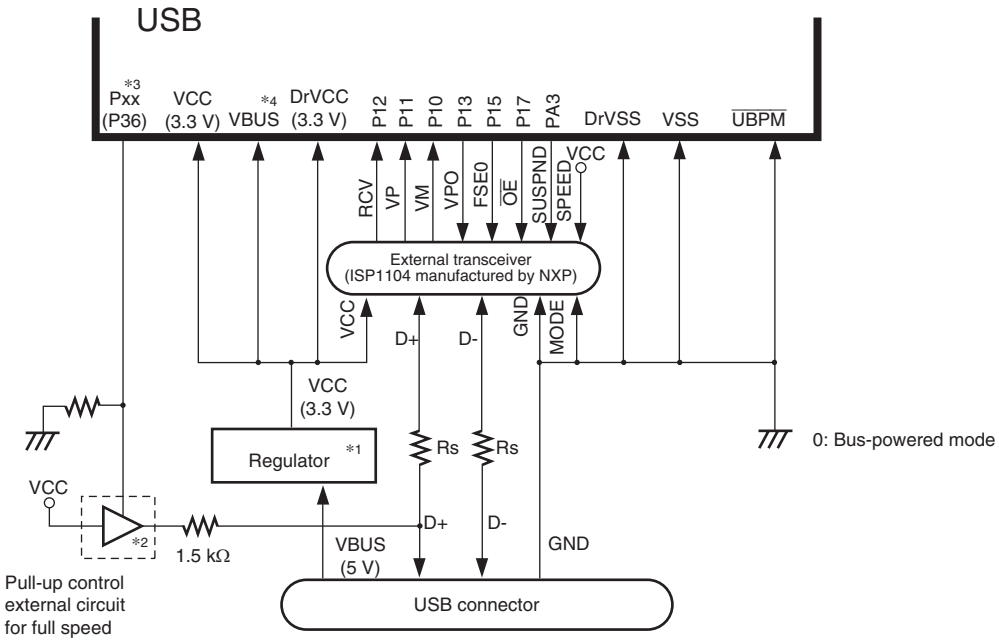
Notes: 1. Step-down to the operating voltage VCC (3.3 V) of this LSI.
2. To protect the LSI, voltage applicable IC such as HD74LV-A series must be used even when the system power is turned off.
3. In HD64F2215, HD64F2215R, HD64F2215T, HD6432215A, HD6432215B, and HD6432215C, Pxx should be assigned to an output port as the D+ pull-up control pin.
In HD64F2215U, HD64F2215RU, and HD64F2215TU, in which on-chip ROM can be programmed by using the USB, P36 should be used as the D+ pull-up control pin.
4. Steps should be taken prevent noise from affecting the VBUS terminal during USB communications.

Figure 15.30 USB External Circuit in Bus-Powered Mode (When On-Chip Transceiver Is Used)



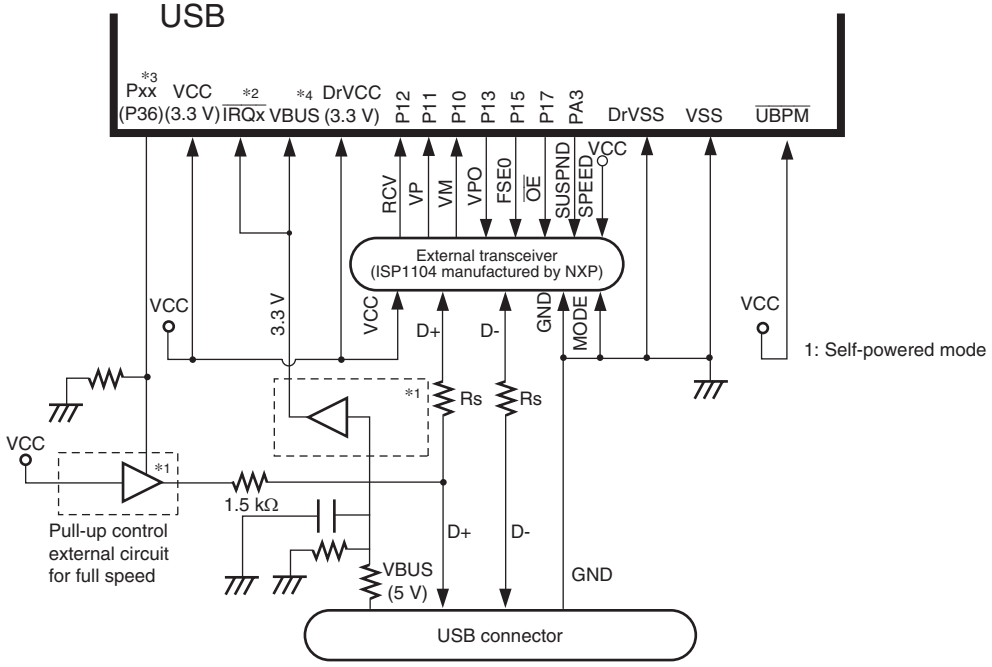
Notes: 1. To protect the LSI, voltage applicable IC such as HD74LV-A series must be used even when the system power is turned off.
2. To cancel software standby state by detecting the USB cable disconnection, the level shifter signal must also be connected to the $\overline{\text{IRQx}}$ pin. Note that the software standby state cannot be canceled by the USB interrupt $\overline{\text{EXIRQx}}$.
3. In HD64F2215, HD64F2215R, HD64F2215T, HD6432215A, HD6432215B, and HD6432215C, Pxx should be assigned to an output port as the D+ pull-up control pin.
In HD64F2215U, HD64F2215RU, and HD64F2215TU, in which on-chip ROM can be programmed by using the USB, P36 should be used as the D+ pull-up control pin.
4. Steps should be taken prevent noise from affecting the VBUS terminal during USB communications.

Figure 15.31 USB External Circuit in Self-Powered Mode (When On-Chip Transceiver Is Used)



Notes: 1. Step-down to the operating voltage VCC (3.3 V) of this LSI.
2. To protect the LSI, voltage applicable IC such as HD74LV-A series must be used even when the system power is turned off.
3. In HD64F2215, HD64F2215R, HD64F2215T, HD6432215A, HD6432215B, and HD6432215C, Pxx should be assigned to an output port as the D+ pull-up control pin.
In HD64F2215U, HD64F2215RU, and HD64F2215TU, in which on-chip ROM can be programmed by using the USB, P36 should be used as the D+ pull-up control pin.
4. Steps should be taken prevent noise from affecting the VBUS terminal during USB communications.

Figure 15.32 USB External Circuit in Bus-Powered Mode (When External Transceiver Is Used)



Notes: 1. To protect the LSI, voltage applicable IC such as HD74LV-A series must be used even when the system power is turned off.

2. To cancel software standby state by detecting the USB cable disconnection, the VBUS signal must also be connected to the $\overline{\text{IRQx}}$ pin (Note that the software standby state cannot be canceled by the USB internal interrupt $\overline{\text{EXIRQx}}$).
3. In HD64F2215, HD64F2215R, HD64F2215T, HD6432215A, HD6432215B, and HD6432215C, Pxx should be assigned to an output port as the D+ pull-up control pin.
   In HD64F2215U, HD64F2215RU, and HD64F2215TU, in which on-chip ROM can be programmed by using the USB, P36 should be used as the D+ pull-up control pin.
4. Steps should be taken prevent noise from affecting the VBUS terminal during USB communications.

Figure 15.33 USB External Circuit in Self-Powered Mode (When External Transceiver Is Used)

15.9 Usage Notes

15.9.1 Operating Frequency

- In H8S/2215

  When the on-chip PLL circuit is used, the system clock of this LSI must be 16 MHz. This 16-MHz system clock, used as base clock, is tripled in the on-chip PLL circuit to generate the 48-MHz USB operating clock. When the USB operating clock (48 MHz) oscillator or 48-MHz external clock is used, the system clock of the LSI must be 13-MHz to 16-MHz. Medium-speed mode is not supported; use full-speed mode.
- In H8S/2215R and H8S/2215T

  When the on-chip PLL circuit is used, the system clock of this LSI must be 16 MHz or 24 MHz. If the system clock frequency is 16 MHz, it is tripled by the on-chip PLL circuit, and if the system clock frequency is 25 MHz, it is doubled, to generate the 48-MHz USB operating clock. When the USB operating clock (48 MHz) oscillator or 48-MHz external clock is used, the system clock of the LSI must be 13 MHz to 24 MHz. Medium-speed mode is not supported; use full-speed mode.

Note: On the H8S/2215T, use a 16-MHz or 24-MHz system clock for the MCU, even if a 48-MHz oscillator or 48-MHz external clock is used as the USB operation clock.

15.9.2 Bus Interface

This module's interface is based on the bus specifications of external area 6. Before accessing the USB, area 6 must be specified as having an 8-bit bus width and 3-state access using the bus controller register. In mode 7 (single-chip mode), the USB module cannot be accessed. In mode 6 (internal ROM enabled mode), $\overline{CS6}$ and A7 to A0 pins are used as inputs at initialization and USB cannot be accessed. Before access to this module, set P72DDR to 1 and PC7DDR to PC0DDR to H'FF, respectively, to use $\overline{CS6}$ and A7 to A0 pins as outputs. In mode 4 or 5 (on-chip ROM disabled mode), set P72DDR to 1 to use the $\overline{CS6}$ pin as an output.

15.9.3 Setup Data Reception

The following must be noted for the EP0s FIFO used to receive 8-byte setup data. The USB is designed to always receive setup commands. Accordingly, write from the UDC has higher priority than read from the LSI. If the reception of the next setup command starts while the is LSI reading data after completing reception, this data read from the LSI is forcibly cancelled and the next setup command write starts. After the next setup command write, data read from the LSI is thus

undefined. Read operation is forcibly disabled because data cannot be guaranteed if DP-RAM used as FIFO accesses the same address for write and read.

15.9.4 FIFO Clear

If the USB cable is disconnected during communication, old data may be contained in the FIFO. Accordingly, FIFO must be cleared immediately after USB cable connection. In addition, after bus reset, all FIFO must also be cleared. Note, however, that FIFOs that are currently used for data transfer to or from the host must not be cleared.

15.9.5 $\overline{\text{IRQ6}}$ Interrupt

A suspend/resume interrupt requested by $\overline{\text{IRQ6}}$ must be specified as falling-edge sensitive.

15.9.6 Data Register Overread or Overwrite

When the CPU reads or writes to data registers, the following must be noted:

- Transmit data registers (UEDR0i, UEDR1i, UEDR2i, UEDR3i, UEDR4i, UEDR5i)
  Data to be written to the transmit data registers must be within the maximum packet size. For the transmit data registers of EP2i, EP3i, and EP4i having a dual-FIFO configuration, data to be written at any time must be within the maximum packet size. In this case, after a data write, the FIFO is switched to the other FIFO, enabling an further data write when the PKTE bit of UTRG is set to 1 (in EP3i, the same operation is automatically performed when the SOF packet is received). Accordingly, data of size corresponding to two FIFO must not be written to the transmit data registers of EP2i, EP3i, and EP4i at a time.
- Receive data registers (UEDR0o, UEDR2o, UEDR3o, UEDR4o)
  Receive data registers must not read a data size that is greater than the effective size of the read data item. In other words, receive data registers must not read data with data size larger than that specified by the receive data size register. For the receive data registers of EP2o, EP3o, and EP4o having a dual-FIFO configuration, data to be read at any time must be within the maximum packet size. In this case, after reading the currently selected FIFO, set the RDFN bit of UTRG to 1 (in EP3o, the same operation is automatically performed when the SOF packet is received). This switches the FIFO to the other FIFO and updates the receive data size, enabling the next data read. In addition, if there is no receive data in a FIFO, data must not be read. Otherwise, the pointer that controls the internal module FIFO is updated and correct operation cannot be guaranteed.

15.9.7 EP3o Isochronous Transfer

- Reception of EP3o data larger than the maximum packet size

  The EP3o data FIFO cannot receive data with size larger than the maximum packet size; the excessive data is lost. In this case, the receive size register 3o (UESZ3o) can count up to the maximum packet size and the EP3o abnormal transfer flag (EP3oTF) is set to 1.

  Figure 15.34 shows the 10-byte data reception when the maximum packet size is specified as 9 bytes.

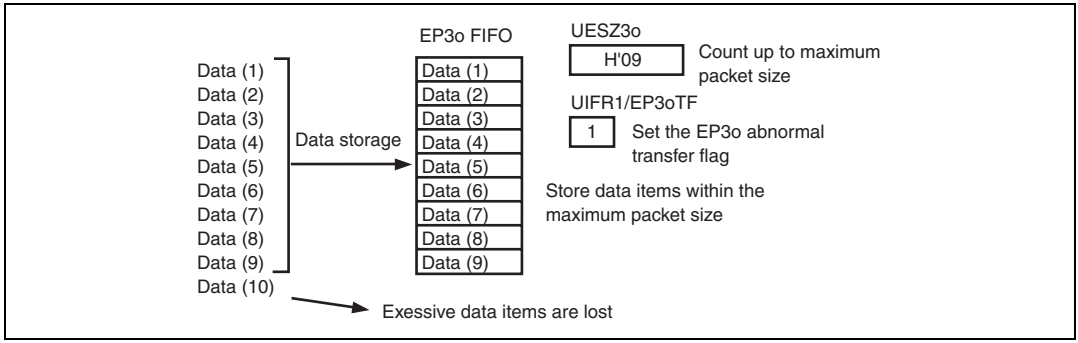


Figure 15.34 10-Byte Data Reception

- EP3o receive data and status bit reading

  As shown in figure 15.35, FIFO are switched on SOF packet reception. FIFOs thus store the latest data. Accordingly, receive data sent from the host in frame [N] can only be read in frame [N+1]. In addition, the EP3oTF and EP3oTS status bits of UIFR1 are automatically switched on with each SOF packet reception; the EP3oTF and EP3oTS status in frame [N] can only be read in frame [N + 1].

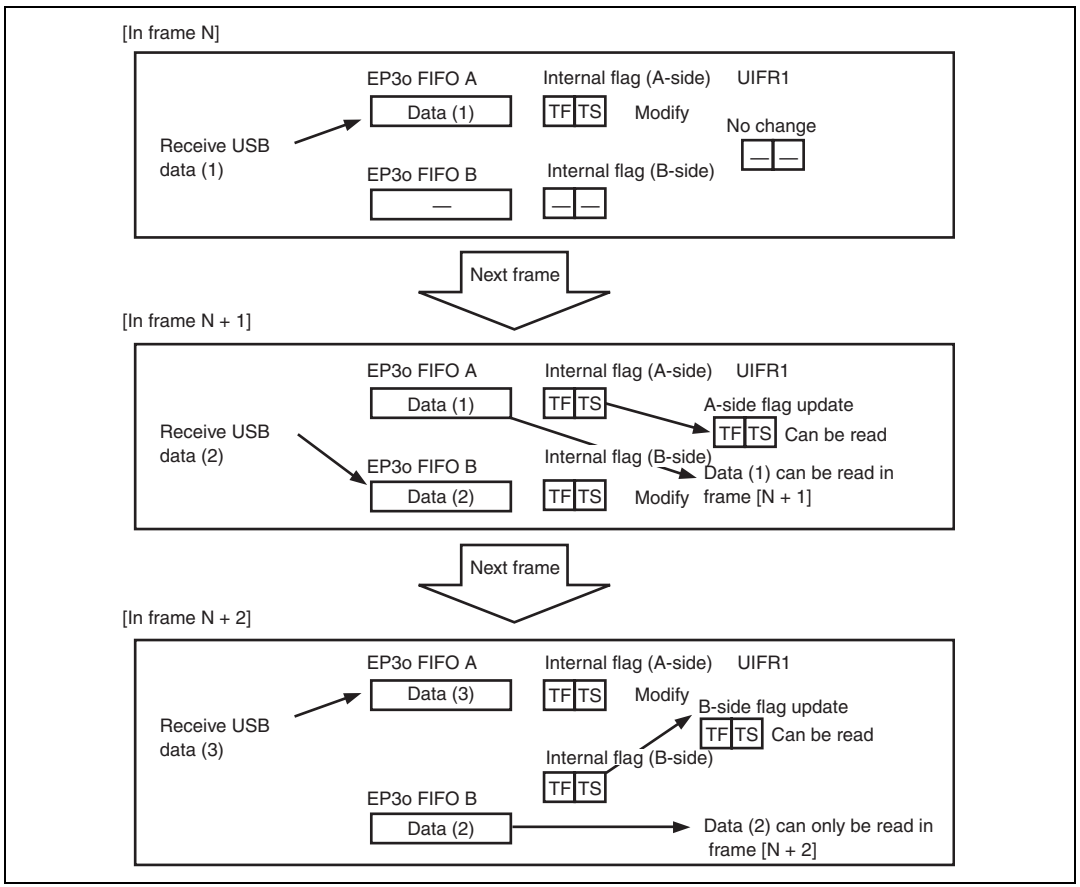


Figure 15.35 EP3o Data Reception

15.9.8 Reset

- A manual reset should not be performed during USB communication as the LSI will stop with the USD+, USD- pin state maintained. This USB module uses synchronous reset for some registers. The reset state of these registers must be cancelled after the clock oscillation stabilization time has passed. At initialization, reset must be cancelled using the following procedure:

  1. Select the USB operating clock: Specify the UCKS3 to UCKS0 bits in UCTLR.
  2. Cancel the USB module stop mode: Clear the MSTPB0 bit in MSTPCRB to 0.
  3. Wait for the USB clock stabilization time: Wait until the CK48READY bit in UIFR3 is set to 1.
  4. Cancel the USB interface reset state: Clear the UIFRST bit in UCTLR to 0.
  5. Cancel the UDC core reset state: Clear the UDCRST bit in UCTLR to 0.

  For detail, see the flowcharts in section 15.5.1, Initialization, and section 15.5.2, USB Cable Connection/Disconnection.

- The USB registers are not initialized when the watchdog timer (WDT) triggers a power-on reset. Therefore, the USB may not operate properly after a power-on reset is triggered by the WDT due to CPU runaway or a similar cause. (If a power-on reset is triggered by input of a power-on reset signal from the $\overline{\text{RES}}$ pin, the USB registers are initialized and there is no problem.) Consequently, an initialization routine should be used to write the initial values listed below to the following three registers, thereby ensuring that all the USB registers are properly initialized, immediately following a reset.

  UCTLR = H'03, UIER3 = H'80, UIFR3 = H'00

15.9.9 EP0 Interrupt Assignment

EP0 interrupt sources assigned to bits 3 to 0 in UIFR0 must be assigned to the same interrupt sign ($\overline{\text{EXIRQx}}$) by setting UISR0. There are no other restrictions on EP0 interrupt sources.

15.9.10 Level Shifter for VBUS and $\overline{\text{IRQx}}$ Pins

The VBUS and $\overline{\text{IRQx}}$ pins of this USB module must be connected to the USB connector's VBUS pin via a level shifter. This is because the USB module has a circuit that operates by detecting USB cable connection or disconnection.

Even if the power of the device incorporating this USB module is turned off, 5-V power is applied to the USB connector's VBUS pin while the USB cable is connected to the device set. To protect the LSI from destruction, use a level shifter such as the HD74LV-A series, which allows voltage application to the pin even when the power is off.

15.9.11 Read and Write to USB Endpoint Data Register

To write data to an USB endpoint data register (UEDRni) on the transmit side using a CPU word or longword transfer instruction, the correct size of data must be written to the USB endpoint data register. Otherwise, an error may occur.

For example, when 7-byte data is transferred to the host, 8-byte data is sent to the host if data is written twice by the longword transfer instructions or if data is written four times by the word transfer instructions. To write 7-byte data correctly, data must be written once by a longword transfer instruction, once by a word transfer instruction, and once by a byte transfer instruction, or data must be written three times by a word transfer instruction and once by a byte transfer instruction.

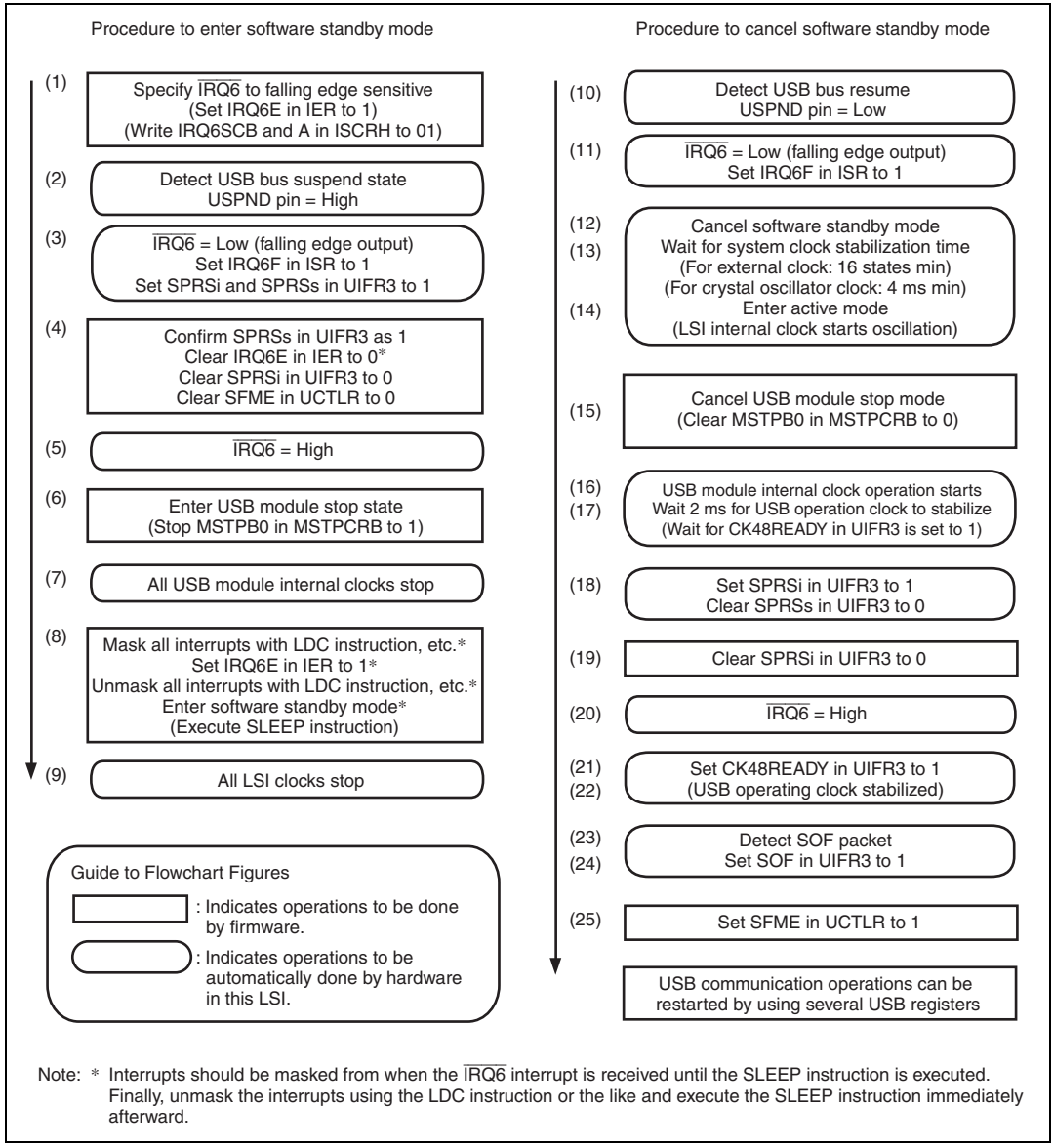
To read data from the USB endpoint data register (UEDRno) on the receive side, the correct size of data must be read. In this case, the data size is specified by the USB endpoint receive size data register (UESZno).

To execute DMA transfer on data in the USB endpoint data register using the on-chip DMAC, byte transfer musts be used. In word transfer, odd-byte data cannot be transferred. Word transfer is thus disabled.

15.9.12 Restrictions for Software Standby Mode Transition

Before entering the software standby mode, disabled the SOF marker function and set the USB module stop state as shown in figure 15.34. The UDC core must not be reset.

To access the USB module after software standby mode, cancel the USB module stop state and wait for the USB operating clock (48 MHz) stabilization time as shown in figure 15.34.



Note: * Interrupts should be masked from when the IRQ6 interrupt is received until the SLEEP instruction is executed. Finally, unmask the interrupts using the LDC instruction or the like and execute the SLEEP instruction immediately afterward.

Figure 15.36 Transition to and from Software Standby Mode

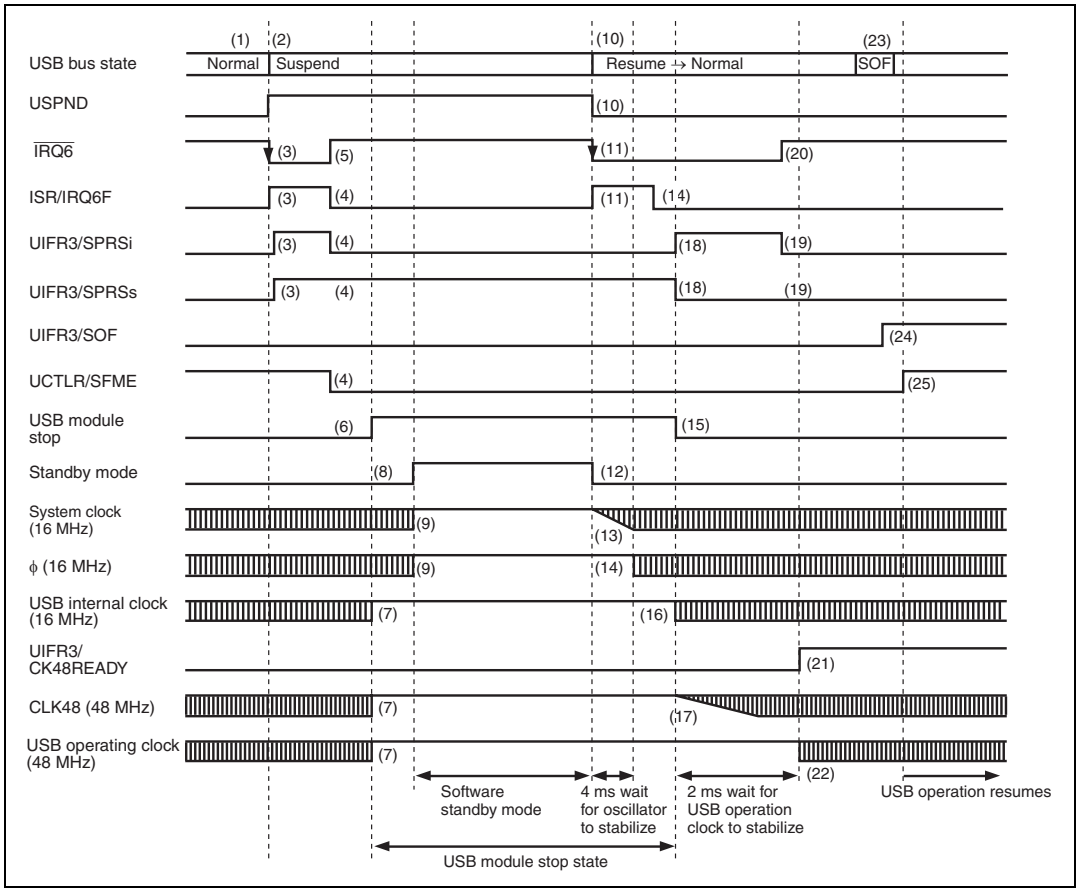


Figure 15.37 USB Software Standby Mode Transition Timing

15.9.13 USB External Circuit Example

The USB external circuit examples are used for reference only. In actual board design, carefully check the system operation. In addition, the USB external circuits examples cannot guarantee correct system operation. The user must individually take measures against external surges or ESD noise by incorporating protective diodes or other components if necessary.

15.9.14 Pin Processing when USB Not Used

Pin processing should be performed as follows.

DrVCC = Vcc, DrVSS = 0 V, USD+ = USD- = USPND = open state, VBUS = $\overline{\text{UBPM}}$ = 0 V

15.9.15 Notes on Emulator Usage

Using the I/O register window function, or the like, to display UEDR0o, UEDR2o, UEDR3o, and UEDR4o can cause the EP0oFIFO, EP2oFIFO, EP3oFIFO, and EP4oFIFO read pointers to malfunction, preventing UEDR0o to UEDR4o and UESZ0o to UESZ4o from being read correctly. Therefore, UEDR0o to UEDR4o should not be displayed.

15.9.16 Notes on TR Interrupt

Note the following when using the transfer request interrupt (TR interrupt) for IN transfer to EP0i, EP2i, EP3i, EP4i, or EP5i.

The TR interrupt flag is set if the FIFO for the target EP has no data when the IN token is sent from the USB host. However, at the timing shown in figure 15.38, multiple TR interrupts occur successively. Take appropriate measures against malfunction in such a case.

Note: This module determines whether to return NAK if the FIFO of the target EP has no data when receiving the IN token, but the TR interrupt flag is set only after a NAK handshake is sent. If the next IN token is sent before PKTE of UTRGx is written to, the TR interrupt flag is set again.

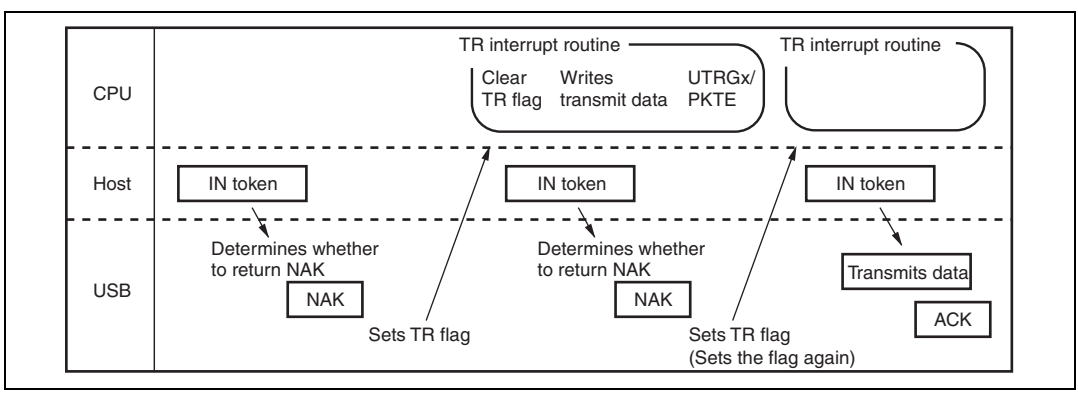


Figure 15.38 TR Interrupt Flag Set Timing

15.9.17 Notes on UIFRO

The bit-clear instruction cannot be used to clear a flag in some USB interrupt flag registers to 0. These registers have flags which are cleared to 0 by writing 0 and to which writing 1 is ignored. The concerning registers are USB interrupt flag registers 0 to 3 (UIFR0 to UIFR3) in the H8S/2215 Group.

A single bit-clear instruction actually executes reading the value of a register, modifying the read value, and writing the modified value. When clearing a flag with the bit-clear instruction, if a source which will set another flag is activated between reading and writing, the flag is unintentionally cleared to 0. Therefore, the bit-clear instruction cannot be used.

To clear these flags, write 0 to a flag which should be cleared and write 1 to other flags with the MOVE instruction. For example, to clear only bit 7, write H'7F and to clear bits 6 and 7, write H'3F.

Section 16 A/D Converter

This LSI includes a successive approximation type 10-bit A/D converter that allows up to six analog input channels to be selected. The block diagram of the A/D converter is shown in figure 16.1.

16.1 Features

- 10-bit resolution
- Six input channels
- Conversion time: 8.1 μs per channel (at 16-MHz operation), 10.7 μs per channel (at 24-MHz operation)*

Note: * Available only in H8S/2215R and H8S/2215T.

- Two operating modes
  - — Single mode: Single-channel A/D conversion
  - — Scan mode: Continuous A/D conversion on 1 to 4 channels
- Four data registers
  - — Conversion results are held in a 16-bit data register for each channel
- Sample and hold function
- Three methods conversion start
  - — Software
  - — Timer (TPU or TMR) conversion start trigger
  - — External trigger signal ($\overline{\text{ADTRG}}$)
- Interrupt request
  - — An A/D conversion end interrupt request (ADI) can be generated
- Module stop mode can be set
- Settable analog conversion voltage range

  Analog conversion voltage range settable using the reference voltage pin (Vref) as the reference voltage

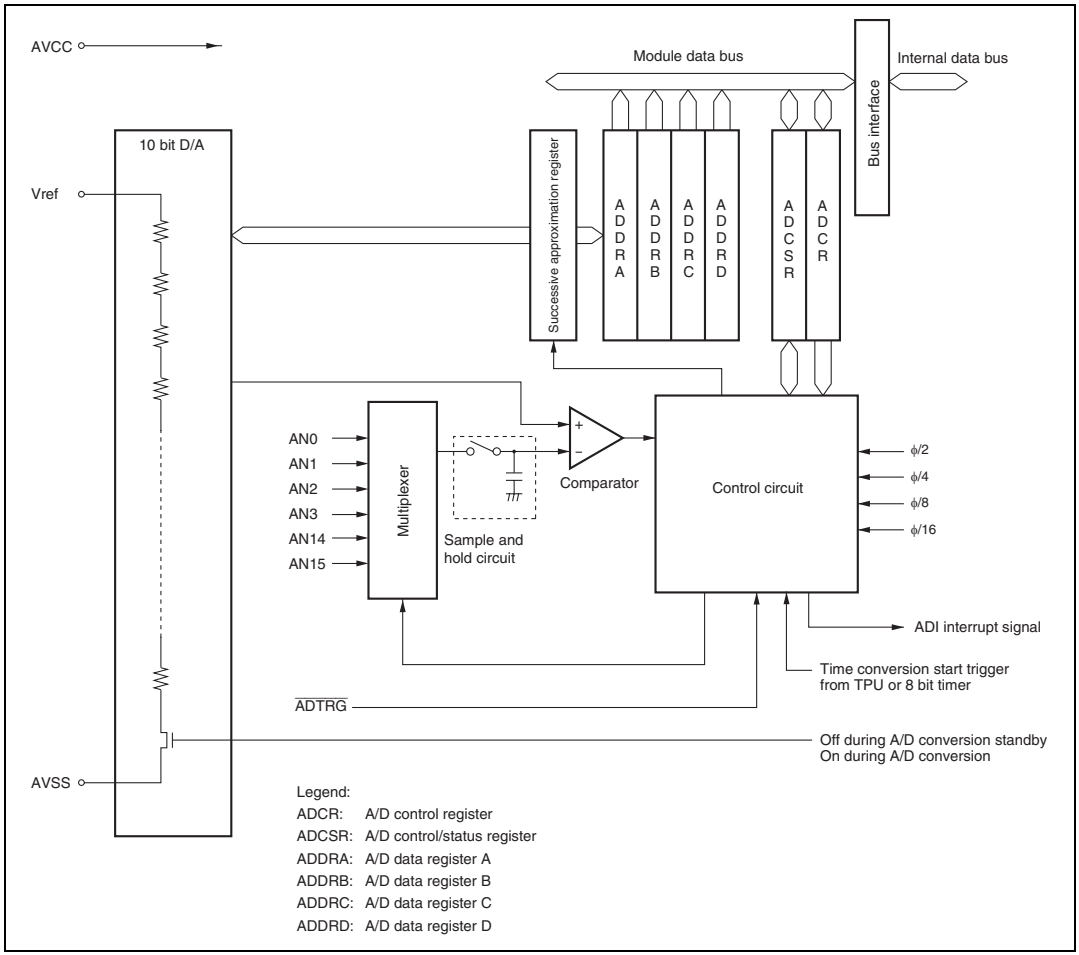


Figure 16.1 Block Diagram of A/D Converter

16.2 Input/Output Pins

Table 16.1 summarizes the input pins used by the A/D converter. The AN0 to AN3 and AN14 to AN15 pins are analog input pins. The AVCC and AVSS pins are the power supply pins for the analog block in the A/D converter. The Vref pin is the reference voltage pin for the A/D conversion.

Table 16.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Analog power supply pin	AVCC	Input	Analog block power supply and reference voltage
Analog ground pin	AVSS	Input	Analog block ground and reference voltage
Analog reference voltage pin	Vref	Input	Reference voltage pin for the A/D
Analog input pin 0	AN0	Input	Analog input pins
Analog input pin 1	AN1	Input	
Analog input pin 2	AN2	Input	
Analog input pin 3	AN3	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 pin for starting A/D conversion

16.3 Register Descriptions

The A/D converter has the following registers.

- A/D data register A (ADDRA)
- A/D data register B (ADDRB)
- A/D data register C (ADDRC)
- A/D data register D (ADDRD)
- A/D control/status register (ADCSR)
- A/D control register (ADCR)

16.3.1 A/D Data Registers A to D (ADDRA to ADDRD)

There are four 16-bit read-only ADDR registers; ADDRA to ADDRD, used to store the results of A/D conversion. The ADDR registers, which store a conversion result for each channel, are shown in table 16.2.

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

The data bus between the CPU and the A/D converter is 8 bits wide. The upper byte can be read directly from the CPU, however the lower byte should be read via a temporary register. The temporary register contents are transferred from the ADDR when the upper byte data is read. When reading the ADDR, read the upper byte before the lower byte, or read in word unit.

The initial value of ADDR is H'0000.

Table 16.2 Analog Input Channels and Corresponding ADDR Registers

Analog Input Channel	A/D Data Register to Be Stored the Results of A/D Conversion
AN0	ADDRA
AN1	ADDRB
AN2, AN14	ADDRC
AN3, AN15	ADDRD

16.3.2 A/D Control/Status Register (ADCSR)

ADCSR controls A/D conversion operations.

Bit	Bit Name	Initial Value	R/W	Description
7	ADF	0	R/(W)*	A/D End Flag A status flag that indicates the end of A/D conversion. [Setting conditions] • When A/D conversion ends • When A/D conversion ends on all channels specified in scan mode [Clearing conditions] • When 0 is written after reading ADF = 1 • When the DMAC or DTC is activated by an ADI interrupt and ADDR is read when DISEL = 0 and the transfer counter ≠ 0

Bit	Bit Name	Initial Value	R/W	Description	
6	ADIE	0	R/W	A/D Interrupt Enable	
				A/D conversion end interrupt (ADI) request enabled when 1 is set.	
5	ADST	0	R/W	A/D Start	
				Clearing this bit to 0 stops A/D conversion, and the A/D converter enters the unit state.	
				Setting this bit to 1 starts A/D conversion. It can be set to 1 by software, the timer conversion start trigger, and the A/D external trigger ($\overline{\text{ADTRG}}$). In single mode, this bit is cleared to 0 automatically when conversion on the specified channel is complete. In scan mode, conversion continues sequentially on the specified channels until this bit is cleared to 0 by software, a reset, a transition to standby mode, or module stop mode.	
4	SCAN	0	R/W	Scan Mode	
				Selects single mode or scan mode as the A/D conversion operating mode.	
				0: Single mode	
				1: Scan mode	
3	CH3	0	R/W	Channel Select 3 to 0	
2	CH2	0	R/W	Select analog input channels.	
1	CH1	0	R/W	When SCAN = 0	When SCAN = 1
0	CH0	0	R/W	0000: AN0	0000: AN0
				0001: AN1	0001: AN0 to AN1
				0010: AN2	0010: AN0 to AN2
				0011: AN3	0011: AN0 to AN3
				01××: Setting prohibited	01××: Setting prohibited
				10××: Setting prohibited	1×××: Setting prohibited
				110×: Setting prohibited	
				1110: AN14	
				1111: AN15	
				Legend: ×: Don't care	

Note: * The write value should always be 0 to clear this flag.

16.3.3 A/D Control Register (ADCR)

The ADCR enables A/D conversion started by an external trigger signal.

Bit	Bit Name	Initial Value	R/W	Description
7	TRGS1	0	R/W	Timer Trigger Select 1 and 0
6	TRGS0	0	R/W	Enables the start of A/D conversion by a trigger signal. Only set bits TRGS1 and TRGS0 while conversion is stopped (ADST = 0). 00: A/D conversion start by software 01: A/D conversion start by TPU 10: A/D conversion start by TMR 11: A/D conversion start by external trigger pin ($\overline{\text{ADTRG}}$)
5, 4	—	All 1	—	Reserved These bits are always read as 1 cannot be modified.
3	CKS1	0	R/W	Clock Select 1 and 0
2	CKS0	0	R/W	These bits specify the A/D conversion time. The conversion time should be changed only when ADST = 0. 00: Conversion time = 530 states (max.) 01: Conversion time = 266 states (max.) 10: Conversion time = 134 states (max.) 11: Conversion time = 68 states (max.) The conversion time setting should exceed the conversion time shown in section 24.6, A/D Converter Characteristics.
1, 0	—	All 1	—	Reserved These bits are always read as 1 cannot be modified.

16.4 Interface to Bus Master

ADDRA to ADDRD are 16-bit registers. As the data bus to the bus master is 8 bits wide, the bus master accesses to the upper byte of the registers directly while to the lower byte of the registers via the temporary register (TEMP).

Data in ADDR is read in the following way: When the upper-byte data is read, the upper-byte data will be transferred to the CPU and the lower-byte data will be transferred to TEMP. Then, when the lower-byte data is read, the lower-byte data will be transferred to the CPU.

When data in ADDR is read, the data should be read from the upper byte and lower byte in the order. When only the upper-byte data is read, the data is guaranteed. However, when only the lower-byte data is read, the data is not guaranteed.

Figure 16.2 shows data flow when accessing to ADDR.

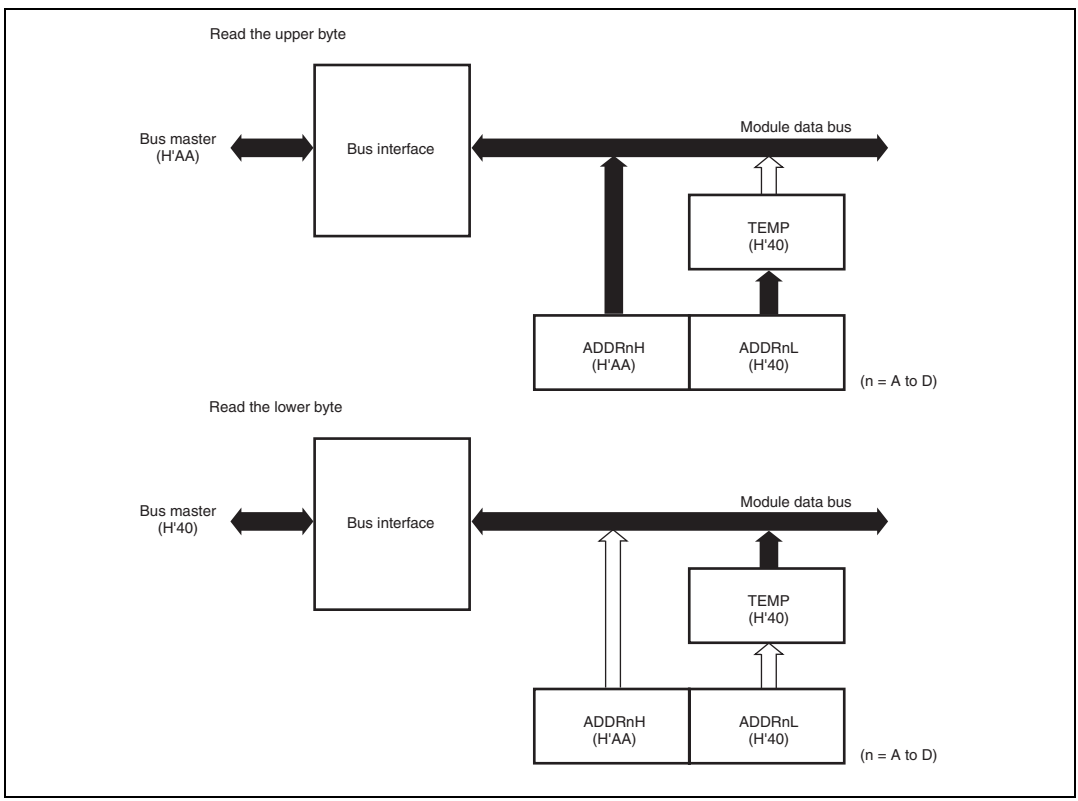


Figure 16.2 Access to ADDR (When Reading H'AA40)

16.5 Operation

The A/D converter operates by successive approximation with 10-bit resolution. It has two operating modes; single mode and scan mode. When changing the operating mode or analog input channel, in order to prevent incorrect operation, first clear the bit ADST to 0 in ADCSR. The ADST bit can be set at the same time as the operating mode or analog input channel is changed.

16.5.1 Single Mode

In single mode, A/D conversion is to be performed only once on the specified single channel. The operations are as follows.

1. A/D conversion is started when the ADST bit is set to 1, according to software, TPU, or external trigger input.
2. When A/D conversion is completed, the result is transferred to the corresponding A/D data register to the channel.
3. On completion of conversion, the ADF bit in ADCSR is set to 1. If the ADIE bit is set to 1 at this time, an ADI interrupt request is generated.
4. The ADST bit remains set to 1 during A/D conversion. When A/D conversion ends, the ADST bit is automatically cleared to 0 and the A/D converter enters the wait state.

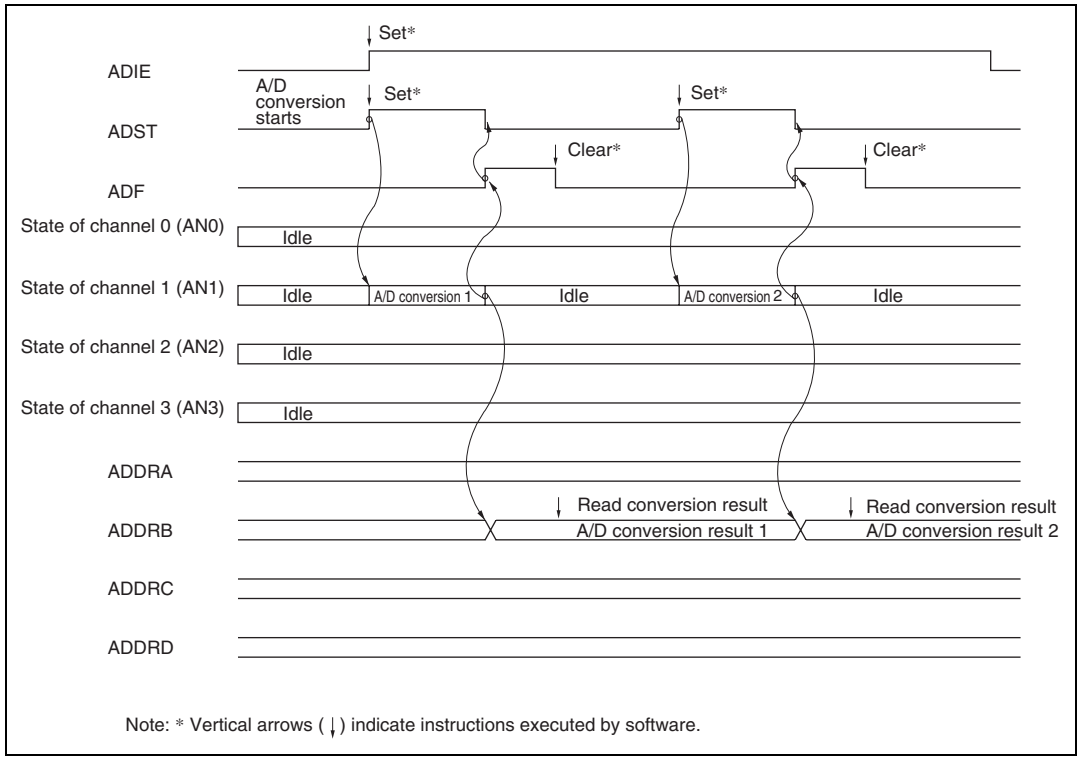
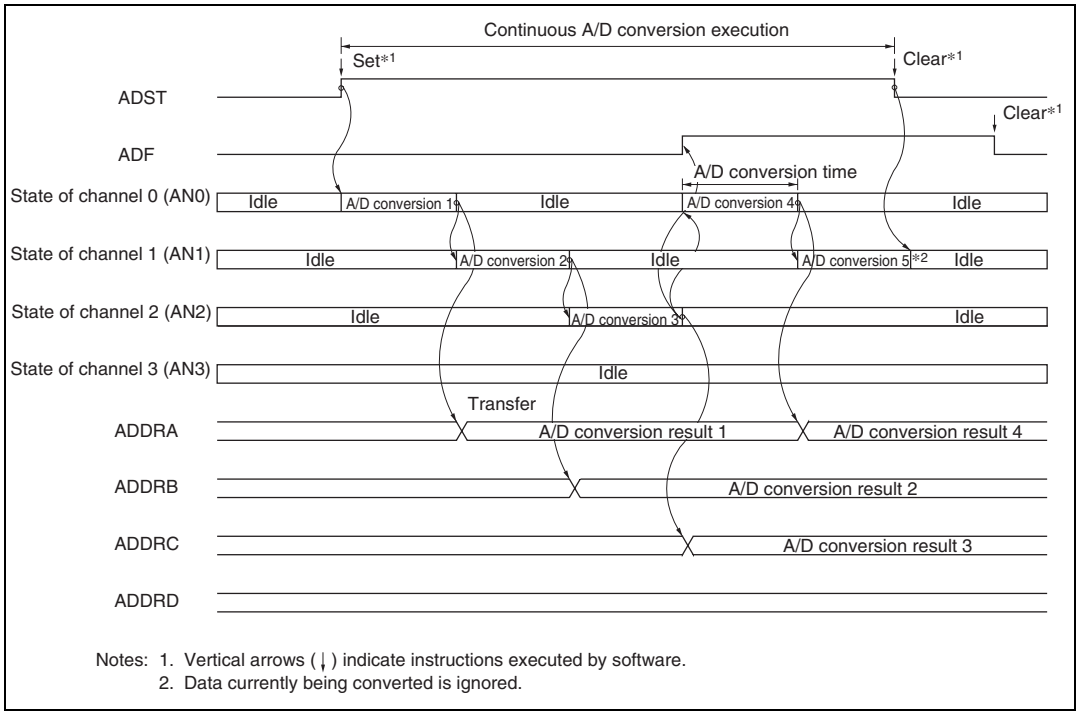


Figure 16.3 A/D Conversion Timing (Single-Chip Mode, Channel 1 Selected)

16.5.2 Scan Mode

In scan mode, A/D conversion is to be performed sequentially on the specified channels (four channels maximum). The operations are as follows.

1. When the ADST bit is set to 1 by software, TPU, or external trigger input, A/D conversion starts on the first channel in the group (AN0 when CH3 and CH2 = 00, AN4 when CH3 and CH2 = 01, or AN8 when CH3 and CH2 = 10).
2. When A/D conversion for each channel is completed, the result is sequentially transferred to the A/D data register corresponding to each channel.
3. When conversion of all the selected channels is completed, the ADF flag is set to 1. If the ADIE bit is set to 1 at this time, an ADI interrupt is requested after A/D conversion ends. Conversion of the first channel in the group starts again.
4. Steps 2 to 3 are repeated as long as the ADST bit remains set to 1. When the ADST bit is cleared to 0, A/D conversion stops.



Notes: 1. Vertical arrows (↓) indicate instructions executed by software.
2. Data currently being converted is ignored.

Figure 16.4 A/D Conversion Timing (Scan Mode, Channels AN0 to AN3 Selected)

16.5.3 Input Sampling and A/D Conversion Time

The A/D converter has a built-in sample-and-hold circuit. The A/D converter samples the analog input when the A/D conversion start delay time (t_D) has passed after the ADST bit is set to 1, then starts conversion. Figure 16.5 shows the A/D conversion timing. Tables 16.3 and 16.4 show the A/D conversion time.

As indicated in figure 16.5, the A/D conversion time (t_{CONV}) includes t_D and the input sampling time (t_{SPL}). The length of t_D varies depending on the timing of 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. The values given in table 16.5 apply to the second and subsequent conversions.

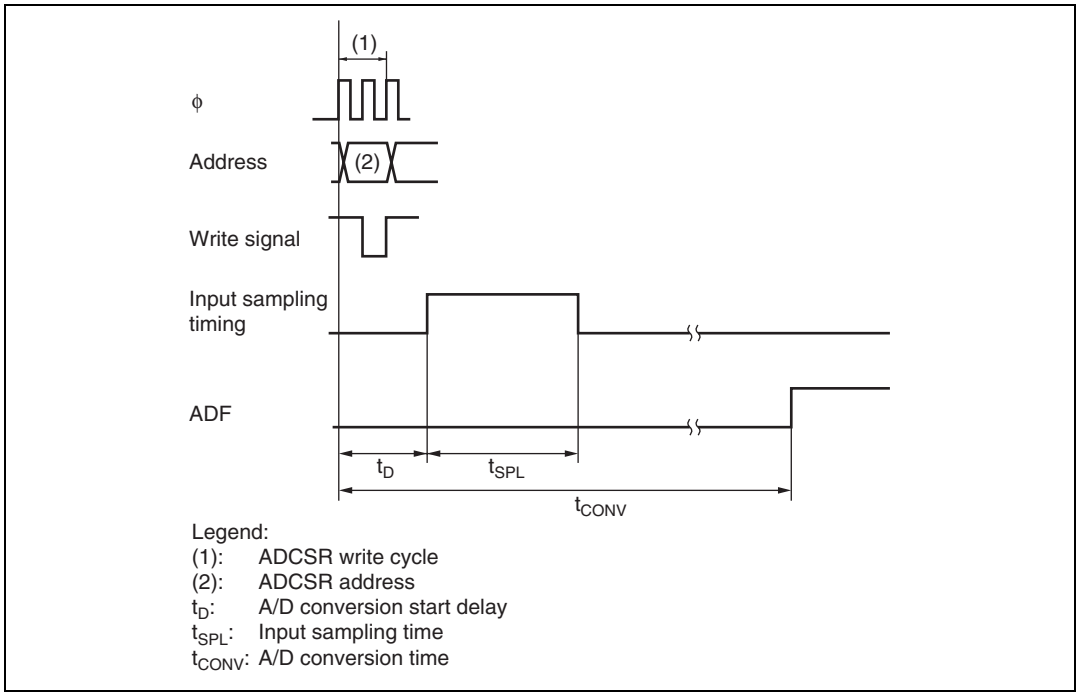


Figure 16.5 A/D Conversion Timing

Table 16.3 A/D Conversion Time (Single Mode)

Item	Symbol	CKS1 = 0						CKS1 = 1					
		CKS0 = 0			CKS0 = 1			CKS0 = 0			CKS0 = 1		
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max
A/D conversion start delay	t_D	18	—	33	10	—	17	6	—	9	4	—	5
Input sampling time	t_{SPL}	—	127	—	—	63	—	—	31	—	—	15	—
A/D conversion time	t_{CONV}	515	—	530	259	—	266	131	—	134	67	—	68

Note: All values represent the number of states.

Table 16.4 A/D Conversion Time (Scan Mode)

CKS1	CKS0	Conversion Time (State)
0	0	512 (Fixed)
	1	256 (Fixed)
1	0	128 (Fixed)
	1	64 (Fixed)

16.5.4 External Trigger Input Timing

A/D conversion can be externally triggered. When the TRGS0 and TRGS1 bits are set to 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 bit ADST has been set to 1 by software. Figure 16.6 shows the timing.

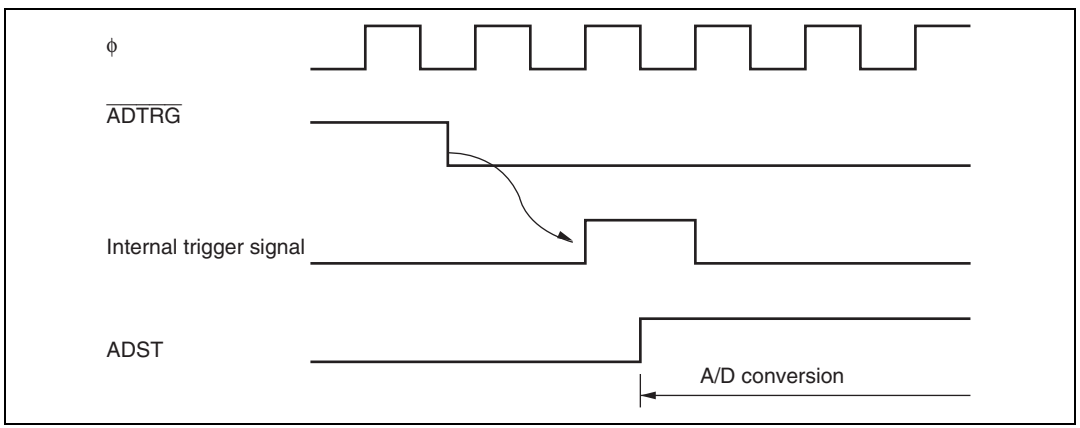


Figure 16.6 External Trigger Input Timing

16.6 Interrupts

The A/D converter generates an A/D conversion end interrupt (ADI) at the end of A/D conversion. Setting the ADIE bit to 1 enables ADI interrupt requests while the bit ADF in ADCSR is set to 1 after A/D conversion is completed. The DMAC or DTC can be activated by an ADI interrupt.

Table 16.5 A/D Converter Interrupt Source

Name	Interrupt Source	Interrupt Source Flag	DMAC or DTC Activation
ADI	A/D conversion completed	ADF	Possible

16.7 A/D Conversion Precision Definitions

This LSI's A/D conversion precision definitions are given below.

- Resolution

  The number of A/D converter digital output codes
- Quantization error

  The deviation inherent in the A/D converter, given by 1/2 LSB (see figure 16.7).
- 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 (H'000) to B'0000000001 (H'001) (see figure 16.8).
- 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 (H'3FE) to B'1111111111 (H'3FF) (see figure 16.8).
- Nonlinearity error

  The error with respect to the ideal A/D conversion characteristic between zero voltage and full-scale voltage. Does not include offset error, full-scale error, or quantization error (see figure 16.8).
- Absolute precision

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

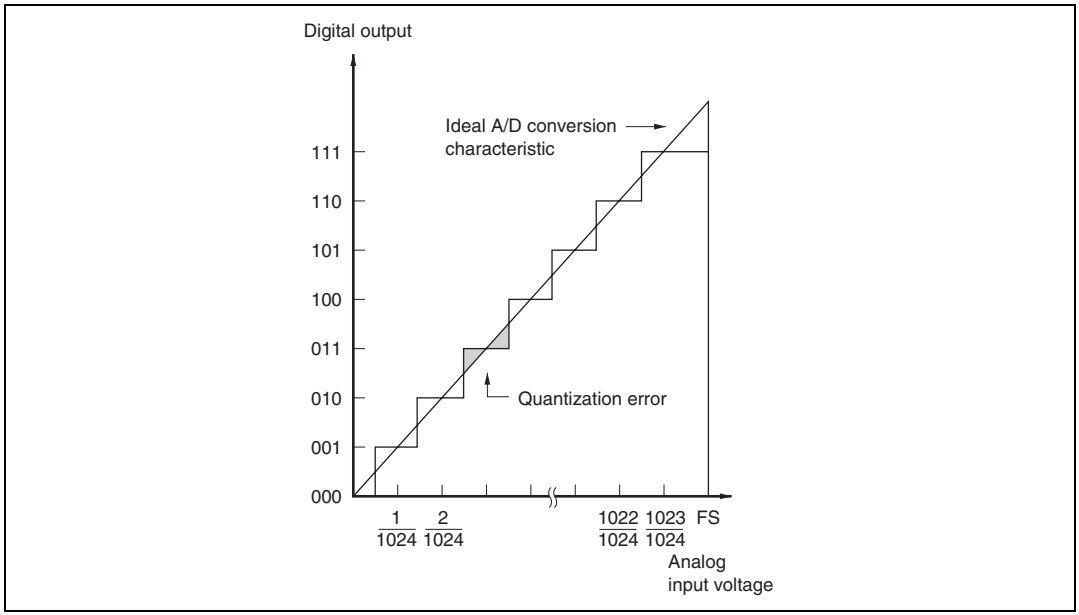


Figure 16.7 A/D Conversion Precision Definitions (1)

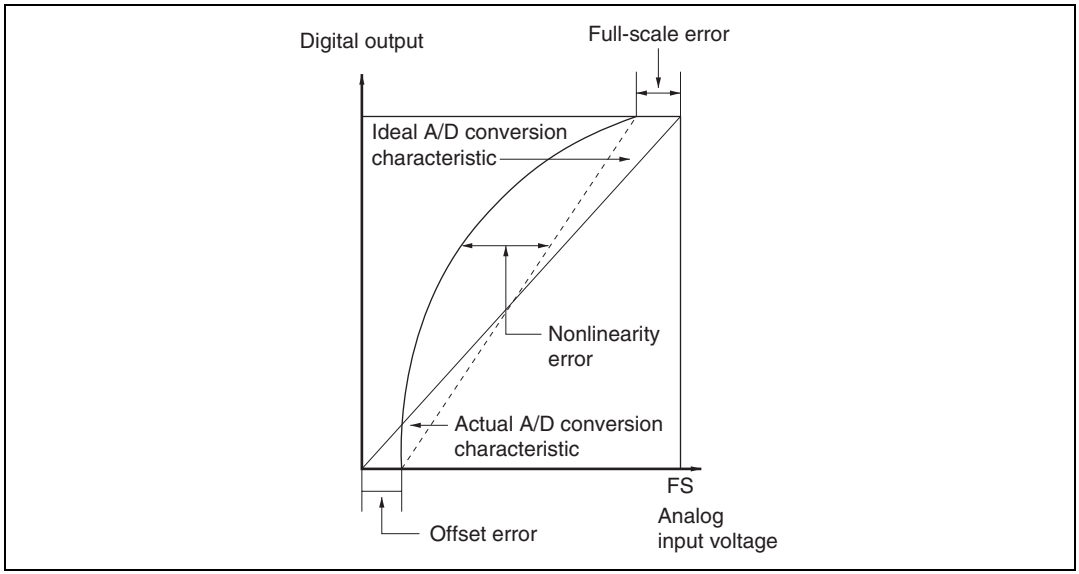


Figure 16.8 A/D Conversion Precision Definitions (2)

16.8 Usage Notes

16.8.1 Permissible Signal Source Impedance

This LSI's analog input is designed such that conversion precision is guaranteed for an input signal for which the signal source impedance is 5 kΩ or less. This specification is provided to enable the A/D converter's sample-and-hold circuit input capacitance to be charged within the sampling time; if the sensor output impedance exceeds 5 kΩ, charging may be insufficient and it may not be possible to guarantee A/D conversion precision. However, for A/D conversion in single mode with a large capacitance provided externally, the input load will essentially comprise only the internal input resistance of 10 kΩ, and the signal source impedance is ignored. However, as a low-pass filter effect is obtained in this case, it may not be possible to follow an analog signal with a large differential coefficient (e.g., 5 mV/Ωs or greater) (see figure 16.9). When converting a high-speed analog signal, a low-impedance buffer should be inserted.

16.8.2 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 AVSS.

Care is also required to insure that filter circuits do not communicate with digital signals on the mounting board (i.e., acting as antennas).

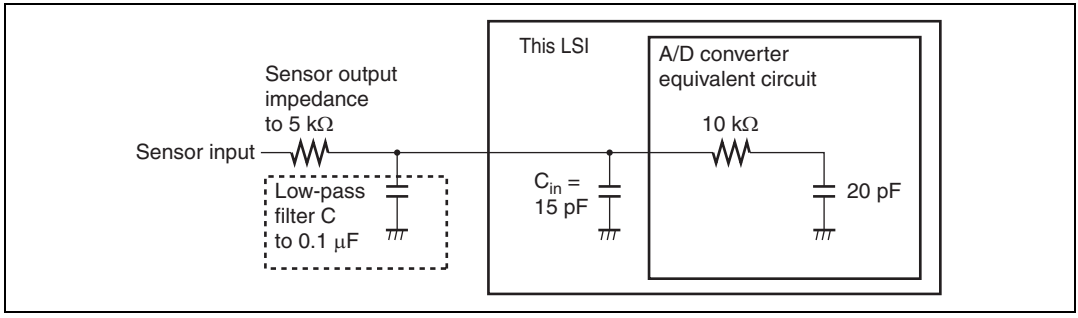


Figure 16.9 Example of Analog Input Circuit

16.8.3 Range of Analog Power Supply and Other Pin Settings

If the conditions below are not met, the reliability of the device may be adversely affected.

- Analog input voltage range
  The voltage applied to analog input pin ANn during A/D conversion should be in the range AVSS ≤ ANn ≤ Vref.
- Relationship between AVcc, AVss and Vcc, Vss
  Set AVss = Vss as the relationship between AVcc, AVss and Vcc, Vss. If the A/D converter is not used, the AVcc and AVss pins must not be left open.
- Vref input range
  The analog reference voltage input at the Vref pin set is the range Vref ≤ AVcc.

16.8.4 Notes on Board Design

In board design, digital circuitry and analog circuitry should be as mutually isolated as possible, and layout in which digital circuit signal lines and analog circuit signal lines cross or are in close proximity should be avoided as far as possible. Failure to do so may result in incorrect operation of the analog circuitry due to inductance, adversely affecting A/D conversion values.

Also, digital circuitry must be isolated from the analog input signals (AN0 to AN3 or AN14 to AN15), analog reference voltage pin (Vref), and analog power supply (AVcc) by the analog ground (AVss). Also, the analog ground (AVss) should be connected at one point to a stable digital ground (Vss) on the board.

16.8.5 Notes on Noise Countermeasures

A protection circuit should be connected in order to prevent damage due to abnormal voltage, such as an excessive surge at the analog input pins (AN0 to AN3 or AN14 to AN15) and analog reference voltage pin (Vref), between AVcc and AVss, as shown in figure 16.10. Also, the bypass capacitors connected to AVcc and the filter capacitor connected to analog input pins (AN0 to AN3 or AN14 to AN15) must be connected to AVss.

If a filter capacitor is connected, the input currents at the analog input pins (AN0 to AN3 or AN14 to AN15) are averaged, and so an error may arise. Also, when A/D conversion is performed frequently, as in scan mode, if the current charged and discharged by the capacitance of the sample-and-hold circuit in the A/D converter exceeds the current input via the input impedance (Rin), an error will arise in the analog input pin voltage. Careful consideration is therefore required when deciding circuit constants.

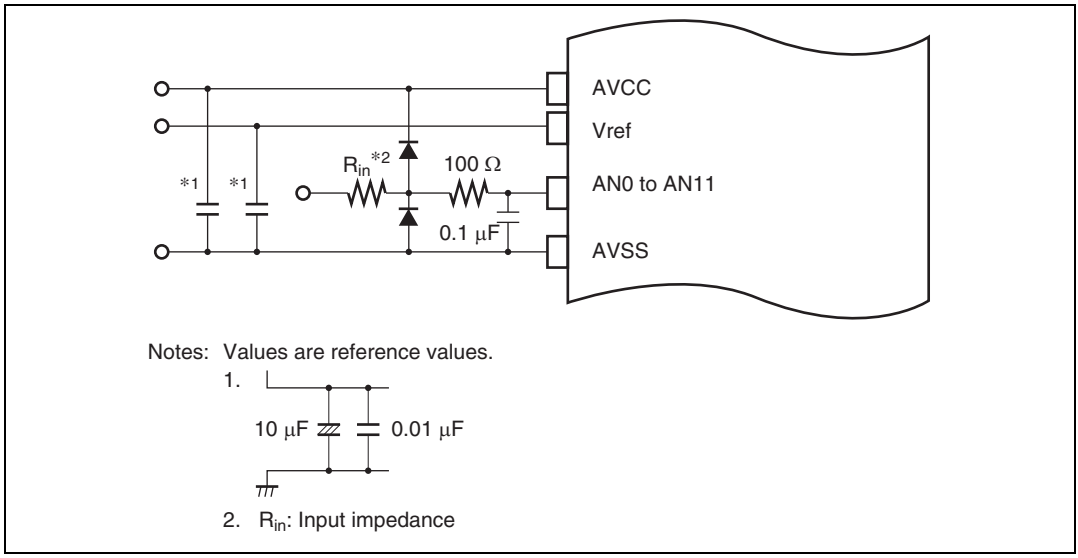


Figure 16.10 Example of Analog Input Protection Circuit

Table 16.6 Analog Pin Specifications

Item	Min	Max	Unit
Analog input capacitance	—	20	pF
Permissible signal source impedance	—	5*	kΩ

Note: * Vcc = 2.7 to 3.6 V

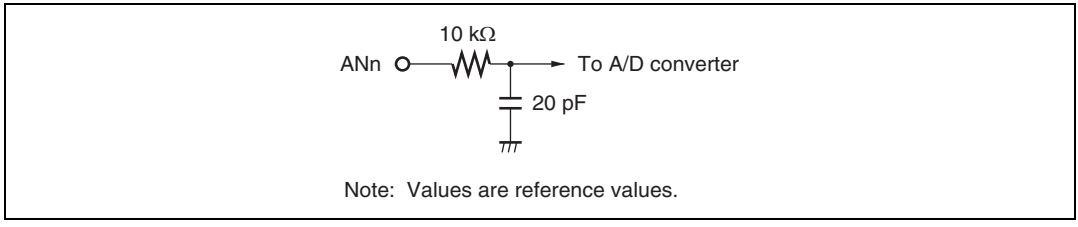


Figure 16.11 Analog Input Pin Equivalent Circuit

16.8.6 Module Stop Mode Setting

Operation of the A/D converter can be disabled or enabled using the module stop control register. The initial setting is for operation of the A/D converter to be halted. Register access is enabled by clearing module stop mode. For details, refer to section 22, Power-Down Modes.

Section 17 D/A Converter

This LSI includes a D/A converter with 2 channels.

17.1 Features

D/A converter features are listed below.

- 8-bit resolution
- Two output channels
- Maximum conversion time of 10 μs (with 20 pF load)
- Output voltage of 0 V to Vref
- D/A output hold function in software standby mode
- Module stop mode can be set

Figure 17.1 shows a block diagram of the D/A converter.

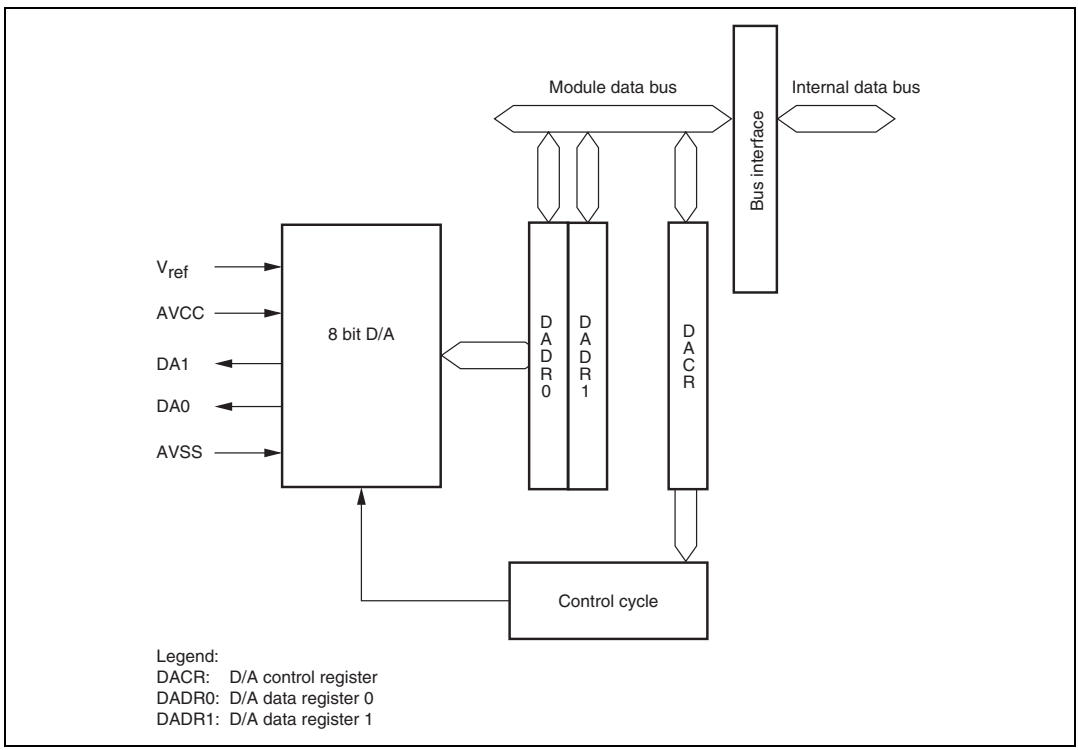


Figure 17.1 Block Diagram of D/A Converter

17.2 Input/Output Pins

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	AVCC	Input	Analog power
Analog ground pin	AVSS	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
Reference voltage pin	Vref	Input	Analog reference voltage

17.3 Register Description

The D/A converter has the following registers.

- D/A data register (DADR)
- D/A control register (DACR)

17.3.1 D/A Data Register (DADR)

DADR is an 8-bit readable/writable register that store data for conversion. Whenever output is enabled, the values in DADR are converted and output from the analog output pins. This register is initialized to H'00 on reset or in hardware standby mode.

17.3.2 D/A Control Register (DACR)

DACR controls the operation of the D/A converter.

DACR01

Bit	Bit Name	Initial Value	R/W	Description
7	DAOE1	0	R/W	D/A Output Enable 1
6	DAOE0	0	R/W	D/A Output Enable 0
5	DAE	0	R/W	D/A Enable Control the D/A conversion and analog output. 00×: Channel 0 and 1 D/A conversions disabled 010: Channel 0 D/A conversion enabled Channel 1 D/A conversion disabled 011: Channel 0 and 1 D/A conversions enabled 100: Channel 0 D/A conversion disabled Channel 1 D/A conversion enabled 101: Channel 0 and 1 D/A conversions enabled 11×: Channel 0 and 1 D/A conversions enabled Legend: ×: Don't care If this LSI 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 the DAOE0, DAOE1, and DAE bits to 0 to disable D/A output.
4 to 0	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.

17.4 Operation

D/A conversion takes place constantly as long as the D/A converter is enabled by the DACR. When DADR_0 and DADR_1 are overwritten, the new data is converted immediately. The conversion result is output by setting the DAOE0 and DAOE1 bits to 1.

The operation example concerns D/A conversion on channel 0. Figure 17.2 shows the timing of this operation.

[1] Write the conversion data to DADR_0.

[2] Set the DAOE0 bit in DACR01 to 1. D/A conversion is started. The conversion result is output after the conversion time t_{DCONV} has elapsed. The output value is expressed by the following formula:

$$\frac{\text{DADR contents}}{256} \times \text{Vref}$$

The conversion results are output continuously until DADR_0 is written to again or the DAOE0 bit is cleared to 0.

[3] If DADR_0 is written to again, the conversion is immediately started. The conversion result is output after the conversion time t_{DCONV} has elapsed.

[4] If the DAOE0 bit is cleared to 0, analog output is disabled.

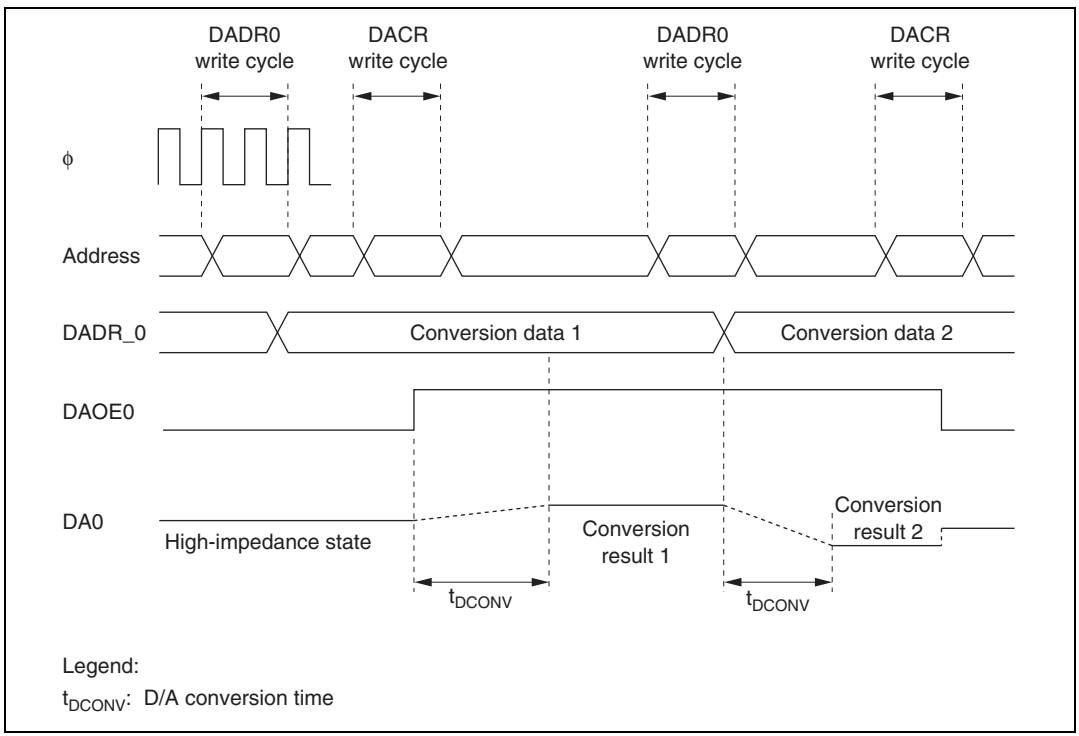


Figure 17.2 Example of D/A Converter Operation

17.5 Usage Note

17.5.1 Module Stop Mode Setting

Operation of the D/A converter can be disabled or enabled using the module stop control register. The initial setting is for operation of the D/A converter to be halted. Register access is enabled by clearing module stop mode. For details, refer to section 22, Power-Down Modes.

Section 18 RAM

This LSI has on-chip high-speed static RAM. The RAM is connected to the CPU by a 16-bit data bus, enabling one-state access by the CPU 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). For details on SYSCR, refer to section 3.2.2, System Control Register (SYSCR).

Product Class		ROM Type	RAM Size	RAM Address
H8S/2215 Group	HD64F2215R	Flash memory Version	20 kbytes	H'FFA000 to H'FFEFBF H'FFFFC0 to H'FFFFFF
	HD64F2215RU			
	HD64F2215T			
	HD64F2215TU			
	HD64F2215		16 kbytes	H'FFB000 to H'FFEFBF H'FFFFC0 to H'FFFFFF
	HD64F2215U			
	HD6432215B	Masked ROM Version		
	HD6432215C		8 kbytes	H'FFD000 to H'FFEFBF H'FFFFC0 to H'FFFFFF

Section 19 Flash Memory (F-ZTAT Version)

The features of the on-chip flash memory are summarized below. The block diagram of the flash memory is shown in figure 19.1.

19.1 Features

- Size

Product Category		ROM Size	ROM Addresses
H8S/2215 Group	HD64F2215, HD64F2215U, HD64F2215R, HD64F2215RU, HD64F2215T, HD64F2215TU	256 kbytes	H'000000 to H'03FFFF (Modes 6 and 7)

- Programming/erase methods
  - The flash memory is programmed 128 bytes at a time. Erase is performed in single-block units. The flash memory is configured as follows: four kbytes × eight blocks, 32 kbytes × 1 block, 64 kbytes × 3 and blocks. To erase the entire flash memory, each block must be erased in turn.
- Reprogramming capability
  - Flash memory can be reprogrammed a minimum of 100 times.
- Two flash memory operating modes
  - Boot mode (SCI boot mode: HD64F2215, HD64F2215R, HD64F2215T. USB boot mode: HD64F2215U, HD64F2215RU, HD64F2215TU)
  - User program mode

    On-board programming/erasing can be done in boot mode in which the boot program built into the chip is started for erase or programming of the entire flash memory. In normal user program mode, individual blocks can be erased or programmed.
- Automatic bit rate adjustment (SCI boot mode)
  - With data transfer in SCI boot mode, this LSI's bit rate can be automatically adjusted to match the transfer bit rate of the host.
- Programming/erasing protection
  - Sets hardware protection, software protection, and error protection against flash memory programming/erasing.
- Programmer mode
  - Flash memory can be programmed/erased in programmer mode, using a PROM programmer, as well as in on-board programming mode.

- Flash memory emulation in RAM
  - — Flash memory programming can be emulated in real time by overlapping a part of RAM onto flash memory.

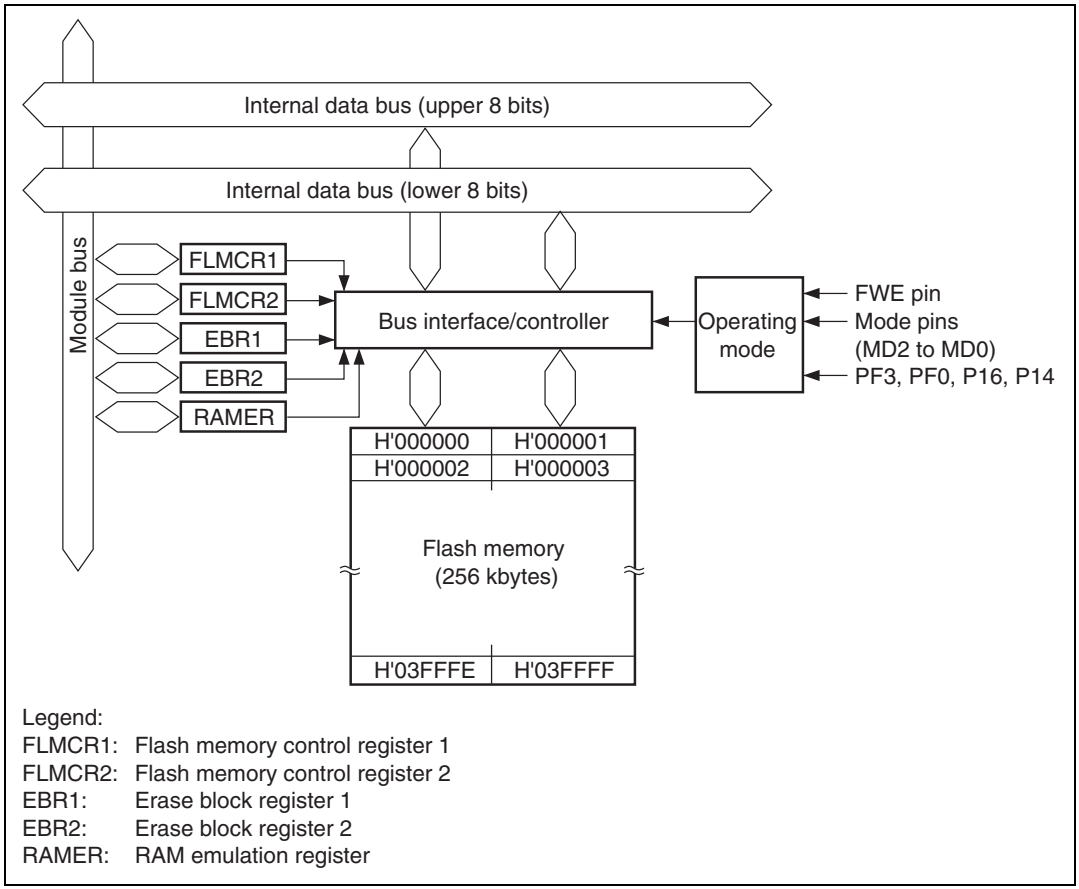
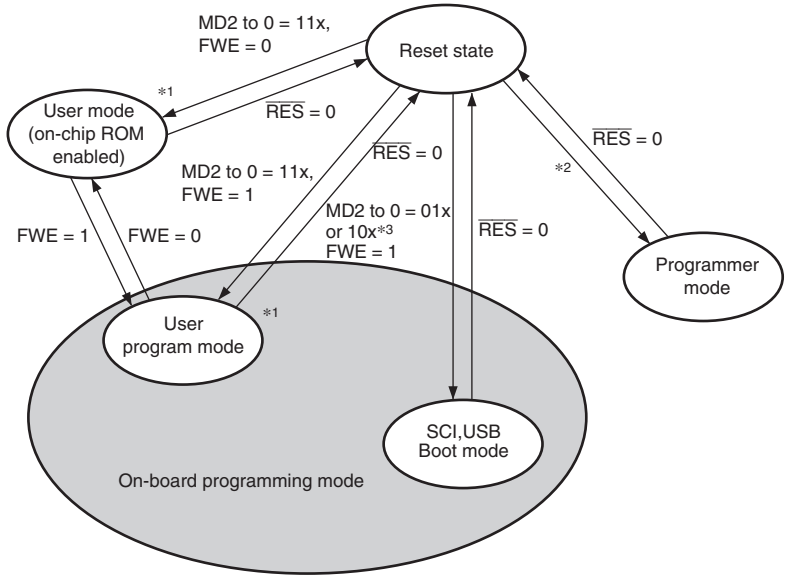


Figure 19.1 Block Diagram of Flash Memory

19.2 Mode Transitions

When the mode pins and the FWE pin are set in the reset state and a reset-start is executed, this LSI enters an operating mode as shown in figure 19.2. In user mode, flash memory can be read but not programmed or erased. The boot and user program modes are provided as modes to write and erase the flash memory.

The differences between boot mode and user program mode are shown in table 19.1. Boot mode and user program mode operations are shown in figures 19.3 and 19.4, respectively.



Notes: Only make a transition between user mode and user program mode when the CPU is not accessing the flash memory.

1. RAM emulation possible
2. MD2 to MD0 = 000, PF3, PF0, P16, P14 = 1100
3. 10x applies only to the HD64F2215RU and HD64F2215TU with 24-MHz system clock.

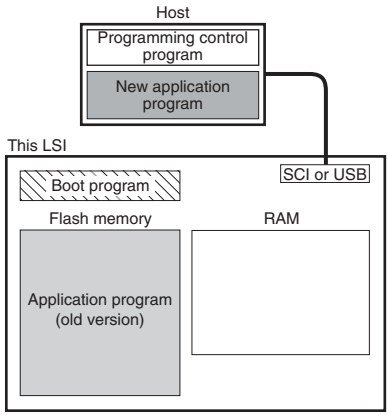
Figure 19.2 Flash Memory State Transitions

Table 19.1 Differences between Boot Mode and User Program Mode

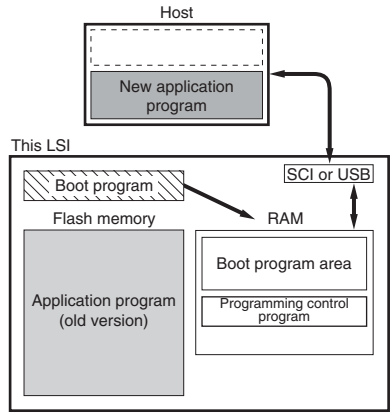
	SCI,USB Boot Mode	User Program Mode	User Mode
Total erase	Yes	Yes	No
Block erase	No	Yes	No
Programming control program*	Program/program-verify	Erase/erase-verify Program/program-verify Emulation	—

Note: * To be provided by the user, in accordance with the recommended algorithm.

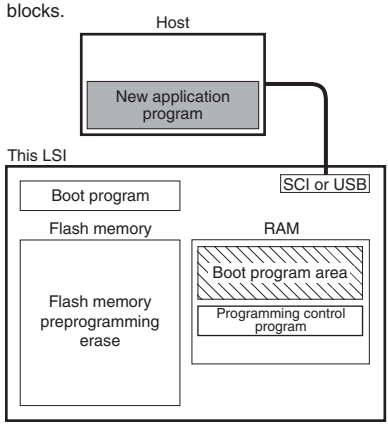
1. Initial state
   The old program version or data remains written in the flash memory. The user should prepare the programming control program and new application program beforehand in the host.



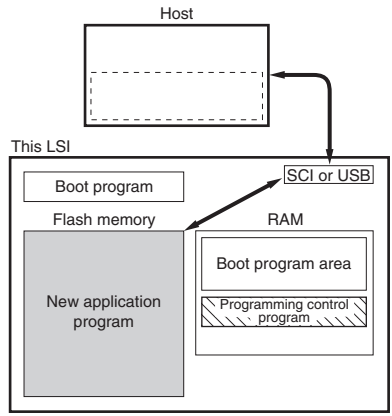
2. Programming control program transfer
   When boot mode is entered, the boot program in this LSI (originally incorporated in the chip) is started and the programming control program in the host is transferred to RAM via SCI or USB communication. The boot program required for flash memory erasing is 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, total flash memory erasure is performed, without regard to blocks.



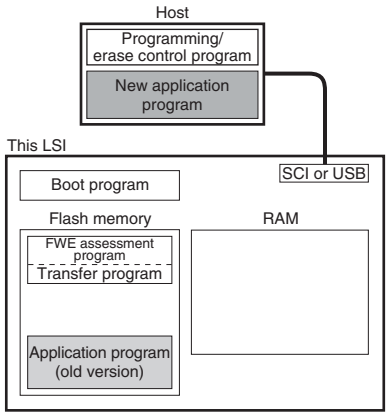
4. Writing new application program
   The programming control program transferred from the host to RAM is executed, and the new application program in the host is written into the flash memory.



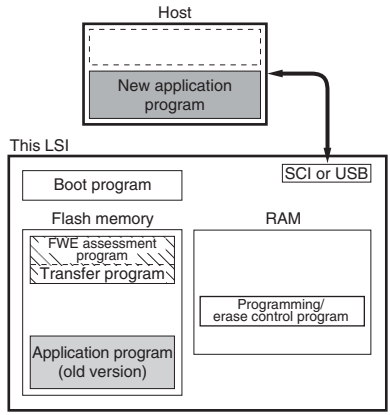
Program execution state

Figure 19.3 Boot Mode (Sample)

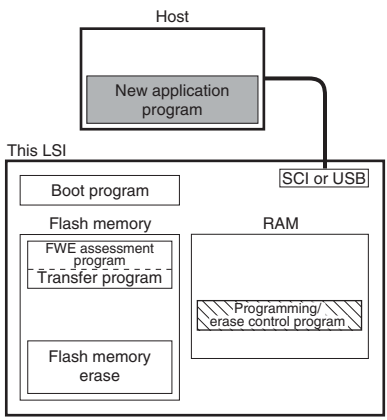
1. Initial state
   The FWE assessment program that confirms that user program mode has been entered, and the program that will transfer the programming/erase control program from flash memory to on-chip RAM should be written into the flash memory by the user beforehand. The programming/erase control program should be prepared in the host or in the flash memory.



2. Programming/erase control program transfer
   When user program mode is entered, user software confirms this fact, executes 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.

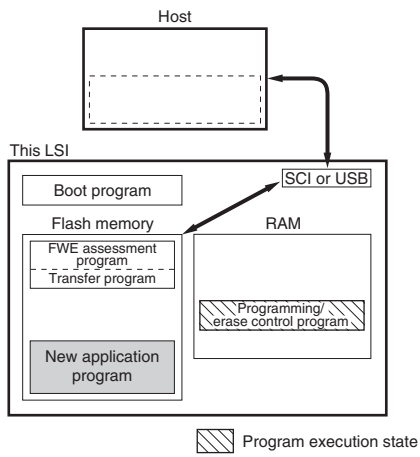


Figure 19.4 User Program Mode (Sample)

19.3 Block Configuration

Figure 19.5 shows the block configuration of 256-kbyte flash memory. The thick lines indicate erasing units, the narrow lines indicate programming units, and the values are addresses. The flash memory is divided into 4 kbytes (eight blocks), 32 kbytes (one block), and 64 kbytes (three blocks). Erasing is performed in these divided units. Programming is performed in 128-byte units starting from an address whose lower eight bits are H'00 or H'80.

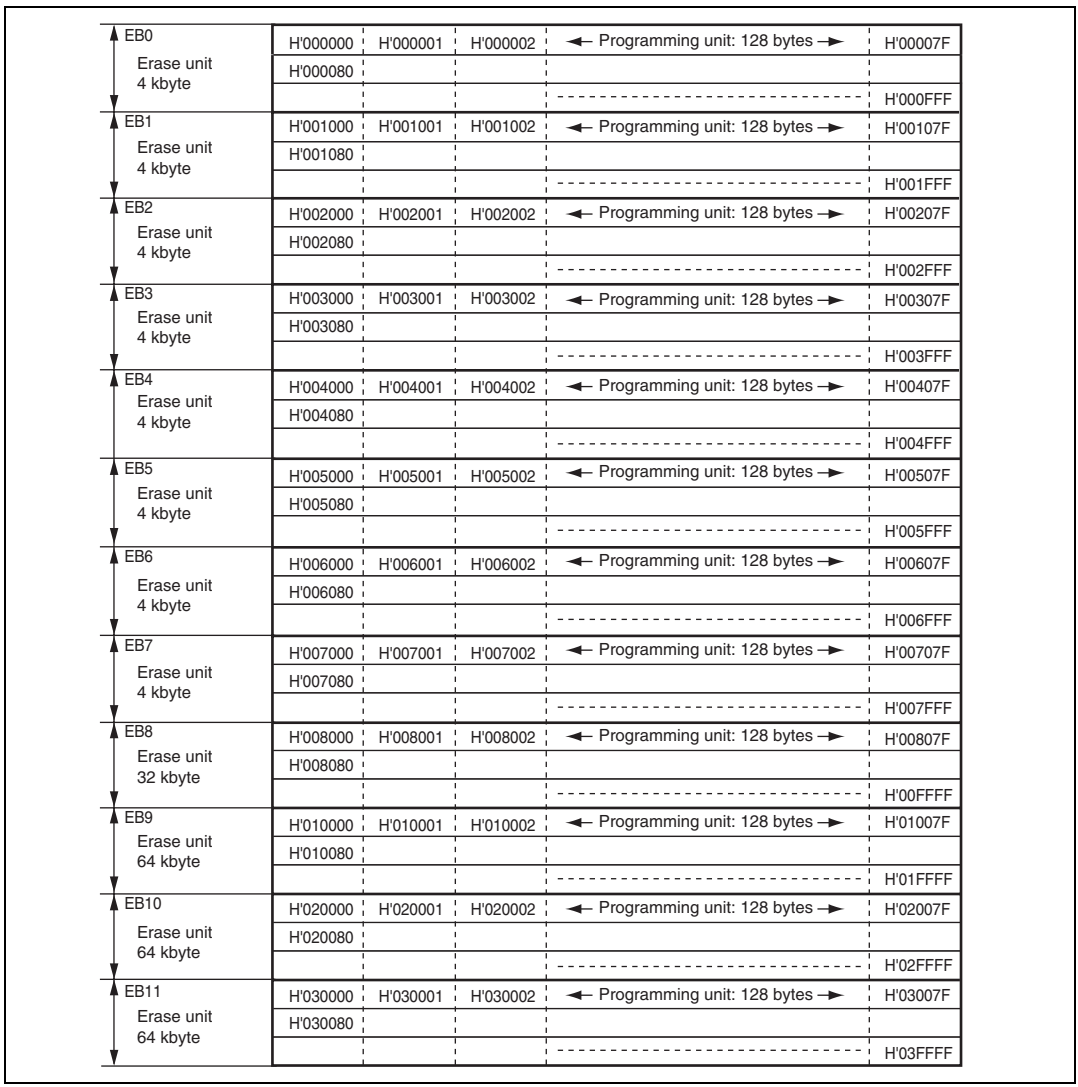


Figure 19.5 Flash Memory Block Configuration

19.4 Input/Output Pins

The flash memory is controlled by means of the pins shown in table 19.2.

Table 19.2 Pin Configuration

Pin Name	I/O	Function	
$\overline{\text{RES}}$	Input	Reset	HD64F2215 and HD64F2215U
FWE	Input	Flash program/erase protection by hardware	
MD2,MD1,MD0	Input	Sets this LSI's operating mode	
PF3,PF0,P16, P14	Input	Sets this LSI's operating mode in programmer mode	
TxD2	Output	Serial transmit data output	HD64F2215
RxD2	Input	Serial receive data input	
USB+,USB-	Input/Output	USB data output	HD64F2215U
VBUS	Input	USB cable connection/disconnection detection	
UBPM	Input	USB bus power mode/self power mode setting	
USPND	Output	USB suspend output	
P36 (PUPD+)	Output	D+ pull-up control	

19.5 Register Descriptions

The flash memory has the following registers. For details on register addresses and register states during each processing, refer to section 23, List of Registers.

- Flash memory control register 1 (FLMCR1)
- Flash memory control register 2 (FLMCR2)
- Erase block register 1 (EBR1)
- Erase block register 2 (EBR2)
- RAM emulation register (RAMER)
- Serial control register X (SCRX)

The above registers are not implemented in the mask ROM version, so attempting to read from them will return undefined values. It is not possible to write to them.

19.5.1 Flash Memory Control Register 1 (FLMCR1)

FLMCR1 is a register that makes the flash memory transit to program mode, program-verify mode, erase mode, or erase-verify mode. For details on register setting, refer to section 19.8, Flash Memory Programming/Erasing.

Bit	Bit Name	Initial Value	R/W	Description
7	FWE	—*	R	Flash Write Enable Reflects the input level at the FWE pin. It is set to 1 when a low level is input to the FWE pin, and cleared to 0 when a high level is input.
6	SWE1	0	R/W	Software Write Enable When this bit is set to 1, flash memory programming/erasing is enabled. When this bit is cleared to 0, other FLMCR1 register bits and all EBR1, EBR2 bits cannot be set. [Setting condition] • When FWE = 1
5	ESU1	0	R/W	Erase Setup When this bit is set to 1, the flash memory transits to the erase setup state. When it is cleared to 0, the erase setup state is cancelled. Set this bit to 1 before setting the E1 bit in FLMCR1. [Setting condition] • When FWE = 1 and SWE1 = 1
4	PSU1	0	R/W	Program Setup When this bit is set to 1, the flash memory transits to the program setup state. When it is cleared to 0, the program setup state is cancelled. Set this bit to 1 before setting the P1 bit in FLMCR1. [Setting condition] • When FWE = 1 and SWE1 = 1
3	EV1	0	R/W	Erase-Verify When this bit is set to 1, the flash memory transits to erase-verify mode. When it is cleared to 0, erase-verify mode is cancelled. [Setting condition] • When FWE = 1 and SWE1 = 1

Bit	Bit Name	Initial Value	R/W	Description
2	PV1	0	R/W	Program-Verify When this bit is set to 1, the flash memory transits to program-verify mode. When it is cleared to 0, program-verify mode is cancelled. [Setting condition] • When FWE = 1 and SWE1 = 1
1	E1	0	R/W	Erase When this bit is set to 1 while the SWE1 and ESU1 bits are 1, the flash memory transits to erase mode. When it is cleared to 0, erase mode is cancelled. [Setting condition] • When FWE = 1, SWE1 = 1, and ESU1 = 1
0	P1	0	R/W	Program When this bit is set to 1 while the SWE1 and PSU1 bits are 1, the flash memory transits to program mode. When it is cleared to 0, program mode is cancelled. [Setting condition] • When FWE = 1, SWE1 = 1, and PSU1 = 1

Note: * Set according to the FWE pin state.

19.5.2 Flash Memory Control Register 2 (FLMCR2)

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

Bit	Bit Name	Initial Value	R/W	Description
7	FLER	0	R	Indicates that an error has occurred during an operation on flash memory (programming or erasing). When FLER is set to 1, flash memory goes to the error-protection state. See section 19.9.3 Error Protection, for details.
6 to 0	—	All 0	—	Reserved These bits always read as 0.

19.5.3 Erase Block Register 1 (EBR1)

EBR1 specifies the flash memory erase area block. EBR1 is initialized to H'00 when the SWE bit in FLMCR is 0. Do not set more than one bit at a time, as this will cause all the bits in EBR1 and EBR2 to be automatically cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
7	EB7	0	R/W	When this bit is set to 1, 4 kbytes of EB7 (H'007000 to H'007FFF) are to be erased.
6	EB6	0	R/W	When this bit is set to 1, 4 kbytes of EB6 (H'006000 to H'006FFF) are to be erased.
5	EB5	0	R/W	When this bit is set to 1, 4 kbytes of EB5 (H'005000 to H'005FFF) are to be erased.
4	EB4	0	R/W	When this bit is set to 1, 4 kbytes of EB4 (H'004000 to H'004FFF) are to be erased.
3	EB3	0	R/W	When this bit is set to 1, 4 kbytes of EB3 (H'003000 to H'003FFF) is to be erased.
2	EB2	0	R/W	When this bit is set to 1, 4 kbytes of EB2 (H'002000 to H'002FFF) is to be erased.
1	EB1	0	R/W	When this bit is set to 1, 4 kbytes of EB1 (H'001000 to H'001FFF) is to be erased.
0	EB0	0	R/W	When this bit is set to 1, 4 kbytes of EB0 (H'000000 to H'000FFF) is to be erased.

19.5.4 Erase Block Register 2 (EBR2)

EBR2 specifies the flash memory erase area block. EBR2 is initialized to H'00 when the SWE bit in FLMCR1 is 0. Do not set more than one bit at a time, as this will cause all the bits in EBR1 and EBR2 to be automatically cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R/W	Reserved The write value should always be 0.
3	EB11	0	R/W	When this bit is set to 1, 64 kbytes of EB11 (H'030000 to H'03FFFF) are to be erased.
2	EB10	0	R/W	When this bit is set to 1, 64 kbytes of EB10 (H'020000 to H'02FFFF) are to be erased.
1	EB9	0	R/W	When this bit is set to 1, 64 kbytes of EB9 (H'010000 to H'01FFFF) are to be erased.
0	EB8	0	R/W	When this bit is set to 1, 32 kbytes of EB8 (H'008000 to H'0'0FFFF) are to be erased.

19.5.5 RAM Emulation Register (RAMER)

RAMER specifies the area of flash memory to be overlapped with part of RAM when emulating real-time flash memory programming. RAMER settings should be made in user mode or user program mode. 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. For details, refer to section 19.7, Flash Memory Emulation in RAM.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	—	Reserved These bits always read as 0.
4	—	0	R/W	Reserved The write value should always be 0.
3	RAMS	0	R/W	RAM Select Specifies selection or non-selection of flash memory emulation in RAM. When RAMS = 1, the flash memory is overlapped with part of RAM, and all flash memory block are program/erase-protected.
2	RAM2	0	R/W	Flash Memory Area Selection
1	RAM1	0	R/W	When the RAMS bit is set to 1, selects one of the following flash memory areas to overlap the RAM area. The areas correspond with 4-kbyte erase blocks.
0	RAM0	0	R/W	000: H'000000 to H'000FFF (EB0) 001: H'001000 to H'001FFF (EB1) 010: H'002000 to H'002FFF (EB2) 011: H'003000 to H'003FFF (EB3) 100: H'004000 to H'004FFF (EB4) 101: H'005000 to H'005FFF (EB5) 110: H'006000 to H'006FFF (EB6) 111: H'007000 to H'007FFF (EB7)

19.5.6 Serial Control Register X (SCRX)

SCRX performs register access control.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R/W	Reserved The write value should always be 0.
3	FLSHE	0	R/W	Flash Memory Control Register Enable Controls CPU access to the flash memory control registers (FLMCR1, FLMCR2, EBR1, and EBR2). Setting the FLSHE bit to 1 enables read/write access to the flash memory control registers. If FLSHE is cleared to 0, the flash memory control registers are deselected. In this case, the flash memory control register contents are retained. 0: Flash control registers deselected in area H'FFFFA8 to H'FFFFAC 1: Flash control registers selected in area H'FFFFA8 to H'FFFFAC
2 to 0	—	All 0	R/W	Reserved The write value should always be 0.

19.6 On-Board Programming Modes

When pins are set to on-board programming mode and a reset-start is executed, a transition is made to the on-board programming state in which 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 19.3. For a diagram of the transitions to the various flash memory modes, see figure 19.2.

Table 19.3 Setting On-Board Programming Modes

Mode		FWE	MD2	MD1	MD0
SCI boot mode (HD64F2215, HD64F2215R, HD64F2215T)	Advanced: On-chip ROM extended mode	1	0	1	0
	Advanced: Single-chip mode	1	0	1	1
USB boot mode (HD64F2215U, HD64F2215RU, HD64F2215TU)*[1]	Advanced: On-chip ROM extended mode	1	0	1	0
	Advanced: Single-chip mode	1	0	1	1
USB boot mode (HD64F2215RU, HD64F2215TU)*[2]	Advanced: On-chip ROM extended mode	1	1	0	0
	Advanced: Single-chip mode	1	1	0	1
User program mode	Advanced: On-chip ROM extended mode (MCU operating mode 6)	1	1	1	0
	Advanced: Single-chip mode (MCU operating mode 7)	1	1	1	1

Notes: 1. When the system clock is 16 MHz.
2. When the system clock is 24 MHz.

19.6.1 SCI Boot Mode (HD64F2215, HD64F2215R, and HD64F2215T)

When a reset-start is executed after the LSI's pins have been set to boot mode, the boot program built into the LSI is started and the programming control program prepared in the host is serially transmitted to the LSI via the SCI. In the LSI, 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 system configuration in SCI boot mode is shown in figure 19.6.

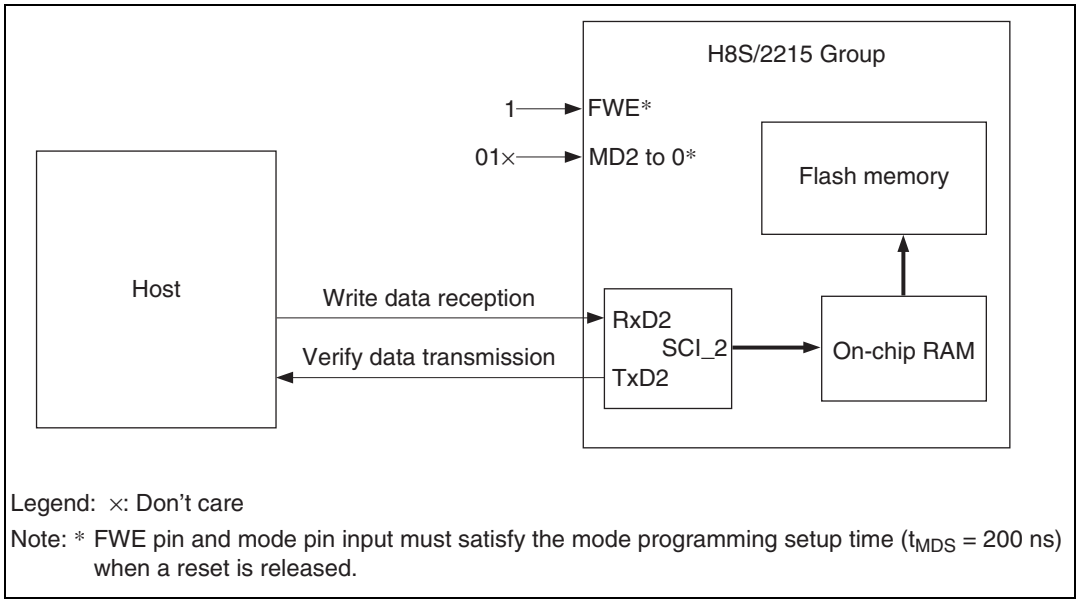


Figure 19.6 System Configuration in SCI Boot Mode

Table 19.4 shows the boot mode operations between reset end and branching to the programming control program.

1. When boot mode is used, the flash memory programming control program must be prepared in the host beforehand. Prepare a programming control program in accordance with the description in section 19.8, Flash Memory Programming/Erasing. 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 in enforced exit 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.
2. The SCI_2 should be set to asynchronous mode, and the transfer format as follows: 8-bit data, 1 stop bit, and no parity.
3. When the boot program is initiated, the chip measures the low-level period of asynchronous SCI communication data (H'00) transmitted continuously from the host. The chip then calculates the bit rate of transmission from the host, and adjusts the SCI_2 bit rate to match that of the host. The reset should end with the RxD pin high. The RxD and TxD pins should be pulled up on the board if necessary. After the reset ends, it takes approximately 100 states before the chip is ready to measure the low-level period.
4. After matching the bit rates, the chip transmits one H'00 byte to the host to indicate the 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 could not be

performed normally, initiate boot mode again by a reset. Depending on the host's transfer bit rate and system clock frequency of this LSI, there will be a discrepancy between the bit rates of the host and the chip. To operate the SCI properly, set the host's transfer bit rate and system clock frequency of this LSI within the ranges listed in table 19.5.

5. In boot mode, a part of the on-chip RAM area (4 kbytes) is used by the boot program. Addresses H'FFE000 to H'FFEFBF is the area to which the programming control program is transferred from the host. The boot program area cannot be used until the execution state in boot mode switches to the programming control program.
6. Before branching to the programming control program, the chip terminates transfer operations by the SCI_2 (by clearing the RE and TE bits in SCR to 0), but the adjusted bit rate value remains set in BRR. Therefore, the programming control program can still use it for transfer of write data or verify data with the host. The TxD pin is high. The contents of the CPU general registers are undefined immediately after branching to the programming control program. These registers must be initialized at the beginning of the programming control program, since the stack pointer (SP), in particular, is used implicitly in subroutine calls, etc.
7. Boot mode can be cleared by a reset. End the reset* after driving the reset pin low, waiting at least 20 states, and then setting the FWE pin and the mode (MD) pins. Boot mode is also cleared when a WDT overflow occurs.
8. Do not change the MD 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 ($\overline{AS}$, $\overline{RD}$, $\overline{WR}$) will change according to the change in the microcomputer's operating mode . 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.
9. All interrupts are disabled during programming or erasing of the flash memory.

Note: * Mode pin and FWE pin input must satisfy the mode programming setup time (t_{MDS} = 200 ns) with respect to the reset release timing.

Table 19.4 SCI Boot Mode Operation

Item	Host Operation	LSI Operation
		Branches to boot program at reset-start.
Bit rate adjustment	Continuously transmits data H'00 at specified bit rate.	Measures low-level period of receive data H'00.
		Calculates bit rate and sets it in BRR of SCI_2.
	Transmits data H'55 when data H'00 is received error-free.	Transmits data H'00 to host as adjustment end indication.
		Transmits data H'AA to host when data H'55 is received.
Transmits number of bytes (N) of programming control program	Transmits number of bytes (N) of programming control program to be transferred as 2-byte data (low-order byte following high-order byte)	Echobacks the 2-byte data received as verification data.
Transmits 1-byte of programming control program (repeated for N times)	Transmits 1-byte of programming control program	Echobacks received data to host and also transfers it to RAM
Flash memory erase		Checks flash memory data, erases all flash memory blocks in case of written data existing, and transmits data H'AA to host. (If erase could not be done, transmits data H'FF to host and aborts operation.)
Programming control program execution		Branches to programming control program transferred to on-chip RAM and starts execution.

Table 19.5 System Clock Frequencies for Which Automatic Adjustment of LSI Bit Rate Is Possible

Host Bit Rate	System Clock Frequency Range of LSI
19,200 bps	HD64F2215: 13 to 16 MHz
9,600 bps	HD64F2215R: 13 to 24 MHz
4,800 bps	HD64F2215T: 16 MHz and 24 MHz

19.6.2 USB Boot Mode (HD64F2215U, HD64F2215RU, and HD64F2215TU)

- Features
  - Selection of bus-powered mode or self-powered mode
  - HD64F2215U: Supports only 16-MHz system clock, with USB operating clock generation by means of PLL3 multiplication

    HD64F2215RU and HD64F2215TU: Supports either 16-MHz or 24-MHz system clock, with USB operating clock generation by means of PLL2 or PLL3 multiplication, respectively.
  - D+ pull-up control connection supported for P36 pin only
  - See table 19.6 for enumeration information

Table 19.6 Enumeration Information

USB standard	Ver.1.1	
Transfer modes	Control (in, out), Bulk (in, out)	
Maximum power	Self power mode ($\overline{\text{UBPM}}$ pin = 1)	100 mA
	Bus power mode ($\overline{\text{UBPM}}$ pin = 0)	500 mA
Endpoint configuration	EP0 Control (in, out) 64 bytes Configuration 1 └ Interface Number 0 └ Alternate Setting 0 ├ EP1 Bulk (out) 64 bytes └ EP2 Bulk (in) 64 bytes	

- Notes on USB Boot Mode Execution
  - With the HD64F2215, use a 16-MHz system clock and an external circuit configuration that allows use of a PLL. With the HD64F2215RU or HD64F2215TU, use a 16-MHz or 24-MHz system clock and an external circuit configuration that allows use of a PLL. USB boot mode execution is not possible with other combinations.
  - Use the P36 pin for D+ pull-up control connection.
  - To ensure stable power supply during flash memory programming/erasing, do not use cable connection via a bus powered HUB.
  - Note in particular that, in the worst case, the LSI may be permanently damaged if the USB cable is detached during flash memory programming/erasing.
  - A transition is not made to software standby mode (a power-down mode) even if the USB bus enters suspend mode when in bus power mode.

- Overview

  When a reset start is performed after the pins of this LSI have been set to boot mode, a boot program incorporated in the microcomputer beforehand is activated, and the prepared programming control program is transmitted sequentially to the host using the USB. With this LSI, the programming control program received by the USB is written to a programming control program area in on-chip RAM. After transfer is completed, control branches to the start address of the programming control program area, and the programming control program execution state is established (flash memory programming is performed). Figure 19.7 shows a system configuration diagram when using USB boot mode.

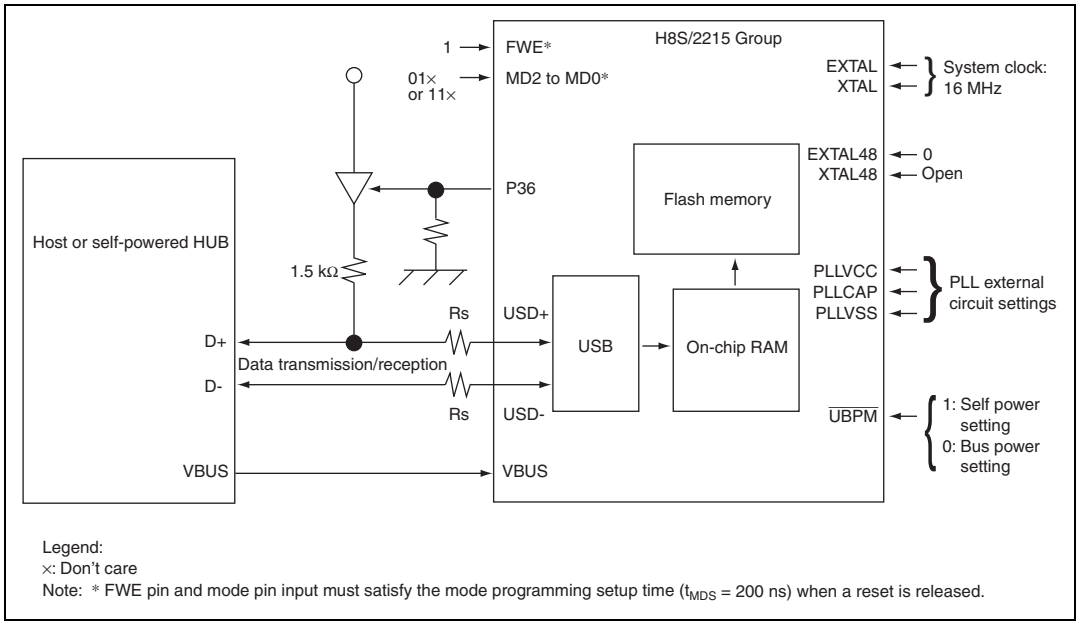


Figure 19.7 System Configuration Diagram when Using USB Boot Mode

Table 19.7 shows operations from reset release in USB boot mode until processing branches to the programming control program.

1. When boot mode is used, the flash memory programming control program must be prepared in the host beforehand. Prepare a programming control program in accordance with the description in section 19.8, Flash Memory Programming/Erasing. 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 in enforced exit when user program mode is unavailable, such a the first time on-board programming is performed, or if the program activated in user program mode is accidentally erased.

2. When the boot program is activated, enumeration with respect to the host is carried out. Enumeration information is shown in table 19.6. When enumeration is completed, transmit a single H'55 byte from the host. If reception has not been performed normally, restart boot mode by means of a reset.

3. Set the frequency for transmission from the host as a numeric value in units of MHz × 100 (ex: 16.00 MHZ → H'0640).

4. In boot mode, the 4-kbyte on-chip RAM area H'FFE000 to H'FFEFBF is used by the boot program. The programming control program transmitted from the host can be stored in the 8-kbyte area H'FFC000 to H'FFDFFF. The boot program area cannot be used until program execution switches to the programming control program. Also note that the boot program remains in RAM even after control passes to the programming control program.

5. When a branch is made to the programming control program, the USB remains connected and can be used immediately for transmission/reception of write data or verify data between the programming control program and the host. The contents of CPU general registers are undefined after a branch to the programming control program. Note, in particular, that since the stack pointer is used implicitly in subroutine calls and the like, it should be initialized at the start of the programming control program.

6. Boot mode is exited by means of a reset. Drive the reset pin low, wait for the elapse of at least 20 states, then set the FWE pin and mode pins to release* the reset. Boot mode is also exited in the event of a WDT overflow reset.

7. Do not change the input level of the mode pins while in boot mode. If the input level of a mode pin is changed (from low to high) during a reset, the states of ports with a dual function as address outputs, and bus control output signals ($\overline{AS}$, $\overline{RD}$, $\overline{WR}$), will change due to switching of the operating mode. Either make pin settings so that these pins do not become output signal pins during a reset, or take precautions to prevent collisions with external signals.

8. Interrupts cannot be used during flash memory programming or erasing.

Note: * FWE pin and mode pin input must satisfy the mode programming setup time (t_{MDS} = 200 ns) when a reset is released.

Table 19.7 USB Boot Mode Operation

Item	Host Operation	Operation of this LSI
		Branches to boot program after reset start
Start of USB boot mode	Transmits one H'55 byte on completion of USB enumeration	Transmits one H'AA byte to host on reception of H'55
Transfer clock information	Transmits frequency (2 bytes), number of multiplication classifications (1 byte), multiplication ratio (1 byte) With this LSI, H'0640, H'01, H'01 are transmitted	If received data are within respective ranges, transmits H'AA to host If any received data is out-of-range, transmits H'FF to host and halts operation
Transfer number of bytes (N) of programming control program	Performs 2-byte transfer of number of bytes (N) of programming control program	If received number of bytes is within range, transmits H'AA to host If received number of bytes is out-of-range, transmits H'FF to host and halts operation
Transfer of programming control program and sum value	Transmits programming control program in N-byte divisions. Transmits sum value (two's complement of sum total of programming control program (1 byte))	Transfers received data to on-chip RAM Calculates sum total of received sum value and 1-byte units of programming control program transferred to on-chip RAM If sum is 0, transmits H'AA to host If sum is not 0, transmits H'FF to host and halts operation
Memory erase	Transmits total erase status command (H'3A)	Starts total erase of flash memory Transmits H'11 to host if total erase processing is being executed when total erase status command is received Transmits H'06 to host if total erase of all blocks has been completed when total erase status command is received
	Retransmits total erase status command (H'3A) when H'11 is received	If erase cannot be performed when total erase status command is received, transmits H'EE to host and halts operation
Execution of programming control program		Branches to programming control program transferred to on-chip RAM and starts execution

19.6.3 Programming/Erasing in User Program Mode

On-board programming/erasing of an individual flash memory block can also be performed in user program mode by branching to a user program/erase control program. The user must set branching conditions and provide on-board FWE control and supply of programming data, and storing a program/erase control program in part of the program area as necessary. The flash memory must contain the user program/erase control program or a program which provides the user program/erase control program from external memory. Because the flash memory itself cannot be read during programming/erasing, transfer the user program/erase control program to on-chip RAM, as like in boot mode. Figure 19.8 shows a sample procedure for programming/erasing in user program mode. Prepare a user program/erase control program in accordance with the description in section 19.8, Flash Memory Programming/Erasing.

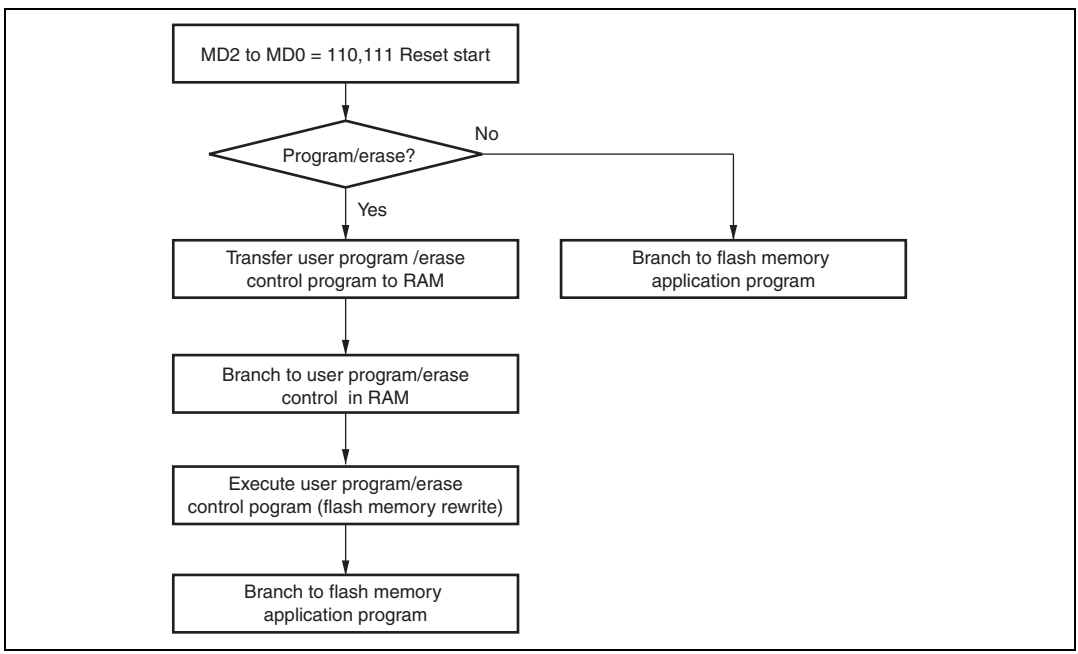


Figure 19.8 Programming/Erasing Flowchart Example in User Program Mode

19.7 Flash Memory 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. Emulation can be performed in user mode or user program mode. Figure 19.9 shows an example of emulation of real-time flash memory programming.

1. Set RAMER to overlap part of RAM onto the area for which real-time programming is required.
2. Emulation is performed using the overlapping RAM.
3. After the program data has been confirmed, the RAMS bit is cleared, thus releasing RAM overlap.
4. The data written in the overlapping RAM is written into the flash memory space (EB0).

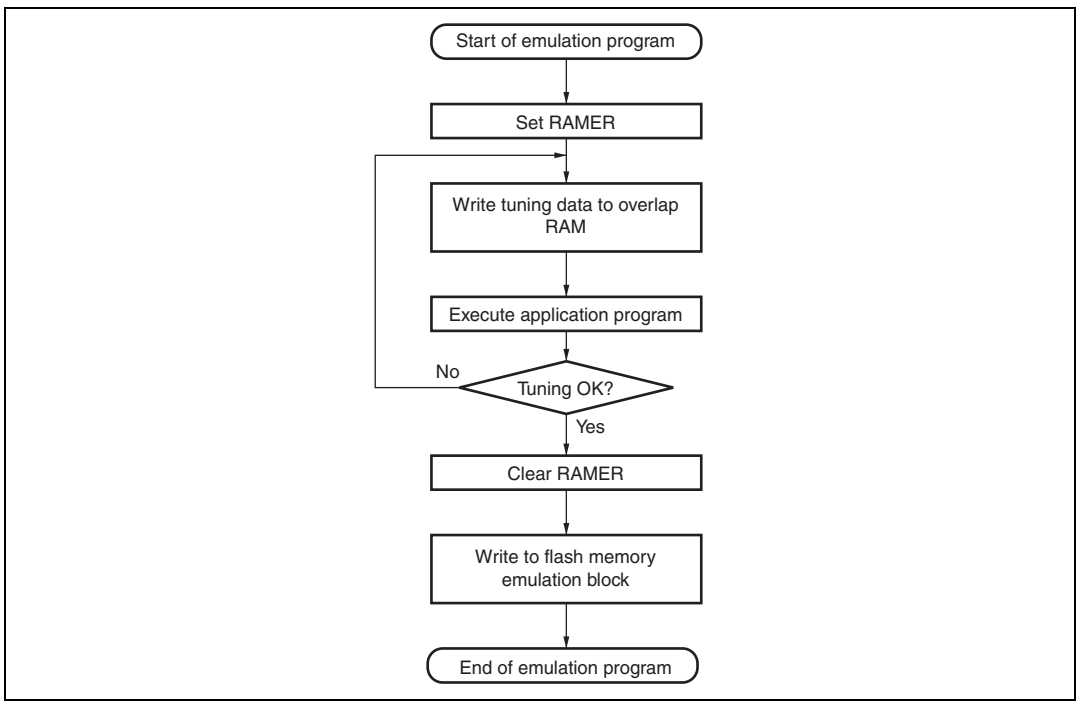


Figure 19.9 Flowchart for Flash Memory Emulation in RAM

Example in which flash memory block area EB0 is overlapped is shown in figure 19.10.

1. The RAM area to be overlapped is fixed at a 4-kbyte area in the range of H'FFD000 to H'FFDFFF.
2. The flash memory area to overlap is selected by RAMER from a 4-kbyte area among one of the EB0 to EB7 blocks.
3. The overlapped RAM area can be accessed from both the flash memory addresses and RAM addresses.
4. When the RAMS bit in RAMER is set to 1, program/erase protection is enabled for all flash memory blocks (emulation protection). In this state, setting the P1 or E1 bit in FLMCR1 to 1 does not cause a transition to program mode or erase mode.
5. A RAM area cannot be erased by execution of software in accordance with the erase algorithm.
6. Block area EB0 contains the vector table. When performing RAM emulation, the vector table is needed in the overlap RAM.

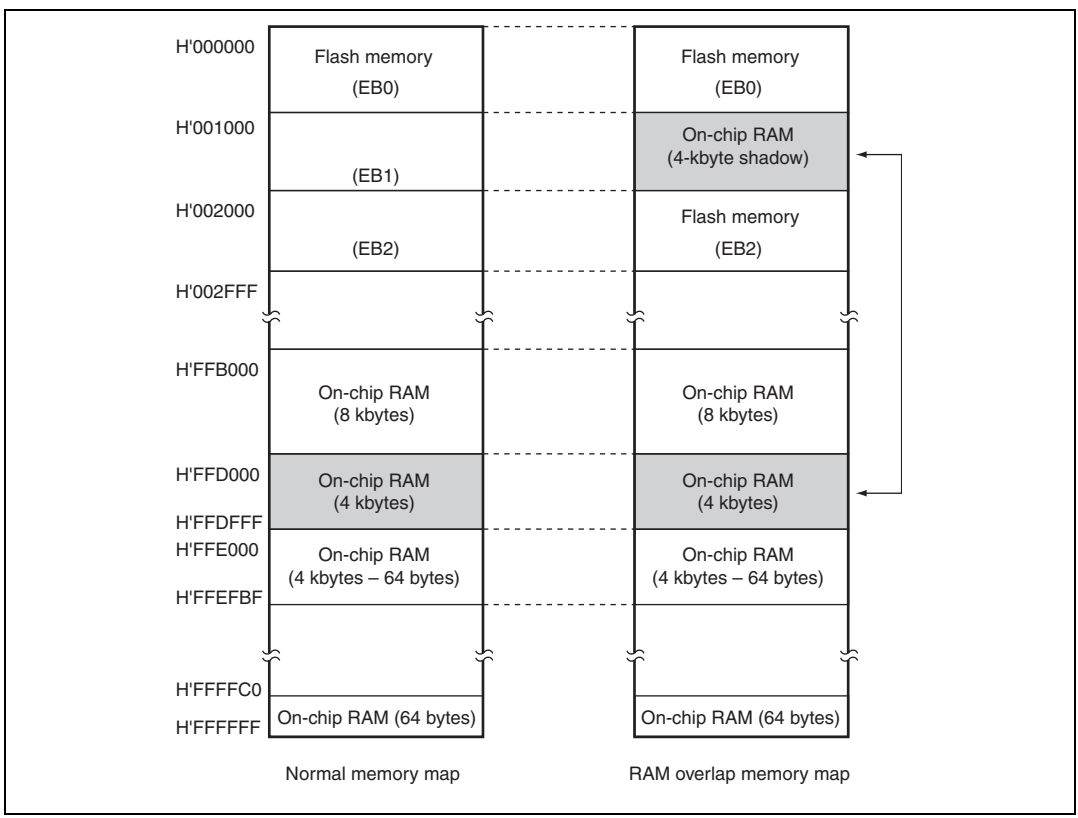


Figure 19.10 Example of RAM Overlap Operation

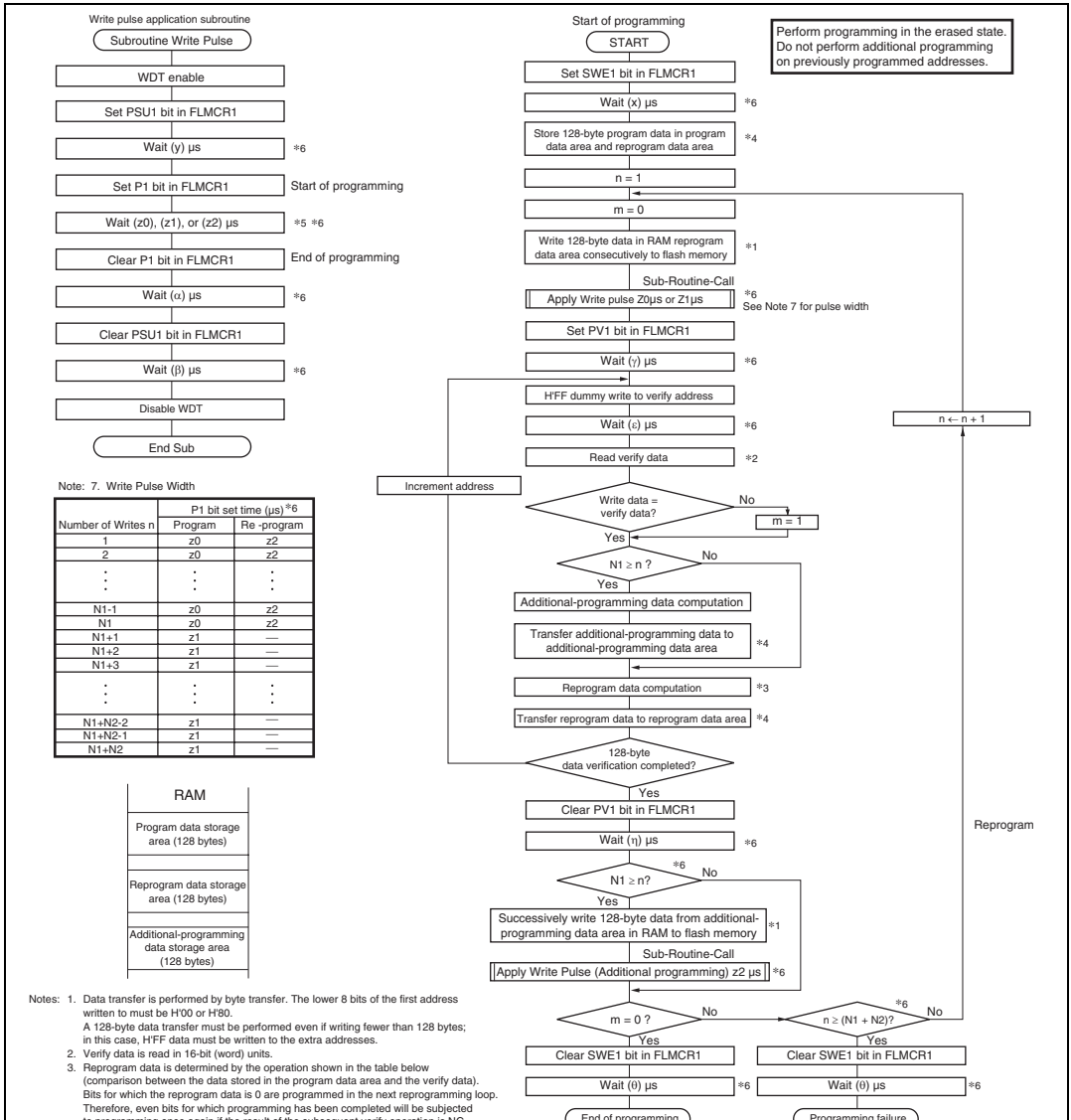
19.8 Flash Memory Programming/Erasing

A software method, using the CPU, is employed to program and erase flash memory in the on-board programming modes. Depending on the FLMCR1 setting, the flash memory operates in one of the following four modes: program mode, erase mode, program-verify mode, and erase-verify mode. The programming control program in boot mode and the user program/erase control program in user program mode use these operating modes in combination to perform programming/erasing. Flash memory programming and erasing should be performed in accordance with the descriptions in section 19.8.1, Program/Program-Verify and section 19.8.2, Erase/Erase-Verify, respectively.

19.8.1 Program/Program-Verify

When writing data or programs to the flash memory, the program/program-verify flowchart shown in figure 19.11 should be followed. Performing programming operations according to this flowchart will enable data or programs to be written to the flash memory without subjecting the chip to voltage stress or sacrificing program data reliability.

1. Programming must be done to an empty address. Do not reprogram an address to which programming has already been performed.
2. Programming should be carried out 128 bytes at a time. A 128-byte data transfer must be performed even if writing fewer than 128 bytes. In this case, H'FF data must be written to the extra addresses.
3. Prepare the following data storage areas in RAM: a 128-byte programming data area, a 128-byte reprogramming data area, and a 128-byte additional-programming data area. Perform reprogramming data computation and additional programming data computation according to figure 19.11.
4. Consecutively transfer 128 bytes of data in byte units from the reprogramming data area or additional-programming data area to the flash memory. The program address and 128-byte data are latched in the flash memory. The lower 8 bits of the start address in the flash memory destination area must be H'00 or H'80.
5. The time during which the P1 bit is set to 1 is the programming time. Figure 19.11 shows the allowable programming times.
6. The watchdog timer (WDT) is set to prevent overprogramming due to program runaway, etc. An overflow cycle of approximately $(y + z1 + \alpha + \beta)$ μs is allowed.
7. For a dummy write to a verify address, write 1-byte data H'FF to an address whose lower 1 bit is B'0. Verify data can be read in words from the address to which a dummy write was performed.
8. The maximum number of repetitions of the program/program-verify sequence to the same bit is (N1 + N2).



Note: 7. Write Pulse Width

Number of Writes n	P1 bit set time (µs)*6 Program	Re -program
1	z0	z2
2	z0	z2
⋮	⋮	⋮
N1-1	z0	z2
N1	z0	z2
N1+1	z1	—
N1+2	z1	—
N1+3	z1	—
⋮	⋮	⋮
N1+N2-2	z1	—
N1+N2-1	z1	—
N1+N2	z1	—

Notes: 1. Data transfer is performed by byte transfer. The lower 8 bits of the first address written to must be H'00 or H'80.
A 128-byte data transfer must be performed even if writing fewer than 128 bytes; in this case, H'FF data must be written to the extra addresses.
2. Verify data is read in 16-bit (word) units.
3. Reprogram data is determined by the operation shown in the table below (comparison between the data stored in the program data area and the verify data). Bits for which the reprogram data is 0 are programmed in the next reprogramming loop. Therefore, even bits for which programming has been completed will be subjected to programming once again if the result of the subsequent verify operation is NG.
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 data must be provided in RAM.
The contents of the reprogram data area and additional data area are modified as programming proceeds.
5. A write pulse of z0 or z1 is applied according to the progress of the programming operation. See note 7 for details of the pulse widths.
When writing of additional-programming data is executed, a z2 write pulse should be applied. Reprogram data X' means reprogram data when the write pulse is applied.
6. x, y, z0, z1, z2, α, β, γ, ε, η, θ, N1, and N2 are shown in section 24.8, Flash Memory Characteristics.

Reprogram Data Computation Table

Original Data (D)	Verify Data (V)	Reprogram Data (X)	Comments
0	0	1	Programming completed
0	1	0	Programming incomplete; reprogram
1	0	1	
1	1	1	Still in erased state; no action

Additional-Programming Data Computation Table

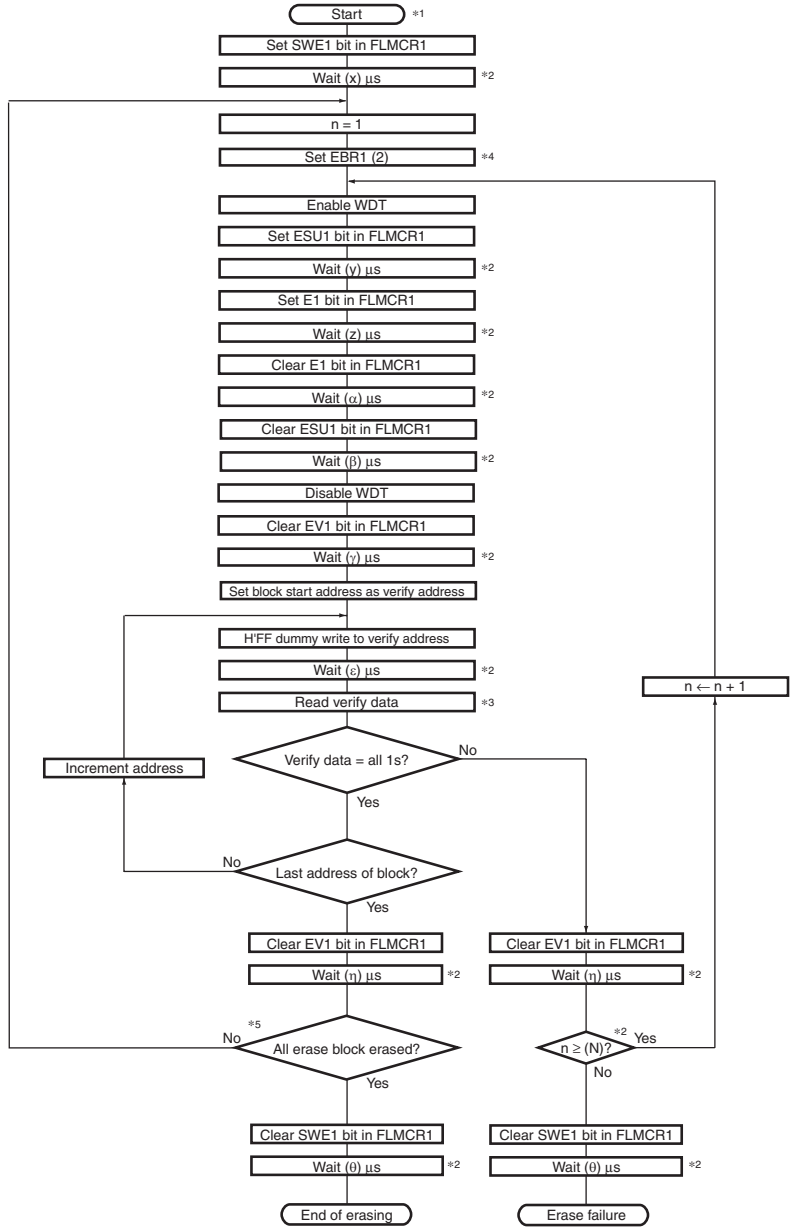
Reprogram Data (X')	Verify Data (V)	Additional-Programming Data (Y)	Comments
0	0	0	Additional programming to be executed
0	1	1	Additional programming not to be executed
1	0	1	Additional programming not to be executed
1	1	1	Additional programming not to be executed

Figure 19.11 Program/Program-Verify Flowchart

19.8.2 Erase/Erase-Verify

When erasing flash memory, the erase/erase-verify flowchart shown in figure 19.12 should be followed.

1. Prewriting (setting erase block data to all 0s) is not necessary.
2. Erasing is performed in block units. Make only a single-bit specification in the erase block register 1 (EBR1), and erase block register 2 (EBR2). To erase multiple blocks, each block must be erased in turn.
3. The time during which the E1 bit is set to 1 is the flash memory erase time.
4. The watchdog timer (WDT) is set to prevent overprogramming due to program runaway, etc. An overflow cycle of approximately (y+z+α+β) ms is allowed.
5. For a dummy write to a verify address, write 1-byte data H'FF to an address whose lower 1 bit is B'0. Verify data can be read in words from the address to which a dummy write was performed.
6. If the read data is unerased, set erase mode again, and repeat the erase/erase-verify sequence as before. The maximum number of repetitions of the erase/erase-verify sequence is N.



Notes: 1. Pre-write (clearing data in the block to be erased to 0) is not required.
2. x, y, z, α, β, γ, ε, η, θ, and N are shown in section 24.8, Flash Memory Characteristics.
3. Veryfy data is read in 16 bits.
4. Only 1 bit in the EBR register must be set. Two or more bits in EBR cannot be set.
5. Erasure is performed in block units. To erase multiple blocks, each block must be erased sequentially.

Figure 19.12 Erase/Erase-Verify Flowchart

19.9 Program/Erase Protection

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

19.9.1 Hardware Protection

Hardware protection refers to a state in which programming/erasing of flash memory is forcibly disabled or aborted because of a transition to reset or standby mode. Flash memory control register 1 (FLMCR1), flash memory control register 2 (FLMCR2), and erase block register 1 (EBR1), and erase block register 2 (EBR2) are initialized. In a reset via the $\overline{\text{RES}}$ pin, the reset state is not entered unless the $\overline{\text{RES}}$ pin is held low until oscillation stabilizes after powering on. In the case of a reset during operation, hold the $\overline{\text{RES}}$ pin low for the $\overline{\text{RES}}$ pulse width specified in the AC Characteristics section.

19.9.2 Software Protection

Software protection can be implemented against programming/erasing of all flash memory blocks by clearing the SWE1 bit in FLMCR1. When software protection is in effect, setting the P1 or E1 bit in FLMCR1 does not cause a transition to program mode or erase mode. By setting the erase block register 1 (EBR1), and erase block register 2 (EBR2), erase protection can be set for individual blocks. When EBR1 and EBR2 are set to H'00, erase protection is set for all blocks.

19.9.3 Error Protection

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

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

Setting Conditions of FLER Bit (Error Protection)

- When the flash memory of the relevant address area is read during programming/erasing (including vector read and instruction fetch)
- Immediately after exception handling (excluding a reset) during programming/erasing
- When a SLEEP instruction is executed during programming/erasing
- When the CPU releases the bus mastership to the DMAC or DTC during programming/erasing

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 P1 or E1 bit. However, PV1 and EV1 bit setting is enabled, and a transition can be made to verify mode.

19.10 Interrupt Handling when Programming/Erasing Flash Memory

All interrupts, including NMI interrupt is disabled when flash memory is being programmed or erased (when the P1 or E1 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 CPU runaway.
3. If interrupt occurred during boot program execution, it would not be possible to execute the normal boot mode sequence.

Notes: 1. Interrupt requests must be disabled inside and outside the CPU 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 P1 or E1 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 Programmer Mode

In programmer mode, a PROM programmer can perform programming/erasing via a socket adapter, just like for a discrete flash memory. Use a PROM programmer which supports the Renesas Technology 256-kbyte flash memory on-chip MCU device type. Memory map in programmer mode is shown in figure 19.13.

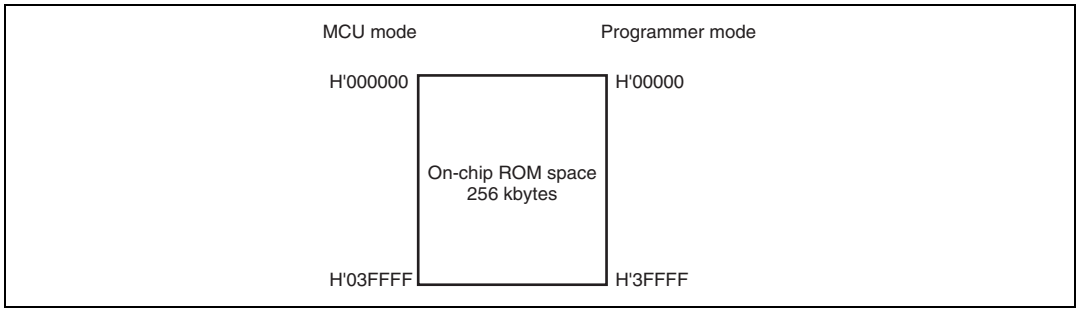


Figure 19.13 Memory Map in Programmer Mode

19.12 Power-Down States for Flash Memory

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

- Normal operating mode
  The flash memory can be read and written to.
- Standby mode
  All flash memory circuits are halted.

Table 19.8 shows the correspondence between the operating modes of this LSI and the flash memory. When the flash memory returns to normal operation from a power-down state, a power supply circuit stabilization period is needed. When the flash memory returns to its normal operating state, bits STS2 to STS0 in SBYCR must be set to provide a wait time of at least 100 μs, even when the external clock is being used and an oscillation stabilization time is not necessary.

Table 19.8 Flash Memory Operating States

LSI Operating State	Flash Memory Operating State
Active mode	Normal operating mode
Sleep mode	Normal operating mode
Standby mode	Standby mode (Before entering to the normal operation mode, wait time of at least 100 μs is required)

19.13 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 Technology 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

  Do not apply a high level to the FWE pin until VCC has stabilized. Also, drive the FWE pin low before turning off VCC. When applying or disconnecting VCC 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

  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. The following points must be observed concerning FWE application and disconnection to prevent unintentional programming or erasing of flash memory:
  - Apply FWE when the VCC voltage has stabilized within its rated voltage range.
  - 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 SWE1, ESU1, PSU1, EV1, PV1, P1, and E1 bits in FLMCR1 are cleared. Make sure that the SWE1, ESU1, PSU1, EV1, PV1, P1, and E1 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 P1 or E1 bit in FLMCR1, the watchdog timer should be set beforehand as a precaution against program runaway, etc.
- Do not set or clear the SWE1 bit during execution of a program in flash memory.

  Wait for at least θ μs* after clearing the SWE1 bit before executing a program or reading data in flash memory. When the SWE1 bit is set, data in flash memory can be rewritten, but access flash memory only for verify operations (verification during programming/erasing). Also, do not clear the SWE1 bit during programming, erasing, or verifying. Similarly, when using emulation by RAM with a high level applied to the FWE pin, the SWE1 bit should be cleared before executing a program or reading data in flash memory. However, read/write accesses can be performed in the RAM area overlapping the flash memory space regardless of whether the SWE1 bit is set or cleared.
- Do not use interrupts while flash memory is being programmed or erased.

  All interrupt requests, including NMI, should be disabled during FWE application 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 programmer 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.

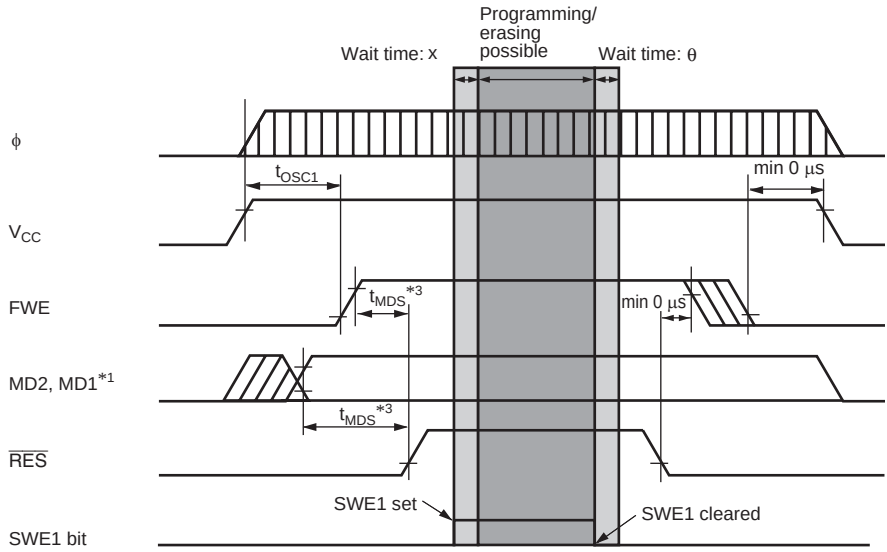
  Overcurrent damage to the device can result if the index marks on the PROM programmer socket, socket adapter, and chip are not correctly aligned.
- Do not touch the socket adapter or chip during programming.

  Touching either of these can cause contact faults and write errors.
- The reset state must be entered after powering on

  Apply the reset signal for at least 100 μs during the oscillation setting period.
- When a reset is applied during operation, this should be done while the SWE1 pin is low.

  Wait at least θ μs* after clearing the SWE1 bit before applying the reset.

Note: * Refer to section 24.8, Flash Memory Characteristics.

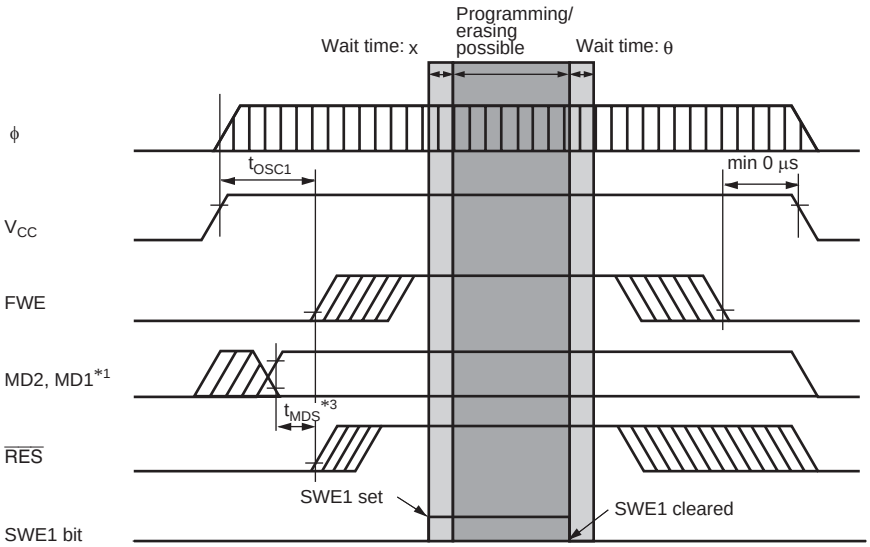


Period during which flash memory access is prohibited
x: Wait time after setting SWE1 bit)*2

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

Notes: 1. Except when switching modes, the level of the mode pins (MD2 and MD1) must be fixed until power-off by pulling the pins up or down.
2. See section 24.8, Flash Memory Characteristics.
3. Mode programming setup time t_{MDS} (min) = 200 ns.

Figure 19.14 Power-On/Off Timing (Boot Mode)

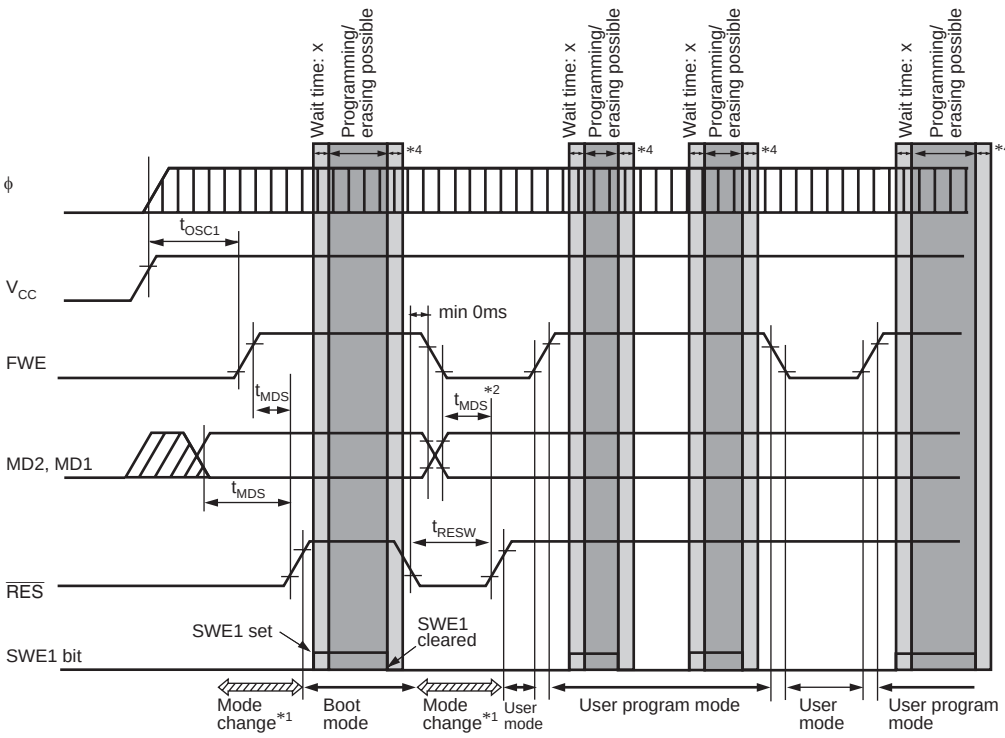


Period during which flash memory access is prohibited
(x: Wait time after setting SWE1 bit)*2

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

Notes: 1. Except when switching modes, the level of the mode pins (MD2 and MD1) must be fixed until power-off by pulling the pins up or down.
2. See section 24.8, Flash Memory Characteristics.
3. Mode programming setup time t_{MDS} (min) = 200 ns.

Figure 19.15 Power-On/Off Timing (User Program Mode)



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
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 24.8, Flash Memory Characteristics.
4. Wait time: θ.

Figure 19.16 Mode Transition Timing
(Example: Boot Mode → User Mode ↔ User Program Mode)

19.14 Note on Switching from F-ZTAT Version to Masked ROM Version

The masked ROM version does not have the internal registers for flash memory control that are provided in the F-ZTAT version. Table 19.7 lists the registers that are present in the F-ZTAT version but not in the masked ROM version. If a register listed in table 19.7 is read in the masked ROM version, an undefined value will be returned. Therefore, if application software developed on the F-ZTAT version is switched to a masked ROM version product, it must be modified to ensure that the registers in table 19.9 have no effect.

Table 19.9 Registers Present in F-ZTAT Version but Absent in Masked ROM Version

Register	Abbreviation	Address
Flash memory control register 1	FLMCR1	H'FFA8
Flash memory control register 2	FLMCR2	H'FFA9
Erase block register 1	EBR1	H'FFAA
Erase block register 2	EBR2	H'FFAB
RAM emulation register	RAMER	H'FEDB
Serial control register x	SCRX	H'FDB4

Section 20 Masked ROM

This LSI incorporates a masked ROM which has the following features.

20.1 Features

- Size:

Product Class		ROM Size	ROM Address (Modes 6 and 7)
H8S/2215 Group	HD6432215B	128 kbytes	H'000000 to H'01FFFF
	HD6432215C	64 kbytes	H'000000 to H'00FFFF

- Connected to the bus master through 16-bit data bus, enabling one-state access to both byte data and word data.

Figure 20.1 shows a block diagram of the on-chip masked ROM.

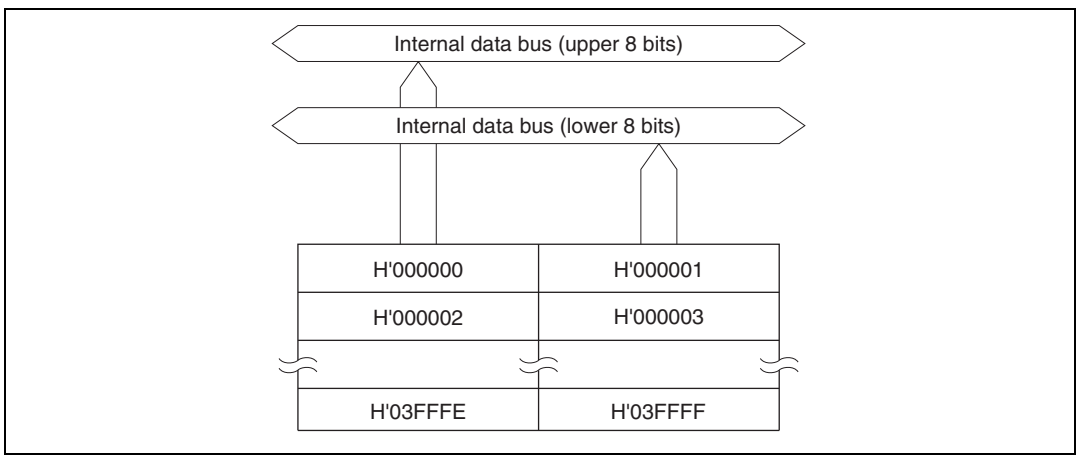


Figure 20.1 Block Diagram of On-Chip Masked ROM (256 kbytes)

Section 21 Clock Pulse Generator

This LSI has an on-chip clock pulse generator that generates the system clock (ϕ), the bus master clock, and internal clocks. The clock pulse generator consists of a system clock oscillator, duty adjustment circuit, medium-speed clock divider, bus master clock selection circuit, USB operating clock oscillator, PLL (Phase Locked Loop) circuit, and USB operating clock selection circuit. A block diagram of clock pulse generator is shown in figure 21.1.

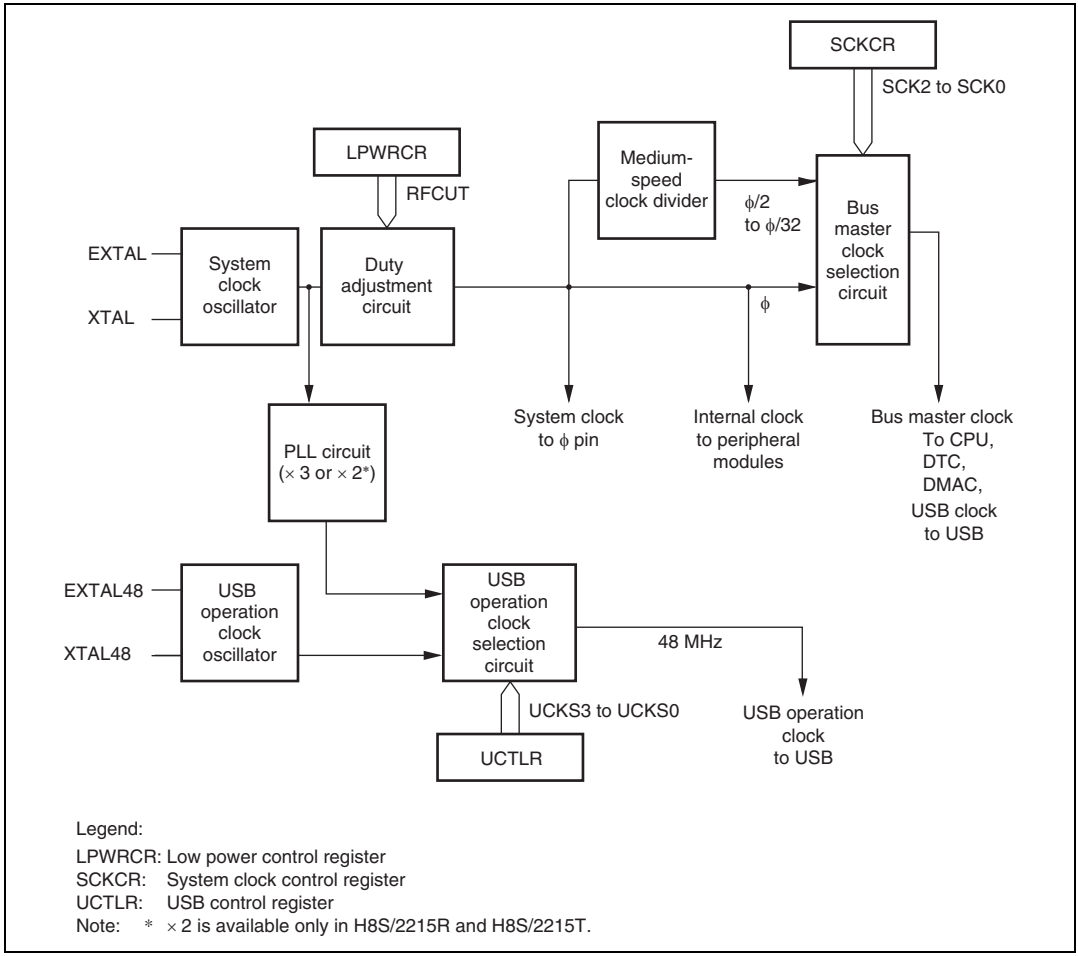


Figure 21.1 Block Diagram of Clock Pulse Generator

The frequency of the system clock oscillator can be changed by software by means of settings in the low-power control register (LPWRCR) and system clock control register (SCKCR). Either USB operating clock (48 MHz) oscillator or PLL 48-MHz clock can be selected by software by means of setting the USB control register (UCTLR). For details, refer to section 15, Universal Serial Bus Interface (USB).

21.1 Register Descriptions

The on-chip clock pulse generator has the following registers.

- System clock control register (SCKCR)
- Low-power control register (LPWRCR)

21.1.1 System Clock Control Register (SCKCR)

SCKCR controls ϕ clock output and medium-speed mode.

Bit	Bit Name	Initial Value	R/W	Description
7	PSTOP	0	R/W	φ Clock Output Disable Controls φ output. Operation differs depending on the mode. For details, see section 22.7, φ Clock Output Disabling Function. 0: φ output, fixed high, or high impedance 1: Fixed high or high impedance
6	—	0	R/W	Reserved This bit can be read from or written to, but the write value should always 0.
5, 4	—	All 0	—	Reserved These bits are always read as 0.
3	—	0	R/W	Reserved This bit can be read from or written to, but the write value should always be 0.
2	SCK2	0	R/W	System Clock Select 2 to 0
1	SCK1	0	R/W	These bits select the bus master clock.
0	SCK0	0	R/W	000: High-speed mode 001: Medium-speed clock is φ/2 010: Medium-speed clock is φ/4 011: Medium-speed clock is φ/8 100: Medium-speed clock is φ/16 101: Medium-speed clock is φ/32 11×: Setting prohibited

Legend:
×: Don't care

21.1.2 Low-Power Control Register (LPWRCR)

LPWRCR selects whether the oscillator's built-in feedback resistor and duty adjustment circuit are used with external clock input.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R/W	Reserved These bits can be read from or written to, but the write value should always be 0.
3	RFCUT	0	R/W	Built-in Feedback Resistor Control Selects whether the oscillator's built-in feedback resistor and duty adjustment circuit are used with external clock input. This bit should not be accessed when a crystal oscillator is used. After this bit is set when using external clock input, a transition should initially be made to software standby mode. Switching between use and non-use of the oscillator's built-in feedback resistor and duty adjustment circuit is performed when the transition is made to software standby mode. 0: System clock oscillator's built-in feedback resistor and duty adjustment circuit are used 1: System clock oscillator's built-in feedback resistor and duty adjustment circuit are not used
2	—	0	R/W	Reserved This bit can be read from or written to, but the write value should always be 0.
1	STC1	0	R/W	Frequency Multiplication Factor
0	STC0	0	R/W	Specify the frequency multiplication factor of the PLL circuit incorporated into the evaluation chip. The specified frequency multiplication factor is valid after a transition to software standby mode, watch mode, or subactive mode. With this LSI, STC1 and STC0 must both be set to 1. After a reset, STC1 and STC0 are both cleared to 0, and so must be set to 1. 00: × 1 01: × 2 (Setting prohibited) 10: × 4 (Setting prohibited) 11: PLL is bypassed

21.2 System Clock Oscillator

The system clock can be supplied by connecting a crystal or ceramic resonator, or by input of an external clock. Suitable resonators differ depending on the product. For details, see table 21.1.

Table 21.1 List of Suitable Resonators

	Crystal Resonator	Ceramic Resonator	External Clock
H8S/2215	13 to 16 MHz	—	13 to 16 MHz
H8S/2215R	13 to 24 MHz	—	13 to 24 MHz
H8S/2215T	—	16, 24 MHz	16, 24 MHz

Legend:
—: Not available

21.2.1 Connecting a Crystal Resonator

A crystal resonator can be connected as shown in the example in figure 21.2. Select the damping resistance R_d according to table 21.2. An AT-cut parallel-resonance crystal should be used.

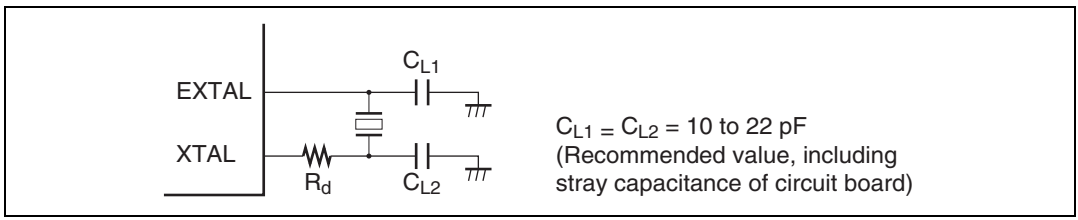


Figure 21.2 Connection of Crystal Resonator (Example)

Table 21.2 Damping Resistance Value

Frequency (MHz)	13	16	24*
R_d (Ω)	0	0	0

Note: * Available only in H8S/2215R.

Figure 21.3 shows the equivalent circuit of the crystal resonator. Use a crystal resonator that has the characteristics shown in table 21.3.

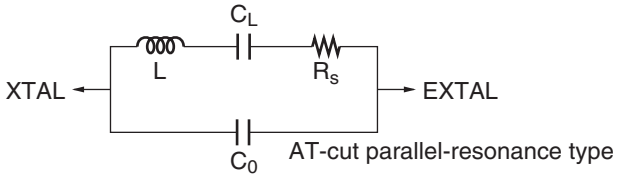


Figure 21.3 Crystal Resonator Equivalent Circuit

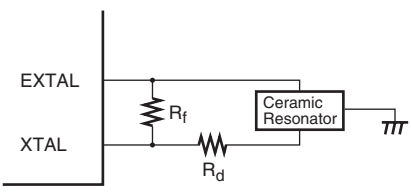
Table 21.3 Crystal Resonator Characteristics

Frequency (MHz)	**13**	**16**	**24***
R_s max (Ω)	60	50	40
C_0 max (pF)	7	7	

Note: * Available only in H8S/2215R.

21.2.2 Connecting a Ceramic Resonator (H8S/2215T)

Figure 21.6 shows an example wiring diagram for connecting a ceramic resonator.



Ceramic resonator
16 MHz: CSTCEV13L**
24 MHz: CSTCWX11***

Ta = 0 to 70 °C
Contact your Renesas Technology sales agency for details of Rf and Rd values.

Note: *** represents three-digit alphanumerics which express "Individual Specification".
Since the frequency for USB requires high accuracy, the official product type name will be defined after the performance of the user's board on which the resonator is mounted is evaluated and its frequency is adjusted.
Please contact your Renesas Technology sales agency.

Figure 21.4 Example Wiring Diagram for Connecting a Ceramic Resonator

21.2.3 Inputting an External Clock

An external clock signal can be input as shown in an example in figure 21.5. If the XTAL pin is left open, make sure that stray capacitance is no more than 10 pF. When complementary clock input to XTAL pin, the external clock input should be fixed high in standby mode.

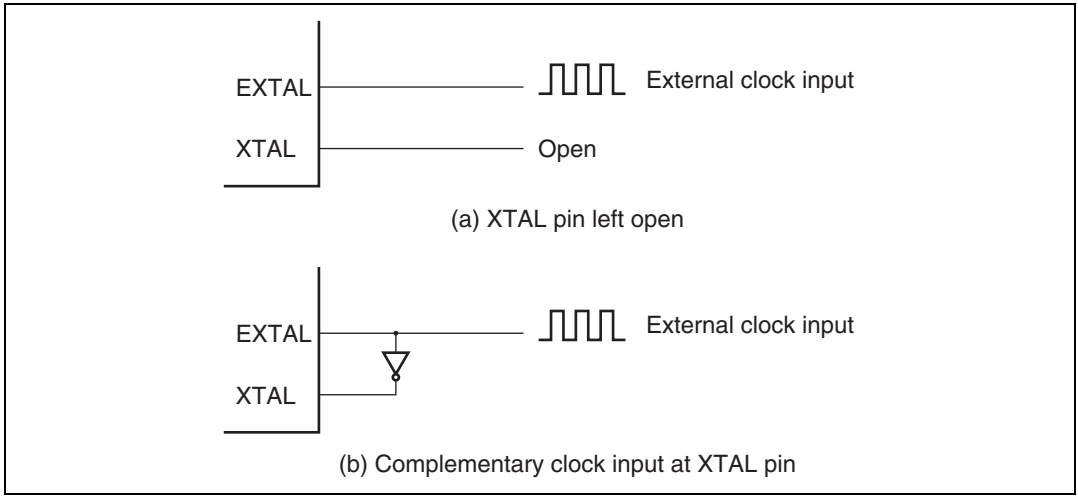


Figure 21.5 External Clock Input (Examples)

Table 21.4 shows the input conditions for the external clock.

Table 21.4 External Clock Input Conditions

		VCC = 2.7 V to 3.6 V		VCC = 3.0 V to 3.6 V*			Test
Item	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Unit**	**Conditions**
External clock input low pulse width	t_{EXL}	25	—	15.5	—	ns	Figure 21.5
External clock input high pulse width	t_{EXH}	25	—	15.5	—	ns	
External clock rise time	t_{EXr}	—	6.25	—	5.25	ns	
External clock fall time	t_{EXf}	—	6.25	—	5.25	ns	
Clock low pulse width level	t_{CL}	0.4	0.6	0.4	0.6	t_{cyc}	Figure 24.3
Clock high pulse width level	t_{CH}	0.4	0.6	0.4	0.6	t_{cyc}	

Note: * Available only in H8S/2215R.

The external clock input conditions when the duty adjustment circuit is not used are shown in table 21.5. When the duty adjustment circuit is not used, note that the maximum operating frequency depends on the external clock input waveform. For example, if $t_{EXL} = T_{EXH} = 31.25$ ns and $t_{EXr} = t_{EXf} = 6.25$ ns, the maximum operating frequency becomes 13.3 MHz depending on the clcok cycle time of 75 ns.

Table 21.5 External Clock Input Conditions when Duty Adjustment Circuit Is Not Used

		VCC = 2.7 V to 3.6 V		VCC = 3.0 V to 3.6 V*			Test
Item	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Unit**	**Conditions**
External clock input low pulse width	t_{EXL}	31.25	—	20.8	—	ns	Figure 21.5
External clock input high pulse width	t_{EXH}	31.25	—	20.8	—	ns	
External clock rise time	t_{EXr}	—	6.25	—	5.25	ns	
External clock fall time	t_{EXf}	—	6.25	—	5.25	ns	

Note: * Available only in H8S/2215R.

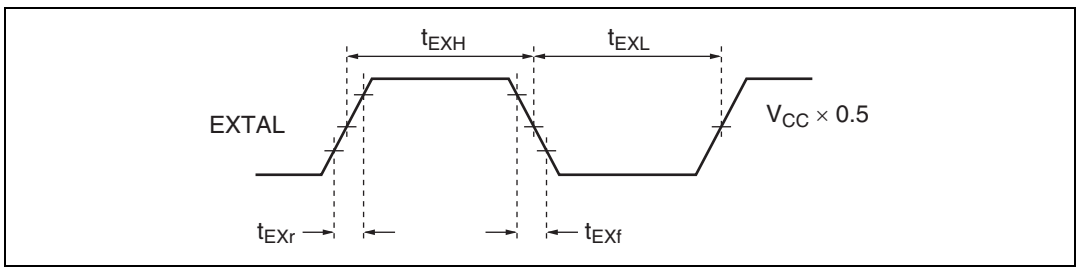


Figure 21.6 External Clock Input Timing

21.3 Duty Adjustment Circuit

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).

21.4 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.

21.5 Bus Master Clock Selection Circuit

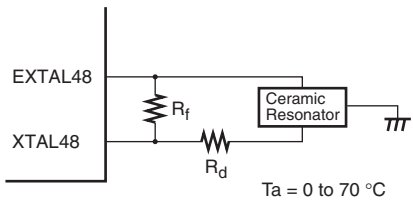
The bus master clock selection circuit selects the clock supplied to the bus master by setting the bits SCK2 to SCK0 in SCKCR. The bus master clock can be selected from high-speed mode, or medium-speed clocks (ϕ/2, ϕ/4, ϕ/8, ϕ/16, ϕ/32).

21.6 USB Operating Clock (48 MHz)

USB operating clock can be supplied by connecting a ceramic resonator, or by input of an external clock.

21.6.1 Connecting a Ceramic Resonator

Circuit Configuration: A crystal resonator can be connected as shown in the example in figure 21.7.



Ceramic resonator: CSTCW48M0X11***-R0
(Murata Manufacturing Co.,Ltd.)

Ta = 0 to 70 °C
Contact your Renesas Technology sales agency for details of Rf and Rd values.

Note: *** represents three-digit alphanumerics which express "Individual Specification".
Since the frequency for USB requires high accuracy, the official product type name will be defined after the performance of the user's board on which the resonator is mounted is evaluated and its frequency is adjusted.
Please contact your Renesas Technology sales agency.

Figure 21.7 Connection of Ceramic Resonator

21.6.2 Inputting an 48-MHz External Clock

An external clock signal can be input as shown in an example in figure 21.8. The XTAL48 pin must be left open.

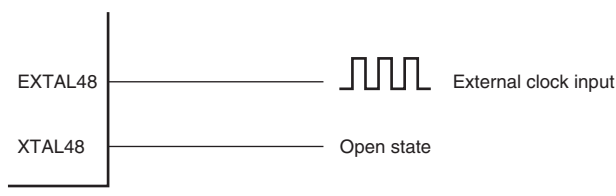


Figure 21.8 Connection of Ceramic Resonator

Table 21.6 shows the input conditions for the 48-MHz external clock.

Table 21.6 External Clock Input Conditions when Duty Adjustment Circuit Is Not Used

Item	Symbol	Min	Max	Unit	Test Conditions
External clock frequency (48 MHz)	t_{FREQ}	47.88	48.12	MHz	Figure 21.10
Clock rise time	t_{R48}	—	5	ns	
Clock fall time	t_{F48}	—	5	ns	
Duty (t_{HIGH}/t_{FREQ})	t_{DUTY}	40	60	%	

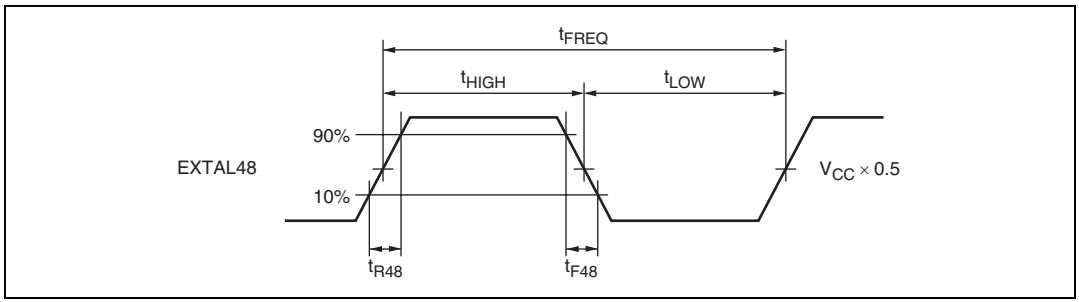


Figure 21.9 48-MHz External Clock Input Timing

21.6.3 Pin Handling when 48-MHz External Clock Is Not Needed (On-chip PLL Circuit Is Used)

When the 48-MHz external clock is not needed, connect the EXTAL48 pin to GND (Vss) and leave the XTAL48 pin open as shown in figure 21.9.

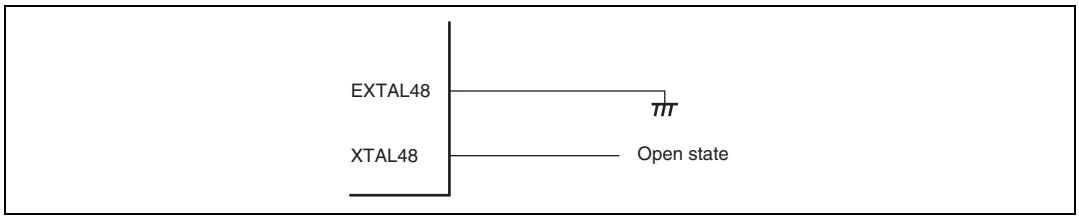


Figure 21.10 Pin Handling when 48-MHz External Clock Is Not Used

21.7 PLL Circuit for USB

The PLL circuit has the function of tripling or doubling* the 16- or 24-MHz* clock from the system oscillator to generate the 48-MHz USB operating clock. When the PLL circuit is used, set the UCKS3 to UCKS0 bits of UCTLR. For details, refer to section 15, Universal Serial Bus Interface (USB). When the PLL circuit is not used, connect the PLLVCC pin to VCC, and leave the PLLCAP pin open as shown in figure 21.11.

Note: * Available only in H8S/2215R.

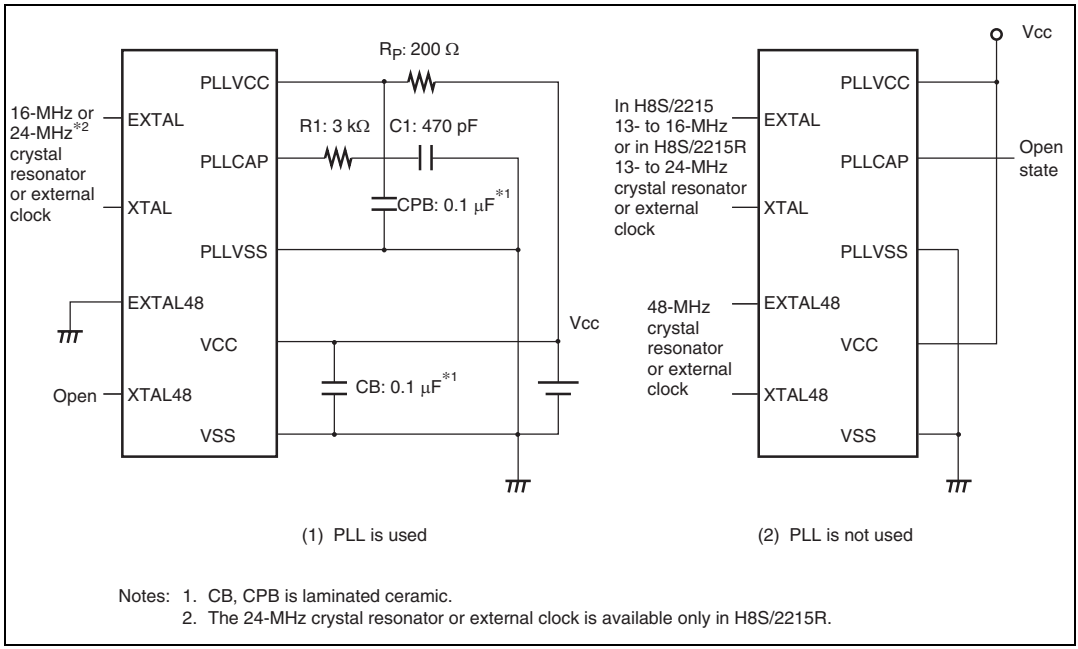


Figure 21.11 Example of PLL Circuit

When designing the board, place the capacitor C1 and resistor R1 as close as possible to the PLLCAP pin. Other signal lines should be routed away from the oscillator circuit to prevent induction from interfering with correct oscillation. C1 must be grounded to PLLVSS. In addition, PLLVCC and PLLVSS must be separated from the VCC and VSS pins. Bypass capacitors CPB and CB must be connected between VCC and VSS and between PLVCC and PLVSS, respectively.

21.8 Usage Notes

21.8.1 Note on Crystal Resonator

Since various characteristics related to the crystal resonator are closely linked to the user's board design, thorough evaluation is necessary on the user's part, using the resonator connection examples shown in this section as a guide. As the resonator circuit ratings will depend on the floating capacitance of the resonator and the mounting circuit, the ratings should be determined in consultation with the resonator manufacturer. The design must ensure that a voltage exceeding the maximum rating is not applied to the oscillator pin.

21.8.2 Note on Board Design

When designing the board, place the crystal resonator and its load capacitors as close as possible to the XTAL or XTAL48 and EXTAL or EXTAL48 pins. Other signal lines should be routed away from the oscillator circuit to prevent induction from interfering with correct oscillation. See figure 21.12.

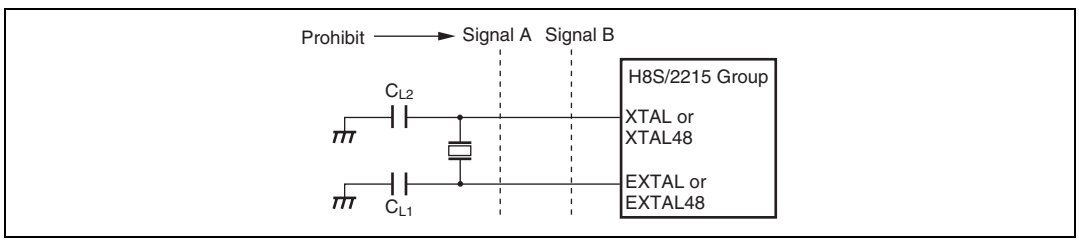


Figure 21.12 Note on Board Design of Oscillator Circuit

21.8.3 Note on Switchover of External Clock

When two or more external clocks (e.g. 16 MHz and 13 MHz) are used as the system clock, switchover of the input clock should be carried out in software standby mode.

An example of an external clock switching circuit is shown in figure 21.13, and an example of the external clock switchover timing in figure 21.14.

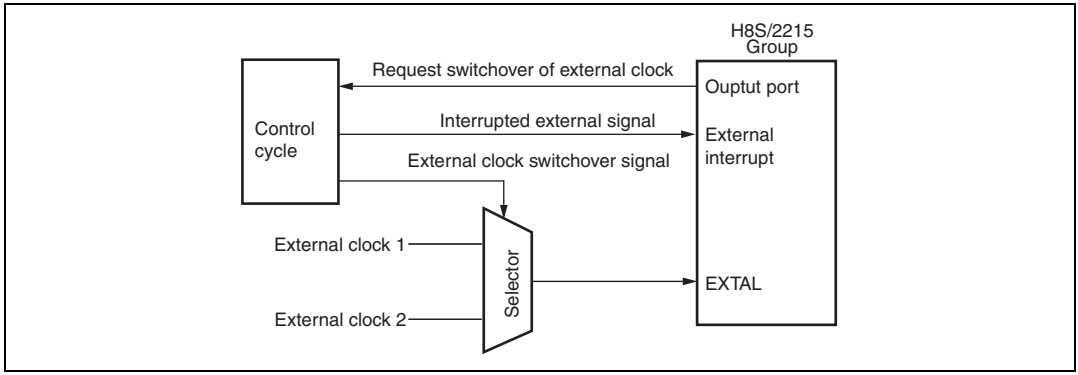


Figure 21.13 Example of External Clock Switching Circuit

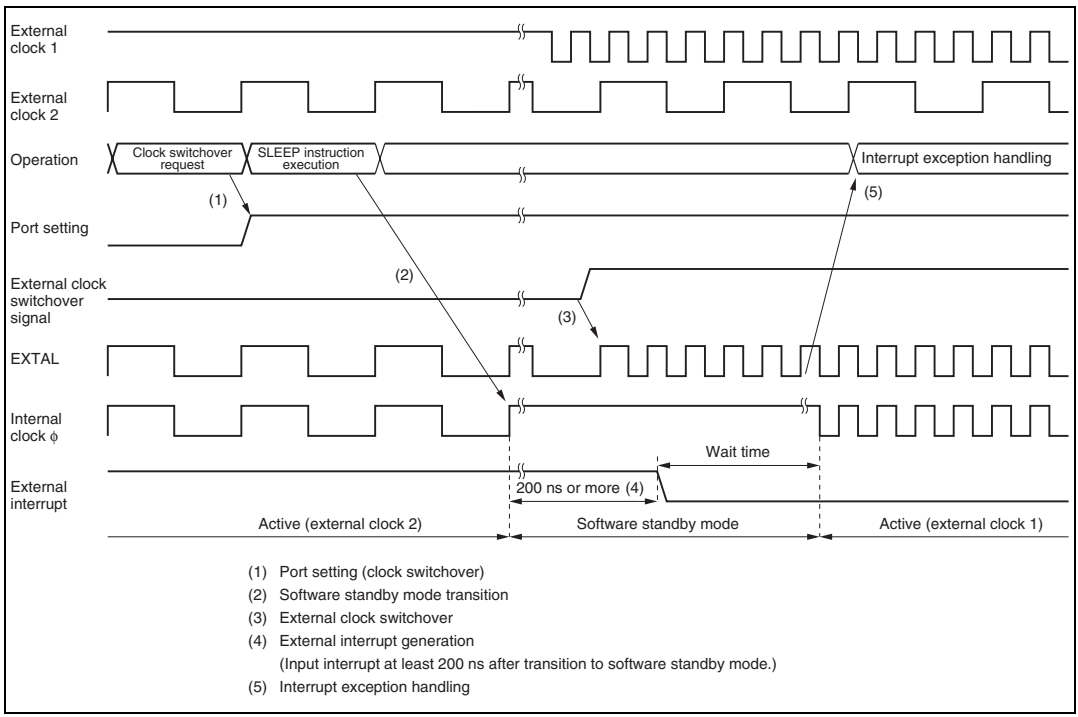


Figure 21.14 Example of External Clock Switchover Timing

Section 22 Power-Down Modes

In addition to the normal program execution state, this LSI 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.

This LSI's operating modes are high-speed mode and five power down modes:

(1) Medium-speed mode
(2) Sleep mode
(3) Module stop mode
(4) Software standby mode
(5) Hardware standby mode

(1) to (5) are power-down modes. Sleep mode is CPU states, medium-speed mode is a CPU and bus master state, and module stop mode is an internal peripheral function (including bus masters other than the CPU) state. Some of these states can be combined.

After a reset, the LSI is in high-speed mode and module stop mode (other than DMAC and DTC).

Table 22.1 shows transition between modes conditions, CPU and on-chip supporting modules states, or mode clearing methods.

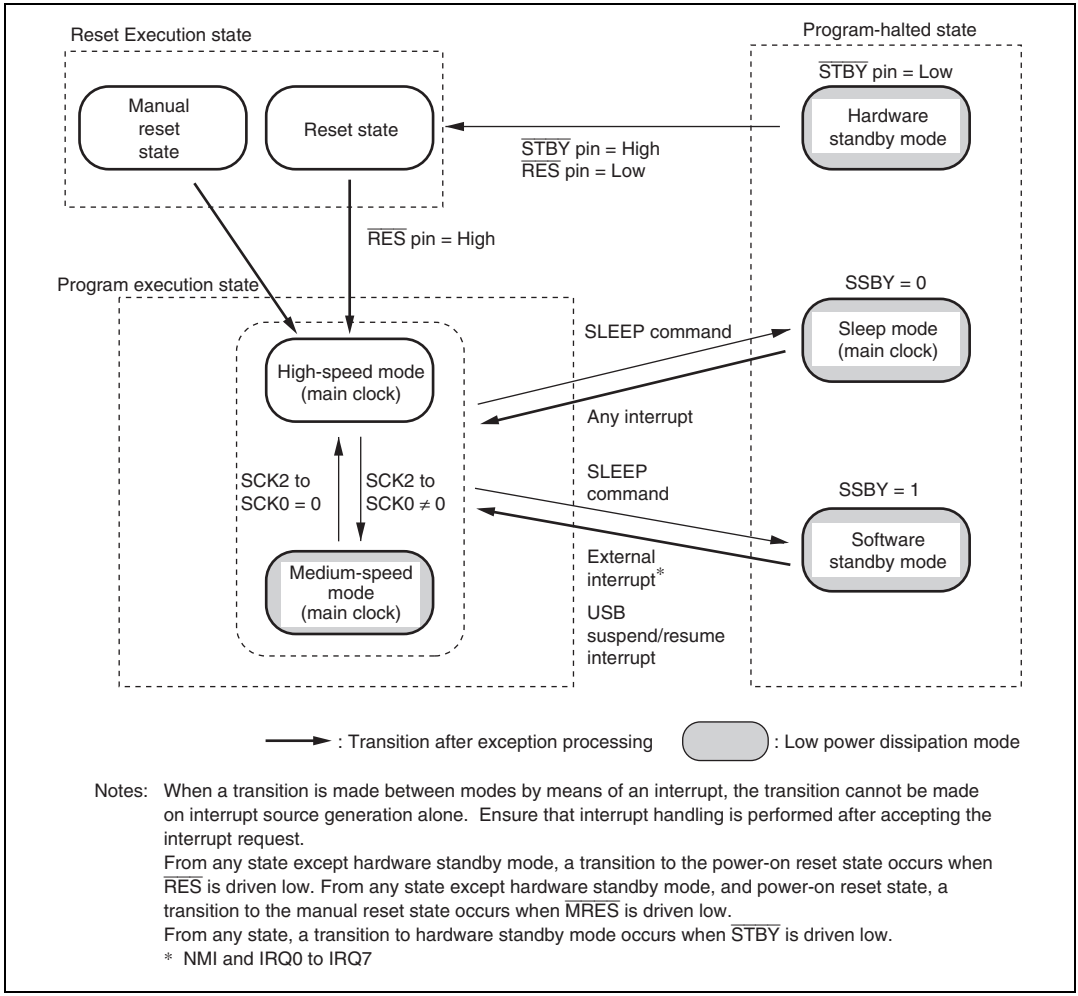
Table 22.1 LSI Internal States in Each Mode

Function		High-Speed	Medium-Speed	Sleep	Module Stop	Software Standby	Hardware Standby
System clock pulse generator		Functioning	Functioning	Functioning	Functioning	Halted	Halted
CPU	Instructions	Functioning	Medium-speed operation	Halted (retained)	High/ medium-speed operation	Halted (retained)	Halted (undefined)
	Registers						
External interrupts	NMI	Functioning	Functioning	Functioning	Functioning	Functioning	Halted
	IRQ0 to 5, 7						
Peripheral functions	DMAC	Functioning	Medium-speed operation	Functioning	Halted (retained)	Halted (retained)	Halted (reset)
	DTC						
	I/O	Functioning	Functioning	Functioning	Functioning	Retained	High impedance
	TPU	Functioning	Functioning	Functioning	Halted (retained)	Halted (retained)	Halted (reset)
	TMR						
	WDT	Functioning	Functioning	Functioning	Functioning	Halted (retained)	Halted (reset)
	D/A	Functioning	Functioning	Functioning	Halted (retained)	Halted (retained)	Halted (reset)
	A/D	Functioning	Functioning	Functioning	Halted (reset)	Halted (reset)	Halted (reset)
	SCI						
	USB	Functioning	Function not guaranteed	Functioning	Halted (retained)	Halted (retained)	Halted (reset)
	USB operating clock oscillator				Halted	Halted	
	PLL circuit						
	RAM	Functioning	Functioning	Functioning	Functioning	Retained	Retained

Notes: “Halted (retained)” means that internal register values are retained. The internal state is “operation suspended.”

“Halted (reset)” means that internal register values and internal states are initialized.

In module stop mode, only modules for which a stop setting has been made are halted (reset or retained).



Notes: When a transition is made between modes by means of an interrupt, the transition cannot be made on interrupt source generation alone. Ensure that interrupt handling is performed after accepting the interrupt request.
From any state except hardware standby mode, a transition to the power-on reset state occurs when RES is driven low. From any state except hardware standby mode, and power-on reset state, a transition to the manual reset state occurs when MRES is driven low.
From any state, a transition to hardware standby mode occurs when STBY is driven low.
* NMI and IRQ0 to IRQ7

Figure 22.1 Mode Transition Diagram

Table 22.2 Low Power Dissipation Mode Transition Conditions

Pre-Transition State	Status of Control Bit at Transition SSBY	State after Transition Invoked by SLEEP Command	State after Transition Back from Low Power Mode Invoked by Interrupt
High-speed/ Medium-speed	0	Sleep	High-speed/Medium-speed
	1	Software standby	High-speed/Medium-speed

22.1 Register Descriptions

The registers relating to the power down mode are shown below.

For details on the system clock control register (SCKCR), refer to section 21.1.1, System Clock Control Register (SCKCR).

- Standby control register (SBYCR)
- System clock control register (SCKCR)
- Module stop control register A (MSTPCRA)
- Module stop control register B (MSTPCRB)
- Module stop control register C (MSTPCRC)

22.1.1 Standby Control Register (SBYCR)

SBYCR performs software standby mode control.

Bit	Bit Name	Initial Value	R/W	Description
7	SSBY	0	R/W	Software Standby This bit specifies the transition mode after executing the SLEEP instruction 0: Shifts to sleep mode when the SLEEP instruction is executed 1: Shifts to software standby mode when the SLEEP instruction is executed This bit does not change when clearing software standby mode by using external interrupts and shifting to normal operation. 0 should be written to this bit for clearing.

Bit	Bit Name	Initial Value	R/W	Description
6	STS2	0	R/W	Standby Timer Select 2 to 0
5	STS1	0	R/W	These bits select the MCU wait time for clock stabilization when cancel software standby mode by an external interrupt. With a crystal oscillator (tables 22.3 and 22.4), select a wait time of t_{OSC2} ms (oscillation stabilization time) or more, depending on the operating frequency. With an external clock, there are no specific wait requirements. However, in the F-ZTAT version a wait time of 16 states cannot be used with an external clock. In this case it is necessary to ensure a wait time of 100 µs or more. 000: Standby time = 8192 states 001: Standby time = 16384 states 010: Standby time = 32768 states 011: Standby time = 65536 states 100: Standby time = 131072 states 101: Standby time = 262144 states 110: Standby time = 2048 states 111: Standby time = 16 states
4	STS0	0	R/W	
3	OPE	1	R/W	Output Port Enable This bit selects whether address bus and bus control signals ($\overline{CS0}$ to $\overline{CS7}$, $\overline{AS}$, $\overline{RD}$, $\overline{HWR}$, and $\overline{LWR}$) are brought to high impedance state or retained in software standby mode. 0: High impedance state 1: Retained
2 to 0	—	All 0	—	Reserved These bits are always read as 0, and cannot be modified.

22.1.2 Module Stop Control Registers A to C (MSTPCRA to MSTPCRC)

MSTPCR, comprising three 8-bit readable/writable registers, performs module stop mode control. Setting a bit to 1, causes the corresponding module to enter module stop mode, while clearing the bit to 0 clears the module stop mode.

MSTPCRA

Bit	Bit Name	Initial Value	R/W	Module
7	MSTPA7	0	R/W	DMA controller (DMAC)
6	MSTPA6	0	R/W	Data transfer controller (DTC)
5	MSTPA5	1	R/W	16-bit timer pulse unit (TPU)
4	MSTPA4	1	R/W	8-bit timer (TMR_0, TMR_1)
3	MSTPA3*	1	R/W	—
2	MSTPA2*	1	R/W	—
1	MSTPA1	1	R/W	A/D converter
0	MSTPA0*	1	R/W	—

MSTPCRB

Bit	Bit Name	Initial Value	R/W	Module
7	MSTPB7	1	R/W	Serial communication interface 0 (SCI_0)
6	MSTPB6	1	R/W	Serial communication interface 1 (SCI_1)
5	MSTPB5	1	R/W	Serial communication interface 2 (SCI_2)
4	MSTPB4*	1	R/W	—
3	MSTPB3*	1	R/W	—
2	MSTPB2*	1	R/W	—
1	MSTPB1*	1	R/W	—
0	MSTPB0	1	R/W	USB

MSTPCRC

Bit	Bit Name	Initial Value	R/W	Module
7	MSTPC7*	1	R/W	—
6	MSTPC6*	1	R/W	—
5	MSTPC5	1	R/W	D/A converter
4	MSTPC4*	1	R/W	—
3	MSTPC3*	1	R/W	—
2	MSTPC2*	1	R/W	—
1	MSTPC1*	1	R/W	—
0	MSTPC0*	1	R/W	—

Note: * MSTPA3, MSTPA2, MSTPA0, MSTPB4 to MSTPB1, MSTPC7, MSTPC6, MSTPC4 to MSTPC0 are readable/writable bits with an initial value of 1 and should always be written with 1.

22.2 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 (ϕ/2, ϕ/4, ϕ/8, ϕ/16, or ϕ/32) specified by the SCK2 to SCK0 bits. The bus masters other than the CPU (DTC or DMAC) 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 ϕ/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.

When the SLEEP instruction is executed with the SSBY bit = 1, operation shifts to the software standby mode. When software standby mode is cleared by an external interrupt, medium-speed mode is restored.

When the $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ pin is set low and medium-speed mode is cancelled, operation shifts to the reset state. The same applies in the case of a reset caused by overflow of the watchdog timer.

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

Figure 22.2 shows the timing for transition to and clearance of medium-speed mode.

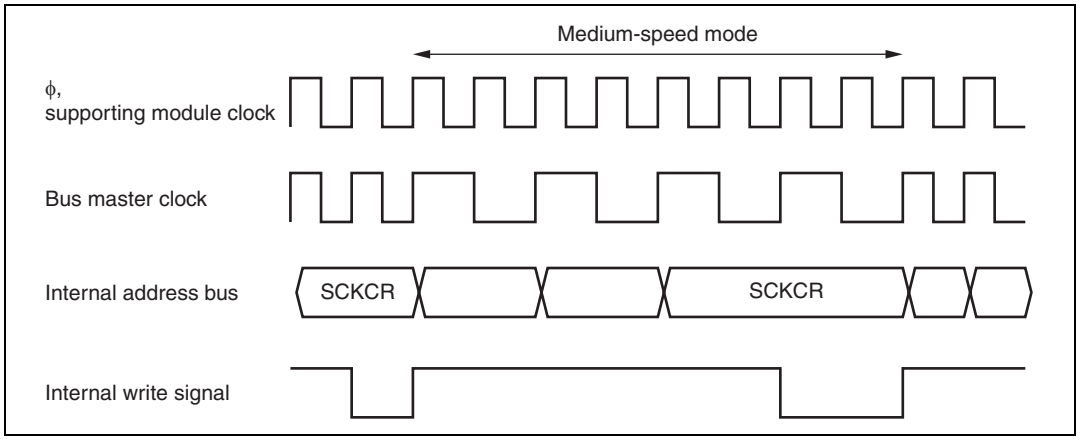


Figure 22.2 Medium-Speed Mode Transition and Clearance Timing

22.3 Sleep Mode

22.3.1 Transition to Sleep Mode

When the SLEEP instruction is executed when the SSBY bit in SBYCR is 0, the CPU enters the 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.

22.3.2 Exiting Sleep Mode

Sleep mode is exited by any interrupt, or signals at the $\overline{\text{RES}}$, $\overline{\text{MRES}}$ and $\overline{\text{STBY}}$ pins.

- Exiting Sleep Mode by Interrupts
  When an interrupt occurs, sleep mode is exited and interrupt exception processing starts. Sleep mode is not exited if the interrupt is disabled, or interrupts other than NMI are masked by the CPU.
- Exiting Sleep Mode by $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ Pin
  Setting the $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ pin level Low selects the reset state. After the stipulated reset input duration, driving the $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ pin High starts the CPU performing reset exception processing.
- Exiting Sleep Mode by $\overline{\text{STBY}}$ Pin
  When the $\overline{\text{STBY}}$ pin level is driven Low, a transition is made to hardware standby mode.

22.4 Software Standby Mode

22.4.1 Transition to Software Standby Mode

A transition is made to software standby mode when the SLEEP instruction is executed when the SSBY bit in SBYCR is 1. 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 A/D converter, and the states of I/O ports, are retained. In this mode the oscillator stops, and therefore power dissipation is significantly reduced.

22.4.2 Clearing Software Standby Mode

Software standby mode is cleared by an external interrupt (NMI pin, $\overline{\text{IRQ7}}$ pin, or $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ5}}$ pins), USB suspend/resume interrupt ($\overline{\text{IRQ6}}$ signal), or by means of the $\overline{\text{RES}}$ pin, $\overline{\text{MRES}}$ 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 SBYCR, 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 IRQ5 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.
- Clearing with the $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ pin

  When the $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ 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 $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ pin must be held low until clock oscillation stabilizes. When the $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ 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.

22.4.3 Setting Oscillation Stabilization Time after Clearing Software 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 t_{OSC2} ms (the oscillation stabilization time).

  Table 22.3 shows the standby times for different operating frequencies and settings of bits STS2 to STS0.
- Using an External Clock

  Set bits STS2 to STS0 as any value. Usually, minimum value is recommended. A 16-state standby time cannot be used in the F-ZTAT version; a standby time of 2,048 states or longer should be used.

Table 22.3 Oscillation Stabilization Time Settings

STS2	STS1	STS0	Standby Time	24 MHz[*2]	20 MHz[*1]	16 MHz	13 MHz	10 MHz	8 MHz	6 MHz	4 MHz	2 MHz	Unit
0	0	0	8192 states	0.3	0.4	0.51	0.6	0.8	1.0	1.3	2.0	4.1	ms
		1	16384 states	0.7	0.8	1.0	1.3	1.6	2.0	2.7	4.1	8.2	
	1	0	32768 states	1.4	1.6	2.0	2.5	3.3	4.1	5.5	8.2	16.4	
		1	65536 states	2.7	3.3	4.1	5.0	6.6	8.2	10.9	16.4	32.8	
1	0	0	131072 states	5.5	6.6	8.2	10.1	13.1	16.4	21.8	32.8	65.5	
		1	262144 states	10.9	13.1	16.4	20.2	26.2	32.8	43.6	65.6	131.2	
	1	0	2048 states	0.09	0.1	0.13	0.16	0.2	0.3	0.3	0.5	1.0	
		1	16 states	0.7	0.8	1.0	1.2	1.6	2.0	1.7	4.0	8.0	µs

Notes: 1. Only in H8S/2215R.
2. Only in H8S/2215R and H8S/2215T.

: Recommended time setting (See the t_{OSC2} item in table 24.4 or table 25.4, for conditions)

22.4.4 Software Standby Mode Application Example

Figure 22.3 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.

Software standby mode is then cleared at the rising edge on the NMI pin.

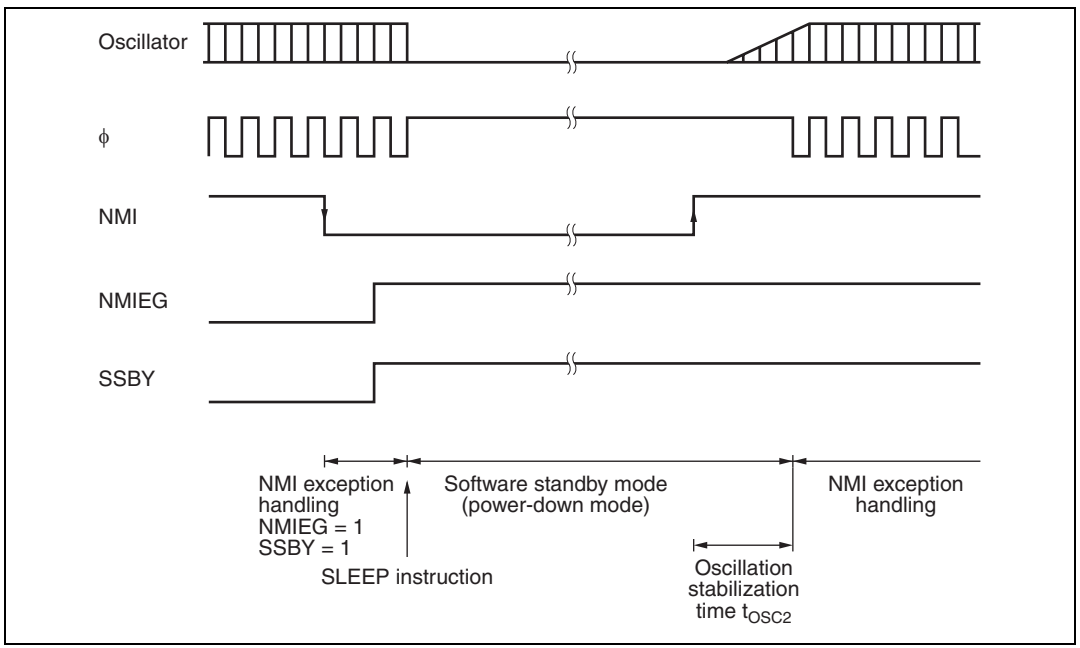


Figure 22.3 Software Standby Mode Application Example

22.5 Hardware Standby Mode

22.5.1 Transition to 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 this LSI is in hardware standby mode.

22.5.2 Clearing 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 t_{osc1}—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.

22.5.3 Hardware Standby Mode Timing

Figure 22.4 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.

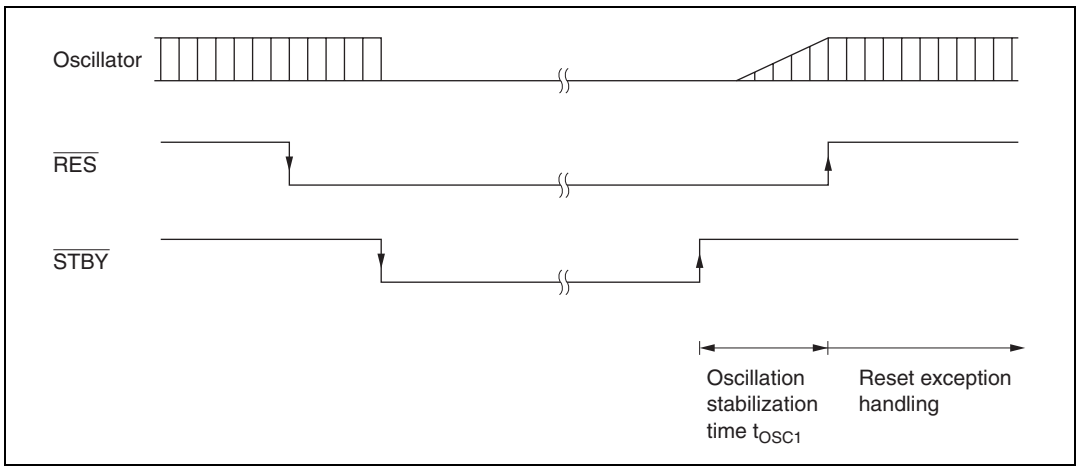


Figure 22.4 Hardware Standby Mode Timing (Example)

22.5.4 Hardware Standby Mode Timings

Timing of Transition to Hardware Standby Mode

1. To retain RAM contents with the RAME bit set to 1 in SYSCR
   Drive the $\overline{RES}$ signal low at least 10 states before the $\overline{STBY}$ signal goes low, as shown in figure 22.5. After $\overline{STBY}$ has gone low, $\overline{RES}$ has to wait for at least 0 ns before becoming high.

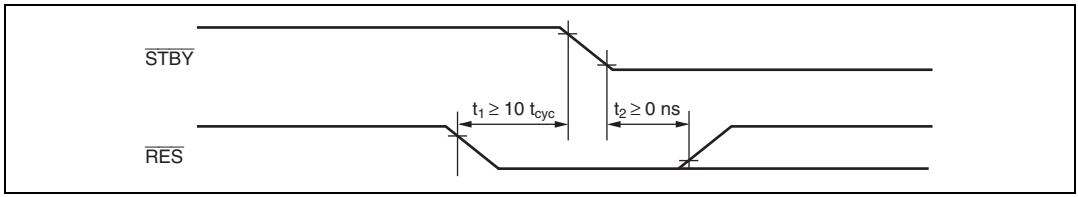


Figure 22.5 Timing of Transition to Hardware Standby Mode

2. To retain RAM contents with the RAME bit cleared to 0 in SYSCR, or when RAM contents do not need to be retained
   $\overline{RES}$ does not have to be driven low as in the above case.

Timing of Recovery from Hardware Standby Mode

Drive the $\overline{RES}$ signal low and NMI signal high approximately 100 ns or more before $\overline{STBY}$ goes high to execute a power-on reset.

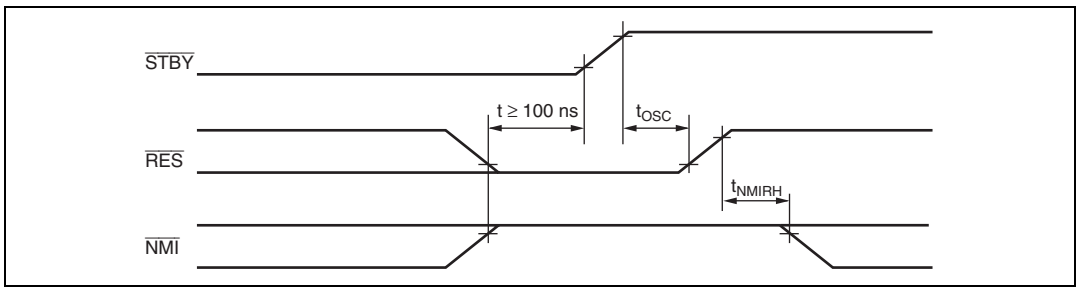


Figure 22.6 Timing of Recovery from Hardware Standby Mode

22.6 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.

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 A/D converter are retained.

After reset clearance, all modules other than DTC and DMAC are in module stop mode.

When an on-chip supporting module is in module stop mode, read/write access to its registers is disabled.

When a transition is made to sleep mode with all modules stopped, the bus controller and I/O ports also stop operating, enabling current dissipation to be further reduced.

22.7 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 22.4 shows the state of the φ pin in each processing state.

Table 22.4 φ Pin State in Each Processing State

Register Settings					
DDR	**PSTOP**	**Normal Mode**	**Sleep Mode**	**Software Standby Mode**	**Hardware Standby Mode**
0	×	High impedance	High impedance	High impedance	High impedance
1	0	φ output	φ output	Fixed high	High impedance
1	1	Fixed high	Fixed high	Fixed high	High impedance

Legend:
×: Don't care

22.8 Usage Notes

22.8.1 I/O Port Status

In software standby mode, I/O port states are retained. In addition, if the OPE bit is set to 1, the address bus and bus control signal output are retained. Therefore, there is no reduction in current dissipation for the output current when a high-level signal is output.

22.8.2 Current Dissipation during Oscillation Stabilization Wait Period

Current dissipation increases during the oscillation stabilization wait period.

22.8.3 DMAC and DTC Module Stop

Depending on the operating status of the DMAC and DTC, the MSTPA7 and MSTPA6 bits may not be set to 1. Setting of the DTC module stop mode should be carried out only when the DTC is not activated.

For details, section 7, DMA Controller (DMAC) and section 8, Data Transfer Controller (DTC).

22.8.4 On-Chip Peripheral Module Interrupts

Module interrupts do not function when in module stop mode. Consequently, it is not possible to clear CPU interrupt sources or DMAC or DTC activation sources if interrupt requests occur while in module stop mode.

For this reason, module interrupts should be disabled before entering module stop mode.

22.8.5 Writing to MSTPCR

MSTPCR should only be written to by the CPU.

Section 23 List of Registers

The register list gives information on the on-chip I/O register addresses, how the register bits are configured, and the register states in each operating mode. The information is given as shown below.

1. Register Addresses (address order)
   - Registers are listed from the lower allocation addresses.
   - Registers are classified by functional modules.
   - The access size is indicated.

2. Register Bits
   - Bit configurations of the registers are described in the same order as the Register Addresses (address order) above.
   - Reserved bits are indicated by "—" in the bit name column.
   - The bit number in the bit-name column indicates that the whole register is allocated as a counter or for holding data.
   - 16-bit or 24-bit registers are indicated from the bit on the MSB side.

3. Register States in Each Operating Mode
   - Register states are described in the same order as the Register Addresses (address order) above.
   - The register states described here are for the basic operating modes. If there is a specific reset for an on-chip peripheral module, refer to the section on that on-chip peripheral module.

23.1 Register Addresses (Address Order)

The data bus width indicates the numbers of bits by which the register is accessed.

The number of access states indicates the number of states based on the specified reference clock.

Register Name	Abbreviation	Number of Bits	Address	Data Bus Width	Number of Access States	Module
USB end point information register 00_0 to 22_4	UEPIR00_0 to UEPIR22_4	8	H'C00000 to H'C00072	8	3	USB
USB control register	UCTLR	8	H'C00080	8	3	
USB test register A	UTSTRA	8	H'C00081	8	3	
USB DMAC transfer request register	UDMAR	8	H'C00082	8	3	
USB device resume register	UDRR	8	H'C00083	8	3	
USB trigger register 0	UTRG0	8	H'C00084	8	3	
USB trigger register 1	UTRG1	8	H'C00085	8	3	
USB FIFO clear register 0	UFCLR0	8	H'C00086	8	3	
USB FIFO clear register 1	UFCLR1	8	H'C00087	8	3	
USB endpoint stall register 0	UESTL0	8	H'C00088	8	3	
USB endpoint stall register 1	UESTL1	8	H'C00089	8	3	
USB endpoint data register 0s	UEDR0s	8	H'C00090 to H'C00093	8	3	
USB endpoint data register 0i	UEDR0i	8	H'C00094 to H'C00097	8	3	
USB endpoint data register 0o	UEDR0o	8	H'C00098 to H'C0009B	8	3	
USB endpoint data register 1i	UEDR1i	8	H'C0009C to H'C0009F	8	3	
USB endpoint data register 2i	UEDR2i	8	H'C000A0 to H'C000A3	8	3	
USB endpoint data register 2o	UEDR2o	8	H'C000A4 to H'C000A7	8	3	
USB endpoint data register 3i	UEDR3i	8	H'C000A8 to H'C000AB	8	3	
USB endpoint data register 3o	UEDR3o	8	H'C000AC to H'C000AF	8	3	
USB endpoint data register 4i	UEDR4i	8	H'C000B0 to H'C000B3	8	3	
USB endpoint data register 4o	UEDR4o	8	H'C000B4 to H'C000B7	8	3	
USB endpoint data register 5i	UEDR5i	8	H'C000B8 to H'C000BB	8	3	

Register Name	Abbreviation	Number of Bits	Address	Data Bus Width	Number of Access States	Module
USB endpoint receive data size register 0o	UESZ0o	8	H'C000BC	8	3	USB
USB endpoint receive data size register 2o	UESZ2o	8	H'C000BD	8	3	
USB endpoint receive data size register 3o	UESZ3o	8	H'C000BE	8	3	
USB endpoint receive data size register 4o	UESZ4o	8	H'C000BF	8	3	
USB interrupt flag register 0	UIFR0	8	H'C000C0	8	3	
USB interrupt flag register 1	UIFR1	8	H'C000C1	8	3	
USB interrupt flag register 2	UIFR2	8	H'C000C2	8	3	
USB interrupt flag register 3	UIFR3	8	H'C000C3	8	3	
USB interrupt enable register 0	UIER0	8	H'C000C4	8	3	
USB interrupt enable register 1	UIER1	8	H'C000C5	8	3	
USB interrupt enable register 2	UIER2	8	H'C000C6	8	3	
USB interrupt enable register 3	UIER3	8	H'C000C7	8	3	
USB interrupt selection register 0	UISR0	8	H'C000C8	8	3	
USB interrupt selection register 1	UISR1	8	H'C000C9	8	3	
USB interrupt selection register 2	UISR2	8	H'C000CA	8	3	
USB interrupt selection register 3	UISR3	8	H'C000CB	8	3	
USB data status register	UDSR	8	H'C000CC	8	3	
USB configuration value register	UCVR	8	H'C000CF	8	3	
USB time stamp register H	UTSRH	8	H'C000D0	8	3	
USB time stamp register L	UTSRL	8	H'C000D1	8	3	
USB test register 0	UTSTR0	8	H'C000F0	8	3	
USB test register 1	UTSTR1	8	H'C000F1	8	3	
USB test register 2	UTSTR2	8	H'C000F2	8	3	
USB test register B	UTSTRB	8	H'C000FB	8	3	
USB test register C	UTSTRC	8	H'C000FC	8	3	
USB test register D	UTSTRD	8	H'C000FD	8	3	
USB test register E	UTSTRE	8	H'C000FE	8	3	
USB test register F	UTSTRF	8	H'C000FF	8	3	
USB reserved area	—	—	H'C00100 to H'DFFFFF	—	—	
DTC mode register A	MRA	8	H'EBC0 to H'EFBF	16/32	1	DTC
DTC source address register	SAR	24		16/32	1	
DTC mode register B	MRB	8		16/32	1	
DTC destination address register	DAR	24		16/32	1	

Register Name	Abbreviation	Number of Bits	Address	Data Bus Width	Number of Access States	Module
DTC transfer count register A	CRA	16	H'EBC0 to H'EFBF	16/32	1	DTC
DTC transfer count register B	CRB	16		16/32	1	
D/A data register_0	DADR_0	8	H'FDAC	8	2	D/A
D/A data register _1	DADR_1	8	H'FDAD	8	2	
D/A control register	DACR	8	H'FDAE	8	2	
Serial control register X	SCRX	8	H'FDB4	8	2	FLASH
Standby control register	SBYCR	8	H'FDE4	8	2	SYSTEM
System control register	SYSCR	8	H'FDE5	8	2	
System clock control register	SCKCR	8	H'FDE6	8	2	
Mode control register	MDCR	8	H'FDE7	8	2	
Module stop control register A	MSTPCRA	8	H'FDE8	8	2	
Module stop control register B	MSTPCRB	8	H'FDE9	8	2	
Module stop control register C	MSTPCRC	8	H'FDEA	8	2	
Pin function control register	PFCR	8	H'FDEB	8	2	BSC
Low power control register	LPWRCR	8	H'FDEC	8	2	SYSTEM
Serial extended mode register_0 (In H8S/2215)	SEMR_0	8	H'FDF8	8	2	SCI_0
Serial extended mode register A_0 (In H8S/2215R)	SEMRA_0	8	H'FDF8	8	2	
Serial extended mode register B_0 (In H8S/2215R)	SEMRB_0	8	H'FDF9	8	2	
IRQ sense control register H	ISCRH	8	H'FE12	8	2	INT
IRQ sense control register L	ISCRL	8	H'FE13	8	2	
IRQ enable register	IER	8	H'FE14	8	2	
IRQ status register	ISR	8	H'FE15	8	2	
DTC enable register A	DTCERA	8	H'FE16	8	2	DTC
DTC enable register B	DTCERB	8	H'FE17	8	2	
DTC enable register C	DTCERC	8	H'FE18	8	2	
DTC enable register D	DTCERD	8	H'FE19	8	2	
DTC enable register E	DTCERE	8	H'FE1A	8	2	
DTC enable register F	DTCERF	8	H'FE1B	8	2	
DTC vector register	DTVECR	8	H'FE1F	8	2	
Port 1 data direction register	P1DDR	8	H'FE30	8	2	PORT
Port 3 data direction register	P3DDR	8	H'FE32	8	2	
Port 7 data direction register	P7DDR	8	H'FE36	8	2	

Register Name	Abbreviation	Number of Bits	Address	Data Bus Width	Number of Access States	Module
Port A data direction register	PADDR	8	H'FE39	8	2	PORT
Port B data direction register	PBDDR	8	H'FE3A	8	2	
Port C data direction register	PCDDR	8	H'FE3B	8	2	
Port D data direction register	PDDDR	8	H'FE3C	8	2	
Port E data direction register	PEDDR	8	H'FE3D	8	2	
Port F data direction register	PFDDR	8	H'FE3E	8	2	
Port G data direction register	PGDDR	8	H'FE3F	8	2	
Port A pull-up MOS control register	PAPCR	8	H'FE40	8	2	
Port B pull-up MOS control register	PBPCR	8	H'FE41	8	2	
Port C pull-up MOS control register	PCPCR	8	H'FE42	8	2	
Port D pull-up MOS control register	PDPCR	8	H'FE43	8	2	
Port E pull-up MOS control register	PEPCR	8	H'FE44	8	2	
Port 3 open drain control register	P3ODR	8	H'FE46	8	2	
Port A open drain control register	PAODR	8	H'FE47	8	2	
Timer start register	TSTR	8	H'FEB0	16	2	TPU
Timer synchro register	TSYR	8	H'FEB1	16	2	
Interrupt priority register A	IPRA	8	H'FEC0	8	2	INT
Interrupt priority register B	IPRB	8	H'FEC1	8	2	
Interrupt priority register C	IPRC	8	H'FEC2	8	2	
Interrupt priority register D	IPRD	8	H'FEC3	8	2	
Interrupt priority register E	IPRE	8	H'FEC4	8	2	
Interrupt priority register F	IPRF	8	H'FEC5	8	2	
Interrupt priority register G	IPRG	8	H'FEC6	8	2	
Interrupt priority register I	IPRI	8	H'FEC8	8	2	
Interrupt priority register J	IPRJ	8	H'FEC9	8	2	
Interrupt priority register K	IPRK	8	H'FECA	8	2	
Interrupt priority register M	IPRM	8	H'FECC	8	2	
Bus width control register	ABWCR	8	H'FED0	8	2	BSC
Access state control register	ASTCR	8	H'FED1	8	2	
Wait control register H	WCRH	8	H'FED2	8	2	
Wait control register L	WCRL	8	H'FED3	8	2	
Bus control register H	BCRH	8	H'FED4	8	2	
Bus control register L	BCRL	8	H'FED5	8	2	
RAM emulation register	RAMER	8	H'FEDB	8	2	FLASH

Register Name	Abbreviation	Number of Bits	Address	Data Bus Width	Number of Access States	Module
Memory address register 0A H	MAR0AH	16	H'FEE0	16	2	DMAC
Memory address register 0A L	MAR0AL	16	H'FEE2	16	2	
I/O address register 0A	IOAR0A	16	H'FEE4	16	2	
Transfer count register 0A	ETCR0A	16	H'FEE6	16	2	
Memory address register 0B H	MAR0BH	16	H'FEE8	16	2	
Memory address register 0B L	MAR0BL	16	H'FEEA	16	2	
I/O address register 0B	IOAR0B	16	H'FEEC	16	2	
Transfer count register 0B	ETCR0B	16	H'FEEE	16	2	
Memory address register 1A H	MAR1AH	16	H'FEF0	16	2	
Memory address register 1A L	MAR1AL	16	H'FEF2	16	2	
I/O address register 1A	IOAR1A	16	H'FEF4	16	2	
Transfer count register 1A	ETCR1A	16	H'FEF6	16	2	
Memory address register 1BH	MAR1BH	16	H'FEF8	16	2	
Memory address register 1BL	MAR1BL	16	H'FEFA	16	2	
I/O address register 1B	IOAR1B	16	H'FEFC	16	2	
Transfer count register 1B	ETCR1B	16	H'FEFE	16	2	
Port 1 data register	P1DR	8	H'FF00	8	2	PORT
Port 3 data register	P3DR	8	H'FF02	8	2	
Port 7 data register	P7DR	8	H'FF06	8	2	
Port A data register	PADR	8	H'FF09	8	2	
Port B data register	PBDR	8	H'FF0A	8	2	
Port C data register	PCDR	8	H'FF0B	8	2	
Port D data register	PDDR	8	H'FF0C	8	2	
Port E data register	PEDR	8	H'FF0D	8	2	
Port F data register	PFDR	8	H'FF0E	8	2	
Port G data register	PGDR	8	H'FF0F	8	2	
Timer control register_0	TCR_0	8	H'FF10	16	2	TPU_0
Timer mode register_0	TMDR_0	8	H'FF11	16	2	
Timer I/O control register H_0	TIORH_0	8	H'FF12	16	2	
Timer I/O control register L_0	TIORL_0	8	H'FF13	16	2	
Timer interrupt enable register_0	TIER_0	8	H'FF14	16	2	
Timer status register_0	TSR_0	8	H'FF15	16	2	
Timer counter_0	TCNT_0	16	H'FF16	16	2	
Timer general register A_0	TGRA_0	16	H'FF18	16	2	

Register Name	Abbreviation	Number of Bits	Address	Data Bus Width	Number of Access States	Module
Timer general register B_0	TGRB_0	16	H'FF1A	16	2	TPU_0
Timer general register C_0	TGRC_0	16	H'FF1C	16	2	
Timer general register D_0	TGRD_0	16	H'FF1E	16	2	
Timer control register_1	TCR_1	8	H'FF20	16	2	TPU_1
Timer mode register_1	TMDR_1	8	H'FF21	16	2	
Timer I/O control register _1	TIOR_1	8	H'FF22	16	2	
Timer interrupt enable register _1	TIER_1	8	H'FF24	16	2	
Timer status register_1	TSR_1	8	H'FF25	16	2	
Timer counter_1	TCNT_1	16	H'FF26	16	2	
Timer general register A_1	TGRA_1	16	H'FF28	16	2	
Timer general register B_1	TGRB_1	16	H'FF2A	16	2	
Timer control register_2	TCR_2	8	H'FF30	16	2	TPU_2
Timer mode register_2	TMDR_2	8	H'FF31	16	2	
Timer I/O control register 2	TIOR_2	8	H'FF32	16	2	
Timer interrupt enable register 2	TIER_2	8	H'FF34	16	2	
Timer status register_2	TSR_2	8	H'FF35	16	2	
Timer counter_2	TCNT_2	16	H'FF36	16	2	
Timer general register A_2	TGRA_2	16	H'FF38	16	2	
Timer general register B_2	TGRB_2	16	H'FF3A	16	2	
DMA write enable register	DMAWER	8	H'FF60	8	2	DMAC
DMA control register 0A	DMACR0A	8	H'FF62	16	2	
DMA control register 0B	DMACR0B	8	H'FF63	16	2	
DMA control register 1A	DMACR1A	8	H'FF64	16	2	
DMA control register 1B	DMACR1B	8	H'FF65	16	2	
DMA band control register	DMABCR	16	H'FF66	16	2	
Timer control register_0	TCR_0	8	H'FF68	8	2	TMR_0
Timer control register_1	TCR_1	8	H'FF69	8	2	TMR_1
Timer control/status register_0	TCSR_0	8	H'FF6A	8	2	TMR_0
Timer control/status register_1	TCSR_1	8	H'FF6B	8	2	TMR_1
Time constant register A0	TCORA_0	8	H'FF6C	8	2	TMR_0
Time constant register A1	TCORA_1	8	H'FF6D	8	2	TMR_1
Time constant register B0	TCORB_0	8	H'FF6E	8	2	TMR_0

Register Name	Abbreviation	Number of Bits	Address	Data Bus Width	Number of Access States	Module
Time constant register B1	TCORB_1	8	H'FF6F	8	2	TMR_1
Timer counter_0	TCNT_0	8	H'FF70	8	2	TMR_0
Timer counter_1	TCNT_1	8	H'FF71	8	2	TMR_1
Timer control/status register	TCSR	8	H'FF74	16	2	WDT
Timer counter	TCNT	8	H'FF74 (write)	16	2	
Timer counter	TCNT	8	H'FF75 (read)	16	2	
Reset control/status register	RSTCSR	8	H'FF76 (write)	16	2	
Reset control/status register	RSTCSR	8	H'FF77 (read)	16	2	
Serial mode register_0	SMR_0	8	H'FF78	8	2	SCI_0
Bit rate register_0	BRR_0	8	H'FF79	8	2	
Serial control register_0	SCR_0	8	H'FF7A	8	2	
Transmit data register_0	TDR_0	8	H'FF7B	8	2	
Serial status register_0	SSR_0	8	H'FF7C	8	2	
Receive data register_0	RDR_0	8	H'FF7D	8	2	
Smart card mode register_0	SCMR_0	8	H'FF7E	8	2	
Serial mode register_1	SMR_1	8	H'FF80	8	2	SCI_1
Bit rate register_1	BRR_1	8	H'FF81	8	2	
Serial control register_1	SCR_1	8	H'FF82	8	2	
Transmit data register _1	TDR_1	8	H'FF83	8	2	
Serial status register_1	SSR_1	8	H'FF84	8	2	
Receive data register _1	RDR_1	8	H'FF85	8	2	
Smart card mode register _1	SCMR_1	8	H'FF86	8	2	
Serial mode register_2	SMR_2	8	H'FF88	8	2	SCI_2
Bit rate register_2	BRR_2	8	H'FF89	8	2	
Serial control register_2	SCR_2	8	H'FF8A	8	2	
Transmit data register_2	TDR_2	8	H'FF8B	8	2	
Serial status register_2	SSR_2	8	H'FF8C	8	2	
Receive data register_2	RDR_2	8	H'FF8D	8	2	
Smart card mode register_2	SCMR_2	8	H'FF8E	8	2	
A/D data register AH	ADDRAH	8	H'FF90	8	2	A/D

Register Name	Abbreviation	Number of Bits	Address	Data Bus Width	Number of Access States	Module
A/D data register AL	ADDRAL	8	H'FF91	8	2	A/D
A/D data register BH	ADDRBH	8	H'FF92	8	2	
A/D data register BL	ADDRBL	8	H'FF93	8	2	
A/D data register CH	ADDRCH	8	H'FF94	8	2	
A/D data register CL	ADDRCL	8	H'FF95	8	2	
A/D data register DH	ADDRDH	8	H'FF96	8	2	
A/D data register DL	ADDRDL	8	H'FF97	8	2	
A/D control/status register	ADCSR	8	H'FF98	8	2	
A/D control register	ADCR	8	H'FF99	8	2	
Flash memory control register 1	FLMCR1	8	H'FFA8	8	2	FLASH
Flash memory control register 2	FLMCR2	8	H'FFA9	8	2	
Erase block register 1	EBR1	8	H'FFAA	8	2	
Erase block register 2	EBR2	8	H'FFAB	8	2	
Port 1 register	PORT1	8	H'FFB0	8	2	PORT
Port 3 register	PORT3	8	H'FFB2	8	2	
Port 4 register	PORT4	8	H'FFB3	8	2	
Port 7 register	PORT7	8	H'FFB6	8	2	
Port 9 register	PORT9	8	H'FFB8	8	2	
Port A register	PORTA	8	H'FFB9	8	2	
Port B register	PORTB	8	H'FFBA	8	2	
Port C register	PORTC	8	H'FFBB	8	2	
Port D register	PORTD	8	H'FFBC	8	2	
Port E register	PORTE	8	H'FFBD	8	2	
Port F register	PORTF	8	H'FFBE	8	2	
Port G register	PORTG	8	H'FFBF	8	2	

23.2 Register Bits

Register addresses and bit names of the on-chip peripheral modules are described below.

Each line covers eight bits, so 16-bit registers are shown as two lines and 32-bit registers as four lines.

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
UEPIRnn_0*1	D39	D38	D37	D36	D35	D34	D33	D32	USB
UEPIRnn_1*1	D31	D30	D29	D28	D27	D26	D25	D24	
UEPIRnn_2*1	D23	D22	D21	D20	D19	D18	D17	D16	
UEPIRnn_3*1	D15	D14	D13	D12	D11	D10	D9	D8	
UEPIRnn_4*1	D7	D6	D5	D4	D3	D2	D1	D0	
UCTLR	FADSEL	SFME	UCKS3	UCKS2	UCKS1	UCKS0	UIFRST	UDCRST	
UTSTRA	—	—	—	—	—	—	—	—	
UDMAR	EP4oT1	EP4oT0	EP4iT1	EP4iT0	EP2oT1	EP2oT0	EP2iT1	EP2iT0	
UDRR	—	—	—	—	—	—	RWUPs	DVR	
UTRG0	—	—	EP2oRDFN	EP2iPKTE	EP1iPKTE	EP0oRDFN	EP0iPKTE	EP0sRDFN	
UTRG1	—	—	—	—	—	EP5iPKTE	EP4oRDFN	EP4iPKTE	
UFCLR0	EP3oCLR	EP3iCLR	EP2oCLR	EP2iCLR	EP1iCLR	EP0oCLR	EP0iCLR	—	
UFCLR1	—	—	—	—	—	EP5iCLR	EP4oCLR	EP4iCLR	
UESTL0	EP3oSTL	EP3iSTL	EP2oSTL	EP2iSTL	EP1iSTL	—	—	EP0STL	
UESTL1	SCME	—	—	—	—	EP5iSTL	EP4oSTL	EP4iSTL	
UEDR0s	D7	D6	D5	D4	D3	D2	D1	D0	
UEDR0i	D7	D6	D5	D4	D3	D2	D1	D0	
UEDR0o	D7	D6	D5	D4	D3	D2	D1	D0	
UEDR1i	D7	D6	D5	D4	D3	D2	D1	D0	
UEDR2i	D7	D6	D5	D4	D3	D2	D1	D0	
UEDR2o	D7	D6	D5	D4	D3	D2	D1	D0	
UEDR3i	D7	D6	D5	D4	D3	D2	D1	D0	
UEDR3o	D7	D6	D5	D4	D3	D2	D1	D0	
UEDR4i	D7	D6	D5	D4	D3	D2	D1	D0	
UEDR4o	D7	D6	D5	D4	D3	D2	D1	D0	
UEDR5i	D7	D6	D5	D4	D3	D2	D1	D0	
UESZ0o	—	D6	D5	D4	D3	D2	D1	D0	
UESZ2o	—	D6	D5	D4	D3	D2	D1	D0	
UESZ3o	D7	D6	D5	D4	D3	D2	D1	D0	
UESZ4o	—	D6	D5	D4	D3	D2	D1	D0	
UIFR0	BRST	—	EP1iTR	EP1iTS	EP0oTS	EP0iTR	EP0iTS	SetupTS	
UIFR1	EP3oTF	EP3oTS	EP3iTF	EP3iTR	—	EP2oREADY	EP2iTR	EP2iEMPTY	
UIFR2	—	—	EP5iTR	EP5iTS	—	EP4oREADY	EP4iTR	EP4iEMPTY	
UIFR3	CK48 READY	SOF	SETC	SETI	SPRSs	SPRSi	VBUSs	VBUSi	

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
UIER0	BRSTE	—	EP1iTRE	EP1iTSE	EP0oTSE	EP0iTRE	EP0iTSE	SetupTSE	USB
UIER1	—	—	EP3iTFE	EP3iTRE	—	EP2o READYE	EP2iTRE	EP2i EMPTYE	
UIER2	—	—	EP5iTRE	EP5iTSE	—	EP4o READYE	EP4iTRE	EP4i EMPTYE	
UIER3	CK48 READYE	SOFE	SETCE	SETIE	—	SPRSiE	—	VBUSiE	
UISR0	BRSTS	—	EP1iTRS	EP1iTSS	EP0oTSS	EP0iTRS	EP0iTSS	SetupTSS	
UISR1	—	—	EP3iTFS	EP3iTRS	—	EP2o READYS	EP2iTRS	EP2i EMPTYS	
UISR2	—	—	EP5iTRS	EP5iTSS	—	EP4o READYS	EP4iTRS	EP4i EMPTYS	
UISR3	CK48 READYS	SOFS	SETCS	SETIS	—	—	—	VBUSiS	
UDSR	—	—	EP5iDE	EP4iDE	—	EP2iDE	EP1iDE	EP0iDE	
UCVR	—	—	CNFV0	INTV1	INTV0	ALTV2	ALTV1	ALTV0	
UTSRH	—	—	—	—	—	D10	D9	D8	
UTSRL	D7	D6	D5	D4	D3	D2	D1	D0	
UTSTR0	PTSTE	—	—	—	SUSPEND	$\overline{OE}$	FSE0	VPO	
UTSTR1	VBUS	$\overline{UBPM}$	—	—	—	RCV	VP	VM	
UTSTR2	—	—	—	—	—	—	—	—	
UTSTRB	—	—	—	—	—	—	—	—	
UTSTRC	—	—	—	—	—	—	—	—	
UTSTRD	—	—	—	—	—	—	—	—	
UTSTRE	—	—	—	—	—	—	—	—	
UTSTRF	—	—	—	—	—	—	—	—	
MRA	SM1	SM0	DM1	DM0	MD1	MD0	DTS	Sz	DTC
SAR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
MRB	CHNE	DISEL	—	—	—	—	—	—	
DAR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CRA	—	—	—	—	—	—	—	—	
CRB	—	—	—	—	—	—	—	—	

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
DADR_0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	D/A
DADR_1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
DACR	DAOE1	DAOE0	DAE	—	—	—	—	—	
SCRX	—	—	—	—	FLSHE	—	—	—	FLASH
SBYCR	SSBY	STS2	STS1	STS0	OPE	—	—	—	SYSTEM
SYSCR	—	—	INTM1	INTM0	NMIEG	MRESE	—	RAME	
SCKCR	PSTOP	—	—	—	—	SCK2	SCK1	SCK0	
MDCR	—	—	—	—	—	MDS2	MDS1	MDS0	
MSTPCRA	MSTPA7	MSTPA6	MSTPA5	MSTPA4	MSTPA3	MSTPA2	MSTPA1	MSTPA0	
MSTPCRB	MSTPB7	MSTPB6	MSTPB5	MSTPB4	MSTPB3	MSTPB2	MSTPB1	MSTPB0	
MSTPCRC	MSTPC7	MSTPC6	MSTPC5	MSTPC4	MSTPC3	MSTPC2	MSTPC1	MSTPC0	
PFCR	—	—	—	—	AE3	AE2	AE1	AE0	BSC
LPWRCR	—	—	—	—	RFCUT	—	STC1	STC0	SYSTEM
SEMR_0	SSE	—	—	—	ABCS	ACS2	ACS1	ACS0	SCI_0
ISCRH	IRQ7SCB	IRQ7SCA	IRQ6SCB	IRQ6SCA	IRQ5SCB	IRQ5SCA	IRQ4SCB	IRQ4SCA	INT
ISCRL	IRQ3SCB	IRQ3SCA	IRQ2SCB	IRQ2SCA	IRQ1SCB	IRQ1SCA	IRQ0SCB	IRQ0SCA	
IER	IRQ7E	IRQ6E	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IRQ0E	
ISR	IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F	
DTCERA	DTCEA7	DTCEA6	DTCEA5	DTCEA4	DTCEA3	DTCEA2	—	DTCEA0	DTC
DTCERB	—	DTCEB6	DTCEB5	DTCEB4	DTCEB3	DTCEB2	DTCEB1	DTCEB0	
DTCERC	DTCEC7	DTCEC6	—	—	—	—	—	—	
DTCERD	—	—	—	—	DTCED3	DTCED2	DTCED1	DTCED0	
DTCERE	DTCEE7	DTCEE6	DTCEE5	DTCEE4	DTCEE3	DTCEE2	DTCEE1	DTCEE0	
DTCERF	DTCEF7	DTCEF6	—	—	—	—	—	—	
DTVECR	SWDTE	DTVEC6	DTVEC5	DTVEC4	DTVEC3	DTVEC2	DTVEC1	DTVEC0	
P1DDR	P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DDR	PORT
P3DDR	—	P36DDR	P35DDR	P34DDR	P33DDR	P32DDR	P31DDR	P30DDR	
P7DDR	—	—	—	P74DDR	P73DDR	P72DDR	P71DDR	P70DDR	
PADDR	—	—	—	—	PA3DDR	PA2DDR	PA1DDR	PA0DDR	
PBDDR	PB7DDR	PB6DDR	PB5DDR	PB4DDR	PB3DDR	PB2DDR	PB1DDR	PB0DDR	
PCDDR	PC7DDR	PC6DDR	PC5DDR	PC4DDR	PC3DDR	PC2DDR	PC1DDR	PC0DDR	
PDDDR	PD7DDR	PD6DDR	PD5DDR	PD4DDR	PD3DDR	PD2DDR	PD1DDR	PD0DDR	
PEDDR	PE7DDR	PE6DDR	PE5DDR	PE4DDR	PE3DDR	PE2DDR	PE1DDR	PE0DDR	
PFDDR	PF7DDR	PF6DDR	PF5DDR	PF4DDR	PF3DDR	PF2DDR	PF1DDR	PF0DDR	
PGDDR	—	—	—	PG4DDR	PG3DDR	PG2DDR	PG1DDR	PG0DDR	
PAPCR	—	—	—	—	PA3PCR	PA2PCR	PA1PCR	PA0PCR	

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
PBPCR	PB7PCR	PB6PCR	PB5PCR	PB4PCR	PB3PCR	PB2PCR	PB1PCR	PB0PCR	PORT
PCPCR	PC7PCR	PC6PCR	PC5PCR	PC4PCR	PC3PCR	PC2PCR	PC1PCR	PC0PCR	
PDPCR	PD7PCR	PD6PCR	PD5PCR	PD4PCR	PD3PCR	PD2PCR	PD1PCR	PD0PCR	
PEPCR	PE7PCR	PE6PCR	PE5PCR	PE4PCR	PE3PCR	PE2PCR	PE1PCR	PE0PCR	
P3ODR	—	P36ODR	P35ODR	P34ODR	P33ODR	P32ODR	P31ODR	P30ODR	
PAODR	—	—	—	—	PA3ODR	PA2ODR	PA1ODR	PA0ODR	
TSTR	—	—	—	—	—	CST2	CST1	CST0	TPU
TSYR	—	—	—	—	—	SYNC2	SYNC1	SYNC0	
IPRA	—	IPRA6	IPRA5	IPRA4	—	IPRA2	IPRA1	IPRA0	INT
IPRB	—	IPRB6	IPRB5	IPRB4	—	IPRB2	IPRB1	IPRB0	
IPRC	—	IPRC6	IPRC5	IPRC4	—	IPRC2	IPRC1	IPRC0	
IPRD	—	IPRD6	IPRD5	IPRD4	—	—	—	—	
IPRE	—	—	—	—	—	IPRE2	IPRE1	IPRE0	
IPRF	—	IPRF6	IPRF5	IPRF4	—	IPRF2	IPRF1	IPRF0	
IPRG	—	IPRG6	IPRG5	IPRG4	—	—	—	—	
IPRI	—	IPRI6	IPRI5	IPRI4	—	IPRI2	IPRI1	IPRI0	
IPRJ	—	IPRJ6	IPRJ5	IPRJ4	—	IPRJ2	IPRJ1	IPRJ0	
IPRK	—	IPRK6	IPRK5	IPRK4	—	IPRK2	IPRK1	IPRK0	
IPRM	—	IPRM6	IPRM5	IPRM4	—	—	—	—	
ABWCR	ABW7	ABW6	ABW5	ABW4	ABW3	ABW2	ABW1	ABW0	BSC
ASTCR	AST7	AST6	AST5	AST4	AST3	AST2	AST1	AST0	
WCRH	W71	W70	W61	W60	W51	W50	W41	W40	
WCRL	W31	W30	W21	W20	W11	W10	W01	W00	
BCRH	ICIS1	ICIS0	BRSTRM	BRSTS1	BRSTS0	—	—	—	
BCRL	BRLE	—	—	—	—	—	—	WAITE	
RAMER	—	—	—	—	RAMS	RAM2	RAM1	RAM0	FLASH
MAR0A	—	—	—	—	—	—	—	—	DMAC
	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
IOAR0A	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
ETCR0A	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
MAR0BH	—	—	—	—	—	—	—	—	DMAC
	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
IOAR0B	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
ETCR0B	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MAR1AH	—	—	—	—	—	—	—	—	
	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
IOAR1A	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
ETCR1A	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MAR1BH	—	—	—	—	—	—	—	—	
	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
MAR1BL	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
IOAR1B	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
ETCR1B	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
P1DR	P17DR	P16DR	P15DR	P14DR	P13DR	P12DR	P11DR	P10DR	PORT
P3DR	—	P36DR	P35DR	P34DR	P33DR	P32DR	P31DR	P30DR	
P7DR	—	—	—	P74DR	P73DR	P72DR	P71DR	P70DR	
PADR	—	—	—	—	PA3DR	PA2DR	PA1DR	PA0DR	
PBDR	PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR	
PCDR	PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR	
PDDR	PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR	
PEDR	PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR	
PFDR	PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1DR	PF0DR	
PGDR	—	—	—	PG4DR	PG3DR	PG2DR	PG1DR	PG0DR	

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
TCR_0	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_0
TMDR_0	—	—	BFB	BFA	MD3	MD2	MD1	MD0	
TIORH_0	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIORL_0	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0	
TIER_0	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	
TSR_0	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
TCNT_0	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRA_0	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRB_0	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRC_0	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRD_0	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TCR_1	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_1
TMDR_1	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_1	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_1	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_1	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNT_1	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRA_1	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRB_1	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
TCR_2	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_2
TMDR_2	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_2	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_2	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_2	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNT_2	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRA_2	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
TGRB_2	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
DMAWER	—	—	—	—	WE1B	WE1A	WE0B	WE0A	DMAC
DMACR0A*2	DTSZ	DTID	RPE	DTDIR	DTF3	DTF2	DTF1	DTF0	
DMACR0A*3	DTSZ	SAID	SAIDE	BLKDIR	BLKE	—	—	—	
DMACR0B*2	DTSZ	DTID	RPE	DTDIR	DTF3	DTF2	DTF1	DTF0	
DMACR0B*3	—	DAID	DAIDE	—	DTF3	DTF2	DTF1	DTF0	
DMACR1A*2	DTSZ	DTID	RPE	DTDIR	DTF3	DTF2	DTF1	DTF0	
DMACR1A*3	DTSZ	SAID	SAIDE	BLKDIR	BLKE	—	—	—	
DMACR1B*2	DTSZ	DTID	RPE	DTDIR	DTF3	DTF2	DTF1	DTF0	
DMACR1B*3	—	DAID	DAIDE	—	DTF3	DTF2	DTF1	DTF0	
DMABCR*2	FAE1	FAE0	—	—	DTA1B	DTA1A	DTA0B	DTA0A	
	DTE1B	DTE1A	DTE0B	DTE0A	DTIE1B	DTIE1A	DTIE0B	DTIE0A	
DMABCR*3	FAE1	FAE0	—	—	DTA1	—	DTA0	—	
	DTME1	DTE1	DTME0	DTE0	DTIE1B	DTIE1A	DTIE0B	DTIE0A	

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
TCR_0	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	TMR_0
TCR_1	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	TMR_1
TCSR_0	CMFB	CMFA	OVF	ADTE	0S3	OS2	OS1	OS0	TMR_0
TCSR_1	CMFB	CMFA	OVF	—	0S3	OS2	OS1	OS0	TMR_1
TCORA_0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	TMR_0
TCORA_1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	TMR_1
TCORB_0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	TMR_0
TCORB_1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	TMR_1
TCNT_0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	TMR_0
TCNT_1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	TMR_1
TCSR	OVF	WT/$\overline{IT}$	TME	—	—	CKS2	CKS1	CKS0	WDT
TCNT	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
RSTCSR	WOVF	RSTE	RSTS	—	—	—	—	—	
SMR_0	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI_0
BRR_0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SCR_0	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR_0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SSR_0	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
RDR_0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SCMR_0	—	—	—	—	DIR	INV	—	—	
SMR_1	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI_1
BRR_1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SCR_1	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR_1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SSR_1	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
RDR_1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SCMR_1	—	—	—	—	DIR	INV	—	—	
SMR_2	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI_2
BRR_2	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SCR_2	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR_2	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SSR_2	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
RDR_2	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SCMR_2	—	—	—	—	DIR	INV	—	—	

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
ADDRAH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D
ADDRAL	AD1	AD0	—	—	—	—	—	—	
ADDRBH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
ADDRBL	AD1	AD0	—	—	—	—	—	—	
ADDRCH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
ADDRCL	AD1	AD0	—	—	—	—	—	—	
ADDRDH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
ADDRDL	AD1	AD0	—	—	—	—	—	—	
ADCSR	ADF	ADIE	ADST	SCAN	CH3	CH2	CH1	CH0	
ADCR	TRGS1	TRGS0	—	—	CKS1	CKS0	—	—	
FLMCR1	FWE	SWE1	ESU1	PSU1	EV1	PV1	E1	P1	FLASH
FLMCR2	FLER	—	—	—	—	—	—	—	
EBR1	EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0	
EBR2	—	—	—	—	EB11	EB10	EB9	EB8	
PORT1	P17	P16	P15	P14	P13	P12	P11	P10	PORT
PORT3	—	P36	P35	P34	P33	P32	P31	P30	
PORT4	—	—	—	—	P43	P42	P41	P40	
PORT7	—	—	—	P74	P73	P72	P71	P70	
PORT9	P97	P96	—	—	—	—	—	—	
PORTA	—	—	—	—	PA3	PA2	PA1	PA0	
PORTB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	
PORTC	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	
PORTD	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0	
PORTE	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0	
PORTF	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0	
PORTG	—	—	—	PG4	PG3	PG2	PG1	PG0	

Notes: 1. nn = 00 to 22
2. Short address mode
3. Full address mode

23.3 Register States in Each Operating Mode

Register Name	Power-on Reset	Manual Reset	High-Speed	Medium-Speed	Sleep	Module Stop	Software Standby	Hardware Standby	Module
UEPIR00_0 to UEPIR22_4	—	—	—	—	—	—	—	—	USB
UCTLR	Initialized*	—	—	—	—	—	—	Initialized	
UTSTRA	Initialized*	—	—	—	—	—	—	Initialized	
UDMAR	Initialized*	—	—	—	—	—	—	Initialized	
UDRR	Initialized*	—	—	—	—	—	—	Initialized	
UTRG0	Initialized*	—	—	—	—	—	—	Initialized	
UTRG1	Initialized*	—	—	—	—	—	—	Initialized	
UFCLR0	Initialized*	—	—	—	—	—	—	Initialized	
UFCLR1	Initialized*	—	—	—	—	—	—	Initialized	
UESTL0	Initialized*	—	—	—	—	—	—	Initialized	
UESTL1	Initialized*	—	—	—	—	—	—	Initialized	
UEDR0s	—	—	—	—	—	—	—	—	
UEDR0i	Initialized*	—	—	—	—	—	—	Initialized	
UEDR0o	—	—	—	—	—	—	—	—	
UEDR1i	Initialized*	—	—	—	—	—	—	Initialized	
UEDR2i	Initialized*	—	—	—	—	—	—	Initialized	
UEDR2o	—	—	—	—	—	—	—	—	
UEDR3i	Initialized*	—	—	—	—	—	—	Initialized	
UEDR3o	—	—	—	—	—	—	—	—	
UEDR4i	Initialized*	—	—	—	—	—	—	Initialized	
UEDR4o	—	—	—	—	—	—	—	Initialized	
UEDR5i	Initialized*	—	—	—	—	—	—	Initialized	
UESZ0o	—	—	—	—	—	—	—	—	
UESZ2o	—	—	—	—	—	—	—	—	
UESZ3o	—	—	—	—	—	—	—	—	
UESZ4o	—	—	—	—	—	—	—	—	
UIFR0	Initialized*	—	—	—	—	—	—	Initialized	
UIFR1	Initialized*	—	—	—	—	—	—	Initialized	
UIFR2	Initialized*	—	—	—	—	—	—	Initialized	
UIFR3	Initialized*	—	—	—	—	—	—	Initialized	
UIER0	Initialized*	—	—	—	—	—	—	Initialized	

Note: * The USB registers are no initialized by a power-on reset triggered by the WDT.

Register Name	Power-on Reset	Manual Reset	High-Speed	Medium-Speed	Sleep	Module Stop	Software Standby	Hardware Standby	Module
UIER1	Initialized*	—	—	—	—	—	—	Initialized	USB
UIER2	Initialized*	—	—	—	—	—	—	Initialized	
UIER3	Initialized*	—	—	—	—	—	—	Initialized	
UISR0	Initialized*	—	—	—	—	—	—	Initialized	
UISR1	Initialized*	—	—	—	—	—	—	Initialized	
UISR2	Initialized*	—	—	—	—	—	—	Initialized	
UISR3	Initialized*	—	—	—	—	—	—	Initialized	
UDSR	Initialized*	—	—	—	—	—	—	Initialized	
UCVR	Initialized*	—	—	—	—	—	—	Initialized	
UTSRH	Initialized*	—	—	—	—	—	—	Initialized	
UTSRL	Initialized*	—	—	—	—	—	—	Initialized	
UTSTR0	Initialized*	—	—	—	—	—	—	Initialized	
UTSTR1	Initialized*	—	—	—	—	—	—	Initialized	
UTSTR2	Initialized*	—	—	—	—	—	—	Initialized	
UTSTRB	Initialized*	—	—	—	—	—	—	Initialized	
UTSTRC	Initialized*	—	—	—	—	—	—	Initialized	
UTSTRD	Initialized*	—	—	—	—	—	—	Initialized	
UTSTRE	Initialized*	—	—	—	—	—	—	Initialized	
UTSTRF	Initialized*	—	—	—	—	—	—	Initialized	
MRA	—	—	—	—	—	—	—	—	DTC
SAR	—	—	—	—	—	—	—	—	
MRB	—	—	—	—	—	—	—	—	
DAR	—	—	—	—	—	—	—	—	
CRA	—	—	—	—	—	—	—	—	
CRB	—	—	—	—	—	—	—	—	
DADR_0	Initialized	Initialized	—	—	—	—	—	Initialized	D/A
DADR_1	Initialized	Initialized	—	—	—	—	—	Initialized	
DACR	Initialized	Initialized	—	—	—	—	—	Initialized	
SCRX	Initialized	Initialized	—	—	—	—	—	Initialized	FLASH
SBYCR	Initialized	Initialized	—	—	—	—	—	Initialized	SYSTEM
SYSCR	Initialized	Initialized	—	—	—	—	—	Initialized	
SCKCR	Initialized	Initialized	—	—	—	—	—	Initialized	
MDCR	Initialized	—	—	—	—	—	—	Initialized	
MSTPCRA	Initialized	Initialized	—	—	—	—	—	Initialized	
MSTPCRB	Initialized	Initialized	—	—	—	—	—	Initialized	SYSTEM
MSTPCRC	Initialized	Initialized	—	—	—	—	—	Initialized	

Note: * The USB registers are no initialized by a power-on reset triggered by the WDT.

Register Name	Power-on Reset	Manual Reset	High-Speed	Medium-Speed	Sleep	Module Stop	Software Standby	Hardware Standby	Module
PFCR	Initialized	—	—	—	—	—	—	Initialized	BSC
LPWRCR	Initialized	—	—	—	—	—	—	Initialized	SYSTEM
SEMR_0	Initialized	Initialized	—	—	—	—	—	Initialized	SCI_0
ISCRH	Initialized	Initialized	—	—	—	—	—	Initialized	INT
ISCRL	Initialized	Initialized	—	—	—	—	—	Initialized	
IER	Initialized	Initialized	—	—	—	—	—	Initialized	
ISR	Initialized	Initialized	—	—	—	—	—	Initialized	
DTCERA	Initialized	Initialized	—	—	—	—	—	Initialized	DTC
DTCERB	Initialized	Initialized	—	—	—	—	—	Initialized	
DTCERC	Initialized	Initialized	—	—	—	—	—	Initialized	
DTCERD	Initialized	Initialized	—	—	—	—	—	Initialized	
DTCERE	Initialized	Initialized	—	—	—	—	—	Initialized	
DTCERF	Initialized	Initialized	—	—	—	—	—	Initialized	
DTVECR	Initialized	Initialized	—	—	—	—	—	Initialized	
P1DDR	Initialized	—	—	—	—	—	—	Initialized	PORT
P3DDR	Initialized	—	—	—	—	—	—	Initialized	
P7DDR	Initialized	—	—	—	—	—	—	Initialized	
PADDR	Initialized	—	—	—	—	—	—	Initialized	
PBDDR	Initialized	—	—	—	—	—	—	Initialized	
PCDDR	Initialized	—	—	—	—	—	—	Initialized	
PDDDR	Initialized	—	—	—	—	—	—	Initialized	
PEDDR	Initialized	—	—	—	—	—	—	Initialized	
PFDDR	Initialized	—	—	—	—	—	—	Initialized	
PGDDR	Initialized	—	—	—	—	—	—	Initialized	
PAPCR	Initialized	—	—	—	—	—	—	Initialized	
PBPCR	Initialized	—	—	—	—	—	—	Initialized	
PCPCR	Initialized	—	—	—	—	—	—	Initialized	
PDPCR	Initialized	—	—	—	—	—	—	Initialized	
PEPCR	Initialized	—	—	—	—	—	—	Initialized	
P3ODR	Initialized	—	—	—	—	—	—	Initialized	
PAODR	Initialized	—	—	—	—	—	—	Initialized	

Register Name	Power-on Reset	Manual Reset	High-Speed	Medium-Speed	Sleep	Module Stop	Software Standby	Hardware Standby	Module
TSTR	Initialized	Initialized	—	—	—	—	—	Initialized	TPU
TSYR	Initialized	Initialized	—	—	—	—	—	Initialized	
IPRA	Initialized	Initialized	—	—	—	—	—	Initialized	INT
IPRB	Initialized	Initialized	—	—	—	—	—	Initialized	
IPRC	Initialized	Initialized	—	—	—	—	—	Initialized	
IPRD	Initialized	Initialized	—	—	—	—	—	Initialized	
IPRE	Initialized	Initialized	—	—	—	—	—	Initialized	
IPRF	Initialized	Initialized	—	—	—	—	—	Initialized	
IPRG	Initialized	Initialized	—	—	—	—	—	Initialized	
IPRI	Initialized	Initialized	—	—	—	—	—	Initialized	
IPRJ	Initialized	Initialized	—	—	—	—	—	Initialized	
IPRK	Initialized	Initialized	—	—	—	—	—	Initialized	
IPRM	Initialized	Initialized	—	—	—	—	—	Initialized	
ABWCR	Initialized	—	—	—	—	—	—	Initialized	BSC
ASTCR	Initialized	—	—	—	—	—	—	Initialized	
WCRH	Initialized	—	—	—	—	—	—	Initialized	
WCRL	Initialized	—	—	—	—	—	—	Initialized	
BCRH	Initialized	—	—	—	—	—	—	Initialized	
BCRL	Initialized	—	—	—	—	—	—	Initialized	
RAMER	Initialized	—	—	—	—	—	—	Initialized	FLASH
MAR0A	—	—	—	—	—	—	—	—	DMAC
IOAR0A	—	—	—	—	—	—	—	—	
ETCR0A	—	—	—	—	—	—	—	—	
MAR0B	—	—	—	—	—	—	—	—	
IOAR0B	—	—	—	—	—	—	—	—	
ETCR0B	—	—	—	—	—	—	—	—	
MAR1A	—	—	—	—	—	—	—	—	
IOAR1A	—	—	—	—	—	—	—	—	
ETCR1A	—	—	—	—	—	—	—	—	
MAR1B	—	—	—	—	—	—	—	—	
IOAR1B	—	—	—	—	—	—	—	—	
ETCR1B	—	—	—	—	—	—	—	—	

Register Name	Power-on Reset	Manual Reset	High-Speed	Medium-Speed	Sleep	Module Stop	Software Standby	Hardware Standby	Module
P1DR	Initialized	—	—	—	—	—	—	Initialized	PORT
P3DR	Initialized	—	—	—	—	—	—	Initialized	
P7DR	Initialized	—	—	—	—	—	—	Initialized	
PADR	Initialized	—	—	—	—	—	—	Initialized	
PBDR	Initialized	—	—	—	—	—	—	Initialized	
PCDR	Initialized	—	—	—	—	—	—	Initialized	
PDDR	Initialized	—	—	—	—	—	—	Initialized	
PEDR	Initialized	—	—	—	—	—	—	Initialized	
PFDR	Initialized	—	—	—	—	—	—	Initialized	
PGDR	Initialized	—	—	—	—	—	—	Initialized	
TCR_0	Initialized	Initialized	—	—	—	—	—	Initialized	TPU_0
TMDR_0	Initialized	Initialized	—	—	—	—	—	Initialized	
TIORH_0	Initialized	Initialized	—	—	—	—	—	Initialized	
TIORL_0	Initialized	Initialized	—	—	—	—	—	Initialized	
TIER_0	Initialized	Initialized	—	—	—	—	—	Initialized	
TSR_0	Initialized	Initialized	—	—	—	—	—	Initialized	
TCNT_0	Initialized	Initialized	—	—	—	—	—	Initialized	
TGRA_0	Initialized	Initialized	—	—	—	—	—	Initialized	
TGRB_0	Initialized	Initialized	—	—	—	—	—	Initialized	
TGRC_0	Initialized	Initialized	—	—	—	—	—	Initialized	
TGRD_0	Initialized	Initialized	—	—	—	—	—	Initialized	
TCR_1	Initialized	Initialized	—	—	—	—	—	Initialized	TPU_1
TMDR_1	Initialized	Initialized	—	—	—	—	—	Initialized	
TIOR_1	Initialized	Initialized	—	—	—	—	—	Initialized	
TIER_1	Initialized	Initialized	—	—	—	—	—	Initialized	
TSR_1	Initialized	Initialized	—	—	—	—	—	Initialized	
TCNT_1	Initialized	Initialized	—	—	—	—	—	Initialized	
TGRA_1	Initialized	Initialized	—	—	—	—	—	Initialized	
TGRB_1	Initialized	Initialized	—	—	—	—	—	Initialized	
TCR_2	Initialized	Initialized	—	—	—	—	—	Initialized	TPU_2
TMDR_2	Initialized	Initialized	—	—	—	—	—	Initialized	
TIOR_2	Initialized	Initialized	—	—	—	—	—	Initialized	
TIER_2	Initialized	Initialized	—	—	—	—	—	Initialized	
TSR_2	Initialized	Initialized	—	—	—	—	—	Initialized	
TCNT_2	Initialized	Initialized	—	—	—	—	—	Initialized	
TGRA_2	Initialized	Initialized	—	—	—	—	—	Initialized	
TGRB_2	Initialized	Initialized	—	—	—	—	—	Initialized	

Register Name	Power-on Reset	Manual Reset	High-Speed	Medium-Speed	Sleep	Module Stop	Software Standby	Hardware Standby	Module
DMAWER	Initialized	Initialized	—	—	—	—	—	Initialized	DMAC
DMACR0A	Initialized	Initialized	—	—	—	—	—	Initialized	
DMACR0B	Initialized	Initialized	—	—	—	—	—	Initialized	
DMACR1A	Initialized	Initialized	—	—	—	—	—	Initialized	
DMACR1B	Initialized	Initialized	—	—	—	—	—	Initialized	
DMABCR	Initialized	Initialized	—	—	—	—	—	Initialized	
TCR_0	Initialized	Initialized	—	—	—	—	—	Initialized	TMR_0
TCR_1	Initialized	Initialized	—	—	—	—	—	Initialized	TMR_1
TCSR_0	Initialized	Initialized	—	—	—	—	—	Initialized	TMR_0
TCSR_1	Initialized	Initialized	—	—	—	—	—	Initialized	TMR_1
TCORA_0	Initialized	Initialized	—	—	—	—	—	Initialized	TMR_0
TCORA_1	Initialized	Initialized	—	—	—	—	—	Initialized	TMR_1
TCORB_0	Initialized	Initialized	—	—	—	—	—	Initialized	TMR_0
TCORB_1	Initialized	Initialized	—	—	—	—	—	Initialized	TMR_1
TCNT_0	Initialized	Initialized	—	—	—	—	—	Initialized	TMR_0
TCNT_1	Initialized	Initialized	—	—	—	—	—	Initialized	TMR_1
TCSR	Initialized	Initialized	—	—	—	—	—	Initialized	WDT
TCNT	Initialized	Initialized	—	—	—	—	—	Initialized	
RSTCSR	Initialized	Initialized	—	—	—	—	—	Initialized	
SMR_0	Initialized	Initialized	—	—	—	—	—	Initialized	SCI_0
BRR_0	Initialized	Initialized	—	—	—	—	—	Initialized	
SCR_0	Initialized	Initialized	—	—	—	—	—	Initialized	
TDR_0	Initialized	Initialized	—	—	—	Initialized	Initialized	Initialized	
SSR_0	Initialized	Initialized	—	—	—	Initialized	Initialized	Initialized	
RDR_0	Initialized	Initialized	—	—	—	Initialized	Initialized	Initialized	
SCMR_0	Initialized	Initialized	—	—	—	—	—	Initialized	
SMR_1	Initialized	Initialized	—	—	—	—	—	Initialized	SCI_1
BRR_1	Initialized	Initialized	—	—	—	—	—	Initialized	
SCR_1	Initialized	Initialized	—	—	—	—	—	Initialized	
TDR_1	Initialized	Initialized	—	—	—	Initialized	Initialized	Initialized	
SSR_1	Initialized	Initialized	—	—	—	Initialized	Initialized	Initialized	
RDR_1	Initialized	Initialized	—	—	—	Initialized	Initialized	Initialized	
SCMR_1	Initialized	Initialized	—	—	—	—	—	Initialized	

Register Name	Power-on Reset	Manual Reset	High-Speed	Medium-Speed	Sleep	Module Stop	Software Standby	Hardware Standby	Module
SMR_2	Initialized	Initialized	—	—	—	—	—	Initialized	SCI_2
BRR_2	Initialized	Initialized	—	—	—	—	—	Initialized	
SCR_2	Initialized	Initialized	—	—	—	—	—	Initialized	
TDR_2	Initialized	Initialized	—	—	—	Initialized	Initialized	Initialized	
SSR_2	Initialized	Initialized	—	—	—	Initialized	Initialized	Initialized	
RDR_2	Initialized	Initialized	—	—	—	Initialized	Initialized	Initialized	
SCMR_2	Initialized	Initialized	—	—	—	—	—	Initialized	
ADDRAH	Initialized	Initialized	—	—	—	Initialized	Initialized	Initialized	A/D
ADDRAL	Initialized	Initialized	—	—	—	Initialized	Initialized	Initialized	
ADDRBH	Initialized	Initialized	—	—	—	Initialized	Initialized	Initialized	
ADDRBL	Initialized	Initialized	—	—	—	Initialized	Initialized	Initialized	
ADDRCH	Initialized	Initialized	—	—	—	Initialized	Initialized	Initialized	
ADDRCL	Initialized	Initialized	—	—	—	Initialized	Initialized	Initialized	
ADDRDH	Initialized	Initialized	—	—	—	Initialized	Initialized	Initialized	
ADDRDL	Initialized	Initialized	—	—	—	Initialized	Initialized	Initialized	
ADCSR	Initialized	Initialized	—	—	—	Initialized	Initialized	Initialized	
ADCR	Initialized	Initialized	—	—	—	Initialized	Initialized	Initialized	
FLMCR1	Initialized	—	—	—	—	—	Initialized	Initialized	FLASH
FLMCR2	Initialized	—	—	—	—	—	Initialized	Initialized	
EBR1	Initialized	—	—	—	—	—	Initialized	Initialized	
EBR2	Initialized	—	—	—	—	—	Initialized	Initialized	
PORT1	—	—	—	—	—	—	—	—	PORT
PORT3	—	—	—	—	—	—	—	—	
PORT4	—	—	—	—	—	—	—	—	
PORT7	—	—	—	—	—	—	—	—	
PORT9	—	—	—	—	—	—	—	—	
PORTA	—	—	—	—	—	—	—	—	
PORTB	—	—	—	—	—	—	—	—	
PORTC	—	—	—	—	—	—	—	—	
PORTD	—	—	—	—	—	—	—	—	
PORTE	—	—	—	—	—	—	—	—	
PORTF	—	—	—	—	—	—	—	—	
PORTG	—	—	—	—	—	—	—	—	

Note: — is not initialized.

Section 24 Electrical Characteristics (H8S/2215)

24.1 Absolute Maximum Ratings

Table 24.1 lists the absolute maximum ratings.

Table 24.1 Absolute Maximum Ratings

Item	Symbol	Value	Unit
Power supply voltage	V_{CC}, $PLLV_{CC}$, DrV_{CC}	–0.3 to +4.3	V
Input voltage (except ports 4 and 9)	V_{in}	–0.3 to V_{CC} +0.3	V
Input voltage (ports 4 and 9)	V_{in}	–0.3 to AV_{CC} +0.3	V
Reference 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	°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.

Note: * The operating temperature ranges for flash memory programming/erasing are T_a = –20°C to +75°C.

24.2 Power Supply Voltage and Operating Frequency Range

Power supply voltage and operating frequency ranges (shaded areas) are shown in figure 24.1.

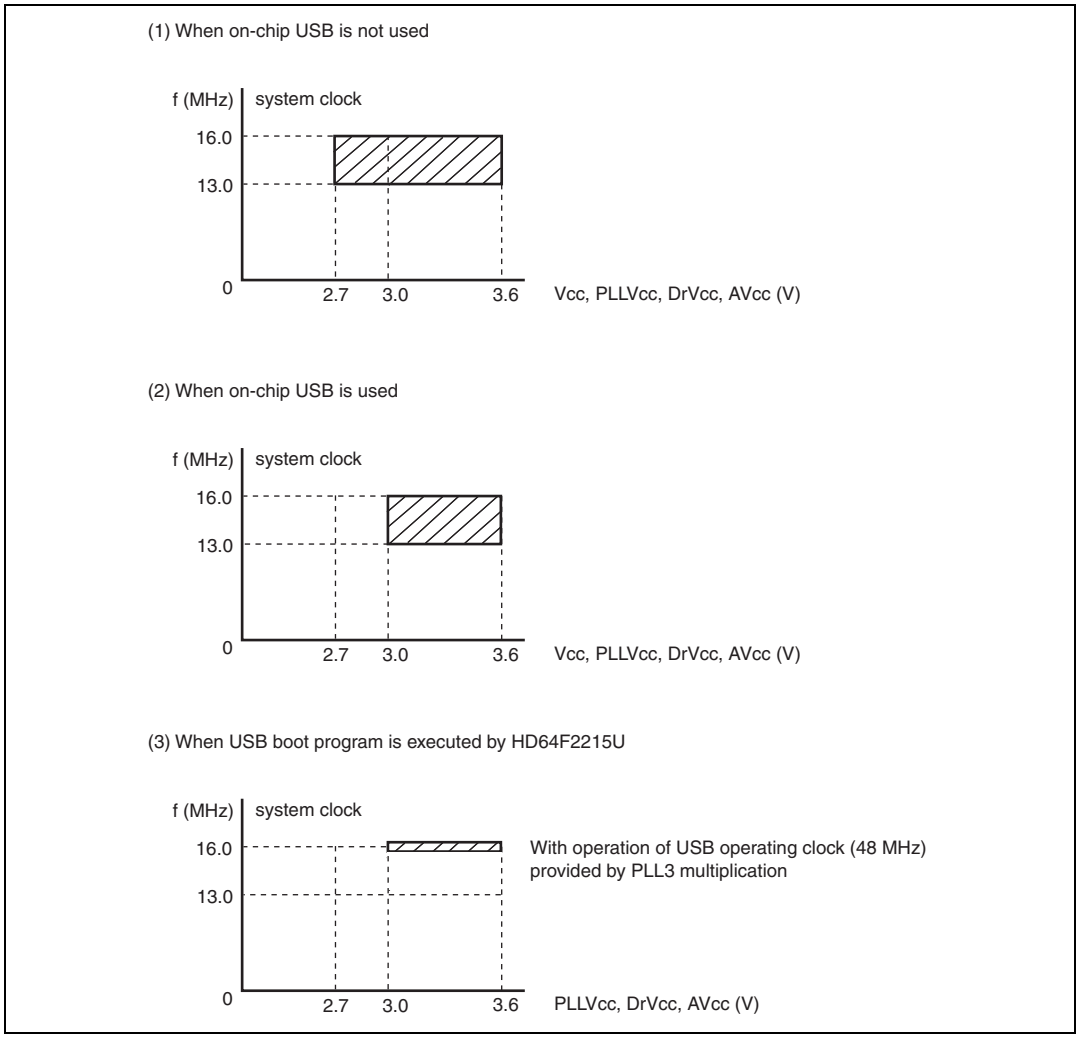


Figure 24.1 Power Supply Voltage and Operating Ranges

24.3 DC Characteristics

Table 24.2 lists the DC characteristics. Table 24.3 lists the permissible output currents.

Table 24.2 DC Characteristics

Conditions: V_{CC} = $PLLV_{CC}$ = DrV_{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} = $PLLV_{SS}$ = DrV_{SS} = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)*1

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Schmitt trigger input voltage	$\overline{IRQ0}$ to $\overline{IRQ5}$, $\overline{IRQ7}$	V_T^-	$V_{CC} \times 0.2$	—	—	V	
		V_T^+	—	—	$V_{CC} \times 0.8$	V	
		$V_T^+ - V_T^-$	$V_{CC} \times 0.05$	—	—	V	
Input high voltage	$\overline{RES}$, $\overline{STBY}$, NMI, MD2 to MD0, $\overline{TRST}$, TCK, TMS, TDI, VBUS, $\overline{UBPM}$, FWE*5	V_{IH}	$V_{CC} \times 0.9$	—	$V_{CC} + 0.3$	V	
	EXTAL, EXTAL48, Ports 1, 3, 7, and A to G		$V_{CC} \times 0.8$	—	$V_{CC} + 0.3$	V	
	Ports 4 and 9		$V_{CC} \times 0.8$	—	$AV_{CC} + 0.3$	V	
Input low voltage	$\overline{RES}$, $\overline{STBY}$, MD2 to MD0, $\overline{TRST}$, TCK, TMS, TDI, VBUS, $\overline{UBPM}$, FWE*5	V_{IL}	–0.3	—	$V_{CC} \times 0.1$	V	
	EXTAL, EXTAL48, NMI, Ports 1, 3, 4, 7, 9, and A to G		–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

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Output low voltage	All output pins	V_{OL}	—	—	0.4	V	I_{OH} = 0.4 mA
			—	—	0.4	V	I_{OL} = 0.8 mA
Input leakage current	$\overline{RES}$, $\overline{STBY}$, NMI, MD2 to MD0, FWE*5, VBUS,$\overline{UBPM}$	$\lvert I_{in} \rvert$	—	—	1.0	µA	V_{in} = 0.5 V to V_{CC} – 0.5 V
	Ports 4, 9	$\lvert I_{in} \rvert$	—	—	1.0	µA	V_{in} = 0.5 V to AV_{CC} – 0.5 V
3-state leak current (off status)	Ports 1, 3, 7, A to G	$\lvert I_{TSI} \rvert$	—	—	1.0	µA	V_{in} = 0.5 V to V_{CC} – 0.5 V
Input pull-up MOS current	Ports A to E	$-I_P$	10	—	300	µA	V_{in} = 0 V
Input capacity	$\overline{RES}$, NMI	C_{in}	—	—	30	pF	V_{in} = 0 V
	All input pins other than $\overline{RES}$ and NMI		—	—	15	pF	f = 1 MHz T_a = 25°C
Current dissipation*2	Normal operation (USB halts)	I_{CC}*3	—	27 V_{CC} = 3.3 V	40 V_{CC} = 3.6 V	mA	f = 16 MHz
	Normal operation (USB operates)		—	36 V_{CC} = 3.3 V	50 V_{CC} = 3.6 V	mA	f = 16 MHz When PLL is used
	Sleep mode		—	22 V_{CC} = 3.3 V	35 V_{CC} = 3.6 V	mA	f = 16 MHz When USB and PLL are halted
	All modules stopped		—	16 V_{CC} = 3.3 V	—	mA	f = 16 MHz (reference value)
	Standby mode*4		—	1.0	10	µA	$T_a \leq$ 50°C
			—	—	50	µA	50°C < T_a
Analog power supply current	During A/D conversion	AI_{CC}	—	0.5	1.5	mA	AV_{CC} = 3.3 V
	Idle		—	0.01	5.0	µA	

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Reference power supply current	During A/D conversion	AI_{CC}	—	1.3	2.5	mA	V_{ref} = 3.3 V
	Idle		—	0.01	5.0	µA	
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Notes: 1. If the A/D or D/A converter is not used, the AVCC, V_{ref}, and AVSS pins should not be open. Even if the A/D or D/A converter is not used, connect the AVCC and V_{ref} pins to V_{CC} and the AVSS pin to V_{SS}, respectively. In this case, the V_{ref} level should be the AVCC level or less.

2. Current dissipation values are for V_{IH} (min.) = V_{CC} – 0.2 V and V_{IL} (max.) = 0.2 V, with all output pins unloaded and the on-chip MOS pull-up transistors in the off state.

3. I_{CC} depends on V_{CC} and f as follows:
   I_{CC} (max.) = 1.0 (mA) + 0.67 (mA/(MHz x V)) × V_{CC} × f (normal operation, USB halted)
   I_{CC} (max.) = 1.0 (mA) + 0.85 (mA/(MHz x V)) × V_{CC} × f (normal operation, USB operated)
   I_{CC} (max.) = 1.0 (mA) + 0.59 (mA/(MHz x V)) × V_{CC} × f (sleep mode)

4. The values are for $V_{RAM} \leq V_{CC} < 2.7$ V, V_{IH} (min.) = V_{CC} × 0.9, and V_{IL} (max.) = 0.3 V.

5. The FWE pin is effective only in the F-ZTAT version.

Table 24.3 Permissible Output Currents

Conditions: V_{CC} = $PLLV_{CC}$ = DrV_{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} = $PLLV_{SS}$ = DrV_{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	V_{CC} = 2.7 to 3.6 V	I_{OL}	—	—	1.0	mA
Permissible output low current (total)	Total of all output pins	V_{CC} = 2.7 to 3.6 V	$\sum I_{OL}$	—	—	60	mA
Permissible output high current (per pin)	All output pins	V_{CC} = 2.7 to 3.6 V	$-I_{OH}$	—	—	1.0	mA
Permissible output high current (total)	Total of all output pins	V_{CC} = 2.7 to 3.6 V	$\sum -I_{OH}$	—	—	30	mA

Note: * To protect chip reliability, do not exceed the output current values in table 24.3.

24.4 AC Characteristics

Figure 24.2 shows, the test conditions for the AC characteristics.

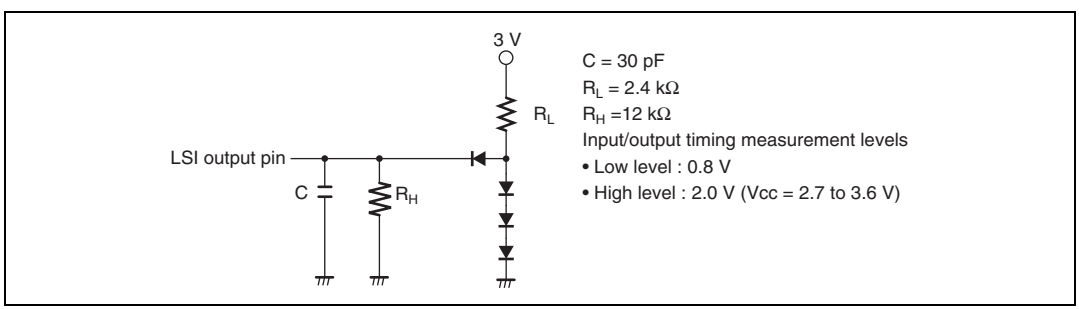


Figure 24.2 Output Load Circuit

24.4.1 Clock Timing

Table 24.4 lists the clock timing

Table 24.4 Clock Timing

Conditions: V_{CC} = $PLLV_{CC}$ = DrV_{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} = $PLLV_{SS}$ = DrV_{SS} = AV_{SS} = 0 V, ϕ = 13 MHz to 16 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}	62.5	76.9	ns	Figure 24.3
Clock high pulse width	t_{CH}	20	—	ns	
Clock low pulse width	t_{CL}	20	—	ns	
Clock rise time	t_{Cr}	—	10	ns	
Clock fall time	t_{Cf}	—	10	ns	
Oscillation stabilization time at reset (crystal)	t_{OSC1}	20	—	ms	Figure 24.4
Oscillation stabilization time in software standby (crystal)	t_{OSC2}	8	—	ms	C_{L1} = C_{L2} = 10 pF to 22 pF in figure 21.2 V_{CC} = 2.7 V to 3.6 V in figure 22.3
		4	—	ms	C_{L1} = C_{L2} = 10 pF to 15 pF in figure 21.2 V_{CC} = 3.0 V to 3.6 V in figure 22.3
External clock output stabilization delay time	t_{DEXT}	500	—	µs	Figure 24.4
USB operating clock (48 MHz) stabilization time	t_{OSC3}	8	—	ms	V_{CC} = 3.0 V to 3.6 V
USB operating clock (48 MHz) oscillator frequency	f_{48}	48		MHz	
USB operating clock (48 MHz) cycle time	f_{48}	20.8		ns	

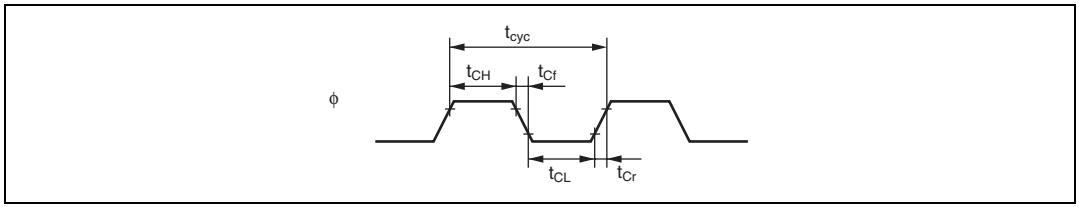


Figure 24.3 System Clock Timing

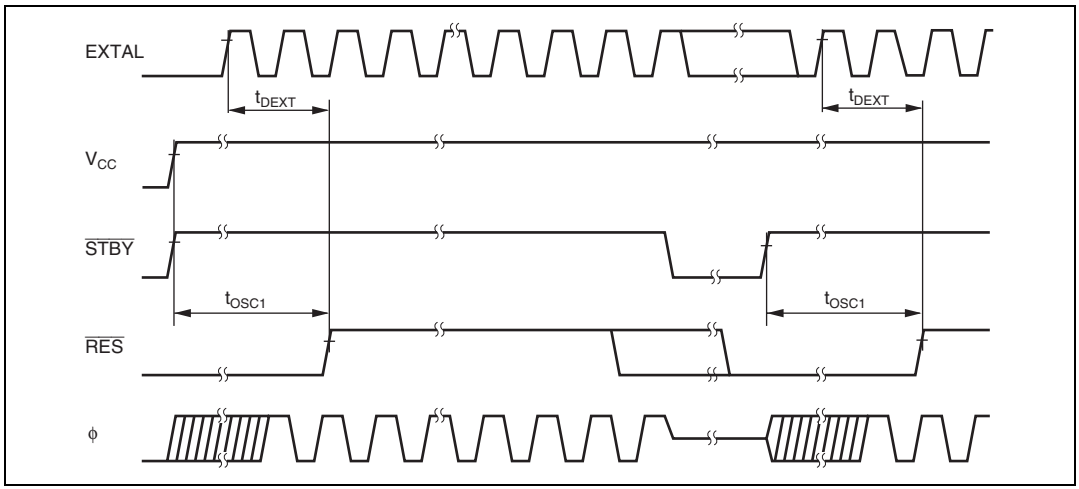


Figure 24.4 Oscillation Stabilization Timing

24.4.2 Control Signal Timing

Table 24.5 lists the control signal timing.

Table 24.5 Control Signal Timing

Conditions: V_{CC} = $PLLV_{CC}$ = DrV_{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} = $PLLV_{SS}$ = DrV_{SS} = AV_{SS} = 0 V, ϕ = 13 MHz to 16 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
$\overline{RES}$ setup time	t_{RESS}	250	—	ns	Figure 24.5
$\overline{RES}$ pulse width	t_{RESW}	20	—	t_{cyc}	
$\overline{MRES}$ setup time	t_{MRESS}	250	—	ns	
$\overline{MRES}$ pulse width	t_{MRESW}	20	—	t_{cyc}	
NMI setup time	t_{NMIS}	250	—	ns	Figure 24.6
NMI hold time	t_{NMIH}	10	—	ns	
NMI pulse width (exiting software standby mode)	t_{NMIW}	200	—	ns	
$\overline{IRQ}$ setup time	t_{IRQS}	250	—	ns	
$\overline{IRQ}$ hold time	t_{IRQH}	10	—	ns	
$\overline{IRQ}$ pulse width (exiting software standby mode)	t_{IRQW}	200	—	ns	

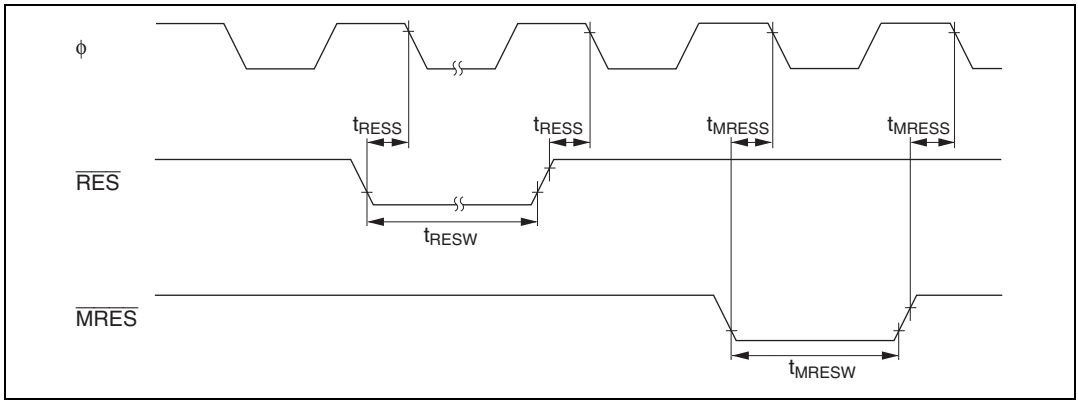


Figure 24.5 Reset Input Timing

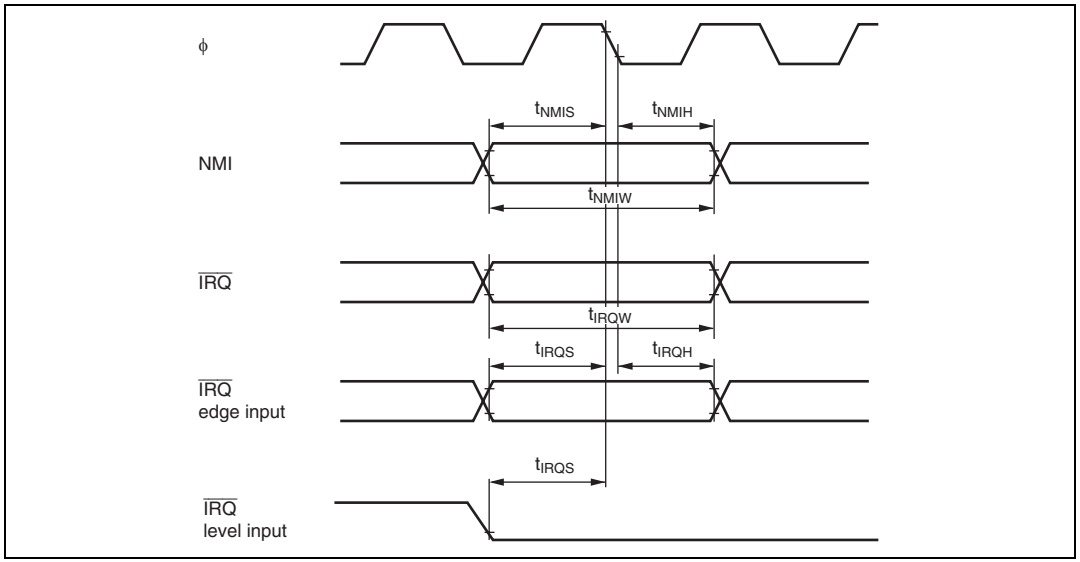


Figure 24.6 Interrupt Input Timing

24.4.3 Bus Timing

Table 24.6 shows, Bus Timing.

Table 24.6 Bus Timing

Conditions: V_{CC} = $PLLV_{CC}$ = DrV_{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} = $PLLV_{SS}$ = DrV_{SS} = AV_{SS} = 0 V, ϕ = 13 MHz to 16 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}	—	50	ns	Figures 24.7, 24.8, 24.10
Address setup time	t_{AS}	$0.5 \times t_{cyc} - 30$	—	ns	
Address hold time	t_{AH}	$0.5 \times t_{cyc} - 15$	—	ns	
$\overline{CS}$ delay time	t_{CSD}	—	50	ns	Figures 24.7, 24.8
$\overline{AS}$ delay time	t_{ASD}	—	50	ns	Figures 24.7, 24.8, 24.10
$\overline{RD}$ delay time 1	t_{RSD1}	—	50	ns	Figures 24.7, 24.8
$\overline{RD}$ delay time 2	t_{RSD2}	—	50	ns	Figures 24.7, 24.8, 24.10
Read data setup time	t_{RDS}	30	—	ns	
Read data hold time	t_{RDH}	0	—	ns	
Read data access time 2	t_{ACC2}	—	$1.5 \times t_{cyc} - 65$	ns	Figure 24.7
Read data access time 3	t_{ACC3}	—	$2.0 \times t_{cyc} - 65$	ns	Figures 24.7, 24.10
Read data access time 4	t_{ACC4}	—	$2.5 \times t_{cyc} - 65$	ns	Figure 24.8
Read data access time 5	t_{ACC5}	—	$3.0 \times t_{cyc} - 65$	ns	
$\overline{WR}$ delay time 1	t_{WRD1}	—	50	ns	
$\overline{WR}$ delay time 2	t_{WRD2}	—	50	ns	Figures 24.7, 24.8
$\overline{WR}$ pulse width 1	t_{WSW1}	$1.0 \times t_{cyc} - 30$	—	ns	Figure 24.7
$\overline{WR}$ pulse width 2	t_{WSW2}	$1.5 \times t_{cyc} - 30$	—	ns	Figure 24.8
Write data delay time	t_{WDD}	—	50	ns	Figures 24.7, 24.8
Write data setup time	t_{WDS}	$0.5 \times t_{cyc} - 30$	—	ns	Figure 24.8
Write data hold time	t_{WDH}	$0.5 \times t_{cyc} - 15$	—	ns	Figures 24.7, 24.8
$\overline{WAIT}$ setup time	t_{WTS}	50	—	ns	Figure 24.9
$\overline{WAIT}$ hold time	t_{WTH}	10	—	ns	
$\overline{BREQ}$ setup time	t_{BRQS}	50	—	ns	Figure 24.11
$\overline{BACK}$ delay time	t_{BACD}	—	50	ns	
Bus-floating time	t_{BZD}	—	80	ns	

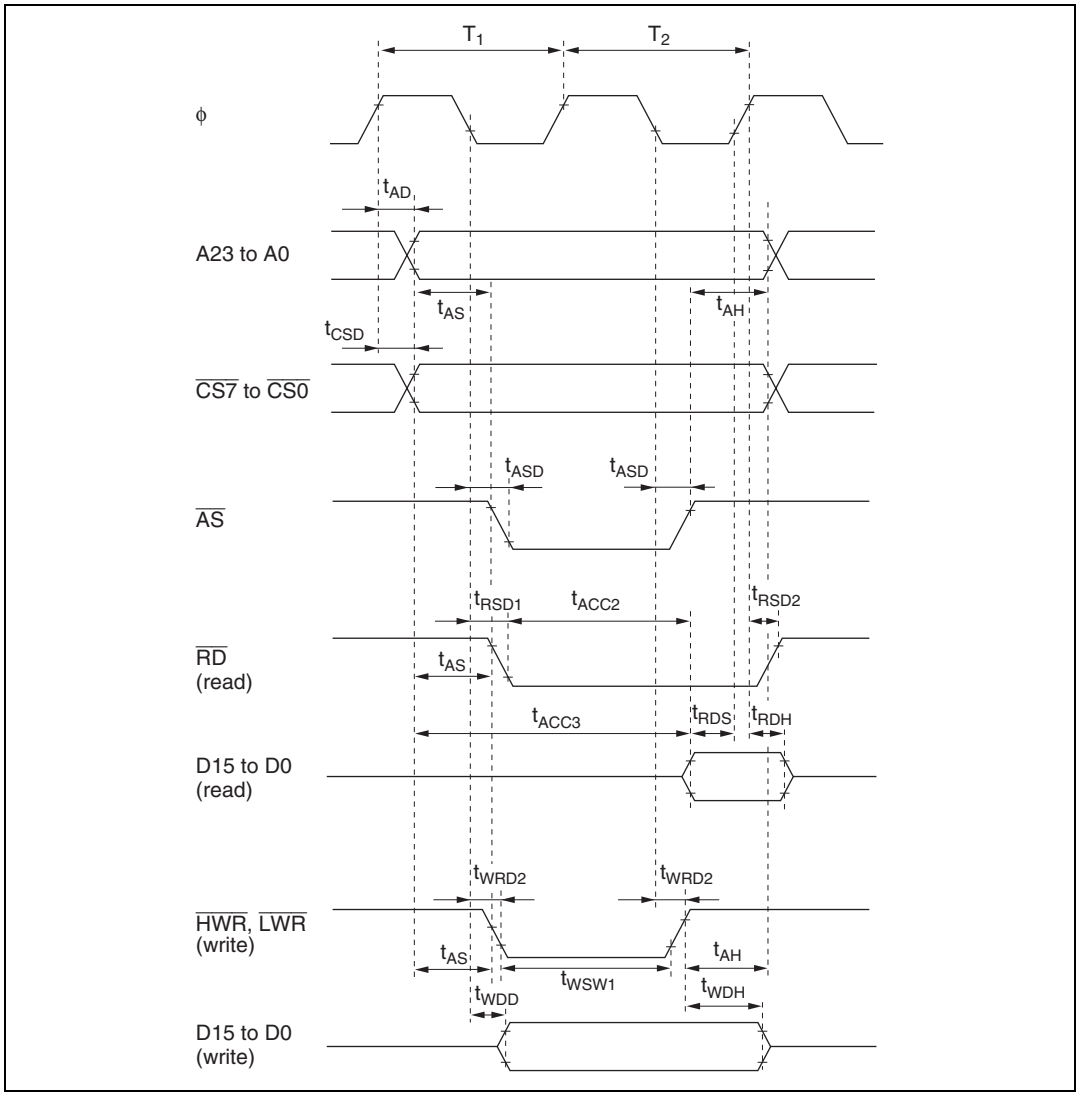


Figure 24.7 Basic Bus Timing (Two-State Access)

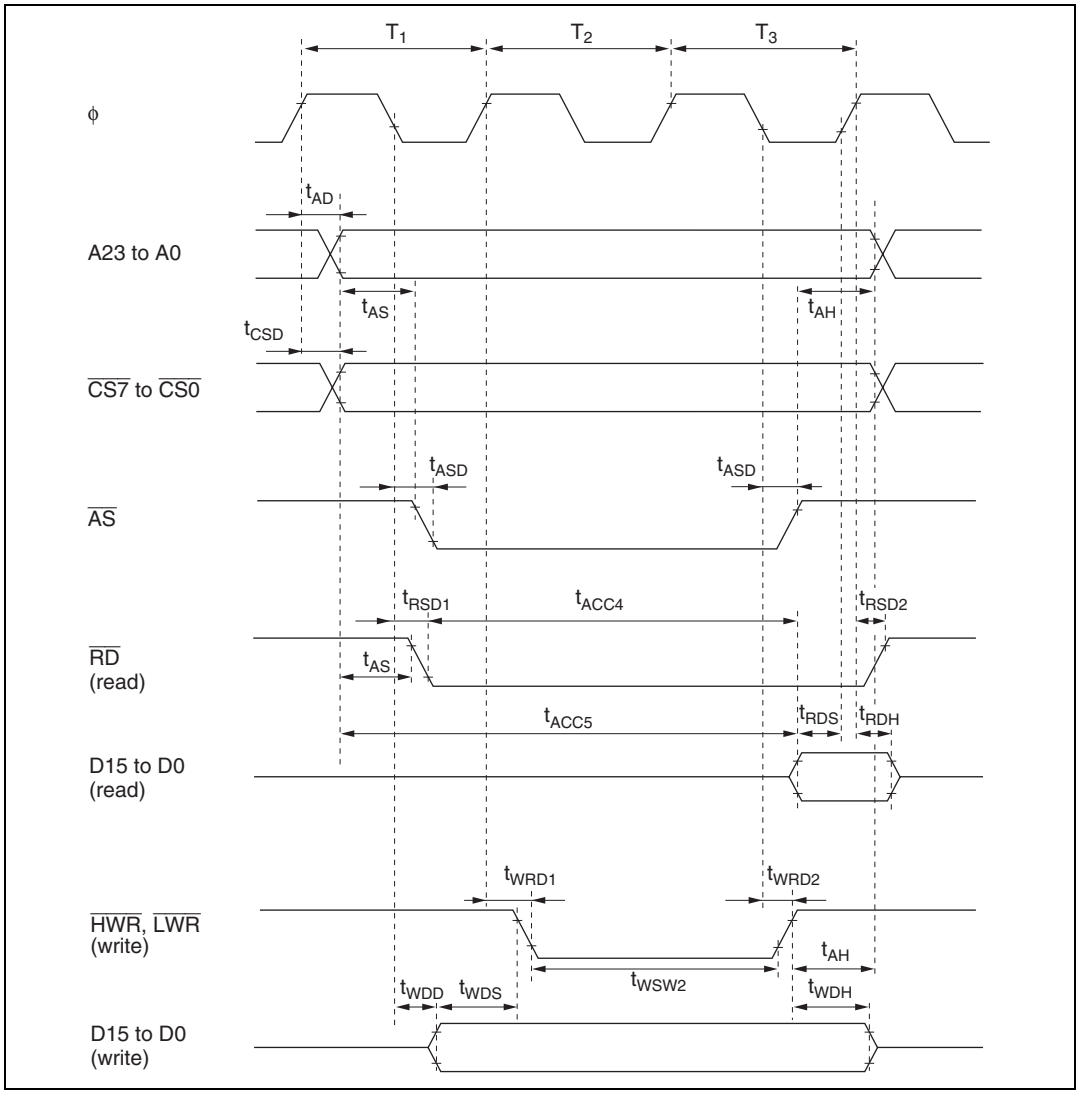


Figure 24.8 Basic Bus Timing (Three-State Access)

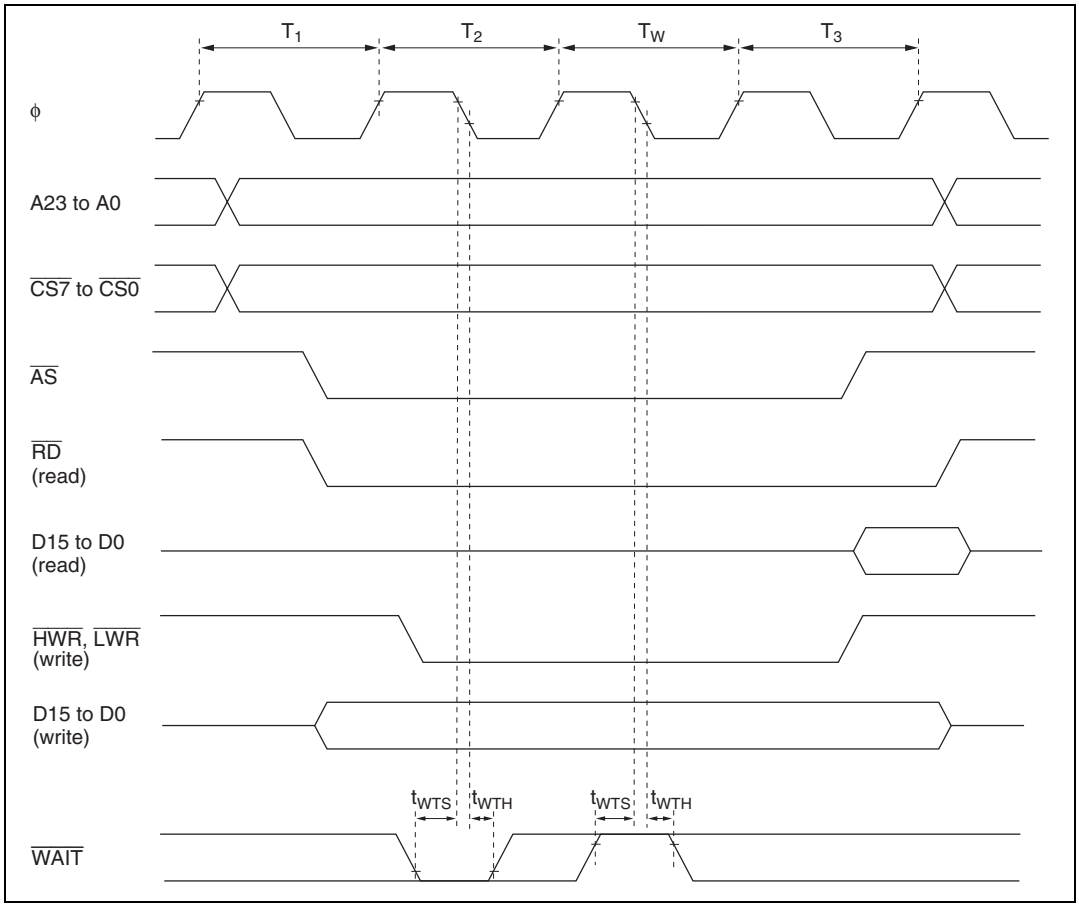


Figure 24.9 Basic Bus Timing (Three-State Access with One Wait State)

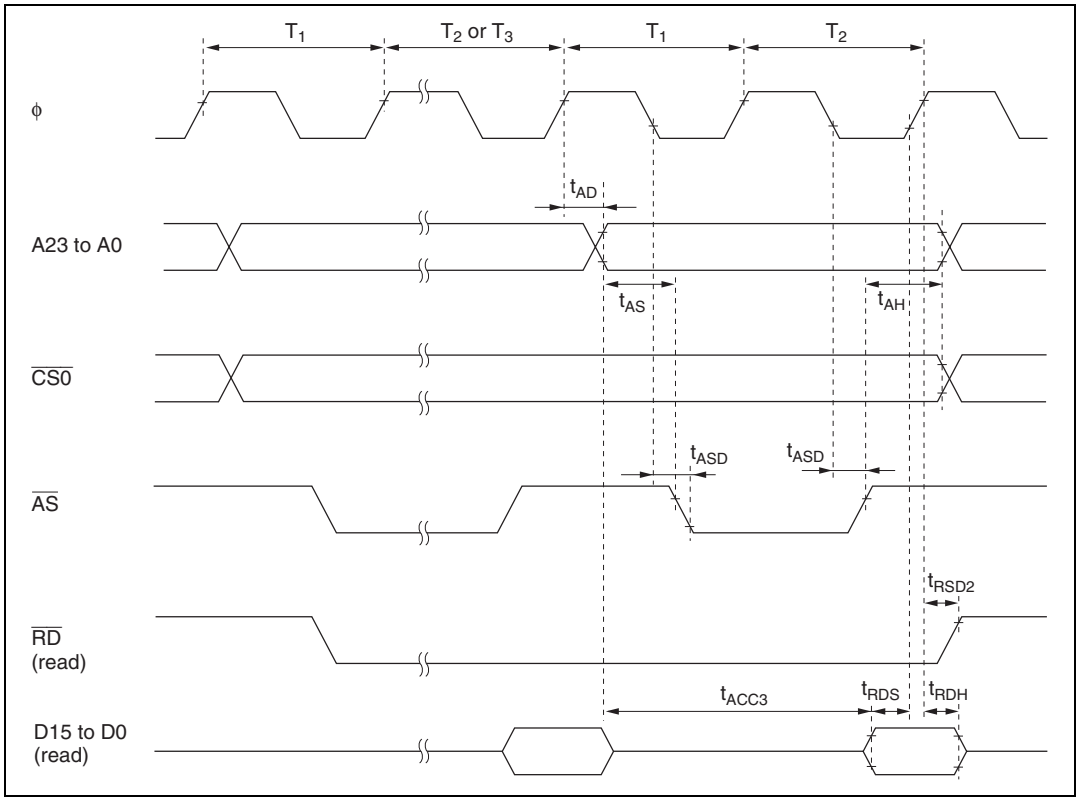


Figure 24.10 Burst ROM Access Timing (Two-State Access)

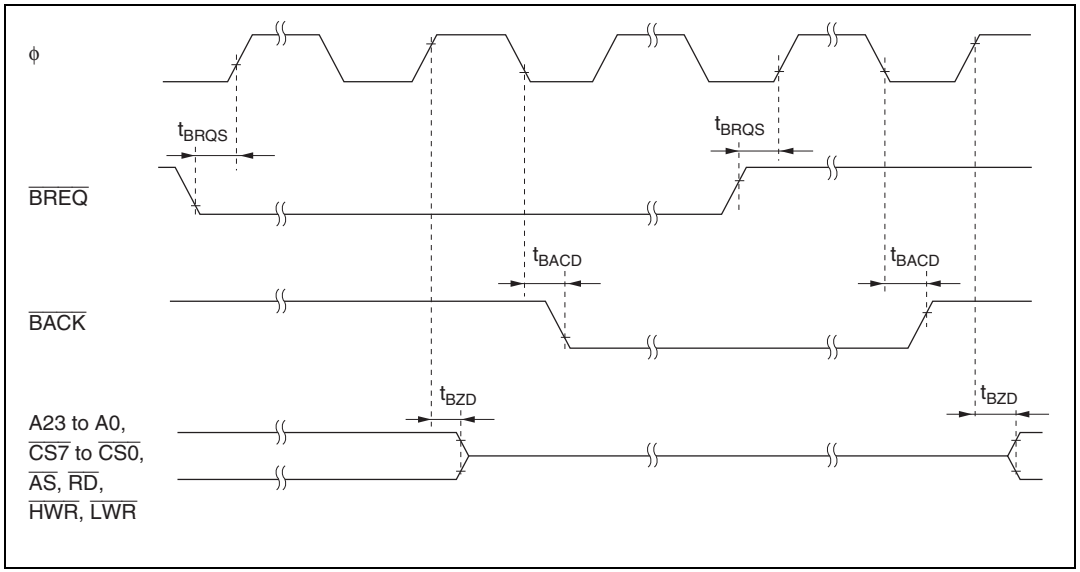


Figure 24.11 External Bus Release Timing

24.4.4 Timing of On-Chip Supporting Modules

Table 24.7 lists the timing of on-chip supporting modules.

Table 24.7 Timing of On-Chip Supporting Modules

Conditions: V_{CC} = $PLLV_{CC}$ = DrV_{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} = $PLLV_{SS}$ = DrV_{SS} = AV_{SS} = 0 V, ϕ = 13 MHz to 16 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 port	Output data delay time		t_{PWD}	—	60	ns	Figure 24.12
	Input data setup time		t_{PRS}	50	—		
	Input data hold time		t_{PRH}	50	—		
TPU	Timer output delay time		t_{TOCD}	—	60	ns	Figure 24.13
	Timer input setup time		t_{TICS}	40	—		
	Timer clock input setup time		t_{TCKS}	40	—	ns	Figure 24.14
	Timer clock pulse width	Single edge	t_{TCKWH}	1.5	—	t_{cyc}	
		Both edges	t_{TCKWL}	2.5	—		
TMR	Timer output delay time		t_{TMOD}	—	60	ns	Figure 24.15
	Timer reset input setup time		t_{TMRS}	50	—	ns	Figure 24.17
	Timer clock input setup time		t_{TMCS}	50	—	ns	Figure 24.16
	Timer clock pulse width	Single edge	t_{TMCWH}	1.5	—	t_{cyc}	
		Both edges	t_{TMCWL}	2.5	—		

Item			Symbol	Min.	Max.	Unit	Test Conditions
SCI	Input clock cycle	Asynchronous	t_{Scyc}	4	—	t_{cyc}	Figure 24.18
		Synchronous		6	—		
	Input clock pulse width		t_{SCKW}	0.4	0.6	t_{Scyc}	
	Input clock rise time		t_{SCKr}	—	1.5	t_{cyc}	
	Input clock fall time		t_{SCKf}	—	1.5		
	Transmit data delay time		t_{TXD}	—	60	ns	Figure 24.19
	Receive data setup time (synchronous)		t_{RXS}	60	—		
	Receive data hold time (synchronous)		t_{RXH}	60	—		
A/D converter	Trigger input setup time		t_{TRGS}	40	—	ns	Figure 24.20
Boundary scan	TCK cycle time		t_{cyc}	62.5	—	ns	Figure 24.21
	TCK high level pulse width		t_{TCKH}	0.4	0.6	t_{cyc}	
	TCK low level pulse width		t_{TCKL}	0.4	0.6	t_{cyc}	
	$\overline{\text{TRST}}$ pulse width		t_{TRSW}	20		t_{cyc}	Figure 24.22
	$\overline{\text{TRST}}$ setup time		t_{TRSS}	250	—	ns	
	TDI setup time		t_{TDIS}	30	—	ns	Figure 24.23
	TDI hold time		t_{TDIH}	10	—		
	TMS setup time		t_{TMSS}	30	—		
	TMS hold time		t_{TMSH}	10	—		
	TDO delay time		t_{TDOD}	—	40		

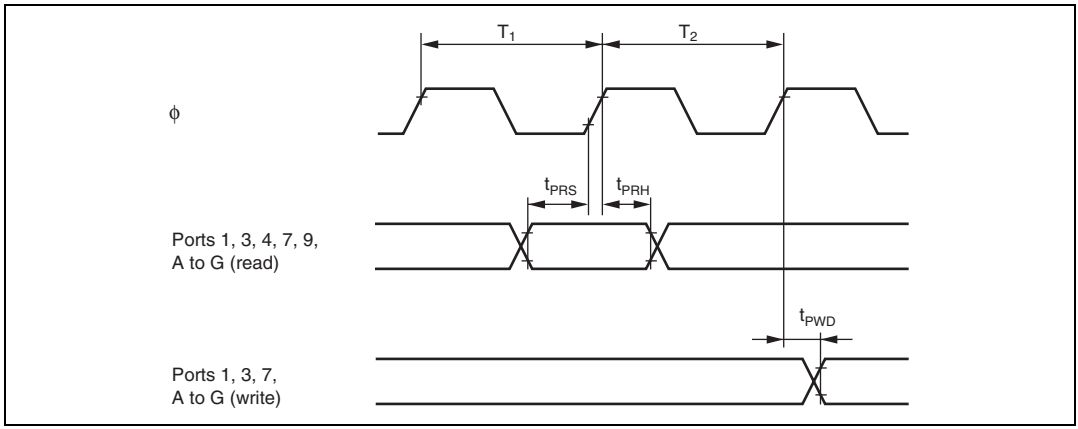


Figure 24.12 I/O Port Input/Output Timing

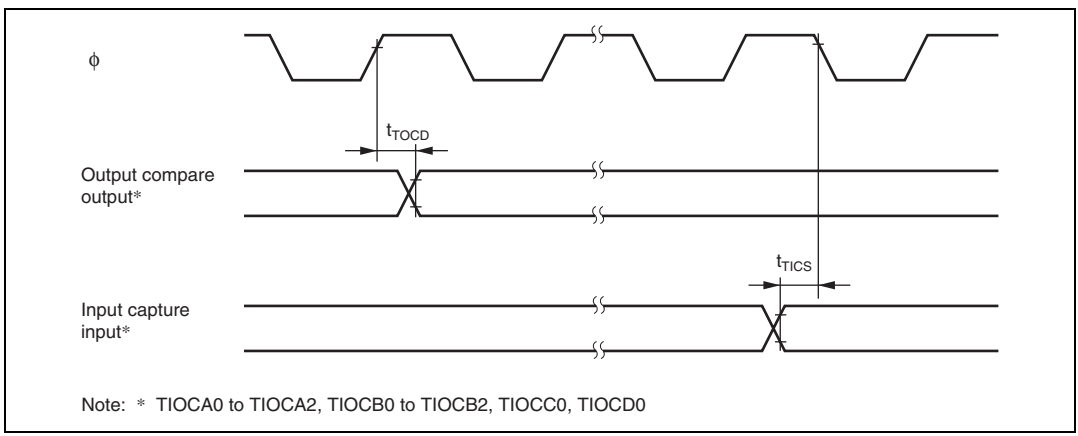


Figure 24.13 TPU Input/Output Timing

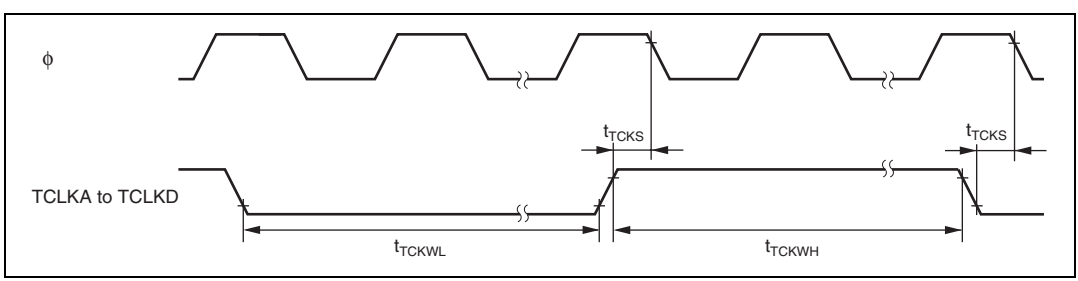


Figure 24.14 TPU Clock Input Timing

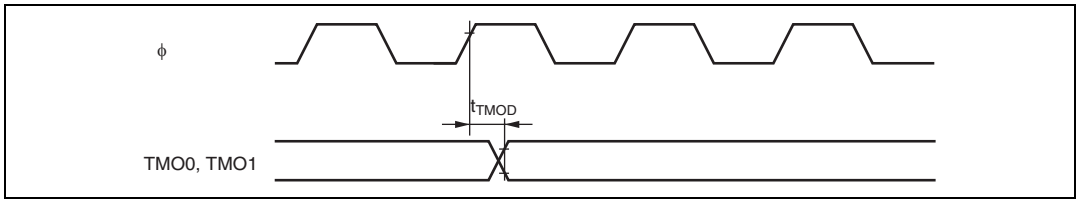


Figure 24.15 8-bit Timer Output Timing

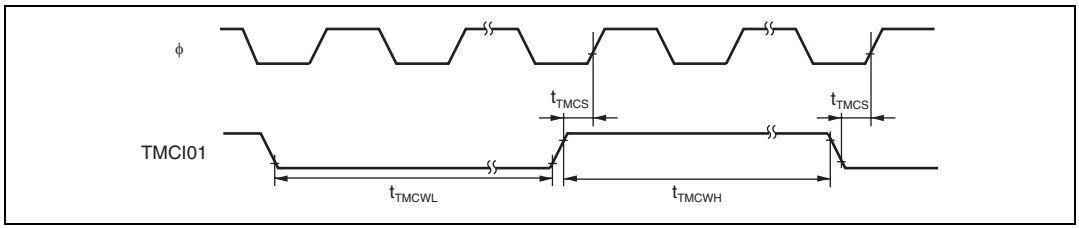


Figure 24.16 8-bit Timer Clock Input Timing

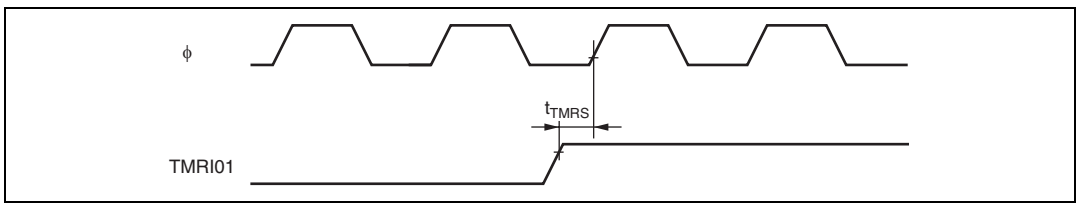


Figure 24.17 8-bit Timer Reset Input Timing

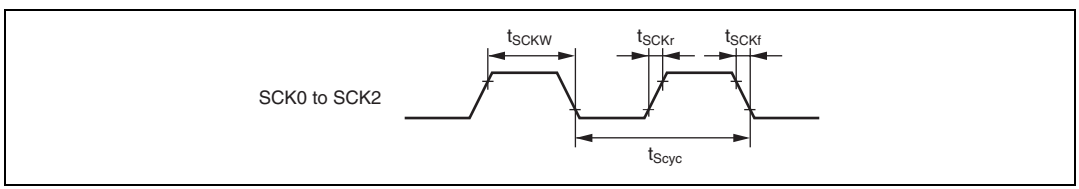


Figure 24.18 SCK Clock Input Timing

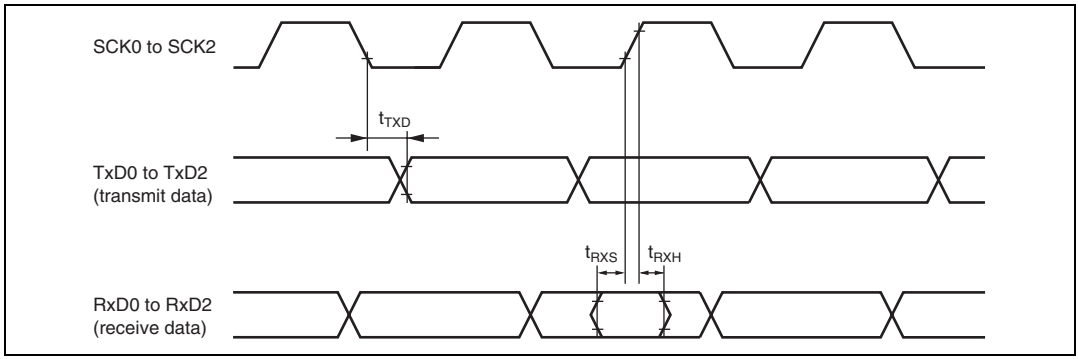


Figure 24.19 SCI Input/Output Timing (Clock Synchronous Mode)

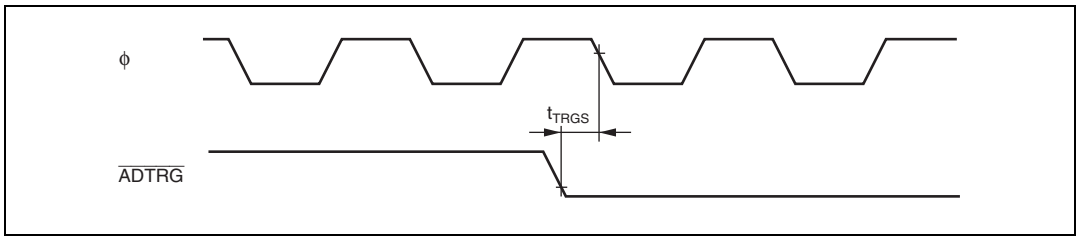


Figure 24.20 A/D Converter External Trigger Input Timing

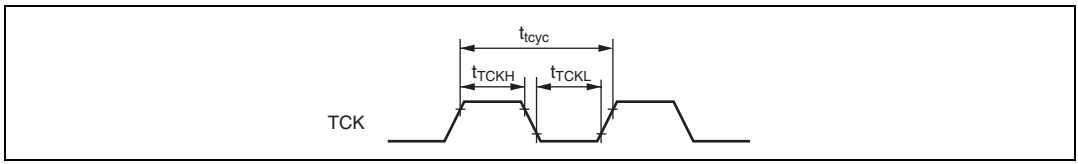


Figure 24.21 Boundary Scan TCK Input Timing

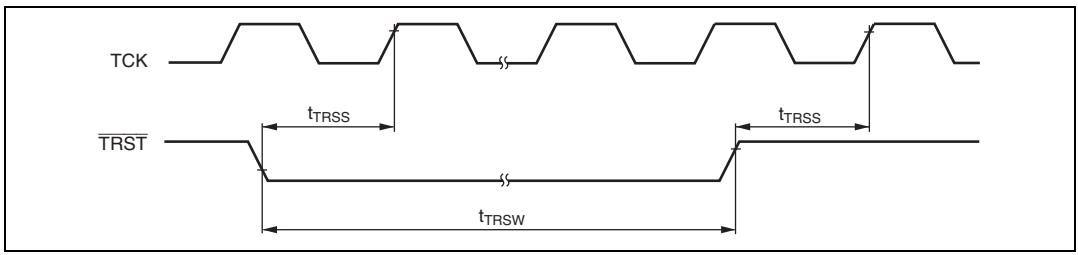


Figure 24.22 Boundary Scan $\overline{TRST}$ Input Timing (At Reset Hold)

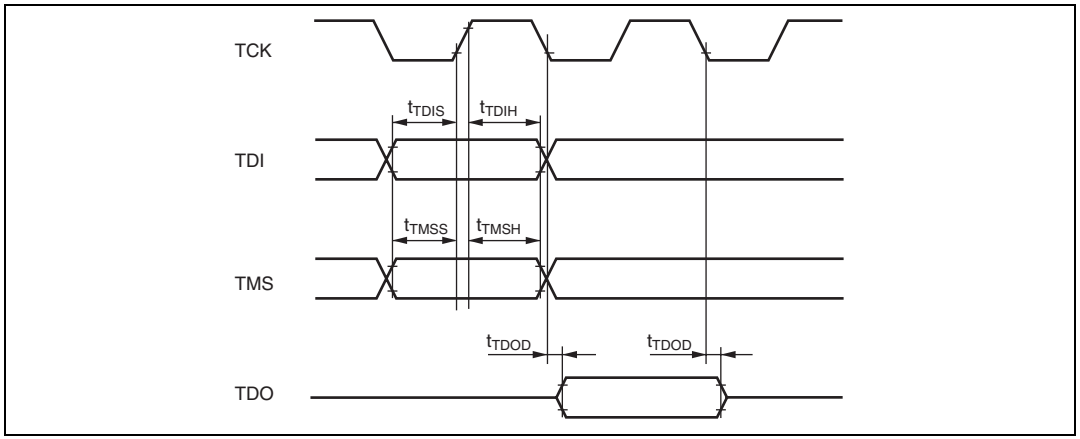


Figure 24.23 Boundary Scan Data Transmission Timing

24.5 USB Characteristics

Table 24.8 lists the USB characteristics (USD+ and USD- pins) when the on-chip USB transceiver is used.

Table 24.8 USB Characteristics (USD+ and USD- pins) when On-Chip USB Transceiver Is Used

Conditions: V_{CC} = $PLLV_{CC}$ = DrV_{CC} = 3.0 V to 3.6 V, V_{SS} = $PLLV_{SS}$ = DrV_{SS} = AV_{SS} = 0 V, ϕ = 13 MHz to 16 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 Condition	
Input characteristics	Input high level voltage	V_{IH}	2.0	—	V		Figures 24.24, 24.25
	Input low level voltage	V_{IL}	—	0.8	V		
	Differential input sense	V_{DI}	0.2	—	V	| (D+)-(D-)| DrVcc = 3.3 to 3.6 V	
	Differential common mode range	V_{CM}	0.8	2.5	V		
Output characteristics	Output high level voltage	V_{OH}	2.8	—	V	I_{OH} = -200 μA	
	Output low level voltage	V_{OL}	—	0.3	V	I_{OL} = 2 mA	
	Crossover voltage	V_{CRS}	1.3	2.0	V		
	Rise time	t_R	4	20	ns		
	Fall time	t_F	4	20	ns		
	Rise time/fall time matching	t_{RFM}	90	111.11	%	(T_R/T_F)	
	Output resistance	Z_{DRV}	28	44	Ω	Including Rs = 24 Ω	

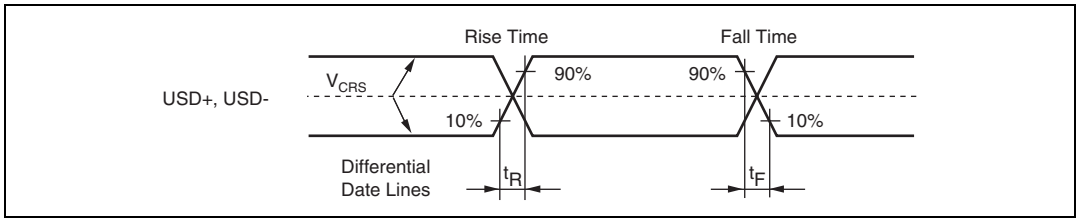


Figure 24.24 Data Signal Timing

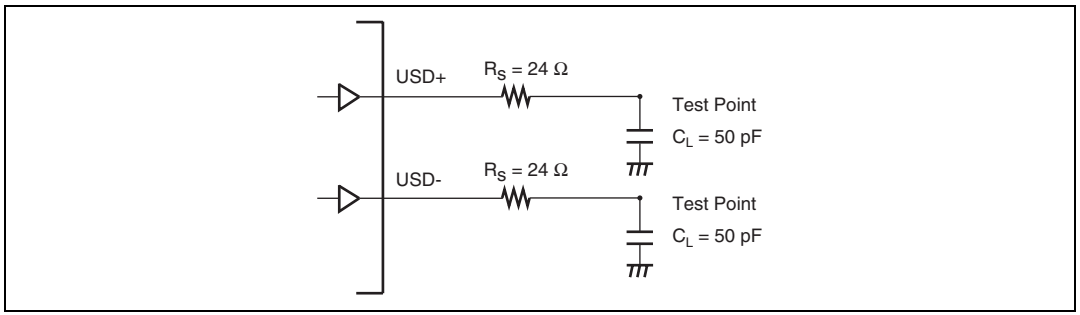


Figure 24.25 Test Load Circuit

24.6 A/D Conversion Characteristics

Table 24.9 lists the A/D conversion characteristics.

Table 24.9 A/D Conversion Characteristics

Conditions: V_{CC} = $PLLV_{CC}$ = DrV_{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} = $PLLV_{SS}$ = DrV_{SS} = AV_{SS} = 0 V, ϕ = 13 MHz to 16 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	8.1	—	—	µs
Analog input capacitance	—	—	20	pF
Permissible signal-source impedance	—	—	5	kΩ
Nonlinearity error	—	—	±6.0	LSB
Offset error	—	—	±4.0	LSB
Full-scale error	—	—	±4.0	LSB
Quantization	—	—	±0.5	LSB
Absolute accuracy	—	—	±8.0	LSB

24.7 D/A Conversion Characteristics

Table 24.10 lists the D/A conversion characteristics.

Table 24.10 D/A Conversion Characteristics

Conditions: V_{CC} = $PLLV_{CC}$ = DrV_{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} = $PLLV_{SS}$ = DrV_{SS} = AV_{SS} = 0 V, ϕ = 13 MHz to 16 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 Condition
Resolution	8	8	8	bits	
Conversion time	—	—	10	µs	Load capacitance = 20 pF
Absolute accuracy	—	±2.0	±3.0	LSB	Load capacitance = 2 MΩ
	—	—	±2.0	LSB	Load capacitance = 4 MΩ

24.8 Flash Memory Characteristics

Table 24.11 Flash Memory Characteristics

Conditions: V_{CC} = $PLLV_{CC}$ = DrV_{CC} = 3.0 V to 3.6 V, AV_{CC} = 2.7 V to 3.6 V, V_{ref} = 2.7 V to AV_{CC}, V_{SS} = $PLLV_{SS}$ = DrV_{SS} = AV_{SS} = 0 V, T_a = –20 to +75°C (Programming/erasing operating temperature range)

Item		Symbol	Min.	Typ.	Max.	Unit
Programming time*1 *2 *4		t_P	—	10	200	ms/128 bytes
Erase time*1 *3 *5		t_E	—	50	1000	ms/block
Reprogramming count		N_{WEC}	100*6	10,000*7	—	Times
Data retention time*8		t_{DRP}	10	—	—	Years
Programming	Wait time after PSU1 bit setting*1	y	50	50	—	µs
	Wait time after P1 bit setting*1*4	z0	28	30	32	µs
		z1	198	200	202	µs
		z2	8	10	12	µs
	Wait time after P1 bit clear*1	α	5	5	—	µs
	Wait time after PSU1 bit clear*1	β	5	5	—	µs
	Wait time after PV1 bit setting*1	γ	4	4	—	µs
	Wait time after H'FF dummy write*1	ε	2	2	—	µs
	Wait time after PV1 bit clear*1	η	2	2	—	µs
	Maximum programming count*1*4	N1	—	—	6*4	Times
		N2	—	—	994*4	Times
Common	Wait time after SWE1 bit setting*1	x	1	1	—	µs
	Wait time after SWE1 bit clear*1	θ	100	100	—	µs
Erase	Wait time after ESU1 bit setting*1	y	100	100	—	µs
	Wait time after E1 bit setting*1*5	z	10	10	100	ms
	Wait time after E1 bit clear*1	α	10	10	—	µs
	Wait time after ESU1 bit clear*1	β	10	10	—	µs
	Wait time after EV1 bit setting*1	γ	20	20	—	µs
	Wait time after H'FF dummy write*1	ε	2	2	—	µs
	Wait time after EV1 bit clear*1	η	4	4	—	µs
	Maximum erase count*1*5	N	—	—	100	Times

Notes: 1. Make each time setting in accordance with the program/program-verify flowchart or erase/erase-verify flowchart.

2. Programming time per 128 bytes (Shows the total period for which the P-bit in the flash memory control register (FLMCR1) is set. It does not include the programming verification time).

3. Block erase time (Shows the total period for which the E1-bit FLMCR1 is set. It does not include the erase verification time).

4. Maximum programming time value

   t_p (max.) = Wait time after P1 bit set (z) × maximum programming count (N1 + N2)

   = (Z0 + Z2) × 6 + Z1 × 994

5. Maximum erasure time value

   t_E (max.) = Wait time after E1 bit set (z) × maximum erasure count (N)

6. Minimum number of times for which all characteristics are guaranteed after rewriting (Guarantee range is 1 to minimum value).
7. Reference value for 25°C (as a guideline, rewriting should normally function up to this value).
8. Data retention characteristic when rewriting is performed within the specification range, including the minimum value.

24.9 Usage Note

General Notice during Design for Printed Circuit Board: Measures for radiation noise caused by the transient current in this LSI should be taken into consideration. The examples of the measures are shown below.

- To use a multilayer printed circuit board which includes layers for Vcc and GND.
- To mount by-pass capacitors (approximately 0.1 μF) between the Vcc and GND (Vss) pins, and the $PLLV_{CC}$ and PLLGND pins, of this LSI.

Section 25 Electrical Characteristics (H8S/2215R)

25.1 Absolute Maximum Ratings

Table 25.1 lists the absolute maximum ratings.

Table 25.1 Absolute Maximum Ratings

Item	Symbol	Value	Unit
Power supply voltage	V_{CC}, $PLLV_{CC}$, DrV_{CC}	–0.3 to +4.3	V
Input voltage (except ports 4 and 9)	V_{in}	–0.3 to V_{CC} +0.3	V
Input voltage (ports 4 and 9)	V_{in}	–0.3 to AV_{CC} +0.3	V
Reference 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	°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.

Note: * The operating temperature ranges for flash memory programming/erasing are T_a = –20°C to +75°C.

25.2 Power Supply Voltage and Operating Frequency Range

Power supply voltage and operating frequency ranges (shaded areas) are shown in figure 25.1.

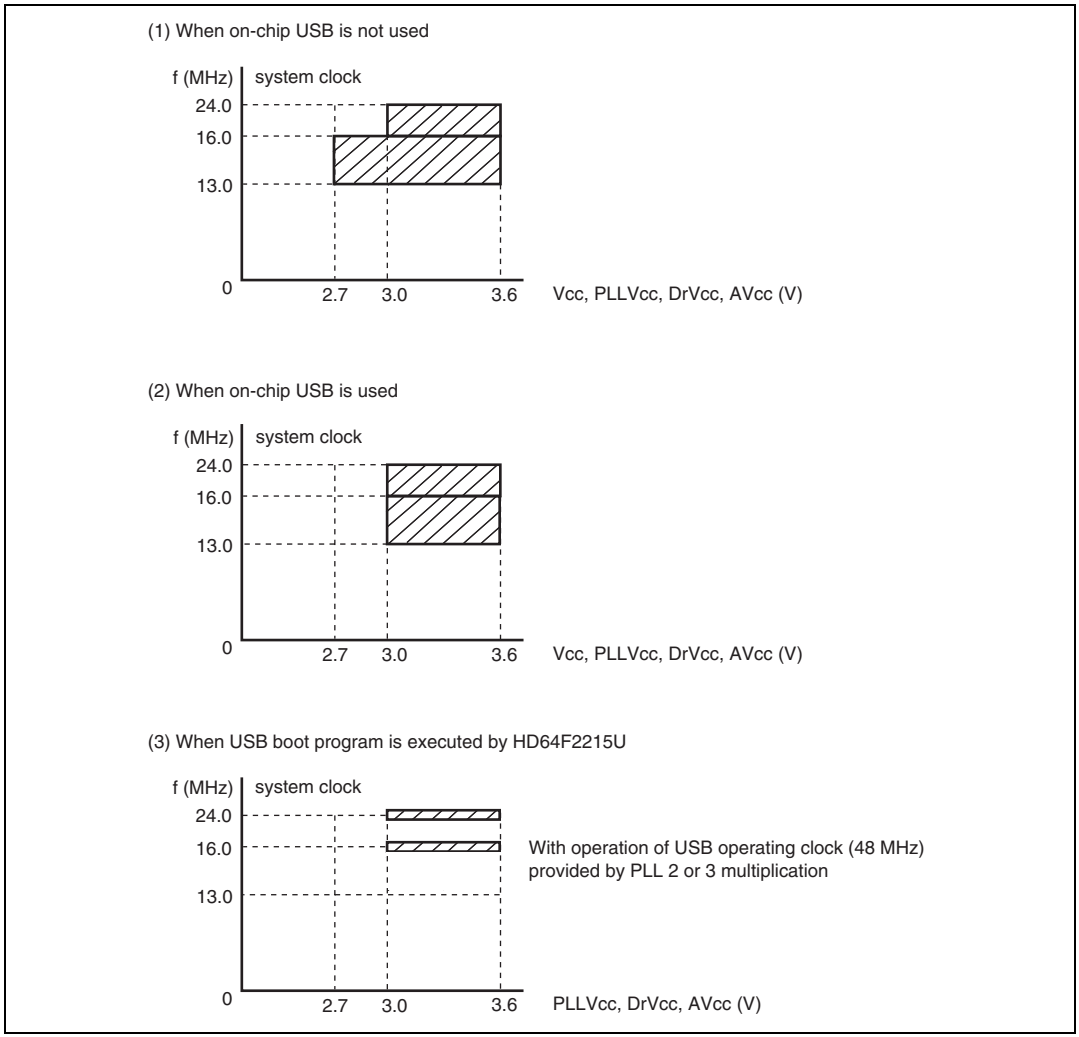


Figure 25.1 Power Supply Voltage and Operating Ranges

25.3 DC Characteristics

Table 25.2 lists the DC characteristics. Table 25.3 lists the permissible output currents.

Table 25.2 DC Characteristics

Conditions: V_{CC} = $PLLV_{CC}$ = DrV_{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} = $PLLV_{SS}$ = DrV_{SS} = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)*1

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Schmitt trigger input voltage	$\overline{IRQ0}$ to $\overline{IRQ5}$, $\overline{IRQ7}$	V_T^-	$V_{CC} \times 0.2$	—	—	V	
		V_T^+	—	—	$V_{CC} \times 0.8$	V	
		$V_T^+ - V_T^-$	$V_{CC} \times 0.05$	—	—	V	
Input high voltage	$\overline{RES}$, $\overline{STBY}$, NMI, MD2 to MD0, $\overline{TRST}$, TCK, TMS, TDI, EMLEN, VBUS, $\overline{UBPM}$, FWE	V_{IH}	$V_{CC} \times 0.9$	—	V_{CC} + 0.3	V	
	EXTAL, EXTAL48, Ports 1, 3, 7, and A to G		$V_{CC} \times 0.8$	—	V_{CC} + 0.3	V	
	Ports 4 and 9		$V_{CC} \times 0.8$	—	AV_{CC} + 0.3	V	
Input low voltage	$\overline{RES}$, $\overline{STBY}$, MD2 to MD0, $\overline{TRST}$, TCK, TMS, TDI, EMLEN, VBUS, $\overline{UBPM}$, FWE	V_{IL}	–0.3	—	$V_{CC} \times 0.1$	V	
	EXTAL, EXTAL48, NMI, Ports 1, 3, 4, 7, 9, and A to G		–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

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Output low voltage	All output pins	V_{OL}	—	—	0.4	V	I_{OH} = 0.4 mA
			—	—	0.4	V	I_{OL} = 0.8 mA
Input leakage current	$\overline{RES}$, $\overline{STBY}$, NMI, MD2 to MD0, FWE*5, VBUS,$\overline{UBPM}$	$\lvert I_{in} \rvert$	—	—	1.0	µA	V_{in} = 0.5 V to V_{CC} – 0.5 V
	Ports 4, 9	$\lvert I_{in} \rvert$	—	—	1.0	µA	V_{in} = 0.5 V to AV_{CC} – 0.5 V
3-state leak current (off status)	Ports 1, 3, 7, A to G	$\lvert I_{TSI} \rvert$	—	—	1.0	µA	V_{in} = 0.5 V to V_{CC} – 0.5 V
Input pull-up MOS current	Ports A to E	$-I_P$	10	—	300	µA	V_{in} = 0 V
Input capacity	$\overline{RES}$, NMI	C_{in}	—	—	30	pF	V_{in} = 0 V
	All input pins other than $\overline{RES}$ and NMI		—	—	15	pF	f = 1 MHz T_a = 25°C
Current dissipation*2	Normal operation (USB halts)	I_{CC}*3	—	23 (V_{CC} = 3.3 V)	40 (V_{CC} = 3.6 V)	mA	f = 16 MHz
			—	34 (V_{CC} = 3.3 V)	55 (V_{CC} = 3.6 V)	mA	f = 24 MHz
	Normal operation (USB operates)		—	28 (V_{CC} = 3.3 V)	50 (V_{CC} = 3.6 V)	mA	f = 16 MHz (PLL 3 multiplication)
			—	40 (V_{CC} = 3.3 V)	60 (V_{CC} = 3.6 V)	mA	f = 24 MHz (PLL 2 multiplication)
	Sleep mode		—	18 (V_{CC} = 3.3 V)	35 (V_{CC} = 3.6 V)	mA	f = 16 MHz (when USB and PLL are halted)
			—	26 (V_{CC} = 3.3 V)	45 (V_{CC} = 3.6 V)	mA	f = 24 MHz (when USB and PLL are halted)

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Current dissipation*2	All modules stopped	I_{CC}*3	—	15 (V_{CC} = 3.3 V)	—	mA	f = 16 MHz (reference value)
			—	21 (V_{CC} = 3.3 V)	—	mA	f = 24 MHz (reference value)
	Standby mode*4		—	1.0	10	µA	$T_a \leq 50°C$
			—	—	50	µA	$50°C < T_a$
Analog power supply current	During A/D conversion	AI_{CC}	—	0.3	1.5	mA	AV_{CC} = 3.3 V
	Idle		—	0.01	5.0	µA	
Reference power supply current	During A/D conversion	AI_{CC}	—	1.2	2.5	mA	V_{ref} = 3.3 V
	Idle		—	0.01	5.0	µA	
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Notes: 1. If the A/D or D/A converter is not used, the AVCC, V_{ref}, and AVSS pins should not be open. Even if the A/D or D/A converter is not used, connect the AVCC and V_{ref} pins to V_{CC} and the AVSS pin to V_{SS}, respectively. In this case, the V_{ref} level should be the AVCC level or less.

2. Current dissipation values are for V_{IH} (min.) = V_{CC} – 0.2 V and V_{IL} (max.) = 0.2 V, with all output pins unloaded and the on-chip MOS pull-up transistors in the off state.

3. I_{CC} depends on V_{CC} and f as follows:
   I_{CC} (max.) = 1.0 (mA) + 0.67 (mA/(MHz x V)) × V_{CC} × f (normal operation, USB halted)
   I_{CC} (max.) = 1.0 (mA) + 0.85 (mA/(MHz x V)) × V_{CC} × f (normal operation, USB operated)
   I_{CC} (max.) = 1.0 (mA) + 0.59 (mA/(MHz x V)) × V_{CC} × f (sleep mode)

4. The values are for $V_{RAM} \leq V_{CC} < 2.7$ V, V_{IH} (min.) = V_{CC} × 0.9, and V_{IL} (max.) = 0.3 V.

Table 25.3 Permissible Output Currents

Conditions: V_{CC} = $PLLV_{CC}$ = DrV_{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} = $PLLV_{SS}$ = DrV_{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	V_{CC} = 2.7 to 3.6 V	I_{OL}	—	—	1.0	mA
Permissible output low current (total)	Total of all output pins	V_{CC} = 2.7 to 3.6 V	$\sum I_{OL}$	—	—	60	mA
Permissible output high current (per pin)	All output pins	V_{CC} = 2.7 to 3.6 V	$-I_{OH}$	—	—	1.0	mA
Permissible output high current (total)	Total of all output pins	V_{CC} = 2.7 to 3.6 V	$\sum -I_{OH}$	—	—	30	mA

Note: * To protect chip reliability, do not exceed the output current values in table 25.3.

25.4 AC Characteristics

Figure 25.2 shows, the test conditions for the AC characteristics.

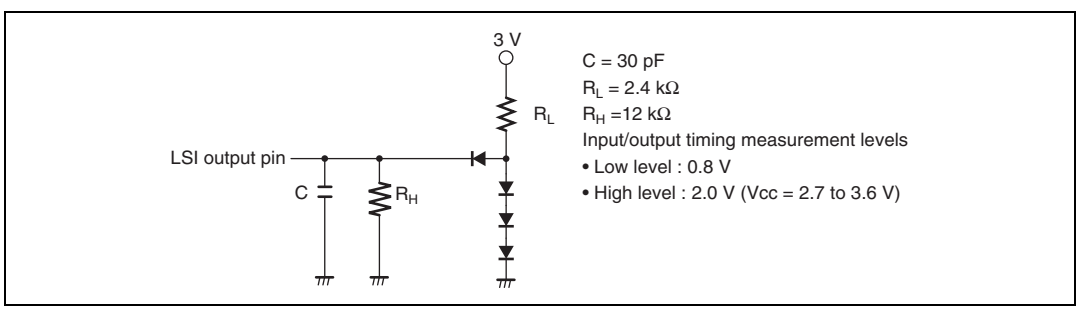


Figure 25.2 Output Load Circuit

25.4.1 Clock Timing

Table 25.4 lists the clock timing

Table 25.4 Clock Timing

Condition A: V_{CC} = $PLLV_{CC}$ = DrV_{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} = $PLLV_{SS}$ = DrV_{SS} = AV_{SS} = 0 V, ϕ = 13 MHz to 16 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Condition B: V_{CC} = $PLLV_{CC}$ = DrV_{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} = $PLLV_{SS}$ = DrV_{SS} = AV_{SS} = 0 V, ϕ = 13 MHz to 24 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}	62.5	76.9	41.6	76.9	ns	Figure 25.3
Clock high pulse width	t_{CH}	20	—	13	—	ns	
Clock low pulse width	t_{CL}	20	—	13	—	ns	
Clock rise time	t_{Cr}	—	10	—	7	ns	
Clock fall time	t_{Cf}	—	10	—	7	ns	
Oscillation stabilization time at reset (crystal)	t_{OSC1}	20	—	20	—	ms	Figure 25.4
Oscillation stabilization time in software standby (crystal)	t_{OSC2}	8	—	8	—	ms	C_{L1} = C_{L2} = 10 pF to 22 pF in figure 21.2 Figure 22.3
		4	—	4	—	ms	C_{L1} = C_{L2} = 10 pF to 15 pF in figure 21.2 V_{CC} = 3.0 V to 3.6 V in figure 22.3

Item	Symbol	Condition A Min.	Condition A Max.	Condition B Min.	Condition B Max.	Unit	Test Conditions
External clock output stabilization delay time	t_{DEXT}	500	—	500	—	µs	Figure 25.4
USB operating clock (48 MHz) stabilization time	t_{OSC3}	8	—	8	—	ms	V_{CC} = 3.0 V to 3.6 V
USB operating clock (48 MHz) oscillator frequency	f_{48}	48		4.8		MHz	
USB operating clock (48 MHz) cycle time	f_{48}	20.8		20.8		ns	

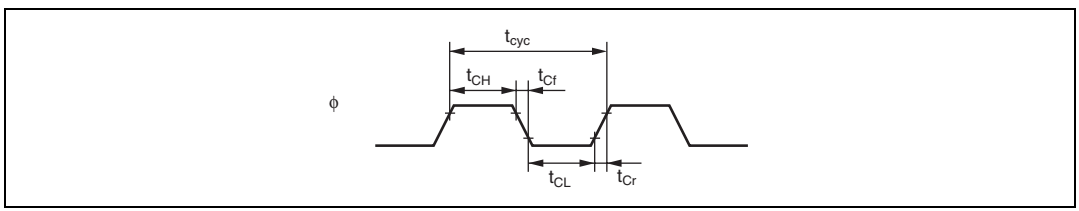


Figure 25.3 System Clock Timing

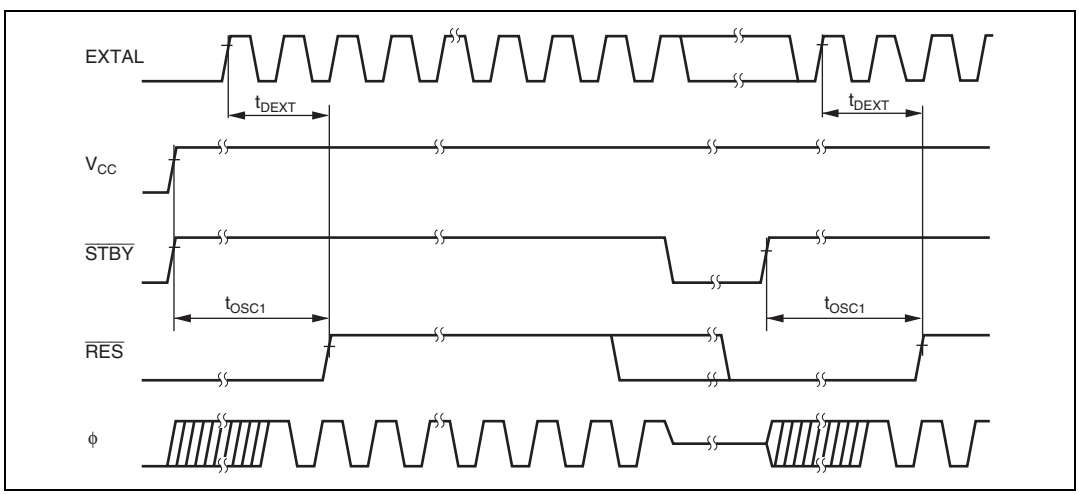


Figure 25.4 Oscillation Stabilization Timing

25.4.2 Control Signal Timing

Table 25.5 lists the control signal timing.

Table 25.5 Control Signal Timing

Condition A: V_{CC} = $PLLV_{CC}$ = DrV_{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} = $PLLV_{SS}$ = DrV_{SS} = AV_{SS} = 0 V, ϕ = 13 MHz to 16 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Condition B: V_{CC} = $PLLV_{CC}$ = DrV_{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} = $PLLV_{SS}$ = DrV_{SS} = AV_{SS} = 0 V, ϕ = 13 MHz to 24 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

		Condition A and B			
Item	**Symbol**	**Min.**	**Max.**	**Unit**	**Test Conditions**
$\overline{RES}$ setup time	t_{RESS}	250	—	ns	Figure 25.5
$\overline{RES}$ pulse width	t_{RESW}	20	—	t_{cyc}	
$\overline{MRES}$ setup time	t_{MRESS}	250	—	ns	
$\overline{MRES}$ pulse width	t_{MRESW}	20	—	t_{cyc}	
NMI setup time	t_{NMIS}	250	—	ns	Figure 25.6
NMI hold time	t_{NMIH}	10	—	ns	
NMI pulse width (exiting software standby mode)	t_{NMIW}	200	—	ns	
$\overline{IRQ}$ setup time	t_{IRQS}	250	—	ns	
$\overline{IRQ}$ hold time	t_{IRQH}	10	—	ns	
$\overline{IRQ}$ pulse width (exiting software standby mode)	t_{IRQW}	200	—	ns	

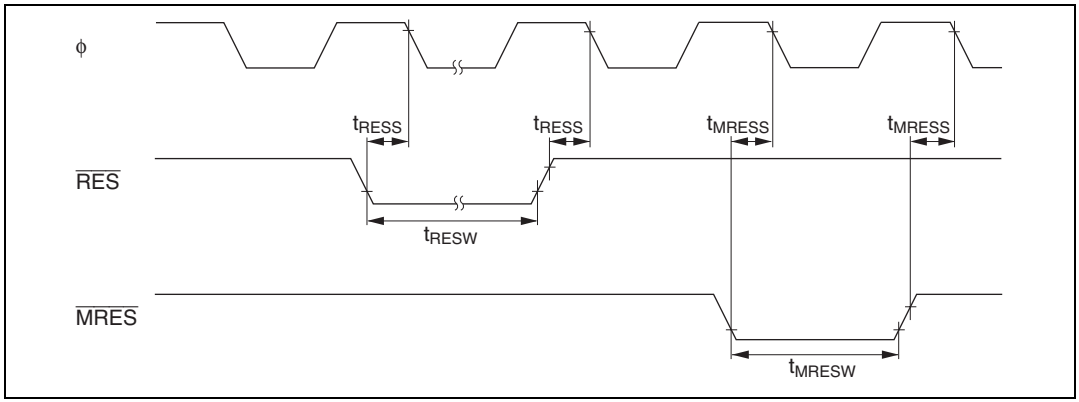


Figure 25.5 Reset Input Timing

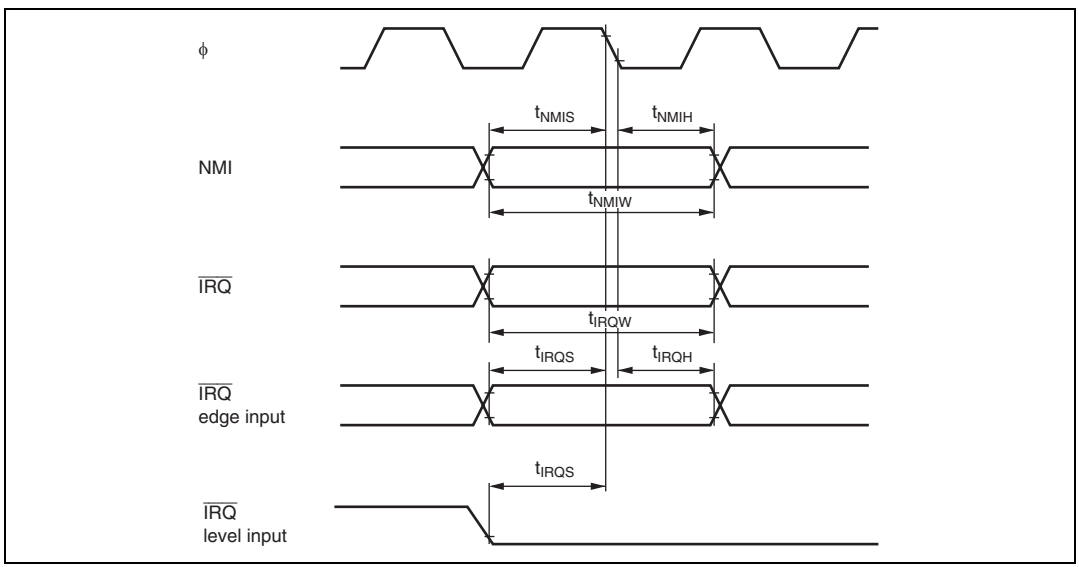


Figure 25.6 Interrupt Input Timing

25.4.3 Bus Timing

Table 25.6 shows, Bus Timing.

Table 25.6 Bus Timing

Condition A: V_{CC} = $PLLV_{CC}$ = DrV_{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} = $PLLV_{SS}$ = DrV_{SS} = AV_{SS} = 0 V, ϕ = 13 MHz to 16 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Condition B: V_{CC} = $PLLV_{CC}$ = DrV_{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} = $PLLV_{SS}$ = DrV_{SS} = AV_{SS} = 0 V, ϕ = 13 MHz to 24 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}	—	50	—	30	ns	Figures 25.7, 25.8, 25.10
Address setup time	t_{AS}	0.5 × t_{cyc} – 30	—	0.5 × t_{cyc} – 20	—	ns	
Address hold time	t_{AH}	0.5 × t_{cyc} – 15	—	0.5 × t_{cyc} – 8	—	ns	
$\overline{CS}$ delay time	t_{CSD}	—	50	—	30	ns	Figures 25.7, 25.8
$\overline{AS}$ delay time	t_{ASD}	—	50	—	25	ns	Figures 25.7, 25.8, 25.10
$\overline{RD}$ delay time 1	t_{RSD1}	—	50	—	25	ns	Figures 25.7, 25.8
$\overline{RD}$ delay time 2	t_{RSD2}	—	50	—	25	ns	Figures 25.7, 25.8, 25.10
Read data setup time	t_{RDS}	30	—	20	—	ns	
Read data hold time	t_{RDH}	0	—	0	—	ns	
Read data access time 2	t_{ACC2}	—	1.5 × t_{cyc} – 65	—	1.5 × t_{cyc} – 35	ns	Figure 25.7
Read data access time 3	t_{ACC3}	—	2.0 × t_{cyc} – 65	—	2.0 × t_{cyc} – 40	ns	Figures 25.7, 25.10
Read data access time 4	t_{ACC4}	—	2.5 × t_{cyc} – 65	—	2.5 × t_{cyc} – 35	ns	Figure 25.8
Read data access time 5	t_{ACC5}	—	3.0 × t_{cyc} – 65	—	3.0 × t_{cyc} – 40	ns	
$\overline{WR}$ delay time 1	t_{WRD1}	—	50	—	20	ns	
$\overline{WR}$ delay time 2	t_{WRD2}	—	50	—	25	ns	Figures 25.7, 25.8

		Condition A		Condition B			
Item	**Symbol**	**Min.**	**Max.**	**Min.**	**Max.**	**Unit**	**Test Conditions**
$\overline{WR}$ pulse width 1	t_{WSW1}	$1.0 \times t_{cyc} - 30$	—	$1.0 \times t_{cyc} - 20$	—	ns	Figure 25.7
$\overline{WR}$ pulse width 2	t_{WSW2}	$1.5 \times t_{cyc} - 30$	—	$1.5 \times t_{cyc} - 20$	—	ns	Figure 25.8
Write data delay time	t_{WDD}	—	50	—	30	ns	Figures 25.7, 25.8
Write data setup time	t_{WDS}	$0.5 \times t_{cyc} - 30$	—	$0.5 \times t_{cyc} - 20$	—	ns	Figure 25.8
Write data hold time	t_{WDH}	$0.5 \times t_{cyc} - 15$	—	$0.5 \times t_{cyc} - 10$	—	ns	Figures 25.7, 25.8
$\overline{WAIT}$ setup time	t_{WTS}	50	—	25	—	ns	Figure 25.9
$\overline{WAIT}$ hold time	t_{WTH}	10	—	5	—	ns	
$\overline{BREQ}$ setup time	t_{BRQS}	50	—	25	—	ns	Figure 25.11
$\overline{BACK}$ delay time	t_{BACD}	—	50	—	40	ns	
Bus-floating time	t_{BZD}	—	80	—	50	ns	

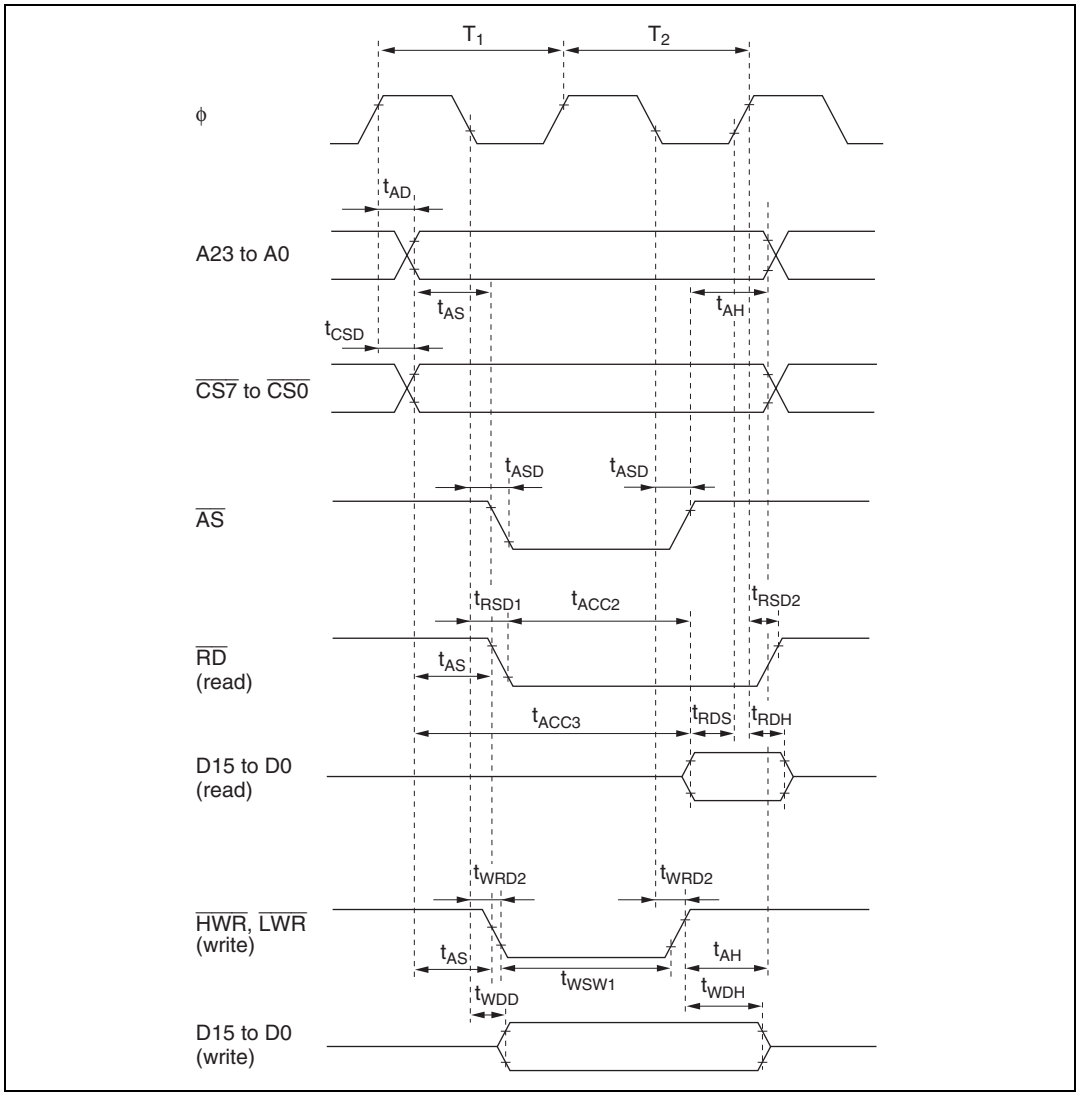


Figure 25.7 Basic Bus Timing (Two-State Access)

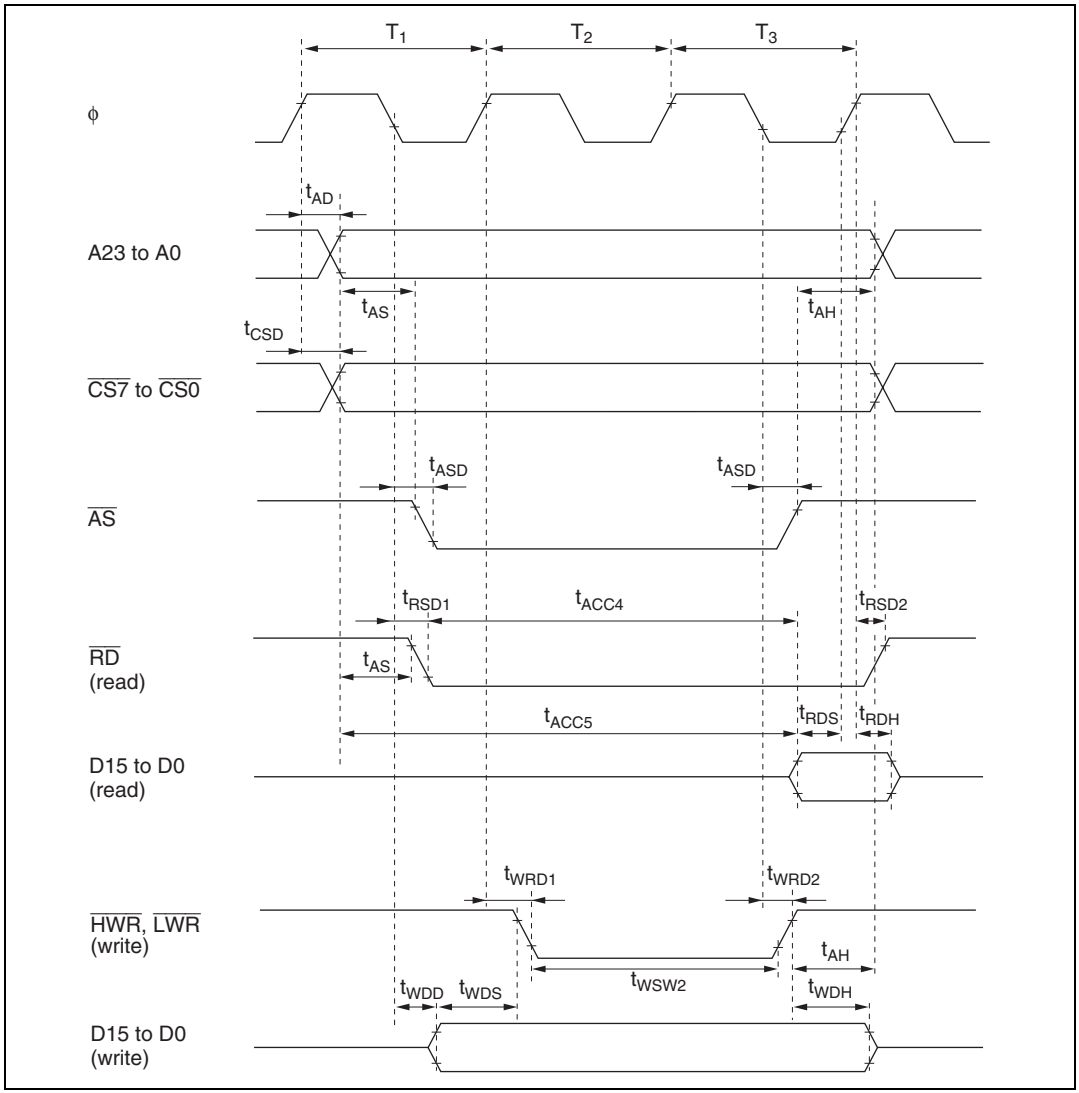


Figure 25.8 Basic Bus Timing (Three-State Access)

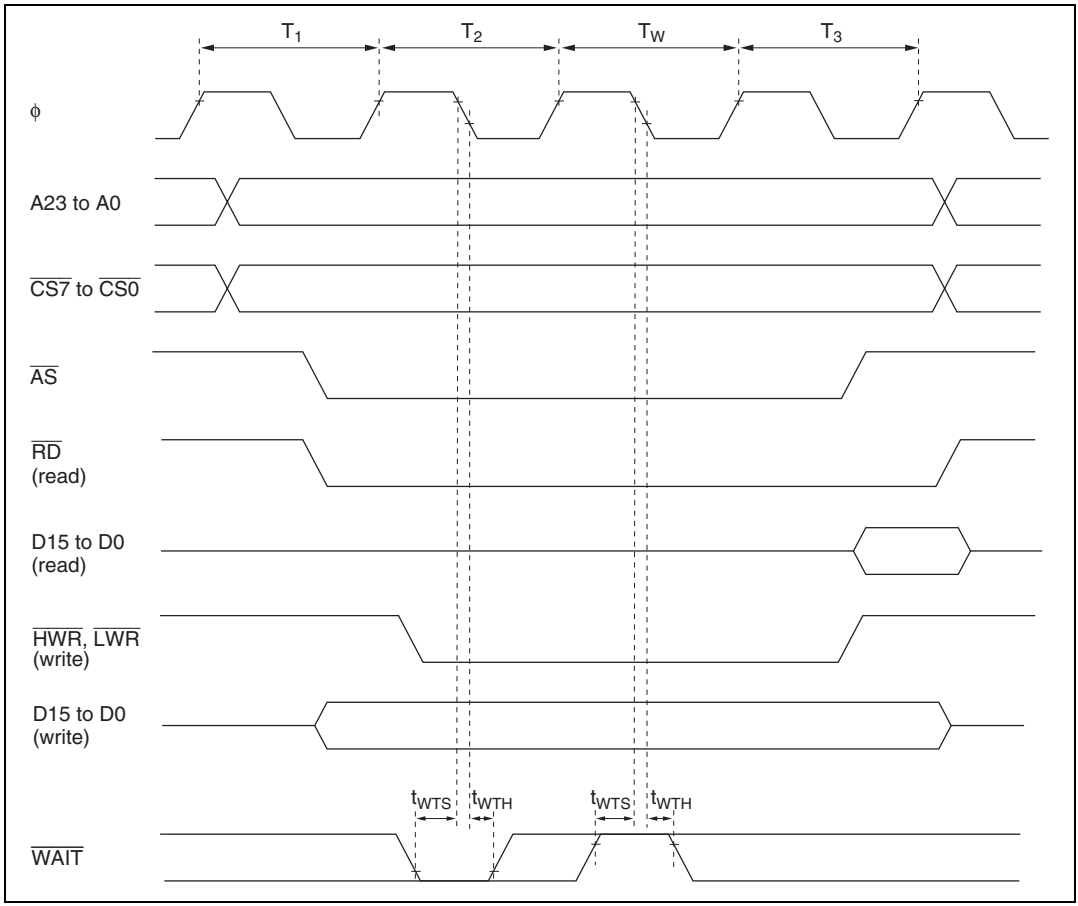


Figure 25.9 Basic Bus Timing (Three-State Access with One Wait State)

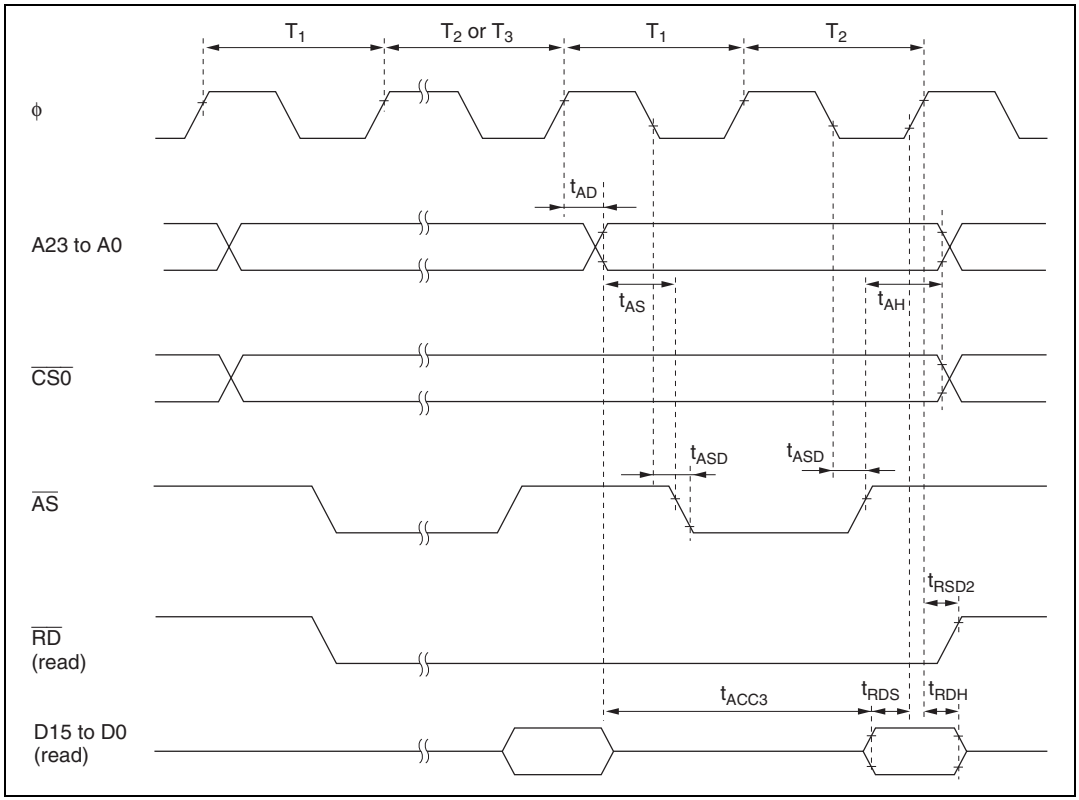


Figure 25.10 Burst ROM Access Timing (Two-State Access)

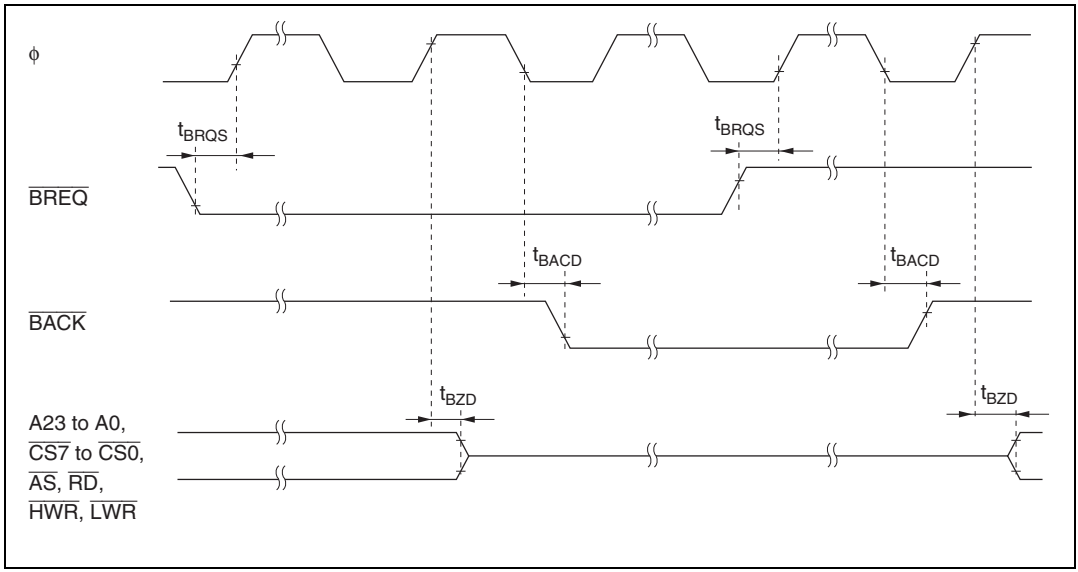


Figure 25.11 External Bus Release Timing

25.4.4 Timing of On-Chip Supporting Modules

Table 25.7 lists the timing of on-chip supporting modules.

Table 25.7 Timing of On-Chip Supporting Modules

Condition A: V_{CC} = $PLLV_{CC}$ = DrV_{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} = $PLLV_{SS}$ = DrV_{SS} = AV_{SS} = 0 V, ϕ = 13 MHz to 16 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Condition B: V_{CC} = $PLLV_{CC}$ = DrV_{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} = $PLLV_{SS}$ = DrV_{SS} = AV_{SS} = 0 V, ϕ = 13 MHz to 24 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

			Condition A		Condition B			Test
Item		**Symbol**	**Min.**	**Max.**	**Min.**	**Max.**	**Unit**	**Conditions**
I/O port	Output data delay time	t_{PWD}	—	60	—	40	ns	Figure 25.12
	Input data setup time	t_{PRS}	50	—	30	—		
	Input data hold time	t_{PRH}	50	—	30	—		
TPU	Timer output delay time	t_{TOCD}	—	60	—	40	ns	Figure 25.13
	Timer input setup time	t_{TICS}	40	—	30	—		
	Timer clock input setup time	t_{TCKS}	40	—	30	—	ns	Figure 25.14
	Timer clock pulse width Single edge	t_{TCKWH}	1.5	—	1.5	—	t_{cyc}	
	Timer clock pulse width Both edges	t_{TCKWL}	2.5	—	2.5	—		
TMR	Timer output delay time	t_{TMOD}	—	60	—	41	ns	Figure 25.15
	Timer reset input setup time	t_{TMRS}	50	—	29	—	ns	Figure 25.17
	Timer clock input setup time	t_{TMCS}	50	—	29	—	ns	Figure 25.16

Item			Symbol	Condition A Min.	Condition A Max.	Condition B Min.	Condition B Max.	Unit	Test Conditions
TMR	Timer clock pulse width	Single edge	t_{TMCWH}	1.5	—	1.5	—	t_{cyc}	Figure 25.16
		Both edges	t_{TMCWL}	2.5	—	2.5	—		
SCI	Input clock cycle	Asynch-ronous	t_{Scyc}	4	—	4	—	t_{cyc}	Figure 25.18
		Synchro-nous		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}	—	60	—	40	ns	Figure 25.19
	Receive data setup time (synchronous)		t_{RXS}	60	—	40	—		
	Receive data hold time (synchronous)		t_{RXH}	60	—	40	—		
A/D converter	Trigger input setup time		t_{TRGS}	40	—	30	—	ns	Figure 25.20
Boundary scan	TCK cycle time		t_{cyc}	62.5	—	41.6	—	ns	Figure 25.21
	TCK high level pulse width		t_{TCKH}	0.4	0.6	0.4	0.6	t_{cyc}	
	TCK low level pulse width		t_{TCKL}	0.4	0.6	0.4	0.6	t_{cyc}	
	$\overline{\text{TRST}}$ pulse width		t_{TRSW}	20		20	—	t_{cyc}	Figure 25.22
	$\overline{\text{TRST}}$ setup time		t_{TRSS}	250	—	250	—	ns	
	TDI setup time		t_{TDIS}	30	—	20	—	ns	Figure 25.23
	TDI hold time		t_{TDIH}	10	—	10	—		
	TMS setup time		t_{TMSS}	30	—	20	—		
	TMS hold time		t_{TMSH}	10	—	10	—		
	TDO delay time		t_{TDOD}	—	40	—	35		

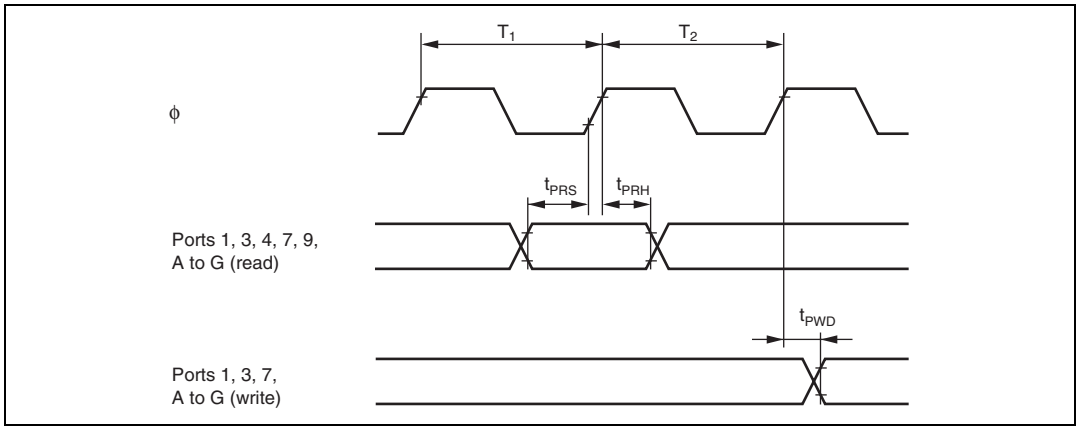


Figure 25.12 I/O Port Input/Output Timing

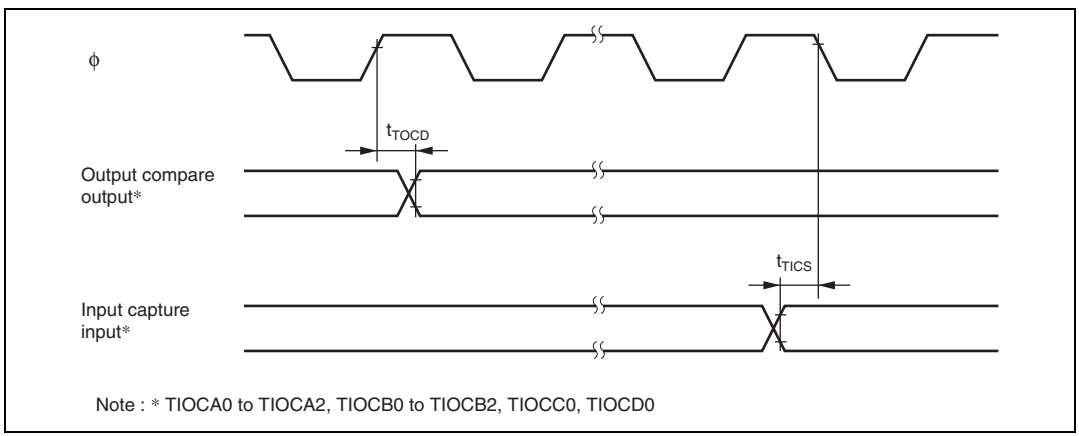


Figure 25.13 TPU Input/Output Timing

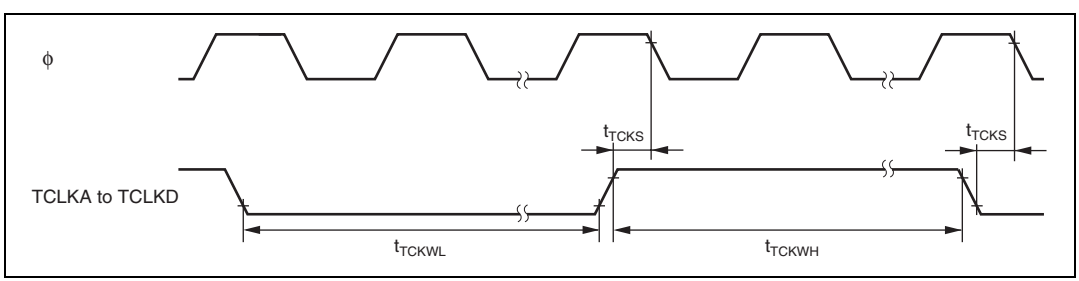


Figure 25.14 TPU Clock Input Timing

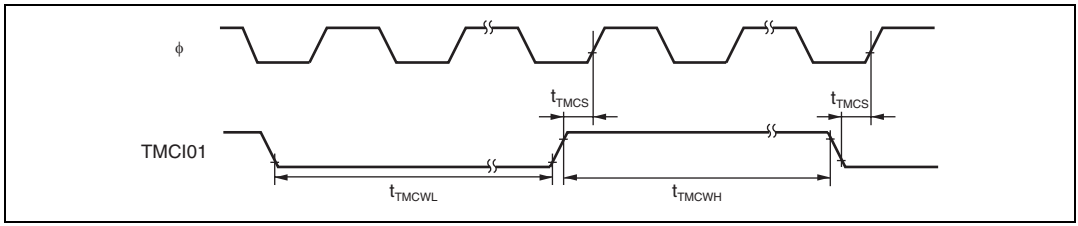


Figure 25.15 8-bit Timer Output Timing

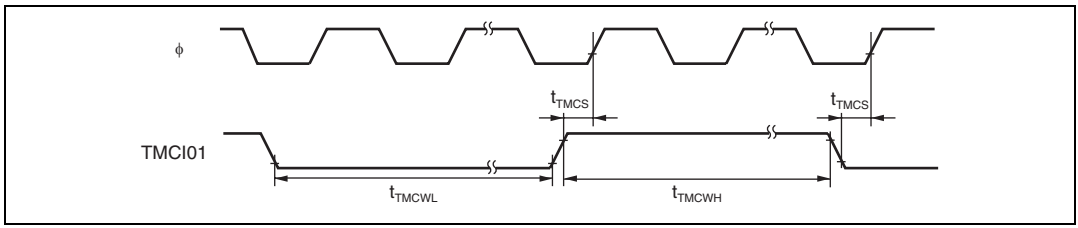


Figure 25.16 8-bit Timer Clock Input Timing

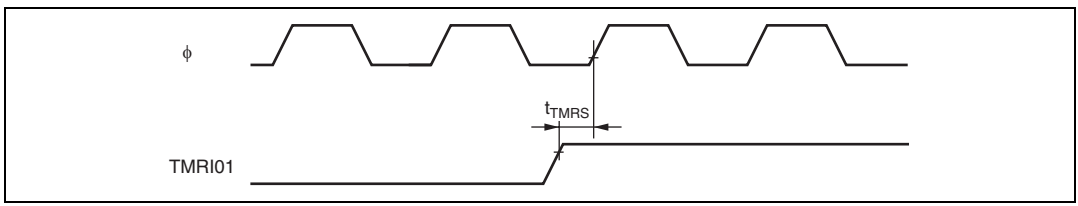


Figure 25.17 8-bit Timer Reset Input Timing

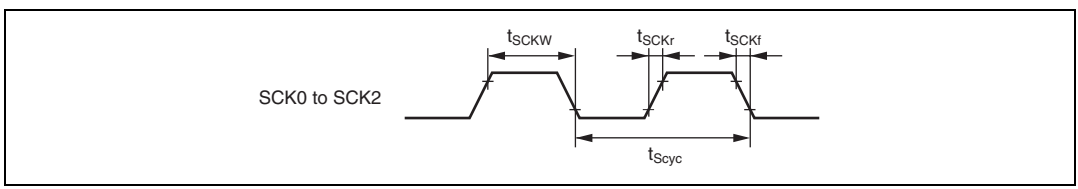


Figure 25.18 SCK Clock Input Timing

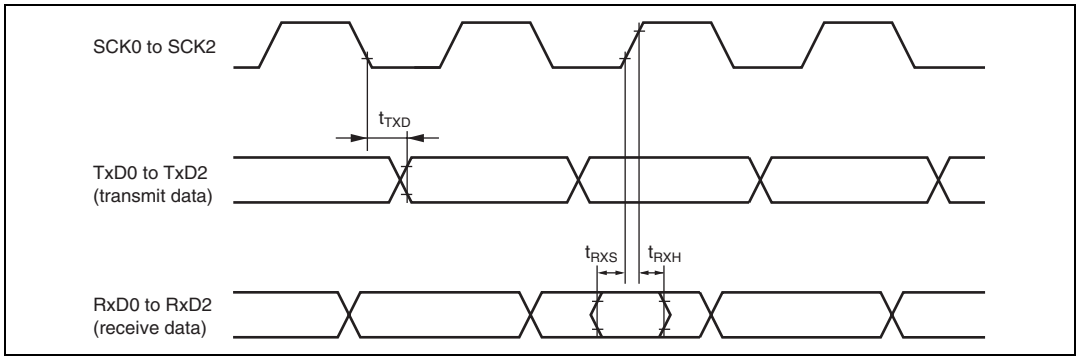


Figure 25.19 SCI Input/Output Timing (Clock Synchronous Mode)

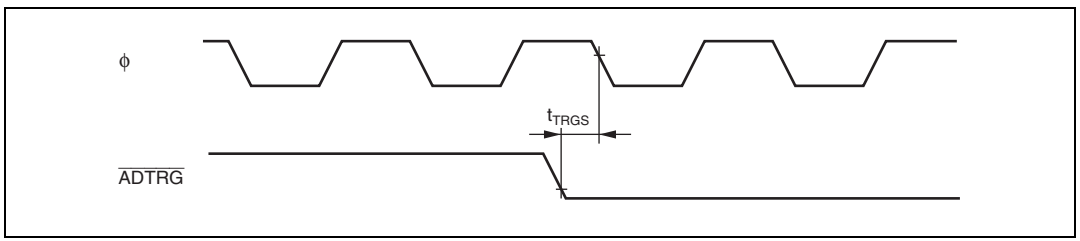


Figure 25.20 A/D Converter External Trigger Input Timing

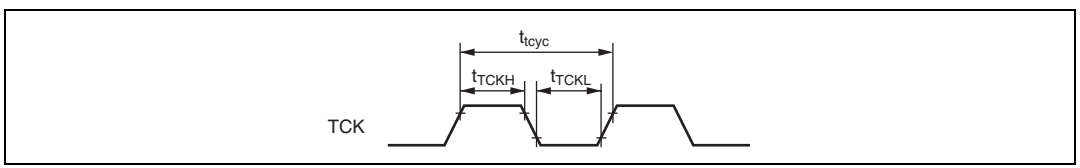


Figure 25.21 Boundary Scan TCK Input Timing

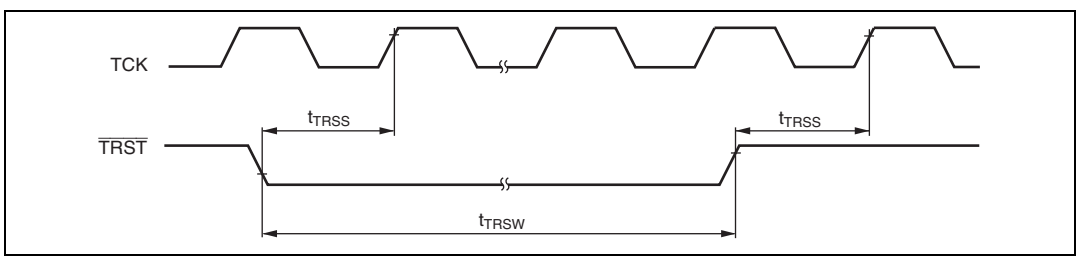


Figure 25.22 Boundary Scan $\overline{TRST}$ Input Timing (At Reset Hold)

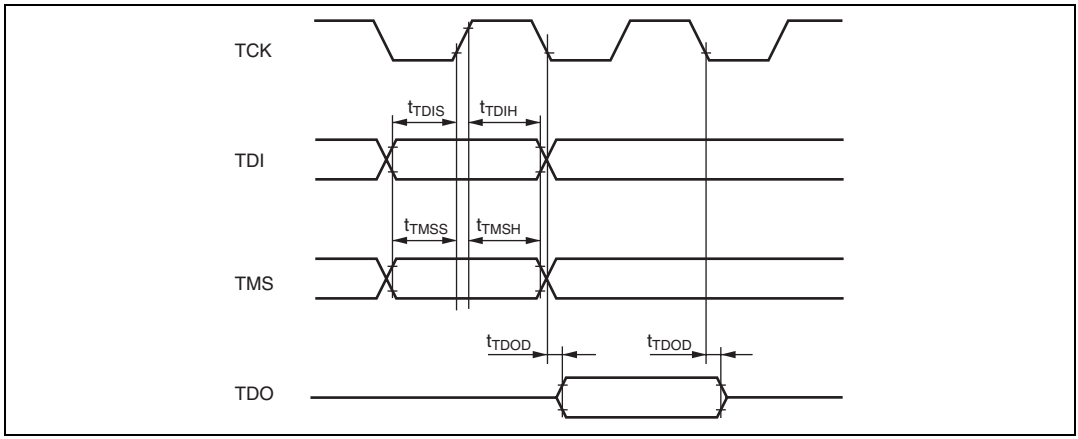


Figure 25.23 Boundary Scan Data Transmission Timing

25.5 USB Characteristics

Table 25.8 lists the USB characteristics (USD+ and USD- pins) when the on-chip USB transceiver is used.

Table 25.8 USB Characteristics (USD+ and USD- pins) when On-Chip USB Transceiver Is Used

Conditions: V_{CC} = $PLLV_{CC}$ = DrV_{CC} = 3.0 V to 3.6 V, V_{SS} = $PLLV_{SS}$ = DrV_{SS} = AV_{SS} = 0 V, ϕ = 13 MHz to 24 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 Condition	
Input characteristics	Input high level voltage	V_{IH}	2.0	—	V		Figures 25.24, 25.25
	Input low level voltage	V_{IL}	—	0.8	V		
	Differential input sense	V_{DI}	0.2	—	V	| (D+)-(D-)| DrVcc = 3.3 to 3.6 V	
	Differential common mode range	V_{CM}	0.8	2.5	V		
Output characteristics	Output high level voltage	V_{OH}	2.8	—	V	I_{OH} = -200 µA	
	Output low level voltage	V_{OL}	—	0.3	V	I_{OL} = 2 mA	
	Crossover voltage	V_{CRS}	1.3	2.0	V		
	Rise time	t_R	4	20	ns		
	Fall time	t_F	4	20	ns		
	Rise time/fall time matching	t_{RFM}	90	111.11	%	(T_R/T_F)	
	Output resistance	Z_{DRV}	28	44	Ω	Including Rs = 24 Ω	

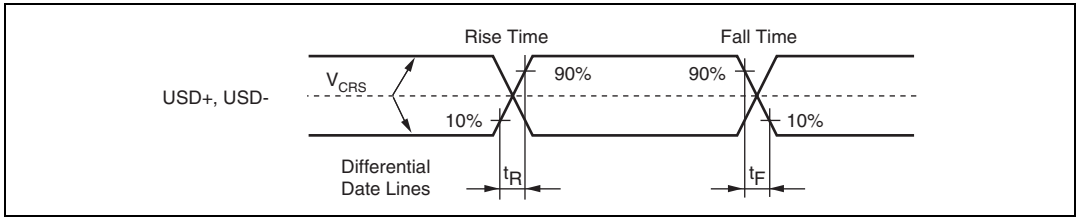


Figure 25.24 Data Signal Timing

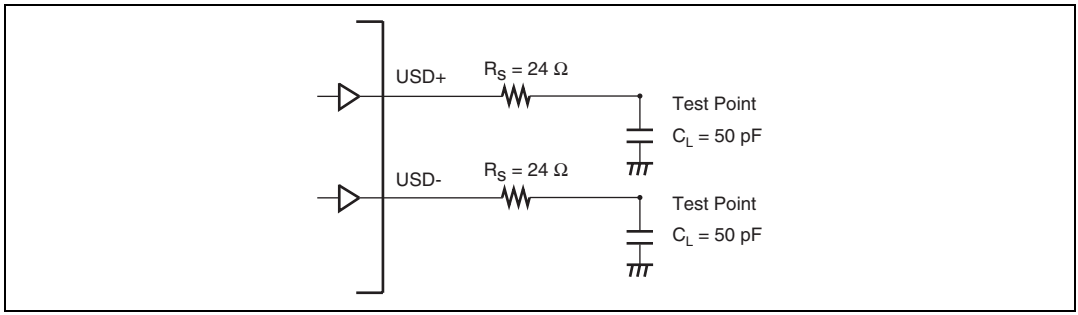


Figure 25.25 Test Load Circuit

25.6 A/D Conversion Characteristics

Table 25.9 lists the A/D conversion characteristics.

Table 25.9 A/D Conversion Characteristics

Condition A: V_{CC} = $PLLV_{CC}$ = DrV_{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} = $PLLV_{SS}$ = DrV_{SS} = AV_{SS} = 0 V, ϕ = 13 MHz to 16 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Condition B: V_{CC} = $PLLV_{CC}$ = DrV_{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} = $PLLV_{SS}$ = DrV_{SS} = AV_{SS} = 0 V, ϕ = 13 MHz to 24 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

	Condition A and B			
Item	**Min.**	**Typ.**	**Max.**	**Unit**
Resolution	10	10	10	bits
Conversion time	8.1	—	—	µs
Analog input capacitance	—	—	20	pF
Permissible signal-source impedance	—	—	5	kΩ
Nonlinearity error	—	—	±6.0	LSB
Offset error	—	—	±4.0	LSB
Full-scale error	—	—	±4.0	LSB
Quantization	—	—	±0.5	LSB
Absolute accuracy	—	—	±8.0	LSB

25.7 D/A Conversion Characteristics

Table 25.10 lists the D/A conversion characteristics.

Table 25.10 D/A Conversion Characteristics

Condition A: V_{CC} = $PLLV_{CC}$ = DrV_{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} = $PLLV_{SS}$ = DrV_{SS} = AV_{SS} = 0 V, ϕ = 13 MHz to 16 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Condition B: V_{CC} = $PLLV_{CC}$ = DrV_{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} = $PLLV_{SS}$ = DrV_{SS} = AV_{SS} = 0 V, ϕ = 13 MHz to 24 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

	Condition A and B				
Item	**Min.**	**Typ.**	**Max.**	**Unit**	**Test Condition**
Resolution	8	8	8	bits	
Conversion time	—	—	10	µs	Load capacitance = 20 pF
Absolute accuracy	—	±2.0	±3.0	LSB	Load capacitance = 2 MΩ
	—	—	±2.0	LSB	Load capacitance = 4 MΩ

25.8 Flash Memory Characteristics

Table 25.11 Flash Memory Characteristics

Condition A: V_{CC} = $PLLV_{CC}$ = DrV_{CC} = 3.0 V to 3.6 V, AV_{CC} = 2.7 V to 3.6 V, V_{ref} = 2.7 V to AV_{CC}, V_{SS} = $PLLV_{SS}$ = DrV_{SS} = AV_{SS} = 0 V, ϕ = 13 MHz to 16 MHz, T_a = –20°C to +75°C (Programming/erasing operating temperature range)

Condition B: V_{CC} = $PLLV_{CC}$ = DrV_{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} = $PLLV_{SS}$ = DrV_{SS} = AV_{SS} = 0 V, ϕ = 13 MHz to 24 MHz, T_a = –20°C to +75°C (Programming/erasing operating temperature range)

Item	**Symbol**	**Min.**	**Typ.**	**Max.**	**Unit**
Programming time*1 *2 *4	t_P	—	10	200	ms/128 bytes
Erase time*1 *3 *5	t_E	—	50	1000	ms/block
Reprogramming count	N_{WEC}	100*6	10,000*7	—	Times
Data retention time*8	t_{DRP}	10	—	—	Years

Item		Symbol	Min.	Typ.	Max.	Unit
Programming	Wait time after PSU1 bit setting*1	y	50	50	—	µs
	Wait time after P1 bit setting*1*4	z0	28	30	32	µs
		z1	198	200	202	µs
		z2	8	10	12	µs
	Wait time after P1 bit clear*1	α	5	5	—	µs
	Wait time after PSU1 bit clear*1	β	5	5	—	µs
	Wait time after PV1 bit setting*1	γ	4	4	—	µs
	Wait time after H'FF dummy write*1	ε	2	2	—	µs
	Wait time after PV1 bit clear*1	η	2	2	—	µs
	Maximum programming count*1*4	N1	—	—	6*4	Times
		N2	—	—	994*4	Times
Common	Wait time after SWE1 bit setting*1	x	1	1	—	µs
	Wait time after SWE1 bit clear*1	θ	100	100	—	µs
Erase	Wait time after ESU1 bit setting*1	y	100	100	—	µs
	Wait time after E1 bit setting*1*5	z	10	10	100	ms
	Wait time after E1 bit clear*1	α	10	10	—	µs
	Wait time after ESU1 bit clear*1	β	10	10	—	µs
	Wait time after EV1 bit setting*1	γ	20	20	—	µs
	Wait time after H'FF dummy write*1	ε	2	2	—	µs
	Wait time after EV1 bit clear*1	η	4	4	—	µs
	Maximum erase count*1*5	N	—	—	100	Times

Notes:
1. Make each time setting in accordance with the program/program-verify flowchart or erase/erase-verify flowchart.
2. Programming time per 128 bytes (Shows the total period for which the P-bit in the flash memory control register (FLMCR1) is set. It does not include the programming verification time).
3. Block erase time (Shows the total period for which the E1-bit FLMCR1 is set. It does not include the erase verification time).
4. Maximum programming time value
   t_p (max.) = Wait time after P1 bit set (z) × maximum programming count (N1 + N2)
   = (Z0 + Z2) × 6 + Z1 × 994
5. Maximum erasure time value
   t_E (max.) = Wait time after E1 bit set (z) × maximum erasure count (N)2
6. Minimum number of times for which all characteristics are guaranteed after rewriting (Guarantee range is 1 to minimum value).
7. Reference value for 25°C (as a guideline, rewriting should normally function up to this value).
8. Data retention characteristic when rewriting is performed within the specification range, including the minimum value.

25.9 Usage Note

General Notice during Design for Printed Circuit Board: Measures for radiation noise caused by the transient current in this LSI should be taken into consideration. The examples of the measures are shown below.

- To use a multilayer printed circuit board which includes layers for Vcc and GND.
- To mount by-pass capacitors (approximately 0.1 μF) between the Vcc and GND (Vss) pins, and the $PLLV_{CC}$ and PLLGND pins, of this LSI.

Section 26 Electrical Characteristics (H8S/2215T)

26.1 Absolute Maximum Ratings

Table 26.1 lists the absolute maximum ratings.

Table 26.1 Absolute Maximum Ratings

Item	Symbol	Value	Unit
Power supply voltage	V_{CC}, $PLLV_{CC}$, DrV_{CC}	–0.3 to +4.3	V
Input voltage (except ports 4 and 9)	V_{in}	–0.3 to V_{CC} +0.3	V
Input voltage (ports 4 and 9)	V_{in}	–0.3 to AV_{CC} +0.3	V
Reference 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	°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.

Note: * The operating temperature ranges for flash memory programming/erasing are T_a = –20°C to +75°C.

26.2 Power Supply Voltage and Operating Frequency Range

Power supply voltage and operating frequency ranges (shaded areas) are shown in figure 26.1.

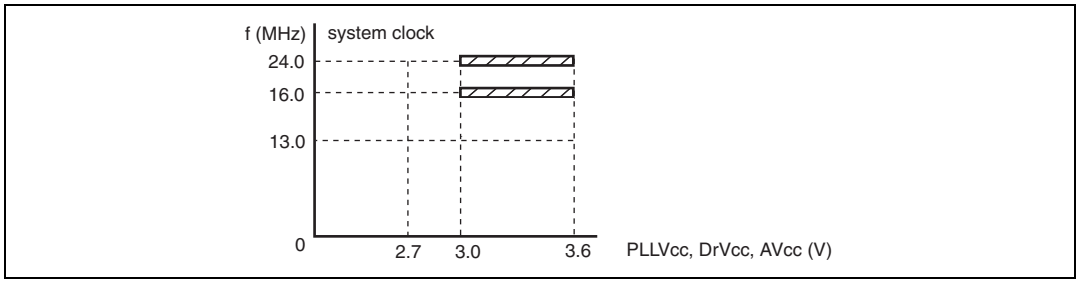


Figure 26.1 Power Supply Voltage and Operating Ranges

26.3 DC Characteristics

Table 26.2 lists the DC characteristics. Table 26.3 lists the permissible output currents.

Table 26.2 DC Characteristics

Conditions: V_{CC} = $PLLV_{CC}$ = DrV_{CC} = 3.0 V to 3.6 V, AV_{CC} = 2.7 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = $PLLV_{SS}$ = DrV_{SS} = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)*1

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Schmitt trigger input voltage	$\overline{IRQ0}$ to $\overline{IRQ5}$ $\overline{IRQ7}$	V_T^-	$V_{CC} \times 0.2$	—	—	V	
		V_T^+	—	—	$V_{CC} \times 0.8$	V	
		$V_T^+ - V_T^-$	$V_{CC} \times 0.05$	—	—	V	
Input high voltage	$\overline{RES}$, $\overline{STBY}$, NMI, MD2 to MD0, $\overline{TRST}$, TCK, TMS, TDI, $\overline{EMLE}$, VBUS, $\overline{UBPM}$, FWE	V_{IH}	$V_{CC} \times 0.9$	—	$V_{CC} + 0.3$	V	
	EXTAL, EXTAL48, Ports 1, 3, 7, and A to G		$V_{CC} \times 0.8$	—	$V_{CC} + 0.3$	V	
	Ports 4 and 9		$V_{CC} \times 0.8$	—	$AV_{CC} + 0.3$	V	
Input low voltage	$\overline{RES}$, $\overline{STBY}$, MD2 to MD0, $\overline{TRST}$, TCK, TMS, TDI, $\overline{EMLE}$, VBUS, $\overline{UBPM}$, FWE	V_{IL}	–0.3	—	$V_{CC} \times 0.1$	V	
	EXTAL, EXTAL48, NMI, Ports 1, 3, 4, 7, 9, and A to G		–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

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Output low voltage	All output pins	V_{OL}	—	—	0.4	V	I_{OH} = 0.4 mA
			—	—	0.4	V	I_{OL} = 0.8 mA
Input leakage current	$\overline{RES}$, $\overline{STBY}$, NMI, MD2 to MD0, FWE*5, VBUS,$\overline{UBPM}$	$\lvert I_{in} \rvert$	—	—	1.0	µA	V_{in} = 0.5 V to V_{CC} – 0.5 V
	Ports 4, 9	$\lvert I_{in} \rvert$	—	—	1.0	µA	V_{in} = 0.5 V to AV_{CC} – 0.5 V
3-state leak current (off status)	Ports 1, 3, 7, A to G	$\lvert I_{TSI} \rvert$	—	—	1.0	µA	V_{in} = 0.5 V to V_{CC} – 0.5 V
Input pull-up MOS current	Ports A to E	$-I_P$	10	—	300	µA	V_{in} = 0 V
Input capacity	$\overline{RES}$, NMI	C_{in}	—	—	30	pF	V_{in} = 0 V
	All input pins other than $\overline{RES}$ and NMI		—	—	15	pF	f = 1 MHz T_a = 25°C
Current dissipation*2	Normal operation (USB halts)	I_{CC}*3	—	23 (V_{CC} = 3.3 V)	40 (V_{CC} = 3.6 V)	mA	f = 16 MHz
			—	34 (V_{CC} = 3.3 V)	55 (V_{CC} = 3.6 V)	mA	f = 24 MHz
	Normal operation (USB operates)		—	28 (V_{CC} = 3.3 V)	50 (V_{CC} = 3.6 V)	mA	f = 16 MHz (PLL 3 multiplication)
			—	40 (V_{CC} = 3.3 V)	60 (V_{CC} = 3.6 V)	mA	f = 24 MHz (PLL 2 multiplication)
	Sleep mode		—	18 (V_{CC} = 3.3 V)	35 (V_{CC} = 3.6 V)	mA	f = 16 MHz (when USB and PLL are halted)
			—	26 (V_{CC} = 3.3 V)	45 (V_{CC} = 3.6 V)	mA	f = 24 MHz (when USB and PLL are halted)

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Current dissipation*2	All modules stopped	I_{CC}*3	—	15 (V_{CC} = 3.3 V)	—	mA	f = 16 MHz (reference value)
			—	21 (V_{CC} = 3.3 V)	—	mA	f = 24 MHz (reference value)
	Standby mode*4		—	1.0	10	µA	$T_a \leq 50°C$
			—	—	50	µA	$50°C < T_a$
Analog power supply current	During A/D conversion	AI_{CC}	—	0.3	1.5	mA	AV_{CC} = 3.3 V
	Idle		—	0.01	5.0	µA	
Reference power supply current	During A/D conversion	AI_{CC}	—	1.2	2.5	mA	V_{ref} = 3.3 V
	Idle		—	0.01	5.0	µA	
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Notes: 1. If the A/D or D/A converter is not used, the AVCC, V_{ref}, and AVSS pins should not be open. Even if the A/D or D/A converter is not used, connect the AVCC and V_{ref} pins to V_{CC} and the AVSS pin to V_{SS}, respectively. In this case, the V_{ref} level should be the AVCC level or less.

2. Current dissipation values are for V_{IH} (min.) = V_{CC} – 0.2 V and V_{IL} (max.) = 0.2 V, with all output pins unloaded and the on-chip MOS pull-up transistors in the off state.

3. I_{CC} depends on V_{CC} and f as follows:
   I_{CC} (max.) = 1.0 (mA) + 0.67 (mA/(MHz x V)) × V_{CC} × f (normal operation, USB halted)
   I_{CC} (max.) = 1.0 (mA) + 0.85 (mA/(MHz x V)) × V_{CC} × f (normal operation, USB operated)
   I_{CC} (max.) = 1.0 (mA) + 0.59 (mA/(MHz x V)) × V_{CC} × f (sleep mode)

4. The values are for $V_{RAM} \leq V_{CC} < 3.0$ V, V_{IH} (min.) = V_{CC} × 0.9, and V_{IL} (max.) = 0.3 V.

Table 26.3 Permissible Output Currents

Conditions: V_{CC} = $PLLV_{CC}$ = DrV_{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} = $PLLV_{SS}$ = DrV_{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	V_{CC} = 3.0 to 3.6 V	I_{OL}	—	—	1.0	mA
Permissible output low current (total)	Total of all output pins	V_{CC} = 3.0 to 3.6 V	$\sum I_{OL}$	—	—	60	mA
Permissible output high current (per pin)	All output pins	V_{CC} = 3.0 to 3.6 V	$-I_{OH}$	—	—	1.0	mA
Permissible output high current (total)	Total of all output pins	V_{CC} = 3.0 to 3.6 V	$\sum -I_{OH}$	—	—	30	mA

Note: * To protect chip reliability, do not exceed the output current values in table 26.3.

26.4 AC Characteristics

Figure 26.2 shows, the test conditions for the AC characteristics.

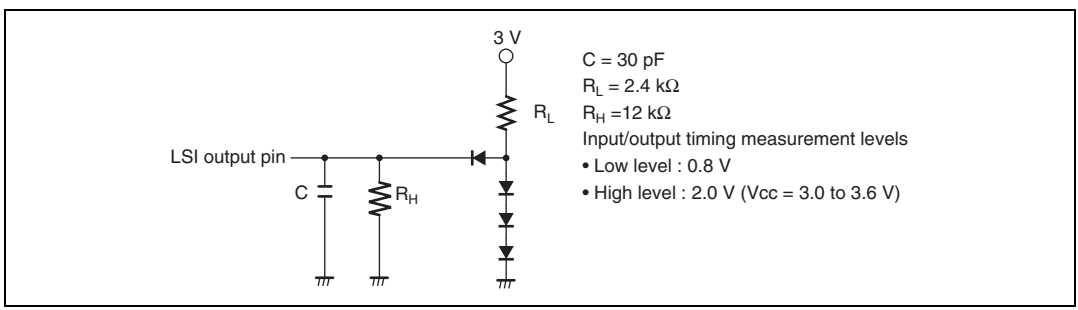


Figure 26.2 Output Load Circuit

26.4.1 Clock Timing

Table 26.4 lists the clock timing

Table 26.4 Clock Timing

Conditions: V_{CC} = $PLLV_{CC}$ = DrV_{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} = $PLLV_{SS}$ = DrV_{SS} = AV_{SS} = 0 V, ϕ = 16 MHz, 24 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}	41.6	62.5	ns	Figure 26.3
Clock high pulse width	t_{CH}	13	—	ns	
Clock low pulse width	t_{CL}	13	—	ns	
Clock rise time	t_{Cr}	—	7	ns	
Clock fall time	t_{Cf}	—	7	ns	
Oscillation stabilization time at reset (crystal)	t_{OSC1}	20	—	ms	Figure 26.4
Oscillation stabilization time in software standby (crystal)	t_{OSC2}	8	—	ms	C_{L1} = C_{L2} = 10 pF to 22 pF in figure 21.2 Figure 22.3
		4	—	ms	C_{L1} = C_{L2} = 10 pF to 15 pF in figure 21.2 V_{CC} = 3.0 V to 3.6 V in figure 22.3
External clock output stabilization delay time	t_{DEXT}	500	—	µs	Figure 26.4
USB operating clock (48 MHz) stabilization time	t_{OSC3}	8	—	ms	V_{CC} = 3.0 V to 3.6 V
USB operating clock (48 MHz) oscillator frequency	f_{48}	4.8		MHz	
USB operating clock (48 MHz) cycle time	f_{48}	20.8		ns	

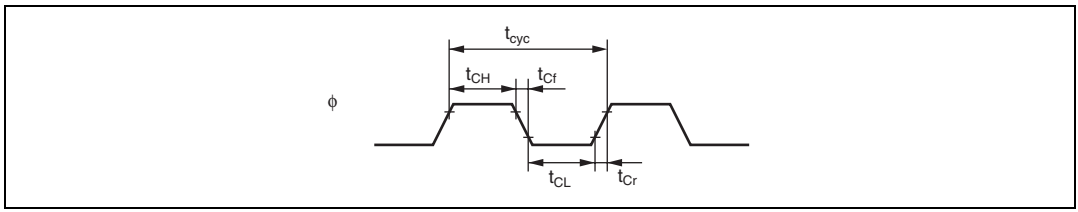


Figure 26.3 System Clock Timing

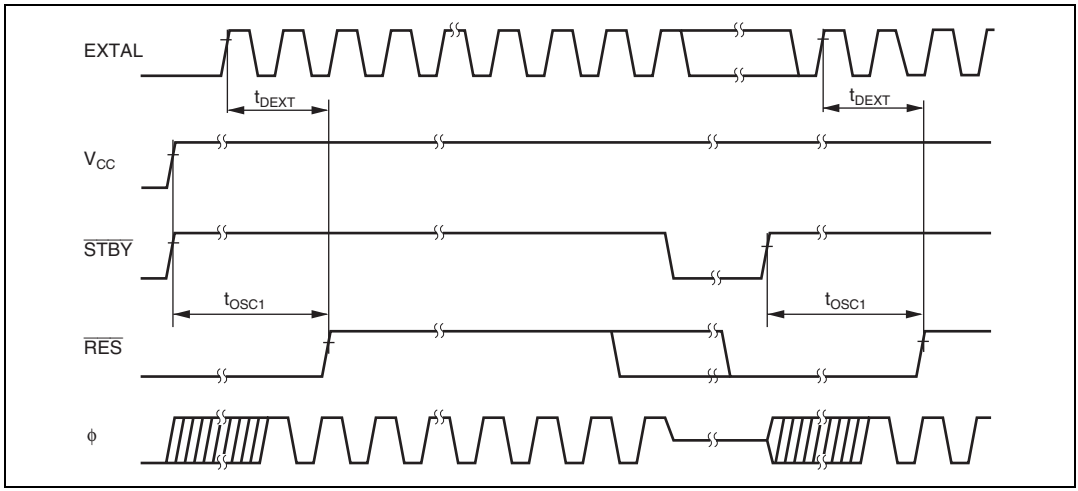


Figure 26.4 Oscillation Stabilization Timing

26.4.2 Control Signal Timing

Table 26.5 lists the control signal timing.

Table 26.5 Control Signal Timing

Conditions: V_{CC} = $PLLV_{CC}$ = DrV_{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} = $PLLV_{SS}$ = DrV_{SS} = AV_{SS} = 0 V, φ = 16 MHz, 24 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
$\overline{RES}$ setup time	t_{RESS}	250	—	ns	Figure 26.5
$\overline{RES}$ pulse width	t_{RESW}	20	—	t_{cyc}	
$\overline{MRES}$ setup time	t_{MRESS}	250	—	ns	
$\overline{MRES}$ pulse width	t_{MRESW}	20	—	t_{cyc}	
NMI setup time	t_{NMIS}	250	—	ns	Figure 26.6
NMI hold time	t_{NMIH}	10	—	ns	
NMI pulse width (exiting software standby mode)	t_{NMIW}	200	—	ns	
$\overline{IRQ}$ setup time	t_{IRQS}	250	—	ns	
$\overline{IRQ}$ hold time	t_{IRQH}	10	—	ns	
$\overline{IRQ}$ pulse width (exiting software standby mode)	t_{IRQW}	200	—	ns	

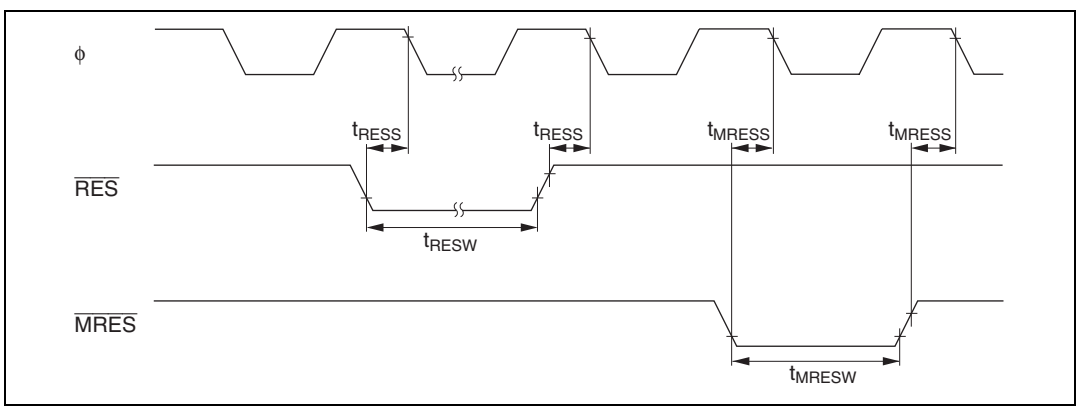


Figure 26.5 Reset Input Timing

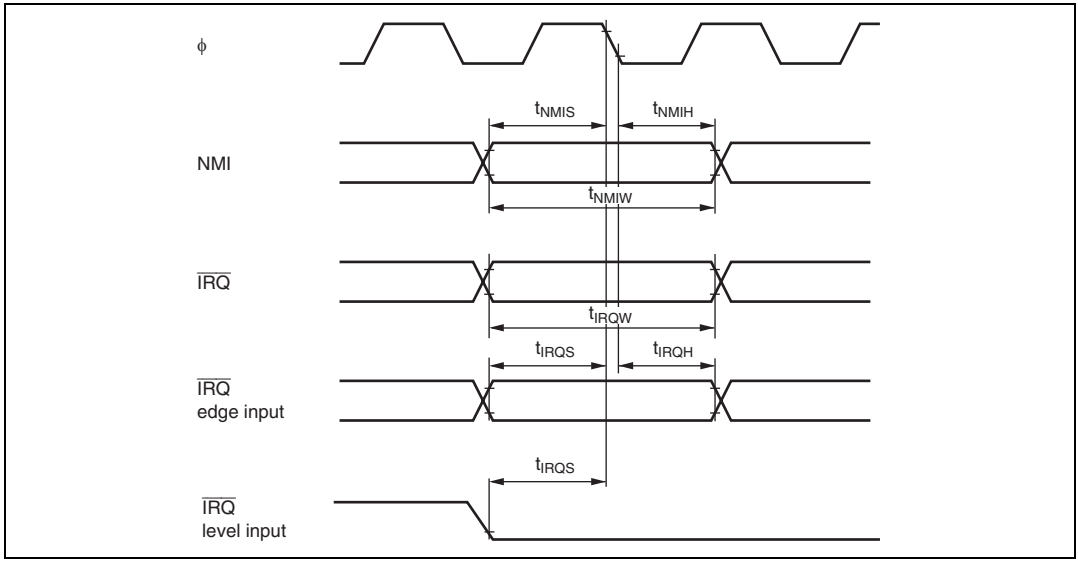


Figure 26.6 Interrupt Input Timing

26.4.3 Bus Timing

Table 26.6 shows, Bus Timing.

Table 26.6 Bus Timing

Conditions: V_{CC} = $PLLV_{CC}$ = DrV_{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} = $PLLV_{SS}$ = DrV_{SS} = AV_{SS} = 0 V, ϕ = 16 MHz, 24 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}	—	30	ns	Figures 26.7, 26.8, 26.10
Address setup time	t_{AS}	0.5 × t_{cyc} – 20	—	ns	
Address hold time	t_{AH}	0.5 × t_{cyc} – 8	—	ns	
$\overline{CS}$ delay time	t_{CSD}	—	30	ns	Figures 26.7, 26.8
$\overline{AS}$ delay time	t_{ASD}	—	25	ns	Figures 26.7, 26.8, 26.10
$\overline{RD}$ delay time 1	t_{RSD1}	—	25	ns	Figures 26.7, 26.8
$\overline{RD}$ delay time 2	t_{RSD2}	—	25	ns	Figures 26.7, 26.8, 26.10
Read data setup time	t_{RDS}	20	—	ns	
Read data hold time	t_{RDH}	0	—	ns	
Read data access time 2	t_{ACC2}	—	1.5 × t_{cyc} – 35	ns	Figure 26.7
Read data access time 3	t_{ACC3}	—	2.0 × t_{cyc} – 40	ns	Figures 26.7, 26.10
Read data access time 4	t_{ACC4}	—	2.5 × t_{cyc} – 35	ns	Figure 26.8
Read data access time 5	t_{ACC5}	—	3.0 × t_{cyc} – 40	ns	
$\overline{WR}$ delay time 1	t_{WRD1}	—	20	ns	
$\overline{WR}$ delay time 2	t_{WRD2}	—	25	ns	Figures 26.7, 26.8
$\overline{WR}$ pulse width 1	t_{WSW1}	1.0 × t_{cyc} – 20	—	ns	Figure 26.7
$\overline{WR}$ pulse width 2	t_{WSW2}	1.5 × t_{cyc} – 20	—	ns	Figure 26.8
Write data delay time	t_{WDD}	—	30	ns	Figures 26.7, 26.8
Write data setup time	t_{WDS}	0.5 × t_{cyc} – 20	—	ns	Figure 26.8
Write data hold time	t_{WDH}	0.5 × t_{cyc} – 10	—	ns	Figures 26.7, 26.8
$\overline{WAIT}$ setup time	t_{WTS}	25	—	ns	Figure 26.9
$\overline{WAIT}$ hold time	t_{WTH}	5	—	ns	

Item	Symbol	Min.	Max.	Unit	Test Conditions
$\overline{BREQ}$ setup time	t_{BRQS}	25	—	ns	Figure 26.11
$\overline{BACK}$ delay time	t_{BACD}	—	40	ns	
Bus-floating time	t_{BZD}	—	50	ns	

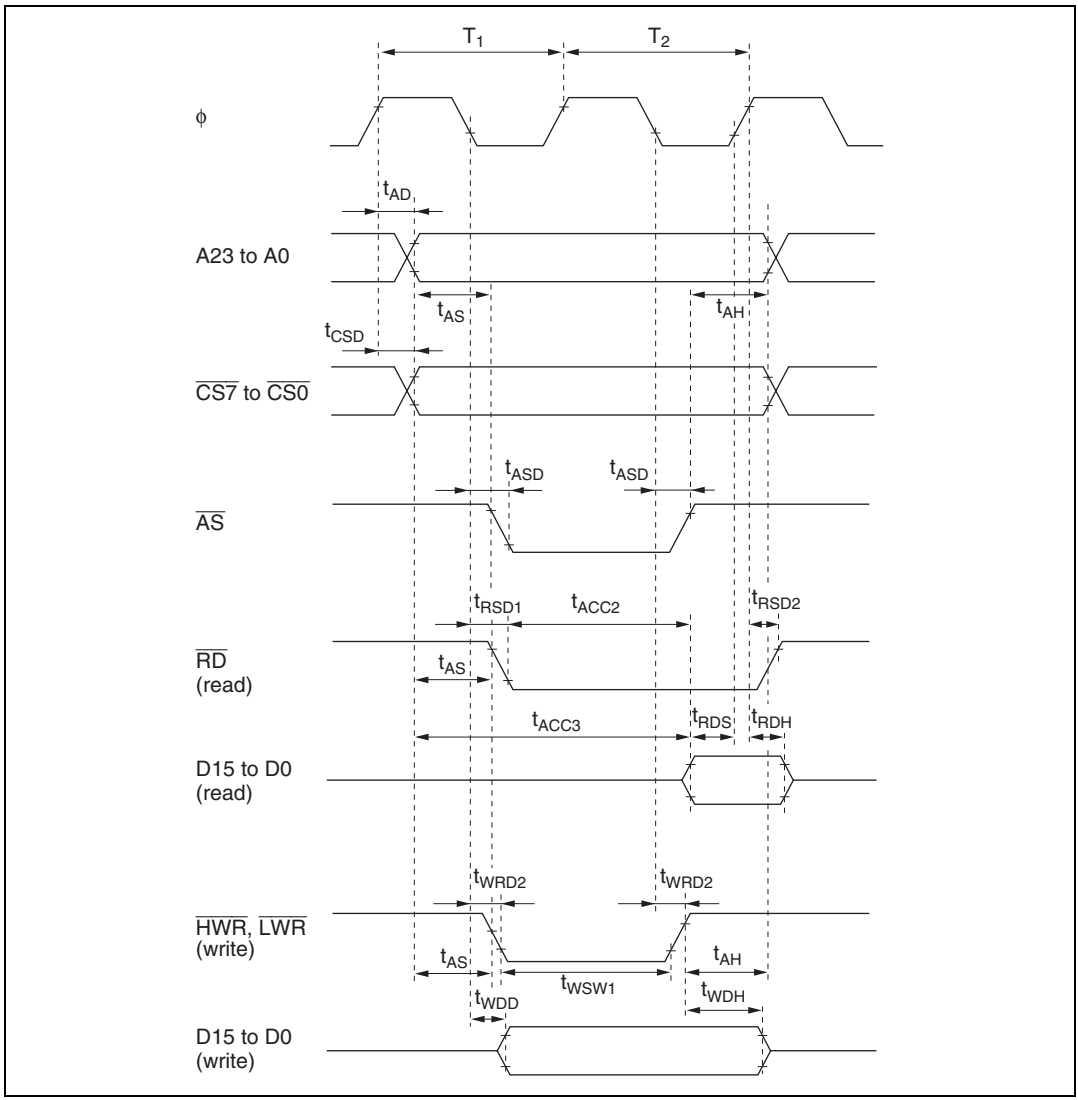


Figure 26.7 Basic Bus Timing (Two-State Access)

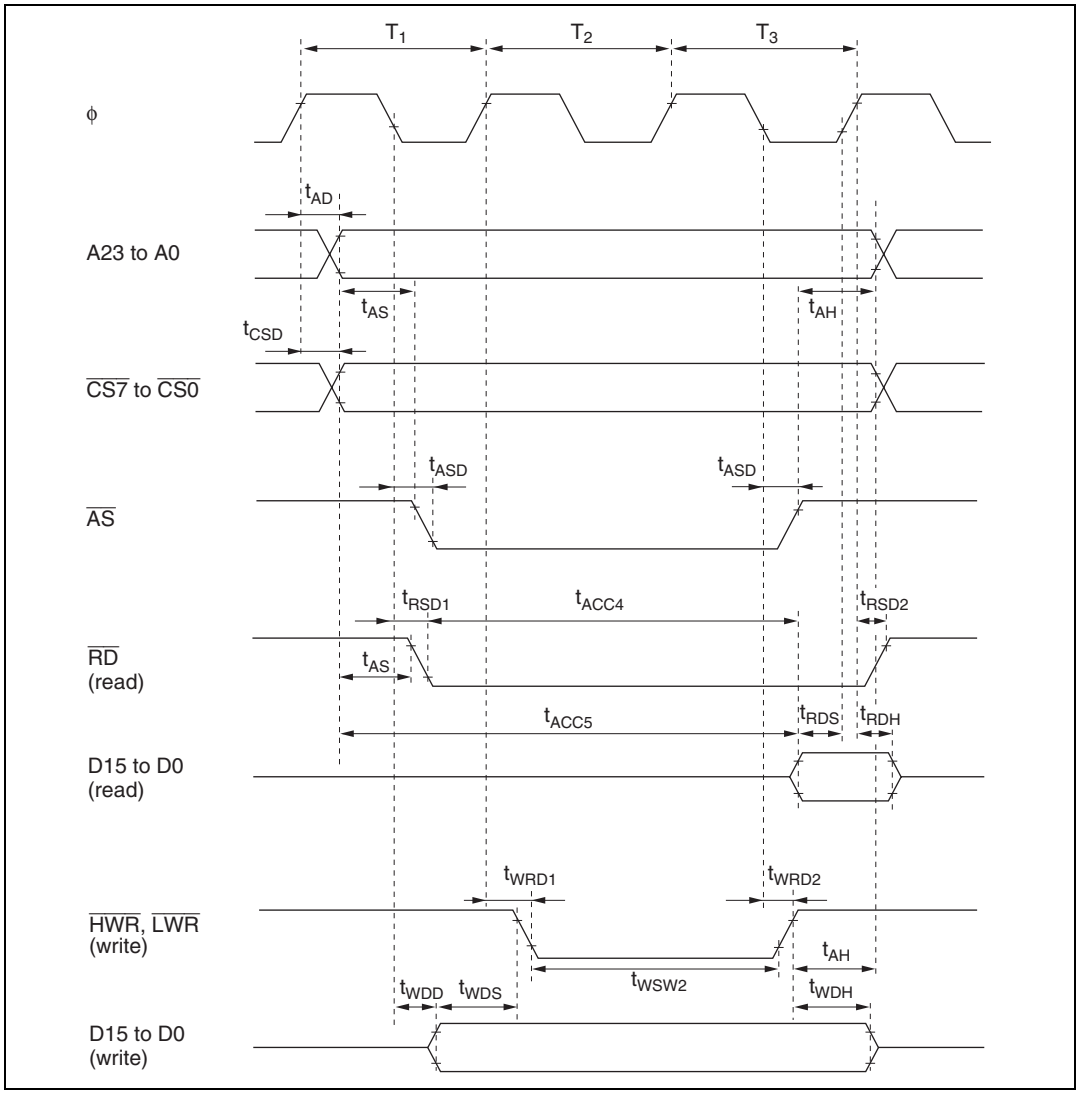


Figure 26.8 Basic Bus Timing (Three-State Access)

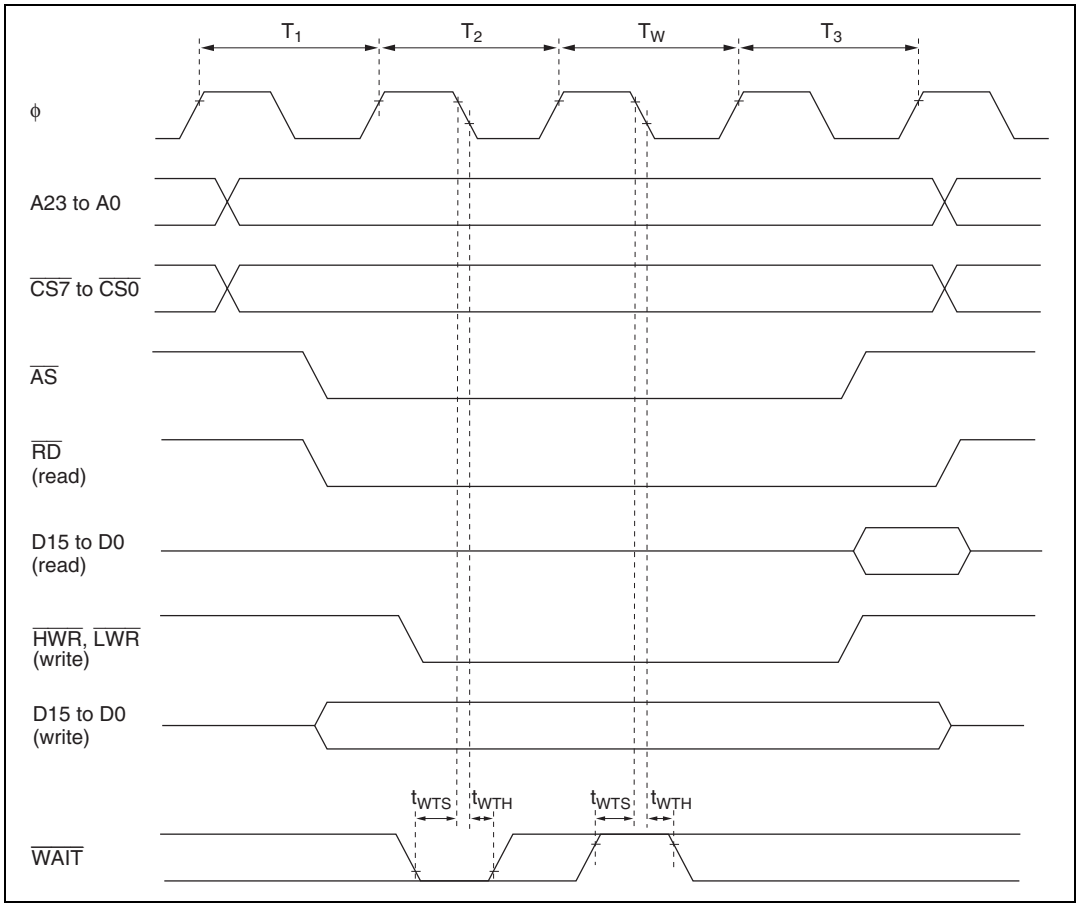


Figure 26.9 Basic Bus Timing (Three-State Access with One Wait State)

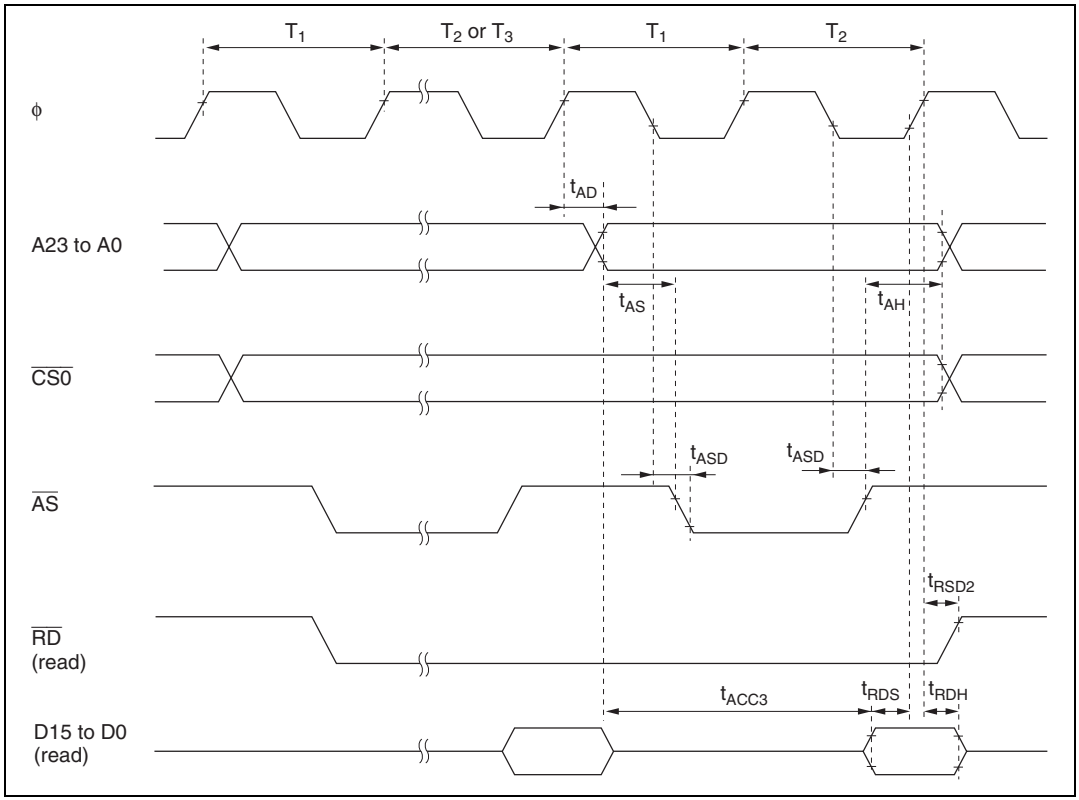


Figure 26.10 Burst ROM Access Timing (Two-State Access)

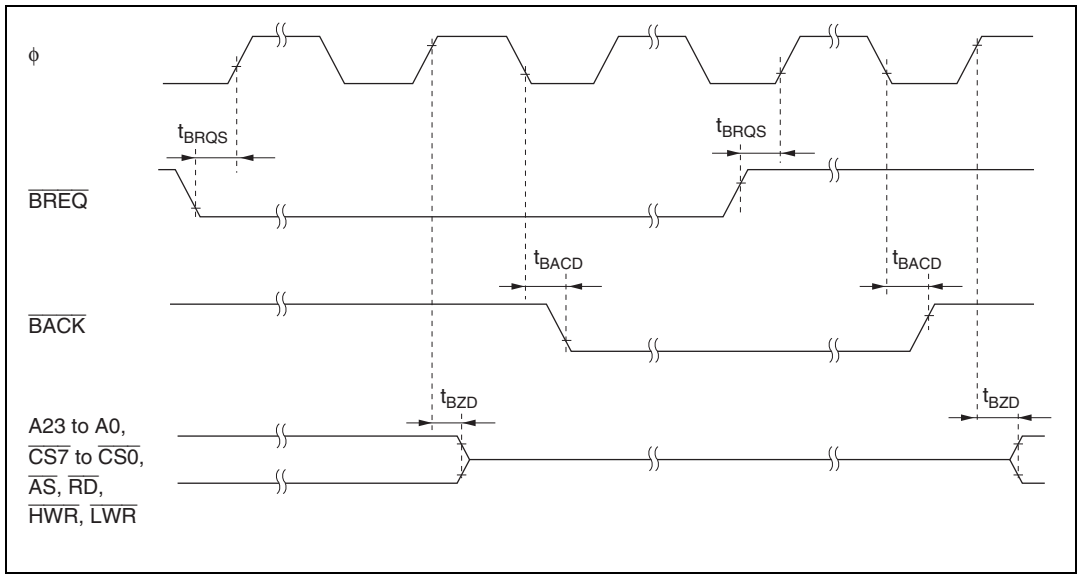


Figure 26.11 External Bus Release Timing

26.4.4 Timing of On-Chip Supporting Modules

Table 26.7 lists the timing of on-chip supporting modules.

Table 26.7 Timing of On-Chip Supporting Modules

Conditions: V_{CC} = $PLLV_{CC}$ = DrV_{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} = $PLLV_{SS}$ = DrV_{SS} = AV_{SS} = 0 V, ϕ = 16 MHz, 24 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
			t_{PWD}	—	40	ns	Figure 26.12
	Input data setup time		t_{PRS}	30	—		
	Input data hold time		t_{PRH}	30	—		
TPU	Timer output delay time		t_{TOCD}	—	40	ns	Figure 26.13
	Timer input setup time		t_{TICS}	30	—		
	Timer clock input setup time		t_{TCKS}	30	—	ns	Figure 26.14
	Timer clock pulse width	Single edge	t_{TCKWH}	1.5	—	t_{cyc}	
		Both edges	t_{TCKWL}	2.5	—		
TMR	Timer output delay time		t_{TMOD}	—	41	ns	Figure 26.15
	Timer reset input setup time		t_{TMRS}	29	—	ns	Figure 26.17
	Timer clock input setup time		t_{TMCS}	29	—	ns	Figure 26.16
TMR	Timer clock pulse width	Single edge	t_{TMCWH}	1.5	—	t_{cyc}	Figure 26.16
		Both edges	t_{TMCWL}	2.5	—		
SCI	Input clock cycle	Asynchronous	t_{Scyc}	4	—	t_{cyc}	Figure 26.18
		Synchronous		6	—		
	Input clock pulse width		t_{SCKW}	0.4	0.6	t_{Scyc}	
	Input clock rise time		t_{SCKr}	—	1.5	t_{cyc}	
	Input clock fall time		t_{SCKf}	—	1.5		
	Transmit data delay time		t_{TXD}	—	40	ns	Figure 26.19
	Receive data setup time (synchronous)		t_{RXS}	40	—		
	Receive data hold time (synchronous)		t_{RXH}	40	—		
A/D converter	Trigger input setup time		t_{TRGS}	30	—	ns	Figure 26.20

Item		Symbol	Min.	Max.	Unit	Test Conditions
Boundary scan	TCK cycle time	t_{cyc}	41.6	—	ns	Figure 26.21
	TCK high level pulse width	t_{TCKH}	0.4	0.6	t_{cyc}	
	TCK low level pulse width	t_{TCKL}	0.4	0.6	t_{cyc}	
	$\overline{\text{TRST}}$ pulse width	t_{TRSW}	20	—	t_{cyc}	Figure 25.22
	$\overline{\text{TRST}}$ setup time	t_{TRSS}	250	—	ns	
	TDI setup time	t_{TDIS}	20	—	ns	Figure 26.23
	TDI hold time	t_{TDIH}	10	—		
	TMS setup time	t_{TMSS}	20	—		
	TMS hold time	t_{TMSH}	10	—		
	TDO delay time	t_{TDOD}	—	35		

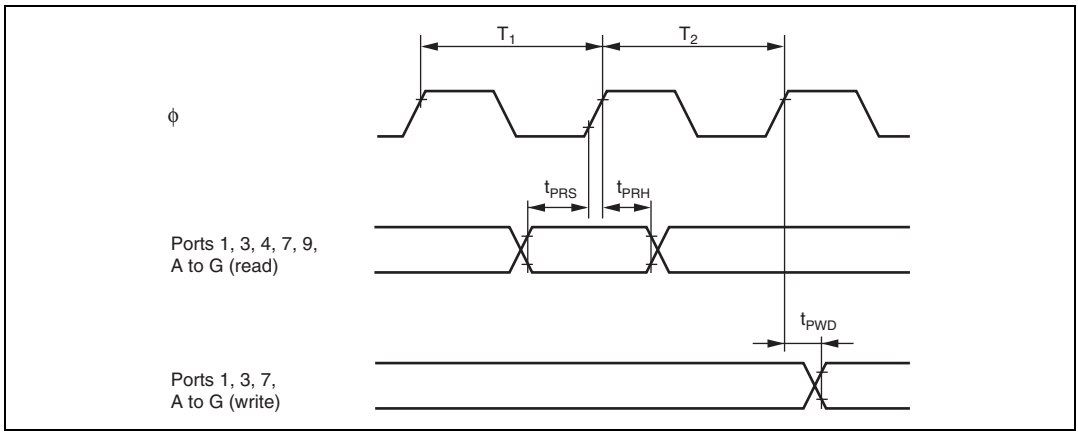


Figure 26.12 I/O Port Input/Output Timing

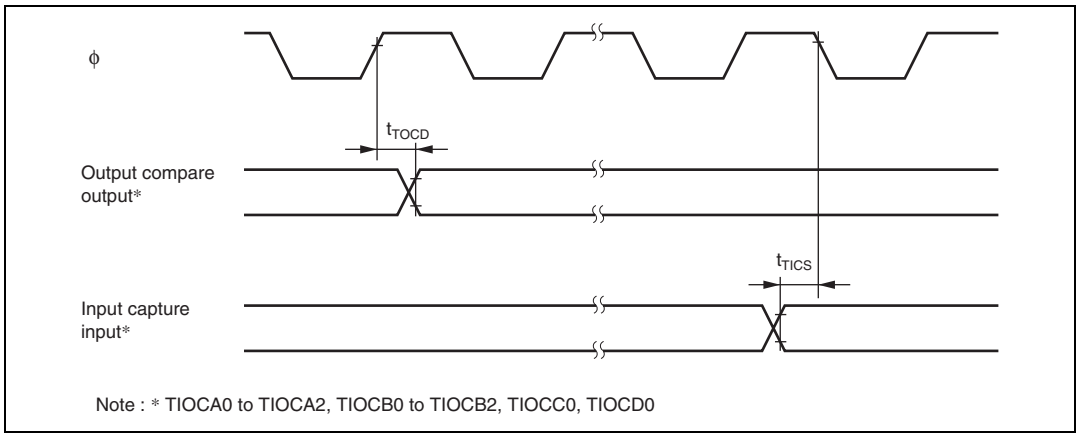


Figure 26.13 TPU Input/Output Timing

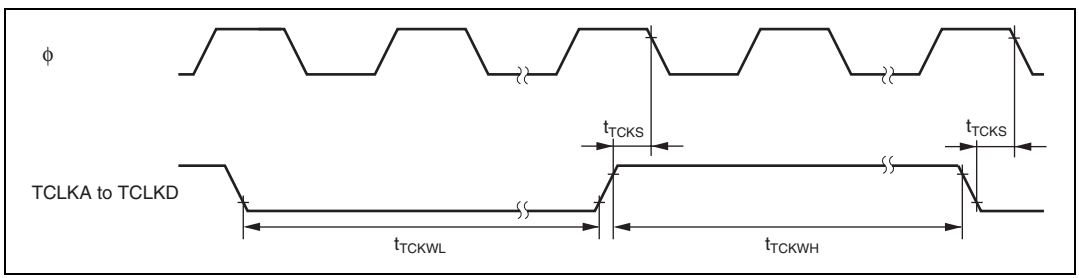


Figure 26.14 TPU Clock Input Timing

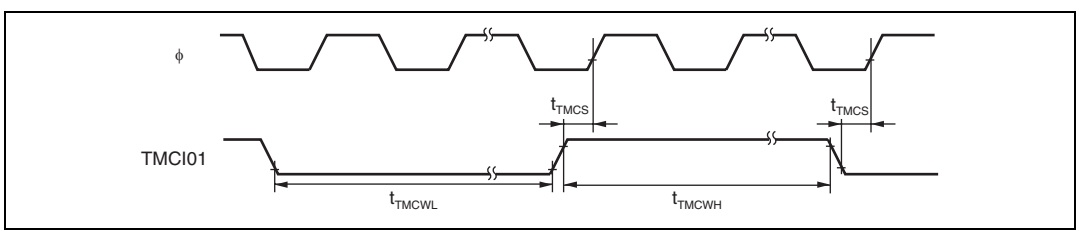


Figure 26.15 8-bit Timer Output Timing

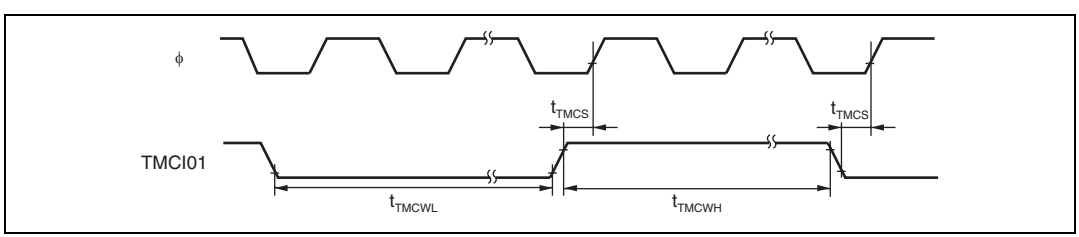


Figure 26.16 8-bit Timer Clock Input Timing

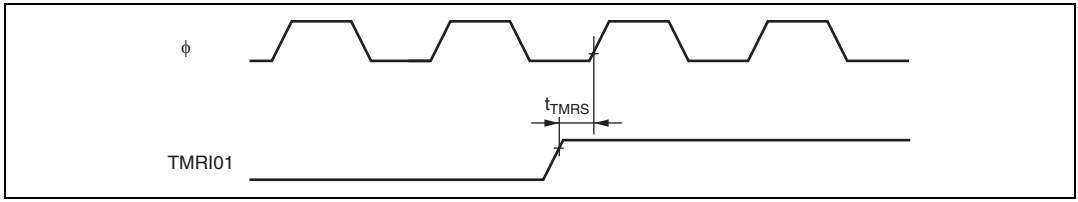


Figure 26.17 8-bit Timer Reset Input Timing

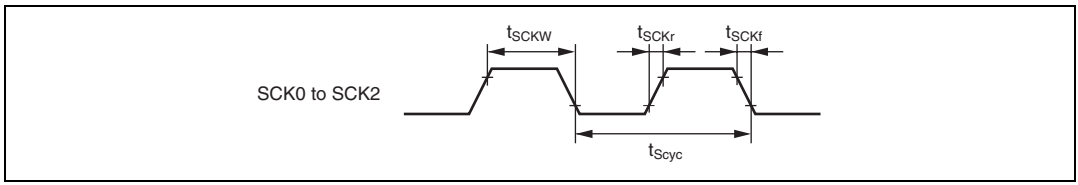


Figure 26.18 SCK Clock Input Timing

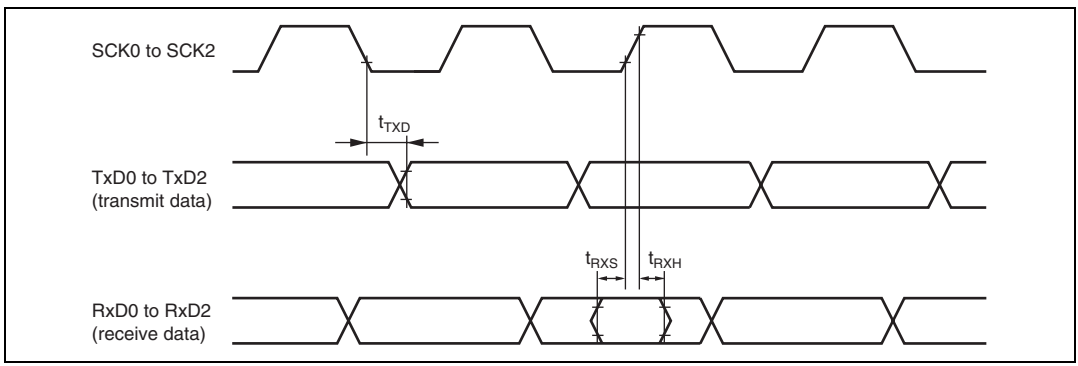


Figure 26.19 SCI Input/Output Timing (Clock Synchronous Mode)

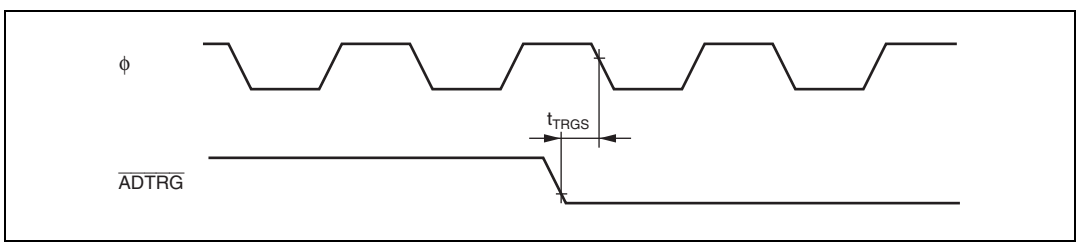


Figure 26.20 A/D Converter External Trigger Input Timing

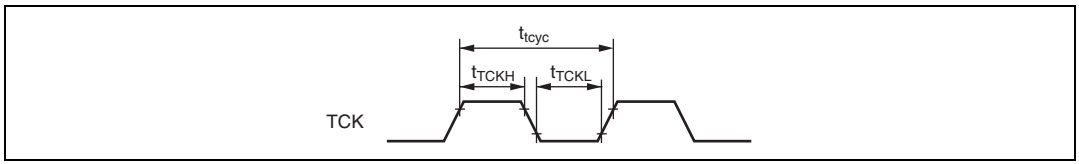


Figure 26.21 Boundary Scan TCK Input Timing

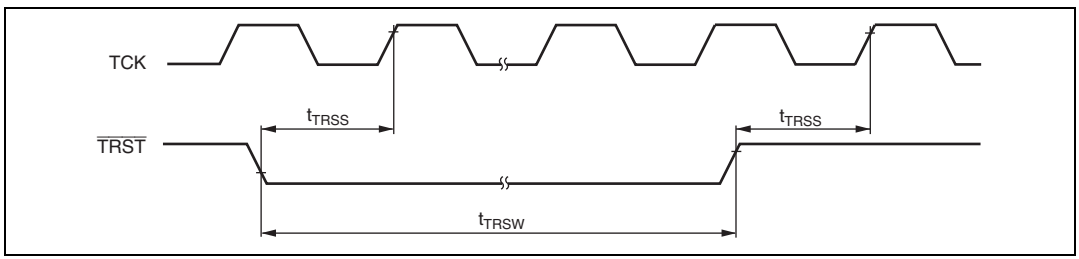


Figure 26.22 Boundary Scan $\overline{TRST}$ Input Timing (At Reset Hold)

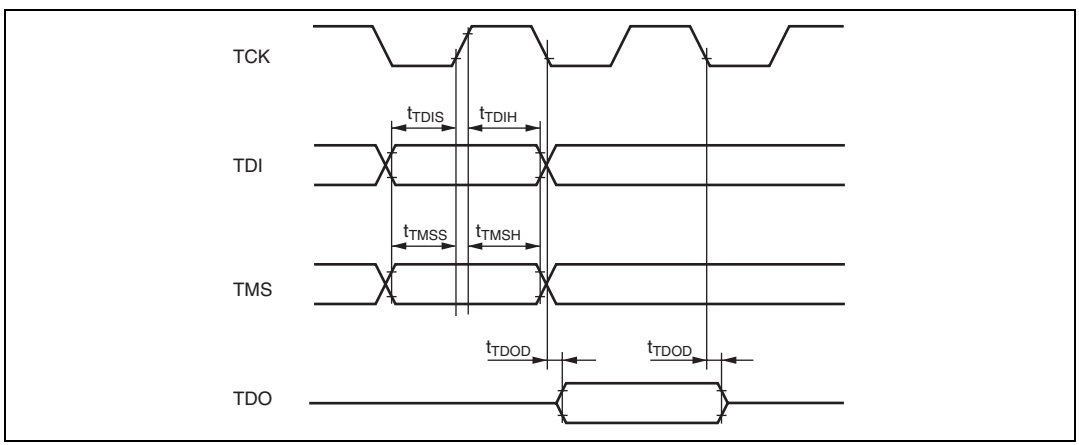


Figure 26.23 Boundary Scan Data Transmission Timing

26.5 USB Characteristics

Table 26.8 lists the USB characteristics (USD+ and USD- pins) when the on-chip USB transceiver is used.

Table 26.8 USB Characteristics (USD+ and USD- pins) when On-Chip USB Transceiver Is Used

Conditions: V_{CC} = $PLLV_{CC}$ = DrV_{CC} = 3.0 V to 3.6 V, V_{SS} = $PLLV_{SS}$ = DrV_{SS} = AV_{SS} = 0 V, ϕ = 16 MHz, 24 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 Condition	
Input characteristics	Input high level voltage	V_{IH}	2.0	—	V		Figures 26.24, 26.25
	Input low level voltage	V_{IL}	—	0.8	V		
	Differential input sense	V_{DI}	0.2	—	V	| (D+)-(D-)| DrVcc = 3.3 to 3.6 V	
	Differential common mode range	V_{CM}	0.8	2.5	V		
Output characteristics	Output high level voltage	V_{OH}	2.8	—	V	I_{OH} = -200 μA	
	Output low level voltage	V_{OL}	—	0.3	V	I_{OL} = 2 mA	
	Crossover voltage	V_{CRS}	1.3	2.0	V		
	Rise time	t_R	4	20	ns		
	Fall time	t_F	4	20	ns		
	Rise time/fall time matching	t_{RFM}	90	111.11	%	(T_R/T_F)	
	Output resistance	Z_{DRV}	28	44	Ω	Including Rs = 24 Ω	

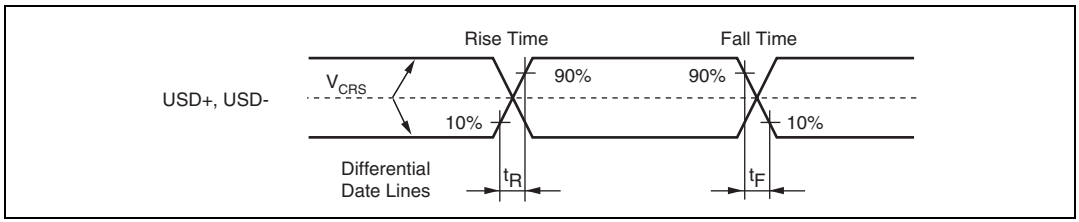


Figure 26.24 Data Signal Timing

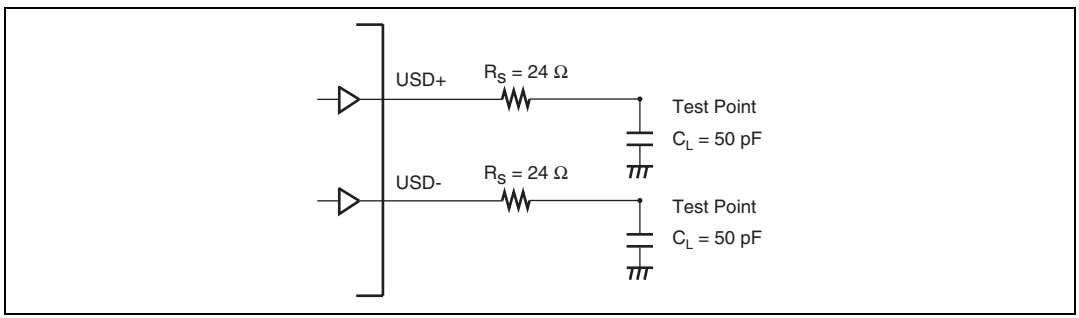


Figure 26.25 Test Load Circuit

26.6 A/D Conversion Characteristics

Table 26.9 lists the A/D conversion characteristics.

Table 26.9 A/D Conversion Characteristics

Conditions: V_{CC} = $PLLV_{CC}$ = DrV_{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} = $PLLV_{SS}$ = DrV_{SS} = AV_{SS} = 0 V, ϕ = 16 MHz to, 24 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	8.1	—	—	µs
Analog input capacitance	—	—	20	pF
Permissible signal-source impedance	—	—	5	kΩ
Nonlinearity error	—	—	±6.0	LSB
Offset error	—	—	±4.0	LSB
Full-scale error	—	—	±4.0	LSB
Quantization	—	—	±0.5	LSB
Absolute accuracy	—	—	±8.0	LSB

26.7 D/A Conversion Characteristics

Table 26.10 lists the D/A conversion characteristics.

Table 26.10 D/A Conversion Characteristics

Conditions: V_{CC} = $PLLV_{CC}$ = DrV_{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} = $PLLV_{SS}$ = DrV_{SS} = AV_{SS} = 0 V, ϕ = 16 MHz, 24 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 Condition
Resolution	8	8	8		
Conversion time	—	—	10	µs	Load capacitance = 20 pF
Absolute accuracy	—	±2.0	±3.0	LSB	Load capacitance = 2 MΩ
	—	—	±2.0	LSB	Load capacitance = 4 MΩ

26.8 Flash Memory Characteristics

Table 26.11 Flash Memory Characteristics

Conditions: V_{CC} = $PLLV_{CC}$ = DrV_{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} = $PLLV_{SS}$ = DrV_{SS} = AV_{SS} = 0 V, ϕ = 16 MHz, 24 MHz, T_a = –20°C to +75°C (Programming/erasing operating temperature range)

Item		Symbol	Min.	Typ.	Max.	Unit
Programming time*1 *2 *4		t_P	—	10	200	ms/128 bytes
Erase time*1 *3 *5		t_E	—	50	1000	ms/block
Reprogramming count		N_{WEC}	100*6	10,000*7	—	Times
Data retention time*8		t_{DRP}	10	—	—	Years
Programming	Wait time after PSU1 bit setting*1	y	50	50	—	µs
	Wait time after P1 bit setting*1*4	z0	28	30	32	µs
		z1	198	200	202	µs
		z2	8	10	12	µs
	Wait time after P1 bit clear*1	α	5	5	—	µs
	Wait time after PSU1 bit clear*1	β	5	5	—	µs
	Wait time after PV1 bit setting*1	γ	4	4	—	µs
	Wait time after H'FF dummy write*1	ε	2	2	—	µs
	Wait time after PV1 bit clear*1	η	2	2	—	µs
	Maximum programming count*1*4	N1	—	—	6*4	Times
		N2	—	—	994*4	Times

Item		Symbol	Min.	Typ.	Max.	Unit
Common	Wait time after SWE1 bit setting*1	x	1	1	—	µs
	Wait time after SWE1 bit clear*1	θ	100	100	—	µs
Erase	Wait time after ESU1 bit setting*1	y	100	100	—	µs
	Wait time after E1 bit setting*1*5	z	10	10	100	ms
	Wait time after E1 bit clear*1	α	10	10	—	µs
	Wait time after ESU1 bit clear*1	β	10	10	—	µs
	Wait time after EV1 bit setting*1	γ	20	20	—	µs
	Wait time after H'FF dummy write*1	ε	2	2	—	µs
	Wait time after EV1 bit clear*1	η	4	4	—	µs
	Maximum erase count*1*5	N	—	—	100	Times

Notes: 1. Make each time setting in accordance with the program/program-verify flowchart or erase/erase-verify flowchart.

2. Programming time per 128 bytes (Shows the total period for which the P-bit in the flash memory control register (FLMCR1) is set. It does not include the programming verification time).
3. Block erase time (Shows the total period for which the E1-bit FLMCR1 is set. It does not include the erase verification time).
4. Maximum programming time value

t_p (max.) = Wait time after P1 bit set (z) × maximum programming count (N1 + N2)
= (Z0 + Z2) × 6 + Z1 × 994

5. Maximum erasure time value
t_E (max.) = Wait time after E1 bit set (z) × maximum erasure count (N)2
6. Minimum number of times for which all characteristics are guaranteed after rewriting (Guarantee range is 1 to minimum value).
7. Reference value for 25°C (as a guideline, rewriting should normally function up to this value).
8. Data retention characteristic when rewriting is performed within the specification range, including the minimum value.

26.9 Usage Note

General Notice during Design for Printed Circuit Board: Measures for radiation noise caused by the transient current in this LSI should be taken into consideration. The examples of the measures are shown below.

- To use a multilayer printed circuit board which includes layers for Vcc and GND.
- To mount by-pass capacitors (approximately 0.1 µF) between the Vcc and GND (Vss) pins, and the $PLLV_{CC}$ and PLLGND pins, of this LSI.

Appendix

A. I/O Port States in Each Processing State

Port Name Pin Name	MCU Operating Mode	Power-on Reset	Manual Reset	Hardware Standby Mode	Software Standby Mode	Bus Right Release State	Program Execution State or Sleep Mode
P17 to P14	4 to 7	T	keep	T	keep	keep	I/O port
P13/A23	7	T	keep	T	keep	keep	I/O port
P12/A22							
P11/A21							
Address output selected by AEn bit	4 to 6	T	keep	T	[OPE=0] T [OPE=1] keep	T	Address output
Port selection	4 to 6	T	keep	T	keep	keep	I/O port
P10/A20	7	T	keep	T	keep	keep	I/O port
Address output selected by AEn bit	4 and 5	L	keep	T	[OPE=0] T [OPE=1] keep	T	Address output
	6	T					
Port selection	4 to 6	T*	keep	T	keep	keep	I/O port
Port 3	4 to 7	T	keep	T	keep	keep	I/O port
Port 4	4 to 7	T	T	T	T	T	Input port
P74	4 to 7	T	keep	T	keep	keep	I/O port
P73/$\overline{CS7}$	7	T	keep	T	keep	keep	I/O port
P72/$\overline{CS6}$	4 to 6	T	keep	T	[DDR•OPE=0] T [DDR•OPE=1] H	T	[DDR=0] Input port [DDR=1] $\overline{CS7}$ to $\overline{CS4}$
P71/$\overline{CS5}$							
P70/$\overline{CS4}$							
Port 9	4 to 7	T	T	T	[DAOEn=1] keep [DAOEn=0] T	keep	Input port
Port A	7	T	keep	T	keep	keep	I/O port
Address output selected by AEn bit	4 and 5	L	keep	T	[OPE=0] T [OPE=1] keep	T	Address output
	6	T					
Port selection	4 to 6	T*	keep	T	keep	keep	I/O port

Port Name / Pin Name		MCU Operating Mode	Power-on Reset	Manual Reset	Hardware Standby Mode	Software Standby Mode	Bus Right Release State	Program Execution State or Sleep Mode
Port B		7	T	keep	T	keep	keep	I/O port
	Address output selected by AEn bit	4 and 5	L	keep	T	[OPE=0] T [OPE=1] keep	T	Address output
		6	T					
	Port selection	4 to 6	T*	keep	T	keep	keep	I/O port
Port C		4 and 5	L	keep	T	[OPE=0] T [OPE=1] keep	T	Address output
		6	T	keep	T	[DDR•OPE=0] T [DDR•OPE=1] keep	T	[DDR=0] Input port [DDR=1] Address output
		7	T	keep	T	keep	keep	I/O port
Port D		4 to 6	T	T	T	T	T	Data bus
		7	T	keep	T	keep	keep	I/O port
Port E	8-bit bus	4 to 6	T	keep	T	keep	keep	I/O port
	16-bit bus	4 to 6	T	T	T	T	T	Data bus
		7	T	keep	T	keep	keep	I/O port
PF7/ϕ		4 to 6	Clock output	[DDR=0] Input port [DDR=1] 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	keep	T	[DDR=0] Input port [DDR=1] H	[DDR=0] Input port [DDR=1] Clock output	[DDR=0] Input port [DDR=1] Clock output
PF6/$\overline{AS}$ PF5/$\overline{RD}$ PF4/$\overline{HWR}$		4 to 6	H	H	T	[OPE=0] T [OPE=1] H	T	$\overline{AS}$, $\overline{RD}$, $\overline{HWR}$
		7	T	keep	T	keep	keep	I/O port

Port Name Pin Name	MCU Operating Mode	Power-on Reset	Manual Reset	Hardware Standby Mode	Software Standby Mode	Bus Right Release State	Program Execution State or Sleep Mode
PF3/$\overline{\text{LWR}}$	7	T	keep	T	keep	keep	I/O port
8-bit bus	4 to 6	(Mode 4)	keep	T	keep	keep	I/O port
16-bit bus	4 to 6	H	H	T	[OPE=0]	T	$\overline{\text{LWR}}$
		(Modes 5 ,6)			T		
		T			[OPE=1]		
					H		
PF2/$\overline{\text{WAIT}}$	4 to 6	T	keep	T	[WAITE=0]	[WAITE=0]	[WAITE=0]
					keep	keep	I/O port
					[WAITE=1]	[WAITE=1]	[WAITE=1]
					T	T	$\overline{\text{WAIT}}$
	7	T	keep	T	keep	keep	I/O port
PF1/$\overline{\text{BACK}}$	4 to 6	T	keep	T	[BRLE=0]	L	[BRLE=0]
					keep		I/O port
					[BRLE=1]		[BRLE=1]
					H		$\overline{\text{BACK}}$
	7	T	keep	T	keep	keep	I/O port
PF0/$\overline{\text{BREQ}}$	4 to 6	T	keep	T	[BRLE=0]	T	[BRLE=0]
					keep		I/O port
					[BRLE=1]		[BRLE=1]
					T		$\overline{\text{BREQ}}$
	7	T	keep	T	keep	keep	I/O port
PG4/$\overline{\text{CS0}}$	4 and 5	H	keep	T	[DDR•OPE=0]	T	[DDR=0]
					T		I/O port
					[DDR•OPE=1]		[DDR=1]
					H		$\overline{\text{CS0}}$
	6	T					(When sleep mode)H
	7	T	keep	T	keep	keep	I/O port
PG3/$\overline{\text{CS1}}$	4 to 6	T	keep	T	[DDR•OPE=0]	T	[DDR=0]
PG2/$\overline{\text{CS2}}$					T		I/O port
PG1/$\overline{\text{CS3}}$					[DDR•OPE=1]		[DDR=1]
					H		$\overline{\text{CS1}}$ to $\overline{\text{CS3}}$
	7	T	keep	T	keep	keep	I/O port
PG0	4 to 7	T	keep	T	keep	keep	I/O port

Legend:

H: High level

L: Low level

T: High impedance

keep: Input port level is high impedance, and output port level is retained.

DDR: Data direction register

OPE: Output port enable

WAITE: Wait port enable

BRLE: Bus release enable

Note: * L (address input) in mode 4 or 5

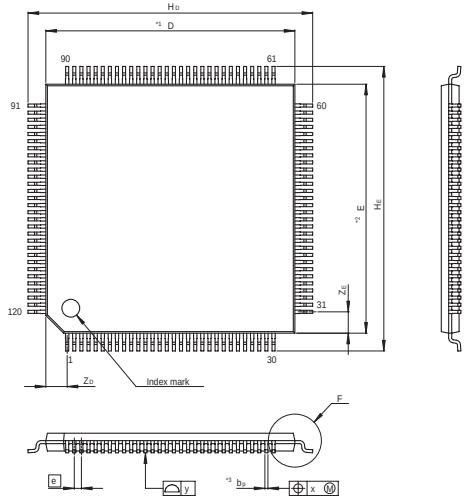
B. Product Model Lineup

Product Class		Part No.	Marking	Package (code)
H8S/2215	Flash memory Version	HD64F2215	HD64F2215TE	120 pin TQFP (TFP-120, TFP-120V)
			HD64F2215BR	112 pin P-LFBGA (BP-112, BP-112V)
		HD64F2215U	HD64F2215UTE	120 pin TQFP (TEP-120, TFP-120V)
			HD64F2215UBR	112 pin P-LFBGA (BP-112, BP-112V)
	Masked ROM Version	HD6432215A	HD6432215A(***)TE	120 pin TQFP (TFP-120, TFP-120V)
			HD6432215A(***)BR	112 pin P-LFBGA (BP-112, BP-112V)
		HD6432215B	HD6432215B(***)TE	120 pin TQFP (TFP-120, TFP-120V)
			HD6432215B(***)BR	112 pin P-LFBGA (BP-112, BP-112V)
		HD6432215C	HD6432215C(***)TE	120 pin TQFP (TFP-120, TFP-120V)
			HD6432215C(***)BR	112 pin P-LFBGA (BP-112, BP-112V)
H8S/2215R	Flash memory Version	HD64F2215R	HD64F2215RTE	120 pin TQFP (TFP-120, TFP-120V)
			HD64F2215RBR	112 pin P-LFBGA (BP-112, BP-112V)
		HD64F2215RU	HD64F2215RUTE	120 pin TQFP (TEP-120, TFP-120V)
			HD64F2215RUBR	112 pin P-LFBGA (BP-112, BP-112V)
H8S/2215T	Flash memory Version	HD64F2215T	HD64F2215TTE	120 pin TQFP (TFP-120, TFP-120V)
			HD64F2215TBR	112 pin P-LFBGA (BP-112, BP-112V)
		HD64F2215TU	HD64F2215TUTE	120 pin TQFP (TEP-120, TFP-120V)
			HD64F2215TUBR	112 pin P-LFBGA (BP-112, BP-112V)

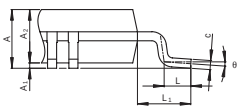
Legend: ***: ROM code

C. Package Dimensions

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-TQFP120-14x14-0.40	PTQP0120LA-A	TFP-120/TFP-120V	0.5g



Terminal cross section



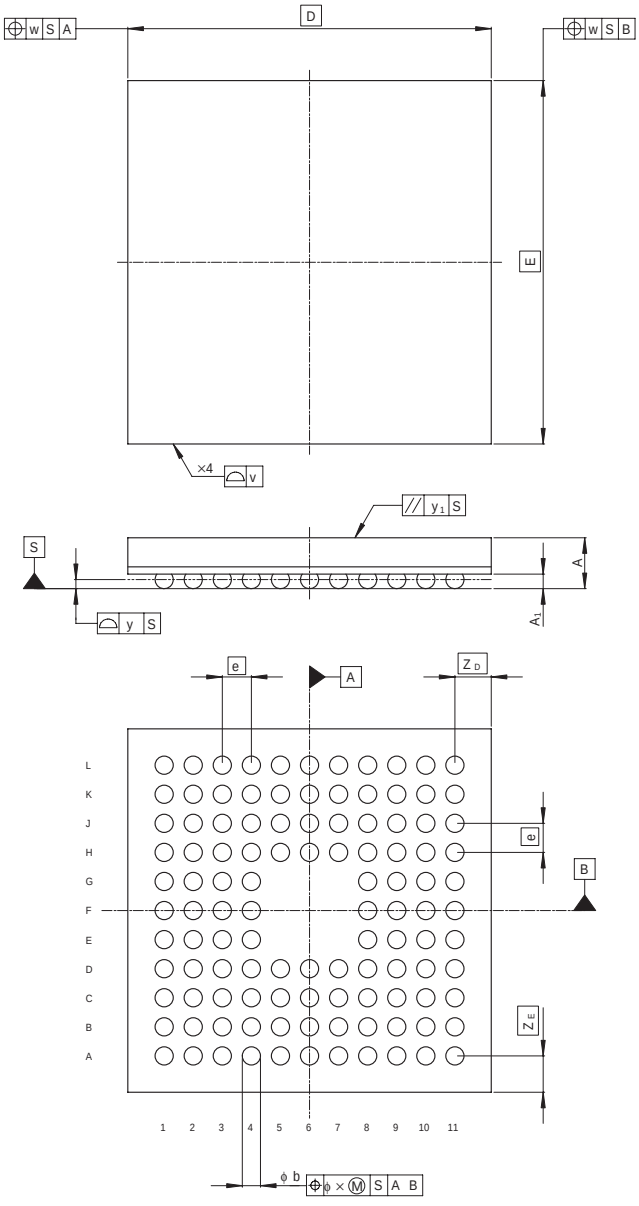
Detail F

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	—	14	—
E	—	14	—
A_2	—	1.00	—
H_D	15.8	16.0	16.2
H_E	15.8	16.0	16.2
A	—	—	1.20
A_1	0.00	0.10	0.20
b_p	0.12	0.17	0.22
b_1	—	0.15	—
c	0.12	0.17	0.22
c_1	—	0.15	—
θ	0°	—	8°
e	—	0.4	—
x	—	—	0.07
y	—	—	0.10
Z_D	—	1.20	—
Z_E	—	1.20	—
L	0.4	0.5	0.6
L_1	—	1.0	—

Figure C.1 TFP-120, TFP-120V Package Dimension

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LFBGA112-10x10-0.80	PLBG0112GA-A	BP-112/BP-112V	0.3g



Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	—	10.00	—
E	—	10.00	—
v	—	—	0.15
w	—	—	0.20
A	—	—	1.40
A_1	0.35	0.40	0.45
e	—	0.80	—
b	0.45	0.50	0.55
x	—	—	0.08
y	—	—	0.10
y_1	—	—	0.2
S_D	—	—	—
S_E	—	—	—
Z_D	—	1.00	—
Z_E	—	1.00	—

Figure C.2 BP-112, BP-112V Package Dimension

Index

Renesas 16-Bit Single-Chip Microcomputer
Hardware Manual
H8S/2215 Group

Publication Date: 1st Edition, April 2001
Rev.7.00, March 27, 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

H8S/2215 Group
Hardware Manual

Renesas Technology Corp.
2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan