To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

RENESAS

Notice

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

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

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

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

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

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

User's Manual

32

H8SX/1657 Group

Hardware Manual

Renesas 32-Bit CISC Microcomputer H8SX Family / H8SX/1600 Series

H8SX/1657C R5F61657C
H8SX/1656C R5F61656C

Renesas Electronics
www.renesas.com

Rev.2.00 2007.06

RENESAS

vicinity of LSI, an associated shoot-through current flows internally, and malfunct
occur due to the false recognition of the pin state as an input signal. Unused pins
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 regist
   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, th
   of pins are not guaranteed from the moment when power is supplied until the res
   process is completed.

   In a similar way, the states of pins in a product that is reset by an on-chip power-
   function are not guaranteed from the moment when power is supplied until the po
   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 functio
   not access these addresses; the correct operation of LSI is not guaranteed if they
   accessed.

4. Clock Signals

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

   — When the clock signal is generated with an external resonator (or from an externa
   oscillator) during a reset, ensure that the reset line is only released after full stabi
   the clock signal. Moreover, when switching to a clock signal produced with an ext
   resonator (or by an external oscillator) while program execution is in progress, wa
   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 numbe
   that the change will not lead to problems.

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

When designing an application system that includes this LSI, take all points to note
account. Points to note are given in their contexts and at the final part of each sect
in the section giving usage notes.

The list of revisions is a summary of major points of revision or addition for earlier
It does not cover all revised items. For details on the revised points, see the actual
in the manual.

The following documents have been prepared for the H8SX/1657 Group. Before usi
the documents, please visit our web site to verify that you have the most up-to-date a
version of the document.

Document Type	Contents	Document Title	Docu
Data Sheet	Overview of hardware and electrical characteristics	—	—
Hardware Manual	Hardware specifications (pin assignments, memory maps, peripheral specifications, electrical characteristics, and timing charts) and descriptions of operation	H8SX/1657 Group Hardware Manual	This r
Software Manual	Detailed descriptions of the CPU and instruction set	H8SX Family Software Manual	REJ0
Application Note	Examples of applications and sample programs	The latest versions are available f web site.	
Renesas Technical Update	Preliminary report on the specifications of a product, document, etc.		

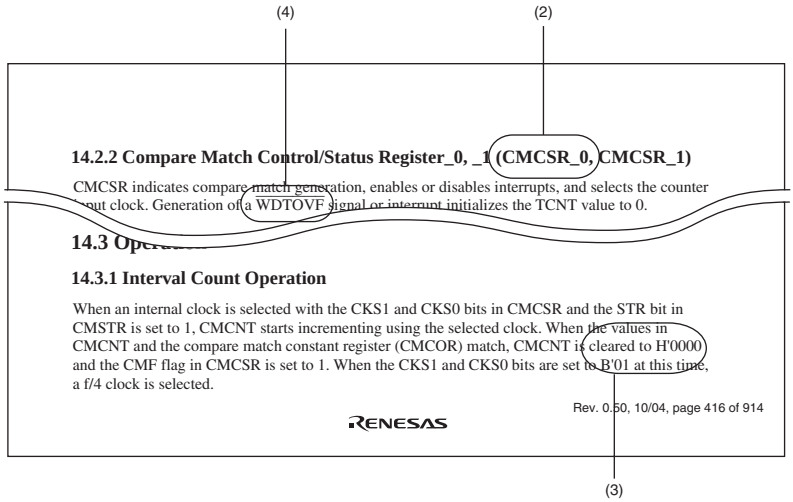
Binary numbers are given as B'nnnn (B' may be omitted if the number is obviously binary), hexadecimal numbers are given as H'nnnn or 0xnnnn, and decimal numbers are given as nnnn.

[Examples] Binary: B'11 or 11
Hexadecimal: H'EFA0 or 0xEFA0
Decimal: 1234

(4) Notation for active-low

An overbar on the name indicates that a signal or pin is active-low.

[Example] $\overline{\text{WDTOVF}}$



Note: The bit names and sentences in the above figure are examples and have nothing to do with the contents of this manual.

Bit	Bit Name	Initial Value	R/W	Description
15	–	0	R	Reserved
14	–	0	R	These bits are always read as 0.
13 to 11	ASID2 to ASID0	All 0	R/W	Address Identifier These bits enable or disable the pin function.
10	–	0	R	Reserved This bit is always read as 0.
9	–	1	R	Reserved This bit is always read as 1.
	–	0		

Note: The bit names and sentences in the above figure are examples, and have nothing to do with the conter manual.

(1) Bit
Indicates the bit number or numbers.
In the case of a 32-bit register, the bits are arranged in order from 31 to 0. In the case of a 16-bit register, the bits are arranged in order from 15 to 0.

(2) Bit name
Indicates the name of the bit or bit field.
When the number of bits has to be clearly indicated in the field, appropriate notation is included (e.g., ASID[3:0]).
A reserved bit is indicated by "–".
Certain kinds of bits, such as those of timer counters, are not assigned bit names. In such cases, the entry under Bit Name is blank.

(3) Initial value
Indicates the value of each bit immediately after a power-on reset, i.e., the initial value.
0: The initial value is 0
1: The initial value is 1
–: The initial value is undefined

(4) R/W
For each bit and bit field, this entry indicates whether the bit or field is readable or writable or both writing to and reading from the bit or field are impossible.
The notation is as follows:

R/W: The bit or field is readable and writable.
R/(W): The bit or field is readable and writable.
However, writing is only performed to flag clearing.
R: The bit or field is readable.
"R" is indicated for all reserved bits. When writing to the register, write the value under Initial Value in the bit chart to reserved bits or fields.
W: The bit or field is writable.

(5) Description
Describes the function of the bit or field and specifies the values for writing.

SCI	Serial communication interface
TMR	8-bit timer
TPU	16-bit timer pulse unit
WDT	Watchdog timer

- Abbreviations other than those listed above

Abbreviation	Description
ACIA	Asynchronous communication interface adapter
bps	Bits per second
CRC	Cyclic redundancy check
DMA	Direct memory access
DMAC	Direct memory access controller
GSM	Global System for Mobile Communications
Hi-Z	High impedance
IEBus	Inter Equipment Bus (IEBus is a trademark of NEC Electronics Corpo
I/O	Input/output
IrDA	Infrared Data Association
LSB	Least significant bit
MSB	Most significant bit
NC	No connection
PLL	Phase-locked loop
PWM	Pulse width modulation
SFR	Special function register
SIM	Subscriber Identity Module
UART	Universal asynchronous receiver/transmitter
VCO	Voltage-controlled oscillator

speed data transfer, and a bus-state controller, which enables direct connection to differe
of memory. The LSI of the Group also includes serial communication interfaces, A/D ar
converters, and a multi-function timer that makes motor control easy. Together, the mod
realize low-cost configurations for end systems. The power consumption of these modul
down dynamically by an on-chip power-management function. The on-chip ROM is a fl
memory (F-ZTAT™) with a capacity of 768 Kbytes (H8SX/1657C) or 512 Kbytes (H8S

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

1.1.1 Applications

Examples of the applications of this LSI include PC peripheral equipment, optical storag
office automation equipment, and industrial equipment.

Notes: The following additions and changes have been made in the switch from the H8
to the H8SX/1657C.

A sleep exception handling function has been added to the H8SX/1657C.

	RAM	• RAM capacity: 24 Kbytes
CPU	CPU	• 32-bit high-speed H8SX CPU (CISC type) Upward-compatibility with H8/300, H8/300H, and H8S C object level • Sixteen 16-bit general registers • Eleven addressing modes • 4-Gbyte address space Program: 4 Gbytes available Data: 4 Gbytes available • 87 basic instructions, classifiable as bit arithmetic and l instructions, multiply and divide instructions, bit manipu instructions, multiply-and-accumulate instructions, and • Minimum instruction execution time: 20.0 ns (for an AD instruction while system clock Iϕ = 50 MHz and V_{CC} = 3.0 to 3.6 V) • On-chip multiplier (16 $\times$ 16 $\rightarrow$ 32 bits) • Supports multiply-and-accumulate instructions (16 $\times$ 16 + 32 $\rightarrow$ 32 bits)
	Operating mode	• Advanced mode

		driving the MD2 pin high) Mode 5: On-chip ROM disabled external extended mode, (selected by driving the MD1 pin low and driving and MD0 pins high) Mode 6: On-chip ROM enabled external extended mode (selected by driving the MD0 pin low and driving and MD1 pins high) Mode 7: Single chip mode (external extension possible) (selected by driving the MD2, MD1, and MD0 pir • Power-down state (transition to the power-down state the SLEEP instruction)
Interrupt (source)	Interrupt controller (INTC)	• Thirteen external interrupt pins (NMI, and $\overline{\text{IRQ11}}$ to $\overline{\text{IR}}$ • 64 internal interrupt sources • Two interrupt control modes (specified by the interrup register) • Eight priority orders specifiable (by setting the interrup register) • Independent vector addresses

	transfer controller (DTC)	sources) • Activated by interrupt sources (chain transfer enabled) • Three transfer modes (normal transfer, repeat transfer, block transfer) • Short-address mode or full-address mode selectable
External bus extension	Bus controller (BSC)	• 16-Mbyte external address space
		• The external address space can be divided into eight a each of which is independently controllable — Chip-select signals ($\overline{CS0}$ to $\overline{CA7}$) can be output — Access in two or three states can be selected for ea — Program wait cycles can be inserted — The period of $\overline{CS}$ assertion can be extended — Idle cycles can be inserted • Bus arbitration function (arbitrates bus mastership amo internal CPU, DMAC, and DTC, and external bus mast
		Bus formats • External memory interfaces (for the connection of ROM ROM, SRAM, and byte control SRAM) • Address/data bus format: Support for both separate an multiplexed buses (8-bit access or 16-bit access)
		• Endian conversion function for connecting devices in lit endian format

		— Modules in the external space run in synchronizati the external bus clock (Bϕ): 8 to 35 MHz • Includes a PLL frequency multiplication circuit and fre divider, so the operating frequency is selectable • Four power-down modes: Sleep mode, all-module-clo mode, software standby mode, and hardware standby
A/D converter	A/D converter (ADC)	• 10-bit resolution × eight input channels • Sample and hold function included • Conversion time: 7.4 μs per channel (with peripheral r clock (Pϕ) at 35-MHz operation) • Two operating modes: single mode and scan mode • Three ways to start A/D conversion: software, timer (T trigger, and external trigger
D/A converter	D/A converter (DAC)	• 8-bit resolution × two output channels • Output voltage: 0 V to Vref, maximum conversion time (with 20-pF load)

		counters (TCNT), simultaneous clearing by compare m input capture possible, simultaneous input/output for re possible by counter synchronous operation, and up to 1 PWM output possible by combination with synchronous operation • Buffered operation, cascaded operation (32 bits × two channels), and phase counting mode (two-phase encod input) settable for each channel • Input capture function supported • Output compare function (by the output of compare ma waveform) supported
	Program-mable pulse generator (PPG)	• 16-bit pulse output • Four output groups, non-overlapping mode, and inverte can be set • Selectable output trigger signals; the PPG can operate conjunction with the data transfer controller (DTC) and controller (DMAC)
Watchdog timer	Watchdog timer (WDT)	• 8 bits × one channel (selectable from eight counter inpu • Switchable between watchdog timer mode and interval mode
Serial interface	Serial communi-cation interface (SCI)	• Four channels (choice of asynchronous or clocked syn serial communication mode) • Full-duplex communication capability • Select the desired bit rate and LSB-first or MSB-first tra
Smart card/ SIM		• The SCI module supports a smart card (SIM) interface.

Operating frequency/ Power supply voltage	• Operating frequency: 8 to 35 MHz • Power supply voltage: Vcc = 3.0 to 3.6 V, AVcc = 3.0 • Supply current: — 35 mA (typ.) (Vcc = 3.3 V, AVcc = 3.3 V, Iϕ = Pϕ = 35 MHz)
Operating peripheral temperature (°C)	• –20 to +75°C (regular specifications) • –40 to +85°C (wide-range specifications)

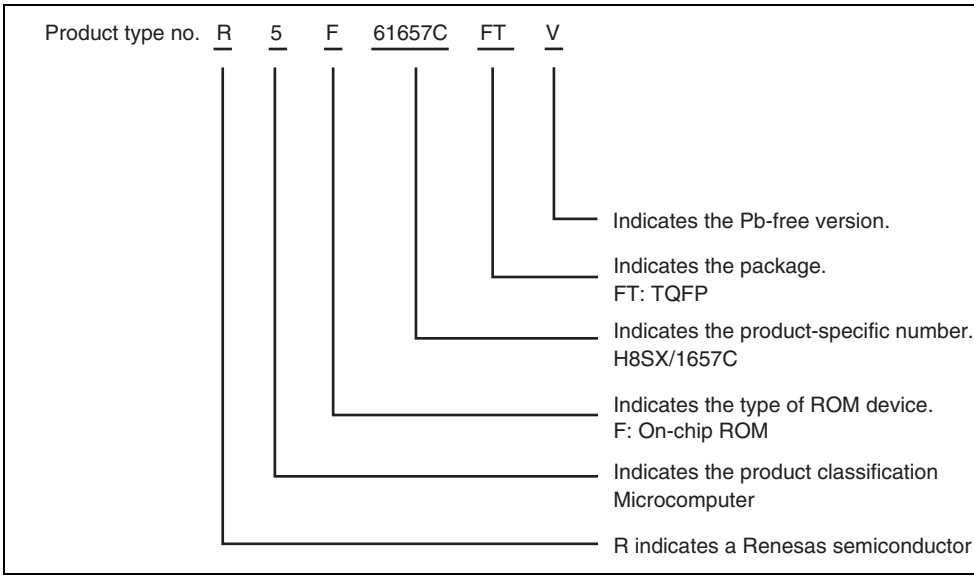


Figure 1.1 How to Read the Product Name Code

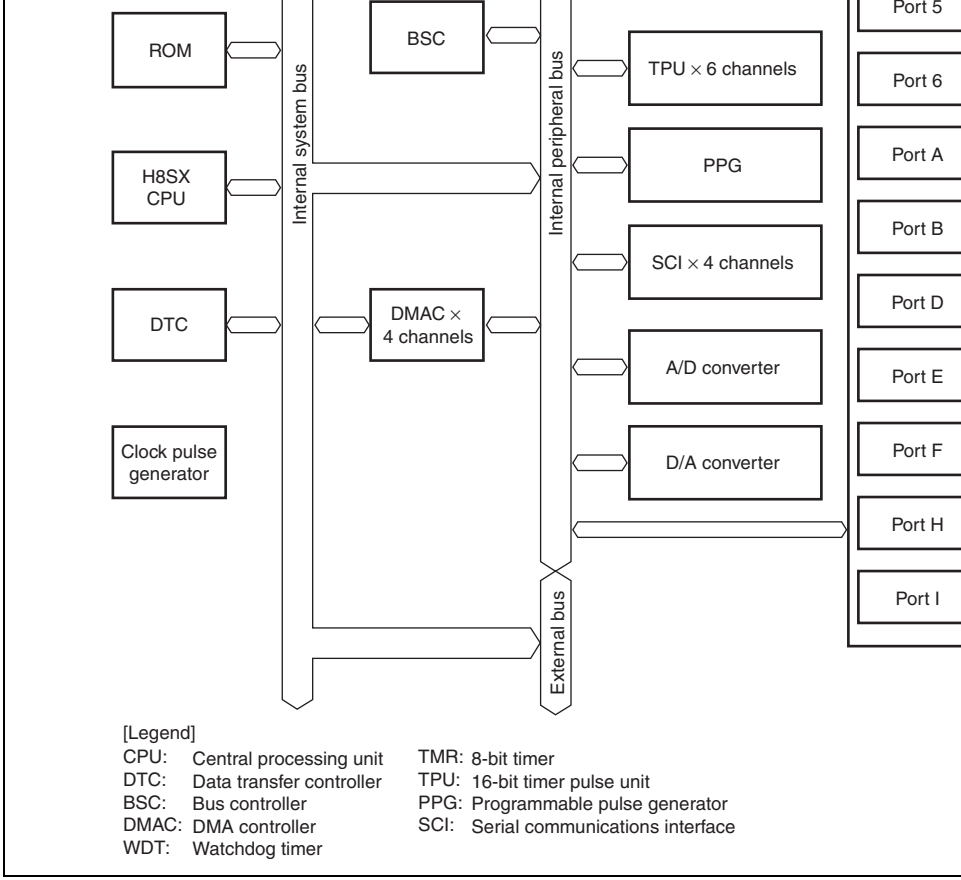


Figure 1.2 Internal Block Diagram

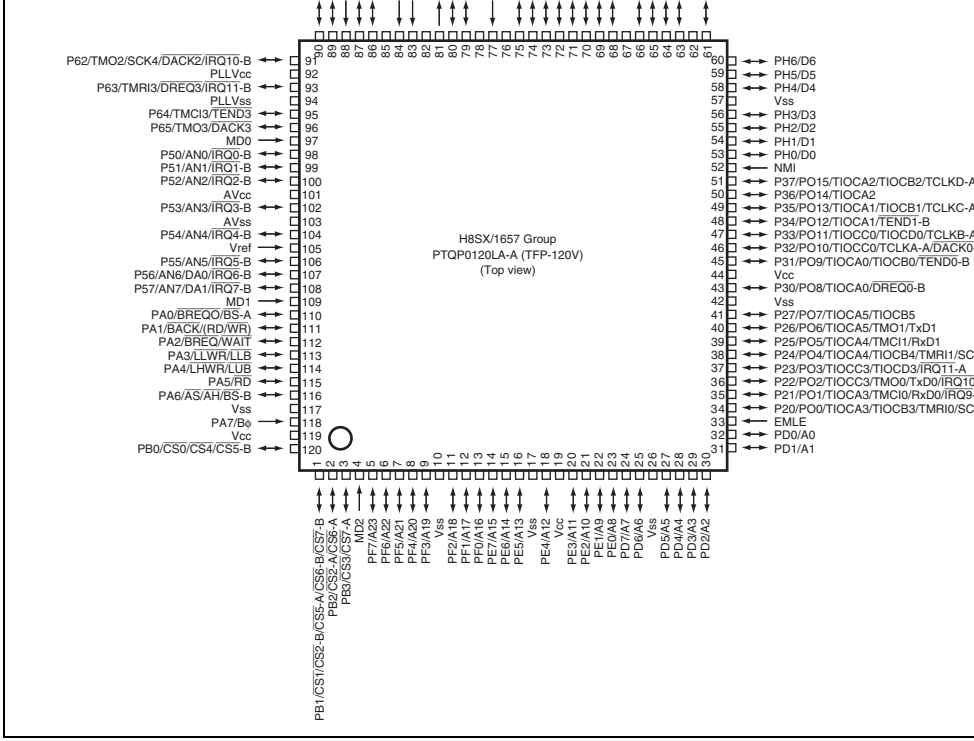


Figure 1.3 Pin Assignments

	$PLLV_{CC}$	Input	Power supply pin for the PLL circuit.
	$PLLV_{SS}$	Input	Ground pin for the PLL circuit.
Clock	XTAL	Input	Pins for a crystal resonator. An external clock sig input through the EXTAL pin. For an example of t connection, see section 19, Clock Pulse Generat
	EXTAL	Input	
	Bϕ	Output	Outputs the system clock for external devices.
Operating mode control	MD2 to MD0	Input	Pins for setting the operating mode. The signal le these pins must not be changed during operation
System control	$\overline{RES}$	Input	Reset pin. This LSI enters the reset state when th goes low.
	$\overline{STBY}$	Input	This LSI enters hardware standby mode when thi goes low.
	EMLE	Input	Input pin for the on-chip emulator enable signal. T level should normally be fixed low.
Address bus	A23 to A0	Output	Output pins for the address bits.
Data bus	D15 to D0	Input/ output	Input and output for the bidirectional data bus. Th also output addresses when accessing an addres multiplexed I/O interface space.
Bus control	$\overline{BREQ}$	Input	External bus-master modules assert this signal to the bus.
	$\overline{BREQO}$	Output	Internal bus-master modules assert this signal to access to the external space via the bus in the ex released state.

RD/$\overline{\text{WR}}$	Output	Indicates the direction (input or output) of the data
$\overline{\text{LHWR}}$	Output	Strobe signal which indicates that the higher-order (D15 to D8) is valid in access to the basic bus inte space.
$\overline{\text{LLWR}}$	Output	Strobe signal which indicates that the lower-order to D0) is valid in access to the basic bus interface
$\overline{\text{LUB}}$	Output	Strobe signal which indicates that the higher-order (D15 to D8) is valid in access to the byte control S interface space.
$\overline{\text{LLB}}$	Output	Strobe signal which indicates that the lower-order to D0) is valid in access to the byte control SRAM space.
$\overline{\text{CS0}}$ $\overline{\text{CS1}}$ $\overline{\text{CS2}}$-A/$\overline{\text{CS2}}$-B $\overline{\text{CS3}}$ $\overline{\text{CS4}}$ $\overline{\text{CS5}}$-A/$\overline{\text{CS5}}$-B $\overline{\text{CS6}}$-A/$\overline{\text{CS6}}$-B $\overline{\text{CS7}}$-A/$\overline{\text{CS7}}$-B	Output	Select signals for areas 7 to 0.
$\overline{\text{WAIT}}$	Input	Requests wait cycles in access to the external spa

	$\overline{\text{IRQ2}}$-A/$\overline{\text{IRQ2}}$-B $\overline{\text{IRQ1}}$-A/$\overline{\text{IRQ1}}$-B $\overline{\text{IRQ0}}$-A/$\overline{\text{IRQ0}}$-B		
DMA controller (DMAC)	$\overline{\text{DREQ0}}$-A/$\overline{\text{DREQ0}}$-B $\overline{\text{DREQ1}}$-A/$\overline{\text{DREQ1}}$-B $\overline{\text{DREQ2}}$ $\overline{\text{DREQ3}}$	Input	Requests DMAC activation.
	$\overline{\text{DACK0}}$-A/$\overline{\text{DACK0}}$-B $\overline{\text{DACK1}}$-A/$\overline{\text{DACK1}}$-B $\overline{\text{DACK2}}$ $\overline{\text{DACK3}}$	Output	DMAC single address-transfer acknowledge sign
	$\overline{\text{TEND0}}$-A/$\overline{\text{TEND0}}$-B $\overline{\text{TEND1}}$-A/$\overline{\text{TEND1}}$-B $\overline{\text{TEND2}}$ $\overline{\text{TEND3}}$	Output	Indicates end of data transfer by the DMAC.
16-bit timer pulse unit (TPU)	TCLKA-A/TCLKA-B TCLKB-A/TCLKB-B TCLKC-A/TCLKC-B TCLKD-A/TCLKD-B	Input	Input pins for the external clock signals.
	TIOCA0 TIOCB0 TIOCC0 TIOCD0	Input/ output	Signals for TGRA_0 to TGRD_0. These pins are input capture inputs, output compare outputs, or outputs.
	TIOCA1 TIOCB1	Input/ output	Signals for TGRA_1 and TGRB_1. These pins ar input capture inputs, output compare outputs, or outputs.

	TIOCA5 TIOCB5	Input/ output	Signals for TGRA_5 and TGRB_5. These pins are input capture inputs, output compare outputs, or P outputs.
Programmable pulse generator (PPG)	PO15 to PO0	Output	Output pins for the pulse signals.
8-bit timer (TMR)	TMO0 to TMO3	Output	Output pins for the compare match signals.
	TMCI0 to TMCI3	Input	Input pins for the external clock signals that drive f counters.
	TMRI0 to TMRI3	Input	Input pins for the counter-reset signals.
Watchdog timer (WDT)	$\overline{\text{WDTOVF}}$	Output	Output pin for the counter-overflow signal in watch mode.
Serial communication interface (SCI)	TxD0 to TxD4	Output	Output pins for data transmission.
	RxD0 to RxD4	Input	Input pins for data reception.
	SCK0 to SCK4	Input/ output	Input/output pins for clock signals.

	Vref	Input	Reference power supply pin for the A/D and D/A When the A/D and D/A converters are not in use, this pin to the system power supply.
I/O ports	P17 to P10	Input/ output	8-bit input/output pins.
	P27 to P20	Input/ output	8-bit input/output pins.
	P37 to P30	Input/ output	8-bit input/output pins.
	P57 to P50	Input	8-bit input/output pins.
	P65 to P60	Input/ output	6-bit input/output pins.
	PA7	Input	Input-only pin.
	PA6 to PA0	Input/ output	7-bit input/output pins.
	PB3 to PB0	Input/ output	4-bit input/output pins.
	PD7 to PD0	Input/ output	8-bit input/output pins.
	PE7 to PE0	Input/ output	8-bit input/output pins.
	PF7 to PF0	Input/ output	8-bit input/output pins.
	PH7 to PH0	Input/ output	8-bit input/output pins.
	PI7 to PI0	Input/ output	8-bit input/output pins.

- Upward-compatible with H8/300, H8/300H, and H8S CPUs
  - Can execute H8/300, H8/300H, and H8S/2000 object programs
- Sixteen 16-bit general registers
  - Also usable as sixteen 8-bit registers or eight 32-bit registers
- 87 basic instructions
  - 8/16/32-bit arithmetic and logic instructions
  - Multiply and divide instructions
  - Bit field transfer instructions
  - Powerful bit-manipulation instructions
  - Bit condition branch instructions
  - Multiply-and-accumulate instruction
- Eleven addressing modes
  - Register direct [Rn]
  - Register indirect [@ERn]
  - Register indirect with displacement [@(d:2,ERn), @(d:16,ERn), or @(d:32,ERn
  - Index register indirect with displacement [@(d:16,RnL.B), @(d:32,RnL.B), @(d:16,Rn.W), @(d:32,Rn.W), @(d:16,ERn.L), or @(d:32,ERn.L)]
  - Register indirect with pre-/post-increment or pre-/post-decrement [@+ERn, @–E @ERn+, or @ERn–]
  - Absolute address [@aa:8, @aa:16, @aa:24, or @aa:32]
  - Immediate [#xx:3, #xx:4, #xx:8, #xx:16, or #xx:32]
  - Program-counter relative [@(d:8,PC) or @(d:16,PC)]
  - Program-counter relative with index register [@(RnL.B,PC), @(Rn.W,PC), or @(ERn.L,PC)]
  - Memory indirect [@@aa:8]
  - Extended memory indirect [@@vec:7]

— 8 × 8-bit register-register multiply: 1 state
— 16 ÷ 8-bit register-register divide: 10 states
— 16 × 16-bit register-register multiply: 1 state
— 32 ÷ 16-bit register-register divide: 18 states
— 32 × 32-bit register-register multiply: 5 states
— 32 ÷ 32-bit register-register divide: 18 states

- Four CPU operating modes
  — Normal mode
  — Middle mode
  — Advanced mode
  — Maximum mode
- Power-down modes
  — Transition is made by execution of SLEEP instruction
  — Choice of CPU operating clocks

Notes: 1. Advanced mode is only supported as the CPU operating mode of the H8SX/16
Group. Normal, middle, and maximum modes are not supported.
2. The multiplier and divider are supported by the H8SX/1657 Group.

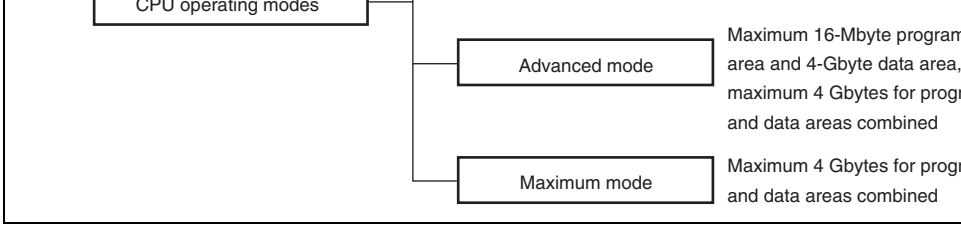


Figure 2.1 CPU Operating Modes

2.2.1 Normal Mode

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

Note: Normal mode is not supported in this LSI.

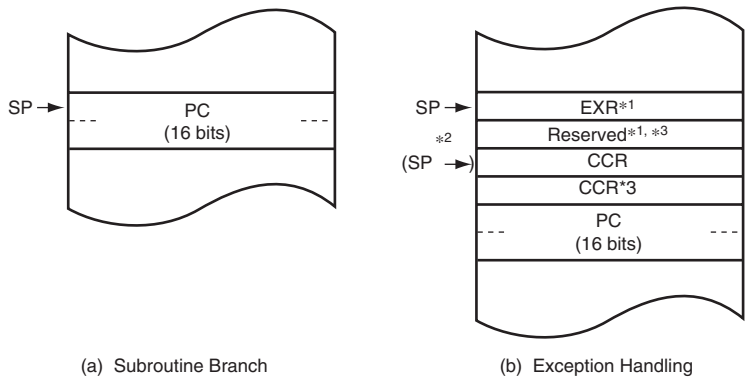
- Address Space
  The maximum address space of 64 Kbytes can be accessed.
- Extended Registers (En)
  The extended registers (E0 to E7) can be used as 16-bit registers, or as the upper 16-
  segments of 32-bit registers. When the extended register En is used as a 16-bit regist
  contain any value, even when the corresponding general register Rn is used as an ad
  register. (If the general register Rn is referenced in the register indirect addressing m
  pre-/post-increment or pre-/post-decrement and a carry or borrow occurs, however, t
  the corresponding extended register En will be affected.)
- Instruction Set
  All instructions and addressing modes can be used. Only the lower 16 bits of effectiv
  addresses (EA) are valid.

Figure 2.2 Exception Vector Table (Normal Mode)

The memory indirect (@@aa:8) and extended memory indirect (@@vec:7) addressin are used in the JMP and JSR instructions. An 8-bit absolute address included in the in code specifies a memory location. Execution branches to the contents of the memory

- Stack Structure

  The stack structure of PC at a subroutine branch and that of PC and CCR at an except handling are shown in figure 2.3. The PC contents are saved or restored in 16-bit unit



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

Figure 2.3 Stack Structure (Normal Mode)

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

- Instruction Set

  All instructions and addressing modes can be used. Only the lower 16 bits of effectiv
  addresses (EA) are valid and the upper eight bits are sign-extended.

- Exception Vector Table and Memory Indirect Branch Addresses

  In middle mode, the top area starting at H'000000 is allocated to the exception vecto
  One branch address is stored per 32 bits. The upper eight bits are ignored and the lov
  are stored. The structure of the exception vector table is shown in figure 2.4.

  The memory indirect (@@aa:8) and extended memory indirect (@@vec:7) addressi
  are used in the JMP and JSR instructions. An 8-bit absolute address included in the i
  code specifies a memory location. Execution branches to the contents of the memory

  In middle mode, an operand is a 32-bit (longword) operand, providing a 32-bit branc
  The upper eight bits are reserved and assumed to be H'00.

- Stack Structure

  The stack structure of PC at a subroutine branch and that of PC and CCR at an excep
  handling are shown in figure 2.5. The PC contents are saved or restored in 24-bit uni

- 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 ve
  table. One branch address is stored per 32 bits. The upper eight bits are ignored and th
  24 bits are stored. The structure of the exception vector table is shown in figure 2.4.

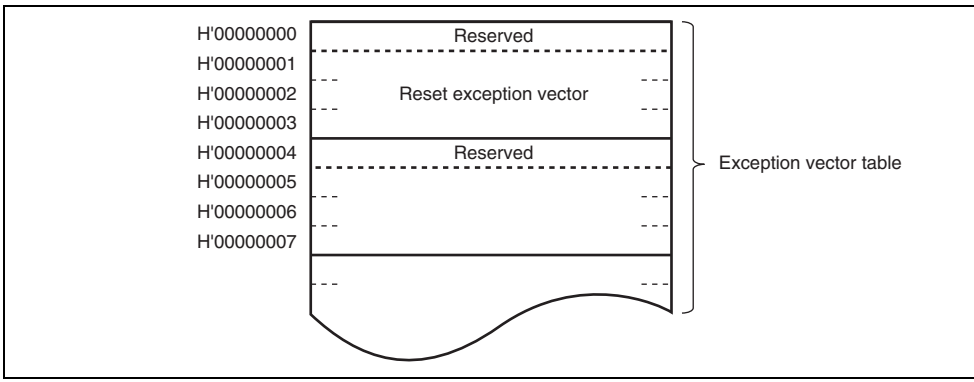


Figure 2.4 Exception Vector Table (Middle and Advanced Modes)

The memory indirect (@@aa:8) and extended memory indirect (@@vec:7) addressin
are used in the JMP and JSR instructions. An 8-bit absolute address included in the in
code specifies a memory location. Execution branches to the contents of the memory

In advanced mode, an operand is a 32-bit (longword) operand, providing a 32-bit bran
address. The upper eight bits are reserved and assumed to be H'00.

(a) Subroutine Branch (b) Exception Handling

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

Figure 2.5 Stack Structure (Middle and Advanced Modes)

2.2.4 Maximum Mode

The program area is extended to 4 Gbytes as compared with that in advanced mode.

- Address Space
  The maximum address space of 4 Gbytes can be linearly accessed.
- Extended Registers (En)
  The extended registers (E0 to E7) can be used as 16-bit registers or as the upper 16-b
  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 maximum mode, the top area starting at H'00000000 is allocated to the exception
  table. One branch address is stored per 32 bits. The structure of the exception vector
  shown in figure 2.6.

Figure 2.6 Exception Vector Table (Maximum Modes)

The memory indirect (@@aa:8) and extended memory indirect (@@vec:7) addressin
are used in the JMP and JSR instructions. An 8-bit absolute address included in the in
code specifies a memory location. Execution branches to the contents of the memory

In maximum mode, an operand is a 32-bit (longword) operand, providing a 32-bit bra
address.

- Stack Structure

  The stack structure of PC at a subroutine branch and that of PC and CCR at an except
handling are shown in figure 2.7. The PC contents are saved or restored in 32-bit unit
EXR contents are saved or restored regardless of whether or not EXR is in use.

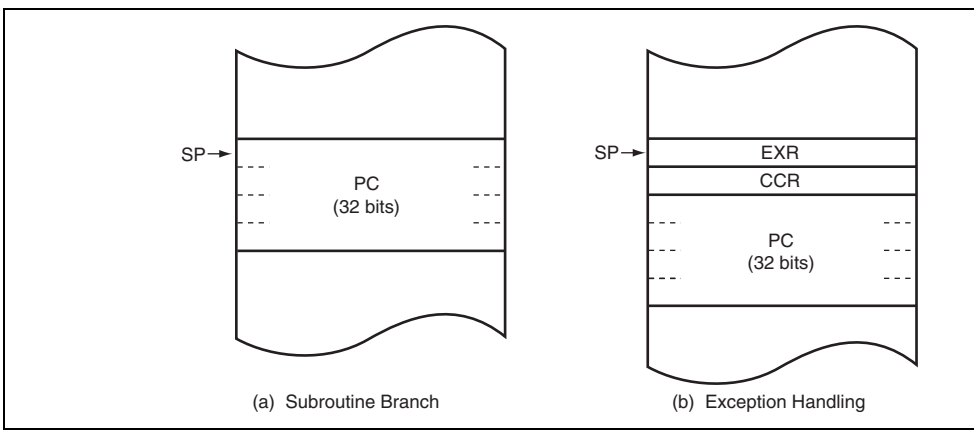


Figure 2.7 Stack Structure (Maximum Mode)

Figure 2.8 shows a memory map of the H8SX CPU. The address space differs dependin
CPU operating mode.

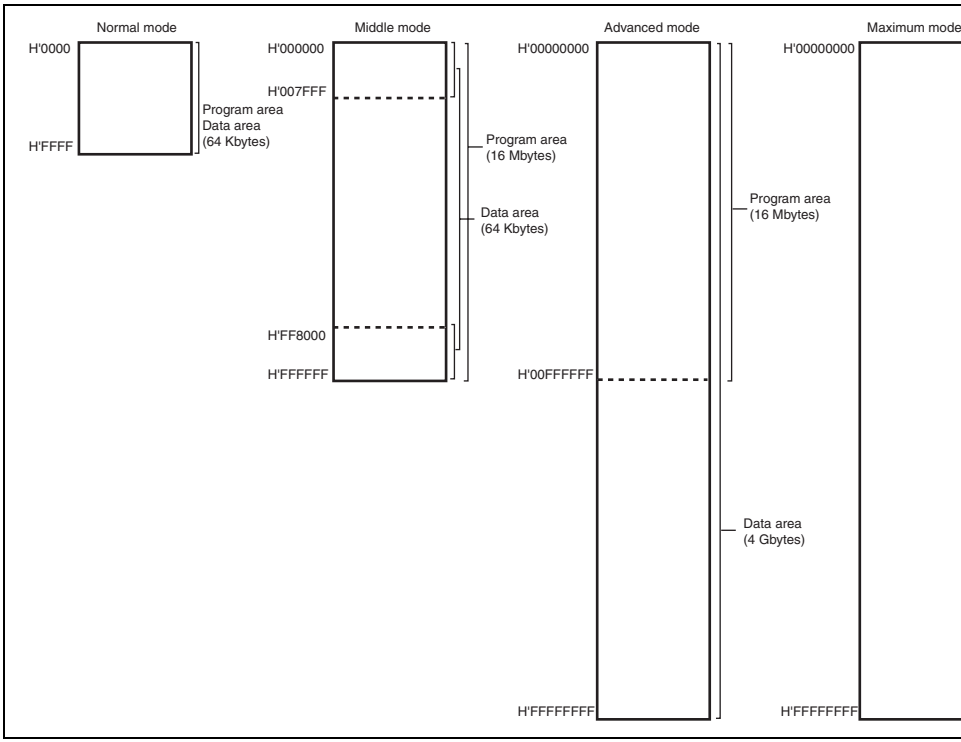


Figure 2.8 Memory Map

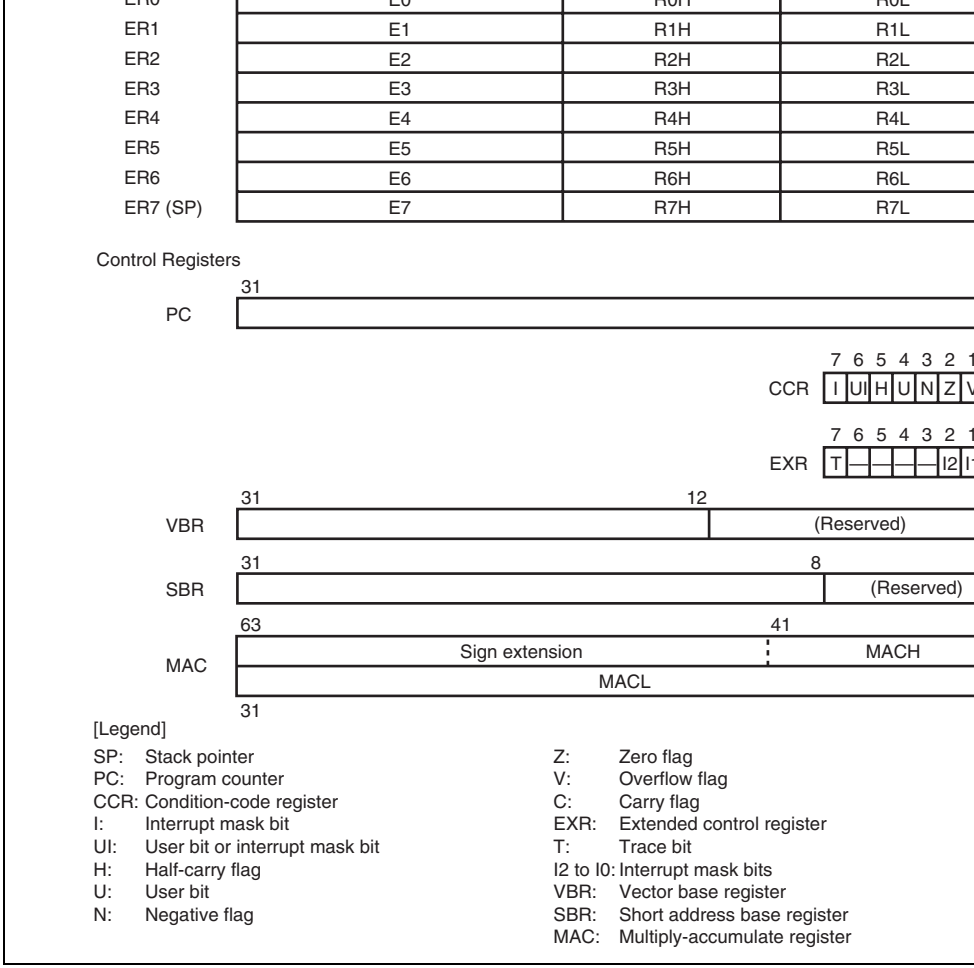


Figure 2.9 CPU Registers

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

When the general registers are used as 8-bit registers, the R registers are divided into 8-b
registers designated by the letters RH (R0H to R7H) and RL (R0L to R7L). These regist
functionally equivalent, providing a maximum sixteen 8-bit registers.

The general registers ER (ER0 to ER7), R (R0 to R7), and RL (R0L to R7L) are also use
registers. The size in the operand field determines which register is selected.

The usage of each register can be selected independently.

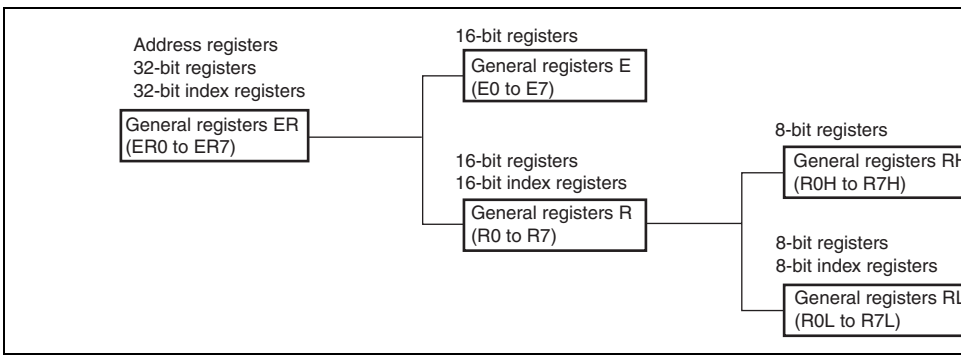


Figure 2.10 Usage of General Registers

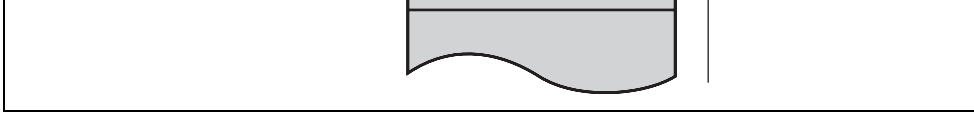


Figure 2.11 Stack

2.5.2 Program Counter (PC)

PC is a 32-bit counter that indicates the address of the next instruction the CPU will exec
length of all CPU instructions is 16 bits (one word) or a multiple of 16 bits, so the least si
bit is ignored. (When the instruction code is fetched, the least significant bit is regarded a

Bit	Bit Name	Value	R/W	Description
7	I	1	R/W	Interrupt Mask Bit Masks interrupts when set to 1. This bit is the start of an exception handling.
6	UI	Undefined	R/W	User Bit or Interrupt Mask Bit Can be written to and read from by softwa the LDC, STC, ANDC, ORC, and XORC instructions. This bit can also be used as a interrupt mask bit.
5	H	Undefined	R/W	Half-Carry Flag When the ADD.B, ADDX.B, SUB.B, SUBX CMP.B, or NEG.B instruction is executed, set to 1 if there is a carry or borrow at bit 3 cleared to 0 otherwise. When the ADD.W, CMP.W, or NEG.W instruction is executed is set to 1 if there is a carry or borrow at bi cleared to 0 otherwise. When the ADD.L, CMP.L, or NEG.L instruction is executed, set to 1 if there is a carry or borrow at bit 2 cleared to 0 otherwise.
4	U	Undefined	R/W	User Bit Can be written to and read from by softwa the LDC, STC, ANDC, ORC, and XORC instructions.
3	N	Undefined	R/W	Negative Flag Stores the value of the most significant bit as sign bit) of data.

	otherwise. A carry has the following types: • Carry from the result of addition • Borrow from the result of subtraction • Carry from the result of shift or rotation The carry flag is also used as a bit accumu bit manipulation instructions.

2.5.4 Extended Control Register (EXR)

EXR is an 8-bit register that contains the trace bit (T) and three interrupt mask bits (I2 to

Operations can be performed on the EXR bits by the LDC, STC, ANDC, ORC, and XOR instructions.

For details, see section 4, Exception Handling.

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 i generated each time an instruction is execu When this bit is cleared to 0, instructions ar executed in sequence.
6 to 3	—	All 1	R/W	Reserved These bits are always read as 1.

reset and a CPU address error (extended memory indirect is also out of the target). The i
value is H'00000000. The VBR contents are changed with the LDC and STC instruction

2.5.6 Short Address Base Register (SBR)

SBR is a 32-bit register in which the upper 24 bits are valid. The lower eight bits are rea
8-bit absolute address addressing mode (@aa:8), this register is used as the upper addres
initial value is H'FFFFFF00. The SBR contents are changed with the LDC and STC inst

2.5.7 Multiply-Accumulate Register (MAC)

MAC is a 64-bit register that stores the results of multiply-and-accumulate operations. It
of two 32-bit registers denoted MACH and MACL. The lower 10 bits of MACH are val
upper bits are sign extended. The MAC contents are changed with the MAC, CLRMAC
and STMAC instructions.

2.5.8 Initial Values of CPU Registers

Reset exception handling loads the start address from the vector table into the PC, clears
in EXR to 0, and sets the I bits in CCR and EXR to 1. The general registers, MAC, and t
bits in CCR are not initialized. In particular, the initial value of the stack pointer (ER7) i
undefined. The SP should therefore be initialized using an MOV.L instruction executed
immediately after a reset.

Figure 2.12 shows the data formats in general registers.

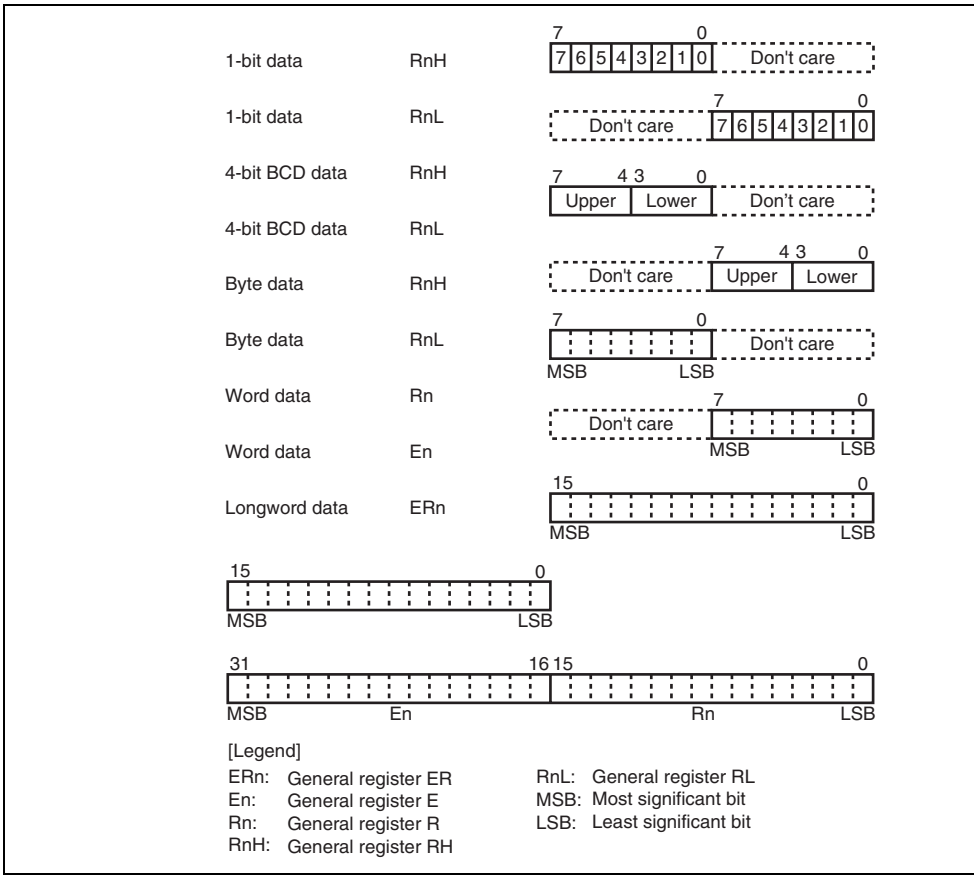


Figure 2.12 General Register Data Formats

the stack manipulation, branch table manipulation, block transfer instructions, and MAC instruction should be located to even addresses.

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

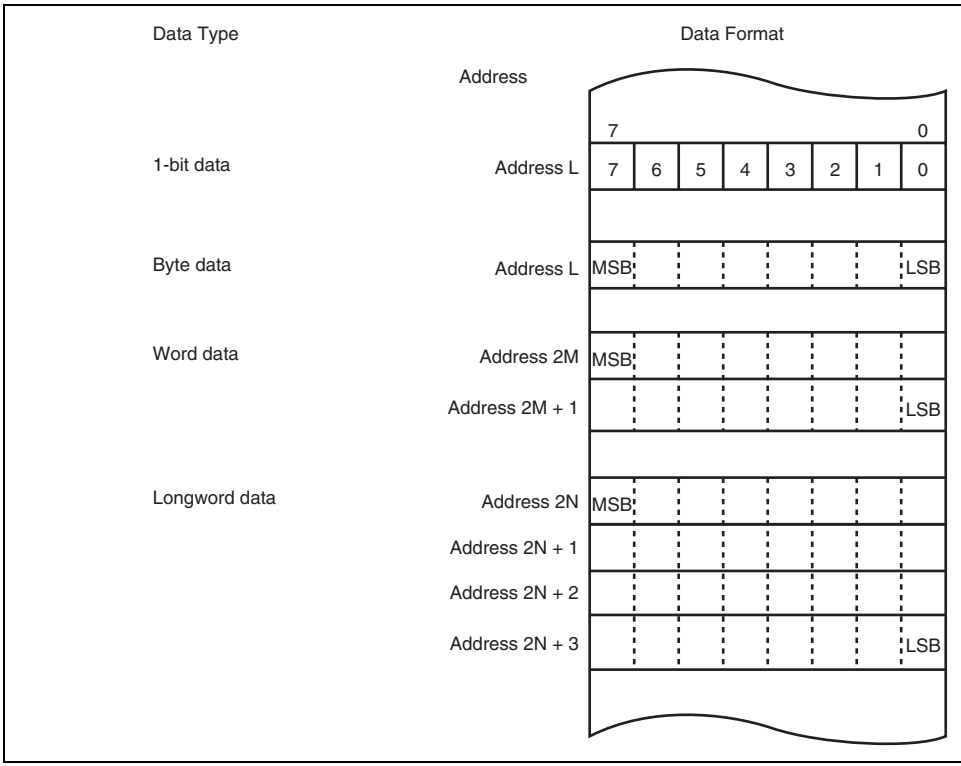


Figure 2.13 Memory Data Formats

	POP, PUSH*[1]	W/L
	LDM, STM	L
	MOVA	B/W*[2]
Block transfer	EEPMOV	B
	MOVMD	B/W/L
	MOVSD	B
Arithmetic operations	ADD, ADDX, SUB, SUBX, CMP, NEG, INC, DEC	B/W/L
	DAA, DAS	B
	ADDS, SUBS	L
	MULXU, DIVXU, MULXS, DIVXS	B/W
	MULU, DIVU, MULS, DIVS	W/L
	MULU/U, MULS/U	L
	EXTU, EXTS	W/L
	TAS	B
	MAC	—
	LDMAC, STMAC	—
	CLRMAC	—
Logic operations	AND, OR, XOR, NOT	B/W/L
Shift	SHLL, SHLR, SHAL, SHAR, ROTL, ROTR, ROTXL, ROTXR	B/W/L
Bit manipulation	BSET, BCLR, BNOT, BTST, BAND, BIAND, BOR, BIOR, BXOR, BIXOR, BLD, BILD, BST, BIST	B
	BSET/EQ, BSET/NE, BCLR/EQ, BCLR/NE, BSTZ, BISTZ	B
	BFLD, BFST	B

[Legend]

B: Byte size

W: Word size

L: Longword size

Notes: 1. POP.W Rn and PUSH.W Rn are identical to MOV.W @SP+, Rn and MOV.W @−SP.

POP.L ERn and PUSH.L ERn are identical to MOV.L @SP+, ERn and MOV.L @−SP.

2. Size of data to be added with a displacement
3. Size of data to specify a branch condition
4. Bcc is the generic designation of a conditional branch instruction.
5. Size of general register to be restored
6. Not available in this LSI.

Data transfer	MOV	B/W/L	S	SD	SD	SD	SD	SD		SD
		B		S/D					S/D	
	MOVFPE, MOVTPE*[12]	B		S/D						S/D*
	POP, PUSH	W/L		S/D				S/D*[2]		
	LDM, STM	L		S/D				S/D*[2]		
	MOVA*[4]	B/W		S	S	S	S	S		S
Block transfer	EEPMOV	B								
	MOVMD	B/W/L								
	MOVSD	B								
Arithmetic operations	ADD, CMP	B	S	D	D	D	D	D	D	D
		B		S	D	D	D	D	D	D
		B		D	S	S	S	S	S	S
		B			SD	SD	SD	SD		SD
		W/L	S	SD	SD	SD	SD	SD		SD
	SUB	B	S		D	D	D	D	D	D
		B		S	D	D	D	D	D	D
		B		D	S	S	S	S	S	S
		B			SD	SD	SD	SD		SD
		W/L	S	SD	SD	SD	SD	SD		SD
	ADDX, SUBX	B/W/L	S	SD						
		B/W/L	S		SD					
		B/W/L	S					SD*[5]		
	INC, DEC	B/W/L		D						

	DIVXS									
	MULS, DIVS	W/L	S:4	SD						
	NEG	B		D	D	D	D	D	D	D
		W/L		D	D	D	D	D		D
	EXTU, EXTS	W/L		D	D	D	D	D		D
	TAS	B			D					
	MAC	—								
	CLRMAC	—								
	LDMAC	—		S						
	STMAC	—		D						
Logic operations	AND, OR, XOR	B		S	D	D	D	D	D	D
		B		D	S	S	S	S	S	S
		B			SD	SD	SD	SD		SD
		W/L	S	SD	SD	SD	SD	SD		SD
	NOT	B		D	D	D	D	D	D	D
		W/L		D	D	D	D	D		D
Shift	SHLL, SHLR	B		D	D	D	D	D	D	D
		B/W/L*[6]		D	D	D	D	D		D
		B/W/L*[7]		D						
	SHAL, SHAR ROTL, ROTR ROTXL, ROTXR	B		D	D	D	D	D	D	D
		W/L		D	D	D	D	D		D

	BILD, BST, BIST, BSTZ, BISTZ									
	BFLD	B		D	S				S	S
	BFST	B		S	D				D	D
Branch	BRA/BS, BRA/BC*[8]	B			S				S	S
	BSR/BS, BSR/BC*[8]	B			S				S	S
System control	LDC (CCR, EXR)	B/W*[9]	S	S	S	S		S*[10]		S
	LDC (VBR, SBR)	L		S						
	STC (CCR, EXR)	B/W*[9]		D	D	D		D*[11]		D
	STC (VBR, SBR)	L		D						
	ANDC, ORC, XORC	B	S							
	SLEEP	—								
	NOP	—								

[Legend]

d: d:16 or d:32

S: Can be specified as a source operand.

D: Can be specified as a destination operand.

SD: Can be specified as either a source or destination operand or both.

S/D: Can be specified as either a source or destination operand.

S:4: 4-bit immediate data can be specified as a source operand.

Notes: 1. Only @aa:16 is available.

2. @ERn+ as a source operand and @−ERn as a destination operand

3. Specified by ER5 as a source address and ER6 as a destination address for d transfer.

			Addressing Mode						
Classification	Instruction	Size	@ERn	@(d,PC)	@(RnL.B/Rn.W/ERn.L, PC)	@aa:24	@ aa:32	@@ aa:8	@@
Branch	BRA/BS, BRA/BC	—		O					
	BSR/BS, BSR/BC	—		O					
	Bcc	—		O					
	BRA	—		O	O				
	BRA/S	—		O*					
	JMP	—	O			O	O	O	O
	BSR	—		O					
	JSR	—	O			O	O	O	O
	RTS, RTS/L	—							
System control	TRAPA	—							
	RTE, RTE/L	—							

[Legend]

d: d:8 or d:16

Note: * Only @(d:8, PC) is available.

ERn	General register (32-bit register)
(EAd)	Destination operand
(EAs)	Source operand
EXR	Extended control register
CCR	Condition-code register
VBR	Vector base register
SBR	Short address base 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
~	Logical 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 regis
to R7, E0 to E7), and 32-bit registers (ER0 to ER7).

		Saves general register contents on the stack.
LDM	L	@SP+ → Rn (register list) Restores the data from the stack to multiple general registers. T or four general registers which have serial register numbers car specified.
STM	L	Rn (register list) → @−SP Saves the contents of multiple general registers on the stack. T or four general registers which have serial register numbers car specified.
MOVA	B/W	EA → Rd Zero-extends and shifts the contents of a specified general regi memory data and adds them with a displacement. The result is general register.

Note: Not available in this LSI.

MOVMD.W	W	Transfers a data block. Transfers word data which begins at a memory location specifiec to a memory location specified by ER6. The number of word data transferred is specified by R4.
MOVMD.L	L	Transfers a data block. Transfers longword data which begins at a memory location spec ER5 to a memory location specified by ER6. The number of long data to be transferred is specified by R4.
MOVSD.B	B	Transfers a data block with zero data detection. Transfers byte data which begins at a memory location specified to a memory location specified by ER6. The number of byte data transferred is specified by R4. When zero data is detected during the transfer stops and execution branches to a specified address

INC	B/W/L	Rd ± 1 → Rd, Rd ± 2 → Rd
DEC		Increments or decrements a general register by 1 or 2. (Byte op 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 genera
DAA	B	Rd (decimal adjust) → Rd
DAS		Decimal-adjusts an addition or subtraction result in a general re referring to the CCR to produce 2-digit 4-bit BCD data.
MULXU	B/W	Rd × Rs → Rd
		Performs unsigned multiplication on data in two general registe bits × 8 bits → 16 bits, or 16 bits × 16 bits → 32 bits.
MULU	W/L	Rd × Rs → Rd
		Performs unsigned multiplication on data in two general registe bits × 8 bits → 16 bits, or 16 bits × 16 bits → 32 bits.
MULU/U	L	Rd × Rs → Rd
		Performs unsigned multiplication on data in two general registe × 32 bits → upper 32 bits).
MULXS	B/W	Rd × Rs → Rd
		Performs signed multiplication on data in two general registers: bits × 8 bits → 16 bits, or 16 bits × 16 bits → 32 bits.
MULS	W/L	Rd × Rs → Rd
		Performs signed multiplication on data in two general registers: bits × 16 bits → 16 bits, or 32 bits × 32 bits → 32 bits.
MULS/U	L	Rd × Rs → Rd
		Performs signed multiplication on data in two general registers 32 bits → upper 32 bits).
DIVXU	B/W	Rd ÷ Rs → Rd
		Performs unsigned division on data in two general registers: eit ÷ 8 bits → 8-bit quotient and 8-bit remainder, or 32 bits ÷ 16 bits quotient and 16-bit remainder.

		Compares data between immediate data, general registers, and and stores the result in CCR.
NEG	B/W/L	0 – (EAd) → (EAd) Takes the two's complement (arithmetic complement) of data in a register or the contents of a memory location.
EXTU	W/L	(EAd) (zero extension) → (EAd) Performs zero-extension on the lower 8 or 16 bits of data in a ge register or memory to word or longword size. The lower 8 bits to word or longword, or the lower 16 bits to long be zero-extended.
EXTS	W/L	(EAd) (sign extension) → (EAd) Performs sign-extension on the lower 8 or 16 bits of data in a ge register or memory to word or longword size. The lower 8 bits to word or longword, or the lower 16 bits to long be sign-extended.
TAS	B	@ERd – 0, 1 → (<bit 7> of @EAd) Tests memory contents, and sets the most significant bit (bit 7) to
MAC	—	(EAs) × (EAd) + MAC → MAC Performs signed multiplication on memory contents and adds the MAC.
CLRMAC	—	0 → MAC Clears MAC to zero.
LDMAC	—	Rs → MAC Loads data from a general register to MAC.
STMAC	—	MAC → Rd Stores data from MAC to a general register.

		Performs a logical exclusive OR operation on data between imm data, general registers, and memory.
NOT	B/W/L	~ (EAd) → (EAd) Takes the one's complement of the contents of a general regist memory location.

Table 2.8 Shift Operation Instructions

Instruction	Size	Function
SHLL SHLR	B/W/L	(EAd) (shift) → (EAd) Performs a logical shift on the contents of a general register or a location. The contents of a general register or a memory location can be 1, 2, 4, 8, or 16 bits. The contents of a general register can be s any bits. In this case, the number of bits is specified by 5-bit imm data or the lower 5 bits of the contents of a general register.
SHAL SHAR	B/W/L	(EAd) (shift) → (EAd) Performs an arithmetic shift on the contents of a general registe memory location. 1-bit or 2-bit shift is possible.
ROTL ROTR	B/W/L	(EAd) (rotate) → (EAd) Rotates the contents of a general register or a memory location 1-bit or 2-bit rotation is possible.
ROTXL ROTXR	B/W/L	(EAd) (rotate) → (EAd) Rotates the contents of a general register or a memory location carry bit. 1-bit or 2-bit rotation is possible.

BCLR	B	0 → (<bit-No.> of <EAd>) Clears a specified bit in the contents of a general register or a me location to 0. The bit number is specified by 3-bit immediate data lower three bits of a general register.
BCLR/cc	B	if cc, 0 → (<bit-No.> of <EAd>) If the specified condition is satisfied, this instruction clears a spec in a memory location to 0. The bit number can be specified by 3- immediate data, or by the lower three bits of a general register. T status can be specified as a condition.
BNOT	B	~ (<bit-No.> of <EAd>) → (<bit-No.> of <EAd>) Inverts a specified bit in the contents of a general register or a m location. The bit number is specified by 3-bit immediate data or t three bits of a general register.
BTST	B	~ (<bit-No.> of <EAd>) → Z Tests a specified bit in the contents of a general register or a me location and sets or clears the Z flag accordingly. The bit number specified by 3-bit immediate data or the lower three bits of a gen register.
BAND	B	C ∧ (<bit-No.> of <EAd>) → C ANDs the carry flag with a specified bit in the contents of a gener register or a memory location and stores the result in the carry fl bit number is specified by 3-bit immediate data.
BIAND	B	C ∧ [~ (<bit-No.> of <EAd>)] → C ANDs the carry flag with the inverse of a specified bit in the conte general register or a memory location and stores the result in the flag. The bit number is specified by 3-bit immediate data.

		Exclusive-ORs the carry flag with a specified bit in the contents of a general register or a memory location and stores the result in the carry flag. The bit number is specified by 3-bit immediate data.
BIXOR	B	C ⊕ [~ (<bit-No.> of <EAd>)] → C Exclusive-ORs the carry flag with the inverse of a specified bit in the contents of a general register or a memory location 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 the contents of a general register or a memory location to the carry flag. The bit number is specified by 3-bit immediate data.
BILD	B	~ (<bit-No.> of <EAd>) → C Transfers the inverse of a specified bit in the contents of a general register or a memory location 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 the contents of a general register or a memory location. The bit number is specified by 3-bit immediate data.
BSTZ	B	Z → (<bit-No.> of <EAd>) Transfers the zero flag value to a specified bit in the contents of a memory location. The bit number is specified by 3-bit immediate data.
BIST	B	~ C → (<bit-No.> of <EAd>) Transfers the inverse of the carry flag value to a specified bit in the contents of a general register or a memory location. The bit number is specified by 3-bit immediate data.

field in memory location contents.

Table 2.10 Branch Instructions

Instruction	Size	Function
BRA/BS BRA/BC	B	Tests a specified bit in memory location contents. If the specified condition is satisfied, execution branches to a specified address.
BSR/BS BSR/BC	B	Tests a specified bit in memory location contents. If the specified condition is satisfied, execution branches to a subroutine at a spe address.
Bcc	—	Branches to a specified address if the specified condition is satis
BRA/S	—	Branches unconditionally to a specified address after executing t instruction. The next instruction should be a 1-word instruction ex the block transfer and branch instructions.
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.
RTS/L	—	Returns from a subroutine, restoring data from the stack to multip general registers.

		location to CCR or EXR. Although CCR and EXR are 8-bit registers, word-size transfers performed between them and memory. The upper 8 bits are val
	L	Rs → VBR, Rs → SBR Transfers the general register contents to VBR or SBR.
STC	B/W	CCR → (EAd), EXR → (EAd) Transfers the contents of CCR or EXR to a general register or r Although CCR and EXR are 8-bit registers, word-size transfers performed between them and memory. The upper 8 bits are val
	L	VBR → Rd, SBR → Rd Transfers the contents of VBR or SBR to a general register.
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 immedia
NOP	—	PC + 2 → PC Only increments the program counter.

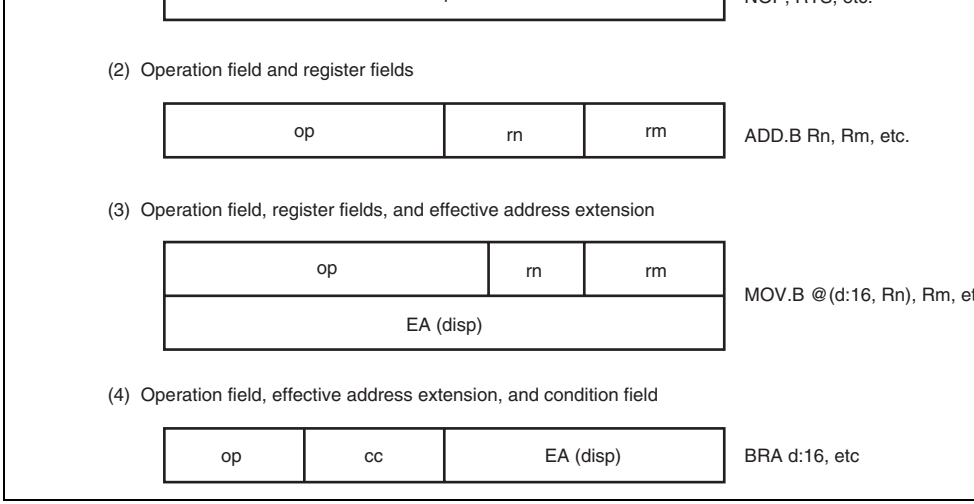


Figure 2.14 Instruction Formats

- Operation Field

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

  Specifies a general register. Address registers are specified by 3 bits, data registers by 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 branch condition of Bcc instructions.

No.	Addressing Mode	Symbol
1	Register direct	Rn
2	Register indirect	@ERn
3	Register indirect with displacement	@(d:2,ERn)/@(d:16,ERn)/@(d:32,E
4	Index register indirect with displacement	@(d:16, RnL.B)/@(d:16,Rn.W)/@(d: @(d:32, RnL.B)/@(d:32,Rn.W)/@(d:
5	Register indirect with post-increment	@ERn+
	Register indirect with pre-decrement	@−ERn
	Register indirect with pre-increment	@+ERn
	Register indirect with post-decrement	@ERn−
6	Absolute address	@aa:8/@aa:16/@aa:24/@aa:32
7	Immediate	#xx:3/#xx:4/#xx:8/#xx:16/#xx:32
8	Program-counter relative	@(d:8,PC)/@(d:16,PC)
9	Program-counter relative with index register	@(RnL.B,PC)/@(Rn.W,PC)/@(ERn.
10	Memory indirect	@@aa:8
11	Extended memory indirect	@@vec:7

address register (ERn). ERn is specified by the register field of the instruction code.

In advanced mode, if this addressing mode is used in a branch instruction, the lower 24 b
valid and the upper 8 bits are all assumed to be 0 (H'00).

2.8.3 Register Indirect with Displacement—@(d:2, ERn), @(d:16, ERn), or @(d
ERn)

The operand value is the contents of a memory location which is pointed to by the sum o
contents of an address register (ERn) and a 16- or 32-bit displacement. ERn is specified b
register field of the instruction code. The displacement is included in the instruction code
16-bit displacement is sign-extended when added to ERn.

This addressing mode has a short format (@(d:2, ERn)). The short format can be used wh
displacement is 1, 2, or 3 and the operand is byte data, when the displacement is 2, 4, or 6
operand is word data, or when the displacement is 4, 8, or 12 and the operand is longwor

2.8.4 Index Register Indirect with Displacement—@(d:16,RnL.B), @(d:32,RnL.
@(d:16,Rn.W), @(d:32,Rn.W), @(d:16,ERn.L), or @(d:32,ERn.L)

The operand value is the contents of a memory location which is pointed to by the sum o
following operation result and a 16- or 32-bit displacement: a specified bits of the conten
address register (RnL, Rn, ERn) specified by the register field in the instruction code are
extended to 32-bit data and multiplied by 1, 2, or 4. The displacement is included in the i
code and the 16-bit displacement is sign-extended when added to ERn. If the operand is b
ERn is multiplied by 1. If the operand is word or longword data, ERn is multiplied by 2 o
respectively.

(2) Register indirect with pre-decrement—@–ERn

The operand value is the contents of a memory location which is pointed to by the follo
operation result: the value 1, 2, or 4 is subtracted from the contents of an address registe
ERn is specified by the register field of the instruction code. After that, the operand valu
in the address register. The value subtracted is 1 for byte access, 2 for word access, or 4
longword access.

(3) Register indirect with pre-increment—@+ERn

The operand value is the contents of a memory location which is pointed to by the follo
operation result: the value 1, 2, or 4 is added to the contents of an address register (ERn)
specified by the register field of the instruction code. After that, the operand value is sto
address register. The value added is 1 for byte access, 2 for word access, or 4 for longwo

(4) Register indirect with post-decrement—@ERn–

The operand value is the contents of a memory location which is pointed to by the conte
address register (ERn). ERn is specified by the register field in the instruction code. Aft
memory location is accessed, 1, 2, or 4 is subtracted from the address register contents a
subtraction result is stored in the address register. The value subtracted is 1 for byte acce
word access, or 4 for longword access.

In the case of (1) to (4) above, if the contents of a general register which is also used as
register is written to memory using this addressing mode, data to be written is the conte
general register after calculating an effective address. If the same general register is spec
instruction and two effective addresses are calculated, the contents of the general registe
first calculation of an effective address is used in the second calculation of an effective a

After execution, ER0 is H'00001002.

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

The operand value is the contents of a memory location which is pointed to by an absolut
included in the instruction code.

There are 8-bit (@aa:8), 16-bit (@aa:16), 24-bit (@aa:24), and 32-bit (@aa:32) absolute
addresses.

To access the data area, the absolute address of 8 bits (@aa:8), 16 bits (@aa:16), or 32 bi
(@aa:32) is used. For an 8-bit absolute address, the upper 24 bits are specified by SBR. F
bit absolute address, the upper 16 bits are sign-extended. A 32-bit absolute address can ac
entire address space.

To access the program area, the absolute address of 24 bits (@aa:24) or 32 bits (@aa:32)
For a 24-bit absolute address, the upper 8 bits are all assumed to be 0 (H'00).

Table 2.13 shows the accessible absolute address ranges.

	(@aa:24)	H'FFFFFF		
	32 bits (@aa:32)		H'00000000 to H'00FFFFFF	H'0000 H'FFF

manipulation instructions contain 3-bit immediate data in the instruction code, for specify number. The BFLD and BFST instructions contain 8-bit immediate data in the instruction for specifying a bit field. The TRAPA instruction contains 2-bit immediate data in the ins code, for specifying a vector address.

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

This mode is used in the Bcc and BSR instructions. The operand value is a 32-bit branch which is the sum of an 8- or 16-bit displacement in the instruction code and the 32-bit add the PC contents. The 8-bit or 16-bit displacement is sign-extended to 32 bits when added contents. The PC contents to which the displacement is added is the address of the first by next instruction, so the possible branching range is –126 to +128 bytes (–63 to +64 words –32766 to +32768 bytes (–16383 to +16384 words) from the branch instruction. The resu value should be an even number. In advanced mode, only the lower 24 bits of this branch are valid; the upper 8 bits are all assumed to be 0 (H'00).

2.8.9 Program-Counter Relative with Index Register—@(RnL.B, PC), @(Rn.W @(ERn.L, PC)

This mode is used in the Bcc and BSR instructions. The operand value is a 32-bit branch which is the sum of the following operation result and the 32-bit address of the PC conter contents of an address register specified by the register field in the instruction code (RnL, ERn) is zero-extended and multiplied by 2. The PC contents to which the displacement is the address of the first byte of the next instruction. In advanced mode, only the lower 24 b this branch address are valid; the upper 8 bits are all assumed to be 0 (H'00).

advanced mode, the first byte of the longword-size data is assumed to be all 0 (H'00).

Note that the top part of the address range is also used as the exception handling vector a
vector address of an exception handling other than a reset or a CPU address error can be
by VBR.

Figure 2.15 shows an example of specification of a branch address using this addressing

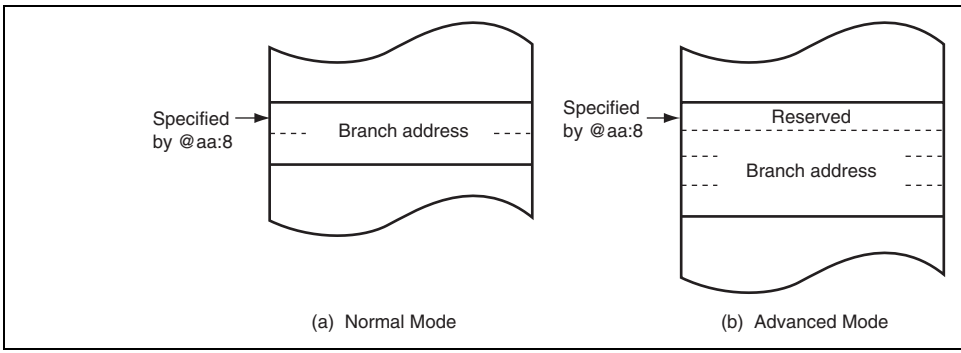


Figure 2.15 Branch Address Specification in Memory Indirect Mode

advanced mode, the first byte of the longword-size data is assumed to be all 0 (H'00).

2.8.12 Effective Address Calculation

Tables 2.14 and 2.15 show how effective addresses are calculated in each addressing mod
lower bits of the effective address are valid and the upper bits are ignored (zero extended
extended) according to the CPU operating mode.

The valid bits in middle mode are as follows:

- The lower 16 bits of the effective address are valid and the upper 16 bits are sign-exte
  the transfer and operation instructions.
- The lower 24 bits of the effective address are valid and the upper eight bits are zero-e
  for the branch instructions.

	Register indirect with 32-bit displacement op r disp	31 General register contents 0 disp	31
5	Index register indirect with 16-bit displacement op r disp	31 Zero extension 0 Contents of general register (RL, R, or ER) 1, 2, or 4 31 15 Sign extension disp 0	31
	Index register indirect with 32-bit displacement op r disp	31 Zero extension 0 Contents of general register (RL, R, or ER) 1, 2, or 4 31 disp 0	31
6	Register indirect with post-increment or post-decrement op r	31 General register contents 0 1, 2, or 4	31
	Register indirect with pre-increment or pre-decrement op r	31 General register contents 0 1, 2, or 4	31
7	8-bit absolute address op aa	31 SBR 7 aa 0	31
	16-bit absolute address op aa	31 15 Sign extension aa 0	31
	32-bit absolute address op aa	31 aa 0	31

3	Program-counter relative with index register op r	31 0 Zero extension Contents of general register (RL, R, or ER) 2 31 0 PC contents	31
4	24-bit absolute address op aa 32-bit absolute address op aa	Zero 31 extension 23 0 aa 31 0 aa	31 31
5	Memory indirect op aa	31 7 0 Zero extension aa 31 0 Memory contents	31
6	Extended memory indirect op vec	31 7 0 Zero extension 1 vec 2 or 4 31 0 31 0 Memory contents	31

2.8.13 MOVA Instruction

The MOVA instruction stores the effective address in a general register.

1. Firstly, data is obtained by the addressing mode shown in item 2of table 2.14.
2. Next, the effective address is calculated using the obtained data as the index by the ad
   mode shown in item 5 of table 2.14. The obtained data is used instead of the general r
   The result is stored in a general register. For details, see H8SX Family Software Manu

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 when available.

- Exception-handling state

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

- Program execution state

  In this state the CPU executes program instructions in sequence.

- Bus-released state

  The bus-released state occurs when the bus has been released in response to a bus re a bus master other than the CPU. While the bus is released, the CPU halts operations

- Program stop state

  This is a power-down state in which the CPU stops operating. The program stop stat when a SLEEP instruction is executed or the CPU enters hardware standby mode. Fo refer to section 20, Power-Down Modes.

Note: * A transition to the reset state occurs whenever the $\overline{RES}$ signal goes low. A transition can also be made to the reset state when the watchdog timer overflows.

Figure 2.16 State Transitions

Operating Mode	MD2	MD1	MD0	Operating Mode	Address Space	LSI Initiation Mode	On-Chip ROM	Defau
1	0	0	1	Advanced	16 Mbytes	User boot mode	Enabled	—
2	0	1	0			Boot mode	Enabled	—
3	0	1	1			Reserved (setting prohibited)		
4	1	0	0	Advanced	16 Mbytes	On-chip ROM disabled extended mode	Disabled	16 bi
5	1	0	1				Disabled	8 bits
6	1	1	0			On-chip ROM enabled extended mode	Enabled	8 bits
7	1	1	1			Single-chip mode	Enabled	—

In this LSI, an advanced mode as the CPU operating mode and a 16-Mbyte address spac
available. The initial bus widths are eight or 16 bits. As the LSI initiation mode, the exte
extended mode, on-chip ROM initiation mode single-chip initiation mode can be selecte

Modes 1 and 2 are the user boot mode and the boot mode in which the flash memory ca
programmed and erased. For details on the user boot mode and the boot mode, refer to s
Flash Memory (0.18-µm F-ZTAT Version).

Mode 7 is a single-chip mode. All I/O ports can be used as general input/output ports. T
address space cannot be accessed in the initial state, but setting the EXPE bit in the syste
register (SYSCR) to 1 enables the external address space. After the external address spa
enabled, ports D, E, and F can be used as an address output bus and ports H and I as a da
specifying the data direction register (DDR) for each port.

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

3.2.1 Mode Control Register (MDCR)

MDCR indicates the current operating mode. When MDCR is read from, the states of sig
MD2 to MD0 are latched. This latch is canceled by a reset.

Bit	15	14	13	12	11	10	9	
Bit Name	—	—	—	—	MDS3	MDS2	MDS1	
Initial Value	0	1	0	1	Undefined*	Undefined*	Undefined*	Un
R/W	R	R	R	R	R	R	R	

Bit	7	6	5	4	3	2	1	
Bit Name	—	—	—	—	—	—	—	
Initial Value	0	1	0	1	Undefined*	Undefined*	Undefined*	Un
R/W	R	R	R	R	R	R	R	

Note: * Determined by pins MD2 to MD0.

				latches are released by a reset.
7	—	0	R	Reserved
6	—	1	R	These are read-only bits and cannot be mod
5	—	0	R	
4	—	1	R	
3	—	Undefined*	R	
2	—	Undefined*	R	
1	—	Undefined*	R	
0	—	Undefined*	R	

Note: * Determined by pins MD2 to MD0.

Table 3.2 Settings of Bits MDS3 to MDS0

MCU Operating Mode	Mode Pins			MDCR		
	MD2	MD1	MD0	MDS3	MDS2	MDS1
1	0	0	1	1	1	0
2	0	1	0	1	1	0
4	1	0	0	0	0	1
5	1	0	1	0	0	0
6	1	1	0	0	1	0
7	1	1	1	0	1	0

Bit Name	—	—	—	—	—	—	DTCMD	
Initial Value	0	0	0	0	0	0	1	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Note: * The initial value depends on the startup mode.

Bit	Bit Name	Initial Value	R/W	Descriptions
15, 14	—	All 1	R/W	Reserved These bits are always read as 1. The write valu always be 1.
13	MACS	0	R/W	MAC Saturation Operation Control Selects either saturation operation or non-satu operation for the MAC instruction. 0: MAC instruction is non-saturation operation 1: MAC instruction is saturation operation
12	—	1	R/W	Reserved This bit is always read as 1. The write value sh always be 1.
11	FETCHMD	0	R/W	Instruction Fetch Mode Select This LSI can prefetch an instruction in units of 32 bits. Select the bus width for instruction fetc depending on the used memory for the storage programs*[1]. 0: 32-bit mode 1: 16-bit mode

				The external bus cycle may be carried out in p with the internal bus cycle depending on the s the write data buffer function. 0: External bus disabled 1: External bus enabled
8	RAME	1	R/W	RAM Enable Enables or disables the on-chip RAM. This bi initialized when the reset state is released. Do 0 during access to the on-chip RAM. 0: On-chip RAM disabled 1: On-chip RAM enabled
7 to 2	—	All 0	R/W	Reserved These bits are always read as 0. The write va always be 0.
1	DTCMD	1	R/W	DTC Mode Select Selects DTC operating mode. 0: DTC is in full-address mode 1: DTC is in short address mode
0	—	1	R/W	Reserved This bit is always read as 1. The write value s always be 1.

Notes: 1. For details on instruction fetch mode, see section 2.3, Instruction Fetch.
2. The initial value depends on the LSI initiation mode.
Since operating modes 4, 5 and 6 are external extended modes, EXPE is 1.

This is the boot mode for the flash memory. The LSI operates in the same way as in mode
except for programming and erasing of the flash memory. For details, refer to section 18,
Memory (0.18-µm F-ZTAT Version).

3.3.3 Mode 4

The CPU operating mode is advanced mode in which the address space is 16 Mbytes, and
chip ROM is disabled.

The initial bus width mode immediately after a reset is 16 bits, with 16-bit access to all ar
Ports D, E, and F function as an address bus, ports H and I function as a data bus, and par
ports A and B function as bus control signals. However, if all areas are designated as an 8
access space by the bus controller, the bus mode switches to eight bits, and only port H fu
as a data bus.

3.3.4 Mode 5

The CPU operating mode is advanced mode in which the address space is 16 Mbytes, and
chip ROM is disabled.

The initial bus width mode immediately after a reset is eight bits, with 8-bit access to all a
Ports D, E, and F function as an address bus, port H functions as a data bus, and parts of p
and B function as bus control signals. However, if any area is designated as a 16-bit acces
by the bus controller, the bus width mode switches to 16 bits, and ports H and I function a
bus.

3.3.6 Mode 7

The CPU operating mode is advanced mode in which the address space is 16 Mbytes, ar
chip ROM is enabled. All I/O ports can be used as general input/output ports. The extern
space cannot be accessed in the initial state, but setting the EXPE bit in the system contr
(SYSCR) to 1 enables the external address space. After the external address space is ena
D, E, and F can be used as an address output bus and ports H and I as a data bus by spec
data direction register (DDR) for each port. For details, refer to section 9, I/O Ports.

4	P/C*	P/C*	P*/C	P*/C	P/C*	A	A	A	P*/A	D
5	P/C*	P/C*	P*/C	P*/C	P/C*	A	A	A	P*/A	D
6	P/C*	P/C*	P*/C	P*/C	P*/C	P*/A	P*/A	P*/A	P*/A	D
7	P*/C	P*/C	P*/C	P*/C	P*/C	P*/A	P*/A	P*/A	P*/A	P*/D

[Legend]

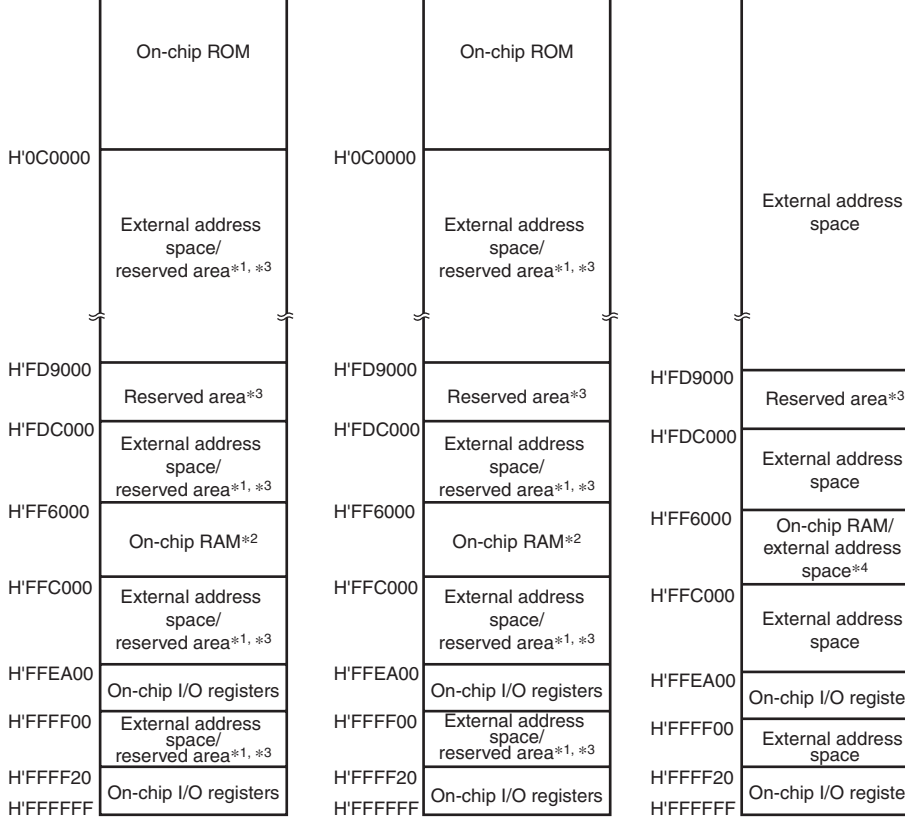
P: I/O port

A: Address bus output

D: Data bus input/output

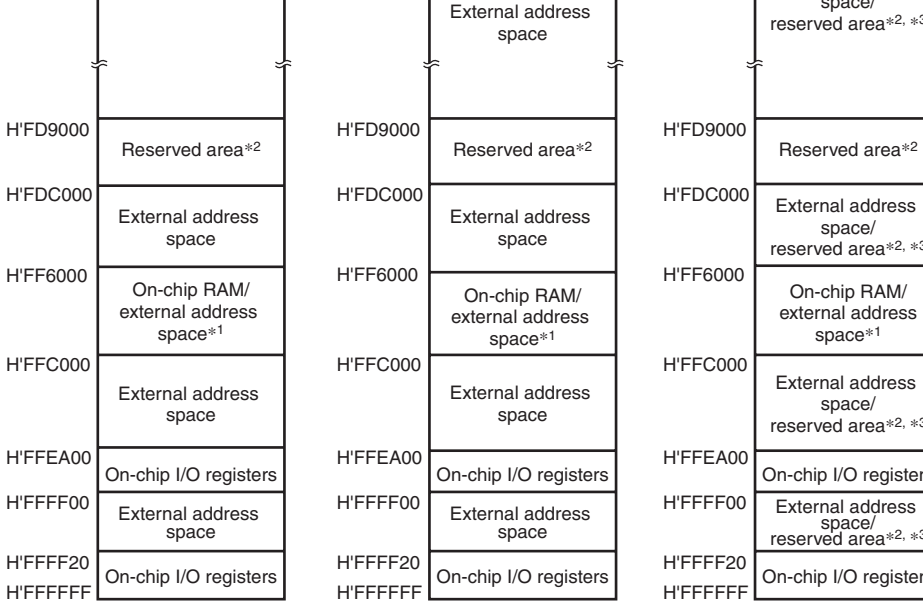
C: Control signals, clock input/output

*: Immediately after a reset



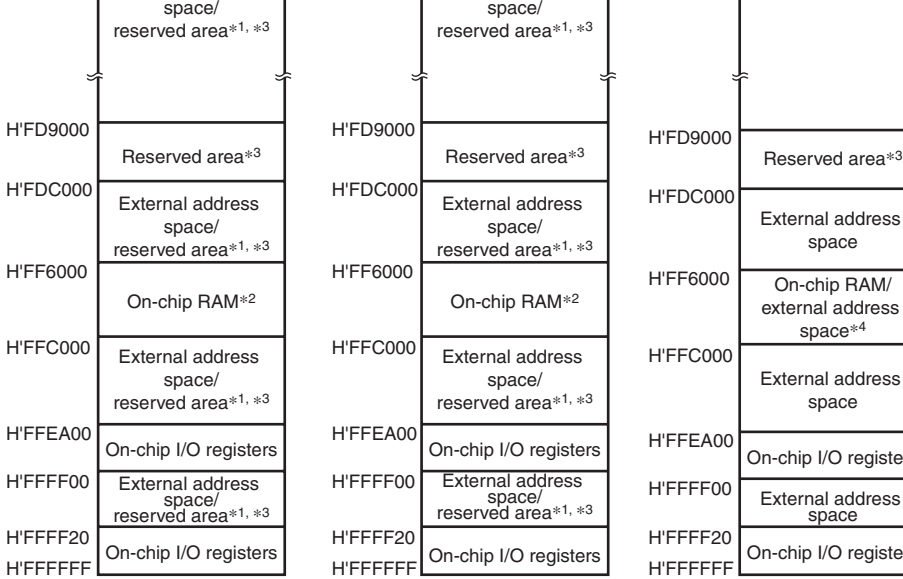
Notes: 1. This area is specified as the external address space when EXPE = 1 and the reserved area when E
2. The on-chip RAM is used for flash memory programming. Do not clear the RAME bit to 0.
3. Do not access the reserved areas.
4. This area is specified as the external address space by clearing the RAME bit to 0.

Figure 3.1 Address Map in each Operating Mode of H8SX/1657C (1)



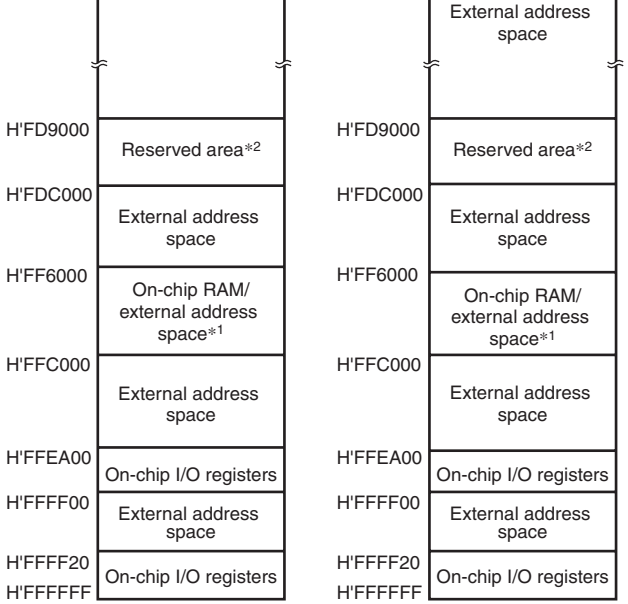
Notes: 1. This area is specified as the external address space by clearing the RAME bit to 0.
2. Do not access the reserved areas.
3. This area is specified as the external address space when EXPE = 1 and the reserved area when EX

Figure 3.1 Address Map in each Operating Mode of H8SX/1657C (2)



Notes: 1. This area is specified as the external address space when EXPE = 1 and the reserved area when E
2. The on-chip RAM is used for flash memory programming. Do not clear the RAME bit to 0.
3. Do not access the reserved areas.
4. This area is specified as the external address space by clearing the RAME bit to 0.

Figure 3.1 Address Map in each Operating Mode of H8SX/1656C (3)



Notes: 1. This area is specified as the external address space by clearing the RAME bit to 0.
2. Do not access the reserved areas.
3. This area is specified as the external address space when EXPE = 1 and the reserved area when EX

Figure 3.1 Address Map in each Operating Mode of H8SX/1656C (4)

Table 4.1 Exception Types and Priority

Priority	Exception Type	Exception Handling Start Timing
High	Reset	Exception handling starts at the timing of level cha low to high on the $\overline{\text{RES}}$ pin, or when the watchdog overflows. The CPU enters the reset state when th pin is low.
	Illegal instruction	Exception handling starts when an undefined code executed.
	Trace*[1]	Exception handling starts after execution of the cu instruction or exception handling, if the trace (T) b is set to 1.
	Address error	After an address error has occurred, exception ha starts on completion of instruction execution.
	Interrupt	Exception handling starts after execution of the cu instruction or exception handling, if an interrupt re occurred.*[2]
	Sleep instruction	Exception handling starts by execution of a sleep (SLEEP), if the SSBY bit in SBYCR is set to 0 and SLPIE bit in SBYCR is set to 1.
Low	Trap instruction*[3]	Exception handling starts by execution of a trap in (TRAPA).

Notes: 1. Traces are enabled only in interrupt control mode 2. Trace exception handling executed after execution of an RTE instruction.

2. Interrupt detection is not performed on completion of ANDC, ORC, XORC, or instruction execution, or on completion of reset exception handling.

3. Trap instruction exception handling requests and sleep instruction exception requests are accepted at all times in the program execution state.

Exception Source	Vector Number	Vector Table Address Offse	
		Normal Mode*[2]	Advanced, M Maximum*[2]
Reset	0	H'0000 to H'0001	H'0000 to H'0
Reserved for system use	1	H'0002 to H'0003	H'0004 to H'0
	2	H'0004 to H'0005	H'0008 to H'0
	3	H'0006 to H'0007	H'000C to H'
Illegal instruction	4	H'0008 to H'0009	H'0010 to H'0
Trace	5	H'000A to H'000B	H'0014 to H'0
Reserved for system use	6	H'000C to H'000D	H'0018 to H'0
Interrupt (NMI)	7	H'000E to H'000F	H'001C to H'
Trap instruction (#0)	8	H'0010 to H'0011	H'0020 to H'0
(#1)	9	H'0012 to H'0013	H'0024 to H'0
(#2)	10	H'0014 to H'0015	H'0028 to H'0
(#3)	11	H'0016 to H'0017	H'002C to H'
CPU address error	12	H'0018 to H'0019	H'0030 to H'0
DMA address error*[3]	13	H'001A to H'001B	H'0034 to H'0
Reserved for system use	14	H'001C to H'001D	H'0038 to H'0
	|	|	|
	27	H'0022 to H'0023	H'0044 to H'0
Sleep instruction	18	H'0024 to H'0025	H'0048 to H'0
Reserved for system use	19	H'0026 to H'0027	H'004C to H'
	|	|	|
	23	H'002E to H'002F	H'005C to H'

IRQ4	68	H'0088 to H'0089	H'0110 to H
IRQ5	69	H'008A to H'008B	H'0114 to H
IRQ6	70	H'008C to H'008D	H'0118 to H
IRQ7	71	H'008E to H'008F	H'011C to H
IRQ8	72	H'0090 to H'0091	H'0120 to H
IRQ9	73	H'0092 to H'0093	H'0124 to H
IRQ10	74	H'0094 to H'0095	H'0128 to H
IRQ11	75	H'0096 to H'0097	H'012C to H
Reserved for system use	76 | 79	H'0098 to H'0099 | H'009E to H'009F	H'0130 to H | H'013C to H
Internal interrupt*4	80 | 255	H'00A0 to H'00A1 | H'01FE to H'01FF	H'0140 to H | H'03FC to H

Notes: 1. Lower 16 bits of the address.
2. Not available in this LSI.
3. A DMA address error is generated by the DTC and DMAC.
4. For details of internal interrupt vectors, see section 5.5, Interrupt Exception H Vector Table.

Table 4.3 Calculation Method of Exception Handling Vector Table Address

Exception Source	Calculation Method of Vector Table Address
Reset, CPU address error	Vector table address = (vector table address offset)
Other than above	Vector table address = VBR + (vector table address offset)

[Legend]
VBR: Vector base register
Vector table address offset: See table 4.2.

The interrupt control mode is 0 immediately after a reset.

4.3.1 Reset Exception Handling

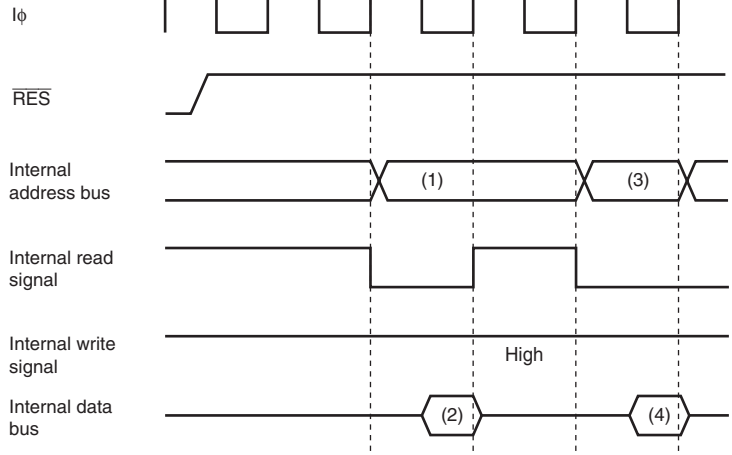
When the $\overline{\text{RES}}$ pin goes high after being held low for the necessary time, this LSI starts r
exception handling as follows:

1. The internal state of the CPU and the registers of the on-chip peripheral modules are initialized, VBR is cleared to H'00000000, the T bit is cleared to 0 in EXR, and the I b set to 1 in EXR and CCR.
2. The reset exception handling vector address is read and transferred to the PC, and pro execution starts from the address indicated by the PC.

Figures 4.1 and 4.2 show examples of the reset sequence.

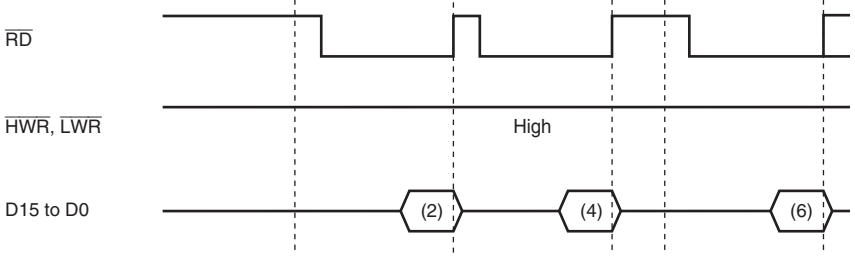
4.3.2 Interrupts after Reset

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



(1): Reset exception handling vector address (when reset, (1) = H'000000)
(2): Start address (contents of reset exception handling vector address)
(3) Start address ((3) = (2))
(4) First instruction in the exception handling routine

Figure 4.1 Reset Sequence (On-chip ROM Enabled Advanced Mode)



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

Note: * Seven program wait cycles are inserted.

Figure 4.2 Reset Sequence
(16-Bit External Access in On-chip ROM Disabled Advanced Mode)

handling routine by the RTE instruction, trace mode resumes. Trace exception handling 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

	CCR			EXR
Interrupt Control Mode	**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 the previous value.

Instruction fetch	CPU	Fetches instructions from even addresses	No (no
		Fetches instructions from odd addresses	Occurs
		Fetches instructions from areas other than on-chip peripheral module space*[1]	No (no
		Fetches instructions from on-chip peripheral module space*[1]	Occurs
		Fetches instructions from external memory space in single-chip mode	Occurs
		Fetches instructions from access prohibited area.*[2]	Occurs
Stack operation	CPU	Accesses stack when the stack pointer value is even address	No (no
		Accesses stack when the stack pointer value is odd	Occurs
Data read/write	CPU	Accesses word data from even addresses	No (no
		Accesses word data from odd addresses	No (no
		Accesses external memory space in single-chip mode	Occurs
		Accesses to access prohibited area*[2]	Occurs
Data read/write	DTC or DMAC	Accesses word data from even addresses	No (no
		Accesses word data from odd addresses	No (no
		Accesses external memory space in single-chip mode	Occurs
		Accesses to access prohibited area*[2]	Occurs
Single address transfer	DMAC	Address access space is the external memory space for single address transfer	No (no
		Address access space is not the external memory space for single address transfer	Occurs

Notes: 1. For on-chip peripheral module space, see section 6, Bus Controller (BSC).
2. For the access prohibited area, refer to figure 3.1 in section 3.4, Address Map.

program execution starts from that address.

Even though an address error occurs during a transition to an address error exception ha
address error is not accepted. This prevents an address error from occurring due to stack
exception handling, thereby preventing infinitive stacking.

If the SP contents are not a multiple of 2 when an address error exception handling occu
stacked values (PC, CCR, and EXR) are undefined.

When an address error occurs, the following is performed to halt the DTC and DMAC.

- The ERR bit of DTCCR in the DTC is set to 1.
- The ERRF bit of DMDR_0 in the DMAC is set to 1.
- The DTE bits of DMDRs for all channels in the DMAC are cleared to 0 to forcibly t
  transfer.

Table 4.6 shows the state of CCR and EXR after execution of the address error exceptio
handling.

Table 4.6 Status of CCR and EXR after Address Error Exception Handling

	CCR		EXR	
Interrupt Control Mode	**I**	**UI**	**T**	**I2 to**
0	1	—	—	—
2	1	—	0	7

[Legend]
1: Set to 1
0: Cleared to 0
—: Retains the previous value.

IRQ0 to IRQ11	Pins IRQ0 to IRQ11 (external input)	12
On-chip peripheral module	DMA controller (DMAC)	8
	Watchdog timer (WDT)	1
	A/D converter	1
	16-bit timer pulse unit (TPU)	26
	8-bit timer (TMR)	12
	Serial communications interface (SCI)	16

Different vector numbers and vector table offsets are assigned to different interrupt sourc
vector number and vector table offset, refer to table 5.2, Interrupt Sources, Vector Addres
Offsets, and Interrupt Priority in section 5, Interrupt Controller.

4.6.2 Interrupt Exception Handling

Interrupts are controlled by the interrupt controller. The interrupt controller has two intern
control modes and can assign interrupts other than NMI to eight priority/mask levels to e
multiple-interrupt control. The source to start interrupt exception handling and the vector
differ depending on the product. For details, refer to section 5, Interrupt Controller.

The interrupt exception handling is as follows:

1. The contents of PC, CCR, and EXR are saved in the stack.
2. The interrupt mask bit is updated and the T bit is cleared to 0.
3. An exception handling vector table address corresponding to the interrupt source is ge
   the start address of the exception service routine is loaded from the vector table to PC
   program execution starts from that address.

1. The contents of PC, CCR, and EXR are saved in the stack.
2. The interrupt mask bit is updated and the T bit is cleared to 0.
3. An exception handling vector table address corresponding to the vector number spec
   the TRAPA instruction is generated, the start address of the exception service routin
   from the vector table to PC, and program execution starts from that address.

A start address is read from the vector table corresponding to a vector number from 0 to
specified in the instruction code.

Table 4.8 shows the state of CCR and EXR after execution of trap instruction exception

Table 4.8 Status of CCR and EXR after Trap Instruction Exception Handling

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

[Legend]
1: Set to 1
0: Cleared to 0
—: Retains the previous value.

the SLEEP instruction is generated, the start address of the exception service routine f
from the vector table to PC, and program execution starts from that address.

Bus masters other than the CPU may gain the bus mastership after a sleep instruction has
executed. In such cases, the sleep instruction will be started when the transactions of a bu
other than the CPU has been completed and the CPU has gained the bus mastership.

Table 4.9 shows the state of CCR and EXR after execution of sleep instruction exception
handling. For details, see section 20.9, Sleep Instruction Exception Handling.

Table 4.9 Status of CCR and EXR after Sleep Instruction Exception Handling

	CCR		EXR	
Interrupt Control Mode	**I**	**UI**	**T**	**I2 to I**
0	1	—	—	—
2	1	—	0	7

[Legend]
1: Set to 1
0: Cleared to 0
—: Retains the previous value.

1. The contents of PC, CCR, and EXR are saved in the stack.
2. The interrupt mask bit is updated and the T bit is cleared to 0.
3. An exception handling vector table address corresponding to the occurred exception generated, the start address of the exception service routine is loaded from the vector PC, and program execution starts from that address.

Table 4.10 shows the state of CCR and EXR after execution of illegal instruction except handling.

Table 4.10 Status of CCR and EXR after Illegal Instruction Exception Handling

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

[Legend]

1: Set to 1

0: Cleared to 0

—: Retains the previous value.

Interrupt control mode 0 Interrupt control mode 2

Note: * Ignored on return.

Figure 4.3 Stack Status after Exception Handling

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

Performing stack manipulation while SP is set to an odd value leads to an address error. shows an example of operation when the SP value is odd.

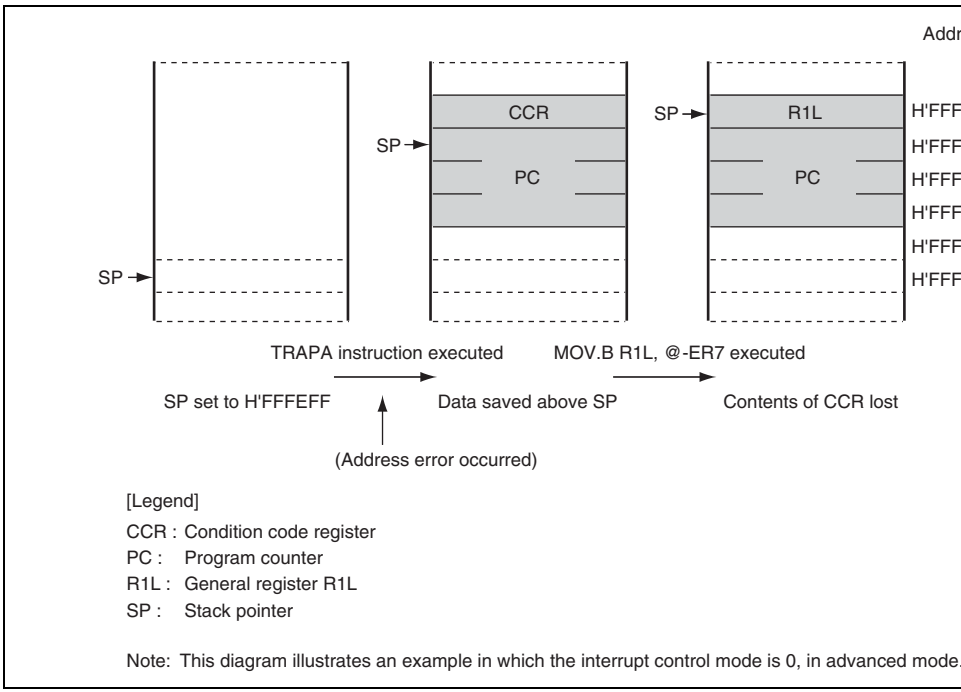


Figure 4.4 Operation when SP Value Is Odd

interrupts except for the interrupt requests listed below. The following seven interrup
are given priority of 8, therefore they are accepted at all times.

— NMI
— Illegal instructions
— Trace
— Trap instructions
— CPU address error
— DMA address error (occurred in the DTC and DMAC)
— Sleep instruction

- Independent vector addresses

  All interrupt sources are assigned independent vector addresses, making it unnecessa
  source to be identified in the interrupt handling routine.

- Thirteen external interrupts

  NMI is the highest-priority interrupt, and is accepted at all times. Rising edge or fall
  detection can be selected for NMI. Falling edge, rising edge, or both edge detection,
  sensing, can be selected for $\overline{\text{IRQ11}}$ to $\overline{\text{IRQ0}}$.

- DTC and DMAC control

  DTC and DMAC can be activated by means of interrupts.

- CPU priority control function

  The priority levels can be assigned to the CPU, DTC, and DMAC. The priority level
  CPU can be automatically assigned on an exception generation. Priority can be giver
  CPU interrupt exception handling over that of the DTC and DMAC transfer.

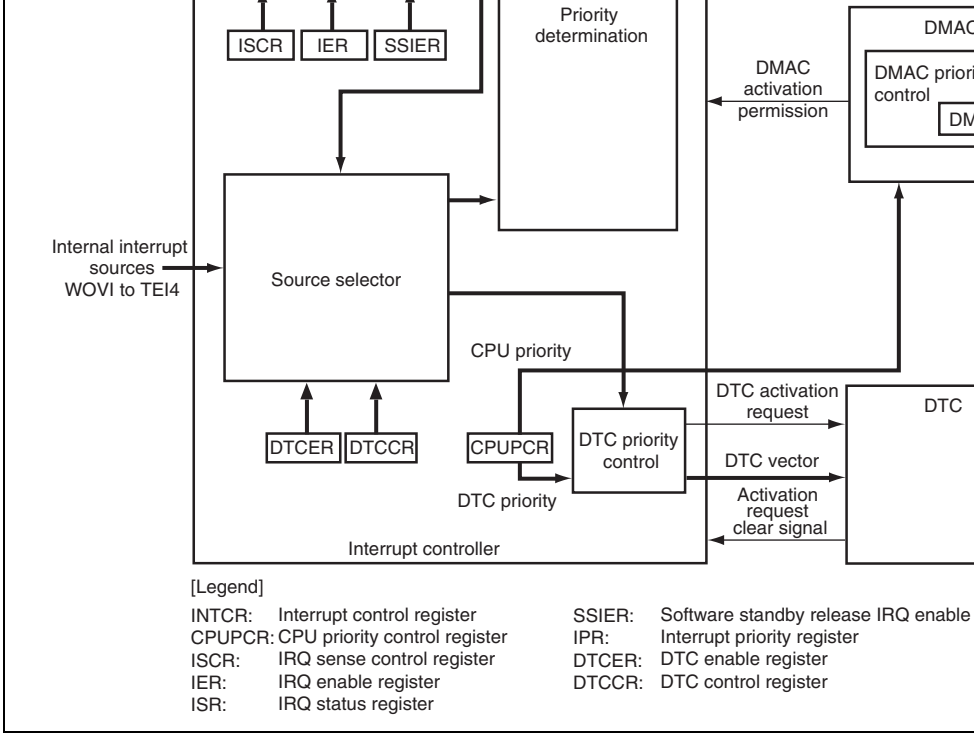


Figure 5.1 Block Diagram of Interrupt Controller

5.3 Register Descriptions

The interrupt controller has the following registers.

- Interrupt control register (INTCR)
- CPU priority control register (CPUPCR)
- Interrupt priority registers A to C, E to I, K, and L (IPRA to IPRC, IPRE to IPRI, IP
  IPRL)
- IRQ enable register (IER)
- IRQ sense control registers H and L (ISCRH, ISCRL)
- IRQ status register (ISR)
- Software standby release IRQ enable register (SSIER)

Bit	Bit Name	Value	R/W	Description
7, 6	—	All 0	R	Reserved These are read-only bits and cannot be modified
5	INTM1	0	R/W	Interrupt Control Select Mode 1 and 0
4	INTM0	0	R/W	These bits select either of two interrupt control m the interrupt controller. 00: Interrupt control mode 0 Interrupts are controlled by I bit in CCR. 01: Setting prohibited. 10: Interrupt control mode 2 Interrupts are controlled by bits I2 to I0 in EX IPR. 11: Setting prohibited.
3	NMIEG	0	R/W	NMI Edge Select Selects the input edge for the NMI pin. 0: Interrupt request generated at falling edge of I 1: Interrupt request generated at rising edge of N
2 to 0	—	All 0	R	Reserved These are read-only bits and cannot be modified

R/W R/W R/W R/W R/W R/W R/(W)* R/(W)* R

Note: * When the IPSETE bit is set to 1, the CPU priority is automatically updated, so these bits cannot be modif

Bit	Bit Name	Initial Value	R/W	Description
7	CPUPCE	0	R/W	CPU Priority Control Enable Controls the CPU priority control function. Setti to 1 enables the CPU priority control over the D DMAC. 0: CPU always has the lowest priority 1: CPU priority control enabled
6	DTCP2	0	R/W	DTC Priority Level 2 to 0
5	DTCP1	0	R/W	These bits set the DTC priority level.
4	DTCP0	0	R/W	000: Priority level 0 (lowest) 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)

1	CPUP1	0	R/(W)*	These bits set the CPU priority level. When the
0	CPUP0	0	R/(W)*	CPUPCE is set to 1, the CPU priority control fu becomes valid and the priority of CPU process assigned in accordance with the settings of bits to CPUP0. 000: Priority level 0 (lowest) 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)

Note: * When the IPSETE bit is set to 1, the CPU priority is automatically updated, so cannot be modified.

Initial Value	0	1	1	1	0	1	1	
R/W	R	R/W	R/W	R/W	R	R/W	R/W	

Bit	7	6	5	4	3	2	1	
Bit Name	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	
Initial Value	0	1	1	1	0	1	1	
R/W	R	R/W	R/W	R/W	R	R/W	R/W	

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This is a read-only bit and cannot be modified.
14 13 12	IPR14 IPR13 IPR12	1 1 1	R/W R/W R/W	Sets the priority level of the corresponding inter source. 000: Priority level 0 (lowest) 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)
11	—	0	R	Reserved This is a read-only bit and cannot be modified.

				110: Priority level 6 111: Priority level 7 (highest)
7	—	0	R	Reserved This is a read-only bit and cannot be modified.
6 5 4	IPR6 IPR5 IPR4	1 1 1	R/W R/W R/W	Sets the priority level of the corresponding interr source. 000: Priority level 0 (lowest) 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	R	Reserved This is a read-only bit and cannot be modified.

	110: Priority level 6 111: Priority level 7 (highest)

5.3.4 IRQ Enable Register (IER)

IER enables or disables interrupt requests IRQ11 to IRQ0.

Bit	15	14	13	12	11	10	9	
Bit Name	—	—	—	—	IRQ11E	IRQ10E	IRQ9E	IR
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	7	6	5	4	3	2	1	
Bit Name	IRQ7E	IRQ6E	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IR
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	—	All 0	R/W	Reserved These bits are always read as 0. The write va always be 0.
11	IRQ11E	0	R/W	IRQ11 Enable The IRQ11 interrupt request is enabled when 1.

				The IRQ7 interrupt request is enabled when th
6	IRQ6E	0	R/W	IRQ6 Enable The IRQ6 interrupt request is enabled when th
5	IRQ5E	0	R/W	IRQ5 Enable The IRQ5 interrupt request is enabled when th
4	IRQ4E	0	R/W	IRQ4 Enable The IRQ4 interrupt request is enabled when th
3	IRQ3E	0	R/W	IRQ3 Enable The IRQ3 interrupt request is enabled when th
2	IRQ2E	0	R/W	IRQ2 Enable The IRQ2 interrupt request is enabled when th
1	IRQ1E	0	R/W	IRQ1 Enable The IRQ1 interrupt request is enabled when th
0	IRQ0E	0	R/W	IRQ0 Enable The IRQ0 interrupt request is enabled when th

Bit	15	14	13	12	11	10	9
Bit Name	—	—	—	—	—	—	—
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1
Bit Name	IRQ11SR	IRQ11SF	IRQ10SR	IRQ10SF	IRQ9SR	IRQ9SF	IRQ8SR
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- ISCRL

Bit	15	14	13	12	11	10	9
Bit Name	IRQ7SR	IRQ7SF	IRQ6SR	IRQ6SF	IRQ5SR	IRQ5SF	IRQ4SR
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1
Bit Name	IRQ3SR	IRQ3SF	IRQ2SR	IRQ2SF	IRQ1SR	IRQ1SF	IRQ0SR
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

				10: Interrupt request generated at rising edge o
				11: Interrupt request generated at both falling a edges of $\overline{\text{IRQ11}}$
5	IRQ10SR	0	R/W	IRQ10 Sense Control Rise
4	IRQ10SF	0	R/W	IRQ10 Sense Control Fall
				00: Interrupt request generated by low level of
				01: Interrupt request generated at falling edge
				10: Interrupt request generated at rising edge o
				11: Interrupt request generated at both falling a edges of $\overline{\text{IRQ10}}$
3	IRQ9SR	0	R/W	IRQ9 Sense Control Rise
2	IRQ9SF	0	R/W	IRQ9 Sense Control Fall
				00: Interrupt request generated by low level of
				01: Interrupt request generated at falling edge
				10: Interrupt request generated at rising edge o
				11: Interrupt request generated at both falling a edges of $\overline{\text{IRQ9}}$
1	IRQ8SR	0	R/W	IRQ8 Sense Control Rise
0	IRQ8SF	0	R/W	IRQ8 Sense Control Fall
				00: Interrupt request generated by low level of
				01: Interrupt request generated at falling edge
				10: Interrupt request generated at rising edge o
				11: Interrupt request generated at both falling a edges of $\overline{\text{IRQ8}}$

				edges of $\overline{IRQ7}$
13	IRQ6SR	0	R/W	IRQ6 Sense Control Rise
12	IRQ6SF	0	R/W	IRQ6 Sense Control Fall 00: Interrupt request generated by low level of $\overline{I}$ 01: Interrupt request generated at falling edge o 10: Interrupt request generated at rising edge o 11: Interrupt request generated at both falling a edges of $\overline{IRQ6}$
11	IRQ5SR	0	R/W	IRQ5 Sense Control Rise
10	IRQ5SF	0	R/W	IRQ5 Sense Control Fall 00: Interrupt request generated by low level of $\overline{I}$ 01: Interrupt request generated at falling edge o 10: Interrupt request generated at rising edge o 11: Interrupt request generated at both falling a edges of $\overline{IRQ5}$
9	IRQ4SR	0	R/W	IRQ4 Sense Control Rise
8	IRQ4SF	0	R/W	IRQ4 Sense Control Fall 00: Interrupt request generated by low level of $\overline{I}$ 01: Interrupt request generated at falling edge o 10: Interrupt request generated at rising edge o 11: Interrupt request generated at both falling a edges of $\overline{IRQ4}$

4	IRQ2SF	0	R/W	IRQ2 Sense Control Fall 00: Interrupt request generated by low level of $\overline{\text{IR}}$ 01: Interrupt request generated at falling edge of 10: Interrupt request generated at rising edge of 11: Interrupt request generated at both falling an edges of $\overline{\text{IRQ2}}$
3	IRQ1SR	0	R/W	IRQ1 Sense Control Rise
2	IRQ1SF	0	R/W	IRQ1 Sense Control Fall 00: Interrupt request generated by low level of $\overline{\text{IR}}$ 01: Interrupt request generated at falling edge of 10: Interrupt request generated at rising edge of 11: Interrupt request generated at both falling an edges of $\overline{\text{IRQ1}}$
1	IRQ0SR	0	R/W	IRQ0 Sense Control Rise
0	IRQ0SF	0	R/W	IRQ0 Sense Control Fall 00: Interrupt request generated by low level of $\overline{\text{IR}}$ 01: Interrupt request generated at falling edge of 10: Interrupt request generated at rising edge of 11: Interrupt request generated at both falling an edges of $\overline{\text{IRQ0}}$

Initial Value	0	0	0	0	0	0	0	
R/W	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R

Note: * Only 0 can be written, to clear the flag. The bit manipulation instructions or memory operation instructio
be used to clear the flag.

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	—	All 0	R/W	Reserved These bits are always read as 0. The write va always be 0.
11	IRQ11F	0	R/(W)*	[Setting condition]
10	IRQ10F	0	R/(W)*	• When the interrupt selected by ISCR occu
9	IRQ9F	0	R/(W)*	[Clearing conditions]
8	IRQ8F	0	R/(W)*	• Writing 0 after reading IRQnF = 1
7	IRQ7F	0	R/(W)*	• When interrupt exception handling is exec
6	IRQ6F	0	R/(W)*	low-level sensing is selected and $\overline{\text{IRQn}}$ inp
5	IRQ5F	0	R/(W)*	• When IRQn interrupt exception handling is
4	IRQ4F	0	R/(W)*	when falling-, rising-, or both-edge sensing
3	IRQ3F	0	R/(W)*	selected
2	IRQ2F	0	R/(W)*	• When the DTC is activated by an IRQn int
1	IRQ1F	0	R/(W)*	and the DISEL bit in MRB of the DTC is cl
0	IRQ0F	0	R/(W)*	

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

Bit	7	6	5	4	3	2	1	0
Bit Name	SSI7	SSI6	SSI5	SSI4	SSI3	SSI2	SSI1	SS
Initial Value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	—	All 0	R/W	Reserved These bits are always read as 0. The write valu always be 0.
11	SSI11	0	R/W	Software Standby Release IRQ Setting
10	SSI10	0	R/W	These bits select the $\overline{\text{IRQn}}$ pins used to leave s standby mode (n = 11 to 0).
9	SSI9	0	R/W	
8	SSI8	0	R/W	0: IRQn requests are not sampled in software s mode
7	SSI7	0	R/W	
6	SSI6	0	R/W	1: When an IRQn request occurs in software st mode, this LSI leaves software standby mod the oscillation settling time has elapsed
5	SSI5	0	R/W	
4	SSI4	0	R/W	
3	SSI3	0	R/W	
2	SSI2	0	R/W	
1	SSI1	0	R/W	
0	SSI0	0	R/W	

the CPU regardless of the interrupt control mode of the settings of the CPU interrupt mas
The NMIEG bit in INTCR selects whether an interrupt is requested at the rising or fallin
the NMI pin.

When an NMI interrupt is generated, the interrupt controller determines that an error has
and performs the following procedure.

- Sets the ERR bit of DTCCR in the DTC to 1.
- Sets the ERRF bit of DMDR_0 in the DMAC to 1
- Clears the DTE bits of DMDRs for all channels in the DMAC to 0 to forcibly termin
  transfer

(2) IRQn Interrupts

An IRQn interrupt is requested by a signal input on pins $\overline{\text{IRQ11}}$ to $\overline{\text{IRQ0}}$. $\overline{\text{IRQn}}$ (n = 11 t
the following features:

- Using ISCR, it is possible to select whether an interrupt is generated by a low level,
  edge, rising edge, or both edges, on pins $\overline{\text{IRQn}}$.
- Enabling or disabling of interrupt requests IRQn can be selected by IER.
- The interrupt priority can be set by IPR.
- The status of interrupt requests IRQn is indicated in ISR. ISR flags can be cleared to
  software. The bit manipulation instructions and memory operation instructions shoul
  to clear the flag.

Detection of IRQn interrupts is enabled through the P1ICR, P2ICR, and P5ICR register
and does not change regardless of the output setting. However, when a pin is used as an
interrupt input pin, the pin must not be used as an I/O pin for another function by clearin
corresponding DDR bit to 0.

Figure 5.2 Block Diagram of Interrupts IRQn

When the IRQ sensing control in ISCR is set to a low level of signal $\overline{\text{IRQn}}$, the level of $\overline{\text{IR}}$
should be held low until an interrupt handling starts. Then set the corresponding input sig
to high in the interrupt handling routine and clear the IRQnF to 0. Interrupts may not be e
when the corresponding input signal $\overline{\text{IRQn}}$ is set to high before the interrupt handling beg

5.4.2 Internal Interrupts

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

- For each on-chip peripheral module there are flags that indicate the interrupt request s
  and enable bits that enable or disable these interrupts. They can be controlled indepen
  When the enable bit is set to 1, an interrupt request is issued to the interrupt controller
- The interrupt priority can be set by means of IPR.
- The DTC and DMAC can be activated by a TPU, SCI, or other interrupt request.
- The priority levels of DTC and DMAC activation can be controlled by the DTC and I
  priority control functions.

Classification	Interrupt Source	Vector Number	Vector Address Offset*	IPR	Priority	DTC Activation
External pin	NMI	7	H'001C	—	High	—
	IRQ0	64	H'0100	IPRA14 to IPRA12	↑	O
	IRQ1	65	H'0104	IPRA10 to IPRA8		O
	IRQ2	66	H'0108	IPRA6 to IPRA4		O
	IRQ3	67	H'010C	IPRA2 to IPRA0		O
	IRQ4	68	H'0110	IPRB14 to IPRB12		O
	IRQ5	69	H'0114	IPRB10 to IPRB8		O
	IRQ6	70	H'0118	IPRB6 to IPRB4		O
	IRQ7	71	H'011C	IPRB2 to IPRB0		O
	IRQ8	72	H'0120	IPRC14 to IPRC12		O
	IRQ9	73	H'0124	IPRC10 to IPRC8		O
	IRQ10	74	H'0128	IPRC6 to IPRC4		O
	IRQ11	75	H'012C	IPRC2 to IPRC0		O
—	Reserved for system use	76	H'0130	—		—
		77	H'0134			—
		78	H'0138			—
		79	H'013C			—
		80	H'0140			—
WDT	WOVI	81	H'0144	IPRE10 to IPRE8	Low	—

	TGI0B	89	H'0164			O
	TGI0C	90	H'0168			O
	TGI0D	91	H'016C			O
	TCI0V	92	H'0170			—
TPU_1	TGI1A	93	H'0174	IPRF2 to IPRF0		O
	TGI1B	94	H'0178			O
	TCI1V	95	H'017C			—
	TCI1U	96	H'0180			—
TPU_2	TGI2A	97	H'0184	IPRG14 to IPRG12		O
	TGI2B	98	H'0188			O
	TCI2V	99	H'018C			—
	TCI2U	100	H'0190			—
TPU_3	TGI3A	101	H'0194	IPRG10 to IPRG8		O
	TGI3B	102	H'0198			O
	TGI3C	103	H'019C			O
	TGI3D	104	H'01A0			O
	TCI3V	105	H'01A4			—
TPU_4	TGI4A	106	H'01A8	IPRG6 to IPRG4		O
	TGI4B	107	H'01AC			O
	TCI4V	108	H'01B0			—
	TCI4U	109	H'01B4		Low	—

	CMIB0	117	H'01D4			O
	OVI0	118	H'01D8			—
TMR_1	CMIA1	119	H'01DC	IPRH10 to IPRH8		O
	CMIB1	120	H'01E0			O
	OVI1	121	H'01E4			—
TMR_2	CMIA2	122	H'01E8	IPRH6 to IPRH4		O
	CMIB2	123	H'01EC			O
	OVI2	124	H'01F0			—
TMR_3	CMIA3	125	H'01F4	IPRH2 to IPRH0		O
	CMIB3	126	H'01F8			O
	OVI3	127	H'01FC			—
DMAC	DMTEND0	128	H'0200	IPRI14 to IPRI12		O
	DMTEND1	129	H'0204	IPRI10 to IPRI8		O
	DMTEND2	130	H'0208	IPRI6 to IPRI4		O
	DMTEND3	131	H'020C	IPRI2 to IPRI0		O
—	Reserved for system use	132	H'0210	—		—
		133	H'0214			—
		134	H'0218			—
		135	H'021C			—
DMAC	DMEEND0	136	H'0220	IPRK14 to IPRK12		O
	DMEEND1	137	H'0224			O
	DMEEND2	138	H'0228			O
	DMEEND3	139	H'022C		Low	O

	TEI0	147	H'024C			—
SCI_1	ERI1	148	H'0250	IPRK2 to IPRK0		—
	RXI1	149	H'0254			O
	TXI1	150	H'0258			O
	TEI1	151	H'025C			—
SCI_2	ERI2	152	H'0260	IPRL14 to IPRL12		—
	RXI2	153	H'0264			O
	TXI2	154	H'0268			O
	TEI2	155	H'026C			—
—	Reserved for system use	156	H'0270	—		—
		157	H'0274			—
		158	H'0278			—
		159	H'027C			—
SCI_4	ERI4	160	H'0280	IPRL6 to IPRL4		—
	RXI4	161	H'0284			O
	TXI4	162	H'0288			O
	TEI4	163	H'028C			—
—	Reserved for system use	164	H'0290	—		—
		|	|			|
		255	H'03FC		Low	—

Note: * Lower 16 bits of the start address in advanced, middle, and maximum modes.

0	Default	I	The priority levels of the interrupt sou fixed default settings. The interrupts except for NMI is mask I bit.
2	IPR	I2 to I0	Eight priority levels can be set for inte sources except for NMI with IPR. 8-level interrupt mask control is perfo bits I2 to I0.

5.6.1 Interrupt Control Mode 0

In interrupt control mode 0, interrupt requests except for NMI are masked by the I bit in
the CPU. Figure 5.3 shows a flowchart of the interrupt acceptance operation in this case

1. If an interrupt request occurs when the corresponding interrupt enable bit is set to 1,
   interrupt request is sent to the interrupt controller.
2. If the I bit in CCR is set to 1, only an NMI interrupt is accepted, and other interrupt r
   are held pending. If the I bit is cleared to 0, an interrupt request is accepted.
3. For multiple interrupt requests, the interrupt controller selects the interrupt request w
   highest priority, sends the request to the CPU, and holds other interrupt requests pen
4. When the CPU accepts the interrupt request, it starts interrupt exception handling aft
   execution of the current instruction has been completed.
5. The PC and CCR contents are saved to the stack area during the interrupt exception h
   The PC contents saved on the stack is the address of the first instruction to be execut
   returning from the interrupt handling routine.
6. Next, the I bit in CCR is set to 1. This masks all interrupts except NMI.

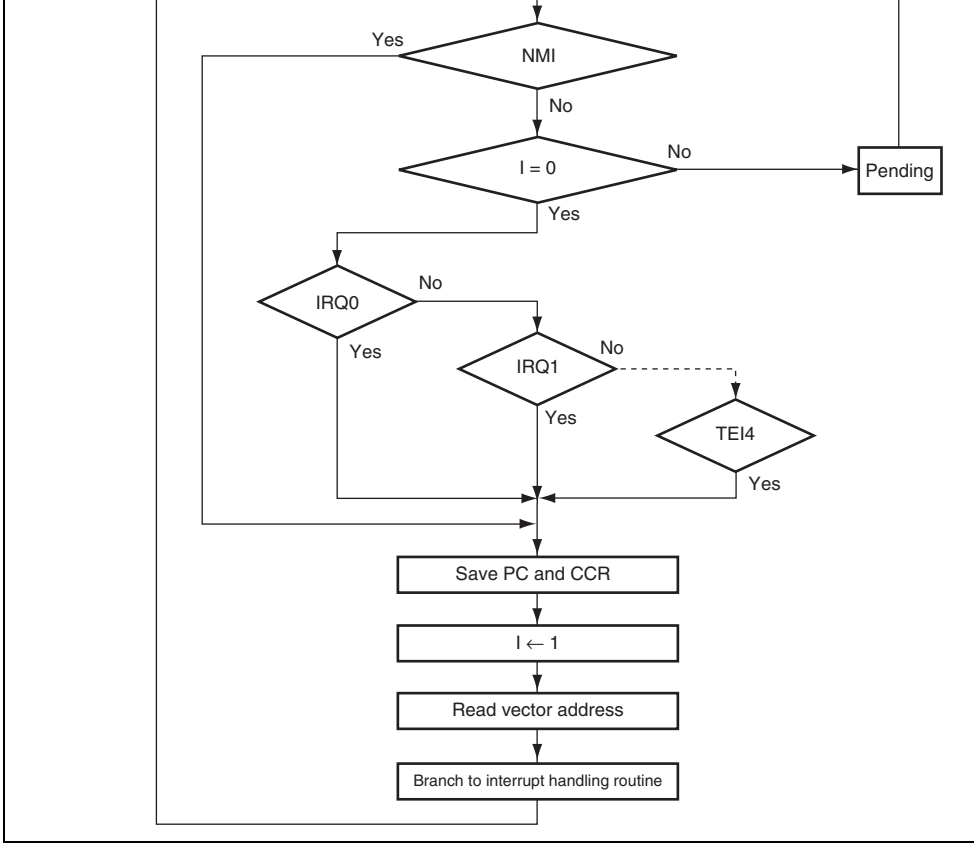


Figure 5.3 Flowchart of Procedure Up to Interrupt Acceptance in Interrupt Control Mode 0

multiple interrupt requests has the same priority, an interrupt request is selected acco
the default setting shown in table 5.2.

3. Next, the priority of the selected interrupt request is compared with the interrupt mas
in EXR. When the interrupt request does not have priority over the mask level set, it
pending, and only an interrupt request with a priority over the interrupt mask level is
4. When the CPU accepts an interrupt request, it starts interrupt exception handling afte
execution of the current instruction has been completed.
5. The PC, CCR, and EXR contents are saved to the stack area during interrupt excepti
handling. The PC saved on the stack is the address of the first instruction to be execu
returning from the interrupt handling routine.
6. The T bit in EXR is cleared to 0. The interrupt mask level is rewritten with the priori
accepted interrupt. If the accepted interrupt is NMI, the interrupt mask level is set to
7. The CPU generates a vector address for the accepted interrupt and starts execution o
interrupt handling routine at the address indicated by the contents of the vector addre
vector table.

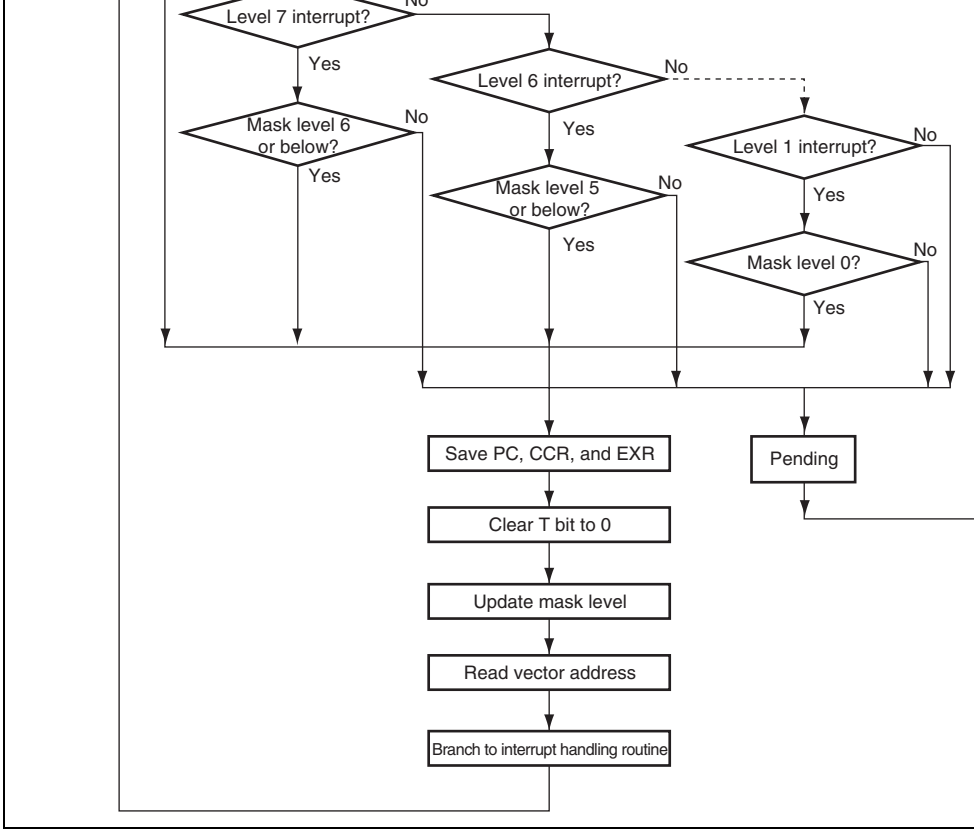


Figure 5.4 Flowchart of Procedure Up to Interrupt Acceptance in Interrupt Control Mode 2

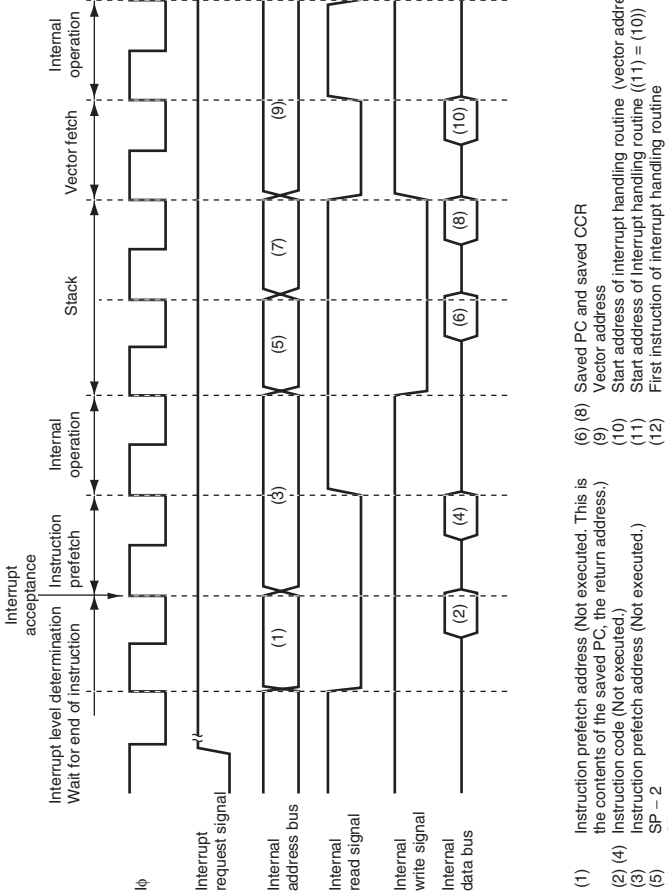


Figure 5.5 Interrupt Exception Handling

Execution State	Normal Mode*		Advanced Mode		Maximum	
	Interrupt Control Mode 0	**Interrupt Control Mode 2**	**Interrupt Control Mode 0**	**Interrupt Control Mode 2**	**Interrupt Control Mode 0**	**Interrupt Control Mode**
Interrupt priority determination*[1]	3					
Number of states until executing instruction ends*[2]	$1 \text{ to } 19 + 2 \cdot S_I$					
PC, CCR, EXR stacking	S_K to $2 \cdot S_K$*[6]	$2 \cdot S_K$	S_K to $2 \cdot S_K$*[6]	$2 \cdot S_K$	$2 \cdot S_K$	2
Vector fetch	S_h					
Instruction fetch*[3]	$2 \cdot S_I$					
Internal processing*[4]	2					
Total (using on-chip memory)	10 to 31	11 to 31	10 to 31	11 to 31	11 to 31	1

Notes: 1. Two states for an internal interrupt.
2. In the case of the MULXS or DIVXS instruction
3. Prefetch after interrupt acceptance or for an instruction in the interrupt handling
4. Internal operation after interrupt acceptance or after vector fetch
5. Not available in this LSI.
6. When setting the SP value to 4n, the interrupt response time is S_K; when settin 2, the interrupt response time is $2 \cdot S_K$.

[Legend]
m: Number of wait cycles in an external device access.

5.6.5 DTC and DMAC Activation by Interrupt

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

- Interrupt request to the CPU
- Activation request to the DTC
- Activation request to the DMAC
- Combination of the above

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

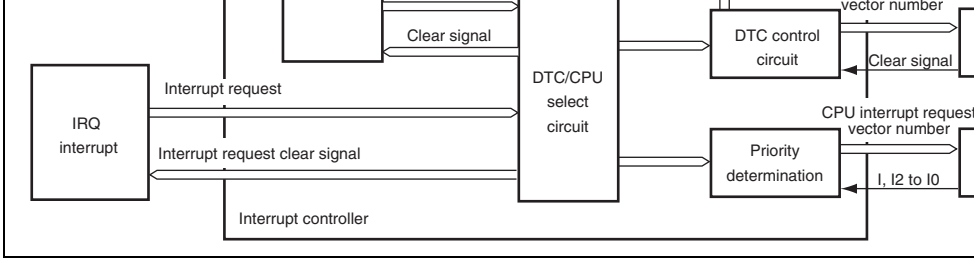


Figure 5.6 Block Diagram of DTC, DMAC, and Interrupt Controller

(1) Selection of Interrupt Sources

The activation source for each DMAC channel is selected by DMRSR. The selected activ
source is input to the DMAC through the select circuit. When transfer by an on-chip mod
interrupt is enabled (DTF1 = 1, DTF0 = 0, and DTE = 1 in DMDR) and the DTA bit in D
set to 1, the interrupt source selected for the DMAC activation source is controlled by the
and cannot be used as a DTC activation source or CPU interrupt source.

Interrupt sources that are not controlled by the DMAC are set for DTC activation sources
interrupt sources by the DTCE bit in DTCERA to DTCERH of the DTC.

Specifying the DISEL bit in MRB of the DTC generates an interrupt request to the CPU b
clearing the DTCE bit to 0 after the individual DTC data transfer.

Note that when the DTC performs a predetermined number of data transfers and the trans
counter indicates 0, an interrupt request is made to the CPU by clearing the DTCE bit to 0
DTC data transfer.

affected by its mask level or priority level. For respective priority levels, see table 8.1, in
8.4, Location of Transfer Information and DTC Vector Table.

(3) Operation Order

If the same interrupt is selected as both the DTC activation source and CPU interrupt so
CPU interrupt exception handling is performed after the DTC data transfer. If the same i
selected as the DTC or DMAC activation source or CPU interrupt source, respective ope
are performed independently.

Table 5.6 lists the selection of interrupt sources and interrupt source clear control by sett
DTA bit in DMDR of the DMAC, the DTCE bit in DTCERA to DTCERH of the DTC,
DISEL bit in MRB of the DTC.

Table 5.6 Interrupt Source Selection and Clear Control

DMAC Setting	DTC Setting		Interrupt Source Selection/Clear		
DTA	DTCE	CISEL	DMAC	DTC	CPU
0	0	*	O	X	√
	1	0	O	√	X
		1	O	O	√
1	*	*	√	X	X

[Legend]

√: The corresponding interrupt is used. The interrupt source is cleared.
(The interrupt source flag must be cleared in the CPU interrupt handling routine.)

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

X: The corresponding interrupt is not available.

*: Don't care.

CPU by assigning different priority levels to the DTC, DMAC, and CPU. Since the priori
can automatically be assigned to the CPU on an interrupt occurrence, it is possible to exe
CPU interrupt exception handling prior to the DTC or DMAC transfer.

The priority level of the CPU is assigned by bits CPUP2 to CPUP0 in CPUPCR. The prio
of the DTC is assigned by bits DTCP2 to DTCP0 in CPUPCR. The priority level of the D
assigned by bits DMAP2 to DMAP0 in DMDR for each channel.

The priority control function over the DTC and DMAC is enabled by setting the CPUPC
CPUPCR to 1. When the CPUPCE bit is 1, the DTC and DMAC activation sources are c
according to the respective priority levels.

The DTC activation source is controlled according to the priority level of the CPU indica
bits CPUP2 to CPUP0 and the priority level of the DTC indicated by bits DTCP2 to DTC
CPU has priority, the DTC activation source is held. The DTC is activated when the cond
which the activation source is held is cancelled (CPUCPCE = 1 and value of bits CPUP2
CPUP0 is greater than that of bits DTCP2 to DTCP0). The priority level of the DTC is as
the DTCP2 to DTCP0 bits regardless of the activation source.

For the DMAC, the priority level can be specified for each channel. The DMAC activatio
is controlled according to the priority level of each DMAC channel indicated by bits DM
DMAP0 and the priority level of the CPU. If the CPU has priority, the DMAC activation
held. The DMAC is activated when the condition by which the activation source is held i
cancelled (CPUCPCE = 1 and value of bits CPUP2 to CPUP0 is greater than that of bits
to DMAP0). If different priority levels are specified for channels, the channels of the high
priority levels continue transfer and the activation sources for the channels of lower prior
than that of the CPU are held.

In interrupt control mode 0, the I bit in CCR of the CPU is reflected in bit CPUP2. Bits C
and CPUP0 are fixed 0. In interrupt control mode 2, the values of bits I2 to I0 in EXR of
are reflected in bits CPUP2 to CPUP0.

Table 5.7 shows the CPU priority control.

Table 5.7 CPU Priority Control

Interrupt Control Mode	Interrupt Priority	Interrupt Mask Bit	IPSETE in CPUPCR	Control Status	
				CPUP2 to CPUP0	Updating of to CPUP0
0	Default	I = any	0	B'111 to B'000	Enabled
		I = 0	1	B'000	Disabled
		I = 1		B'100	
2	IPR setting	I2 to I0	0	B'111 to B'000	Enabled
			1	I2 to I0	Disabled

Table 5.8 shows an setting example of the priority control function over the DTC and D
the transfer request control state. A priority level can be independently set to each DMA
but the table only shows one channel for example. Transfers through the DMAC channe
separately controlled by assigning different priority levels for channels.

2	0	Any	Any	Any	Enabled	Enabl
	1	B'000	B'000	B'000	Enabled	Enabl
		B'000	B'011	B'101	Enabled	Enabl
		B'011	B'011	B'101	Enabled	Enabl
		B'100	B'011	B'101	Masked	Enabl
		B'101	B'011	B'101	Masked	Enabl
		B'110	B'011	B'101	Masked	Mask
		B'111	B'011	B'101	Masked	Mask
		B'101	B'011	B'101	Masked	Enabl
		B'101	B'110	B'101	Enabled	Enabl

be executed on completion of the instruction. However, if there is an interrupt request w
over that interrupt, interrupt exception handling will be executed for the interrupt with p
and another interrupt will be ignored. The same also applies when an interrupt source fla
cleared to 0. Figure 5.7 shows an example in which the TCIEV bit in TIER of the TPU i
to 0. The above conflict will not occur if an enable bit or interrupt source flag is cleared
the interrupt is masked.

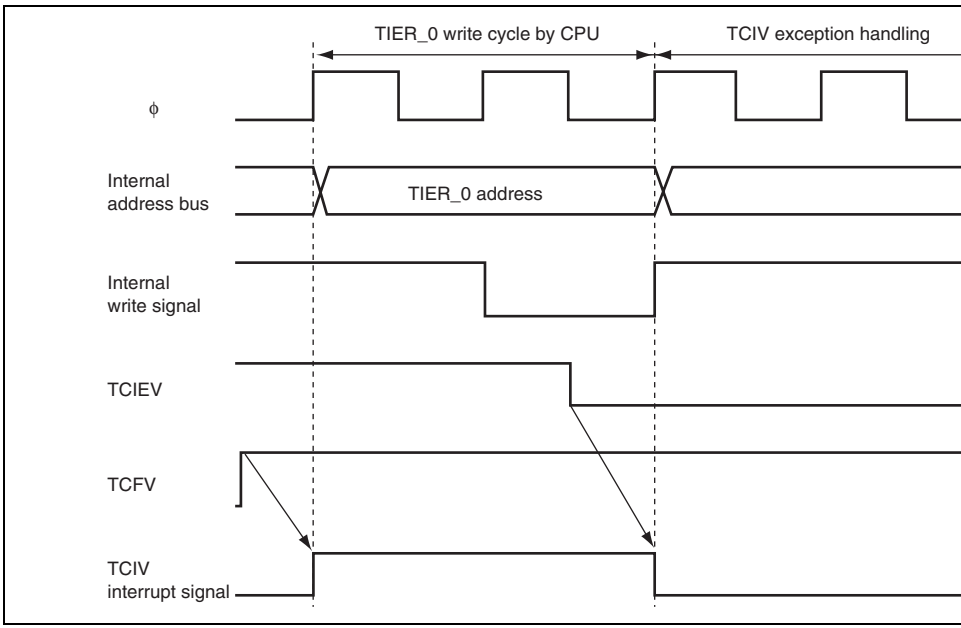


Figure 5.7 Conflict between Interrupt Generation and Disabling

5.8.3 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 ha
updated the mask level with an LDC, ANDC, ORC, or XORC instruction, and for a perio
writing to the registers of the interrupt controller.

5.8.4 Interrupts during Execution of EEPMOV Instruction

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

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

With the EEPMOV.W instruction, if an interrupt request is issued during the transfer, inte
exception handling starts at the end of the individual transfer cycle. The PC value saved o
stack in this case is the address of the next instruction. Therefore, if an interrupt is genera
during execution of an EEPMOV.W instruction, the following coding should be used.

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

after clearing the flag within the interrupt handling routine. This makes the CPU operati
synchronized with the peripheral module clock.

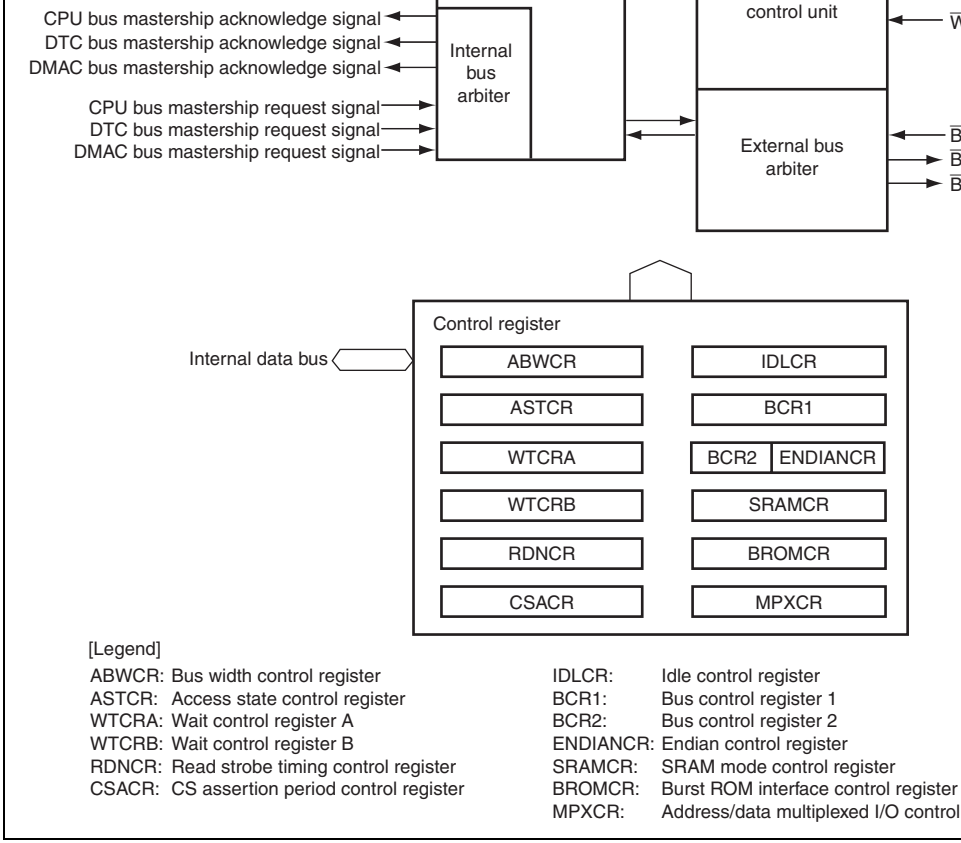
- Manages external address space in area units
  Manages the external address space divided into eight areas
  Chip select signals ($\overline{CS0}$ to $\overline{CS7}$) can be output for each area
  Bus specifications can be set independently for each area
  8-bit access or 16-bit access can be selected for each area
  Burst ROM, byte control SRAM, or address/data multiplexed I/O interface can be se
  An endian conversion function is provided to connect a device of little endian
- Basic bus interface
  This interface can be connected to the SRAM and ROM
  2-state access or 3-state access can be selected for each area
  Program wait cycles can be inserted for each area
  Wait cycles can be inserted by the $\overline{WAIT}$ pin.
  Extension cycles can be inserted while $\overline{CSn}$ is asserted for each area (n = 0 to 7)
  The negation timing of the read strobe signal ($\overline{RD}$) can be modified
- Byte control SRAM interface
  Byte control SRAM interface can be set for areas 0 to 7
  The SRAM that has a byte control pin can be directly connected
- Burst ROM interface
  Burst ROM interface can be set for areas 0 and 1
  Burst ROM interface parameters can be set independently for areas 0 and 1
- Address/data multiplexed I/O interface
  Address/data multiplexed I/O interface can be set for areas 3 to 7

DMAC single address transfers and internal accesses can be executed in parallel

- External bus release function
- Bus arbitration function

  Includes a bus arbiter that arbitrates bus mastership among the CPU, DMAC, and DT
- Multi-clock function

  The internal peripheral functions can be operated in synchronization with the periphe module clock (Pϕ). Accesses to the external address space can be operated in synchro with the external bus clock (Bϕ).
- The bus start ($\overline{BS}$) and read/write (RD/$\overline{WR}$) signals can be output.



[Legend]

ABWCR: Bus width control register
ASTCR: Access state control register
WTCRA: Wait control register A
WTCRB: Wait control register B
RDNCR: Read strobe timing control register
CSACR: CS assertion period control register
IDLCR: Idle control register
BCR1: Bus control register 1
BCR2: Bus control register 2
ENDIANCR: Endian control register
SRAMCR: SRAM mode control register
BROMCR: Burst ROM interface control register
MPXCR: Address/data multiplexed I/O control

Figure 6.1 Block Diagram of Bus Controller

- Idle control register (IDLCR)
- Bus control register 1 (BCR1)
- Bus control register 2 (BCR2)
- Endian control register (ENDIANCR)
- SRAM mode control register (SRAMCR)
- Burst ROM interface control register (BROMCR)
- Address/data multiplexed I/O control register (MPXCR)

Bit Name	ABWL7	ABWL6	ABWL5	ABWL4	ABWL3	ABWL2	ABWL1
Initial Value	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Initial value at 16-bit bus initiation is H'FEFF, and that at 8-bit bus initiation is H'FFFF.

Bit	Bit Name	Initial Value*[1]	R/W	Description
15	ABWH7	1	R/W	Area 7 to 0 Bus Width Control
14	ABWH6	1	R/W	These bits select whether the corresponding ar
13	ABWH5	1	R/W	designated as 8-bit access space or 16-bit acce
12	ABWH4	1	R/W	ABWHn ABWLn (n = 7 to 0)
11	ABWH3	1	R/W	× 0: Setting prohibited
10	ABWH2	1	R/W	0 1: Area n is designated as 16- access space
9	ABWH1	1	R/W	
8	ABWL0	1/0	R/W	1 1: Area n is designated as 8-b space*[2]
7	ABWL7	1	R/W	
6	ABWL6	1	R/W	
5	ABWL5	1	R/W	
4	ABWL4	1	R/W	
3	ABWL3	1	R/W	
2	ABWL2	1	R/W	
1	ABWL1	1	R/W	
0	ABWL0	1	R/W	

[Legend]

×: Don't care

Notes: 1. Initial value at 16-bit bus initiation is H'FEFF, and that at 8-bit bus initiation is

2. An address space specified as byte control SRAM interface must not be spec bit access space.

Bit	7	6	5	4	3	2	1	0
Bit Name	—	—	—	—	—	—	—	
Initial Value	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	

Bit	Bit Name	Initial Value	R/W	Description
15	AST7	1	R/W	Area 7 to 0 Access State Control
14	AST6	1	R/W	These bits select whether the corresponding are
13	AST5	1	R/W	designated as 2-state access space or 3-state a
12	AST4	1	R/W	space. Wait cycle insertion is enabled or disable same time.
11	AST3	1	R/W	0: Area n is designated as 2-state access space
10	AST2	1	R/W	Wait cycle insertion in area n access is disabl
9	AST1	1	R/W	1: Area n is designated as 3-state access space
8	AST0	1	R/W	Wait cycle insertion in area n access is enable
				(n = 7 to 0)
7 to 0	—	All 0	R	Reserved
				These are read-only bits and cannot be modified

Bit	7	6	5	4	3	2	1
Bit Name	—	W52	W51	W50	—	W42	W41
Initial Value	0	1	1	1	0	1	1
R/W	R	R/W	R/W	R/W	R	R/W	R/W

• WTCRB

Bit	15	14	13	12	11	10	9
Bit Name	—	W32	W31	W30	—	W22	W21
Initial Value	0	1	1	1	0	1	1
R/W	R	R/W	R/W	R/W	R	R/W	R/W

Bit	7	6	5	4	3	2	1
Bit Name	—	W12	W11	W10	—	W02	W01
Initial Value	0	1	1	1	0	1	1
R/W	R	R/W	R/W	R/W	R	R/W	R/W

				001: 1 program wait cycle inserted 010: 2 program wait cycles inserted 011: 3 program wait cycles inserted 100: 4 program wait cycles inserted 101: 5 program wait cycles inserted 110: 6 program wait cycles inserted 111: 7 program wait cycles inserted
11	—	0	R	Reserved This is a read-only bit and cannot be modified.
10 9 8	W62 W61 W60	1 1 1	R/W R/W R/W	Area 6 Wait Control 2 to 0 These bits select the number of program wait cy when accessing area 6 while bit AST6 in ASTCR 000: Program wait cycle not inserted 001: 1 program wait cycle inserted 010: 2 program wait cycles inserted 011: 3 program wait cycles inserted 100: 4 program wait cycles inserted 101: 5 program wait cycles inserted 110: 6 program wait cycles inserted 111: 7 program wait cycles inserted
7	—	0	R	Reserved This is a read-only bit and cannot be modified.

				101: 5 program wait cycles inserted 110: 6 program wait cycles inserted 111: 7 program wait cycles inserted
3	—	0	R	Reserved This is a read-only bit and cannot be modified.
2 1 0	W42 W41 W40	1 1 1	R/W R/W R/W	Area 4 Wait Control 2 to 0 These bits select the number of program wait c when accessing area 4 while bit AST4 in ASTC 000: Program wait cycle not inserted 001: 1 program wait cycle inserted 010: 2 program wait cycles inserted 011: 3 program wait cycles inserted 100: 4 program wait cycles inserted 101: 5 program wait cycles inserted 110: 6 program wait cycles inserted 111: 7 program wait cycles inserted

				001: 1 program wait cycle inserted 010: 2 program wait cycles inserted 011: 3 program wait cycles inserted 100: 4 program wait cycles inserted 101: 5 program wait cycles inserted 110: 6 program wait cycles inserted 111: 7 program wait cycles inserted
11	—	0	R	Reserved This is a read-only bit and cannot be modified.
10	W22	1	R/W	Area 2 Wait Control 2 to 0
9	W21	1	R/W	These bits select the number of program wait cy
8	W20	1	R/W	when accessing area 2 while bit AST2 in ASTCR 000: Program wait cycle not inserted 001: 1 program wait cycle inserted 010: 2 program wait cycles inserted 011: 3 program wait cycles inserted 100: 4 program wait cycles inserted 101: 5 program wait cycles inserted 110: 6 program wait cycles inserted 111: 7 program wait cycles inserted
7	—	0	R	Reserved This is a read-only bit and cannot be modified.

				101: 5 program wait cycles inserted
				110: 6 program wait cycles inserted
				111: 7 program wait cycles inserted
3	—	0	R	Reserved This is a read-only bit and cannot be modified.
2	W02	1	R/W	Area 0 Wait Control 2 to 0
1	W01	1	R/W	These bits select the number of program wait c
0	W00	1	R/W	when accessing area 0 while bit AST0 in ASTC
				000: Program wait cycle not inserted
				001: 1 program wait cycle inserted
				010: 2 program wait cycles inserted
				011: 3 program wait cycles inserted
				100: 4 program wait cycles inserted
				101: 5 program wait cycles inserted
				110: 6 program wait cycles inserted
				111: 7 program wait cycles inserted

Bit	7	6	5	4	3	2	1	0
Bit Name	—	—	—	—	—	—	—	
Initial Value	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	

Bit	Bit Name	Initial Value	R/W	Description
15	RDN7	0	R/W	Read Strobe Timing Control
14	RDN6	0	R/W	These bits set the negation timing of the read str
13	RDN5	0	R/W	corresponding area read access.
12	RDN4	0	R/W	As shown in figure 6.2, the read strobe for an are
11	RDN3	0	R/W	which the RDNn bit is set to 1 is negated one ha
10	RDN2	0	R/W	earlier than that for an area for which the RDNn
9	RDN1	0	R/W	cleared to 0. The read data setup and hold time
8	RDN0	0	R/W	given one half-cycle earlier.
				0: In an area n read access, the $\overline{\text{RD}}$ signal is neg the end of the read cycle
				1: In an area n read access, the $\overline{\text{RD}}$ signal is neg half-cycle before the end of the read cycle
				(n = 7 to 0)
7 to 0	—	All 0	R	Reserved
				These are read-only bits and cannot be modified

Notes: 1. In an external address space which is specified as byte control SRAM interfac RDNCR setting is ignored and the same operation when RDNn = 1 is performe

2. In an external address space which is specified as burst ROM interface, the R setting is ignored during CPU read accesses and the same operation when RD performed.

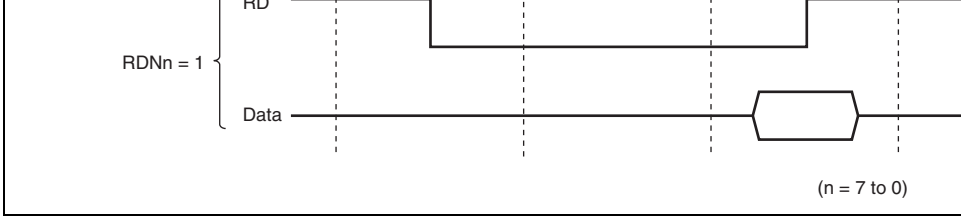


Figure 6.2 Read Strobe Negation Timing (Example of 3-State Access Spac

6.2.5 $\overline{\text{CS}}$ Assertion Period Control Registers (CSACR)

CSACR selects whether or not the assertion periods of the chip select signals ($\overline{\text{CSn}}$) and
signals for the basic bus, byte-control SRAM, burst ROM, and address/data multiplexed
interface are to be extended. Extending the assertion period of the $\overline{\text{CSn}}$ and address sign
the setup time and hold time of read strobe ($\overline{\text{RD}}$) and write strobe ($\overline{\text{LHWR}}$/$\overline{\text{LLWR}}$) to be
and to make the write data setup time and hold time for the write strobe become flexible

Bit	15	14	13	12	11	10	9
Bit Name	CSXH7	CSXH6	CSXH5	CSXH4	CSXH3	CSXH2	CSXH1
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1
Bit Name	CSXT7	CSXT6	CSXT5	CSXT4	CSXT3	CSXT2	CSXT1
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

				period (Th) is extended (n = 7 to 0)
7	CSXT7	0	R/W	$\overline{\text{CS}}$ and Address Signal Assertion Period Control
6	CSXT6	0	R/W	These bits specify whether or not the Tt cycle is
5	CSXT5	0	R/W	inserted (see figure 6.3). When an area for which
4	CSXT4	0	R/W	CSXTn is set to 1 is accessed, one Tt cycle, in w
3	CSXT3	0	R/W	$\overline{\text{CSn}}$ and address signals are retained, is inserte the normal access cycle.
2	CSXT2	0	R/W	0: In access to area n, the $\overline{\text{CSn}}$ and address ass
1	CSXT1	0	R/W	period (Tt) is not extended
0	CSXT0	0	R/W	1: In access to area n, the $\overline{\text{CSn}}$ and address ass period (Tt) is extended (n = 7 to 0)

Note: * In burst ROM interface, the CSXTn settings are ignored during CPU read acce

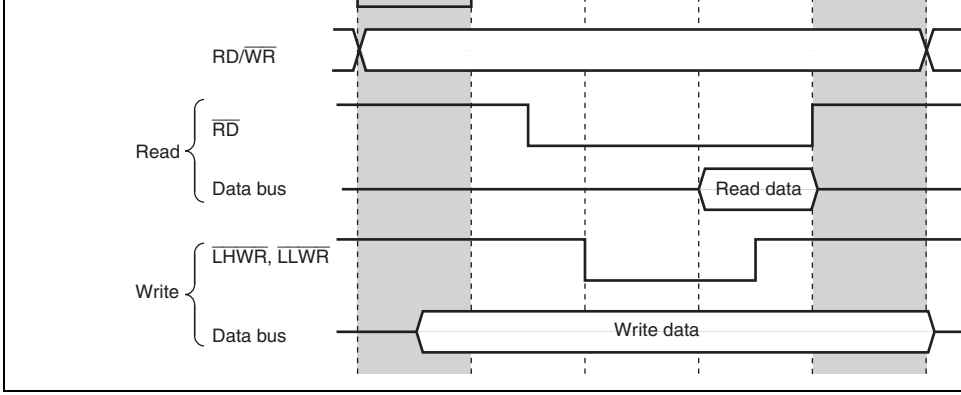


Figure 6.3 $\overline{CS}$ and Address Assertion Period Extension (Example of Basic Bus Interface, 3-State Access Space, and RDNn = 0)

6.2.6 Idle Control Register (IDLCR)

IDLCR specifies the idle cycle insertion conditions and the number of idle cycles.

Bit	15	14	13	12	11	10	9
Bit Name	IDLS3	IDLS2	IDLS1	IDLS0	IDLCB1	IDLCB0	IDLCA1
Initial Value	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1
Bit Name	IDLSEL7	IDLSEL6	IDLSEL5	IDLSEL4	IDLSEL3	IDLSEL2	IDLSEL1
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

				1: An idle cycle is inserted
13	IDLS1	1	R/W	Idle Cycle Insertion 1 Inserts an idle cycle between the bus cycles whe external read cycles of different areas continue. 0: No idle cycle is inserted 1: An idle cycle is inserted
12	IDLS0	1	R/W	Idle Cycle Insertion 0 Inserts an idle cycle between the bus cycles whe external read cycle is followed by external write c 0: No idle cycle is inserted 1: An idle cycle is inserted
11	IDLCB1	1	R/W	Idle Cycle State Number Select B
10	IDLCB0	1	R/W	Specifies the number of idle cycles to be inserted idle condition specified by IDLS1 and IDLS0. 00: No idle cycle is inserted 01: 2 idle cycles are inserted 00: 3 idle cycles are inserted 01: 4 idle cycles are inserted
9	IDLCA1	1	R/W	Idle Cycle State Number Select A
8	IDLCA0	1	R/W	Specifies the number of idle cycles to be inserted idle condition specified by IDLS3 to IDLS0. 00: 1 idle cycle is inserted 01: 2 idle cycles are inserted 10: 3 idle cycles are inserted 11: 4 idle cycles are inserted

(n = 7 to 0)

6.2.7 Bus Control Register 1 (BCR1)

BCR1 is used for selection of the external bus released state protocol, enabling/disablin
write data buffer function, and enabling/disabling of the $\overline{\text{WAIT}}$ pin input.

Bit	15	14	13	12	11	10	9	
Bit Name	BRLE	BREQOE	—	—	—	—	WDBE	
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R	R	R/W	R/W	R/W	

Bit	7	6	5	4	3	2	1	
Bit Name	DKC	—	—	—	—	—	—	
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R	R	R	R	R	

Bit	Bit Name	Initial Value	R/W	Description
15	BRLE	0	R/W	External Bus Release Enable Enables/disables external bus release. 0: External bus release disabled $\overline{\text{BREQ}}$, $\overline{\text{BACK}}$, and $\overline{\text{BREQO}}$ pins can be use ports 1: External bus release enabled* For details, see section 9, I/O Ports.

11, 10	—	All 0	R/W	Reserved These bits are always read as 0. The write value always be 0.
9	WDBE	0	R/W	Write Data Buffer Enable The write data buffer function can be used for an write cycle and a DMAC single address transfer The changed setting may not affect an external a immediately after the change. 0: Write data buffer function not used 1: Write data buffer function used
8	WAITE	0	R/W	$\overline{\text{WAIT}}$ Pin Enable Selects enabling/disabling of wait input by the $\overline{\text{W}}$ 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 For details, see section 9, I/O Ports.
7	DKC	0	R/W	$\overline{\text{DACK}}$ Control Selects the timing of DMAC transfer acknowledg assertion. 0: $\overline{\text{DACK}}$ signal is asserted at the Bϕ falling edge 1: $\overline{\text{DACK}}$ signal is asserted at the Bϕ rising edge
6	—	0	R/W	Reserved This bit is always read as 0. The write value shou always be 0.
5 to 0	—	All 0	R	Reserved These are read-only bits and cannot be modified

Bit	Bit Name	Value	R/W	Description
7, 6	—	All 0	R	Reserved These are read-only bits and cannot be modifie
5	—	0	R/W	Reserved This bit is always read as 0. The write value sh always be 0.
4	IBCCS	0	R/W	Internal Bus Cycle Control Select Selects the internal bus arbiter function. 0: Releases the bus mastership according to th 1: Executes the bus cycles alternatively when a mastership request conflicts with a DMAC or mastership request
3, 2	—	All 0	R	Reserved These are read-only bits and cannot be modifie
1	—	1	R/W	Reserved This bit is always read as 1. The write value sh always be 1.
0	PWDBE	0	R/W	Peripheral Module Write Data Buffer Enable Specifies whether or not to use the write data b function for the peripheral module write cycles. 0: Write data buffer function not used 1: Write data buffer function used

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
7	LE7	0	R/W	Little Endian Select
6	LE6	0	R/W	Selects the endian for the corresponding area.
5	LE5	0	R/W	0: Data format of area n is specified as big endia
4	LE4	0	R/W	1: Data format of area n is specified as little endi
3	LE3	0	R/W	(n = 7 to 2)
2	LE2	0	R/W	
1, 0	—	All 0	R	Reserved These are read-only bits and cannot be modified

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Bit	7	6	5	4	3	2	1	
Bit Name	—	—	—	—	—	—	—	
Initial Value	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	

Bit	Bit Name	Initial Value	R/W	Description
15	BCSEL7	0	R/W	Byte Control SRAM Interface Select
14	BCSEL6	0	R/W	Selects the bus interface for the corresponding
13	BCSEL5	0	R/W	When setting a bit to 1, the bus interface select
12	BCSEL4	0	R/W	BROMCR and MPXCR must be cleared to 0.
11	BCSEL3	0	R/W	0: Area n is basic bus interface
10	BCSEL2	0	R/W	1: Area n is byte control SRAM interface
9	BCSEL1	0	R/W	(n = 7 to 0)
8	BCSEL0	0	R/W	
7 to 0	—	All 0	R	Reserved
				These are read-only bits and cannot be modifie

Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	BSRM0	0	R/W	Area 0 Burst ROM Interface Select Specifies the area 0 bus interface. To set this bit clear bit BCSEL0 in SRAMCR to 0. 0: Basic bus interface or byte-control SRAM inte 1: Burst ROM interface
14	BSTS02	0	R/W	Area 0 Burst Cycle Select
13	BSTS01	0	R/W	Specifies the number of burst cycles of area 0
12	BSTS00	0	R/W	000: 1 cycle 001: 2 cycles 010: 3 cycles 011: 4 cycles 100: 5 cycles 101: 6 cycles 110: 7 cycles 111: 8 cycles
11, 10	—	All 0	R	Reserved These are read-only bits and cannot be modified

				Specifies the area 1 bus interface as a basic int a burst ROM interface. To set this bit to 1, clear BCSEL1 in SRAMCR to 0. 0: Basic bus interface or byte-control SRAM int 1: Burst ROM interface
6	BSTS12	0	R/W	Area 1 Burst Cycle Select
5	BSTS11	0	R/W	Specifies the number of cycles of area 1 burst c
4	BSTS10	0	R/W	000: 1 cycle 001: 2 cycles 010: 3 cycles 011: 4 cycles 100: 5 cycles 101: 6 cycles 110: 7 cycles 111: 8 cycles
3, 2	—	All 0	R	Reserved These are read-only bits and cannot be modifie
1	BSWD11	0	R/W	Area 1 Burst Word Number Select
0	BSWD10	0	R/W	Selects the number of words in burst access to burst ROM interface 00: Up to 4 words (8 bytes) 01: Up to 8 words (16 bytes) 10: Up to 16 words (32 bytes) 11: Up to 32 words (64 bytes)

R/W	R/W	R/W	R/W	R/W	R/W	R	R	
Bit	7	6	5	4	3	2	1	
Bit Name	—	—	—	—	—	—	—	A
Initial Value	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	

Bit	Bit Name	Initial Value	R/W	Description
15	MPXE7	0	R/W	Address/Data Multiplexed I/O Interface Select
14	MPXE6	0	R/W	Specifies the bus interface for the corresponding
13	MPXE5	0	R/W	To set this bit to 1, clear the BCSELn bit in SRAM
12	MPXE4	0	R/W	0.
11	MPXE3	0	R/W	0: Area n is specified as a basic interface or a by control SRAM interface. 1: Area n is specified as an address/data multipl interface (n = 7 to 3)
10 to 1	—	All 0	R	Reserved These are read-only bits and cannot be modified
0	ADDEX	0	R/W	Address Output Cycle Extension Specifies whether a wait cycle is inserted for the output cycle of address/data multiplexed I/O inte 0: No wait cycle is inserted for the address outpu 1: One wait cycle is inserted for the address outp

registers of peripheral modules such as SCI and timer.

- External access cycle

  A bus that accesses external devices via the external bus interface.

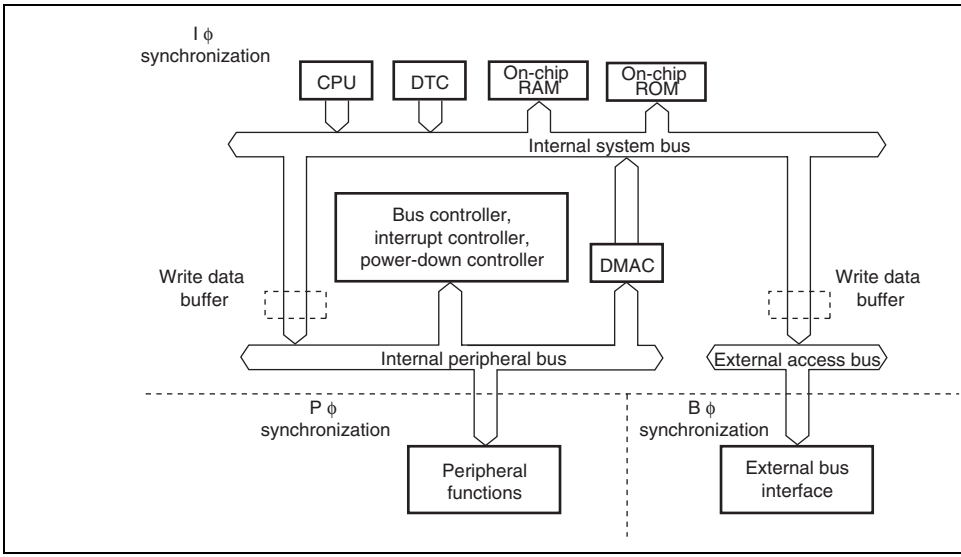


Figure 6.4 Internal Bus Configuration

	Bus controller CPU DTC DMAC Internal memory Clock pulse generator Power down control
Pϕ	I/O ports TPU PPG TMR WDT SCI A/D D/A
Bϕ	External bus interface

of access cycles is counted based on Iϕ.

If the frequencies of Iϕ, Pϕ and Bϕ are different, the start of bus cycle may not synchron
Pϕ or Bϕ according to the bus cycle initiation timing. In this case, clock synchronization
(Tsy) is inserted at the beginning of each bus cycle.

For example, if an external address space access occurs when the frequency rate of Iϕ an
n : 1, 0 to n-1 cycles of Tsy may be inserted. If an internal peripheral module access occ
the frequency rate of Iϕ and Pϕ is m : 1, 0 to m-1 cycles of Tsy may be inserted.

Figure 6.5 shows the external 2-state access timing when the frequency rate of Iϕ and B
Figure 6.6 shows the external 3-state access timing when the frequency rate of Iϕ and B

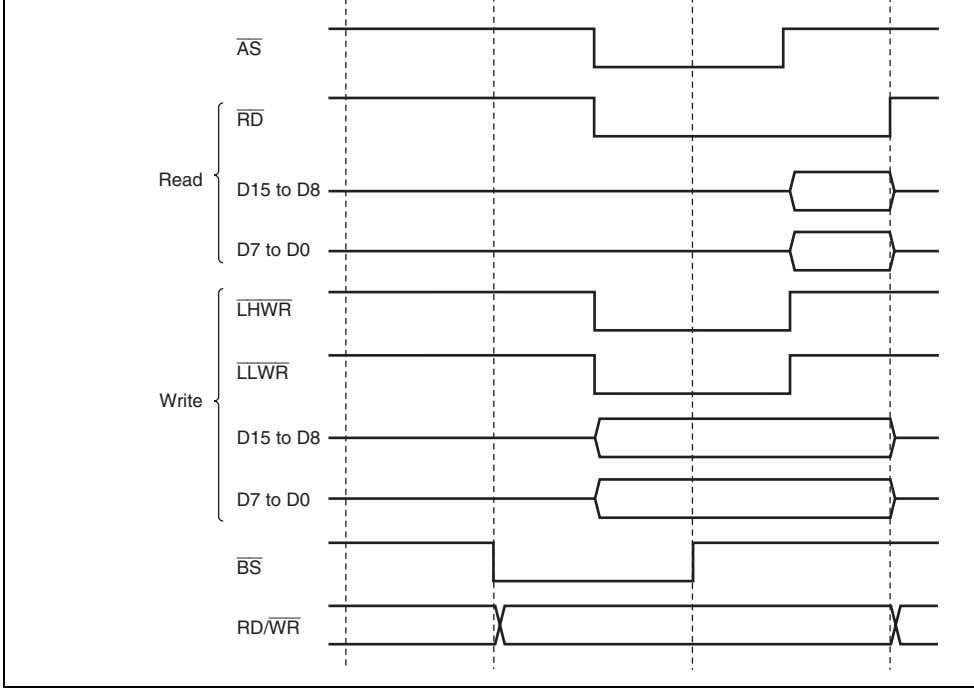


Figure 6.5 System Clock: External Bus Clock = 4:1, External 2-State Acces

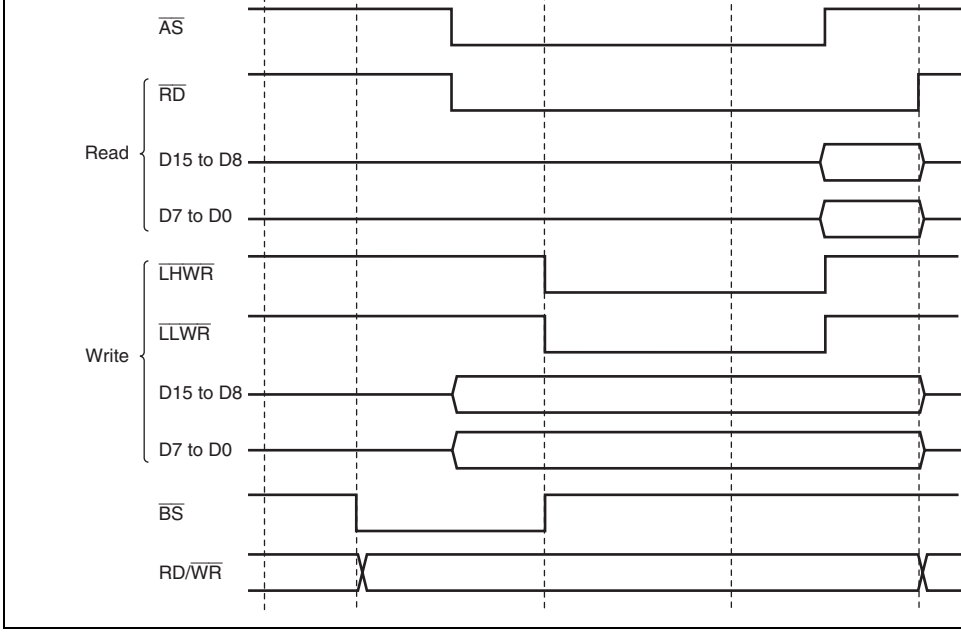


Figure 6.6 System Clock: External Bus Clock = 2:1, External 3-State Acce

Bus cycle start	BS	Output	Signal indicating that the bus cycle started
Address strobe/address hold	$\overline{AS}$/$\overline{AH}$	Output	• Strobe signal indicating that th bus, byte control SRAM, or bu space is accessed and addres on address bus is enabled • Signal to hold the address dur access to the address/data mu I/O interface
Read strobe	$\overline{RD}$	Output	Strobe signal indicating that the ba byte control SRAM, burst ROM, o address/data multiplexed I/O spac being read
Read/write	RD/$\overline{WR}$	Output	• Signal indicating the input or o direction • Write enable signal of the SRA access to the byte control SRA
Low-high write/lower-upper byte select	$\overline{LHWR}$/$\overline{LUB}$	Output	• Strobe signal indicating that th bus, burst ROM, or address/da multiplexed I/O space is writte the upper byte (D15 to D8) of is enabled • Strobe signal indicating that th control SRAM space is access the upper byte (D15 to D8) of is enabled

Chip select 1	$\overline{CS1}$	Output	Strobe signal indicating that area selected
Chip select 2	$\overline{CS2}$	Output	Strobe signal indicating that area selected
Chip select 3	$\overline{CS3}$	Output	Strobe signal indicating that area selected
Chip select 4	$\overline{CS4}$	Output	Strobe signal indicating that area selected
Chip select 5	$\overline{CS5}$	Output	Strobe signal indicating that area selected
Chip select 6	$\overline{CS6}$	Output	Strobe signal indicating that area selected
Chip select 7	$\overline{CS7}$	Output	Strobe signal indicating that area selected
Wait	$\overline{WAIT}$	Input	Wait request signal when access external address space.
Bus request	$\overline{BREQ}$	Input	Request signal for release of bus external bus master
Bus request acknowledge	$\overline{BACK}$	Output	Acknowledge signal indicating th been released to external bus ma
Bus request output	$\overline{BREQO}$	Output	External bus request signal used internal bus master accesses ext address space in the external-bu state
Data transfer acknowledge 3 (DMAC_3)	$\overline{DACK3}$	Output	Data transfer acknowledge signa DMAC_3 single address transfer
Data transfer acknowledge 2 (DMAC_2)	$\overline{DACK2}$	Output	Data transfer acknowledge signa DMAC_2 single address transfer

	Initial State			Basic Bus		SRAM	ROM		I/O		
Pin Name	**16**	**8**	**Single-Chip**	**16**	**8**	**16**	**16**	**8**	**16**	**8**	**Remark**
Bϕ	Output	Output	—	O	O	O	O	O	O	O	
$\overline{CS0}$	Output	Output	—	O	O	O	O	O	—	—	
$\overline{CS1}$	—	—	—	O	O	O	O	O	—	—	
$\overline{CS2}$	—	—	—	O	O	O	—	—	—	—	
$\overline{CS3}$	—	—	—	O	O	O	—	—	O	O	
$\overline{CS4}$	—	—	—	O	O	O	—	—	O	O	
$\overline{CS5}$	—	—	—	O	O	O	—	—	O	O	
$\overline{CS6}$	—	—	—	O	O	O	—	—	O	O	
$\overline{CS7}$	—	—	—	O	O	O	—	—	O	O	
BS	—	—	—	O	O	O	O	O	O	O	
RD/$\overline{WR}$	—	—	—	O	O	O	O	O	O	O	
$\overline{AS}$	Output	Output	—	O	O	O	O	O	—	—	
$\overline{AH}$	—	—	—	—	—	—	—	—	O	O	
$\overline{RD}$	Output	Output	—	O	O	O	O	O	O	O	
$\overline{LHWR}$/$\overline{LUB}$	Output	Output	—	O	—	O	O	—	O	—	
$\overline{LLWR}$/$\overline{LLB}$	Output	Output	—	O	O	O	O	O	O	O	
$\overline{WAIT}$	—	—	—	O	O	O	O	O	O	O	Controlle WAITE

[Legend]

O: Used as a bus control signal

—: Not used as a bus control signal (used as a port input when initialized)

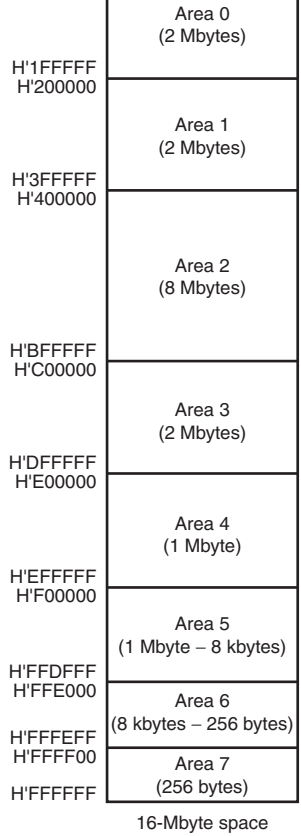


Figure 6.7 Address Space Area Division

be set to 1 when outputting signals $\overline{CS1}$ to $\overline{CS7}$.

In on-chip ROM enabled extended mode, pins $\overline{CS0}$ to $\overline{CS7}$ are all placed in the input state reset and so the corresponding PFCR bits should be set to 1 when outputting signals $\overline{CS0}$

The PFCR can specify multiple $\overline{CS}$ outputs for a pin. If multiple $\overline{CSn}$ outputs are specifie single pin by the PFCR, $\overline{CS}$ to be output are generated by mixing all the $\overline{CS}$ signals. In th the settings for the external bus interface areas in which the $\overline{CSn}$ signals are output to a si should be the same.

Figure 6.9 shows the signal output timing when the $\overline{CS}$ signals to be output to areas 5 and output to the same pin.

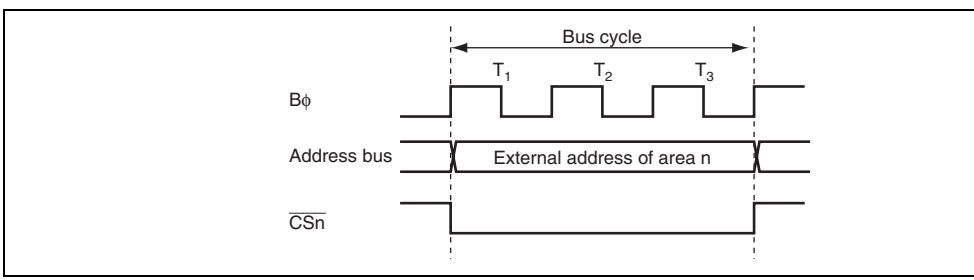


Figure 6.8 $\overline{CSn}$ Signal Output Timing (n = 0 to 7)

Figure 6.9 Timing When $\overline{CS}$ Signal is Output to the Same Pin

6.5.4 External Bus Interface

The type of the external bus interfaces, bus width, endian format, number of access cycl
strobe assert/negate timings can be set for each area in the external address space. The b
and the number of access cycles for both on-chip memory and internal I/O registers are
are not affected by the external bus settings.

Type of External Bus Interface: Four types of external bus interfaces are provided and
selected in area units. Table 6.4 shows each interface name, description, area name to be
each interface. Table 6.5 shows the areas that can be specified for each interface. The in
of each area is a basic bus interface.

Table 6.4 Interface Names and Area Names

Interface	Description	Area Name
Basic interface	Directly connected to ROM and RAM	Basic bus space
Byte control SRAM interface	Directly connected to byte SRAM with byte control pin	Byte control SRAM s
Burst ROM interface	Directly connected to the ROM that allows page access	Burst ROM space
Address/data multiplexed I/O interface	Directly connected to the peripheral LSI that requires address and data multiplexing	Address/data multipl space

(1) Bus Width

A bus width of 8 or 16 bits can be selected with ABWCR. An area for which an 8-bit bus selected functions as an 8-bit access space and an area for which a 16-bit bus is selected f as a 16-bit access space. In addition, the bus width of address/data multiplexed I/O space or 16 bits, and the bus width for the byte control SRAM space is 16 bits.

The initial state of the bus width is specified by the operating mode.

If all areas are designated as 8-bit access space, 8-bit bus mode is set; if any area is desig 16-bit access space, 16-bit bus mode is set.

(2) Endian Format

Though the endian format of this LSI is big endian, data can be converted into little endia when reading or writing to the external address space.

Areas 7 to 2 can be specified as either big endian or little endian format by the LE7 to LE ENDIANCR.

The initial state of each area is the big endian format.

Note that the data format for the areas used as a program area or a stack area should be bi

Number of access cycles in the basic bus interface
= number of basic cycles (2, 3) + number of program wait cycles (0 to 7)
+ number of $\overline{CS}$ extension cycles (0, 1, 2)
[+ number of external wait cycles by the $\overline{WAIT}$ pin]

Assertion period of the chip select signal can be extended by CSACR.

(b) Byte Control SRAM Interface

The number of access cycles in the byte control SRAM interface is the same as that in th
bus interface.

Number of access cycles in byte control SRAM interface
= number of basic cycles (2, 3) + number of program wait cycles (0 to 7)
+ number of $\overline{CS}$ extension cycles (0, 1, 2)
[+ number of external wait cycles by the $\overline{WAIT}$ pin]

(c) Burst ROM Interface

The number of access cycles at full access in the burst ROM interface is the same as tha
basic bus interface. The number of access cycles in the burst access can be specified as o
eight cycles by the BSTS bit in BROMCR.

Number of access cycles in the burst ROM interface
= number of basic cycles (2, 3) + number of program wait cycles (0 to 7)
+ number of $\overline{CS}$ extension cycles (0, 1)
[+number of external wait cycles by the $\overline{WAIT}$ pin]
+ number of burst access cycles (1 to 8) × number of burst accesses (0 to 63)

Table 6.6 lists the number of access cycles for each interface.

Table 6.6 Number of Access Cycles

Basic bus interface	=	Th [0,1]	+T1 [1]	+T2 [1]				+Tt [0,1]		
	=	Th [0,1]	+T1 [1]	+T2 [1]	+Tpw [0 to 7]	+TtW [n]	+T3 [1]	+Tt [0,1]		
Byte control SRAM interface	=	Th [0,1]	+T1 [1]	+T2 [1]				+Tt [0,1]		
	=	Th [0,1]	+T1 [1]	+T2 [1]	+Tpw [0 to 7]	+TtW [n]	+T3 [1]	+Tt [0,1]		
Burst ROM interface	=	Th [0,1]	+T1 [1]	+T2 [1]					+Tb [(1 to 8) × m]	[(2 to 3) +
	=	Th [0,1]	+T1 [1]	+T2 [1]	+Tpw [0 to 7]	+TtW [n]	+T3 [1]		+Tb [(1 to 8) × m]	[(2 to 11 + n) +
Address/data multiplexed I/O interface	= Tma [2,3]	+Th [0,1]	+T1 [1]	+T2 [1]				+Tt [0,1]		
	= Tma [2,3]	+Th [0,1]	+T1 [1]	+T2 [1]	+Tpw [0 to 7]	+TtW [n]	+T3 [1]	+Tt [0,1]		

[Legend]
Numbers: Number of access cycles
n: Pin wait (0 to ∞)
m: Number of burst accesses (0 to 63)

(4) Strobe Assert/Negate Timings

The assert and negate timings of the strobe signals can be modified as well as number of
cycles.

- Read strobe ($\overline{RD}$) in the basic bus interface
- Chip select assertion period extension cycles in the basic bus interface
- Data transfer acknowledge ($\overline{DACK3}$ to $\overline{DACK0}$) output for DMAC single address tra

selected for area 0 by bit BSRM0 in BROMCR and bit BCSEL0 in SRAMCR. Table 6.7 the external interface of area 0.

Note: Applied to the LSI version that incorporates the ROM.

Table 6.7 Area 0 External Interface

	Register Setting	
Interface	**BSRM0 of BROMCR**	**BCSEL0 of SRAMCR**
Basic bus interface	0	0
Byte control SRAM interface	0	1
Burst ROM interface	1	0
Setting prohibited	1	1

(2) Area 1

In externally extended mode, all of area 1 is external address space. In on-chip ROM en extended mode, the space excluding on-chip ROM* is external address space.

When area 1 external address space is accessed, the $\overline{CS1}$ signal can be output.

Either of the basic bus interface, byte control SRAM, or burst ROM interface can be sel area 1 by bit BSRM1 in BROMCR and bit BCSEL1 in SRAMCR. Table 6.8 shows the interface of area 1.

Note: Applied to the LSI version that incorporates the ROM.

In externally extended mode, all of area 2 is external address space.

When area 2 external address space is accessed, the $\overline{CS2}$ signal can be output.

Either the basic bus interface or byte control SRAM interface can be selected for area 2 b
BCSEL2 in SRAMCR. Table 6.9 shows the external interface of area 2.

Table 6.9 Area 2 External Interface

	Register Setting
Interface	**BCSEL2 of SRAMCR**
Basic bus interface	0
Byte control SRAM interface	1

(4) Area 3

In externally extended mode, all of area 3 is external address space.

When area 3 external address space is accessed, the $\overline{CS3}$ signal can be output.

Either of the basic bus interface, byte control SRAM interface, or address/data multiplexe
interface can be selected for area 3 by bit MPXE3 in MPXCR and bit BCSEL3 in SRAM
Table 6.10 shows the external interface of area 3.

(5) Area 4

In externally extended mode, all of area 4 is external address space.

When area 4 external address space is accessed, the $\overline{CS4}$ signal can be output.

Either of the basic bus interface, byte control SRAM interface, or address/data multiplex
interface can be selected for area 4 by bit MPXE4 in MPXCR and bit BCSEL4 in SRAM
Table 6.11 shows the external interface of area 4.

Table 6.11 Area 4 External Interface

	Register Setting	
Interface	**MPXE4 of MPXCR**	**BCSEL4 of SRAMCR**
Basic bus interface	0	0
Byte control SRAM interface	0	1
Address/data multiplexed I/O interface	1	0
Setting prohibited	1	1

interface can be selected for area 5 by the MPXE5 bit in MPXCR and the BCSEL5 bit in SRAMCR. Table 6.12 shows the external interface of area 5.

Table 6.12 Area 5 External Interface

	Register Setting	
Interface	**MPXE5 of MPXCR**	**BCSEL5 of SRAMCR**
Basic bus interface	0	0
Byte control SRAM interface	0	1
Address/data multiplexed I/O interface	1	0
Setting prohibited	1	1

(7) Area 6

Area 6 includes internal I/O registers. In external extended mode, area 6 other than on-ch register area is external address space.

When area 6 external address space is accessed, the $\overline{CS6}$ signal can be output.

Either of the basic bus interface, byte control SRAM interface, or address/data multiplexe interface can be selected for area 6 by the MPXE6 bit in MPXCR and the BCSEL6 bit in SRAMCR. Table 6.13 shows the external interface of area 6.

(8) Area 7

Area 7 includes internal I/O registers. In external extended mode, area 7 other than inter
register area is external address space.

When area 7 external address space is accessed, the $\overline{CS7}$ signal can be output.

Either of the basic bus interface, byte control SRAM interface, or address/data multiplex
interface can be selected for area 7 by the MPXE7 bit in MPXCR and the BCSEL7 bit i
SRAMCR. Table 6.14 shows the external interface of area 7.

Table 6.14 Area 7 External Interface

	Register Setting	
Interface	**MPXE7 of MPXCR**	**BCSEL7 of SRAMCR**
Basic bus interface	0	0
Byte control SRAM interface	0	1
Address/data multiplexed I/O interface	1	0
Setting prohibited	1	1

amount of data that can be accessed at one time is one byte: a word access is performed a
byte accesses, and a longword access, as four byte accesses.

Figures 6.10 and 6.11 illustrate data alignment control for the 8-bit access space. Figure 6
shows the data alignment when the data endian format is specified as big endian. Figure 6
shows the data alignment when the data endian format is specified as little endian.

Data Size	Access Address	Access Count	Bus Cycle	Data Size	Strobe sig LHWR/LUB RD Data bus D15 D8 D7
Byte	n	1	1st	Byte	7
Word	n	2	1st	Byte	15
			2nd	Byte	7
Longword	n	4	1st	Byte	31
			2nd	Byte	23
			3rd	Byte	15
			4th	Byte	7

Figure 6.10 Access Sizes and Data Alignment Control for 8-Bit Access Space (Big

			2nd	Byte
			3rd	Byte
			4th	Byte

Figure 6.11 Access Sizes and Data Alignment Control for 8-Bit Access Spa (Little Endian)

(2) 16-Bit Access Space

With the 16-bit access space, the upper byte data bus (D15 to D8) and lower byte data b D0) are used for accesses. The amount of data that can be accessed at one time is one by word.

Figures 6.12 and 6.13 illustrate data alignment control for the 16-bit access space. Figur shows the data alignment when the data endian format is specified as big endian. Figure shows the data alignment when the data endian format is specified as little endian.

In big endian, byte access for an even address is performed by using the upper byte data byte access for an odd address is performed by using the lower byte data bus.

In little endian, byte access for an even address is performed by using the lower byte dat byte access for an odd address is performed by using the third byte data bus.

Longword	Even (2n)	2	1st	Word	31 … 24 23
			2nd	Word	15 … 8 7
	Odd (2n+1)	3	1st	Byte	31
			2nd	Word	23 … 16 15
			3rd	Byte	7 … 0

Figure 6.12 Access Sizes and Data Alignment Control for 16-Bit Access Space (Big

Access Size	Access Address	Access Count	Bus Cycle	Data Size	Strobe sig LHWR/LUB RI Data b D15 … D8 D
Byte	Even (2n)	1	1st	Byte	7
	Odd (2n+1)	1	1st	Byte	7 … 0
Word	Even (2n)	1	1st	Word	15 … 8 7
	Odd (2n+1)	2	1st	Byte	7 … 0
			2nd	Byte	15
Longword	Even (2n)	2	1st	Word	15 … 8 7
			2nd	Word	31 … 24 23
	Odd (2n+1)	3	1st	Byte	7 … 0
			2nd	Word	23 … 16 15
			3rd	Byte	31

Figure 6.13 Access Sizes and Data Alignment Control for 16-Bit Access Spa (Little Endian)

accessed (8-bit access space or 16-bit access space), the data size, and endian format wh
accessing external address space,. For details, see section 6.5.6, Endian and Data Alignm

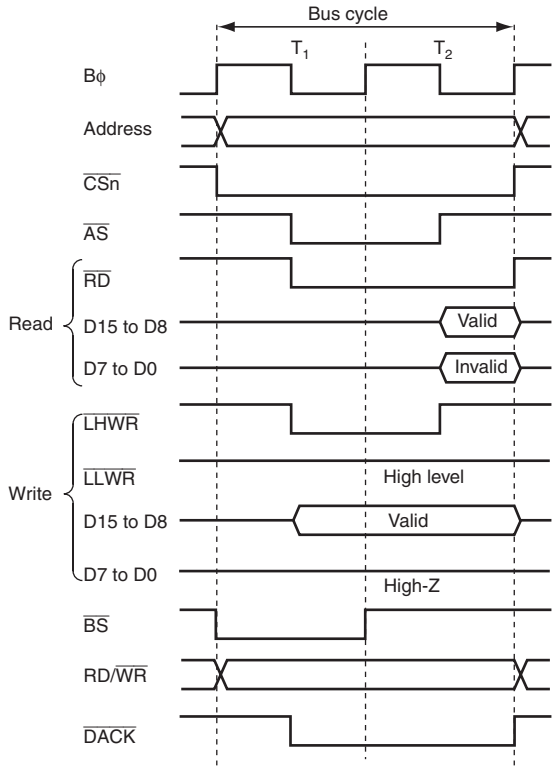
6.6.2 I/O Pins Used for Basic Bus Interface

Table 6.15 shows the pins used for basic bus interface.

Table 6.15 I/O Pins for Basic Bus Interface

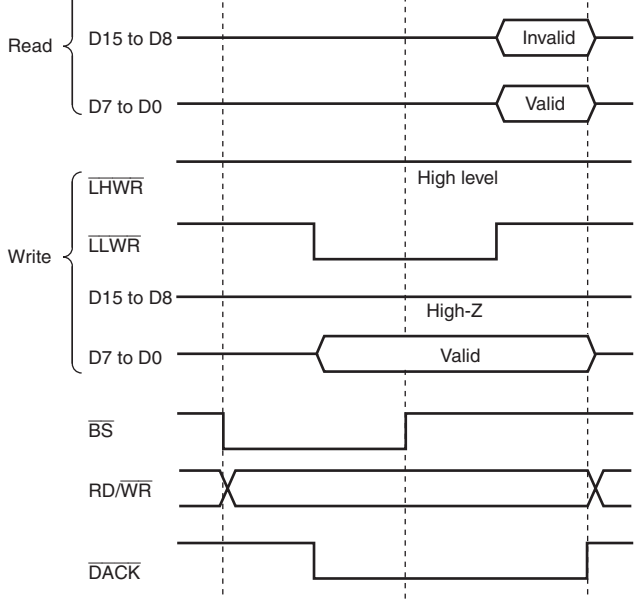
Name	Symbol	I/O	Function
Bus cycle start	$\overline{BS}$	Output	Signal indicating that the bus cycle has sta
Address strobe	$\overline{AS}$*	Output	Strobe signal indicating that an address ou address bus is valid during access
Read strobe	$\overline{RD}$	Output	Strobe signal indicating the read access
Read/write	RD/$\overline{WR}$	Output	Signal indicating the data bus input or outp direction
Low-high write	$\overline{LHWR}$	Output	Strobe signal indicating that the upper byte D8) is valid during write access
Low-low write	$\overline{LLWR}$	Output	Strobe signal indicating that the lower byte D0) is valid during write access
Chip select 0 to 7	$\overline{CS0}$ to $\overline{CS7}$	Output	Strobe signal indicating that the area is sel
Wait	$\overline{WAIT}$	Input	Wait request signal used when an external space is accessed

Note: * When the address/data multiplexed I/O is selected, this pin only functions as output and does not function as the AS output.



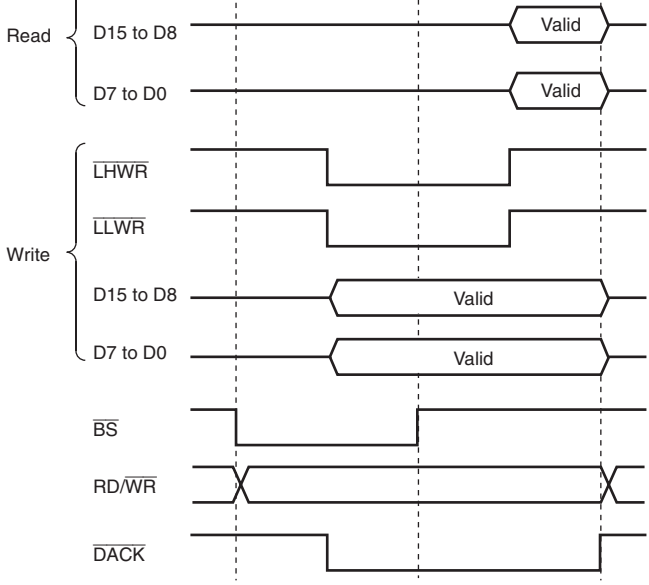
Notes: 1. n = 0 to 7
2. When RDNn = 0
3. When DKC = 0

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



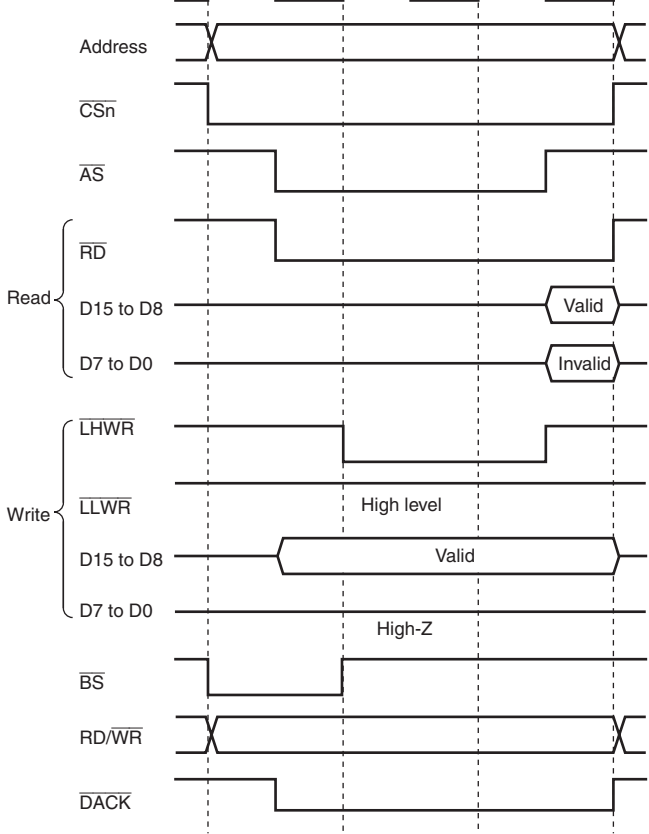
Notes: 1. n = 0 to 7
2. When RDNn = 0
3. When DKC = 0

Figure 6.15 16-Bit 2-State Access Space Bus Timing (Byte Access for Odd Address)



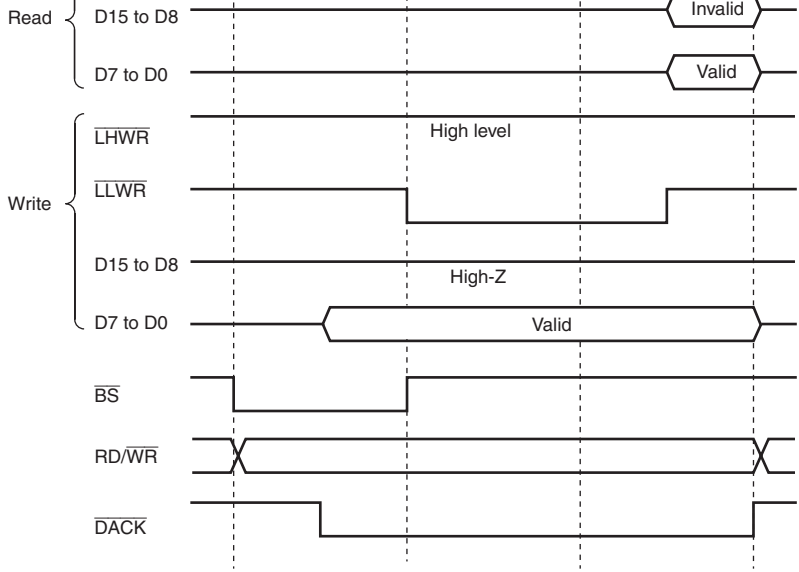
Notes: 1. n = 0 to 7
2. When RDNn = 0
3. When DKC = 0

Figure 6.16 16-Bit 2-State Access Space Bus Timing (Word Access for Even Address)



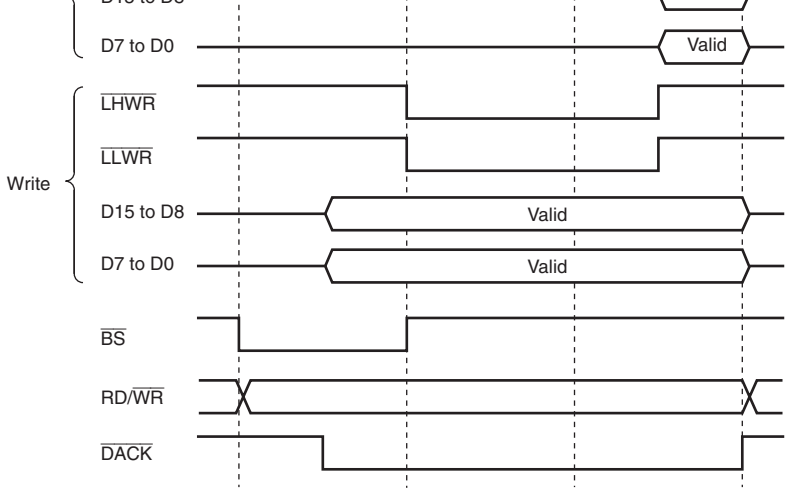
Notes: 1. n = 0 to 7
2. When RDNn = 0
3. When DKC = 0

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



Notes: 1. n = 0 to 7
2. When RDNn = 0
3. When DKC = 0

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



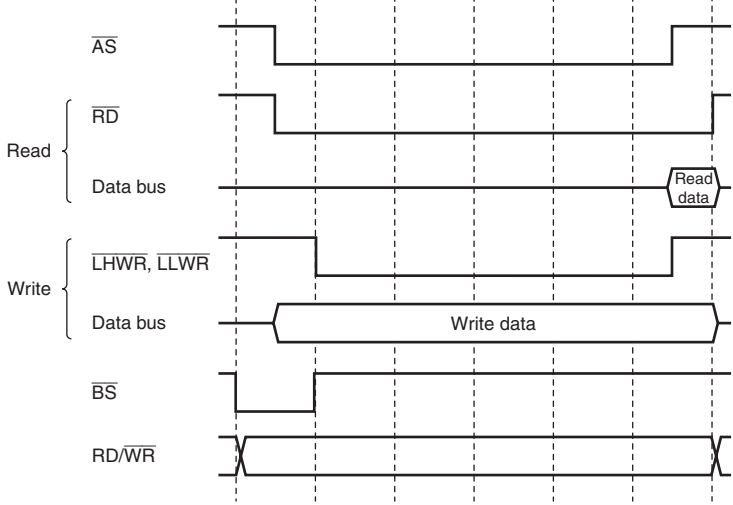
Notes: 1. n = 0 to 7
2. When RDNn = 0
3. When DKC = 0

Figure 6.19 16-Bit 3-State Access Space Bus Timing (Word Access for Even Address)

(2) Pin Wait Insertion

For 3-state access space, when the WAITE bit in BCR1 is set to 1 and the corresponding
is set to 1, wait input by means of the $\overline{\text{WAIT}}$ pin is enabled. When the external address sp
accessed in this state, a program wait (Tw) is first inserted according to the WTCRA and
settings. If the $\overline{\text{WAIT}}$ pin is low at the falling edge of Bϕ in the last T2 or Tpw cycle, ano
cycle is inserted until the $\overline{\text{WAIT}}$ pin is brought high. The pin wait insertion is effective w
Tw cycles are inserted to seven cycles or more, or when the number of Tw cycles to be in
changed according to the external devices. The WAITE bit is common to all areas. For de
ICR, see section 9, I/O Ports.

Figure 6.20 shows an example of wait cycle insertion timing. After a reset, the 3-state acc
specified, the program wait is inserted for seven cycles, and the $\overline{\text{WAIT}}$ input is disabled.



Notes: 1. Upward arrows indicate the timing of $\overline{\text{WAIT}}$ pin sampling.
2. n = 0 to 7
3. When RDNn = 0

Figure 6.20 Example of Wait Cycle Insertion Timing

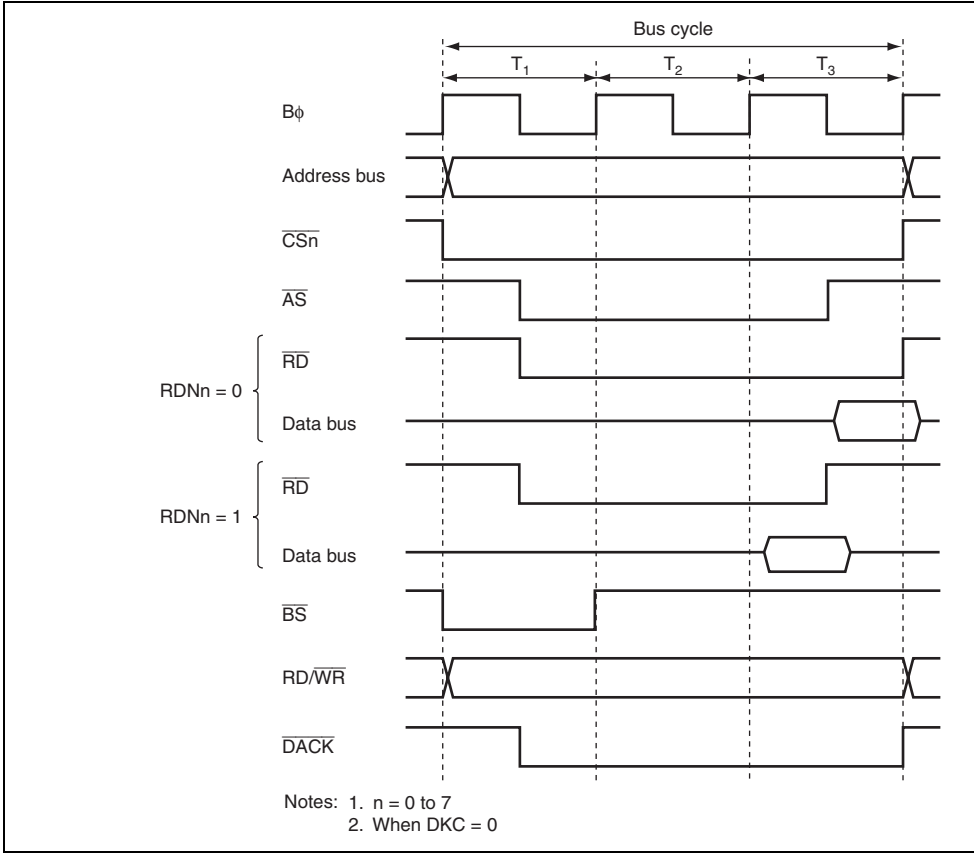
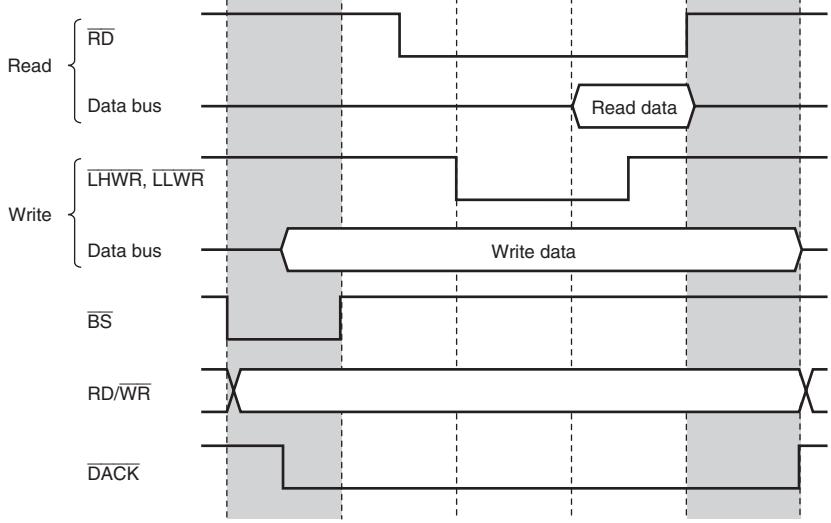


Figure 6.21 Example of Read Strobe Timing

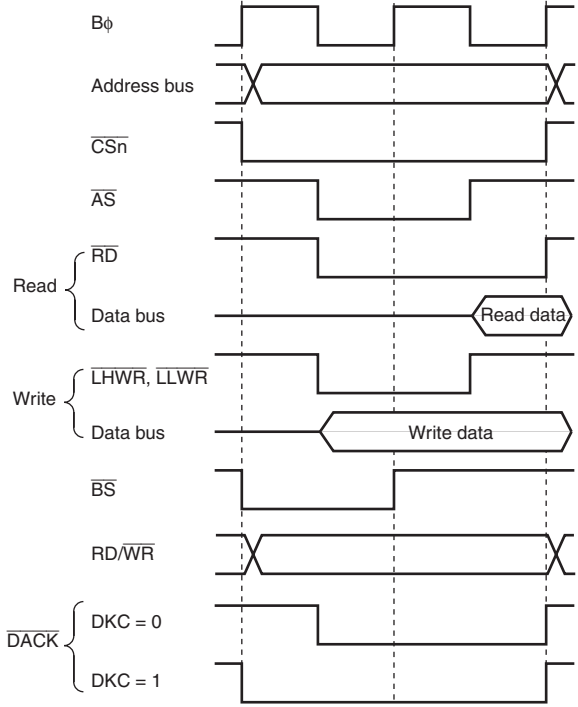
3-state access space.

Both extension cycle Th inserted before the basic bus cycle and extension cycle Tt inser
the basic bus cycle, or only one of these, can be specified for individual areas. Insertion
insertion can be specified for the Th cycle with the upper eight bits (CSXH7 to CSXH0)
CSACR, and for the Tt cycle with the lower eight bits (CSXT7 to CSXT0).



Notes: 1. n = 0 to 7
2. When DKC = 0

Figure 6.22 Example of Timing when Chip Select Assertion Period is Extend



Notes: 1. n = 7 to 0
2. RDNn = 0

Figure 6.23 DACK Signal Output Timing

6.7.1 Byte Control SRAM Space Setting

Byte control SRAM interface can be specified for areas 0 to 7. Each area can be specified control SRAM interface by setting bits BCSELn (n = 0 to 7) in SRAMCR. For the area sp as burst ROM interface or address/data multiplexed I/O interface, the SRAMCR setting is and byte control SRAM interface cannot be used.

6.7.2 Data Bus

The bus width of the byte control SRAM space can be specified as 16-bit byte control SR space according to bits ABWHn and ABWLn (n = 0 to 7) in ABWCR. The area specified access space cannot be specified as the byte control SRAM space.

For the 16-bit byte control SRAM space, data bus (D15 to D0) is valid.

Access size and data alignment are the same as the basic bus interface. For details, see se 6.5.6, Endian and Data Alignment.

AS/AH	AS	Address strobe	Output	Strobe signal indicating that tr output on the address bus is v a basic bus interface space or control SRAM space is access
$\overline{CSn}$	$\overline{CSn}$	Chip select	Output	Strobe signal indicating that a selected
$\overline{RD}$	$\overline{RD}$	Read strobe	Output	Output enable for the SRAM w byte control SRAM space is a
RD/$\overline{WR}$	RD/$\overline{WR}$	Read/write	Output	Write enable signal for the SR the byte control SRAM space accessed
$\overline{LHWR}$/$\overline{LUB}$	$\overline{LUB}$	Lower-upper byte select	Output	Upper byte select when the 16 control SRAM space is access
$\overline{LLWR}$/$\overline{LLB}$	$\overline{LLB}$	Lower-lower byte select	Output	Lower byte select when the 16 control SRAM space is access
$\overline{WAIT}$	$\overline{WAIT}$	Wait	Input	Wait request signal used whe external address space is acc
A23 to A0	A23 to A0	Address pin	Output	Address output pin
D15 to D0	D15 to D0	Data pin	Input/ output	Data input/output pin

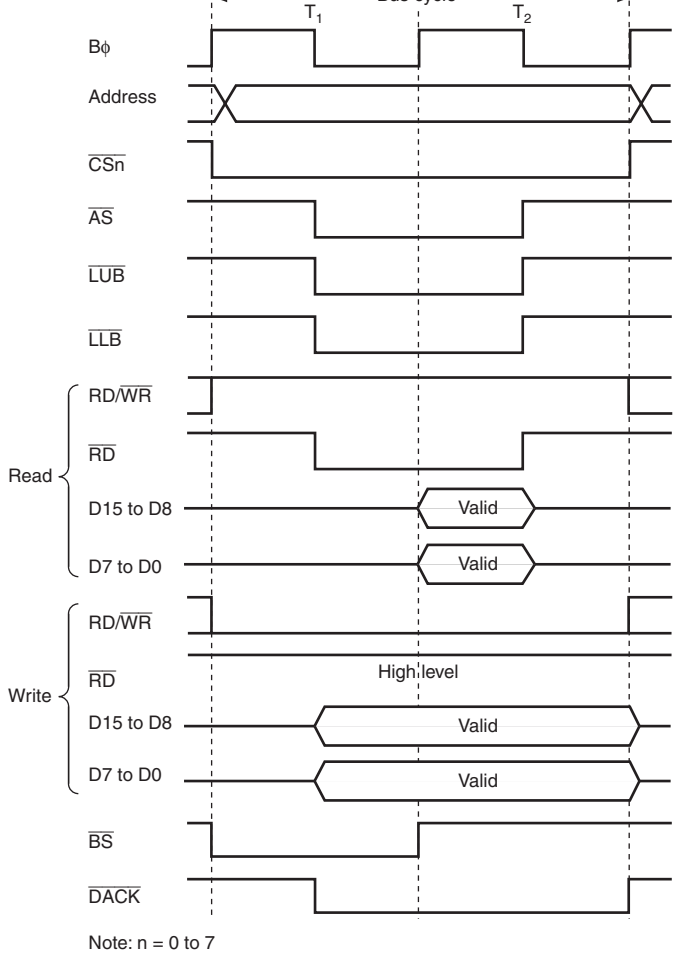


Figure 6.24 16-Bit 2-State Access Space Bus Timing

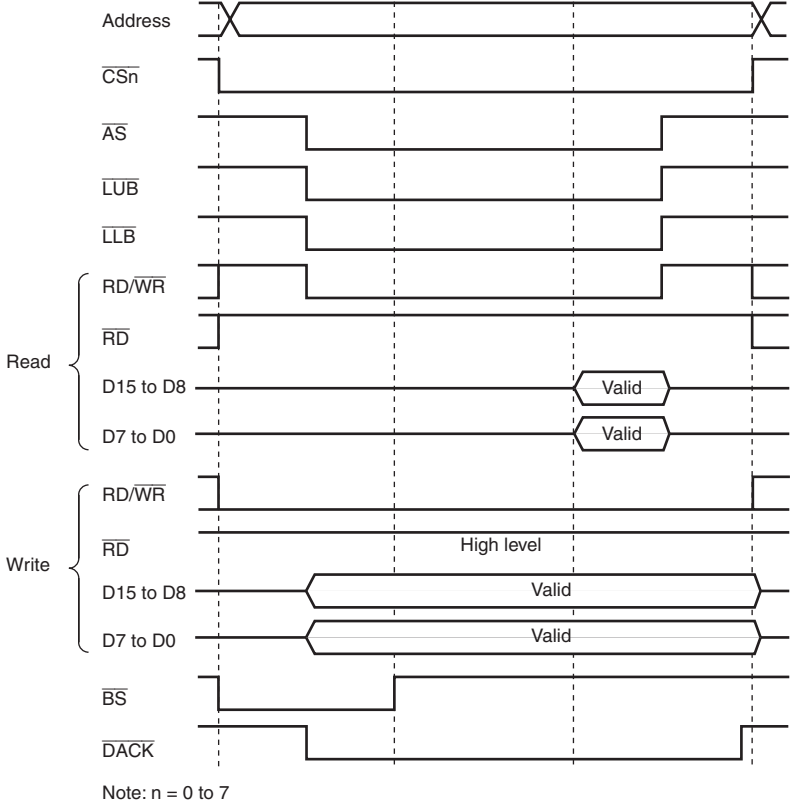
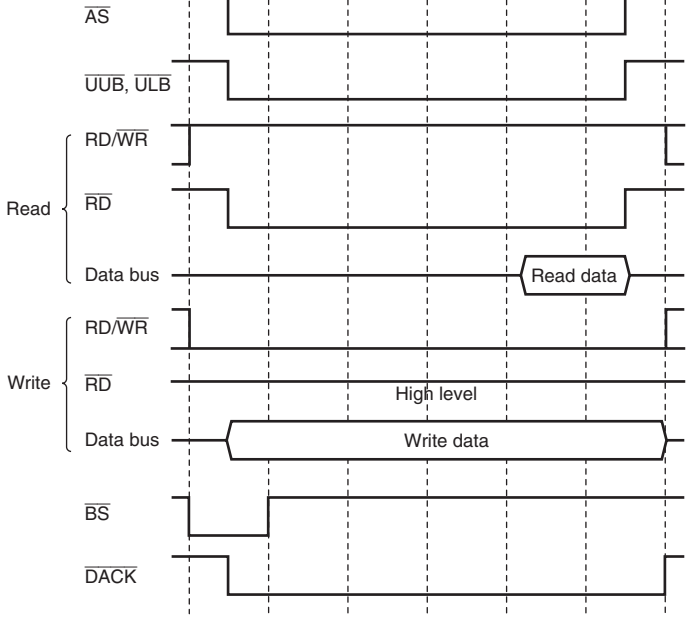


Figure 6.25 16-Bit 3-State Access Space Bus Timing

For 3-state access space, when the WAITE bit in BCR1 is set to 1, the corresponding DD
cleared to 0, and the ICR bit is set to 1, wait input by means of the $\overline{\text{WAIT}}$ pin is enabled.
details on DDR and ICR, refer to section 9, I/O Ports.

Figure 6.26 shows an example of wait cycle insertion timing.



Notes: 1. Upward arrows indicate the timing of $\overline{\text{WAIT}}$ pin sampling.
2. n = 0 to 7

Figure 6.26 Example of Wait Cycle Insertion Timing

In the byte control SRAM interface, the extension cycles can be inserted before and after cycle in the same way as the basic bus interface. For details, refer to section 6.6.6, Extens Chip Select ($\overline{CS}$) Assertion Period.

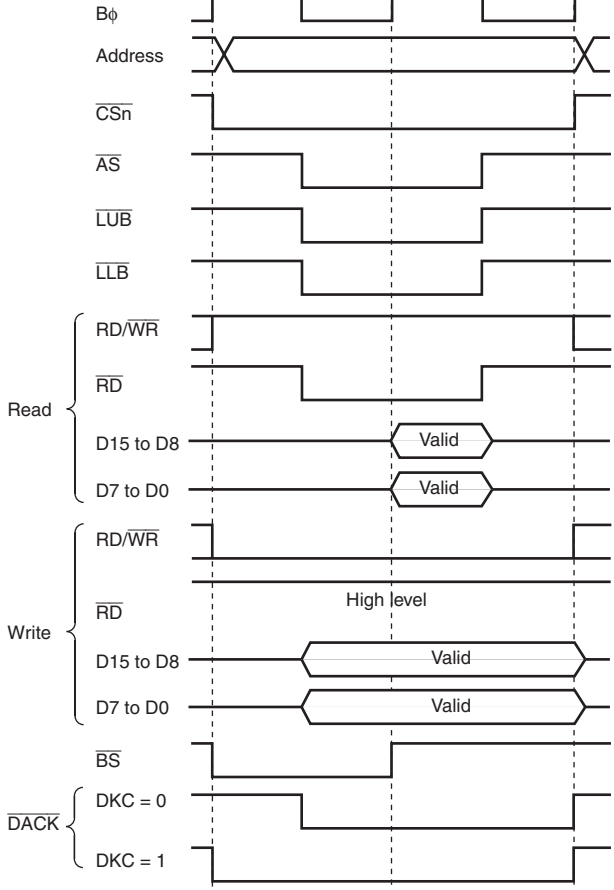


Figure 6.27 $\overline{\text{DACK}}$ Signal Output Timing

Settings can be made independently for area 0 and area 1.

In the burst ROM interface, the burst access covers only CPU read accesses. Other access performed with the similar method to the basic bus interface.

6.8.1 Burst ROM Space Setting

Burst ROM interface can be specified for areas 0 and 1. Areas 0 and 1 can be specified as ROM space by setting bits BSRMn (n = 0, 1) in BROMCR.

6.8.2 Data Bus

The bus width of the burst ROM space can be specified as 8-bit or 16-bit burst ROM inte space according to the ABWHn and ABWLn bits (n = 0, 1) in ABWCR.

For the 8-bit bus width, data bus (D7 to D0) is valid. For the 16-bit bus width, data bus (D D0) is valid.

Access size and data alignment are the same as the basic bus interface. For details, see se 6.5.6, Endian and Data Alignment.

Read strobe	$\overline{RD}$	Output	Strobe signal indicating the read access
Read/write	RD/$\overline{WR}$	Output	Signal indicating the data bus input or outp
Low-high write	$\overline{LHWR}$	Output	Strobe signal indicating that the upper byte D8) is valid during write access
Low-low write	$\overline{LLWR}$	Output	Strobe signal indicating that the lower byte is valid during write access
Chip select 0, 1	$\overline{CS0}$, $\overline{CS1}$	Output	Strobe signal indicating that the area is sel
Wait	$\overline{WAIT}$	Input	Wait request signal used when an external space is accessed

The basic access timing for burst ROM space is shown in figures 6.28 and 6.29.

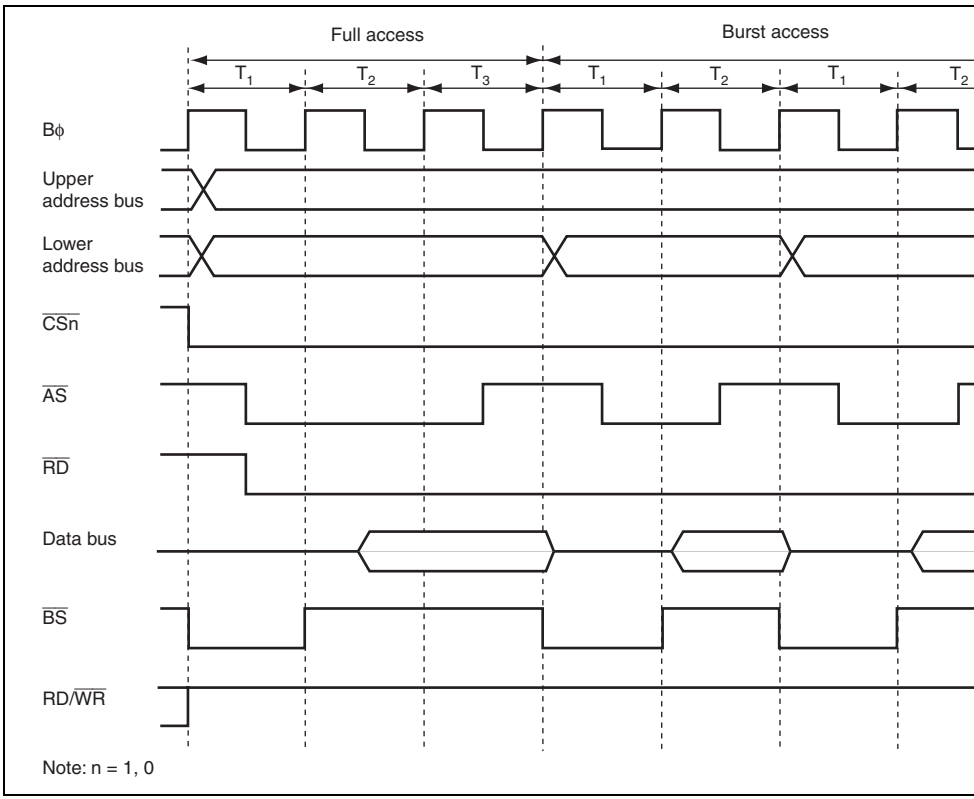
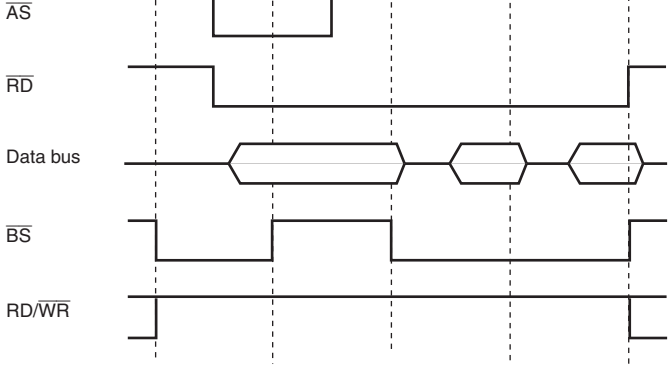


Figure 6.28 Example of Burst ROM Access Timing (ASTn = 1, Two Burst Cy



Note: n = 1, 0

Figure 6.29 Example of Burst ROM Access Timing (ASTn = 0, One Burst C

The read strobe negation timing is the same timing as when RDNn = 0 in the basic bus in

6.8.7 Extension of Chip Select ($\overline{CS}$) Assertion Period

In the burst ROM interface, the extension cycles can be inserted in the same way as the b interface.

For the burst ROM space, the burst access can be enabled only in read access by the CPU case, the setting of the corresponding CSXTn bit in CSACR is ignored and an extension c be inserted only before the full access cycle. Note that no extension cycle can be inserted after the burst access cycles.

In accesses other than read accesses by the CPU, the burst ROM space is equivalent to th bus interface space. Accordingly, extension cycles can be inserted before and after the bu cycles.

specified as the address/data multiplexed I/O space by setting bits MPXEn (n = 3 to 7) i
MPXCR.

6.9.2 Address/Data Multiplex

In the address/data multiplexed I/O space, data bus is multiplexed with address bus. Tab
shows the relationship between the bus width and address output.

Table 6.18 Address/Data Multiplex

Bus Width	Cycle	Data Pins													
		PI7	PI6	PI5	PI4	PI3	PI2	PI1	PI0	PH7	PH6	PH5	PH4	PH3	PH2
8 bits	Address	-	-	-	-	-	-	-	-	A7	A6	A5	A4	A3	A2
	Data	-	-	-	-	-	-	-	-	D7	D6	D5	D4	D3	D2
16 bits	Address	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2
	Data	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2

6.9.3 Data Bus

The bus width of the address/data multiplexed I/O space can be specified for either 8-bit
space or 16-bit access space by the ABWHn and ABWLn bits (n = 3 to 7) in ABWCR.

For the 8-bit access space, D7 to D0 are valid for both address and data. For 16-bit acces
D15 to D0 are valid for both address and data. If the address/data multiplexed I/O space
accessed, the corresponding address will be output to the address bus.

For details on access size and data alignment, see section 6.5.6, Endian and Data Alignm

				as the address/data multiplexed I/O spa
$\overline{AS}$/$\overline{AH}$	$\overline{AH}$*	Address hold	Output	Signal to hold an address when the address/data multiplexed I/O space is s
$\overline{RD}$	$\overline{RD}$	Read strobe	Output	Signal indicating that the address/data multiplexed I/O space is being read
$\overline{LHWR}$/$\overline{LUB}$	$\overline{LHWR}$	Low-high write	Output	Strobe signal indicating that the upper b to D8) is valid when the address/data multiplexed I/O space is written
$\overline{LLWR}$/$\overline{LLB}$	$\overline{LLWR}$	Low-low write	Output	Strobe signal indicating that the lower b to D0) is valid when the address/data multiplexed I/O space is written
D15 to D0	D15 to D0	Address/dat a	Input/ output	Address and data multiplexed pins for t address/data multiplexed I/O space. Only D7 to D0 are valid when the 8-bit s specified. D15 to D0 are valid when the space is specified.
A23 to A0	A23 to A0	Address	Output	Address output pin
$\overline{WAIT}$	$\overline{WAIT}$	Wait	Input	Wait request signal used when the exte address space is accessed
$\overline{BS}$	$\overline{BS}$	Bus cycle start	Output	Signal to indicate the bus cycle start
RD/$\overline{WR}$	RD/$\overline{WR}$	Read/write	Output	Signal indicating the data bus input or o direction

Note: * The $\overline{AH}$ output is multiplexed with the $\overline{AS}$ output. At the timing that an area is s as address/data multiplexed I/O, this pin starts to function as the $\overline{AH}$ output me that this pin cannot be used as the $\overline{AS}$ output. At this time, when other areas s basic bus interface is accessed, this pin does not function as the $\overline{AS}$ output. U area is specified as address/data multiplexed I/O, be aware that this pin functio the $\overline{AS}$ output.

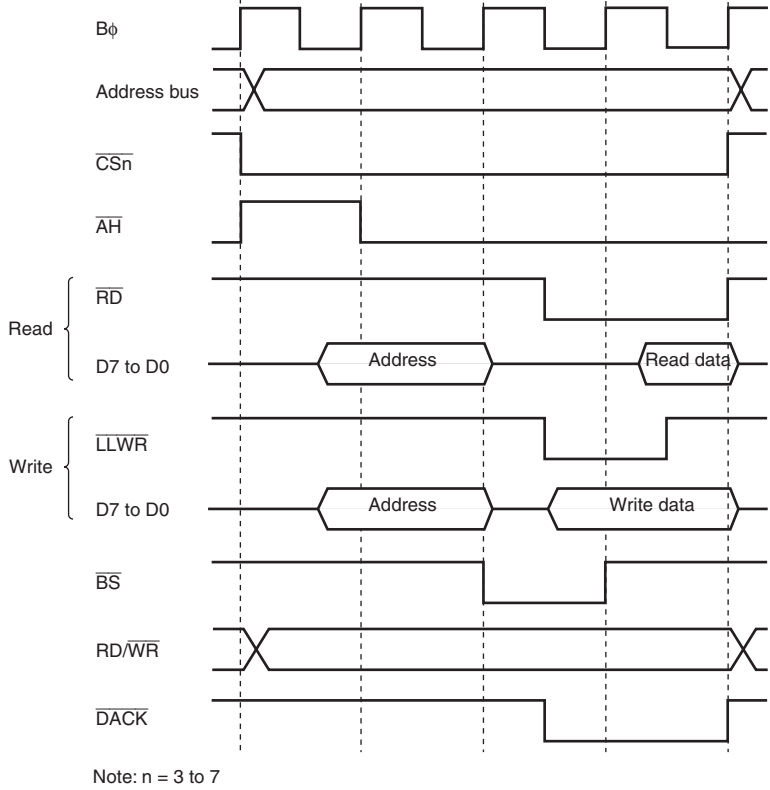


Figure 6.30 8-Bit Access Space Access Timing (ABWHn = 1, ABWLn = 1

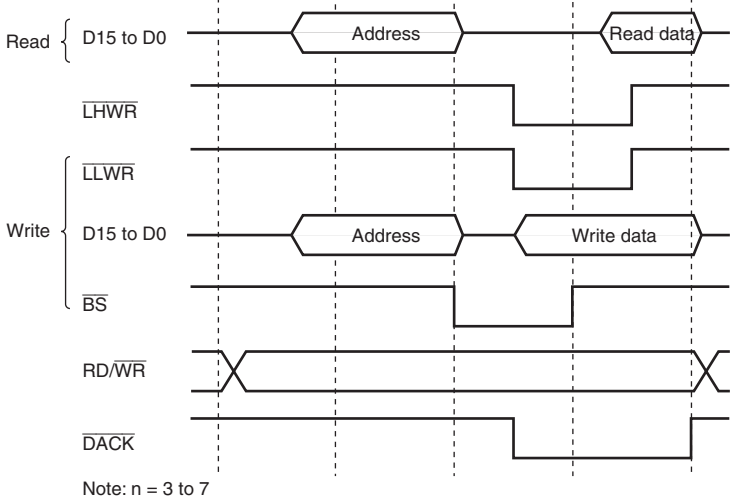


Figure 6.31 16-Bit Access Space Access Timing (ABWHn = 0, ABWLn = 1

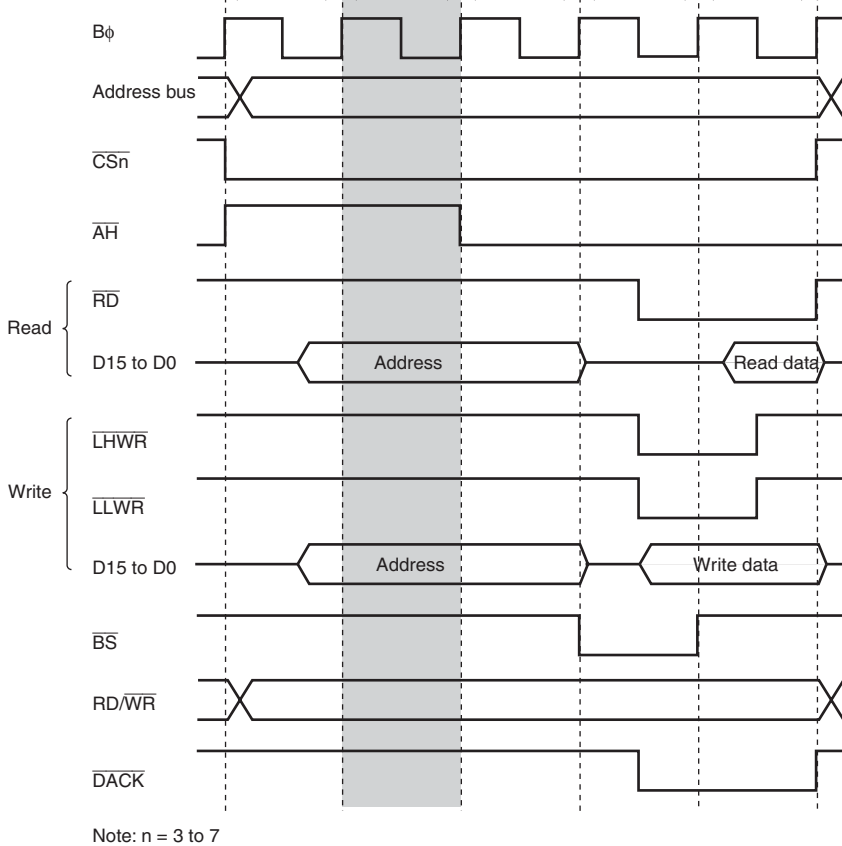


Figure 6.32 Access Timing of 3 Address Cycles (ADDEX = 1)

in the same way as in basic bus interface. For details, refer to section 6.6.5, Read Strobe (
Timing.

Figure 6.33 shows an example when the read strobe timing is modified.

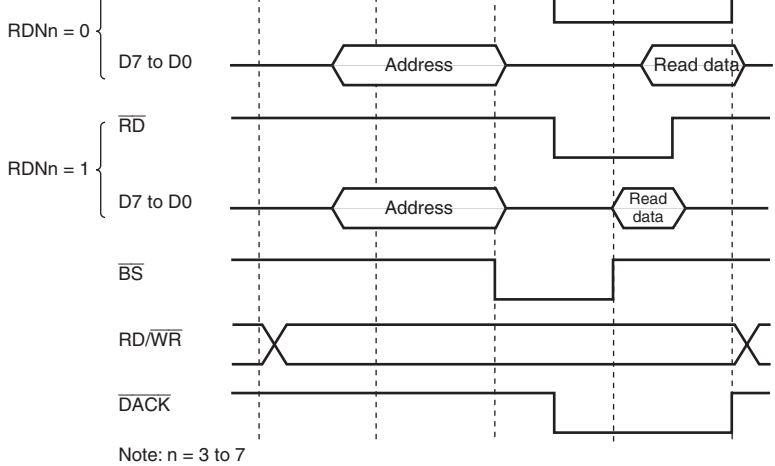


Figure 6.33 Read Strobe Timing

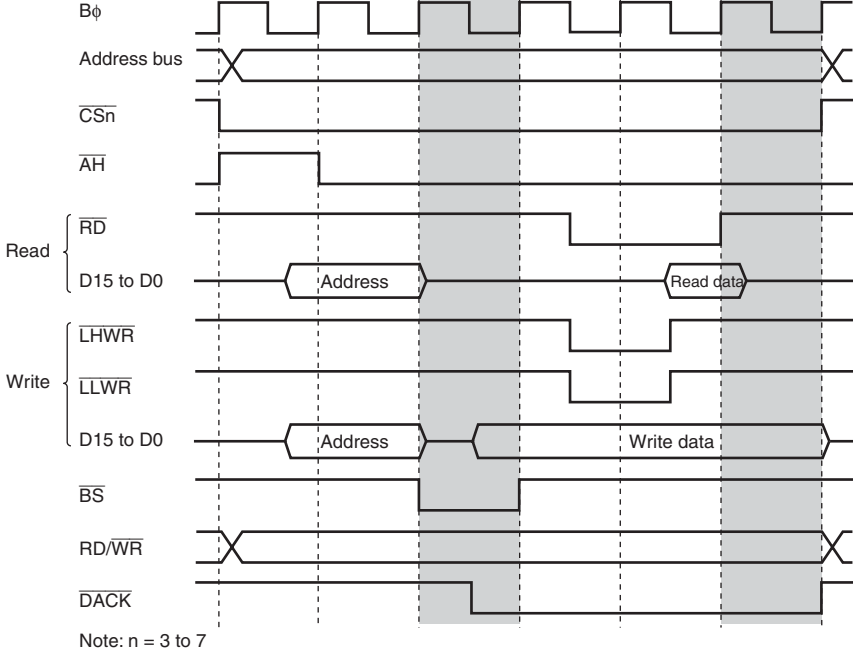
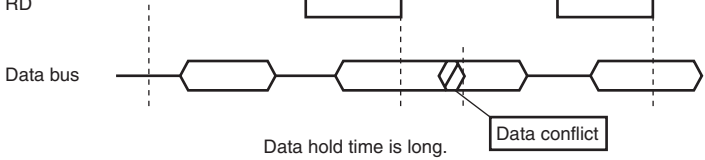


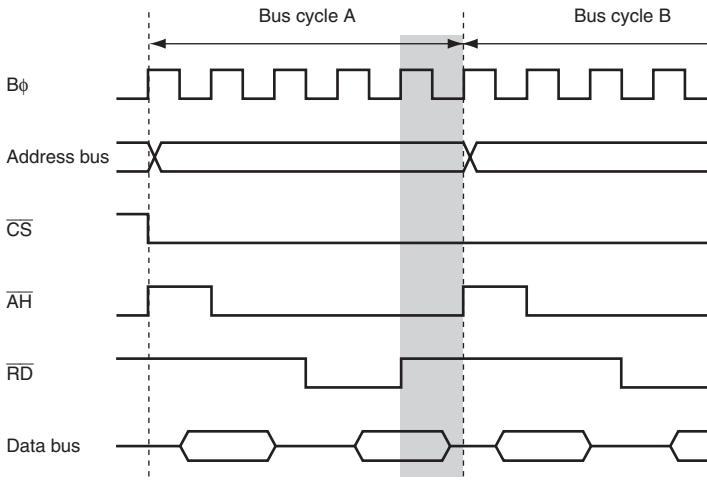
Figure 6.34 Chip Select ($\overline{CS}$) Assertion Period Extension Timing in Data Cy

When consecutively reading from the same area connected to a peripheral LSI whose dat time is long, data outputs from the peripheral LSI and this LSI may conflict. Inserting the select assertion period extension cycle after the access cycle can avoid the data conflict.

Figure 6.35 shows an example of the operation. In the figure, both bus cycles A and B are access cycles to the address/data multiplexed I/O space. An example of the data conflict i in (a), and an example of avoiding the data conflict by the $\overline{CS}$ assertion period extension (b).



(a) Without $\overline{CS}$ assertion period extension cycle (CSXTn = 0)



(b) With $\overline{CS}$ assertion period extension cycle (CSXTn = 1)

Figure 6.35 Consecutive Read Accesses to Same Area (Address/Data Multiplexed I/O Space)

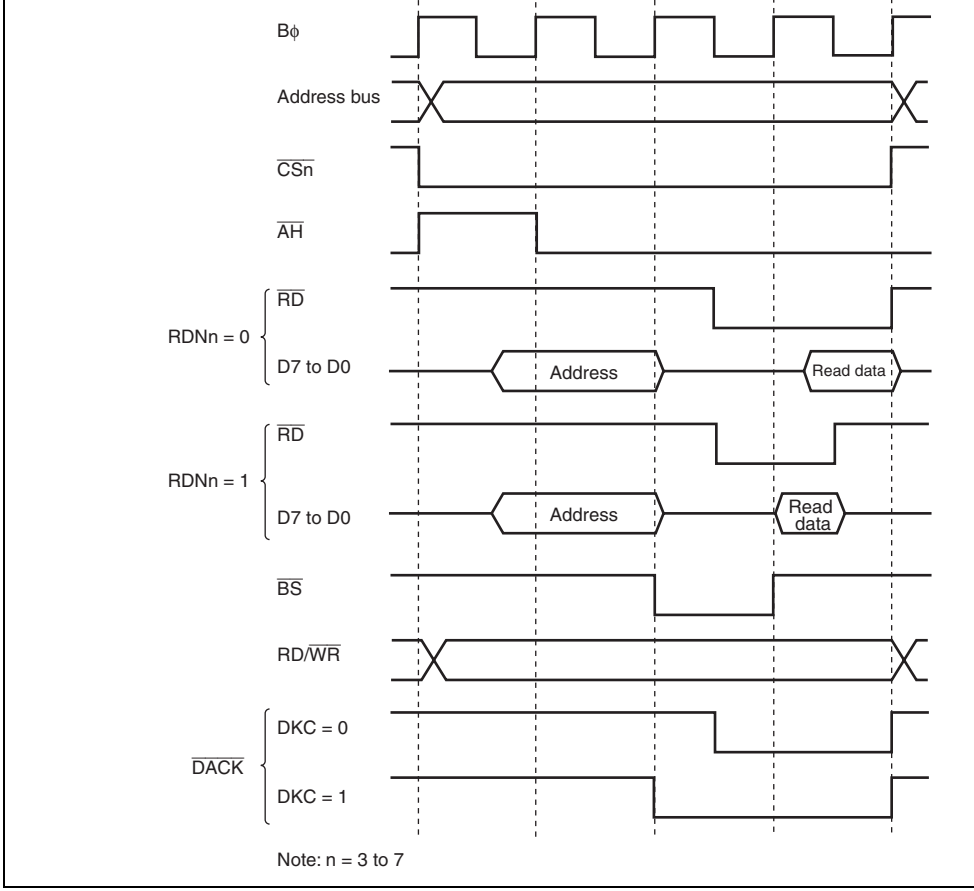


Figure 6.36 $\overline{\text{DACK}}$ Signal Output Timing

and write and previously accessed area.

1. When read cycles of different areas in the external address space occur consecutivel
2. When an external write cycle occurs immediately after an external read cycle
3. When an external read cycle occurs immediately after an external write cycle
4. When an external access occurs immediately after a DMAC single address transfer (
   cycle)

Up to four idle cycles can be inserted under the conditions shown above. The number of
cycles to be inserted should be specified to prevent data conflicts between the output dat
previously accessed device and data from a subsequently accessed device.

Under conditions 1 and 2, which are the conditions to insert idle cycles after read, the nu
idle cycles can be selected from setting A specified by bits IDLCA1 and IDLCA0 in IDL
setting B specified by bits IDLCB1 and IDLCB0 in IDLCR: Setting A can be selected fr
four cycles, and setting B can be selected from one or two to four cycles. Setting A or B
specified for each area by setting bits IDLSEL7 to IDLSEL0 in IDLCR. Note that bits ID
to IDLSEL0 correspond to the previously accessed area of the consecutive accesses.

The number of idle cycles to be inserted under conditions 3 and 4, which are conditions
idle cycles after write, can be determined by setting A as described above.

After the reset release, IDLCR is initialized to four idle cycle insertion under all conditio
shown above.

Table 6.20 shows the correspondence between conditions 1 to 4 and number of idle cycl
inserted for each area. Table 6.21 shows the correspondence between the number of idle
be inserted specified by settings A and B, and number of cycles to be inserted.

			1	B	B	B	B	B	B	
Read after write	2	0	—	Invalid						
		1		A						
External access after single address transfer	3	0	—	Invalid						
		1		A						

[Legend]

A: Number of idle cycle insertion A is selected.

B: Number of idle cycle insertion B is selected.

Invalid: No idle cycle is inserted for the corresponding condition.

Table 6.21 Number of Idle Cycle Insertions

Bit Settings				
A		B		
IDLCA1	IDLCA0	IDLCB1	IDLCB0	Number of Cyc
—	—	0	0	0
0	0	—	—	1
0	1	0	1	2
1	0	1	0	3
1	1	1	1	4

and a data conflict is prevented.

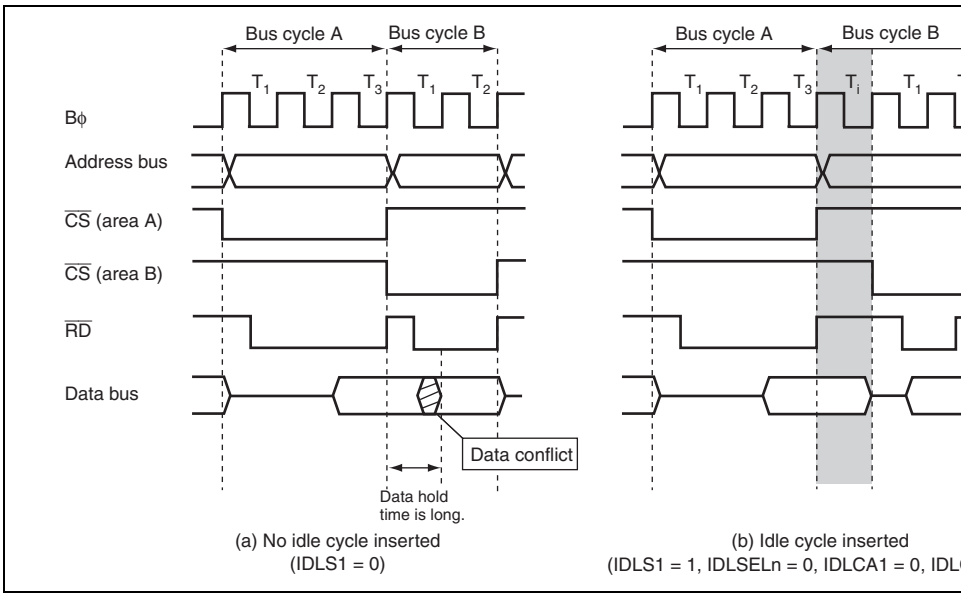


Figure 6.37 Example of Idle Cycle Operation (Consecutive Reads in Different

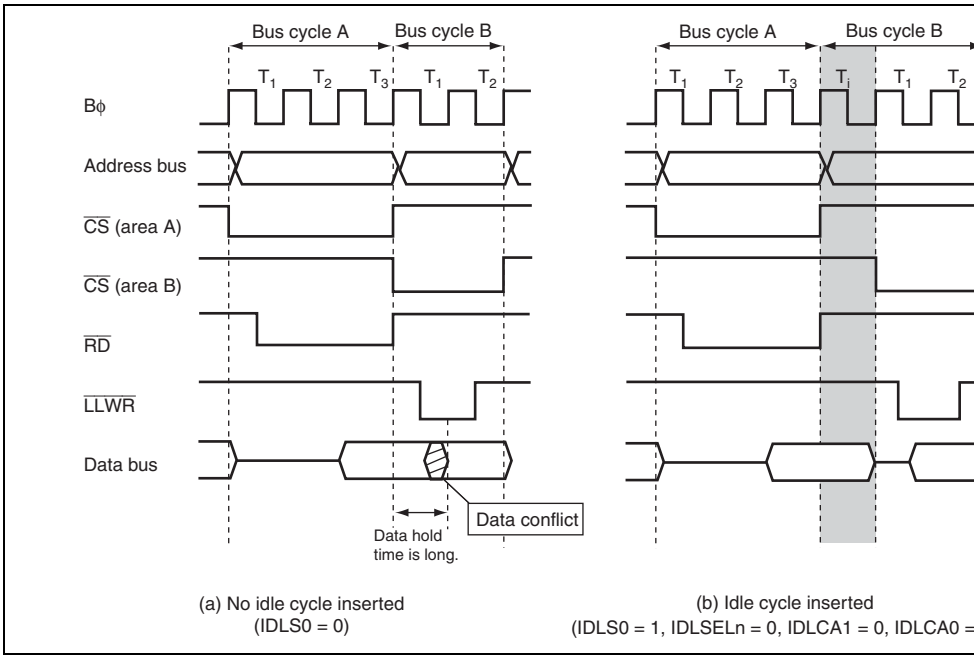


Figure 6.38 Example of Idle Cycle Operation (Write after Read)

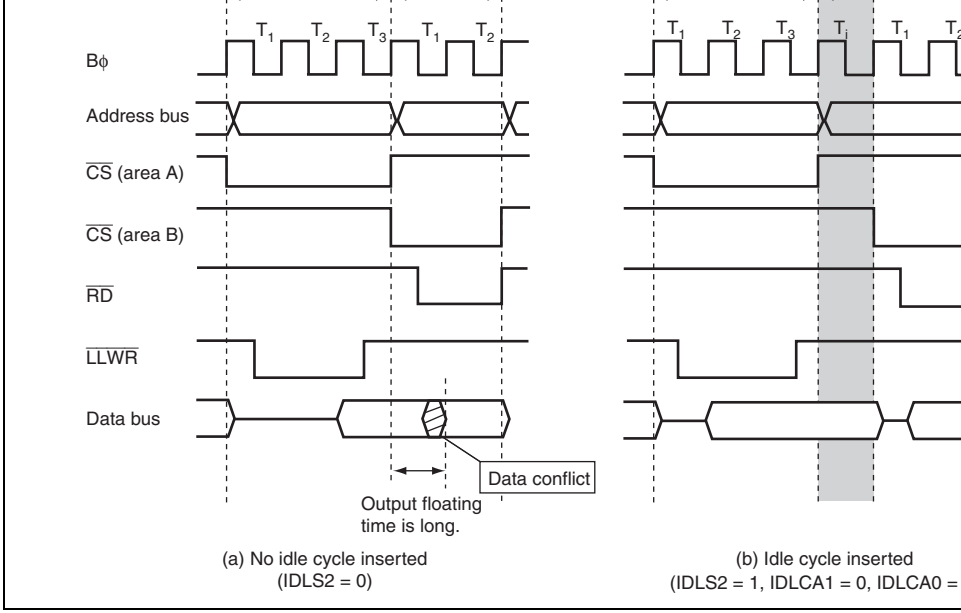


Figure 6.39 Example of Idle Cycle Operation (Read after Write)

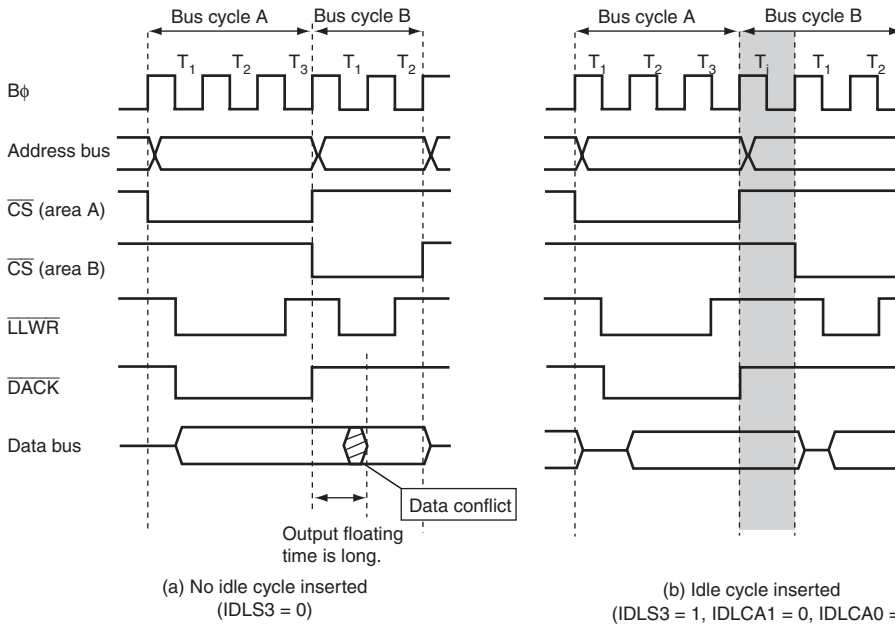


Figure 6.40 Example of Idle Cycle Operation (Write after Single Address Transfe

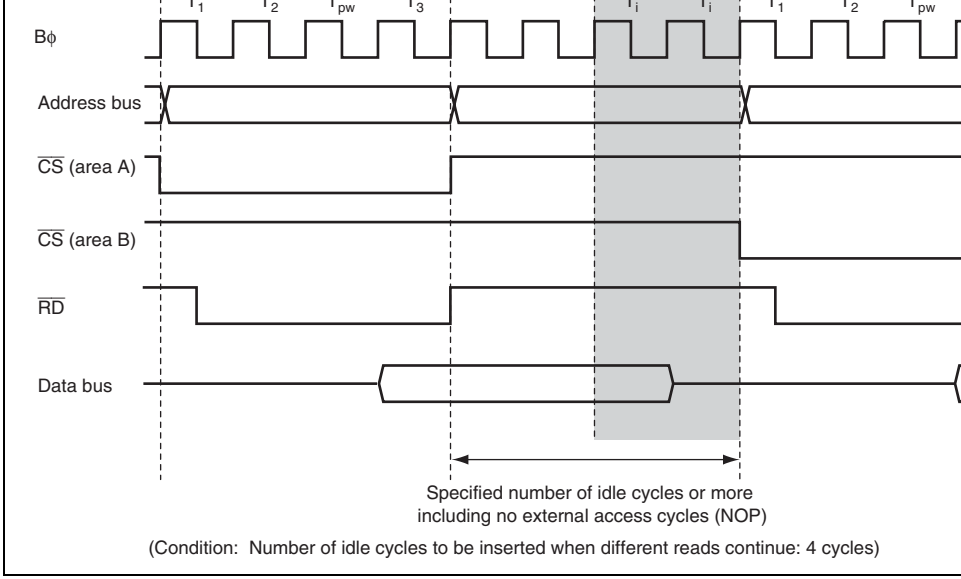


Figure 6.41 Idle Cycle Insertion Example

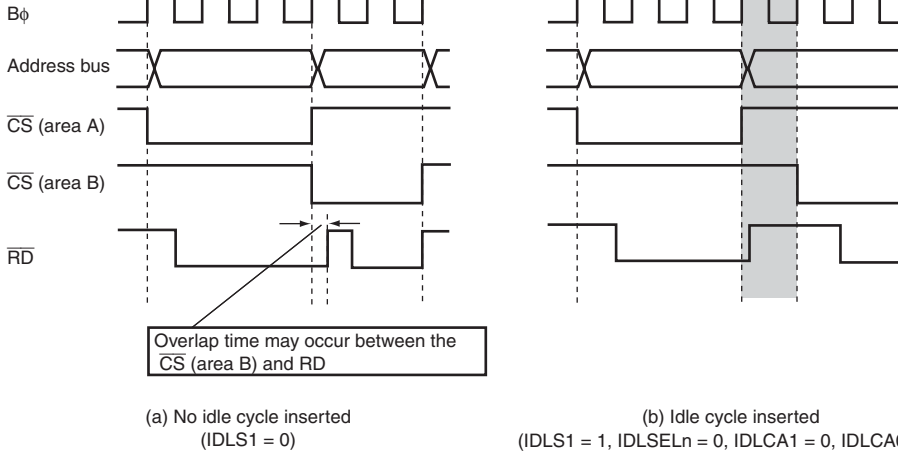


Figure 6.42 Relationship between Chip Select ($\overline{CS}$) and Read ($\overline{RD}$)

									0	1	2 cycle i
									1	0	3 cycles
									1	1	4 cycles
Normal space read	Normal space write	—	—	—	0	—	—	—	—	—	Disablec
		—	—	—	1	0	0	0	—	—	1 cycle i
							0	1			2 cycles
							1	0			3 cycles
							1	1			4 cycles
						1	—	—	0	0	0 cycle i
									0	1	2 cycle i
									1	0	3 cycles
									1	1	4 cycles
Normal space write	Normal space read	—	0	—	—	—	—	—	—	—	Disablec
		—	1	—	—	—	0	0	—	—	1 cycle i
							0	1			2 cycles
							1	0			3 cycles
							1	1			4 cycles
Single address transfer write	Normal space read	0	—	—	—	—	—	—	—	—	Disablec
		1	—	—	—	—	0	0	—	—	1 cycle i
							0	1			2 cycles
							1	0			3 cycles
							1	1			4 cycles

$\overline{AS}$	High
$\overline{RD}$	High
$\overline{BS}$	High
RD/$\overline{WR}$	High
$\overline{AH}$	low
$\overline{LHWR}$, $\overline{LLWR}$	High
$\overline{DACKn}$ (n = 3 to 0)	High

In external extended mode, when the BRLE bit in BCR1 is set to 1 and the ICR bits for
corresponding pin are set to 1, the bus can be released to the external. Driving the $\overline{\text{BREQ}}$
issues an external bus request to this LSI. When the $\overline{\text{BREQ}}$ pin is sampled, at the prescri
timing, the $\overline{\text{BACK}}$ pin is driven low, and the address bus, data bus, and bus control signa
placed in the high-impedance state, establishing the external bus released state. For deta
DDR and ICR, see section 9, I/O Ports.

In the external bus released state, the CPU, DTC, and DMAC can access the internal spa
the internal bus. When the CPU, DTC, or DMAC attempts to access the external address
temporarily defers initiation of the bus cycle, and waits for the bus request from the exte
master to be canceled.

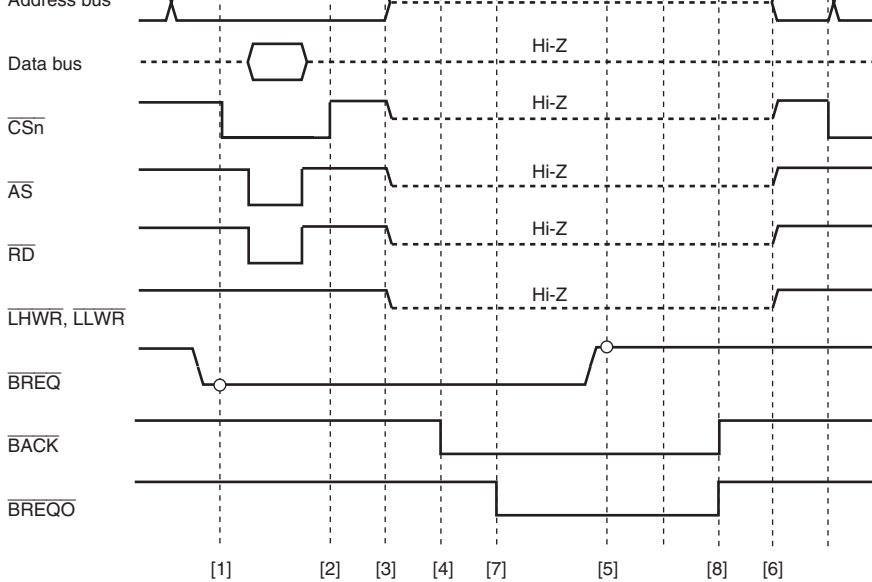
If the BREQOE bit in BCR1 is set to 1, the $\overline{\text{BREQO}}$ pin can be driven low when any of
following requests are issued, to request cancellation of the bus request externally.

- When the CPU, DTC, or DMAC attempts to access the external address space
- When a SLEEP instruction is executed to place the chip in software standby mode o
  module-clock-stop mode
- When SCKCR is written to for setting the clock frequency

If an external bus release request and external access occur simultaneously, the priority i
follows:

(High) External bus release > External access by CPU, DTC, or DMAC (Low)

CSn (n = 7 to 0)	High impedance
$\overline{AS}$	High impedance
$\overline{AH}$	High impedance
RD/$\overline{WR}$	High impedance
$\overline{RD}$	High impedance
$\overline{LUB}$, $\overline{LLB}$	High impedance
$\overline{LHWR}$, $\overline{LLWR}$	High impedance
$\overline{DACKn}$ (n = 3 to 0)	High level



[1] A low level of the $\overline{\text{BREQ}}$ signal is sampled at the rising edge of the Bϕ signal.
[2] The bus control signals are driven high at the end of the external space access cycle. It takes two cycles more after the low level of the $\overline{\text{BREQ}}$ signal is sampled.
[3] The $\overline{\text{BACK}}$ signal is driven low, releasing bus to the external bus master.
[4] The $\overline{\text{BREQ}}$ signal state sampling is continued in the external bus released state.
[5] A high level of the $\overline{\text{BREQ}}$ signal is sampled.
[6] The external bus released cycles are ended one cycle after the $\overline{\text{BREQ}}$ signal is driven high.
[7] When the external space is accessed by an internal bus master during external bus released while the B bit is set to 1, the $\overline{\text{BREQO}}$ signal goes low.
[8] Normally the $\overline{\text{BREQO}}$ signal goes high at the rising edge of the $\overline{\text{BACK}}$ signal.

Figure 6.43 Bus Released State Transition Timing

Table 6.25 Number of Access Cycles for On-Chip Memory Spaces

Access Space	Access	Number of Access C
On-chip ROM space	Read	One Iϕ cycle
	Write	Six Iϕ cycles
On-chip RAM space	Read	One Iϕ cycle
	Write	One Iϕ cycle

In access to the registers for on-chip peripheral modules, the number of access cycles diff
according to the register to be accessed. When the dividing ratio of the operating clock of
master and that of a peripheral module is 1 : n, synchronization cycles using a clock divid
to n-1 are inserted for register access in the same way as for external bus clock division.

Table 6.26 lists the number of access cycles for registers of on-chip peripheral modules.

Table 6.26 Number of Access Cycles for Registers of On-Chip Peripheral Modules

	Number of Cycles		
Module to be Accessed	Read	Write	Write Data Buffer F
DMAC registers	2Iϕ		Disabled
MCU operating mode, clock pulse generator, power-down control registers, interrupt controller, bus controller, and DTC registers	2Iϕ	3Iϕ	Disabled
I/O port PFCR registers and WDT registers	2Pϕ	3Pϕ	Disabled
I/O port registers other than PFCR, TPU, PPG, TMR, SCI, A/D, and D/A registers	2Pϕ		Enabled

for two cycles or longer, and there is an internal access next, an external write only is ex
the first two cycles. However, from the next cycle onward, internal accesses (on-chip me
internal I/O register read/write) and the external address space write rather than waiting
ends are executed in parallel.

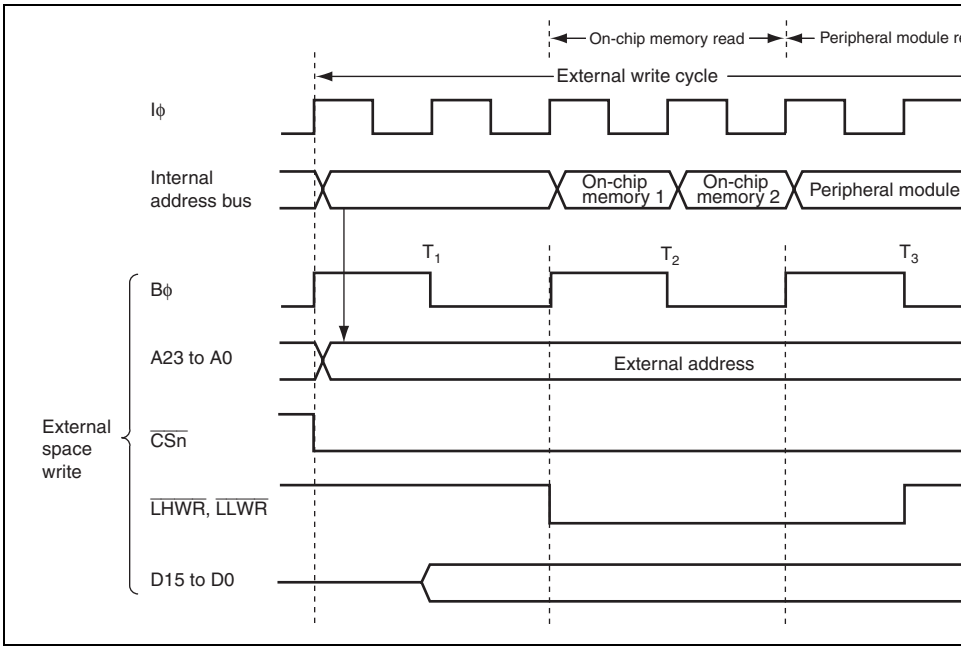


Figure 6.44 Example of Timing when Write Data Buffer Function is Use

performed in the first two cycles. However, from the next cycle onward an internal memo
external access and internal I/O register write are executed in parallel rather than waiting
ends.

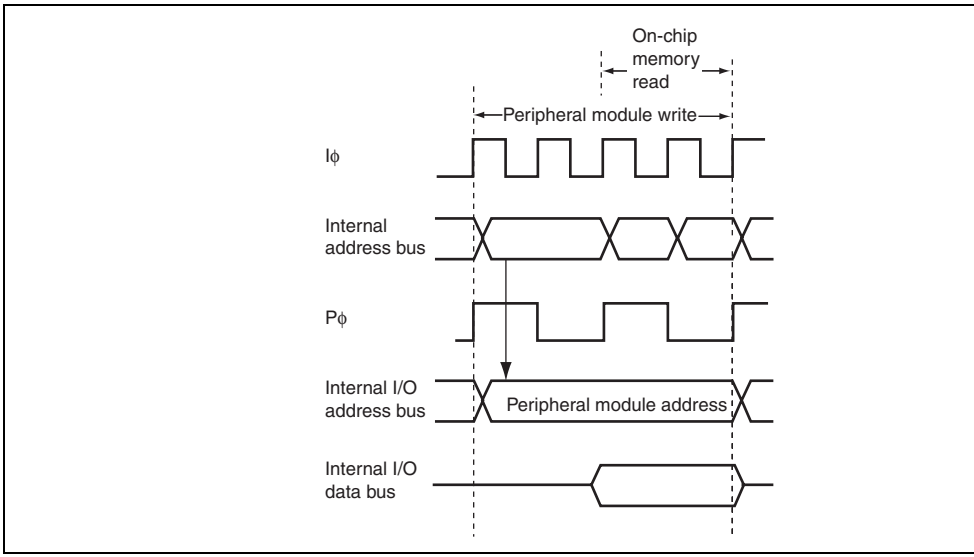


Figure 6.45 Example of Timing when Peripheral Module Write Data Buffer Function is Used

6.14.1 Operation

The bus arbiter detects the bus masters' bus request signals, and if the bus is requested, s
request acknowledge signal to the bus master. If there are bus requests from more than o
master, the bus request acknowledge signal is sent to the one with the highest priority. W
master receives the bus request acknowledge signal, it takes possession of the bus until t
is canceled.

The priority of the internal bus arbitration:

(High) DMAC > DTC > CPU (Low)

The priority of the external bus arbitration:

(High) External bus release request > External access by the CPU, DTC, and DM

If the DMAC or DTC accesses continue, the CPU can be given priority over the DMAC
to execute the bus cycles alternatively between them by setting the IBCCS bit in BCR2.
case, the priority between the DMAC and DTC does not change.

An internal bus access by the CPU, DTC, or DMAC and an external bus access by an ex
release request can be executed in parallel.

The timing for transfer of the bus is at the end of the bus cycle. In sleep mode, the bus is transferred synchronously with the clock.

Note, however, that the bus cannot be transferred in the following cases.

- The word or longword access is performed in some divisions.
- Stack handling is performed in multiple bus cycles.
- Transfer data read or write by memory transfer instructions, block transfer instruction instruction.

  (In the block transfer instructions, the bus can be transferred in the write cycle and the following transfer data read cycle.)
- From the target read to write in the bit manipulation instructions or memory operation instructions.

  (In an instruction that performs no write operation according to the instruction conditi a cycle corresponding the write cycle)

(2) DTC

The DTC sends the internal bus arbiter a request for the bus when an activation request is generated. When the DTC accesses an external bus space, the DTC first takes control of t from the internal bus arbiter and then requests a bus to the external bus arbiter.

Once the DTC takes control of the bus, the DTC continues the transfer processing cycles. master whose priority is higher than the DTC requests the bus, the DTC transfers the bus higher priority bus master. If the IBCSS bit in BCR2 is set to 1, the DTC transfers the bus CPU.

Note, however, that the bus cannot be transferred in the following cases.

After the DMAC takes control of the bus, it may continue the transfer processing cycles
the bus at the end of every bus cycle depending on the conditions.

The DMAC continues transfers without releasing the bus in the following case:

- Between the read cycle in the dual-address mode and the write cycle corresponding t
  cycle

If no bus master of a higher priority than the DMAC requests the bus and the IBCSS bit
is cleared to 0, the DMAC continues transfers without releasing the bus in the following

- During 1-block transfers in the block transfer mode
- During transfers in the burst mode

In other cases, the DMAC transfers the bus at the end of the bus cycle.

(4) External Bus Release

When the $\overline{\text{BREQ}}$ pin goes low and an external bus release request is issued while the BR
BCR1 is set to 1 with the corresponding ICR bit set to 1, a bus request is sent to the bus

External bus release can be performed on completion of an external bus cycle.

6.15 Bus Controller Operation in Reset

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

with the setting for all peripheral module clocks to be stopped (MSTPCRA and MSTPCR
H'FFFFFFFF) or for operation of the 8-bit timer module alone (MSTPCRA and MSTPCR
H'F[E to 0]FFFFFF), and a transition is made to the sleep state, the all-module-clock-stop
entered in which the clock is also stopped for the bus controller and I/O ports. For details
section 20, Power-Down Modes.

In this state, the external bus release function is halted. To use the external bus release fun
sleep mode, the ACSE bit in MSTPCRA must be cleared to 0. Conversely, if a SLEEP in
to place the chip in all-module-clock-stop mode is executed in the external bus released s
transition to all-module-clock-stop mode is deferred and performed until after the bus is
recovered.

(3) External Bus Release Function and Software Standby

In this LSI, internal bus master operation does not stop even while the bus is released, as
the program is running in on-chip ROM, etc., and no external access occurs. If a SLEEP
instruction to place the chip in software standby mode is executed while the external bus
released, the transition to software standby mode is deferred and performed after the bus
recovered.

Also, since clock oscillation halts in software standby mode, if the $\overline{\text{BREQ}}$ signal goes low
mode, indicating an external bus release request, the request cannot be answered until the
recovered from the software standby mode.

Note that the $\overline{\text{BACK}}$ and $\overline{\text{BREQO}}$ pins are both in the high-impedance state in software st
mode.

- DMAC activation methods are auto-request, on-chip module interrupt, and external r

  Auto request: CPU activates (cycle stealing or burst access can be sele

  On-chip module interrupt: Interrupt requests from on-chip peripheral modules can b as an activation source

  External request: Low level or falling edge detection of the $\overline{\text{DREQ}}$ signal selected. External request is available for all four chann

  In block transfer mode, low level detection is only avail

- Dual or single address mode can be selected as address mode

  Dual address mode: Both source and destination are specified by addresses

  Single address mode: Either source or destination is specified by the $\overline{\text{DREQ}}$ signal an other is specified by address

- Normal, repeat, or block transfer can be selected as transfer mode

  Normal transfer mode: One byte, one word, or one longword data is transferred single transfer request

  Repeat transfer mode: One byte, one word, or one longword data is transferred single transfer request

  Repeat size of data is transferred and then a transfer adc returns to the transfer start address

  Up to 65536 transfers (65,536 bytes/words/longwords) as repeat size

  Block transfer mode: One block data is transferred at a single transfer request

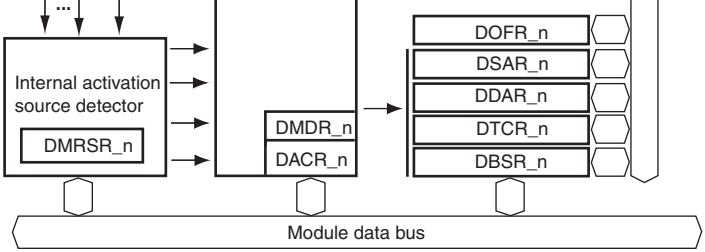
  Up to 65,536 bytes/words/longwords can be set as block

respective boundary

Data is divided according to its address (byte or word) when it is transferred

- Two types of interrupts can be requested to the CPU

  A transfer end interrupt is generated after the number of data specified by the transfer is transferred. A transfer escape end interrupt is generated when the remaining total tr size is less than the transfer data size at a single transfer request, when the repeat size transfer is completed, or when the extended repeat area overflows.



[Legend]

DSAR_n:	DMA source address register	$\overline{\text{DREQn}}$:	DMA transfer request
DDAR_n:	DMA destination address register	$\overline{\text{DACKn}}$:	DMA transfer acknowledge
DOFR_n:	DMA offset register	$\overline{\text{TENDn}}$:	DMA transfer end
DTCR_n:	DMA transfer count register	n = 0 to 3	
DBSR_n:	DMA block size register		
DMDR_n:	DMA mode control register		
DACR_n:	DMA address control register		
DMRSR_n:	DMA module request select register		

Figure 7.1 Block Diagram of DMAC

1	DMA transfer request 1	$\overline{\text{DREQ1}}$	Input	Channel 1 external reque
	DMA transfer acknowledge 1	$\overline{\text{DACK1}}$	Output	Channel 1 single address acknowledge
	DMA transfer end 1	$\overline{\text{TEND1}}$	Output	Channel 1 transfer end
2	DMA transfer request 2	$\overline{\text{DREQ2}}$	Input	Channel 2 external reque
	DMA transfer acknowledge 2	$\overline{\text{DACK2}}$	Output	Channel 2 single address acknowledge
	DMA transfer end 2	$\overline{\text{TEND2}}$	Output	Channel 2 transfer end
3	DMA transfer request 3	$\overline{\text{DREQ3}}$	Input	Channel 3 external reque
	DMA transfer acknowledge 3	$\overline{\text{DACK3}}$	Output	Channel 3 single address acknowledge
	DMA transfer end 3	$\overline{\text{TEND3}}$	Output	Channel 3 transfer end

- DMA block size register_0 (DBSR_0)
- DMA mode control register_0 (DMDR_0)
- DMA address control register_0 (DACR_0)
- DMA module request select register_0 (DMRSR_0)

Channel 1:

- DMA source address register_1 (DSAR_1)
- DMA destination address register_1 (DDAR_1)
- DMA offset register_1 (DOFR_1)
- DMA transfer count register_1 (DTCR_1)
- DMA block size register_1 (DBSR_1)
- DMA mode control register_1 (DMDR_1)
- DMA address control register_1 (DACR_1)
- DMA module request select register_1 (DMRSR_1)

Channel 2:

- DMA source address register_2 (DSAR_2)
- DMA destination address register_2 (DDAR_2)
- DMA offset register_2 (DOFR_2)
- DMA transfer count register_2 (DTCR_2)
- DMA block size register_2 (DBSR_2)
- DMA mode control register_2 (DMDR_2)
- DMA address control register_2 (DACR_2)
- DMA module request select register_2 (DMRSR_2)

7.3.1 DMA Source Address Register (DSAR)

DSAR is a 32-bit readable/writable register that specifies the transfer source address. DSA
updates the transfer source address every time data is transferred. When DDAR is specifi
destination address (the DIRS bit in DACR is 1) in single address mode, DSAR is ignore

Although DSAR can always be read from by the CPU, it must be read from in longwords
must not be written to while data for the channel is being transferred.

Bit	31	30	29	28	27	26	25	
Bit Name								
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	23	22	21	20	19	18	17	
Bit Name								
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	15	14	13	12	11	10	9	
Bit Name								
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	7	6	5	4	3	2	1	
Bit Name								
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	23	22	21	20	19	18	17	
Bit Name								
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Bit	15	14	13	12	11	10	9	
Bit Name								
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Bit	7	6	5	4	3	2	1	
Bit Name								
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Bit	23	22	21	20	19	18	17	
Bit Name								
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	15	14	13	12	11	10	9	
Bit Name								
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	7	6	5	4	3	2	1	
Bit Name								
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Although DTCR can always be read from by the CPU, it must be read from in longword must not be written to while data for the channel is being transferred.

Bit	31	30	29	28	27	26	25
Bit Name							
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	23	22	21	20	19	18	17
Bit Name							
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9
Bit Name							
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1
Bit Name							
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
Bit	15	14	13	12	11	10	9	
Bit Name	BKSZ15	BKSZ14	BKSZ13	BKSZ12	BKSZ11	BKSZ10	BKSZ9	BK
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
Bit	7	6	5	4	3	2	1	
Bit Name	BKSZ7	BKSZ6	BKSZ5	BKSZ4	BKSZ3	BKSZ2	BKSZ1	BK
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	BKSZH31 to BKSZH16	Undefined	R/W	Specify the repeat size or block size. When H'0001 is set, the repeat or block size is one word, or one longword. When H'0000 is se means the maximum value (refer to table 7.1). DMA is in operation, the setting is fixed.
15 to 0	BKSZ15 to BKSZ0	Undefined	R/W	Indicate the remaining repeat or block size whil DMA is in operation. The value is decremented every time data is transferred. When the remain becomes 0, the value of the BKSZH bits is load the same value as the BKSZH bits.

DMDR controls the DMAC operation.

- DMDR_0

Bit	31	30	29	28	27	26	25
Bit Name	DTE	DACKE	TENDE	—	DREQS	NRD	—
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	23	22	21	20	19	18	17
Bit Name	ACT	—	—	—	ERRF	—	ESIF
Initial Value	0	0	0	0	0	0	0
R/W	R	R	R	R	R/(W)*	R	R/(W)*

Bit	15	14	13	12	11	10	9
Bit Name	DTSZ1	DTSZ0	MDS1	MDS0	TSEIE	—	ESIE
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R	R/W

Bit	7	6	5	4	3	2	1
Bit Name	DTF1	DTF0	DTA	—	—	DMAP2	DMAP1
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R	R/W	R/W

Note: * Only 0 can be written to this bit after having been read as 1, to clear the flag.

Bit Name	DTSZ1	DTSZ0	MDS1	MDS0	TSIZE		ESIZE	[illegible]
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	[illegible]

Bit	7	6	5	4	3	2	1	
Bit Name	DTF1	DTF0	DTA	—	—	DMAP2	DMAP1	DM
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R	R	R/W	R/W	[illegible]

Note: * Only 0 can be written to this bit after having been read as 1, to clear the flag.

transfer.

In block transfer mode, if writing 0 to this bit w
being transferred, this bit is cleared to 0 after t
1-block size data transfer.

If an event which stops (sustains) a transfer o
externally, this bit is automatically cleared to 0
the transfer.

Operating modes and transfer methods must
changed while this bit is set to 1.

0: Disables a data transfer

1: Enables a data transfer (DMA is in operatio

[Clearing conditions]

- When the specified total transfer size of tra
  completed
- When a transfer is stopped by an overflow
  by a repeat size end
- When a transfer is stopped by an overflow
  by an extended repeat size end
- When a transfer is stopped by a transfer s
  interrupt
- When clearing this bit to 0 to stop a transfe

In block transfer mode, this bit changes after t
block transfer.

- When an address error or an NMI interrup
  requested
- In the reset state or hardware standby mo

28	—	0	R/W	Reserved Initial value should not be changed.
27	DREQS	0	R/W	$\overline{\text{DREQ}}$ Select Selects whether a low level or the falling edge o $\overline{\text{DREQ}}$ signal used in external request mode is When a block transfer is performed in external mode, clear this bit to 0. 0: Low level detection 1: Falling edge detection (the first transfer afte transfer enabled is detected on a low level)
26	NRD	0	R/W	Next Request Delay Selects the accepting timing of the next transfe 0: Starts accepting the next transfer request a completion of the current transfer 1: Starts accepting the next transfer request o after completion of the current transfer
25, 24	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
23	ACT	0	R	Active State Indicates the operating state for the channel. 0: Waiting for a transfer request or a transfer c state by clearing the DTE bit to 0 1: Active state
22 to 20	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.

				generated [Clearing condition] • When clearing to 0 after reading ERRF = 1 [Setting condition] • When an address error or an NMI interrupt generated However, when an address error or an NMI in been generated in DMAC module stop mode, not set to 1.
18	—	0	R	Reserved This bit is always read as 0 and cannot be mo
17	ESIF	0	R/(W)*	Transfer Escape Interrupt Flag Indicates that a transfer escape end interrupt requested. A transfer escape end means that is terminated before the transfer counter reach 0: A transfer escape end interrupt has not be requested 1: A transfer escape end interrupt has been r [Clearing conditions] • When setting the DTE bit to 1 • When clearing to 0 before reading ESIF = [Setting conditions] • When a transfer size error interrupt is requ • When a repeat size end interrupt is reques • When a transfer end interrupt by an extend area overflow is requested

				• When setting the DTE bit to 1 • When clearing to 0 after reading DTIF = 1 [Setting condition] • When DTCR reaches 0 and the transfer is completed
15	DTSZ1	0	R/W	Data Access Size 1 and 0
14	DTSZ0	0	R/W	Select the data access size for a transfer. 00: Byte size (eight bits) 01: Word size (16 bits) 10: Longword size (32 bits) 11: Setting prohibited
13	MDS1	0	R/W	Transfer Mode Select 1 and 0
12	MDS0	0	R/W	Select the transfer mode. 00: Normal transfer mode 01: Block transfer mode 10: Repeat transfer mode 11: Setting prohibited

				• In normal or repeat transfer mode, the tota size set in DTCR is less than the data acc • In block transfer mode, the total transfer si DTCR is less than the block size 0: Disables a transfer size error interrupt requ 1: Enables a transfer size error interrupt reque
10	—	0	R	Reserved This bit is always read as 0 and cannot be mo
9	ESIE	0	R/W	Transfer Escape Interrupt Enable Enables/disables a transfer escape end interr request. When the ESIF bit is set to 1 with this 1, a transfer escape end interrupt is requested CPU or DTC. The transfer end interrupt reque cleared by clearing this bit or the ESIF bit to 0 0: Disables a transfer escape end interrupt 1: Enables a transfer escape end interrupt
8	DTIE	0	R/W	Data Transfer End Interrupt Enable Enables/disables a transfer end interrupt requ transfer counter. When the DTIF bit is set to 1 bit set to 1, a transfer end interrupt is requeste CPU or DTC. The transfer end interrupt reque cleared by clearing this bit or the DTIF bit to 0 0: Disables a transfer end interrupt 1: Enables a transfer end interrupt

				11: External request
5	DTA	0	R/W	Data Transfer Acknowledge This bit is valid in DMA transfer by the on-chip r interrupt source. This bit enables or disables to source flag selected by DMRSR. 0: To clear the source in DMA transfer is disabl Since the on-chip module interrupt source is cleared in DMA transfer, it should be clearec CPU or DTC transfer. 1: To clear the source in DMA transfer is enabl Since the on-chip module interrupt source is in DMA transfer, it does not require an interru CPU or DTC transfer.
4, 3	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.

	000: Priority level 0 (low)
	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 (high)

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

Initial Value								
R/W	R	R	R/W	R/W	R	R	R/W	R

Bit	15	14	13	12	11	10	9	
Bit Name	SARIE	—	—	SARA4	SARA3	SARA2	SARA1	SA
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R	R	R/W	R/W	R/W	R/W	R

Bit	7	6	5	4	3	2	1	
Bit Name	DARIE	—	—	DARA4	DARA3	DARA2	DARA1	DA
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R	R	R/W	R/W	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
31	AMS	0	R/W	Address Mode Select Selects address mode from single or dual addr mode. In single address mode, the $\overline{\text{DACK}}$ pin is according to the DACKE bit. 0: Dual address mode 1: Single address mode
30	DIRS	0	R/W	Single Address Direction Select Specifies the data transfer direction in single ad mode. This bit s ignored in dual address mode. 0: Specifies DSAR as source address 1: Specifies DDAR as destination address
29 to 27	—	0	R/W	Reserved These bits are always read as 0 and cannot be modified.

				In addition, in block transfer mode, when the transfer is requested after 1-block data transfe this bit is set to 1, the DTE bit in DMDR is clea At this time, the ESIF bit in DMDR is set to 1 t that a repeat size end interrupt is requested. 0: Disables a repeat size end interrupt 1: Enables a repeat size end interrupt
25	ARS1	0	R/W	Area Select 1 and 0
24	ARS0	0	R/W	Specify the block area or repeat area in block transfer mode. 00: Specify the block area or repeat area on th address 01: Specify the block area or repeat area on th destination address 10: Do not specify the block area or repeat are 11: Setting prohibited
23, 22	—	All 0	R	Reserved These bits are always read as 0 and cannot b modified.
21	SAT1	0	R/W	Source Address Update Mode 1 and 0
20	SAT0	0	R/W	Select the update method of the source addre (DSAR). When DSAR is not specified as the t source in single address mode, this bit is igno 00: Source address is fixed 01: Source address is updated by adding the 10: Source address is updated by adding 1, 2 according to the data access size 11: Source address is updated by subtracting according to the data access size

				10: Destination address is updated by adding 1 according to the data access size 11: Destination address is updated by subtracti or 4 according to the data access size
15	SARIE	0	R/W	Interrupt Enable for Source Address Extended Overflow Enables/disables an interrupt request for an ex area overflow on the source address. When an extended repeat area overflow on the address occurs while this bit is set to 1, the DT DMDR is cleared to 0. At this time, the ESIF bit DMDR is set to 1 to indicate an interrupt by an repeat area overflow on the source address is requested. When block transfer mode is used with the exte repeat area function, an interrupt is requested a completion of a 1-block size transfer. When set DTE bit in DMDR of the channel for which a tra been stopped to 1, the transfer is resumed from state when the transfer is stopped. When the extended repeat area is not specifiec is ignored. 0: Disables an interrupt request for an extendec overflow on the source address 1: Enables an interrupt request for an extended overflow on the source address
14, 13	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.

				area for address addition and subtraction, resp When an overflow in the extended repeat area with the SARIE bit set to 1, an interrupt can be requested. Table 7.3 shows the settings and a the extended repeat area.
7	DARIE	0	R/W	Destination Address Extended Repeat Area O Interrupt Enable Enables/disables an interrupt request for an ex area overflow on the destination address. When an extended repeat area overflow on th destination address occurs while this bit is set DTE bit in DMDR is cleared to 0. At this time, bit in DMDR is set to 1 to indicate an interrupt extended repeat area overflow on the destinat address is requested. When block transfer mode is used with the ext repeat area function, an interrupt is requested completion of a 1-block size transfer. When se DTE bit in DMDR of the channel for which the has been stopped to 1, the transfer is resume state when the transfer is stopped. When the extended repeat area is not specifie is ignored. 0: Disables an interrupt request for an extende overflow on the destination address 1: Enables an interrupt request for an extende overflow on the destination address
6, 5	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.

area for address addition and subtraction, resp

When an overflow in the extended repeat area
with the DARIE bit set to 1, an interrupt can be
requested. Table 7.3 shows the settings and ar
the extended repeat area.

00101	32 bytes specified as extended repeat area by the lower 5 bits of the addres
00110	64 bytes specified as extended repeat area by the lower 6 bits of the addres
00111	128 bytes specified as extended repeat area by the lower 7 bits of the addre
01000	256 bytes specified as extended repeat area by the lower 8 bits of the addre
01001	512 bytes specified as extended repeat area by the lower 9 bits of the addre
01010	1 kbyte specified as extended repeat area by the lower 10 bits of the addres
01011	2 kbytes specified as extended repeat area by the lower 11 bits of the addre
01100	4 kbytes specified as extended repeat area by the lower 12 bits of the addre
01101	8 kbytes specified as extended repeat area by the lower 13 bits of the addre
01110	16 kbytes specified as extended repeat area by the lower 14 bits of the addr
01111	32 kbytes specified as extended repeat area by the lower 15 bits of the addr
10000	64 kbytes specified as extended repeat area by the lower 16 bits of the addr
10001	128 kbytes specified as extended repeat area by the lower 17 bits of the ad
10010	256 kbytes specified as extended repeat area by the lower 18 bits of the ad
10011	512 kbytes specified as extended repeat area by the lower 19 bits of the ad
10100	1 Mbyte specified as extended repeat area by the lower 20 bits of the addre
10101	2 Mbytes specified as extended repeat area by the lower 21 bits of the addr
10110	4 Mbytes specified as extended repeat area by the lower 22 bits of the addr
10111	8 Mbytes specified as extended repeat area by the lower 23 bits of the addr
11000	16 Mbytes specified as extended repeat area by the lower 24 bits of the ad
11001	32 Mbytes specified as extended repeat area by the lower 25 bits of the ad
11010	64 Mbytes specified as extended repeat area by the lower 26 bits of the ad
11011	128 Mbytes specified as extended repeat area by the lower 27 bits of the a
111××	Setting prohibited

[Legend]

×: Don't care

7.4 Transfer Modes

Table 7.4 shows the DMAC transfer modes. The transfer modes can be specified to the in
channels.

Table 7.4 Transfer Modes

Address Mode	Transfer mode	Activation Source	Common Function	Address R	
				Source	
Dual address	• Normal transfer • Repeat transfer • Block transfer Repeat or block size = 1 to 65,536 bytes, 1 to 65,536 words, or 1 to 65,536 longwords	• Auto request (activated by CPU) • On-chip module interrupt • External request	• Total transfer size: 1 to 4 Gbytes or not specified • Offset addition • Extended repeat area function	DSAR	
Single address	• Instead of specifying the source or destination address registers, data is directly transferred from/to the external device using the $\overline{\text{DACK}}$ pin • The same settings as above are available other than address register setting (e.g., above transfer modes can be specified) • One transfer can be performed in one bus cycle (the types of transfer modes are the same as those of dual address modes)			DSAR/ $\overline{\text{DACK}}$	

In dual address mode, the transfer source address is specified in DSAR and the transfer
address is specified in DDAR. A transfer at a time is performed in two bus cycles (when
bus width is less than the data access size or the access address is not aligned with the b
the data access size, the number of bus cycles are needed more than two because one bu
divided into multiple bus cycles).

In the first bus cycle, data at the transfer source address is read and in the next cycle, the
is written to the transfer destination address.

The read and write cycles are not separated. Other bus cycles (bus cycle by other bus ma
refresh cycle, and external bus release cycle) are not generated between read and write c

The $\overline{\text{TEND}}$ signal output is enabled or disabled by the TENDE bit in DMDR. The $\overline{\text{TEND}}$
output in two bus cycles. When an idle cycle is inserted before the bus cycle, the $\overline{\text{TEND}}$
also output in the idle cycle. The $\overline{\text{DACK}}$ signal is not output.

Figure 7.2 shows an example of the signal timing in dual address mode and figure 7.3 sh
operation in dual address mode.

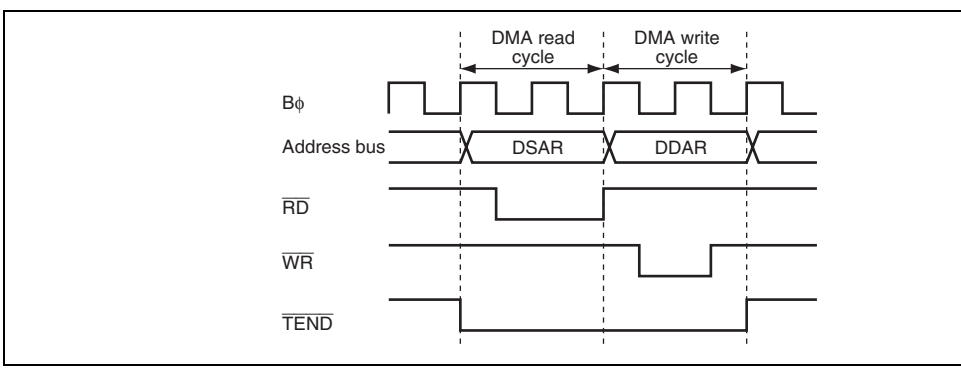


Figure 7.2 Example of Signal Timing in Dual Address Mode

(2) Single Address Mode

In single address mode, data between an external device and an external memory is direct
transferred using the $\overline{\text{DACK}}$ pin instead of DSAR or DDAR. A transfer at a time is perfo
one bus cycle. In this mode, the data bus width must be the same as the data access size. I
details on the data bus width, see section 6, Bus Controller (BSC).

The DMAC accesses an external device as the transfer source or destination by outputting
strobe signal ($\overline{\text{DACK}}$) to the external device with $\overline{\text{DACK}}$ and accesses the other transfer t
outputting the address. Accordingly, the DMA transfer is performed in one bus cycle. Fig
shows an example of a transfer between an external memory and an external device with
$\overline{\text{DACK}}$ pin. In this example, the external device outputs data on the data bus and the data
to the external memory in the same bus cycle.

The transfer direction is decided by the DIRS bit in DACR which specifies an external de
the $\overline{\text{DACK}}$ pin as the transfer source or destination. When DIRS = 0, data is transferred fr
external memory (DSAR) to an external device with the $\overline{\text{DACK}}$ pin. When DIRS = 1, dat
transferred from an external device with the $\overline{\text{DACK}}$ pin to an external memory (DDAR).
settings of registers which are not used as the transfer source or destination are ignored.

The $\overline{\text{DACK}}$ signal output is enabled in single address mode by the DACKE bit in DMDR
$\overline{\text{DACK}}$ signal is low active.

The $\overline{\text{TEND}}$ signal output is enabled or disabled by the TENDE bit in DMDR. The $\overline{\text{TEND}}$
output in one bus cycle. When an idle cycle is inserted before the bus cycle, the $\overline{\text{TEND}}$ si
also output in the idle cycle.

Figure 7.5 shows an example of timing charts in single address mode and figure 7.6 show
example of operation in single address mode.

Figure 7.4 Data Flow in Single Address Mode

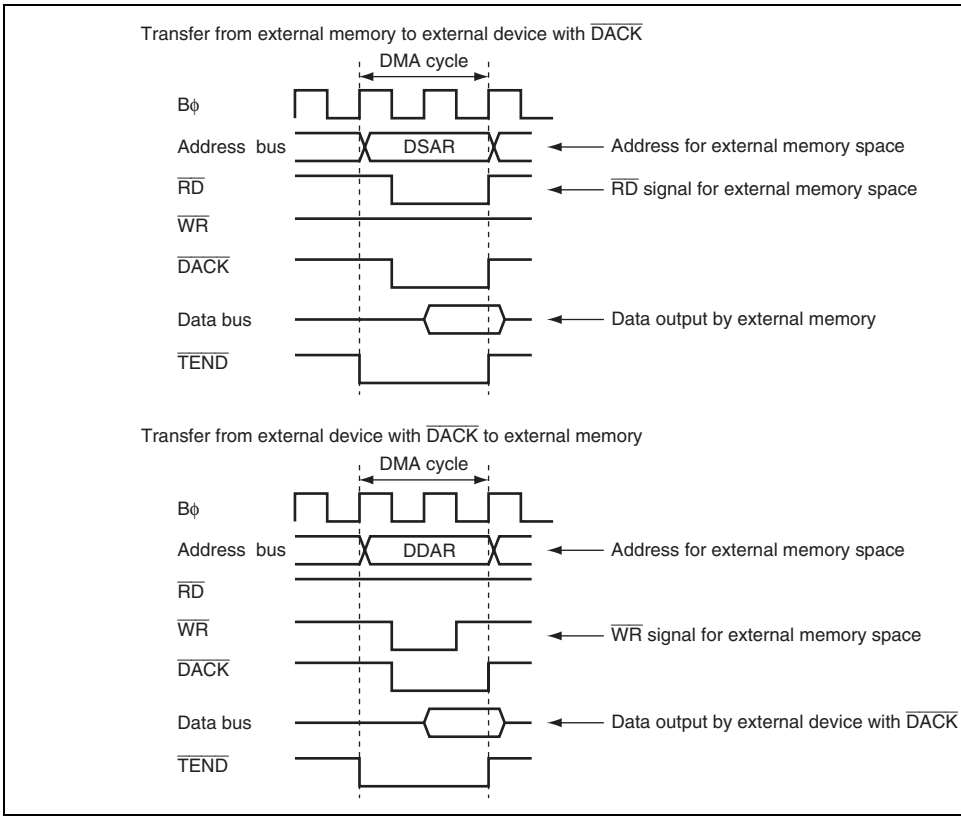


Figure 7.5 Example of Signal Timing in Single Address Mode

Figure 7.6 Operations in Single Address Mode

7.5.2 Transfer Modes

(1) Normal Transfer Mode

In normal transfer mode, one data access size of data is transferred at a single transfer req
to 4 Gbytes can be specified as a total transfer size by DTCR. DBSR is ignored in normal
mode.

The $\overline{\text{TEND}}$ signal is output only in the last DMA transfer. The $\overline{\text{DACK}}$ signal is output eve
transfer request is received and a transfer starts.

Figure 7.7 shows an example of the signal timing in normal transfer mode and figure 7.8
the operation in normal transfer mode.

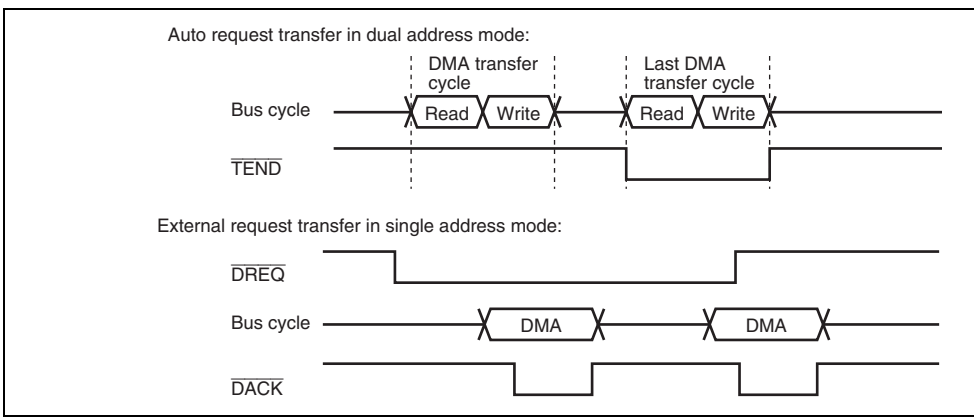


Figure 7.7 Example of Signal Timing in Normal Transfer Mode

(2) Repeat Transfer Mode

In repeat transfer mode, one data access size of data is transferred at a single transfer req
to 4 Gbytes can be specified as a total transfer size by DTCR. The repeat size can be spe
DBSR up to 65536 × data access size.

The repeat area can be specified for the source or destination address side by bits ARS1
in DACR. The address specified as the repeat area returns to the transfer start address w
repeat size of transfers is completed. This operation is repeated until the total transfer siz
specified in DTCR is completed. When H'00000000 is specified in DTCR, it is regarded
free running mode and repeat transfer is continued until the DTE bit in DMDR is cleared

In addition, a DMA transfer can be stopped and a repeat size end interrupt can be reques
CPU or DTC when the repeat size of transfers is completed. When the next transfer is re
after completion of a 1-repeat size data transfer while the RPTIE bit is set to 1, the DTE
DMDR is cleared to 0 and the ESIF bit in DMDR is set to 1 to complete the transfer. At
an interrupt is requested to the CPU or DTC when the ESIE bit in DMDR is set to 1.

The timings of the $\overline{\text{TEND}}$ and $\overline{\text{DACK}}$ signals are the same as in normal transfer mode.

Figure 7.9 shows the operation in repeat transfer mode while dual address mode is set.

When the repeat area is specified as neither source nor destination address side, the oper
the same as the normal transfer mode operation shown in figure 7.8. In this case, a repea
interrupt can also be requested to the CPU when the repeat size of transfers is completed

to the source side

Figure 7.9 Operations in Repeat Transfer Mode

(3) Block Transfer Mode

In block transfer mode, one block size of data is transferred at a single transfer request. U
Gbytes can be specified as total transfer size by DTCR. The block size can be specified in
up to 65536 × data access size.

While one block of data is being transferred, transfer requests from other channels are sus
When the transfer is completed, the bus is released to the other bus master.

The block area can be specified for the source or destination address side by bits ARS1 an
in DACR. The address specified as the block area returns to the transfer start address whe
block size of data is completed. When the block area is specified as neither source nor de
address side, the operation continues without returning the address to the transfer start ad
repeat size end interrupt can be requested.

The $\overline{\text{TEND}}$ signal is output every time 1-block data is transferred in the last DMA transfe
When the external request is selected as an activation source, the low level detection of th
signal (DREQS = 0) should be selected.

When an interrupt request by an extended repeat area overflow is used in block transfer m
settings should be selected carefully. For details, see section 7.5.5, Extended Repeat Area
Function.

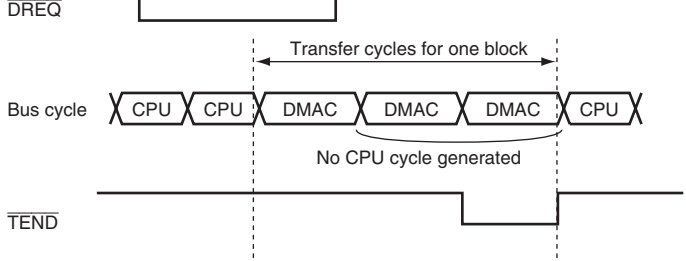


Figure 7.10 Operations in Block Transfer Mode

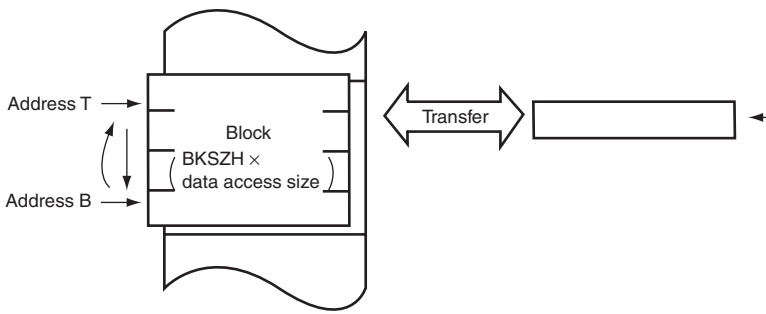


Figure 7.11 Operation in Single Address Mode in Block Transfer Mode (Block Area Specified)

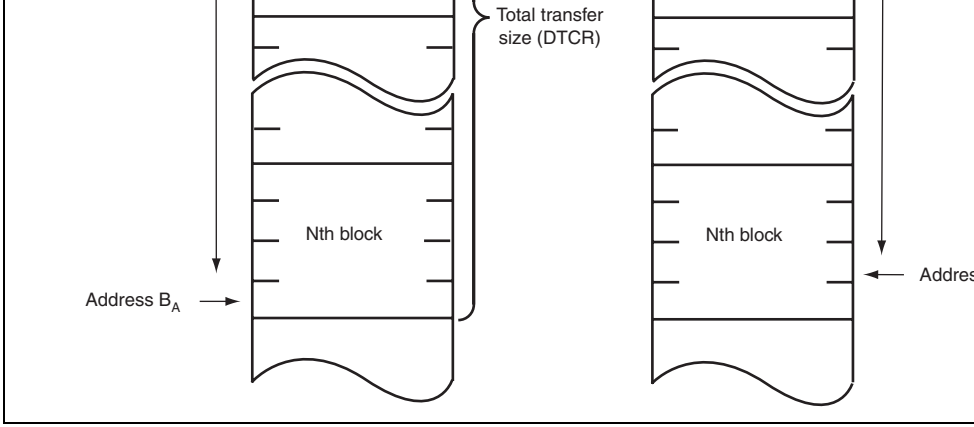


Figure 7.12 Operation in Dual Address Mode in Block Transfer Mode (Block Area Not Specified)

DMDR starts a transfer. The bus mode can be selected from cycle stealing and burst mo

(2) Activation by On-Chip Module Interrupt

An interrupt request from an on-chip peripheral module (on-chip peripheral module inte used as a transfer request. When a DMA transfer is enabled (DTE = 1), the DMA transfe started by an on-chip module interrupt.

The activation source of the on-chip module interrupt is selected by the DMA module re select register (DMRSR). The activation sources are specified to the individual channels 7.5 is a list of on-chip module interrupts for the DMAC. The interrupt request selected a activation source can generate an interrupt request simultaneously to the CPU or DTC. F refer to section 5, Interrupt Controller.

The DMAC receives interrupt requests by on-chip peripheral modules independent of th controller. Therefore, the DMAC is not affected by priority given in the interrupt contro

When the DMAC is activated while DTA = 1, the interrupt request flag is automatically a DMA transfer. If multiple channels use a single transfer request as an activation sourc the channel having priority is activated, the interrupt request flag is cleared. In this case, channels may not be activated because the transfer request is not held in the DMAC.

When the DMAC is activated while DTA = 0, the interrupt request flag is not cleared by DMAC and should be cleared by the CPU or DTC transfer.

When an activation source is selected while DTE = 0, the activation source does not req transfer to the DMAC. It requests an interrupt to the CPU or DTC.

In addition, make sure that an interrupt request flag as an on-chip module interrupt sourc cleared to 0 before writing 1 to the DTE bit.

TGI4A (TGI4A input capture/compare match)	TPU_4	10
TGI5A (TGI5A input capture/compare match)	TPU_5	11
RXI0 (receive data full interrupt for SCI channel 0)	SCI_0	14
TXI0 (transmit data empty interrupt for SCI channel 0)	SCI_0	14
RXI1 (receive data full interrupt for SCI channel 1)	SCI_1	14
TXI1 (transmit data empty interrupt for SCI channel 1)	SCI_1	15
RXI2 (receive data full interrupt for SCI channel 2)	SCI_2	15
TXI2 (transmit data empty interrupt for SCI channel 2)	SCI_2	15
RXI4 (receive data full interrupt for SCI channel 4)	SCI_4	16
TXI4 (transmit data empty interrupt for SCI channel 4)	SCI_4	16

(3) Activation by External Request

A transfer is started by a transfer request signal ($\overline{\text{DREQ}}$) from an external device. When a transfer is enabled (DTE = 1), the DMA transfer is started by the $\overline{\text{DREQ}}$ assertion. When transfer between on-chip peripheral modules is performed, select an activation source fro auto request and on-chip module interrupt (the external request cannot be used).

A transfer request signal is input to the $\overline{\text{DREQ}}$ pin. The $\overline{\text{DREQ}}$ signal is detected on the fa edge or low level. Whether the falling edge or low level detection is used is selected by th DREQS bit in DMDR. To perform a block transfer, select the low level detection (DREQ

When an external request is selected as an activation source, clear the DDR bit to 0 and se ICR bit to 1 for the corresponding pin. For details, see section 9, I/O Ports.

longword, or 1-block size) is completed. After that, when a transfer is requested, the DM
obtains the bus to transfer 1-unit data and then releases the bus on completion of the tran
operation is continued until the transfer end condition is satisfied.

When a transfer is requested to another channel during a DMA transfer, the DMAC rele
bus and then transfers data for the requested channel. For details on operations when a tr
requested to multiple channels, see section 7.5.8, Priority of Channels.

Figure 7.13 shows an example of timing in cycle stealing mode. The transfer conditions
follows:

- Address mode: Single address mode
- Sampling method of the $\overline{\text{DREQ}}$ signal: Low level detection

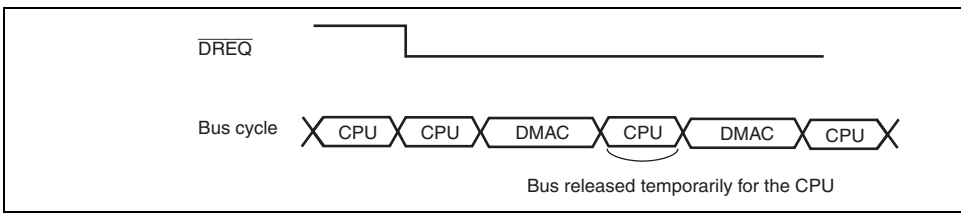


Figure 7.13 Example of Timing in Cycle Stealing Mode

Clearing the DTE bit in DMDR stops a DMA transfer. A transfer requested before the DT
cleared to 0 by the DMAC is executed. When an interrupt by a transfer size error, a repea
end, or an extended repeat area overflow occurs, the DTE bit is cleared to 0 and the trans

Figure 7.14 shows an example of timing in burst mode.

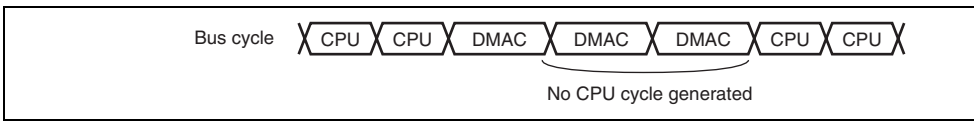


Figure 7.14 Example of Timing in Burst Mode

7.5.5 Extended Repeat Area Function

The source and destination address sides can be specified as the extended repeat area. Th
of the address register repeat addresses within the area specified as the extended repeat ar
example, to use a ring buffer as the transfer target, the contents of the address register sho
return to the start address of the buffer every time the contents reach the end address of th
(overflow on the ring buffer address). This operation can automatically be performed usin
extended repeat area function of the DMAC.

The extended repeat areas can be specified independently to the source address register (I
and destination address register (DDAR).

The extended repeat area on the source address is specified by bits SARA4 to SARA0 in
The extended repeat area on the destination address is specified by bits DARA4 to DARA
DACR. The extended repeat area sizes for each side can be specified independently.

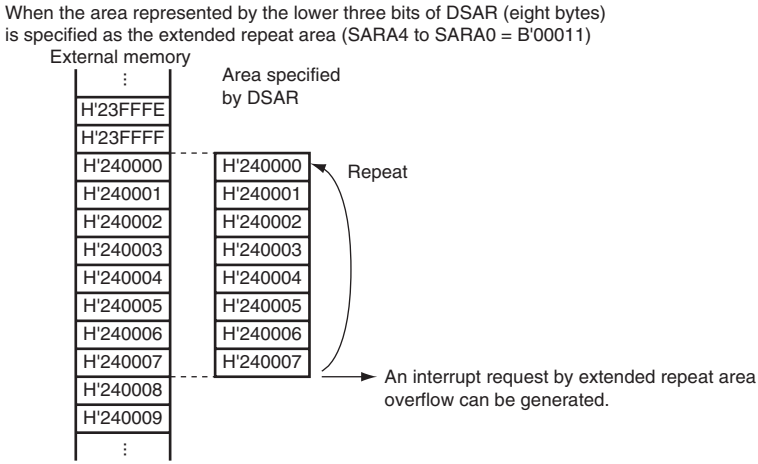


Figure 7.15 Example of Extended Repeat Area Operation

When an interrupt by an extended repeat area overflow is used in block transfer mode, t
following should be taken into consideration.

When a transfer is stopped by an interrupt by an extended repeat area overflow, the addr
register must be set so that the block size is a power of 2 or the block size boundary is al
the extended repeat area boundary. When an overflow on the extended repeat area occur
transfer of one block, the interrupt by the overflow is suspended and the transfer overrun

Figure 7.16 shows examples when the extended repeat area function is used in block tra
mode.

Figure 7.16 Example of Extended Repeat Area Function in Block Transfer M

7.5.6 Address Update Function using Offset

The source and destination addresses are updated by fixing, increment/decrement by 1, 2, offset addition. When the offset addition is selected, the offset specified by the offset regi (DOFR) is added to the address every time the DMAC transfers the data access size of da function realizes a data transfer where addresses are allocated to separated areas.

Figure 7.17 shows the address update method.

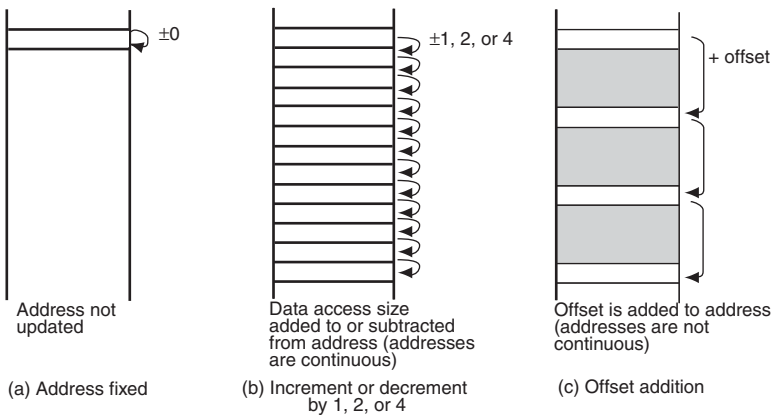


Figure 7.17 Address Update Method

size.

The address is calculated by the offset set in DOFR and the contents of DSAR and DDA
Although the DMAC calculates only addition, an offset subtraction can be realized by se
negative value in DOFR. In this case, the negative value must be 2's complement.

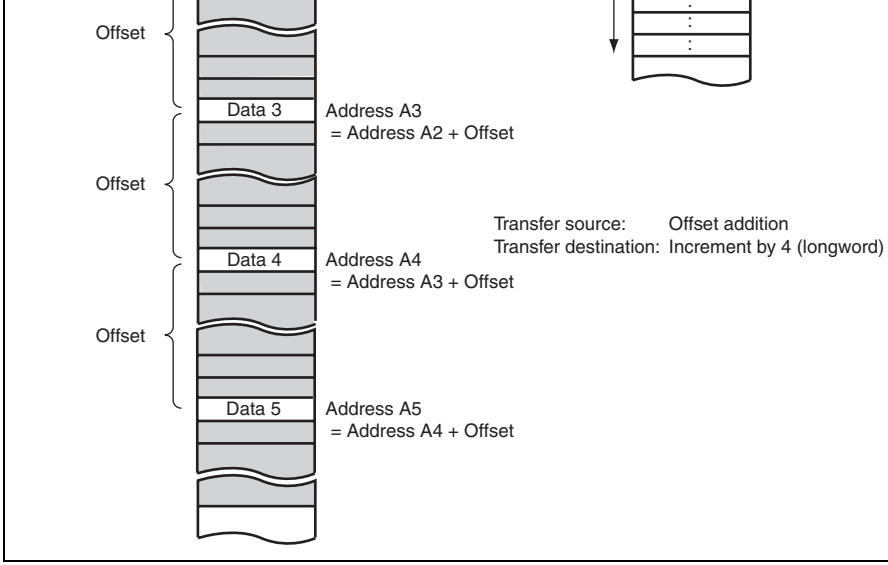


Figure 7.18 Operation of Offset Addition

In figure 7.18, the offset addition is selected as the transfer source address update and inc
decrement by 1, 2, or 4 is selected as the transfer destination address. The address update
that data at the address which is away from the previous transfer source address by the of
read from. The data read from the address away from the previous address is written to th
consecutive area in the destination side.

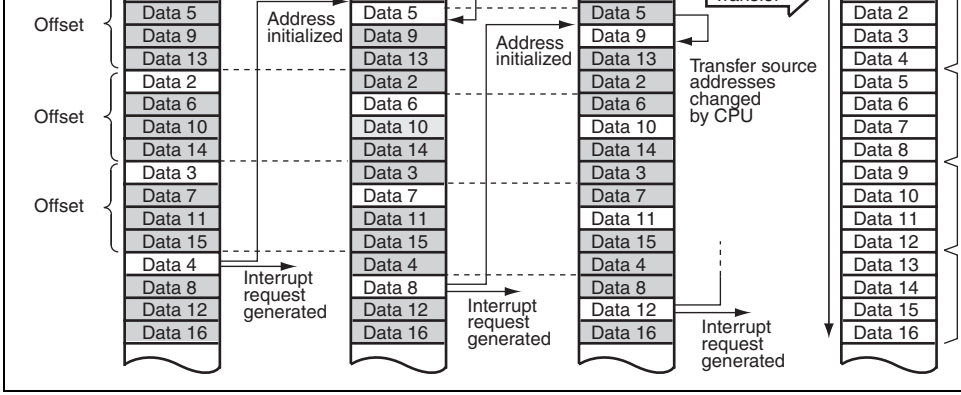


Figure 7.19 XY Conversion Operation Using Offset Addition in Repeat Transfe

In figure 7.19, the source address side is specified to the repeat area by DACR and the o addition is selected. The offset value is set to 4 × data access size (when the data access longword, H'00000010 is set in DOFR, as an example). The repeat size is set to 4 × data size (when the data access size is longword, the repeat size is set to 4 × 4 = 16 bytes, as example). The increment or decrement by 1, 2, or 4 is specified as the transfer destinatio A repeat size end interrupt is requested when the repeat size of transfers is completed.

When a transfer starts, the transfer source address is added to the offset every time data transferred. The transfer data is written to the destination continuous addresses. When da transferred meaning that the repeat size of transfers is completed, the transfer source adc returns to the transfer start address (address of data 1 on the transfer source) and a repea interrupt is requested. While this interrupt stops the transfer temporarily, the contents of written to the address of data 5 by the CPU (when the data access size is longword, write 1 address + 4). When the DTE bit in DMDR is set to 1, the transfer is resumed from the the transfer is stopped. Accordingly, operations are repeated and the transfer source data transposed to the destination area (XY conversion).

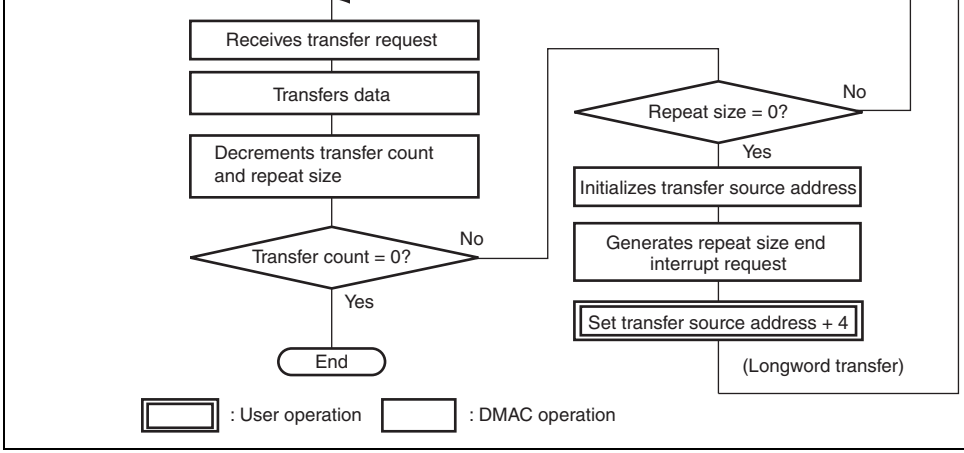


Figure 7.20 XY Conversion Flowchart Using Offset Addition in Repeat Transfer

(3) Offset Subtraction

When setting the negative value in DOFR, the offset value must be 2's complement. The
complement is obtained by the following formula.

2's complement of offset = 1 + ~offset (~: bit inversion)

Example: 2's complement of H'0001FFFF
= H'FFFE0000 + H'00000001
= H'FFFE0001

The value of 2's complement can be obtained by the NEG.L instruction.

The increment or decrement can be specified by bits SAT1 and SAT0 in DACR. When S
SAT0 = B'00, the address is fixed. When SAT1 and SAT0 = B'01, the address is added v
offset. When SAT1 and SAT0 = B'10, the address is incremented. When SAT1 and SAT
the address is decremented. The size of increment or decrement depends on the data acc

The data access size is specified by bits DTSZ1 and DTSZ0 in DMDR. When DTSZ1 an
= B'00, the data access size is byte and the address is incremented or decremented by 1.
DTSZ1 and DTSZ0 = B'01, the data access size is word and the address is incremented o
decremented by 2. When DTSZ1 and DTSZ0 = B'10, the data access size is longword an
address is incremented or decremented by 4. Even if the access data size of the source ad
word or longword, when the source address is not aligned with the word or longword bo
the read bus cycle is divided into byte or word cycles. While data of one word or one lon
being read, the size of increment or decrement is changing according to the actual data a
for example, +1 or +2 for byte or word data. After one word or one longword of data is r
address when the read cycle is started is incremented or decremented by the value accord
bits SAT1 and SAT0.

In block or repeat transfer mode, when the block or repeat size of data transfers is compl
the block or repeat area is specified to the source address side, the source address returns
transfer start address and is not affected by the address update.

When the extended repeat area is specified to the source address side, operation follows
setting. The upper address bits are fixed and is not affected by the address update.

While data is being transferred, DSAR must be accessed in longwords. If the upper word
lower word are read separately, incorrect data may be read from since the contents of DS
during the transfer may be updated regardless of the access by the CPU. Moreover, DSA
channel being transferred must not be written to.

The data access size is specified by bits DTSZ1 and DTSZ0 in DMDR. When DTSZ1 and
= B'00, the data access size is byte and the address is incremented or decremented by 1. W
DTSZ1 and DTSZ0 = B'01, the data access size is word and the address is incremented or
decremented by 2. When DTSZ1 and DTSZ0 = B'10, the data access size is longword and
address is incremented or decremented by 4. Even if the access data size of the destinatio
is word or longword, when the destination address is not aligned with the word or longwo
boundary, the write bus cycle is divided into byte and word cycles. While one word or on
longword of data is being written, the incrementing or decrementing size is changing acco
the actual data access size, for example, +1 or +2 for byte or word data. After the one wor
longword of data is written, the address when the write cycle is started is incremented or
decremented by the value according to bits SAT1 and SAT0.

In block or repeat transfer mode, when the block or repeat size of data transfers is comple
the block or repeat area is specified to the destination address side, the destination addres
to the transfer start address and is not affected by the address update.

When the extended repeat area is specified to the destination address side, operation follo
setting. The upper address bits are fixed and is not affected by the address update.

While data is being transferred, DDAR must be accessed in longwords. If the upper word
lower word are read separately, incorrect data may be read from since the contents of DD
during the transfer may be updated regardless of the access by the CPU. Moreover, DDA
channel being transferred must not be written to.

When a conflict occurs between the address update by DMA transfer and write access b
the CPU has priority. When a conflict occurs between change from 1, 2, or 4 to 0 in DT
write access by the CPU (other than 0), the CPU has priority in writing to DTCR. Howe
transfer is stopped.

(4) DMA Block Size Register (DBSR)

DBSR is enabled in block or repeat transfer mode. Bits 31 to 16 in DBSR function as B
bits 15 to 0 in DBSR function as BKSZ. The BKSZH bits (16 bits) store the block size a
size and its value is not changed. The BKSZ bits (16 bits) function as a counter for the b
and repeat size and its value is decremented every transfer by 1. When the BKSZ value
change from 1 to 0 by a DMA transfer, 0 is not stored but the BKSZH value is loaded in
BKSZ bits.

Since the upper 16 bits of DBSR are not updated, DBSR can be accessed in words.

DBSR for the channel being transferred must not be written to.

- When a transfer is stopped by an NMI interrupt
- When a transfer is stopped by and address error
- Reset state
- Hardware standby mode
- When a transfer is stopped by writing 0 to the DTE bit

Writing to the registers for the channels when the corresponding DTE bit is set to 1 is pro
(except for the DTE bit). When changing the register settings after writing 0 to the DTE b
confirm that the DTE bit has been cleared to 0.

Figure 7.21 show the procedure for changing the register settings for the channel being transferred.

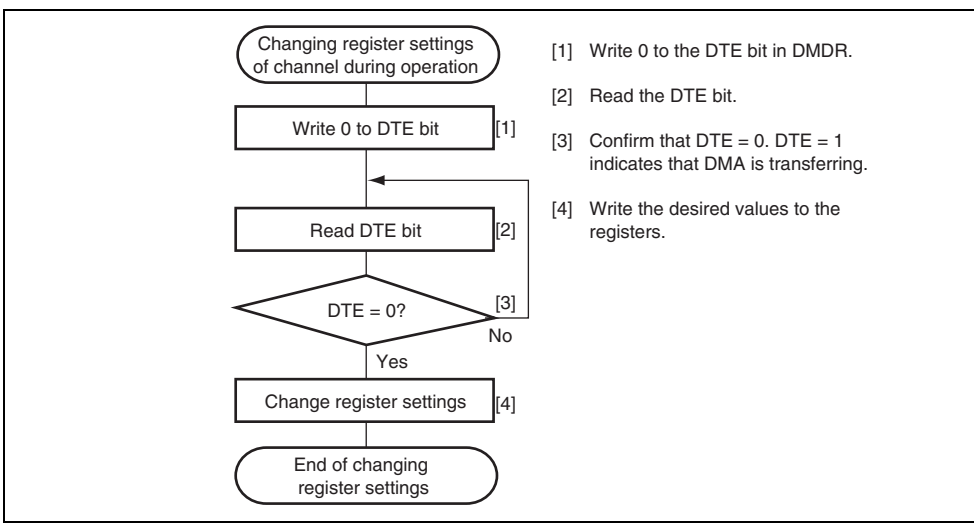


Figure 7.21 Procedure for Changing Register Setting For Channel being Transf

In burst mode, up to three times of DMA transfer are performed from the cycle in which
bit is written to 0. The ACT bit retains 1 from writing 0 to the DTE bit to completion of
transfer.

(7) ERRF Bit in DMDR

When an address error or an NMI interrupt occur, the DMAC clears the DTE bits for all
channels to stop a transfer. In addition, it sets the ERRF bit in DMDR_0 to 1 to indicate
address error or an NMI interrupt has occurred regardless of whether or not the DMAC
operation.

(8) ESIF Bit in DMDR

When an interrupt by an transfer size error, a repeat size end, or an extended repeat area
is requested, the ESIF bit in DMDR is set to 1. When both the ESIF and ESIE bits are se
transfer escape interrupt is requested to the CPU or DTC.

The ESIF bit is set to 1 when the ACT bit in DMDR is cleared to 0 to stop a transfer afte
cycle of the interrupt source is completed.

The ESIF bit is automatically cleared to 0 and a transfer request is cleared if the transfer
resumed by setting the DTE bit to 1 during interrupt handling.

For details on interrupts, see section 7.8, Interrupt Sources.

For details on interrupts, see section 7.8, Interrupt Sources.

7.5.8 Priority of Channels

The channels of the DMAC are given following priority levels: channel 0 > channel 1 > channel 2 > channel3. Table 7.6 shows the priority levels among the DMAC channels.

Table 7.6 Priority among DMAC Channels

Channel	Pric
Channel 0	Hig
Channel 1	↑
Channel 2	
Channel 3	Low

The channel having highest priority other than the channel being transferred is selected w transfer is requested from other channels. The selected channel starts the transfer after the being transferred releases the bus. At this time, when a bus master other than the DMAC the bus, the cycle for the bus master is inserted.

In a burst transfer or a block transfer, channels are not switched.

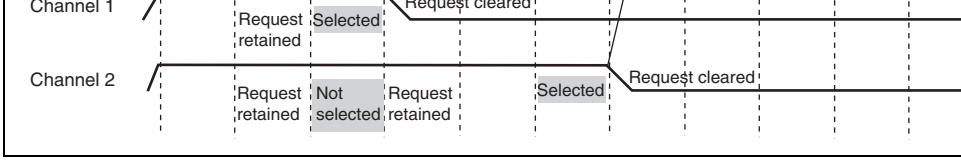


Figure 7.22 Example of Timing for Channel Priority

7.5.9 DMA Basic Bus Cycle

Figure 7.23 shows an examples of signal timing of a basic bus cycle. In figure 7.23, data transferred in words from the 16-bit 2-state access space to the 8-bit 3-state access space the bus mastership is passed from the DMAC to the CPU, data is read from the source a it is written to the destination address. The bus is not released between the read and writ by other bus requests. DMAC bus cycles follows the bus controller settings.

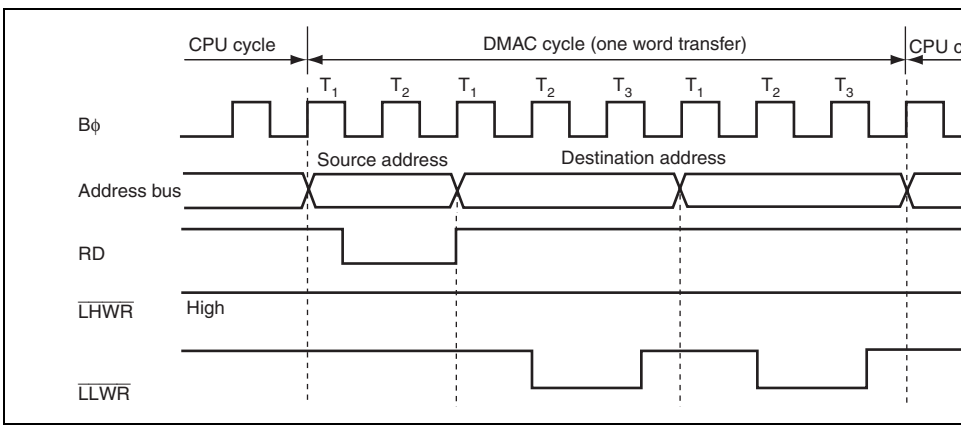


Figure 7.23 Example of Bus Timing of DMA Transfer

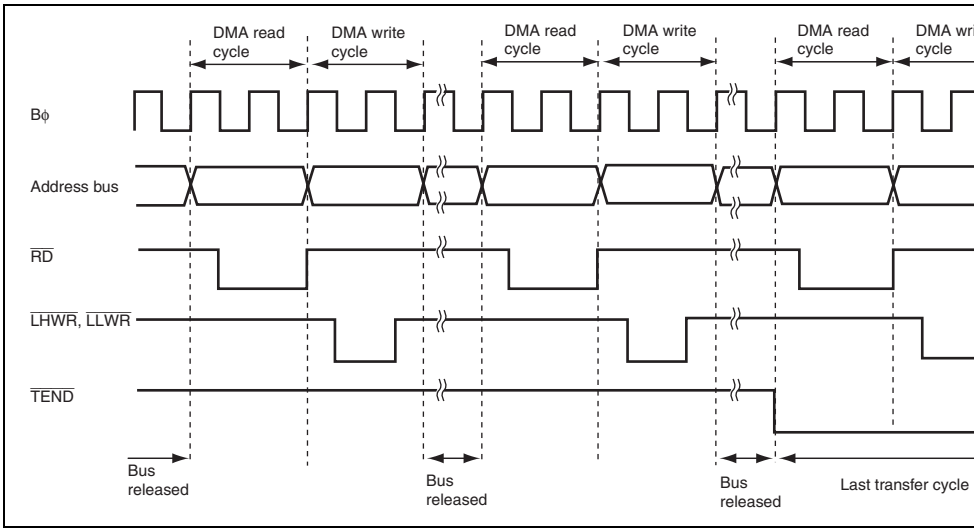


Figure 7.24 Example of Transfer in Normal Transfer Mode by Cycle Stealin

In figures 7.25 and 7.26, the $\overline{\text{TEND}}$ signal output is enabled and data is transferred in lon from the external 16-bit 2-state access space to the 16-bit 2-state access space in normal t mode by cycle stealing.

In figure 7.25, the transfer source (DSAR) is not aligned with a longword boundary and t transfer destination (DDAR) is aligned with a longword boundary.

In figure 7.26, the transfer source (DSAR) is aligned with a longword boundary and the t destination (DDAR) is not aligned with a longword boundary.

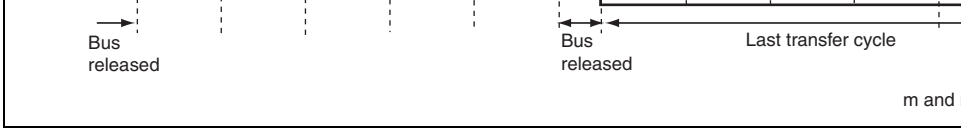


Figure 7.25 Example of Transfer in Normal Transfer Mode by Cycle Steal
(Transfer Source DSAR = Odd Address and Source Address Increment)

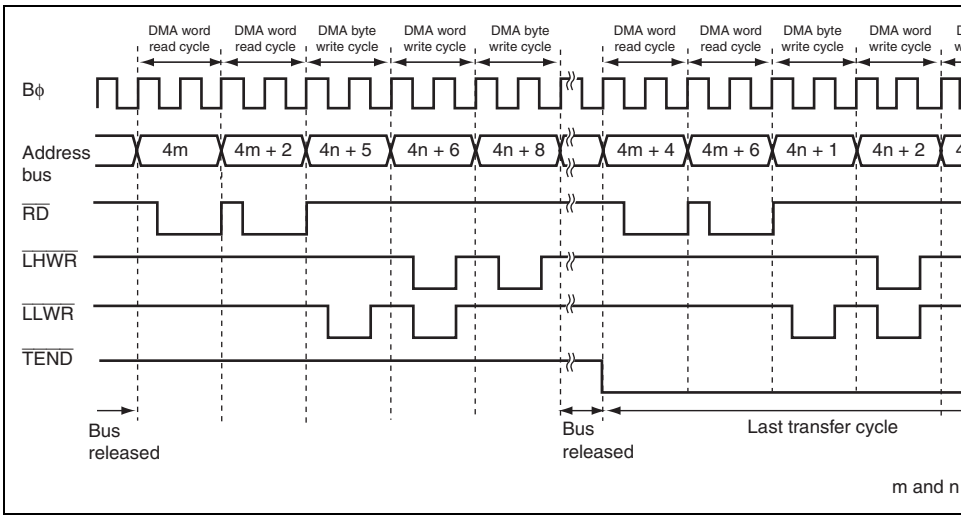


Figure 7.26 Example of Transfer in Normal Transfer Mode by Cycle Steal
(Transfer Destination DDAR = Odd Address and Destination Address Decren

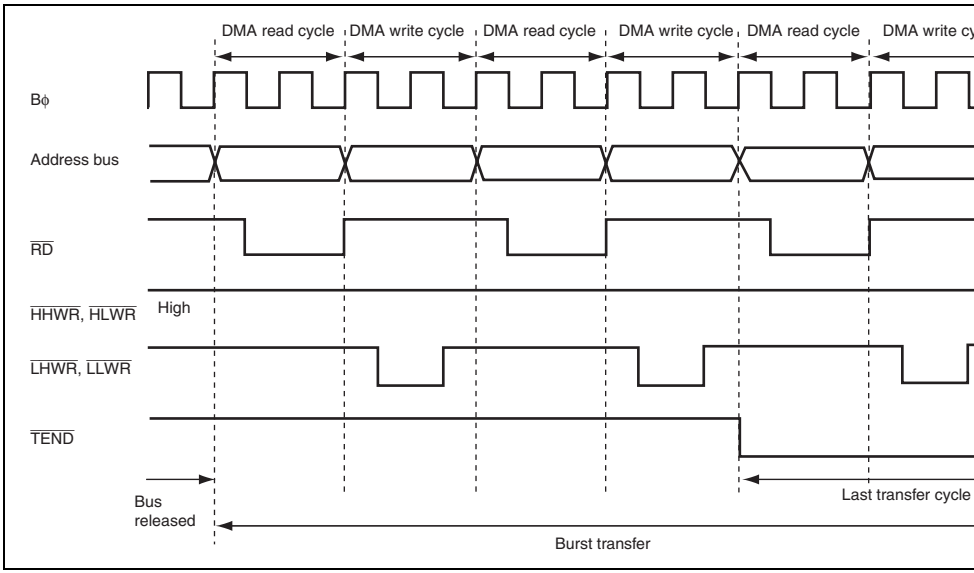


Figure 7.27 Example of Transfer in Normal Transfer Mode by Burst Acces

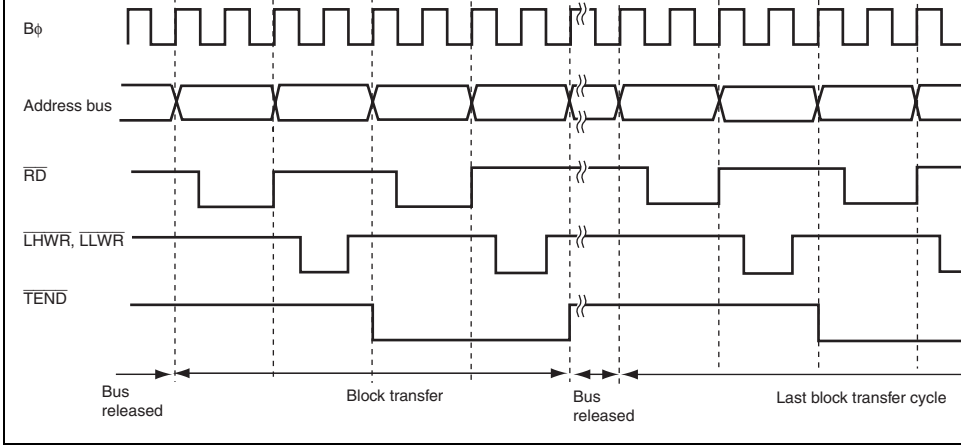
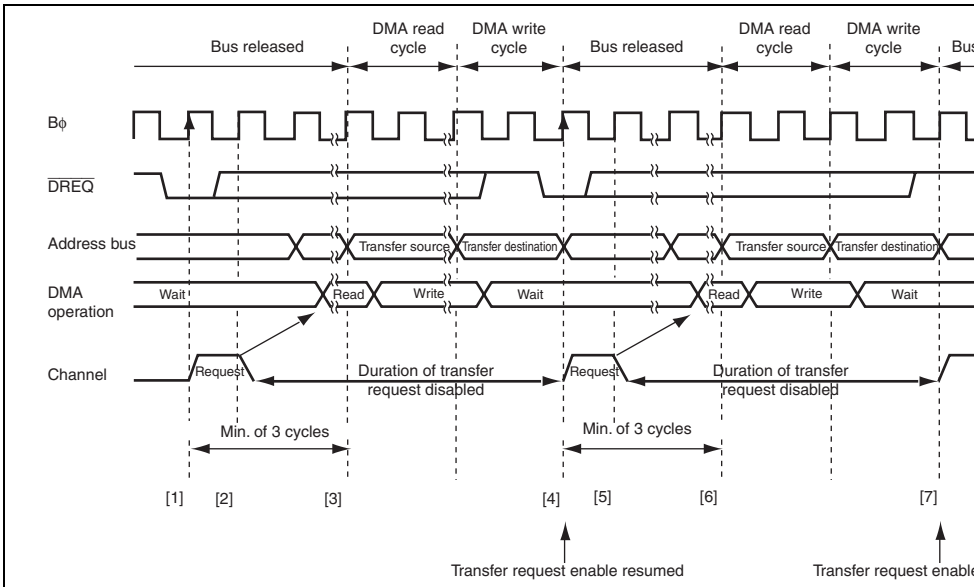


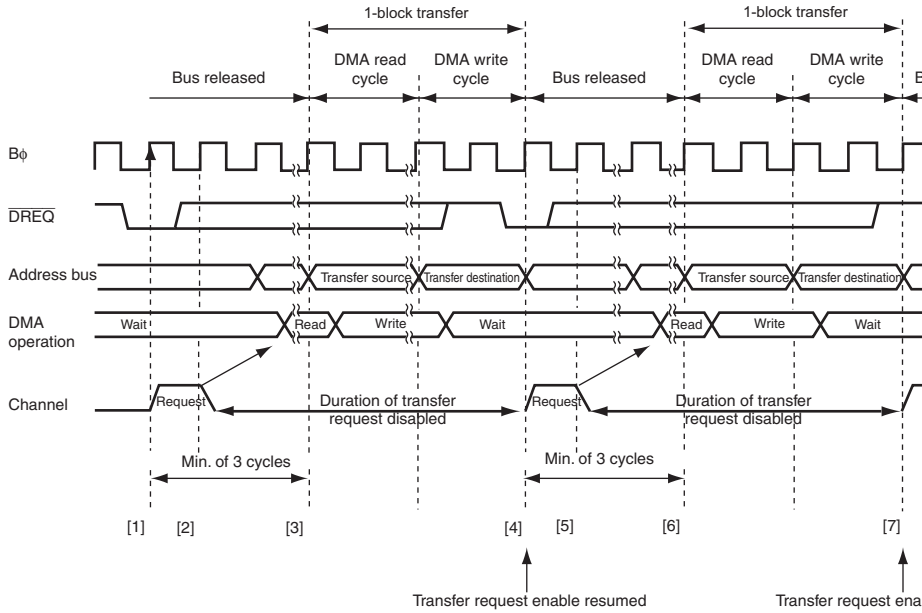
Figure 7.28 Example of Transfer in Block Transfer Mode

receiving the next transfer request resumes and then a low level of the DREQ signal is de
This operation is repeated until the transfer is completed.



[1] After DMA transfer request is enabled, a low level of the DREQ signal is detected at the rising edge of the Bϕ signal and a request is held.
[2][5] The DMAC is activated and the transfer request is cleared.
[3][6] A DMA cycle is started and sampling the DREQ signal at the rising edge of the Bϕ signal is started to detect a high level o DREQ signal.
[4][7] When a high level of the DREQ signal has been detected, transfer request enable is resumed after completion of the write (A low level of the DREQ signal is detected at the rising edge of the Bϕ signal and a transfer request is held. This is the sa

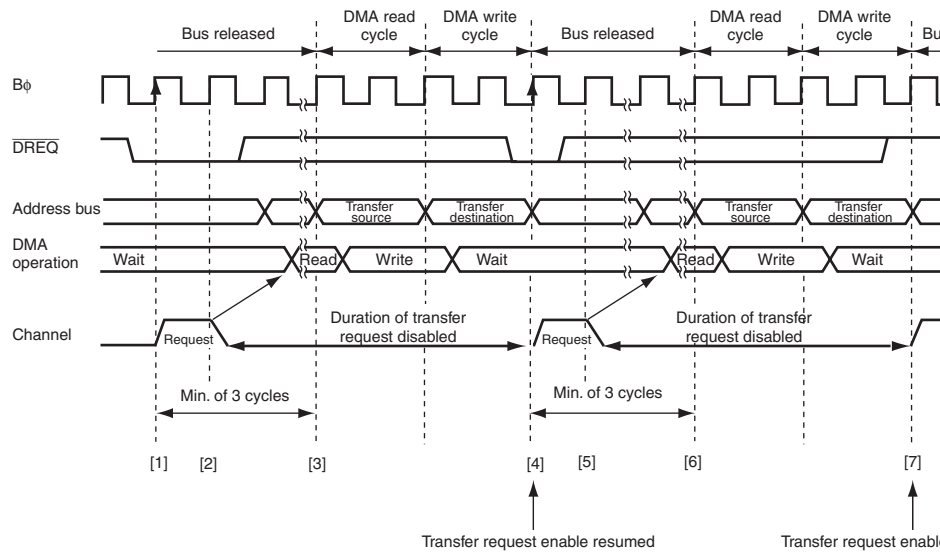
Figure 7.29 Example of Transfer in Normal Transfer Mode Activated by DREQ Falling Edge



[1] After DMA transfer request is enabled, a low level of the $\overline{\text{DREQ}}$ signal is detected at the rising edge of the Bϕ signal anc
request is held.
[2][5] The DMAC is activated and the transfer request is cleared.
[3][6] A DMA cycle is started and sampling the $\overline{\text{DREQ}}$ signal at the rising edge of the Bϕ signal is started to detect a high leve
$\overline{\text{DREQ}}$ signal.
[4][7] When a high level of the $\overline{\text{DREQ}}$ signal has been detected, transfer request enable is resumed after completion of the wr
(A low level of the $\overline{\text{DREQ}}$ signal is detected at the rising edge of the Bϕ signal and a transfer request is held. This is the s

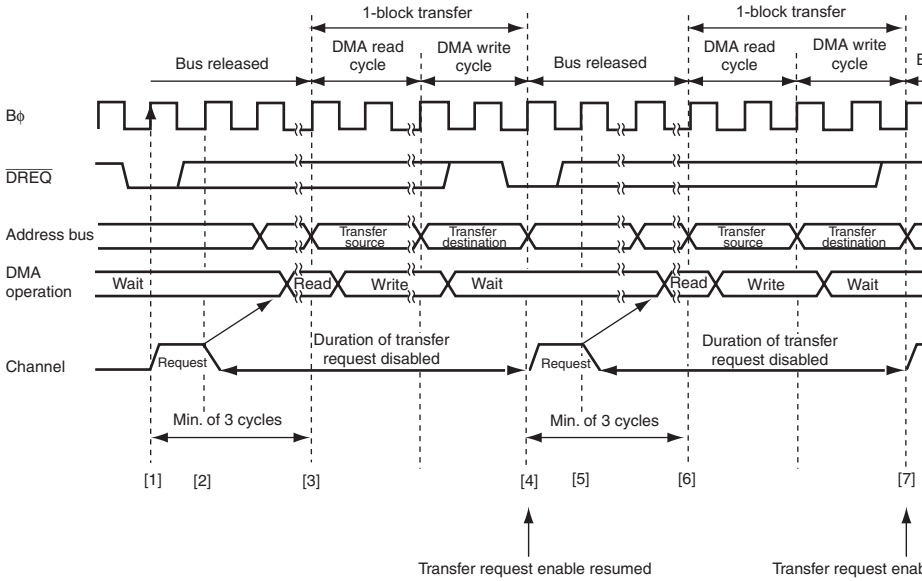
Figure 7.30 Example of Transfer in Block Transfer Mode Activated by $\overline{\text{DREQ}}$ Falling Edge

completed.



[1] After DMA transfer request is enabled, a low level of the $\overline{\text{DREQ}}$ signal is detected at the rising edge of the Bϕ signal and a request is held.
[2][5] The DMAC is activated and the transfer request is cleared.
[3][6] A DMA cycle is started.
[4][7] Transfer request enable is resumed after completion of the write cycle.
(A low level of the $\overline{\text{DREQ}}$ signal is detected at the rising edge of the Bϕ signal and a transfer request is held. This is the sa

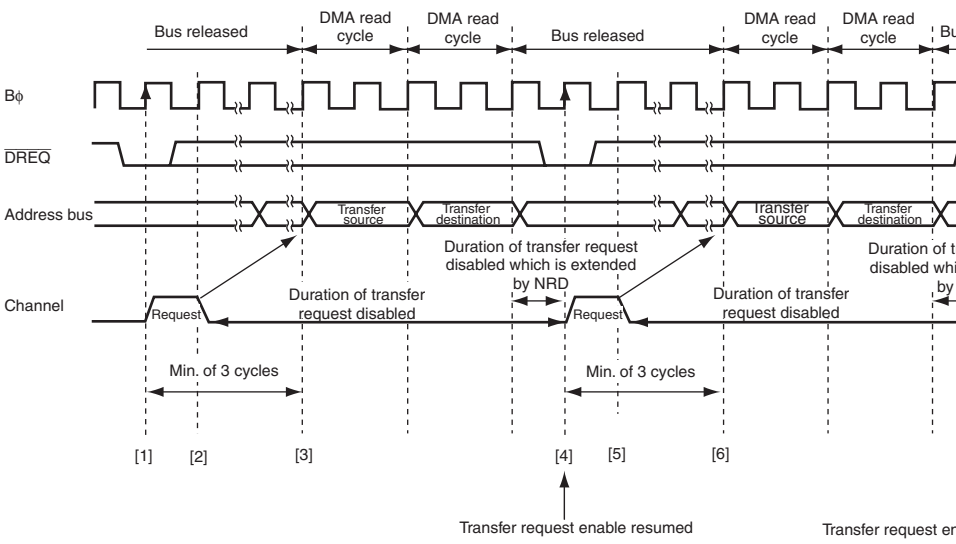
Figure 7.31 Example of Transfer in Normal Transfer Mode Activated by $\overline{\text{DREQ}}$ Low Level



[1] After DMA transfer request is enabled, a low level of the $\overline{\text{DREQ}}$ signal is detected at the rising edge of the Bϕ signal and
request is held.
[2][5] The DMAC is activated and the transfer request is cleared.
[3][6] A DMA cycle is started.
[4][7] Transfer request enable is resumed after completion of the write cycle.
(A low level of the $\overline{\text{DREQ}}$ signal is detected at the rising edge of the Bϕ signal and a transfer request is held. This is the

Figure 7.32 Example of Transfer in Block Transfer Mode Activated by $\overline{\text{DREQ}}$ Low Level

enabled, a transfer request is held in the DMAC. When the DMAC is activated, the transf
request is cleared. Receiving the next transfer request resumes after completion of the wri
and then a low level of the $\overline{\text{DREQ}}$ signal is detected. This operation is repeated until the t
completed.



[1] After DMA transfer request is enabled, a low level of the $\overline{\text{DREQ}}$ signal is detected at the rising edge of the Bϕ signal and a request is held.
[2][5] The DMAC is activated and the transfer request is cleared.
[3][6] A DMA cycle is started.
[4][7] Transfer request enable is resumed one cycle after completion of the write cycle.
(A low level of the $\overline{\text{DREQ}}$ signal is detected at the rising edge of the Bϕ signal and a transfer request is held. This is the sa

Figure 7.33 Example of Transfer in Normal Transfer Mode Activated by $\overline{\text{DREQ}}$ Low Level with NRD = 1

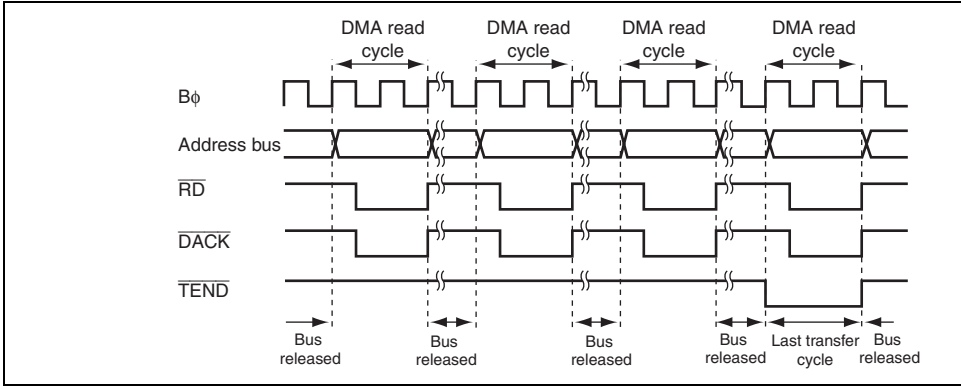


Figure 7.34 Example of Transfer in Single Address Mode (Byte Read)

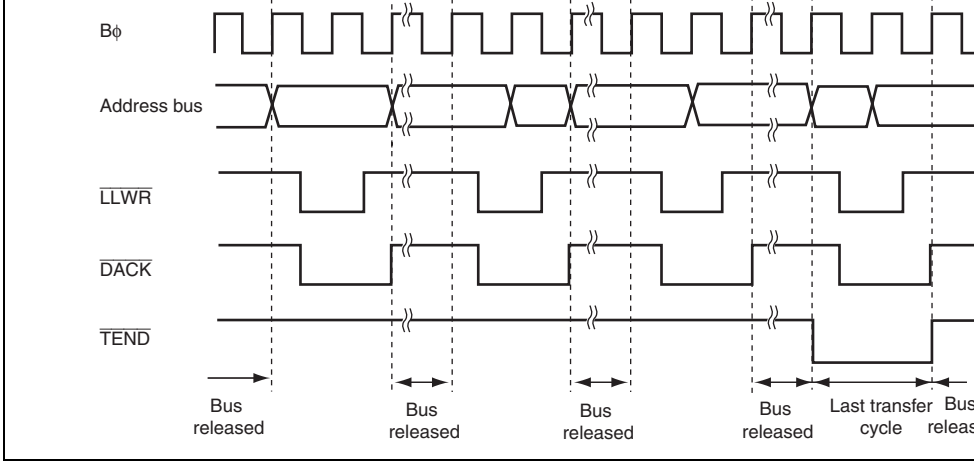


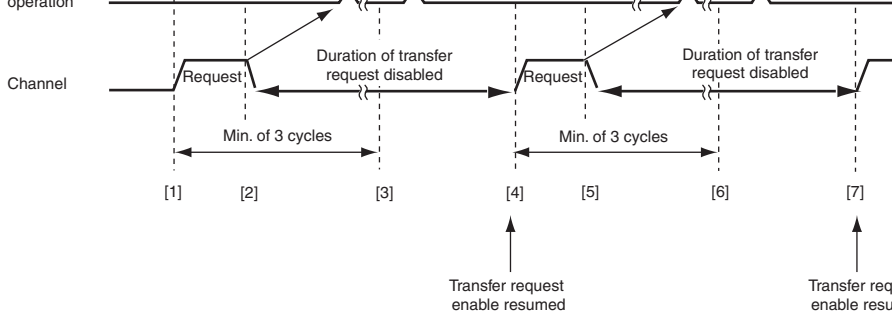
Figure 7.35 Example of Transfer in Single Address Mode (Byte Write)

(3) Activation Timing by $\overline{\text{DREQ}}$ Falling Edge

Figure 7.36 shows an example of single address mode activated by the $\overline{\text{DREQ}}$ signal falli

The $\overline{\text{DREQ}}$ signal is sampled every cycle from the next rising edge of the Bϕ signal imme after the DTE bit write cycle.

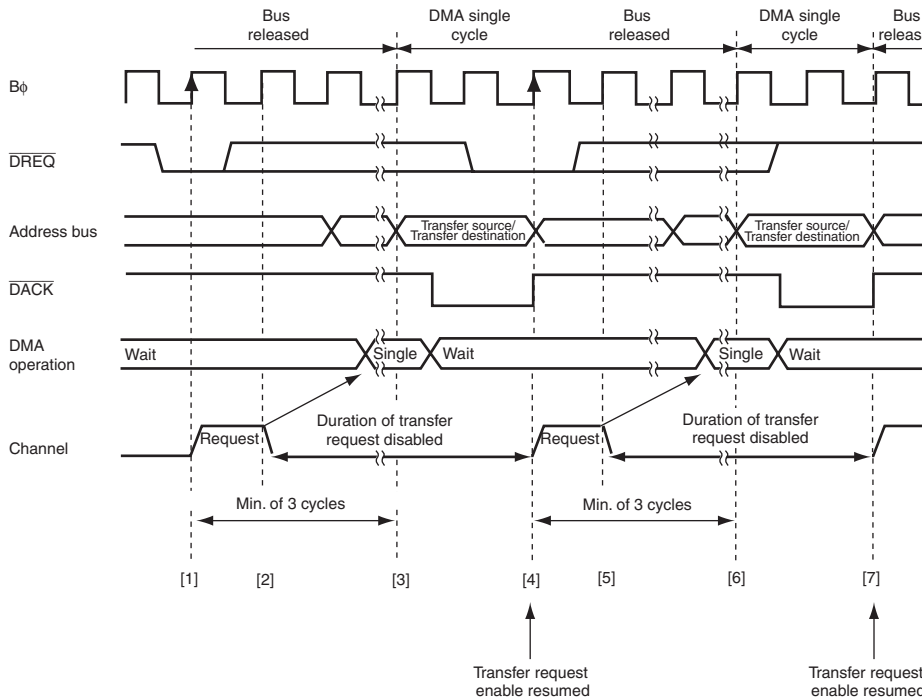
When a low level of the $\overline{\text{DREQ}}$ signal is detected while a transfer request by the $\overline{\text{DREQ}}$ s enabled, a transfer request is held in the DMAC. When the DMAC is activated, the transf request is cleared and starts detecting a high level of the $\overline{\text{DREQ}}$ signal for falling edge de a high level of the $\overline{\text{DREQ}}$ signal has been detected until completion of the single cycle, re the next transfer request resumes and then a low level of the $\overline{\text{DREQ}}$ signal is detected. Th operation is repeated until the transfer is completed.



[1] After DMA transfer request is enabled, a low level of the $\overline{\text{DREQ}}$ signal is detected at the rising edge of the Bϕ signal and a
request is held.
[2][5] The DMAC is activated and the transfer request is cleared.
[3][6] A DMA cycle is started and sampling the $\overline{\text{DREQ}}$ signal at the rising edge of the Bϕ signal is started to detect a high level o
$\overline{\text{DREQ}}$ signal.
[4][7] When a high level of the $\overline{\text{DREQ}}$ signal has been detected, transfer enable is resumed after completion of the write cycle.
(A low level of the $\overline{\text{DREQ}}$ signal is detected at the rising edge of the Bϕ signal and a transfer request is held. This is the sa

Figure 7.36 Example of Transfer in Single Address Mode Activated by $\overline{\text{DREQ}}$ Falling Edge

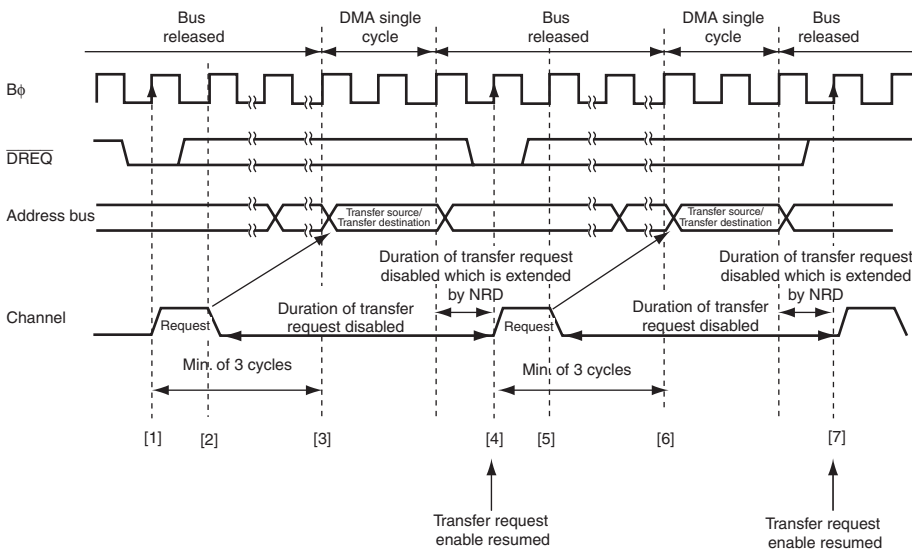
completed.



[1] After DMA transfer request is enabled, a low level of the DREQ signal is detected at the rising edge of the Bϕ signal and a tr
request is held.
[2][5] The DMAC is activated and the transfer request is cleared.
[3][6] A DMA cycle is started.
[4][7] Transfer request enable is resumed after completion of the single cycle.
(A low level of the DREQ signal is detected at the rising edge of the Bϕ signal and a transfer request is held. This is the sam

Figure 7.37 Example of Transfer in Single Address Mode Activated by DREQ Low Level

enabled, a transfer request is held in the DMAC. When the DMAC is activated, the trans
request is cleared. Receiving the next transfer request resumes after one cycle of the tran
request duration inserted by NRD = 1 on completion of the single cycle and then a low l
$\overline{\text{DREQ}}$ signal is detected. This operation is repeated until the transfer is completed.



[1] After DMA transfer request is enabled, a low level of the $\overline{\text{DREQ}}$ signal is detected at the rising edge of the Bϕ signal and
request is held.
[2][5] The DMAC is activated and the transfer request is cleared.
[3][6] A DMA cycle is started.
[4][7] Transfer request enable is resumed one cycle after completion of the single cycle.
(A low level of the $\overline{\text{DREQ}}$ signal is detected at the rising edge of the Bϕ signal and a transfer request is held. This is the s

Figure 7.38 Example of Transfer in Single Address Mode Activated by $\overline{\text{DREQ}}$ Low Level with NRD = 1

(2) Transfer End by Transfer Size Error Interrupt

When the following conditions are satisfied while the TSEIE bit in DMDR is set to 1, a tr
size error occurs and a DMA transfer is terminated. At this time, the DTE bit in DMR is c
0 and the ESIF bit in DMDR is set to 1.

- In normal transfer mode and repeat transfer mode, when the next transfer is requested
  transfer is disabled due to the DTCR value less than the data access size
- In block transfer mode, when the next transfer is requested while a transfer is disablec
  the DTCR value less than the block size

When the TSEIE bit in DMDR is cleared to 0, data is transferred until the DTCR value re
A transfer size error is not generated. Operation in each transfer mode is shown below.

- In normal transfer mode and repeat transfer mode, when the DTCR value is less than
  access size, data is transferred in bytes
- In block transfer mode, when the DTCR value is less than the block size, the specifiec
  data in DTCR is transferred instead of transferring the block size of data. The transfer
  performed in bytes.

(3) Transfer End by Repeat Size End Interrupt

In repeat transfer mode, when the next transfer is requested after completion of a 1-repeat
transfer while the RPTIE bit in DACR is set to 1, a repeat size end interrupt is requested.
the interrupt is requested to complete DMA transfer, the DTE bit in DMDR is cleared to 0
ESIF bit in DMDR is set to 1. Under this condition, setting the DTE bit to 1 resumes the t

In block transfer mode, when the next transfer is requested after completion of a 1-block
transfer, a repeat size end interrupt can be requested.

block transfer, the remaining data is transferred. The transfer is not terminated by an ext
repeat area overflow interrupt unless the current transfer is complete.

(5) Transfer End by Clearing DTE Bit in DMDR

When the DTE bit in DMDR is cleared to 0 by the CPU, a transfer is completed after the
DMA cycle and a DMA cycle in which the transfer request is accepted are completed.

In block transfer mode, a DMA transfer is completed after 1-block data is transferred.

(6) Transfer End by NMI Interrupt

When an NMI interrupt is requested, the DTE bits for all the channels are cleared to 0 an
ERRF bit in DMDR_0 is set to 1. When an NMI interrupt is requested during a DMA tra
transfer is forced to stop. To perform DMA transfer after an NMI interrupt is requested,
ERRF bit to 0 and then set the DTE bits for the channels to 1.

The transfer end timings after an NMI interrupt is requested are shown below.

(a) Normal Transfer Mode and Repeat Transfer Mode

In dual address mode, a DMA transfer is completed after completion of the write cycle f
transfer unit.

In single address mode, a DMA transfer is completed after completion of the bus cycle f
transfer unit.

(b) Block Transfer Mode

A DMA transfer is forced to stop. Since a 1-block size of transfers is not completed, ope
not guaranteed.

In dual address mode, the write cycle corresponding to the read cycle is performed. This
to (a) in normal transfer mode.

transfer is not guaranteed.

7.7 Relationship among DMAC and Other Bus Masters

7.7.1 CPU Priority Control Function Over DMAC

The CPU priority control function over DMAC can be used according to the CPU priorit register (CPUPCR) setting. For details, see section 5.7, CPU Priority Control Function O and DMAC.

The priority level of the DMAC is specified by bits DMAP2 to DMAP0 and can be speci each channel.

The priority level of the CPU is specified by bits CPUP2 to CPUP0. The value of bits CP CPUP0 is updated according to the exception handling priority.

If the CPU priority control is enabled by the CPUPCE bit in CPUPCR, when the CPU ha over the DMAC, a transfer request for the corresponding channel is masked and the trans activated. When another channel has priority over or the same as the CPU, a transfer requ received regardless of the priority between channels and the transfer is activated.

The transfer request masked by the CPU priority control function is suspended. When the channel is given priority over the CPU by changing priority levels of the CPU or channel transfer request is received and the transfer is resumed. Writing 0 to the DTE bit clears th suspended transfer request.

When the CPUPCE bit is cleared to 0, it is regarded as the lowest priority.

a DMA transfer.

In block transfer mode and an auto request transfer by burst access, bus cycles of the DM
transfer are consecutively performed. For this duration, since the DMAC has priority ov
CPU and DTC, accesses to the external space is suspended (the IBCCS bit in the bus co
register 2 (BCR2) is cleared to 0).

When the bus is passed to another channel or an auto request transfer by cycle stealing,
of the DMAC and on-chip bus master are performed alternatively.

When the arbitration function among the DMAC and on-chip bus masters is enabled by
IBCCS bit in BCR2, the bus is used alternatively except the bus cycles which are not se
For details, see section 6, Bus Controller (BSC).

A conflict may occur between external space access of the DMAC and an external bus r
cycle. Even if a burst or block transfer is performed by the DMAC, the transfer is stoppe
temporarily and a cycle of external bus release is inserted by the BSC according to the e
bus priority (when the CPU external access and the DTC external access do not have pri
a DMAC transfer, the transfers are not operated until the DMAC releases the bus).

In dual address mode, the DMAC releases the external bus after the external space write
Since the read and write cycles are not separated, the bus is not released.

An internal space (on-chip memory and internal I/O registers) access of the DMAC and
external bus release cycle may be performed at the same time.

DMTEND2	Transfer end interrupt by channel 2 transfer counter	
DMTEND3	Transfer end interrupt by channel 3 transfer counter	
DMEEND0	Interrupt by channel 0 transfer size error	
	Interrupt by channel 0 repeat size end	
	Interrupt by channel 0 extended repeat area overflow on source address	
	Interrupt by channel 0 extended repeat area overflow on destination address	
DMEEND1	Interrupt by channel 1 transfer size error	
	Interrupt by channel 1 repeat size end	
	Interrupt by channel 1 extended repeat area overflow on source address	
	Interrupt by channel 1 extended repeat area overflow on destination address	
DMEEND2	Interrupt by channel 2 transfer size error	
	Interrupt by channel 2 repeat size end	
	Interrupt by channel 2 extended repeat area overflow on source address	
	Interrupt by channel 2 extended repeat area overflow on destination address	
DMEEND3	Interrupt by channel 3 transfer size error	
	Interrupt by channel 3 repeat size end	
	Interrupt by channel 3 extended repeat area overflow on source address	
	Interrupt by channel 3 extended repeat area overflow on destination address	Lo

Each interrupt is enabled or disabled by the DTIE and ESIE bits in DMDR for the corresp
channel. A DMTEND interrupt is generated by the combination of the DTIF and DTIE bi
DMDR. A DMEEND interrupt is generated by the combination of the ESIF and ESIE bit
DMDR. The DMEEND interrupt sources are not distinguished. The priority among chann
decided by the interrupt controller and it is shown in table 7.7. For details, see section 5, I
Controller.

An interrupt other than the transfer end interrupt by the transfer counter is generated whe
ESIF bit in DMDR is set to 1. The ESIF bit is set to 1 when the conditions are satisfied b
transfer while the enable bit is set to 1.

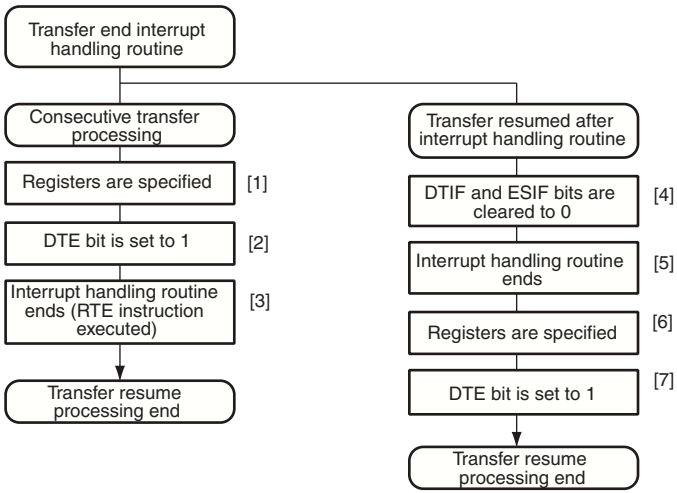
A transfer size error interrupt is generated when the next transfer cannot be performed b
DTCR value is less than the data access size, meaning that the data access size of transfe
be performed. In block transfer mode, the block size is compared with the DTCR value i
transfer error decision.

A repeat size end interrupt is generated when the next transfer is requested after complet
repeat size of transfers in repeat transfer mode. Even when the repeat area is not specifie
address register, the transfer can be stopped periodically according to the repeat size. At
when a transfer end interrupt by the transfer counter is generated, the ESIF bit is set to 1

An interrupt by an extended repeat area overflow on the source and destination addresse
generated when the address exceeds the extended repeat area (overflow). At this time, w
transfer end interrupt by the transfer counter, the ESIF bit is set to 1.

Figure 7.39 is a block diagram of interrupts and interrupt flags. To clear an interrupt, cle
DTIF or ESIF bit in DMDR to 0 in the interrupt handling routine or continue the transfe
setting the DTE bit in DMDR after setting the register. Figure 7.40 shows procedure to r
transfer by clearing a interrupt.

Figure 7.39 Interrupt and Interrupt Sources



[1] Specify the values in the registers such as transfer counter and address register.
[2] Set the DTE bit in DMDR to 1 to resume DMA operation. Setting the DTE bit to 1 automatically clears the D
ESIF bit in DMDR to 0 and an interrupt source is cleared.
[3] End the interrupt handling routine by the RTE instruction.
[4] Read that the DTIF or the ESIF bit in DMDR = 1 and then write 0 to the bit.
[5] Complete the interrupt handling routine and clear the interrupt mask.
[6] Specify the values in the registers such as transfer counter and address register.
[7] Set the DTE bit to 1 to resume DMA operation.

Figure 7.40 Procedure Example of Resuming Transfer by Clearing Interrupt S

enters the module stop state. However, when a transfer for a channel is enabled or w
interrupt is being requested, bit MSTPA13 cannot be set to 1. Clear the DTE bit to 0
DTIF or DTIE bit in DMDR to 0, and then set bit MSTPA13.

When the clock is stopped, the DMAC registers cannot be accessed. However, the fo
register settings are valid in the module stop state. Disable them before entering the
stop state, if necessary.

— TENDE bit in DMDR is 1 (the TEND signal output enabled)

— DACKE bit in DMDR is 1 (the DACK signal output enabled)

3. Activation by $\overline{\text{DREQ}}$ Falling Edge

   The $\overline{\text{DREQ}}$ falling edge detection is synchronized with the DMAC internal operatio

   A. Activation request waiting state: Waiting for detecting the $\overline{\text{DREQ}}$ low level. A tr
      2. is made.

   B. Transfer waiting state: Waiting for a DMAC transfer. A transition to 3. is made.

   C. Transfer prohibited state: Waiting for detecting the $\overline{\text{DREQ}}$ high level. A transitio
      made.

   After a DMAC transfer enabled, a transition to 1. is made. Therefore, the $\overline{\text{DREQ}}$ sig
   sampled by low level detection at the first activation after a DMAC transfer enabled.

4. Acceptation of Activation Source

   At the beginning of an activation source reception, a low level is detected regardless
   setting of $\overline{\text{DREQ}}$ falling edge or low level detection. Therefore, if the $\overline{\text{DREQ}}$ signal i
   low before setting DMDR, the low level is received as a transfer request.

   When the DMAC is activated, clear the $\overline{\text{DREQ}}$ signal of the previous transfer.

- Three transfer modes

  Normal/repeat/block transfer modes selectable

  Transfer source and destination addresses can be selected from increment/decrement
- Short address mode or full address mode selectable
  - Short address mode

    Transfer information is located on a 3-longword boundary

    The transfer source and destination addresses can be specified by 24 bits to selec Mbyte address space directly
  - Full address mode

    Transfer information is located on a 4-longword boundary

    The transfer source and destination addresses can be specified by 32 bits to selec Gbyte address space directly
- Size of data for data transfer can be specified as byte, word, or longword

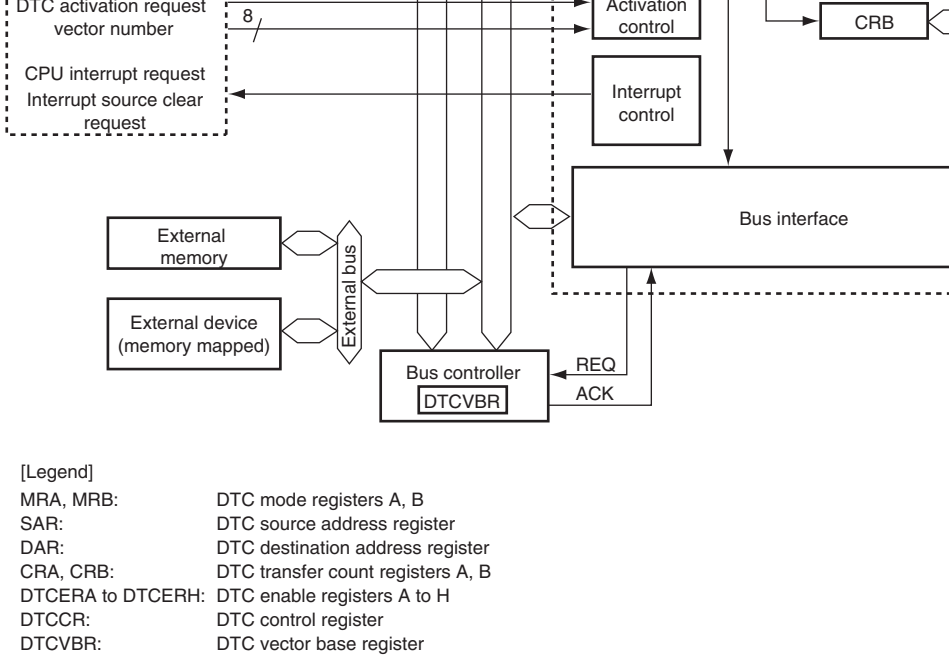
  The bus cycle is divided if an odd address is specified for a word or longword transf

  The bus cycle is divided if address 4n + 2 is specified for a longword transfer.
- A CPU interrupt can be requested for the interrupt that activated the DTC

  A CPU interrupt can be requested after one data transfer completion

  A CPU interrupt can be requested after the specified data transfer completion
- Read skip of the transfer information specifiable
- Writeback skip executed for the fixed transfer source and destination addresses
- Module stop function specifiable

Figure 8.1 shows a block diagram of the DTC. The DTC transfer information can be allo the data area*. When the transfer information is allocated to the on-chip RAM, a 32-bit connects the DTC to the on-chip RAM, enabling 32-bit/1-state reading and writing of th transfer information.



[Legend]

MRA, MRB:	DTC mode registers A, B
SAR:	DTC source address register
DAR:	DTC destination address register
CRA, CRB:	DTC transfer count registers A, B
DTCERA to DTCERH:	DTC enable registers A to H
DTCCR:	DTC control register
DTCVBR:	DTC vector base register

Figure 8.1 Block Diagram of DTC

These six registers MRA, MRB, SAR, DAR, CRA, and CRB cannot be directly accesse
CPU. The contents of these registers are stored in the data area as transfer information. V
DTC activation request occurs, the DTC reads a start address of transfer information tha
in the data area according to the vector address, reads the transfer information, and trans
After the data transfer, it writes a set of updated transfer information back to the data are

- DTC enable registers A to H (DTCERA to DTCERH)
- DTC control register (DTCCR)
- DTC vector base register (DTCVBR)

Bit	Bit Name	Value	R/W	Description
7	MD1	Undefined	—	DTC Mode 1 and 0
6	MD0	Undefined	—	Specify DTC transfer mode. 00: Normal mode 01: Repeat mode 10: Block transfer mode 11: Setting prohibited
5	Sz1	Undefined	—	DTC Data Transfer Size 1 and 0
4	Sz0	Undefined	—	Specify the size of data to be transferred. 00: Byte-size transfer 01: Word-size transfer 10: Longword-size transfer 11: Setting prohibited
3	SM1	Undefined	—	Source Address Mode 1 and 0
2	SM0	Undefined	—	Specify an SAR operation after a data transfer. 0x: SAR is fixed (SAR writeback is skipped) 10: SAR is incremented after a transfer (by 1 when Sz1 and Sz0 = B'00; by 2 when S Sz0 = B'01; by 4 when Sz1 and Sz0 = B'10) 11: SAR is decremented after a transfer (by 1 when Sz1 and Sz0 = B'00; by 2 when S Sz0 = B'01; by 4 when Sz1 and Sz0 = B'10)

Bit	7	6	5	4	3	2	1	
Bit Name	CHNE	CHNS	DISEL	DTS	DM1	DM0	—	
Initial Value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	U
R/W	—	—	—	—	—	—	—	

Bit	Bit Name	Initial Value	R/W	Description
7	CHNE	Undefined	—	DTC Chain Transfer Enable Specifies the chain transfer. For details, see se 8.5.7, Chain Transfer. The chain transfer condit selected by the CHNS bit. 0: Disables the chain transfer 1: Enables the chain transfer
6	CHNS	Undefined	—	DTC Chain Transfer Select Specifies the chain transfer condition. If the foll transfer is a chain transfer, the completion chec specified transfer count is not performed and ac source flag or DTCER is not cleared. 0: Chain transfer every time 1: Chain transfer only when transfer counter = 0

				0: Specifies the destination as repeat or block ar 1: Specifies the source as repeat or block area
3	DM1	Undefined	—	Destination Address Mode 1 and 0
2	DM0	Undefined	—	Specify a DAR operation after a data transfer. 0X: DAR is fixed (DAR writeback is skipped) 10: DAR is incremented after a transfer (by 1 when Sz1 and Sz0 = B'00; by 2 when S Sz0 = B'01; by 4 when Sz1 and Sz0 = B'10) 11: SAR is decremented after a transfer (by 1 when Sz1 and Sz0 = B'00; by 2 when S Sz0 = B'01; by 4 when Sz1 and Sz0 = B'10)
1, 0	—	Undefined	—	Reserved The write value should always be 0.

[Legend]
X: Don't care

SAR cannot be accessed directly from the CPU.

8.2.4 DTC Destination Address Register (DAR)

DAR is a 32-bit register that designates the destination address of data to be transferred
DTC.

In full address mode, 32 bits of DAR are valid. In short address mode, the lower 24 bits
valid and bits 31 to 24 are ignored. At this time, the upper eight bits are filled with the v
bit 23.

If a word or longword access is performed while an odd address is specified in DAR or
longword access is performed while address 4n + 2 is specified in DAR, the bus cycle is
into multiple cycles to transfer data. For details, see section 8.5.1, Bus Cycle Division.

DAR cannot be accessed directly from the CPU.

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

In normal transfer mode, CRA functions as a 16-bit transfer counter (1 to 65,536). It is
decremented by 1 every time data is transferred, and bit DTCEn (n = 15 to 0) correspon
activation source is cleared and then an interrupt is requested to the CPU when the coun
H'0000. The transfer count is 1 when CRA = H'0001, 65,535 when CRA = H'FFFF, and
when CRA = H'0000.

count reaches H'00. The block size is 1 byte (word or longword) when CRAH = CRAL = 255 bytes (words or longwords) when CRAH = CRAL = H'FF, and 256 bytes (words or longwords) when CRAH = CRAL =H'00.

CRA cannot be accessed directly from the CPU.

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 th block transfer mode. It functions as a 16-bit transfer counter (1 to 65,536) that is decreme every time data is transferred, and bit DTCEn (n = 15 to 0) corresponding to the activatio is cleared and then an interrupt is requested to the CPU when the count reaches H'0000. T transfer count is 1 when CRB = H'0001, 65,535 when CRB = H'FFFF, and 65,536 when C H'0000.

CRB is not available in normal and repeat modes and cannot be accessed directly by the C

Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1
Bit Name	DTCE7	DTCE6	DTCE5	DTCE4	DTCE3	DTCE2	DTCE1
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	DTCE15	0	R/W	DTC Activation Enable 15 to 0 Setting this bit to 1 specifies a relevant interrup a DTC activation source. [Clearing conditions] • When writing 0 to the bit to be cleared after • When the DISEL bit is 1 and the data transf ended • When the specified number of transfers hav These bits are not cleared when the DISEL bit i the specified number of transfers have not ende
14	DTCE14	0	R/W	
13	DTCE13	0	R/W	
12	DTCE12	0	R/W	
11	DTCE11	0	R/W	
10	DTCE10	0	R/W	
9	DTCE9	0	R/W	
8	DTCE8	0	R/W	
7	DTCE7	0	R/W	
6	DTCE6	0	R/W	
5	DTCE5	0	R/W	
4	DTCE4	0	R/W	
3	DTCE3	0	R/W	
2	DTCE2	0	R/W	
1	DTCE1	0	R/W	
0	DTCE0	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R/W	Reserved These bits are always read as 0. The write value always be 0.
4	RRS	0	R/W	DTC Transfer Information Read Skip Enable Controls the vector address read and transfer inf read. A DTC vector number is always compared vector number for the previous activation. If the v numbers match and this bit is set to 1, the DTC c transfer is started without reading a vector addre transfer information. If the previous DTC activatio chain transfer, the vector address read and trans information read are always performed. 0: Transfer read skip is not performed. 1: Transfer read skip is performed when the vect numbers match.
3	RCHNE	0	R/W	Chain Transfer Enable After DTC Repeat Transf Enables/disables the chain transfer while transfe (CRAL) is 0 in repeat transfer mode. In repeat transfer mode, the CRAH value is writt CRAL when CRAL is 0. Accordingly, chain trans not occur when CRAL is 0. If this bit is set to 1, t transfer is enabled when CRAH is written to CRA 0: Disables the chain transfer after repeat transfe 1: Enables the chain transfer after repeat transfe
2, 1	—	All 0	R	Reserved These are read-only bits and cannot be modified

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

8.2.9 DTC Vector Base Register (DTCVBR)

DTCVBR is a 32-bit register that specifies the base address for vector table address calc
Bits 31 to 28 and bits 11 to 0 are fixed 0 and cannot be written to. The initial value of D
H'00000000.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	1
Bit Name															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
Bit Name															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	

Locate the transfer information in the data area. The start address of transfer information
located at the address that is a multiple of four (4n). Otherwise, the lower two bits are ign
during access ([1:0] = B'00.) Transfer information can be located in either short address m
(three longwords) or full address mode (four longwords). The DTCMD bit in SYSCR spe
either short address mode (DTCMD = 1) or full address mode (DTCMD = 0). For details
section 3.2.2, System Control Register (SYSCR). Transfer information located in the data
shown in figure 8.2

The DTC reads the start address of transfer information from the vector table according to
activation source, and then reads the transfer information from the start address. Figure 8.
correspondences between the DTC vector address and transfer information.

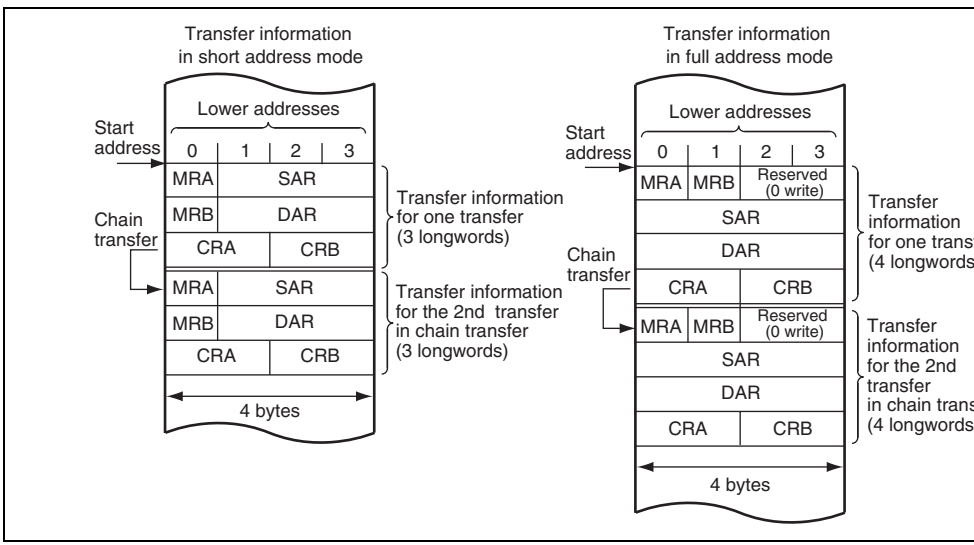


Figure 8.2 Transfer Information on Data Area

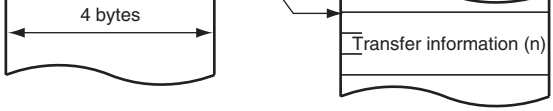


Figure 8.3 Correspondence between DTC Vector Address and Transfer Inform

	IRQ5	69	H'514	DTCEA10	
	IRQ6	70	H'518	DTCEA9	
	IRQ7	71	H'51C	DTCEA8	
	IRQ8	72	H'520	DTCEA7	
	IRQ9	73	H'524	DTCEA6	
	IRQ10	74	H'528	DTCEA5	
	IRQ11	75	H'52C	DTCEA4	
A/D	ADI	86	H'558	DTCEB15	
TPU_0	TGI0A	88	H'560	DTCEB13	
	TGI0B	89	H'564	DTCEB12	
	TGI0C	90	H'568	DTCEB11	
	TGI0D	91	H'56C	DTCEB10	
TPU_1	TGI1A	93	H'574	DTCEB9	
	TGI1B	94	H'578	DTCEB8	
TPU_2	TGI2A	97	H'584	DTCEB7	
	TGI2B	98	H'588	DTCEB6	
TPU_3	TGI3A	101	H'594	DTCEB5	
	TGI3B	102	H'598	DTCEB4	
	TGI3C	103	H'59C	DTCEB3	
	TGI3D	104	H'5A0	DTCEB2	
TPU_4	TGI4A	106	H'5A8	DTCEB1	
	TGI4B	107	H'5AC	DTCEB0	
TPU_5	TGI5A	110	H'5B8	DTCEC15	
	TGI5B	111	H'5BC	DTCEC14	Lo

DMAC	DMTEND0	128	H'600	DTCEC5
	DMTEND1	129	H'604	DTCEC4
	DMTEND2	130	H'608	DTCEC3
	DMTEND3	131	H'60C	DTCEC2
DMAC	DMEEND0	136	H'620	DTCED13
	DMEEND1	137	H'624	DTCED12
	DMEEND2	138	H'628	DTCED11
	DMEEND3	139	H'62C	DTCED10
SCI_0	RXI0	145	H'644	DTCED5
	TXI0	146	H'648	DTCED4
SCI_1	RXI1	149	H'654	DTCED3
	TXI1	150	H'658	DTCED2
SCI_2	RXI2	153	H'664	DTCED1
	TXI2	154	H'668	DTCED0
SCI_4	RXI4	161	H'684	DTCEE13
	TXI4	162	H'688	DTCEE12

Note: * The DTCE bits with no corresponding interrupt are reserved, and the write va
always be 0. To leave software standby mode or all-module-clock-stop mode
interrupt, write 0 to the corresponding DTCE bit.

Table 8.2 shows the DTC transfer modes.

Table 8.2 DTC Transfer Modes

Transfer Mode	Size of Data Transferred at One Transfer Request	Memory Address Increment or Decrement	Tr Co
Normal	1 byte/word/longword	Incremented/decremented by 1, 2, or 4, or fixed	1
Repeat*[1]	1 byte/word/longword	Incremented/decremented by 1, 2, or 4, or fixed	1
Block*[2]	Block size specified by CRAH (1 to 256 bytes/words/longwords)	Incremented/decremented by 1, 2, or 4, or fixed	1

Notes: 1. Either source or destination is specified to repeat area.
2. Either source or destination is specified to block area.
3. After transfer of the specified transfer count, initial state is recovered to continu operation.

Setting the CHNE bit in MRB to 1 makes it possible to perform a number of transfers wit single activation (chain transfer). Setting the CHNS bit in MRB to 1 can also be made to chain transfer performed only when the transfer counter value is 0.

Figure 8.4 shows a flowchart of DTC operation, and table 8.3 summarizes the chain trans conditions (combinations for performing the second and third transfers are omitted).

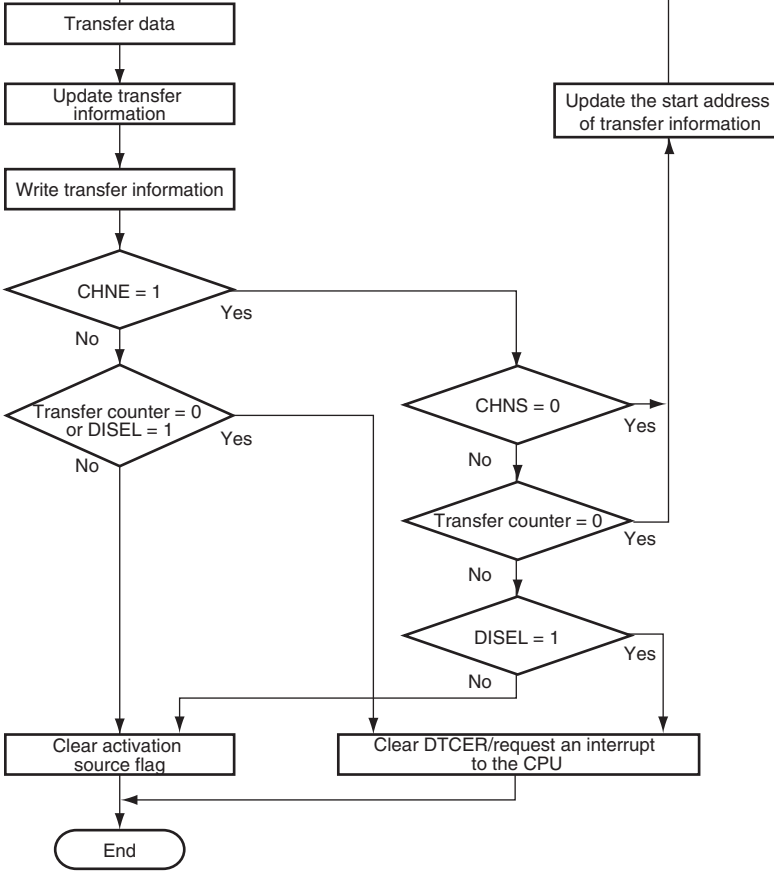


Figure 8.4 Flowchart of DTC Operation

1	1	0	Not 0	—	—		—	Ends at 1st tran
1	1	—	0*[2]	0	—	0	Not 0	Ends at 2nd tra
				0	—	0	0*[2]	Ends at 2nd tra
				0	—	1		Interrupt reques
1	1	1	Not 0	—	—	—	—	Ends at 1st tran
								Interrupt reques

Notes: 1. CRA in normal mode transfer, CRAL in repeat transfer mode, or CRB in block mode
2. When the contents of the CRAH is written to the CRAL in repeat transfer mode

8.5.1 Bus Cycle Division

When the transfer data size is word and the SAR and DAR values are not a multiple of 2, cycle is divided and the transfer data is read from or written to in bytes. Similarly, when t transfer data size is longword and the SAR and DAR values are not a multiple of 4, the bu is divided and the transfer data is read from or written to in words.

Table 8.4 shows the relationship among, SAR, DAR, transfer data size, bus cycle division access data size. Figure 8.5 shows the bus cycle division example.

Table 8.4 Number of Bus Cycle Divisions and Access Size

	Specified Data Size		
SAR and DAR Values	**Byte (B)**	**Word (W)**	**Longword (LW**
Address 4n	1 (B)	1 (W)	1 (LW)
Address 2n + 1	1 (B)	2 (B-B)	3 (B-W-B)
Address 4n + 2	1 (B)	1 (W)	2 (W-W)

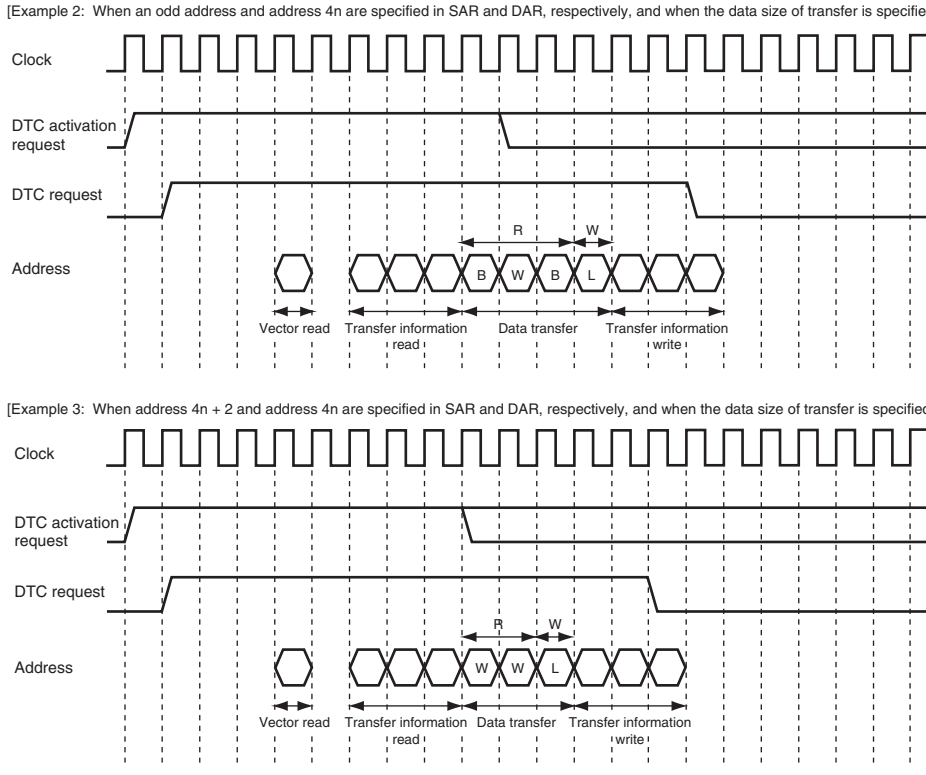
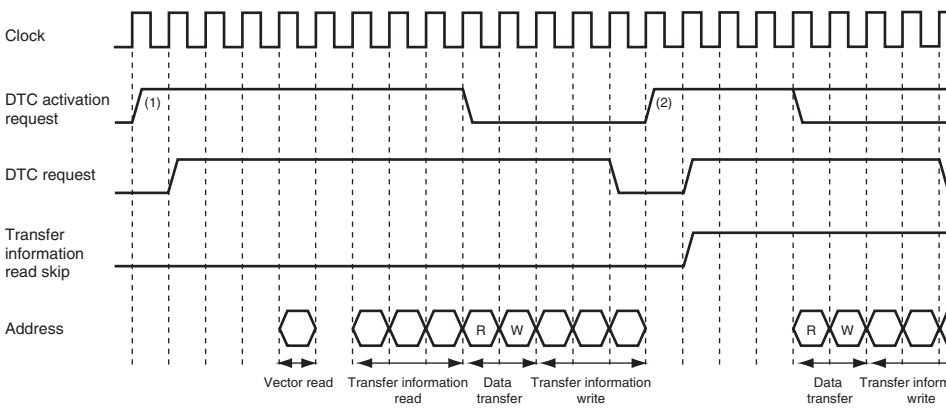


Figure 8.5 Bus Cycle Division Example

cleared to 0, the stored vector number is deleted, and the updated vector table and transfe
information are read at the next activation.



Note: Transfer information read is skipped when the activation sources of (1) and (2) (vector numbers) are the same while RRS = 1.

Figure 8.6 Transfer Information Read Skip Timing

SM1	DM1	SAR	DAR
0	0	Skipped	Skipped
0	1	Skipped	Written back
1	0	Written back	Skipped
1	1	Written back	Written back

8.5.4 Normal Transfer Mode

In normal transfer mode, one operation transfers one byte, one word, or one longword o
From 1 to 65,536 transfers can be specified. The transfer source and destination addresse
specified as incremented, decremented, or fixed. When the specified number of transfers
interrupt can be requested to the CPU.

Table 8.6 lists the register function in normal transfer mode. Figure 8.7 shows the memo
normal transfer mode.

Table 8.6 Register Function in Normal Transfer Mode

Register	Function	Written Back Value
SAR	Source address	Incremented/decremented/fixe
DAR	Destination address	Incremented/decremented/fixe
CRA	Transfer count A	CRA – 1
CRB	Transfer count B	Not updated

Note: * Transfer information writeback is skipped.

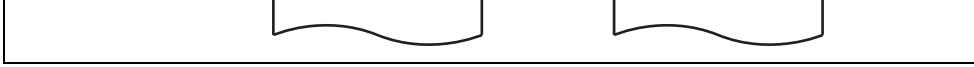


Figure 8.7 Memory Map in Normal Transfer Mode

8.5.5 Repeat Transfer Mode

In repeat transfer mode, one operation transfers one byte, one word, or one longword of d
the DTS bit in MRB, either the source or destination can be specified as a repeat area. Fro
256 transfers can be specified. When the specified number of transfers ends, the transfer c
and address register specified as the repeat area is restored to the initial state, and transfer
repeated. The other address register is then incremented, decremented, or left fixed. In rep
transfer mode, the transfer counter (CRAL) is updated to the value specified in CRAH wh
CRAL becomes H'00. Thus the transfer counter value does not reach H'00, and therefore
interrupt cannot be requested when DISEL = 0.

Table 8.7 lists the register function in repeat transfer mode. Figure 8.8 shows the memory
repeat transfer mode.

CRAH	Transfer count storage	CRAH	CRAH
CRAL	Transfer count A	CRAL – 1	CRAH
CRB	Transfer count B	Not updated	Not updated

Note: * Transfer information writeback is skipped.

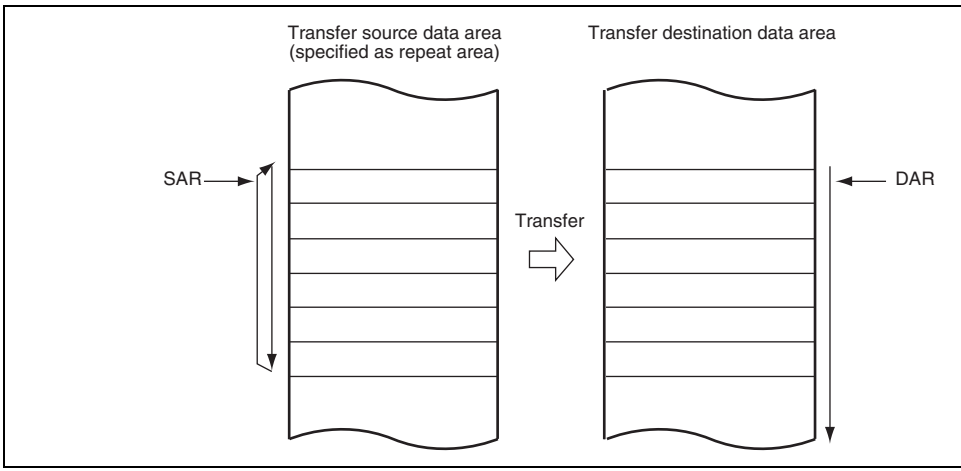


Figure 8.8 Memory Map in Repeat Transfer Mode (When Transfer Source is Specified as Repeat Area)

Table 8.8 lists the register function in block transfer mode. Figure 8.9 shows the memory block transfer mode.

Table 8.8 Register Function in Block Transfer Mode

Register	Function	Written Back Value
SAR	Source address	DTS =0: Incremented/decremented/fixed*
		DTS = 1: SAR initial value
DAR	Destination address	DTS = 0: DAR initial value
		DTS =1: Incremented/decremented/fixed*
CRAH	Block size storage	CRAH
CRAL	Block size counter	CRAH
CRB	Block transfer counter	CRB – 1

Note: * Transfer information writeback is skipped.

Figure 8.9 Memory Map in Block Transfer Mode (When Transfer Destination is Specified as Block Area)

8.5.7 Chain Transfer

Setting the CHNE bit in MRB to 1 enables a number of data transfers to be performed consecutively in response to a single transfer request. Setting the CHNE and CHNS bits set to 1 enables a chain transfer only when the transfer counter reaches 0. SAR, DAR, C MRA, and MRB, which define data transfers, can be set independently. Figure 8.10 sho chain transfer operation.

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

In repeat transfer mode, setting the RCHNE bit in DTCCR and the CHNE and CHNS bi to 1 enables a chain transfer after transfer with transfer counter = 1 has been completed.

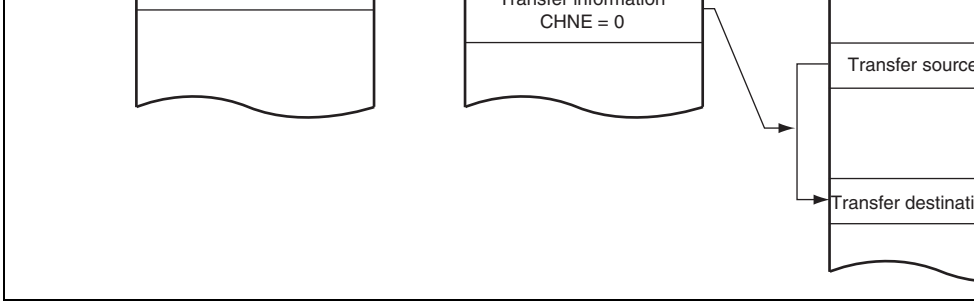


Figure 8.10 Operation of Chain Transfer

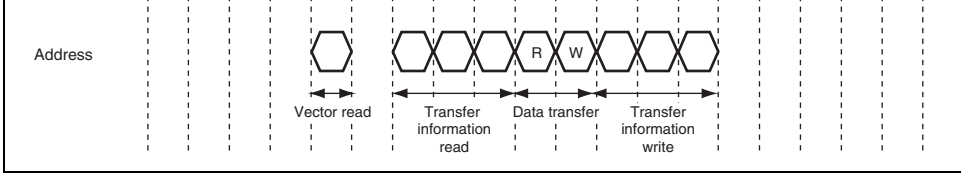


Figure 8.11 DTC Operation Timing
(Example of Short Address Mode in Normal Transfer Mode or Repeat Transfer

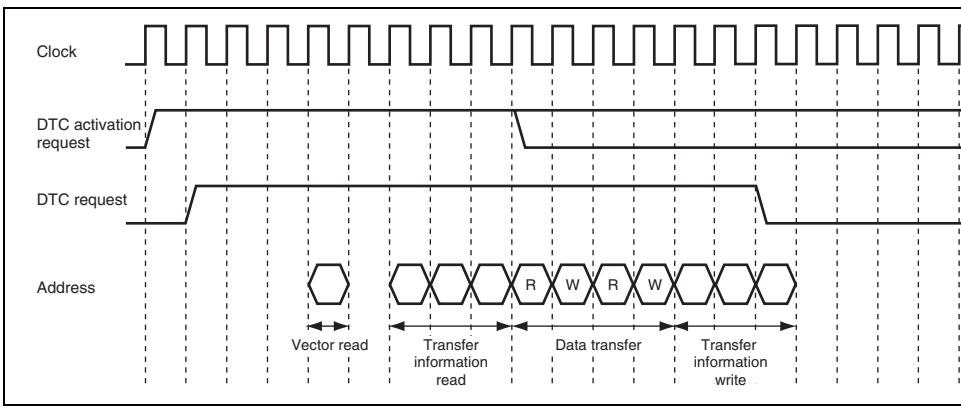


Figure 8.12 DTC Operation Timing
(Example of Short Address Mode in Block Transfer Mode with Block Size o

Figure 8.13 DTC Operation Timing (Example of Short Address Mode in Chain T

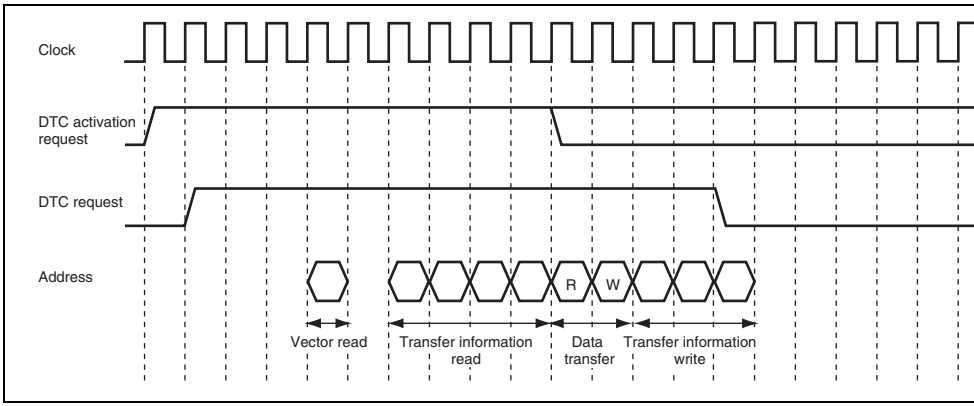


Figure 8.14 DTC Operation Timing
(Example of Full Address Mode in Normal Transfer Mode or Repeat Transfer M

Normal	1	0*[1]	4*[2]	3*[3]	0*[1]	3*[2,3]	2*[4]	1*[5]	3*[6]	2*[7]	1	3*[6]	2*[7]	1
Repeat	1	0*[1]	4*[2]	3*[3]	0*[1]	3*[2,3]	2*[4]	1*[5]	3*[6]	2*[7]	1	3*[6]	2*[7]	1
Block transfer	1	0*[1]	4*[2]	3*[3]	0*[1]	3*[2,3]	2*[4]	1*[5]	3•P*[6]	2•P*[7]	1•P	3•P*[6]	2•P*[7]	1•P

[Legend]

P: Block size (CRAH and CRAL value)

Note:
1. When transfer information read is skipped
2. In full address mode operation
3. In short address mode operation
4. When the SAR or DAR is in fixed mode
5. When the SAR and DAR are in fixed mode
6. When a longword is transferred while an odd address is specified in the addr register
7. When a word is transferred while an odd address is specified in the address r when a longword is transferred while address 4n + 2 is specified

Byte data read S_L	1	1	2	2	2	2	3 + m	2
Word data read S_L	1	1	4	2	2	4	4 + 2m	2
Longword data read S_L	1	1	8	4	2	8	12 + 4m	4
Byte data write S_M	1	1	2	2	2	2	3 + m	2
Word data write S_M	1	1	4	2	2	4	4 + 2m	2
Longword data write S_M	1	1	8	4	2	8	12 + 4m	4
Internal operation S_N	1							

[Legend]
m: Number of wait cycles 0 to 7 (For details, see section 6, Bus Controller (BSC).)

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

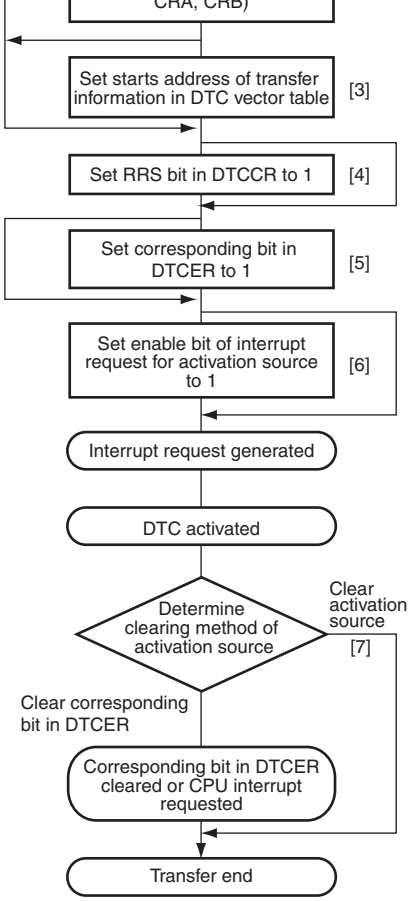
$$\text{Number of execution cycles} = I \cdot S_I + \Sigma (J \cdot S_J + K \cdot S_K + L \cdot S_L + M \cdot S_M) + N \cdot S$$

8.5.10 DTC Bus Release Timing

The DTC requests the bus mastership to the bus arbiter when an activation request occurs
DTC releases the bus after a vector read, transfer information read, a single data transfer,
transfer information writeback. The DTC does not release the bus during transfer informa
read, single data transfer, or transfer information writeback.

8.5.11 DTC Priority Level Control to the CPU

The priority of the DTC activation sources over the CPU can be controlled by the CPU pr
level specified by bits CPUP2 to CPUP0 in CPUPCR and the DTC priority level specifie
DTCP2 to DTCP0. For details, see section 5, Interrupt Controller.



information in the data area. For details on setting tra
information, see section 8.2, Register Descriptions. Fo
on location of transfer information, see section 8.4, Lo
Transfer Information and DTC Vector Table.

[3] Set the start address of the transfer information in the
vector table. For details on setting DTC vector table, s
8.4, Location of Transfer Information and DTC Vector

[4] Setting the RRS bit to 1 performs a read skip of secon
later transfer information when the DTC is activated c
tively by the same interrupt source. Setting the RRS b
always allowed. However, the value set during transfe
valid from the next transfer.

[5] Set the bit in DTCER corresponding to the DTC activa
interrupt source to 1. For the correspondence of intern
DTCER, refer to table 8.1. The bit in DTCER may be s
the second or later transfer. In this case, setting the bi
needed.

[6] Set the enable bits for the interrupt sources to be use
activation sources to 1. The DTC is activated when an
used as an activation source is generated. For details
settings of the interrupt enable bits, see the correspon
descriptions of the corresponding module.

[7] After the end of one data transfer, the DTC clears the
source flag or clears the corresponding bit in DTCER
requests an interrupt to the CPU. The operation after
depends on the transfer information. For details, see s
8.2, Register Descriptions and figure 8.4.

Figure 8.15 DTC with Interrupt Activation

the data will be received in DAR, and 128 (H'0080) in CRA. CRB can be set to any va

2. Set the start address of the transfer information for an RXI interrupt at the DTC vecto
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 end (RXI) interrupt. Since the generation of a receive error during the SCI reception c 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 RXI interrupt is generated, and the DTC is activated. The receive data is transferred fr to RAM by the DTC. DAR is incremented and CRA is decremented. The RDRF flag automatically cleared to 0.
6. When CRA becomes 0 after the 128 data transfers have ended, the RDRF flag is held DTCE bit is cleared to 0, and an RXI interrupt request is sent to the CPU. Terminatio processing should be performed in the interrupt handling routine.

MD0 = 1), and word size (Sz1 = 0, Sz0 = 1). Set the source side as a repeat area (DT
Set MRB to chain transfer mode (CHNE = 1, CHNS = 0, DISEL = 0). Set the data ta
address in SAR, the NDRH address in DAR, and the data table size in CRAH and C
CRB can be set to any value.

2. Perform settings for transfer to the TPU's TGR. Set MRA to source address increme
(SM1 = 1, SM0 = 0), fixed destination address (DM1 = DM0 = 0), normal mode (M
= 0), and word size (Sz1 = 0, Sz0 = 1). Set the data table start address in SAR, the T
address in DAR, and the data table size in CRA. CRB can be set to any value.
3. Locate the TPU transfer information consecutively after the NDR transfer informatio
4. Set the start address of the NDR transfer information to the DTC vector address.
5. Set the bit corresponding to the TGIA interrupt in DTCER to 1.
6. Set TGRA as an output compare register (output disabled) with TIOR, and enable th
interrupt with TIER.
7. Set the initial output value in PODR, and the next output value in NDR. Set bits in D
NDER for which output is to be performed to 1. Using PCR, select the TPU compare
be used as the output trigger.
8. Set the CST bit in TSTR to 1, and start the TCNT count operation.
9. Each time a TGRA compare match occurs, the next output value is transferred to ND
set value of the next output trigger period is transferred to TGRA. The activation sou
flag is cleared.
10. When the specified number of transfers are completed (the TPU transfer CRA value
TGFA flag is held at 1, the DTCE bit is cleared to 0, and a TGIA interrupt request is
CPU. Termination processing should be performed in the interrupt handling routine.

2. Prepare the upper 8-bit addresses of the start addresses for 65,536-transfer units for th data transfer in a separate area (in ROM, etc.). For example, if the input buffer is conf addresses H'200000 to H'21FFFF, prepare H'21 and H'20.
3. For the second transfer, set repeat transfer mode (with the source side as the repeat ar setting the transfer destination address for the first data transfer. Use the upper eight b DAR in the first transfer information area as the transfer destination. Set CHNE = DIS If the above input buffer is specified as H'200000 to H'21FFFF, set the transfer counte
4. Execute the first data transfer 65536 times by means of interrupts. When the transfer c for the first data transfer reaches 0, the second data transfer is started. Set the upper ei of the transfer source address for the first data transfer to H'21. The lower 16 bits of th transfer destination address of the first data transfer and the transfer counter are H'000
5. Next, execute the first data transfer the 65536 times specified for the first data transfe means of interrupts. When the transfer counter for the first data transfer reaches 0, the data transfer is started. Set the upper eight bits of the transfer source address for the fi transfer to H'20. The lower 16 bits of the transfer destination address of the first data t and the transfer counter are H'0000.
6. Steps 4 and 5 are repeated endlessly. As repeat mode is specified for the second data t no interrupt request is sent to the CPU.

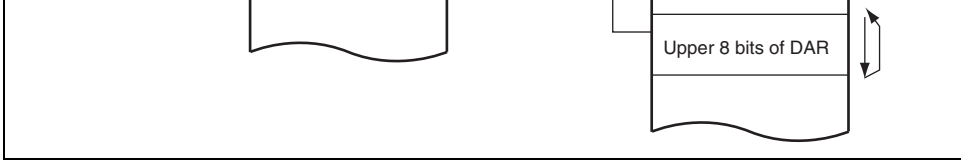


Figure 8.16 Chain Transfer when Counter = 0

8.8 Interrupt Sources

An interrupt request is issued to the CPU when the DTC finishes the specified number o
transfers or a data transfer for which the DISEL bit was set to 1. In the case of interrupt
the interrupt set as the activation source is generated. These interrupts to the CPU are su
CPU mask level and priority level control in the interrupt controller.

Transfer information can be located in on-chip RAM. In this case, the RAME bit in SYS
not be cleared to 0.

8.9.3 DMAC Transfer End Interrupt

When the DTC is activated by a DMAC transfer end interrupt, the DTE bit of DMDR is
controlled by the DTC but its value is modified with the write data regardless of the trans
counter value and DISEL bit setting. Accordingly, even if the DTC transfer counter value
becomes 0, no interrupt request may be sent to the CPU in some cases.

8.9.4 DTCE Bit Setting

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

8.9.5 Chain Transfer

When chain transfer is used, clearing of the activation source or DTCER is performed wh
last of the chain of data transfers is executed. SCI and A/D converter interrupt/activation
are cleared when the DTC reads from or writes to the relevant register.

Therefore, when the DTC is activated by an interrupt or activation source, if a read/write
relevant register is not included in the last chained data transfer, the interrupt or activation
will be retained.

When IDCS = 1 and the DMAC is used, clear the IDCS bit to 0 and then set to 1 again modifying the DTC transfer information in the CPU exception handling routine initiated transfer end interrupt.

8.9.8 Endian Format

The DTC supports big and little endian formats. The endian formats used when transfer information is written to and when transfer information is read from by the DTC must b same.

Ports 2 and F include an open-drain control register (ODR) that controls on/off of the ou
buffer PMOSs.

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

All of the I/O ports can drive Darlington transistors when functioning as output ports.

Ports 2 and 3 are Schmitt-trigger inputs. Schmitt-trigger inputs for other ports are enable
used as the $\overline{\text{IRQ}}$, TPU, or TMR inputs.

and IRQ inputs

4	P14	$\overline{\text{DREQ1}}$-A/ $\overline{\text{IRQ4}}$-A/ TCLKA-B	—	$\overline{\text{IRQ4}}$-A, TCLKA-B
3	P13	$\overline{\text{ADTRG0}}$/ $\overline{\text{IRQ3}}$-A	—	$\overline{\text{IRQ3}}$-A
2	P12/SCK2	$\overline{\text{IRQ2}}$-A	$\overline{\text{DACK0}}$-A	$\overline{\text{IRQ2}}$-A
1	P11	RxD2/ $\overline{\text{IRQ1}}$-A	$\overline{\text{TEND0}}$-A	$\overline{\text{IRQ1}}$-A
0	P10	$\overline{\text{DREQ0}}$-A/ $\overline{\text{IRQ0}}$-A	TxD2	$\overline{\text{IRQ0}}$-A

				TMCI1
4	P24/ TIOCB4/ SCK1	TIOCA4/ TMRI1	PO4	P24, TIOCB4, TIOCA4, TMRI1
3	P23/ TIOCD3	$\overline{\text{IRQ11}}$-A/ TIOCC3	PO3	All input functions
2	P22/ TIOCC3	$\overline{\text{IRQ10}}$-A	PO2/TMO0/ TxD0	All input functions
1	P21/ TIOCA3	TMCI0/ RxD0/ $\overline{\text{IRQ9}}$-A	PO1	P21, $\overline{\text{IRQ9}}$-A, TIOCA3, TMCI0
0	P20/ TIOCB3/ SCK0	TIOCA3/ TMRI0/ $\overline{\text{IRQ8}}$-A	PO0	P20, $\overline{\text{IRQ8}}$-A, TIOCB3, TIOCA3, TMRI0

			TIOCA1		TEND1-B	functions	
		3	P33/ TIOCD0	TIOCC0/ TCLKB-A $\overline{\text{DREQ1}}$-B	PO11	All input functions	
		2	P32/ TIOCC0	TCLKA-A	PO10/ $\overline{\text{DACK0}}$-B	All input functions	
		1	P31/ TIOCB0	TIOCA0	PO9/ $\overline{\text{TEND0}}$-B	All input functions	
		0	P30/ TIOCA0	$\overline{\text{DREQ0}}$-B	PO8	All input functions	
Port 5	General input port also functioning as A/D converter inputs and D/A converter outputs	7	—	P57/AN7 $\overline{\text{IRQ7}}$-B	DA1	$\overline{\text{IRQ7}}$-B	—
		6	—	P56/AN6 $\overline{\text{IRQ6}}$-B	DA0	$\overline{\text{IRQ6}}$-B	
		5	—	P55/AN5 $\overline{\text{IRQ5}}$-B	—	$\overline{\text{IRQ5}}$-B	
		4	—	P54/AN4 $\overline{\text{IRQ4}}$-B	—	$\overline{\text{IRQ4}}$-B	
		3	—	P53/AN3 $\overline{\text{IRQ3}}$-B	—	$\overline{\text{IRQ3}}$-B	
		2	—	P52/AN2 $\overline{\text{IRQ2}}$-B	—	$\overline{\text{IRQ2}}$-B	
		1	—	P51/AN1 $\overline{\text{IRQ1}}$-B	—	$\overline{\text{IRQ1}}$-B	
		0	—	P50/AN0 $\overline{\text{IRQ0}}$-B	—	$\overline{\text{IRQ0}}$-B	

				$\overline{\text{IRQ11}}$-B				
		2	P62/SCK4	$\overline{\text{IRQ10}}$-B	TMO2/ $\overline{\text{DACK2}}$	$\overline{\text{IRQ10}}$-B		
		1	P61	TMCI2/ RxD4/ $\overline{\text{IRQ9}}$-B	$\overline{\text{TEND2}}$	TMCI2, $\overline{\text{IRQ9}}$-B		
		0	P60	TMRI2/ $\overline{\text{DREQ2}}$/ $\overline{\text{IRQ8}}$-B	TxD4	TMRI2, $\overline{\text{IRQ8}}$-B		
Port A	General I/O port also functioning as system clock output and bus control I/Os	7	—	PA7	Bϕ	—	—	
		6	PA6	—	$\overline{\text{AS}}$/$\overline{\text{AH}}$/ $\overline{\text{BS}}$-B			
		5	PA5	—	$\overline{\text{RD}}$			
		4	PA4	—	$\overline{\text{LHWR}}$/$\overline{\text{LUB}}$			
		3	PA3	—	$\overline{\text{LLWR}}$/$\overline{\text{LLB}}$			
		2	PA2	$\overline{\text{BREQ}}$/ $\overline{\text{WAIT}}$	—			
		1	PA1	—	$\overline{\text{BACK}}$/ (RD/$\overline{\text{WR}}$)			
		0	PA0	—	$\overline{\text{BREQO}}$/ $\overline{\text{BS}}$-A			

		2	PB2	—	$\overline{CS2}$-A/ $\overline{CS6}$-A			
		1	PB1	—	$\overline{CS1}$/ $\overline{CS2}$-B/ $\overline{CS5}$-A/ $\overline{CS6}$-B/ $\overline{CS7}$-B			
		0	PB0	—	$\overline{CS0}$/$\overline{CS4}$/ $\overline{CS5}$-B			
Port D	General I/O port also functioning as address outputs	7	PD7	—	A7	—	O	
		6	PD6	—	A6			
		5	PD5	—	A5			
		4	PD4	—	A4			
		3	PD3	—	A3			
		2	PD2	—	A2			
		1	PD1	—	A1			
		0	PD0	—	A0			
Port E	General I/O port also functioning as address outputs	7	PE7	—	A15	—	O	
		6	PE6	—	A14			
		5	PE5	—	A13			
		4	PE4	—	A12			
		3	PE3	—	A11			
		2	PE2	—	A10			
		1	PE1	—	A9			
		0	PE0	—	A8			

		1	PF1	—	A17		
		0	PF0	—	A16		
Port H	General I/O port also functioning as bi-directional data bus	7	PH7/D7*2	—	—	—	O
		6	PH6/D6*2	—	—		
		5	PH5/D5*2	—	—		
		4	PH4/D4*2	—	—		
		3	PH3/D3*2	—	—		
		2	PH2/D2*2	—	—		
		1	PH1/D1*2	—	—		
		0	PH0/D0*2	—	—		
Port I	General I/O port also functioning as bi-directional data bus	7	PI7/D15*2	—	—	—	O
		6	PI6/D14*2	—	—		
		5	PI5/D13*2	—	—		
		4	PI4/D12*2	—	—		
		3	PI3/D11*2	—	—		
		2	PI2/D10*2	—	—		
		1	PI1/D9*2	—	—		
		0	PI0/D8*2	—	—		

Notes: 1. Pins without Schmitt-trigger input buffer have CMOS input buffer.
2. Addresses are also output when accessing to the address/data multiplexed I/

Port 3	8	O	O	O	O	—	—
Port 5	8	—	—	O	O	—	—
Port 6*[1]	6	O	O	O	O	—	—
Port A	8	O	O	O	O	—	—
Port B*[2]	4	O	O	O	O	—	—
Port D	8	O	O	O	O	O	—
Port E	8	O	O	O	O	O	—
Port F	8	O	O	O	O	O	O
Port H	8	O	O	O	O	O	—
Port I	8	O	O	O	O	O	—

[Legend]

O: Register exists

—: No register exists

Notes: 1. The lower six bits are valid and the upper two bits are reserved. The write valu
always be the initial value.

2. The lower four bits are valid and the upper four bits are reserved. The write val
should always be the initial value.

Bit	7	6	5	4	3	2	1
Bit Name	Pn7DDR	Pn6DDR	Pn5DDR	Pn4DDR	Pn3DDR	Pn2DDR	Pn1DDR
Initial Value	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W

Note: The lower six bits are valid and the upper two bits are reserved for port 6 registers.
The lower four bits are valid and the upper four bits are reserved for port B registers.

Table 9.3 Startup Mode and Initial Value

	Startup Mode	
Port	**External Extended Mode**	**Single-Chip Mode**
Port A	H'80	H'00
Other ports	H'00	

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Note: The lower six bits are valid and the upper two bits are reserved for port 6 registers.
The lower four bits are valid and the upper four bits are reserved for port B registers.

9.1.3 Port Register (PORTn) (n = 1 to 3, 5, 6, A, B, D to F, H, and I)

PORT is an 8-bit read-only register that reflects the port pin state. A write to PORT is inv
When PORT is read, the DR bits that correspond to the respective DDR bits set to 1 are r
the status of each pin whose corresponding DDR bit is cleared to 0 is also read regardless
ICR value.

The initial value of PORT is undefined and is determined based on the port pin state.

Bit	7	6	5	4	3	2	1	
Bit Name	Pn7	Pn6	Pn5	Pn4	Pn3	Pn2	Pn1	
Initial Value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Ur
R/W	R	R	R	R	R	R	R	

Note: The lower six bits are valid and the upper two bits are reserved for port 6 registers.
The lower four bits are valid and the upper four bits are reserved for port B registers.

When PORT is read, the pin state is always read regardless of the ICR value. When the
is cleared to 0 at this time, the read pin state is not reflected in a corresponding on-chip
module.

If ICR is modified, an internal edge may occur depending on the pin state. Accordingly,
should be modified when the corresponding input pins are not used. For example, an IR
modify ICR while the corresponding interrupt is disabled, clear the IRQF flag in ISR of
interrupt controller to 0, and then enable the corresponding interrupt. If an edge occurs a
ICR setting, the edge should be cancelled.

The initial value of ICR is H'00.

Bit	7	6	5	4	3	2	1
Bit Name	Pn7ICR	Pn6ICR	Pn5ICR	Pn4ICR	Pn3ICR	Pn2ICR	Pn1ICR
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: The lower six bits are valid and the upper two bits are reserved for port 6 registers.
The lower four bits are valid and the upper four bits are reserved for port B registers.

Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 9.4 Input Pull-Up MOS State

Port	Pin State	Reset	Hardware Standby Mode	Software Standby Mode	Oth Op
Port D	Address output			OFF	
	Port output			OFF	
	Port input		OFF		ON/OFF
Port E	Address output			OFF	
	Port output			OFF	
	Port input		OFF		ON/OFF
Port F	Address output			OFF	
	Port output			OFF	
	Port input		OFF		ON/OFF
Port H	Data input/output			OFF	
	Port output			OFF	
	Port input		OFF		ON/OFF
Port I	Data input/output			OFF	
	Port output			OFF	
	Port input		OFF		ON/OFF

[Legend]

OFF: The input pull-up MOS is always off.

ON/OFF: If PCR is set to 1, the input pull-up MOS is on; if PCR is cleared to 0, the input MOS is off.

Bit Name	Pn7ODR	Pn6ODR	Pn5ODR	Pn4ODR	Pn3ODR	Pn2ODR	Pn1ODR	
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

9.2 Output Buffer Control

This section describes the output priority of each pin.

The name of each peripheral module pin is followed by "_OE". This (for example: TIO
indicates whether the output of the corresponding function is valid (1) or if another setti
specified (0). Table 9.5 lists each port output signal's valid setting. For details on the
corresponding output signals, see the register description of each peripheral module. If t
of each peripheral module pin is followed by A or B, the pin function can be modified b
function control register (PFCR). For details, see section 9.3, Port Function Controller.

For a pin whose initial value changes according to the activation mode, "Initial value E"
the initial value when the LSI is started up in external extended mode and "Initial value
indicates the initial value when the LSI is started in single-chip mode.

	P17 input (initial setting)	0

(2) P16/$\overline{\text{DACK1}}$-A/$\overline{\text{IRQ6}}$-A/TCLKC-B

The pin function is switched as shown below according to the combination of the DMAC setting and P16DDR bit setting.

		Setting	
		DMAC	**I/O Port**
Module Name	**Pin Function**	**$\overline{\text{DACK1A}}$_OE**	**P16DDR**
DMAC	$\overline{\text{DACK1}}$-A output	1	—
I/O port	P16 output	0	1
	P16 input (initial setting)	0	0

	P15 input (initial setting)	0	0

(4) P14/$\overline{\text{DREQ1}}$-A/$\overline{\text{IRQ4}}$-A/TCLKA-B

The pin function is switched as shown below according to the P14DDR bit setting.

		Setting
		I/O Port
Module Name	**Pin Function**	**P14DDR**
I/O port	P14 output	1
	P14 input (initial setting)	0

(5) P13/ADTRG0/IRQ3-A

The pin function is switched as shown below according to the P13DDR bit setting.

		Setting
		I/O Port
Module Name	**Pin Function**	**P13DDR**
I/O port	P13 output	1
	P13 input (initial setting)	0

I/O port	P12 output	0	0	1
	P12 input (initial setting)	0	0	0

(7) P11/RxD2/$\overline{\text{TEND0}}$-A/$\overline{\text{IRQ1}}$-A

The pin function is switched as shown below according to the combination of the DMAC setting and P11DDR bit setting.

		Setting	
		DMAC	**I/O Port**
Module Name	**Pin Function**	**$\overline{\text{TEND0A}}$_OE**	**P11DDR**
DMAC	$\overline{\text{TEND0}}$-A output	1	—
I/O port	P11 output	0	1
	P11 input (initial setting)	0	0

	P10 input (initial setting)	0	0

9.2.2 Port 2

(1) P27/PO7/TIOCA5/TIOCB5

The pin function is switched as shown below according to the combination of the TPU a register settings and P27DDR bit setting.

		Setting		
		TPU	PPG	I/O Por
Module Name	Pin Function	TIOCB5_OE	PO7_OE	P27DD
TPU	TIOCB5 output	1	—	—
PPG	PO7 output	0	1	—
I/O port	P27 output	0	0	1
	P27 input (initial setting)	0	0	0

SCI	TxD1 output	0	0	1	—
PPG	PO6 output	0	0	0	1
I/O port	P26 output	0	0	0	0
	P26 input (initial setting)	0	0	0	0

(3) P25/PO5/TIOCA4/TMCI1/RxD1

The pin function is switched as shown below according to the combination of the TPU an
register settings and P25DDR bit setting.

		Setting		
		TPU	**PPG**	**I/O Port**
Module Name	**Pin Function**	**TIOCA4_OE**	**PO5_OE**	**P25DDR**
TPU	TIOCA4 output	1	—	—
PPG	PO5 output	0	1	—
I/O port	P25 output	0	0	1
	P25 input (initial setting)	0	0	0

PPG	PO4 output	0	0	1	—
I/O port	P24 output	0	0	0	1
	P24 input (initial setting)	0	0	0	0

(5) P23/PO3/TIOCC3/TIOCD3/$\overline{\text{IRQ11}}$-A

The pin function is switched as shown below according to the combination of the TPU a
register settings and P23DDR bit setting.

		Setting		
		TPU	PPG	I/O Por
Module Name	**Pin Function**	**TIOCD3_OE**	**PO3_OE**	**P23DD**
TPU	TIOCD3 output	1	—	—
PPG	PO3 output	0	1	—
I/O port	P23 output	0	0	1
	P23 input (initial setting)	0	0	0

SCI	TxD0 output	0	0	1	—
PPG	PO2 output	0	0	0	1
I/O port	P22 output	0	0	0	0
	P22 input (initial setting)	0	0	0	0

(7) P21/PO1/TIOCA3/TMCI0/RxD0/$\overline{\text{IRQ9}}$-A

The pin function is switched as shown below according to the combination of the TPU an
register settings and P21DDR bit setting.

		Setting		
		TPU	**PPG**	**I/O Port**
Module Name	**Pin Function**	**TIOCA3_OE**	**PO1_OE**	**P21DDR**
TPU	TIOCA3 output	1	—	—
PPG	PO1 output	0	1	—
I/O port	P21 output	0	0	1
	P21 input (initial setting)	0	0	0

PPG	PO0 output	0	0	1	—
I/O port	P20 output	0	0	0	1
	P20 input (initial setting)	0	0	0	0

9.2.3 Port 3

(1) P37/PO15/TIOCA2/TIOCB2/TCLKD-A

The pin function is switched as shown below according to the combination of the TPU a
register settings and P37DDR bit setting.

		Setting		
		TPU	PPG	I/O Por
Module Name	Pin Function	TIOCB2_OE	PO15_OE	P37DD
TPU	TIOCB2 output	1	—	—
PPG	PO15 output	0	1	—
I/O port	P37 output	0	0	1
	P37 input (initial setting)	0	0	0

I/O port	P36 output	0	0	1
	P36 input (initial setting)	0	0	0

(3) P35/PO13/TIOCA1/TIOCB1/TCLKC-A/$\overline{\text{DACK1}}$-B

The pin function is switched as shown below according to the combination of the DMAC and PPG register settings and P35DDR bit setting.

		Setting			
		DMAC	**TPU**	**PPG**	**I/O**
Module Name	**Pin Function**	**$\overline{\text{DACK1B}}$_OE**	**TIOCB1_OE**	**PO13_OE**	**P35**
DMAC	$\overline{\text{DACK1}}$-B output	1	—	—	—
TPU	TIOCB1 output	0	1	—	—
PPG	PO13 output	0	0	1	—
I/O port	P35 output	0	0	0	1
	P35 input (initial setting)	0	0	0	0

PPG	PO12 output	0	0	1	—
I/O port	P34 output	0	0	0	1
	P34 input (initial setting)	0	0	0	0

(5) P33/PO11/TIOCC0/TIOCD0/TCLKB-A/$\overline{\text{DREQ1}}$-B

The pin function is switched as shown below according to the combination of the TPU a
register settings and P33DDR bit setting.

		Setting		
		TPU	PPG	I/O Por
Module Name	**Pin Function**	**TIOCD0_OE**	**PO11_OE**	**P33DD**
TPU	TIOCD0 output	1	—	—
PPG	PO11 output	0	1	—
I/O port	P33 output	0	0	1
	P33 input (initial setting)	0	0	0

PPG	PO10 output	0	0	1	—
I/O port	P32 output	0	0	0	1
	P32 input (initial setting)	0	0	0	0

(7) P31/PO9/TIOCA0/TIOCB0/$\overline{\text{TEND0}}$-B

The pin function is switched as shown below according to the combination of the DMAC and PPG register settings and P31DDR bit setting.

		Setting			
		DMAC	**TPU**	**PPG**	**I/O**
Module Name	**Pin Function**	**$\overline{\text{TEND0B}}$_OE**	**TIOCB0_OE**	**PO9_OE**	**P31**
DMAC	$\overline{\text{TEND0}}$-B output	1	—	—	—
TPU	TIOCB0 output	0	1	—	—
PPG	PO9 output	0	0	1	—
I/O port	P31 output	0	0	0	1
	P31 input (initial setting)	0	0	0	0

I/O port	P30 output	0	0	1
	P30 input (initial setting)	0	0	0

9.2.4 Port 5

(1) P57/AN7/DA1/$\overline{\text{IRQ7}}$-B

Module Name	Pin Function
D/A converter	DA1 output

(2) P56/AN6/DA0/$\overline{\text{IRQ6}}$-B

Module Name	Pin Function
D/A converter	DA0 output

DMAC	DACK3 output	1	—	—
TMR	TMO3 output	0	1	—
I/O port	P65 output	0	0	1
	P65 input (initial setting)	0	0	0

(2) P64/TMCI3/$\overline{\text{TEND3}}$

The pin function is switched as shown below according to the combination of the DMAC setting and P64DDR bit setting.

		Setting	
		DMAC	**I/O Port**
Module Name	**Pin Function**	**$\overline{\text{TEND3}}$_OE**	**P64DDR**
DMAC	$\overline{\text{TEND3}}$ output	1	—
I/O port	P64 output	0	1
	P64 input (initial setting)	0	0

(4) P62/TMO2/SCK4/$\overline{\text{DACK2}}$/$\overline{\text{IRQ10}}$-B

The pin function is switched as shown below according to the combination of the DMA
and SCI register settings and P62DDR bit setting.

		Setting			
		DMAC	TMR	SCI	I/O
Module Name	Pin Function	$\overline{\text{DACK2}}$_OE	TMO2_OE	SCK4_OE	P6
DMAC	$\overline{\text{DACK2}}$ output	1	—	—	—
TMR	TMO2 output	0	1	—	—
SCI	SCK4 output	0	0	1	—
I/O port	P62 output	0	0	0	1
	P62 input (initial setting)	0	0	0	0

(5) P61/TMCI2/RxD4/$\overline{\text{TEND2}}$/$\overline{\text{IRQ9}}$-B

The pin function is switched as shown below according to the combination of the DMA
setting and P61DDR bit setting.

		Setting	
		DMAC	I/O Port
Module Name	Pin Function	$\overline{\text{TEND2}}$_OE	P61DDR
DMAC	$\overline{\text{TEND2}}$ output	1	—
I/O port	P61 output	0	1
	P61 input (initial setting)	0	0

	P60 input (initial setting)	0	0

9.2.6 Port A

(1) PA7/Bϕ

The pin function is switched as shown below according to the PA7DDR bit setting.

		Setting	
		I/O Port	
Module Name	**Pin Function**	**PA7DDR**	
I/O port	Bϕ output* (initial setting E)	1	
	PA7 input (initial setting S)	0	

[Legend]

Initial setting E: Initial setting in external extended mode

Initial setting S: Initial setting in single-chip mode

Note: * The type of ϕ to be output switches according to the POSEL1 bit in SCKCR. Fo
see section 19.1.1, System Clock Control Register (SCKCR).

	$\overline{AS}$ output* (initial setting E)	0	0	1	—
I/O port	PA6 output	0	0	0	1
	PA6 input (initial setting S)	0	0	0	0

[Legend]
Initial setting E: Initial setting in external extended mode
Initial setting S: Initial setting in single-chip mode
Note: * Valid in external extended mode (EXPE = 1)

(3) PA5/$\overline{RD}$

The pin function is switched as shown below according to the combination of operating
EXPE bit, and the PA5DDR bit settings.

		Setting	
		MCU Operating Mode	**I/O Port**
Module Name	**Pin Function**	**EXPE**	**PA5DDR**
Bus controller	$\overline{RD}$ output* (Initial setting E)	1	—
I/O port	PA5 output	0	1
	PA5 input (initial setting S)	0	0

[Legend]
Initial setting E: Initial setting in external extended mode
Initial setting S: Initial setting in single-chip mode
Note: * Valid in external extended mode (EXPE = 1)

	LHWR output (initial setting E)		1	
I/O port	PA4 output	0	0	1
	PA4 input (initial setting S)	0	0	0

[Legend]

Initial setting E: Initial setting in external extended mode

Initial setting S: Initial setting in single-chip mode

Notes: 1. Valid in external extended mode (EXPE = 1)

2. When the byte control SRAM space is accessed while the byte control SRAM s
specified or while LHWROE =1, this pin functions as the $\overline{\text{LUB}}$ output; otherwise
$\overline{\text{LHWR}}$ output.

I/O port	PA3 output	0	0	1
	PA3 input (initial setting S)	0	0	0

[Legend]

Initial setting E: Initial setting in external extended mode

Initial setting S: Initial setting in single-chip mode

Notes: 1. Valid in external extended mode (EXPE = 1)

2. If the byte control SRAM space is accessed, this pin functions as the $\overline{\text{LLB}}$ out otherwise, the $\overline{\text{LLWR}}$.

(6) PA2/BREQ/WAIT

The pin function is switched as shown below according to the combination of the bus co register setting and the PA2DDR bit setting.

		Setting		
		Bus Controller		I/O Port
Module Name	**Pin Function**	**BCR_BRLE**	**BCR_WAITE**	**PA2DDR**
Bus controller	$\overline{\text{BREQ}}$ input	1	—	—
	$\overline{\text{WAIT}}$ input	0	1	—
I/O port	PA2 output	0	0	1
	PA2 input (initial setting)	0	0	0

Bus controller	$\overline{\text{BACK}}$ output *	1	—	—	—
	RD/$\overline{\text{WR}}$ output *	0	1	—	—
		0	0	1	—
I/O port	PA1 output	0	0	0	1
	PA1 input (initial setting)	0	0	0	0

Note: * Valid in external extended mode (EXPE = 1)

(8) PA0/$\overline{\text{BREQO}}$/$\overline{\text{BS}}$-A

The pin function is switched as shown below according to the combination of operating m
EXPE bit, bus controller register, port function control register (PFCR), and the PA0DDR
settings.

		Setting		
		I/O Port	**Bus Controller**	**I/O Port**
Module Name	**Pin Function**	**$\overline{\text{BSA}}$_OE**	**$\overline{\text{BREQ}}$_OE**	**PA0DDR**
Bus controller	$\overline{\text{BS}}$-A output*	1	—	—
	$\overline{\text{BREQO}}$ output*	0	1	—
I/O port	PA0 output	0	0	1
	PA0 input (initial setting)	0	0	0

Note: * Valid in external extended mode (EXPE = 1)

I/O port	PB3 output	0	0	1
	PB3 input (initial setting)	0	0	0

Note: * Valid in external extended mode (EXPE = 1)

(1) PB2/$\overline{\text{CS2}}$-A/$\overline{\text{CS6}}$-A

The pin function is switched as shown below according to the combination of operating
EXPE bit, port function control register (PFCR), and the PB2DDR bit settings.

		Setting		
		I/O Port		
Module Name	**Pin Function**	**$\overline{\text{CS2A}}$_OE**	**$\overline{\text{CS6A}}$_OE**	**PB2DD**
Bus controller	$\overline{\text{CS2}}$-A output*	1	—	—
	$\overline{\text{CS6}}$-A output*	—	1	—
I/O port	PB2 output	0	0	1
	PB2 input (initial setting)	0	0	0

Note: * Valid in external extended mode (EXPE = 1)

	CS5-A output*	—	—	1	—	—
	$\overline{CS6}$-B output*	—	—	—	1	—
	$\overline{CS7}$-B output*	—	—	—	—	1
I/O port	PB1 output	0	0	0	0	0
	PB1 input (initial setting)	0	0	0	0	0

Note: * Valid in external extended mode (EXPE = 1)

(3) PB0/$\overline{CS0}$/$\overline{CS4}$/$\overline{CS5}$-B

The pin function is switched as shown below according to the combination of operating m
EXPE bit, port function control register (PFCR), and the PB0DDR bit settings.

		Setting			
		I/O Port			
Module Name	**Pin Function**	**$\overline{CS0}$_OE**	**$\overline{CS4}$_OE**	**$\overline{CS5B}$_OE**	**PB0**
Bus controller	$\overline{CS0}$ output* (initial setting E)	1	—	—	—
	$\overline{CS4}$ output*	—	1	—	—
	$\overline{CS5}$-B output*	—	—	1	—
I/O port	PB0 output	0	0	0	1
	PB0 input (initial setting S)	0	0	0	0

[Legend]
Initial setting E: Initial setting in on-chip ROM disabled external extended mode
Initial setting S: Initial setting in other modes
Note: * Valid in external extended mode (EXPE = 1)

Bus controller	Address output	On-chip ROM disabled extended mode	—
		On-chip ROM enabled extended mode	1
I/O port	PDn output	Single-chip mode*	1
	PDn input (initial setting)	Modes other than on-chip ROM disabled extended mode	0

[Legend]

n = 0 to 7

Note: * Address output is enabled by setting PDnDDR = 1 in external extended mode (EXPE = 1)

Bus controller	Address output	On-chip ROM disabled extended mode	—
		On-chip ROM enabled extended mode	1
I/O port	PEn output	Single-chip mode*	1
	PEn input (initial setting)	Modes other than on-chip ROM disabled extended mode	0

[Legend]

n = 0 to 7

Note: * Address output is enabled by setting PDnDDR = 1 in external extended mode (EXPE = 1)

9.2.10 Port F

(1) PF7/A23

The pin function is switched as shown below according to the combination of operating m
EXPE bit, port function control register (PFCR), and the PF7DDR bit settings.

			Setting	
			I/O Port	
MCU Operating Mode	**Module Name**	**Pin Function**	**A23_OE**	**PF7**
Modes other than on-chip ROM disabled extended mode	Bus controller	A23 output*	1	—
	I/O port	PF7 output	0	1
		PF7 input (initial setting)	0	0

Note: * Valid in external extended mode (EXPE = 1)

mode	PF6 input (initial setting)	0	0

Note: * Valid in external extended mode (EXPE = 1)

(3) PF5/A21

The pin function is switched as shown below according to the combination of operating
EXPE bit, port function control register (PFCR), and the PF5DDR bit settings.

MCU Operating Mode	Module Name	Pin Function	Settin	
			I/O Po	
			A21_OE	PF
Modes other than on-chip ROM disabled extended mode	Bus controller	A21 output*	1	—
	I/O port	PF5 output	0	1
		PF5 input (initial setting)	0	0

Note: * Valid in external extended mode (EXPE = 1)

Modes other than on-chip ROM disabled extended mode	Bus controller	A20 output*	1	—
	I/O port	PF4 output	0	1
		PF4 input (initial setting)	0	0

Note: * Valid in external extended mode (EXPE = 1)

(5) PF3/A19

The pin function is switched as shown below according to the combination of operating m
EXPE bit, port function control register (PFCR), and the PF3DDR bit settings.

			Setting	
			I/O Port	
MCU Operating Mode	**Module Name**	**Pin Function**	**A19_OE**	**PF3**
On-chip ROM disabled extended mode	Bus controller	A19 output	—	—
Modes other than on-chip ROM disabled extended mode	Bus controller	A19 output*	1	—
	I/O port	PF3 output	0	1
		PF3 input (initial setting)	0	0

Note: * Valid in external extended mode (EXPE = 1)

Modes other than on-chip ROM disabled extended mode	Bus controller	A18 output*	1	—
	I/O port	PF2 output	0	1
		PF2 input (initial setting)	0	0

Note: * Valid in external extended mode (EXPE = 1)

(7) PF1/A17

The pin function is switched as shown below according to the combination of operating
EXPE bit, port function control register (PFCR), and the PF1DDR bit settings.

MCU Operating Mode	Module Name	Pin Function	Settin	
			I/O Po	
			A17_OE	PF
On-chip ROM disabled extended mode	Bus controller	A17 output	—	—
Modes other than on-chip ROM disabled extended mode	Bus controller	A17 output*	1	—
	I/O port	PF1 output	0	1
		PF1 input (initial setting)	0	0

Note: * Valid in external extended mode (EXPE = 1)

Modes other than on-chip ROM disabled extended mode	Bus controller	A16 output*	1	—
	I/O port	PF0 output	0	1
		PF0 input (initial setting)	0	0

Note: * Valid in external extended mode (EXPE = 1)

9.2.11 Port H

(1) PH7/D7, PH6/D6, PH5/D5, PH4/D4, PH3/D3, PH2/D2, PH1/D1, PH0/D0

The pin function is switched as shown below according to the combination of operating m
EXPE bit, and the PHnDDR bit settings.

		Setting	
		MCU Operating Mode	**I/O Port**
Module Name	**Pin Function**	**EXPE**	**PHnDDR**
Bus controller	Data I/O* (initial setting E)	1	—
I/O port	PHn output	0	1
	PHn input (initial setting S)	0	0

[Legend]
Initial setting E: Initial setting in external extended mode
Initial setting S: Initial setting in single-chip mode
n = 0 to 7
Note: * Valid in external extended mode (EXPE = 1)

Bus controller	Data I/O* (initial setting E)	1	—
I/O port	PIn output	0	1
	PIn input (initial setting S)	0	0

[Legend]

Initial setting E: Initial setting in external extended mode

Initial setting S: Initial setting in single-chip mode

n = 0 to 7

Note: * Valid in external extended mode (EXPE = 1)

					When SCMR_2.SMIF = 0: SCR_2.TE = 1 or SCR_2.RE = 1 w SMR_2.C/A = 0, SCR_2.CKE [1, 0 while SMR_2.C/A = 1, SCR_2.CKE
	1	$\overline{\text{TEND0A}}$_OE	$\overline{\text{TEND0}}$	PFCR7.DMAS0[A,B] = 00	DMDR.TENDE = 1
	0	TxD2_OE	TxD2		SCR.TE = 1
P2	7	TIOCB5_OE	TIOCB5		TPU.TIOR5.IOB3 = 0, TPU.TIOR5 = 01/10/11
		PO7_OE	PO7		NDERL.NDER7 = 1
	6	TIOCA5_OE	TIOCA5		TPU.TIOR5.IOA3 = 0, TPU.TIOR5 = 01/10/11
		TMO1_OE	TMO1		TCSR.OS3,2 = 01/10/11 or TCSR. = 01/10/11
		TxD1_OE	TxD1		SCR.TE = 1
		PO6_OE	PO6		NDERL.NDER6 = 1
	5	TIOCA4_OE	TIOCA4		TPU.TIOR4.IOA3 = 0, TPU.TIOR4 = 01/10/11
		PO5_OE	PO5		NDERL.NDER5 = 1
	4	TIOCB4_OE	TIOCB4		TPU.TIOR4.IOB3 = 0, TPU.TIOR4 = 01/10/11
		SCK1_OE	SCK1		When SCMR_1.SMIF = 1: SCR_1.TE = 1 or SCR_1.RE = 1 w SMR_1.GM = 0, SCR_1.CKE [1, 0 while SMR_1.GM = 1 When SCMR_1.SMIF = 0: SCR_1.TE = 1 or SCR_1.RE = 1 w SMR_1.C/A = 0, SCR_1.CKE [1, 0 while SMR_1.C/A = 1, SCR_1.CKE
		PO4_OE	PO4		NDERL.NDER4 = 1

	1	TIOCA3_OE	TIOCA3		TPU.TIORH3.IOA3 = 0, TPU.TIORH3.IOA[1,0] = 01/10/1
		PO1_OE	PO1		NDERL.NDER1 = 1
	0	TIOCB3_OE	TIOCB3		TPU.TIORH3.IOB3 = 0, TPU.TIORH3.IOB[1,0] = 01/10/1
		SCK0_OE	SCK0		When SCMR_0.SMIF = 1: SCR_0.TE = 1 or SCR_0.RE = 1 SMR_0.GM = 0, SCR_0.CKE [1, while SMR_0.GM = 1 When SCMR_0.SMIF = 0: SCR_0.TE = 1 or SCR_0.RE = 1 SMR_0.C/A = 0, SCR_0.CKE [1, while SMR_0.C/A = 1, SCR_0.CK
		PO0_OE	PO0		NDERL.NDER0 = 1
P3	7	TIOCB2_OE	TIOCB2		TPU.TIOR2.IOB3 = 0, TPU.TIOR = 01/10/11
		PO15_OE	PO15		NDERH.NDER15 = 1
	6	TIOCA2_OE	TIOCA2		TPU.TIOR2.IOA3 = 0, TPU.TIOR = 01/10/11
		PO14_OE	PO14		NDERH.NDER14 = 1
	5	$\overline{\text{DACK1B}}$_OE	$\overline{\text{DACK1}}$	PFCR7.DMAS1[A,B] = 01	DACR.AMS = 1, DMDR.DACKE
		TIOCB1_OE	TIOCB1		TPU.TIOR1.IOB3 = 0, TPU.TIOR = 01/10/11
		PO13_OE	PO13		NDERH.NDER13 = 1
	4	$\overline{\text{TEND1B}}$_OE	$\overline{\text{TEND1}}$	PFCR7.DMAS1[A,B] = 01	DMDR.TENDE = 1
		TIOCA1_OE	TIOCA1		TPU.TIOR1.IOA3 = 0, TPU.TIOR = 01/10/11
		PO12_OE	PO12		NDERH.NDER12 = 1

		TIOCB0_OE	TIOCB0		TPU.TIORH0.IOB[1,0] = 01/10/11
		PO9_OE	PO9		NDERH.NDER9 = 1
	0	TIOCA0_OE	TIOCA0		TPU.TIORH0.IOA3 = 0, TPU.TIORH0.IOA[1,0] = 01/10/11
		PO8_OE	PO8		NDERH.NDER8 = 1
P6	5	$\overline{\text{DACK3}}$_OE	$\overline{\text{DACK3}}$	PFCR7.DMAS3[A,B] = 01	DACR.AMS = 1, DMDR.DACKE =
		TMO3_OE	TMO3		TCSR.OS[3,2] = 01/10/11 or TCSR = 01/10/11
	4	$\overline{\text{TEND3}}$_OE	$\overline{\text{TEND3}}$	PFCR7.DMAS3[A,B] = 01	DMDR.TENDE = 1
	2	$\overline{\text{DACK2}}$_OE	$\overline{\text{DACK2}}$	PFCR7.DMAS2[A,B] = 01	DACR.AMS = 1, DMDR.DACKE =
		TMO2_OE	TMO2		TCSR.OS[3,2] = 01/10/11 or TCSR = 01/10/11
		SCK4_OE	SCK4		When SCMR_4.SMIF = 1: SCR_4.TE = 1 or SCR_4.RE = 1 w SMR_4.GM = 0, SCR_4.CKE [1, 0 while SMR_4.GM = 1 When SCMR_4.SMIF = 0: SCR_4.TE = 1 or SCR_4.RE = 1 w SMR_4.C/A = 0, SCR_4.CKE [1, 0 while SMR_4.C/A = 1, SCR_4.CKE
	1	$\overline{\text{TEND2}}$_OE	$\overline{\text{TEND2}}$	PFCR7.DMAS2[A,B] = 01	DMDR.TENDE = 1
	0	TxD4_OE	TxD4		SCR.TE = 1
PA	7	Bϕ_OE	Bϕ		PADDR.PA7DDR = 1, SCKCR.PC
	6	$\overline{\text{AH}}$_OE	$\overline{\text{AH}}$		SYSCR.EXPE = 1, MPXCR.MPXEn (n = 7 to 3) = 1
		$\overline{\text{BSB}}$_OE	$\overline{\text{BS}}$	PFCR2.BSS = 1	SYSCR.EXPE = 1, PFCR2.BSE =
		$\overline{\text{AS}}$_OE	$\overline{\text{AS}}$		SYSCR.EXPE = 1, PFCR2.ASOE
	5	$\overline{\text{RD}}$_OE	$\overline{\text{RD}}$		SYSCR.EXPE = 1

					SRAMCR.BCSELn = 1
	0	$\overline{\text{BSA}}$_OE	$\overline{\text{BS}}$	PFCR2.BSS = 0	SYSCR.EXPE = 1, PFCR2.BSE =
		$\overline{\text{BREQO}}$_OE	$\overline{\text{BREQO}}$		SYSCR.EXPE = 1, BCR1.BRLE = BCR1.BREQOE = 1
PB	3	$\overline{\text{CS3}}$_OE	$\overline{\text{CS3}}$		SYSCR.EXPE = 1, PFCR0.CS3E
		$\overline{\text{CS7A}}$_OE	$\overline{\text{CS7}}$	PFCR1.CS7S[A,B] = 00	SYSCR.EXPE = 1, PFCR0.CS7E
	2	$\overline{\text{CS2A}}$_OE	$\overline{\text{CS2}}$	PFCR2.CS2S = 0	SYSCR.EXPE = 1, PFCR0.CS2E
		$\overline{\text{CS6A}}$_OE	$\overline{\text{CS6}}$	PFCR1.CS6S[A,B] = 00	SYSCR.EXPE = 1, PFCR0.CS6E
	1	$\overline{\text{CS1}}$_OE	$\overline{\text{CS1}}$		SYSCR.EXPE = 1, PFCR0.CS1E
		$\overline{\text{CS2B}}$_OE	$\overline{\text{CS2}}$	PFCR2.CS2S = 1	SYSCR.EXPE = 1, PFCR0.CS2E
		$\overline{\text{CS5A}}$_OE	$\overline{\text{CS5}}$	PFCR1.CS5S[A,B] = 00	SYSCR.EXPE = 1, PFCR0.CS5E
		$\overline{\text{CS6B}}$_OE	$\overline{\text{CS6}}$	PFCR1.CS6S[A,B] = 01	SYSCR.EXPE = 1, PFCR0.CS6E
		$\overline{\text{CS7B}}$_OE	$\overline{\text{CS7}}$	PFCR1.CS7S[A,B] = 01	SYSCR.EXPE = 1, PFCR0.CS7E
	0	$\overline{\text{CS0}}$_OE	$\overline{\text{CS0}}$		SYSCR.EXPE = 1, PFCR0.CS0E
		$\overline{\text{CS4}}$_OE	$\overline{\text{CS4}}$		SYSCR.EXPE = 1, PFCR0.CS4E
		$\overline{\text{CS5B}}$_OE	$\overline{\text{CS5}}$	PFCR1.CS5S[A,B] = 01	SYSCR.EXPE = 1, PFCR0.CS5E
PD	7	A7_OE	A7		SYSCR.EXPE = 1, PDDDR.PD7
	6	A6_OE	A6		SYSCR.EXPE = 1, PDDDR.PD6
	5	A5_OE	A5		SYSCR.EXPE = 1, PDDDR.PD5
	4	A4_OE	A4		SYSCR.EXPE = 1, PDDDR.PD4
	3	A3_OE	A3		SYSCR.EXPE = 1, PDDDR.PD3
	2	A2_OE	A2		SYSCR.EXPE = 1, PDDDR.PD2
	1	A1_OE	A1		SYSCR.EXPE = 1, PDDDR.PD1
	0	A0_OE	A0		SYSCR.EXPE = 1, PDDDR.PD0

	0	A8_OE	A8	SYSCR.EXPE = 1, PDDDR.PE0D
PF	7	A23_OE	A23	SYSCR.EXPE = 1, PFCR4.A23E =
	6	A22_OE	A22	SYSCR.EXPE = 1, PFCR4.A22E =
	5	A21_OE	A21	SYSCR.EXPE = 1, PFCR4.A21E =
	4	A20_OE	A20	SYSCR.EXPE = 1, PFCR4.A20E =
	3	A19_OE	A19	SYSCR.EXPE = 1, PFCR4.A19E =
	2	A18_OE	A18	SYSCR.EXPE = 1, PFCR4.A18E =
	1	A17_OE	A17	SYSCR.EXPE = 1, PFCR4.A17E =
	0	A16_OE	A16	SYSCR.EXPE = 1, PFCR4.A16E =
PH	7	D7_E	D7	SYSCR.EXPE = 1
	6	D6_E	D6	SYSCR.EXPE = 1
	5	D5_E	D5	SYSCR.EXPE = 1
	4	D4_E	D4	SYSCR.EXPE = 1
	3	D3_E	D3	SYSCR.EXPE = 1
	2	D2_E	D2	SYSCR.EXPE = 1
	1	D1_E	D1	SYSCR.EXPE = 1
	0	D0_E	D0	SYSCR.EXPE = 1
PI	7	D15_E	D15	SYSCR.EXPE = 1, ABWCR.ABW[
	6	D14_E	D14	SYSCR.EXPE = 1, ABWCR.ABW[
	5	D13_E	D13	SYSCR.EXPE = 1, ABWCR.ABW[
	4	D12_E	D12	SYSCR.EXPE = 1, ABWCR.ABW[
	3	D11_E	D11	SYSCR.EXPE = 1, ABWCR.ABW[
	2	D10_E	D10	SYSCR.EXPE = 1, ABWCR.ABW[
	1	D9_E	D9	SYSCR.EXPE = 1, ABWCR.ABW[
	0	D8_E	D8	SYSCR.EXPE = 1, ABWCR.ABW[

- Port function control register 6 (PFCR6)
- Port function control register 7 (PFCR7)
- Port function control register 9 (PFCR9)
- Port function control register B (PFCRB)
- Port function control register C (PFCRC)

9.3.1 Port Function Control Register 0 (PFCR0)

PFCR0 enables/disables the $\overline{CS}$ output.

Bit	7	6	5	4	3	2	1	
Bit Name	CS7E	CS6E	CS5E	CS4E	CS3E	CS2E	CS1E	
Initial Value	0	0	0	0	0	0	0	U
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Note: * 1 in external extended mode; 0 in other modes.

Bit	Bit Name	Initial Value	R/W	Description
7	CS7E	0	R/W	CS7 to CS0 Enable
6	CS6E	0	R/W	These bits enable/disable the corresponding $\overline{CS}$ output.
5	CS5E	0	R/W	0: Pin functions as I/O port
4	CS4E	0	R/W	1: Pin functions as $\overline{CSn}$ output pin
3	CS3E	0	R/W	(n = 7 to 0)
2	CS2E	0	R/W	
1	CS1E	0	R/W	
0	CS0E	Undefined*	R/W	

Note: * 1 in external extended mode; 0 in other modes.

Bit	Bit Name	Value	R/W	Description
7	CS7SA*	0	R/W	$\overline{\text{CS7}}$ Output Pin Select
6	CS7SB*	0	R/W	Selects the output pin for $\overline{\text{CS7}}$ when $\overline{\text{CS7}}$ output enabled (CS7E = 1) 00: Specifies pin PB3 as $\overline{\text{CS7}}$-A output 01: Specifies pin PB1 as $\overline{\text{CS7}}$-B output 10: Setting prohibited 11: Setting prohibited
5	CS6SA*	0	R/W	$\overline{\text{CS6}}$ Output Pin Select
4	CS6SB*	0	R/W	Selects the output pin for $\overline{\text{CS6}}$ when $\overline{\text{CS6}}$ output enabled (CS6E = 1) 00: Specifies pin PB2 as $\overline{\text{CS6}}$-A output 01: Specifies pin PB1 as $\overline{\text{CS6}}$-B output 10: Setting prohibited 11: Setting prohibited
3	CS5SA*	0	R/W	$\overline{\text{CS5}}$ Output Pin Select
2	CS5SB*	0	R/W	Selects the output pin for $\overline{\text{CS5}}$ when $\overline{\text{CS5}}$ output enabled (CS5E = 1) 00: Specifies pin PB1 as $\overline{\text{CS5}}$-A output 01: Specifies pin PB0 as $\overline{\text{CS5}}$-B output 10: Setting prohibited 11: Setting prohibited

PFCR1 selects the $\overline{CS}$ output pin, enables/disables bus control I/O, and selects the bus c
pins.

Bit	7	6	5	4	3	2	1
Bit Name	—	CS2S	BSS	BSE	—	RDWRE	ASOE
Initial Value	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R/W	Reserved This bit is always read as 0. The write value sh always be 0.
6	CS2S*[1]	0	R/W	$\overline{CS2}$ Output Pin Select Selects the output pin for $\overline{CS2}$ when $\overline{CS2}$ outpu enabled (CS2E = 1) 0: Specifies pin PB2 as $\overline{CS2}$-A output pin 1: Specifies pin PB1 as $\overline{CS2}$-B output pin
5	BSS	0	R/W	$\overline{BS}$ Output Pin Select Selects the $\overline{BS}$ output pin 0: Specifies pin PA0 as $\overline{BS}$-A output pin 1: Specifies pin PA6 as $\overline{BS}$-B output pin

				Enables/disables the RD/$\overline{\text{WR}}$ output 0: Disables the RD/$\overline{\text{WR}}$ output 1: Enables the RD/$\overline{\text{WR}}$ output
1	ASOE	1	R/W	$\overline{\text{AS}}$ Output Enable Enables/disables the $\overline{\text{AS}}$ output 0: Specifies pin PA6 as I/O port 1: Specifies pin PA6 as $\overline{\text{AS}}$ output pin
0	—	0	R/W	Reserved This bit is always read as 0. The write value sho always be 0.

Notes: 1. If multiple $\overline{\text{CS}}$ outputs are specified to a single pin according to the $\overline{\text{CS2}}$ output select bit, multiple $\overline{\text{CS}}$ signals are output from the pin. For details, see section Chip Select Signals.

2. If an area is specified as a byte control SDRAM space, the pin functions as RD output regardless of the RDWRE bit value.

Bit	Bit Name	Initial Value	R/W	Description
7	A23E	0	R/W	Address A23 Enable Enables/disables the address output (A23) 0: Disables the A23 output 1: Enables the A23 output
6	A22E	0	R/W	Address A22 Enable Enables/disables the address output (A22) 0: Disables the A22 output 1: Enables the A22 output
5	A21E	0	R/W	Address A21 Enable Enables/disables the address output (A21) 0: Disables the A21 output 1: Enables the A21 output
4	A20E	1/0*	R/W	Address A20 Enable Enables/disables the address output (A20) 0: Disables the A20 output 1: Enables the A20 output
3	A19E	1/0*	R/W	Address A19 Enable Enables/disables the address output (A19) 0: Disables the A19 output 1: Enables the A19 output

0	A16E	1/0*	R/W	Address A16 Enable Enables/disables the address output (A16) 0: Disables the A16 output 1: Enables the A16 output

9.3.5 Port Function Control Register 6 (PFCR6)

PFCR6 selects the TPU clock input pin.

Bit	7	6	5	4	3	2	1
Bit Name	—	LHWROE	—	—	TCLKS	—	—
Initial Value	1	1	1	0	0	0	0
R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	—	1	R/W	Reserved This bit is always read as 1. The write value sho always be 1.
6	LHWROE	1	R/W	$\overline{\text{LHWR}}$ Output Enable Enables/disables $\overline{\text{LHWR}}$ output (valid in external extended mode). 0: Specifies pin PA4 as I/O port 1: Specifies pin PA4 as $\overline{\text{LHWR}}$ output pin

				clock inputs 1: Specifies pins P14 to P17 as external clock i
2 to 0	—	All 0	R/W	Reserved These bits are always read as 0. The write valu always be 0.

9.3.6 Port Function Control Register 7 (PFCR7)

PFCR7 selects the DMAC I/O pins ($\overline{\text{DREQ}}$, $\overline{\text{DACK}}$, and $\overline{\text{TEND}}$).

Bit	7	6	5	4	3	2	1	
Bit Name	DMAS3A	DMAS3B	DMAS2A	DMAS2B	DMAS1A	DMAS1B	DMAS0A	
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Bit	Bit Name	Initial Value	R/W	Description
7	DMAS3A	0	R/W	DMAC control pin select
6	DMAS3B	0	R/W	Selects the I/O port to control DMAC_3. 00: Setting prohibited 01: Specifies pins P63 to P65 as DMAC control 10: Setting prohibited 11: Setting prohibited

				00: Specifies pins P14 to P16 as DMAC control p
				01: Specifies pins P33 to P35 as DMAC control p
				10: Setting prohibited
				11: Setting prohibited
1	DMAS0A	0	R/W	DMAC control pin select
0	DMAS0B	0	R/W	Selects the I/O port to control DMAC_0.
				00: Specifies pins P10 to P12 as DMAC control p
				01: Specifies pins P30 to P32 as DMAC control p
				10: Setting prohibited
				11: Setting prohibited

Bit	Bit Name	Value	R/W	Description
7	TPUMS5	0	R/W	TPU I/O Pin Multiplex Function Select Selects TIOCA5 function 0: Specifies pin P26 as output compare output capture 1: Specifies P27 as input capture input and P26 compare
6	TPUMS4	0	R/W	TPU I/O Pin Multiplex Function Select Selects TIOCA4 function 0: Specifies P25 as output compare output and capture 1: Specifies P24 as input capture input and P25 compare
5	TPUMS3A	0	R/W	TPU I/O Pin Multiplex Function Select Selects TIOCA3 function 0: Specifies P21 as output compare output and capture 1: Specifies P20 as input capture input and P21 compare
4	TPUMS3B	0	R/W	TPU I/O Pin Multiplex Function Select Selects TIOCC3 function 0: Specifies P22 as output compare output and capture 1: Specifies P23 as input capture input and P22 compare

				0: Specifies P34 as output compare output and i capture 1: Specifies P35 as input capture input and P34 compare
1	TPUMS0A	0	R/W	TPU I/O Pin Multiplex Function Select Selects TIOCA0 function 0: Specifies P30 as output compare output and i capture 1: Specifies P31 as input capture input and P30 compare
0	TPUMS0B	0	R/W	TPU I/O Pin Multiplex Function Select Selects TIOCC0 function 0: Specifies P32 as output compare output and i capture 1: Specifies P33 as input capture input and P32 compare

Bit	Bit Name	Value	R/W	Description
7 to 4	—	All 0	R/W	Reserved These bits are always read as 0. The write valu always be 0.
3	ITS11	0	R/W	$\overline{\text{IRQ11}}$ Pin Select Selects an input pin for $\overline{\text{IRQ11}}$. 0: Selects pin P23 as $\overline{\text{IRQ11}}$-A input 1: Selects pin P63 as $\overline{\text{IRQ11}}$-B input
2	ITS10	0	R/W	$\overline{\text{IRQ10}}$ Pin Select Selects an input pin for $\overline{\text{IRQ10}}$. 0: Selects pin P22 as $\overline{\text{IRQ10}}$-A input 1: Selects pin P62 as $\overline{\text{IRQ10}}$-B input
1	ITS9	0	R/W	$\overline{\text{IRQ9}}$ Pin Select Selects an input pin for $\overline{\text{IRQ9}}$. 0: Selects pin P21 as $\overline{\text{IRQ9}}$-A input 1: Selects pin P61 as $\overline{\text{IRQ9}}$-B input
0	ITS8	0	R/W	$\overline{\text{IRQ8}}$ Pin Select Selects an input pin for $\overline{\text{IRQ8}}$. 0: Selects pin P20 as $\overline{\text{IRQ8}}$-A input 1: Selects pin P60 as $\overline{\text{IRQ8}}$-B input

Bit	Bit Name	Value	R/W	Description
7	ITS7	0	R/W	$\overline{\text{IRQ7}}$ Pin Select Selects an input pin for $\overline{\text{IRQ7}}$. 0: Selects pin P17 as $\overline{\text{IRQ7}}$-A input 1: Selects pin P57 as $\overline{\text{IRQ7}}$-B output
6	ITS6	0	R/W	$\overline{\text{IRQ6}}$ Pin Select Selects an input pin for $\overline{\text{IRQ6}}$. 0: Selects pin P16 as $\overline{\text{IRQ6}}$-A input 1: Selects pin P56 as $\overline{\text{IRQ6}}$-B output
5	ITS5	0	R/W	$\overline{\text{IRQ5}}$ Pin Select Selects an input pin for $\overline{\text{IRQ5}}$. 0: Selects pin P15 as $\overline{\text{IRQ5}}$-A input 1: Selects pin P55 as $\overline{\text{IRQ5}}$-B output
4	ITS4	0	R/W	$\overline{\text{IRQ4}}$ Pin Select Selects an input pin for $\overline{\text{IRQ4}}$. 0: Selects pin P14 as $\overline{\text{IRQ4}}$-A input 1: Selects pin P54 as $\overline{\text{IRQ4}}$-B output
3	ITS3	0	R/W	$\overline{\text{IRQ3}}$ Pin Select Selects an input pin for $\overline{\text{IRQ3}}$. 0: Selects pin P13 as $\overline{\text{IRQ3}}$-A input 1: Selects pin P53 as $\overline{\text{IRQ3}}$-B output

0	ITS0	0	R/W	$\overline{\text{IRQ0}}$ Pin Select Selects an input pin for $\overline{\text{IRQ0}}$. 0: Selects pin P10 as $\overline{\text{IRQ0}}$-A input 1: Selects pin P50 as $\overline{\text{IRQ0}}$-B output

- When a pin is used as an output, data to be output from the pin will be latched as the p
  if the input function corresponding to the pin is enabled. To use the pin as an output, d
  the input function for the pin by setting ICR.

9.4.2 Notes on Port Function Control Register (PFCR) Settings

- Port function controller controls the I/O port.
  Before enabling a port function, select the input/output destination.
- When changing input pins, this LSI may malfunction due to the internal edge.
  To change input pins, the following procedure must be performed.
  — Disable the input function by the corresponding on-chip peripheral module setting
  — Select another input pin by PFCR
  — Enable its input function by the corresponding on-chip peripheral module settings
- If a pin function has both a select bit that modifies the input/output destination and an
  bit that enables the pin function, first specify the input/output destination by the select
  and then enable the pin function by the enable bit.

- The following operations can be set for each channel:
  - Waveform output at compare match
  - Input capture function
  - Counter clear operation
  - Synchronous operations:
    - Multiple timer counters (TCNT) can be written to simultaneously
    - Simultaneous clearing by compare match and input capture possible
    - Simultaneous input/output for registers possible by counter synchronous oper
    - Maximum of 15-phase PWM output possible by combination with synchronc operation
- Buffer operation settable for channels 0 and 3
- Phase counting mode settable independently for each of channels 1, 2, 4, and 5
- Cascaded operation
- Fast access via internal 16-bit bus
- 26 interrupt sources
- Automatic transfer of register data
- Programmable pulse generator (PPG) output trigger can be generated
- Conversion start trigger for the A/D converter can be generated
- Module stop state specifiable

(TGR)		TGRB_0	TGRB_1	TGRB_2	TGRB_3	TGRB_4	TG
General registers/ buffer registers		TGRC_0 TGRD_0	—	—	TGRC_3 TGRD_3	—	—
I/O pins		TIOCA0 TIOCB0 TIOCC0 TIOCD0	TIOCA1 TIOCB1	TIOCA2 TIOCB2	TIOCA3 TIOCB3 TIOCC3 TIOCD3	TIOCA4 TIOCB4	TI TI
Counter clear function		TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TG co ma inp ca
Compare match output	0 output	O	O	O	O	O	O
	1 output	O	O	O	O	O	O
	Toggle output	O	O	O	O	O	O
Input capture function		O	O	O	O	O	O
Synchronous operation		O	O	O	O	O	O
PWM mode		O	O	O	O	O	O
Phase counting mode		—	O	O	—	O	O
Buffer operation		O	—	—	O	—	—
DTC activation		TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TG co ma inp ca

PPG trigger	TGRA_0/ TGRB_0 compare match or input capture	TGRA_1/ TGRB_1 compare match or input capture	TGRA_2/ TGRB_2 compare match or input capture	TGRA_3/ TGRB_3 compare match or input capture		
Interrupt sources	5 sources Compare match or input capture 0A Compare match or input capture 0B Compare match or input capture 0C Compare match or input capture 0D Overflow	4 sources Compare match or input capture 1A Compare match or input capture 1B Overflow Underflow	4 sources Compare match or input capture 2A Compare match or input capture 2B Overflow Underflow	5 sources Compare match or input capture 3A Compare match or input capture 3B Compare match or input capture 3C Compare match or input capture 3D Overflow	4 sources Compare match or input capture 4A Compare match or input capture 4B Overflow Underflow	4 C m in c C m in c C U

[Legend]

○ : Possible

— : Not possible

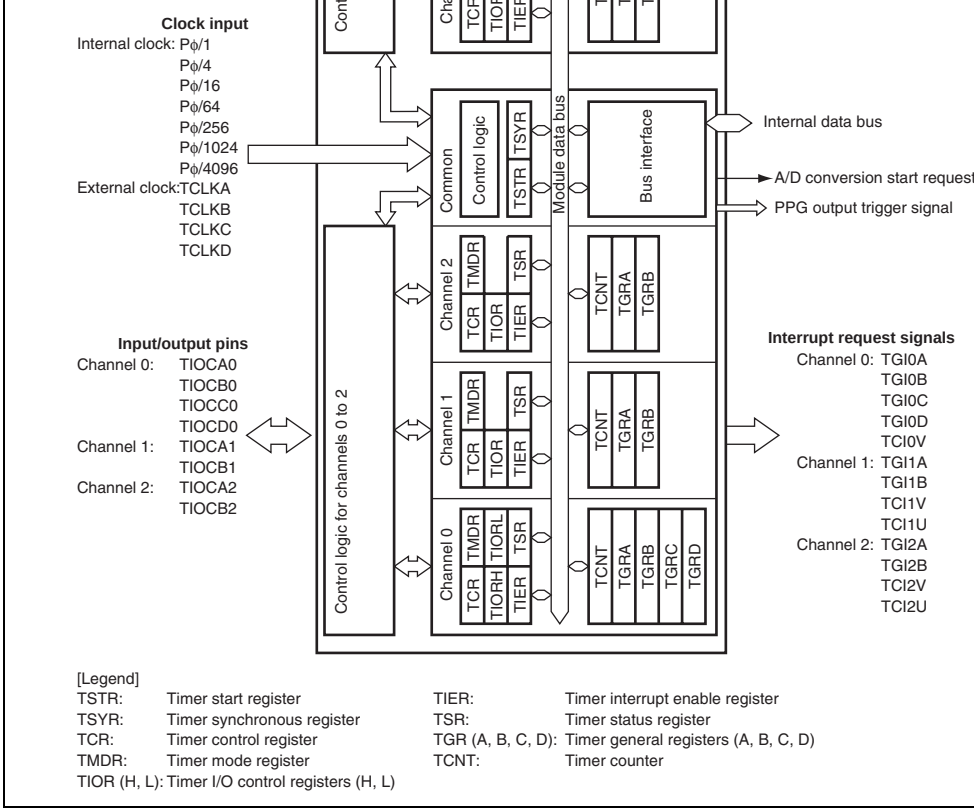


Figure 10.1 Block Diagram of TPU

			(Channel 1 and 5 phase counting mode B phase input)
	TCLKC	Input	External clock C input pin (Channel 2 and 4 phase counting mode A phase input)
	TCLKD	Input	External clock D input pin (Channel 2 and 4 phase counting mode B phase input)
0	TIOCA0	I/O	TGRA_0 input capture input/output compare output/PWM o
	TIOCB0	I/O	TGRB_0 input capture input/output compare output/PWM o
	TIOCC0	I/O	TGRC_0 input capture input/output compare output/PWM o
	TIOCD0	I/O	TGRD_0 input capture input/output compare output/PWM o
1	TIOCA1	I/O	TGRA_1 input capture input/output compare output/PWM o
	TIOCB1	I/O	TGRB_1 input capture input/output compare output/PWM o
2	TIOCA2	I/O	TGRA_2 input capture input/output compare output/PWM o
	TIOCB2	I/O	TGRB_2 input capture input/output compare output/PWM o
3	TIOCA3	I/O	TGRA_3 input capture input/output compare output/PWM o
	TIOCB3	I/O	TGRB_3 input capture input/output compare output/PWM o
	TIOCC3	I/O	TGRC_3 input capture input/output compare output/PWM o
	TIOCD3	I/O	TGRD_3 input capture input/output compare output/PWM o
4	TIOCA4	I/O	TGRA_4 input capture input/output compare output/PWM o
	TIOCB4	I/O	TGRB_4 input capture input/output compare output/PWM o
5	TIOCA5	I/O	TGRA_5 input capture input/output compare output/PWM o
	TIOCB5	I/O	TGRB_5 input capture input/output compare output/PWM o

— 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)

- Channel 1
  — 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)
- Channel 2
  — 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)

— Timer general register B_3 (TGRB_3)
— Timer general register C_3 (TGRC_3)
— Timer general register D_3 (TGRD_3)

- Channel 4
  — Timer control register_4 (TCR_4)
  — Timer mode register_4 (TMDR_4)
  — Timer I/O control register _4 (TIOR_4)
  — Timer interrupt enable register_4 (TIER_4)
  — Timer status register_4 (TSR_4)
  — Timer counter_4 (TCNT_4)
  — Timer general register A_4 (TGRA_4)
  — Timer general register B_4 (TGRB_4)
- Channel 5
  — Timer control register_5 (TCR_5)
  — Timer mode register_5 (TMDR_5)
  — Timer I/O control register_5 (TIOR_5)
  — Timer interrupt enable register_5 (TIER_5)
  — Timer status register_5 (TSR_5)
  — Timer counter_5 (TCNT_5)
  — Timer general register A_5 (TGRA_5)
  — Timer general register B_5 (TGRB_5)
- Common Registers
  — Timer start register (TSTR)
  — Timer synchronous register (TSYR)

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 TCNT counter clearing sou
5	CCLR0	0	R/W	details, see tables 10.3 and 10.4.
4	CKEG1	0	R/W	Clock Edge 1 and 0
3	CKEG0	0	R/W	These bits select the input clock edge. For detail table 10.5. When the input clock is counted using edges, the input clock period is halved (e.g. Pϕ/4 edges = Pϕ/2 rising edge). If phase counting mo used on channels 1, 2, 4, and 5, this setting is ig and the phase counting mode setting has priority clock edge selection is valid when the input clock or slower. This setting is ignored if the input cloc or when overflow/underflow of another channel is selected.
2	TPSC2	0	R/W	Timer Prescaler 2 to 0
1	TPSC1	0	R/W	These bits select the TCNT counter clock. The c
0	TPSC0	0	R/W	source can be selected independently for each c For details, see tables 10.6 to 10.11. To select th external clock as the clock source, the DDR bit a bit for the corresponding pin should be set to 0 a respectively. For details, see section 9, I/O Ports

			synchronous operation
1	0	0	TCNT clearing disabled
1	0	1	TCNT cleared by TGRC compare mat capture*[2]
1	1	0	TCNT cleared by TGRD compare mat capture*[2]
1	1	1	TCNT cleared by counter clearing for a channel performing synchronous clear synchronous operation*[1]

Notes: 1. Synchronous operation is selected by setting the SYNC bit in TSYR to 1.
2. When TGRC or TGRD is used as a buffer register, TCNT is not cleared beca buffer register setting has priority, and compare match/input capture does not

Table 10.4 CCLR2 to CCLR0 (Channels 1, 2, 4, and 5)

Channel	Bit 7 Reserved *[2]	Bit 6 CCLR1	Bit 5 CCLR0	Description
1, 2, 4, 5	0	0	0	TCNT clearing disabled
	0	0	1	TCNT cleared by TGRA compare mat capture
	0	1	0	TCNT cleared by TGRB compare mat capture
	0	1	1	TCNT cleared by counter clearing for a channel performing synchronous clear synchronous operation*[1]

Notes: 1. Synchronous operation is selected by setting the SYNC bit in TSYR to 1.
2. Bit 7 is reserved in channels 1, 2, 4, and 5. It is always read as 0 and cannot modified.

Table 10.6 TPSC2 to TPSC0 (Channel 0)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
0	0	0	0	Internal clock: counts on Pϕ/1
	0	0	1	Internal clock: counts on Pϕ/4
	0	1	0	Internal clock: counts on Pϕ/16
	0	1	1	Internal clock: counts on Pϕ/64
	1	0	0	External clock: counts on TCLKA pin inp
	1	0	1	External clock: counts on TCLKB pin inp
	1	1	0	External clock: counts on TCLKC pin inp
	1	1	1	External clock: counts on TCLKD pin inp

Table 10.7 TPSC2 to TPSC0 (Channel 1)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
1	0	0	0	Internal clock: counts on Pϕ/1
	0	0	1	Internal clock: counts on Pϕ/4
	0	1	0	Internal clock: counts on Pϕ/16
	0	1	1	Internal clock: counts on Pϕ/64
	1	0	0	External clock: counts on TCLKA pin inp
	1	0	1	External clock: counts on TCLKB pin inp
	1	1	0	Internal clock: counts on Pϕ/256
	1	1	1	Counts on TCNT2 overflow/underflow

Note: This setting is ignored when channel 1 is in phase counting mode.

	1	1	0	External clock: counts on TCLKC pin i
	1	1	1	Internal clock: counts on Pϕ/1024

Note: This setting is ignored when channel 2 is in phase counting mode.

Table 10.9 TPSC2 to TPSC0 (Channel 3)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
3	0	0	0	Internal clock: counts on Pϕ/1
	0	0	1	Internal clock: counts on Pϕ/4
	0	1	0	Internal clock: counts on Pϕ/16
	0	1	1	Internal clock: counts on Pϕ/64
	1	0	0	External clock: counts on TCLKA pin i
	1	0	1	Internal clock: counts on Pϕ/1024
	1	1	0	Internal clock: counts on Pϕ/256
	1	1	1	Internal clock: counts on Pϕ/4096

	1	1	0	Internal clock: counts on Pϕ/1024
	1	1	1	Counts on TCNT5 overflow/underflow

Note: This setting is ignored when channel 4 is in phase counting mode.

Table 10.11 TPSC2 to TPSC0 (Channel 5)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
5	0	0	0	Internal clock: counts on Pϕ/1
	0	0	1	Internal clock: counts on Pϕ/4
	0	1	0	Internal clock: counts on Pϕ/16
	0	1	1	Internal clock: counts on Pϕ/64
	1	0	0	External clock: counts on TCLKA pin in
	1	0	1	External clock: counts on TCLKC pin in
	1	1	0	Internal clock: counts on Pϕ/256
	1	1	1	External clock: counts on TCLKD pin in

Note: This setting is ignored when channel 5 is in phase counting mode.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 1	R	Reserved These are read-only bits and cannot be modifie
5	BFB	0	R/W	Buffer Operation B Specifies whether TGRB is to normally operate and TGRD are to be used together for buffer op When TGRD is used as a buffer register, TGRD capture/output compare is not generated. In channels 1, 2, 4, and 5, which have no TGRD reserved. It is always read as 0 and cannot be 0: TGRB operates normally 1: TGRB and TGRD used together for buffer op
4	BFA	0	R/W	Buffer Operation A Specifies whether TGRA is to normally operate and TGRC are to be used together for buffer op When TGRC is used as a buffer register, TGRC capture/output compare is not generated. In channels 1, 2, 4, and 5, which have no TGRC reserved. It is always read as 0 and cannot be 0: TGRA operates normally 1: TGRA and TGRC used together for buffer op
3	MD3	0	R/W	Modes 3 to 0
2	MD2	0	R/W	Set the timer operating mode.
1	MD1	0	R/W	MD3 is a reserved bit. The write value should a
0	MD0	0	R/W	0. For details, see table 10.12 for details.

0	1	1	0	Phase counting mode 3
0	1	1	1	Phase counting mode 4
1	X	X	X	—

[Legend]

X: Don't care

Notes: 1. MD3 is a reserved bit. The write value should always be 0.

2. Phase counting mode cannot be set for channels 0 and 3. In this case, 0 shou be written to MD2.

10.3.3 Timer I/O Control Register (TIOR)

TIOR controls TGR. The TPU has eight TIOR registers, two each for channels 0 and 3, a each for channels 1, 2, 4, and 5. Care is required since TIOR is affected by the TMDR set

The initial output specified by TIOR is valid when the counter is stopped (the CST bit in cleared to 0). Note also that, in PWM mode 2, the output at the point at which the counter cleared to 0 is specified.

When TGRC or TGRD is designated for buffer operation, this setting is invalid and the re operates as a buffer register.

To designate the input capture pin in TIOR, the DDR bit and ICR bit for the correspondin should be set to 0 and 1, respectively. For details, see section 9, I/O Ports.

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- TIORH_0, TIOR_1, TIOR_2, TIORH_3, TIOR_4, TIOR_5

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	For details, see tables 10.13, 10.15, 10.16, 10.1
4	IOB0	0	R/W	and 10.20.
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	For details, see tables 10.21, 10.23, 10.24, 10.2
0	IOA0	0	R/W	and 10.28.

- TIORL_0, TIORL_3:

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	For details, see tables 10.14 and 10.18.
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	For details, see tables 10.22 and 10.26.
0	IOC0	0	R/W	

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	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	0	0	1		Capture input source is TIOCB0 pin Input capture at falling edge
1	0	1	x		Capture input source is TIOCB0 pin Input capture at both edges
1	1	x	x		Capture input source is channel 1/count Input capture at TCNT_1 count-up/count

[Legend]

X: Don't care

Note: When bits TPSC2 to TPSC0 in TCR_1 are set to B'000 and Pϕ/1 is used as the TC count clock, this setting is invalid and input capture is not generated.

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register*2	Capture input source is TIOCD0 pin Input capture at rising edge
1	0	0	1		Capture input source is TIOCD0 pin Input capture at falling edge
1	0	1	X		Capture input source is TIOCD0 pin Input capture at both edges
1	1	X	X		Capture input source is channel 1/coun Input capture at TCNT_1 count-up/cou

[Legend]

X: Don't care

Notes: 1. When bits TPSC2 to TPSC0 in TCR_1 are set to B'000 and Pϕ/1 is used as t
TCNT_1 count clock, this setting is invalid and input capture is not generated
2. When the BFB bit in TMDR_0 is set to 1 and TGRD_0 is used as a buffer reg
setting is invalid and input capture/output compare is not generated.

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	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	0	0	1		Capture input source is TIOCB1 pin Input capture at falling edge
1	0	1	X		Capture input source is TIOCB1 pin Input capture at both edges
1	1	X	X		TGRC_0 compare match/input capture Input capture at generation of TGRC_0 c match/input capture

[Legend]

X: Don't care

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	1		Initial output is 1 output Toggle output at compare match
1	X	0	0	Input capture register	Capture input source is TIOCB2 pin Input capture at rising edge
1	X	0	1		Capture input source is TIOCB2 pin Input capture at falling edge
1	X	1	X		Capture input source is TIOCB2 pin Input capture at both edges

[Legend]
X: Don't care

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCB3 pin Input capture at rising edge
1	0	0	1		Capture input source is TIOCB3 pin Input capture at falling edge
1	0	1	x		Capture input source is TIOCB3 pin Input capture at both edges
1	1	x	x		Capture input source is channel 4/count Input capture at TCNT_4 count-up/count

[Legend]

X: Don't care

Note: When bits TPSC2 to TPSC0 in TCR_4 are set to B'000 and Pϕ/1 is used as the TC count clock, this setting is invalid and input capture is not generated.

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register*2	Capture input source is TIOCD3 pin Input capture at rising edge
1	0	0	1		Capture input source is TIOCD3 pin Input capture at falling edge
1	0	1	x		Capture input source is TIOCD3 pin Input capture at both edges
1	1	x	x		Capture input source is channel 4/coun Input capture at TCNT_4 count-up/cou

[Legend]

X: Don't care

Notes: 1. When bits TPSC2 to TPSC0 in TCR_4 are set to B'000 and Pϕ/1 is used as t
TCNT_4 count clock, this setting is invalid and input capture is not generated

2. When the BFB bit in TMDR_3 is set to 1 and TGRD_3 is used as a buffer reg
setting is invalid and input capture/output compare is not generated.

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCB4 pin Input capture at rising edge
1	0	0	1		Capture input source is TIOCB4 pin Input capture at falling edge
1	0	1	x		Capture input source is TIOCB4 pin Input capture at both edges
1	1	x	x		Capture input source is TGRC_3 compa match/input capture Input capture at generation of TGRC_3 c match/input capture

[Legend]
X: Don't care

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	1		Initial output is 1 output Toggle output at compare match
1	x	0	0	Input capture register	Capture input source is TIOCB5 pin Input capture at rising edge
1	x	0	1		Capture input source is TIOCB5 pin Input capture at falling edge
1	x	1	x		Capture input source is TIOCB5 pin Input capture at both edges

[Legend]
X: Don't care

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	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	0	0	1		Capture input source is TIOCA0 pin Input capture at falling edge
1	0	1	X		Capture input source is TIOCA0 pin Input capture at both edges
1	1	X	X		Capture input source is channel 1/count Input capture at TCNT_1 count-up/count

[Legend]
X: Don't care

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	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	0	0	1		Capture input source is TIOCC0 pin Input capture at falling edge
1	0	1	X		Capture input source is TIOCC0 pin Input capture at both edges
1	1	X	X		Capture input source is channel 1/coun Input capture at TCNT_1 count-up/cou

[Legend]

X: Don't care

Note: 1. When the BFA bit in TMDR_0 is set to 1 and TGRC_0 is used as a buffer reg setting is invalid and input capture/output compare is not generated.

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	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	0	0	1		Capture input source is TIOCA1 pin Input capture at falling edge
1	0	1	X		Capture input source is TIOCA1 pin Input capture at both edges
1	1	X	X		Capture input source is TGRA_0 compa match/input capture Input capture at generation of channel 0/ compare match/input capture

[Legend]
X: Don't care

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	1		Initial output is 1 output Toggle output at compare match
1	X	0	0	Input capture register	Capture input source is TIOCA2 pin Input capture at rising edge
1	X	0	1		Capture input source is TIOCA2 pin Input capture at falling edge
1	X	1	X		Capture input source is TIOCA2 pin Input capture at both edges

[Legend]
X: Don't care

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA3 pin Input capture at rising edge
1	0	0	1		Capture input source is TIOCA3 pin Input capture at falling edge
1	0	1	X		Capture input source is TIOCA3 pin Input capture at both edges
1	1	X	X		Capture input source is channel 4/count Input capture at TCNT_4 count-up/count

[Legend]
X: Don't care

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register*	Capture input source is TIOCC3 pin Input capture at rising edge
1	0	0	1		Capture input source is TIOCC3 pin Input capture at falling edge
1	0	1	X		Capture input source is TIOCC3 pin Input capture at both edges
1	1	X	X		Capture input source is channel 4/coun Input capture at TCNT_4 count-up/cou

[Legend]

X: Don't care

Note: * When the BFA bit in TMDR_3 is set to 1 and TGRC_3 is used as a buffer reg setting is invalid and input capture/output compare is not generated.

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA4 pin Input capture at rising edge
1	0	0	1		Capture input source is TIOCA4 pin Input capture at falling edge
1	0	1	X		Capture input source is TIOCA4 pin Input capture at both edges
1	1	X	X		Capture input source is TGRA_3 compa match/input capture Input capture at generation of TGRA_3 c match/input capture

[Legend]
X: Don't care

0	0	1	1		Initial output is 0 output Toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is 1 output 0 output at compare match
0	1	1	0		Initial output is 1 output 1 output at compare match
0	1	1	1		Initial output is 1 output Toggle output at compare match
1	X	0	0	Input capture register	Input capture source is TIOCA5 pin Input capture at rising edge
1	X	0	1		Input capture source is TIOCA5 pin Input capture at falling edge
1	X	1	X		Input capture source is TIOCA5 pin Input capture at both edges

[Legend]
X: Don't care

Bit	Bit Name	Initial value	R/W	Description
7	TTGE	0	R/W	A/D Conversion Start Request Enable Enables/disables generation of A/D conversion s requests by TGRA input capture/compare match 0: A/D conversion start request generation disab 1: A/D conversion start request generation enabl
6	—	1	R	Reserved This is a read-only bit and cannot be modified.
5	TCIEU	0	R/W	Underflow Interrupt Enable Enables/disables interrupt requests (TCIU) by th flag when the TCFU flag in TSR is set to 1 in cha 2, 4, and 5. In channels 0 and 3, bit 5 is reserved. It is alway 0 and cannot be modified. 0: Interrupt requests (TCIU) by TCFU disabled 1: Interrupt requests (TCIU) by TCFU enabled
4	TCIEV	0	R/W	Overflow Interrupt Enable Enables/disables interrupt requests (TCIV) by th 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

2	TGIEC	0	R/W	TGR Interrupt Enable C Enables/disables interrupt requests (TGIC) by t bit when the TGFC bit in TSR is set to 1 in char and 3. In channels 1, 2, 4, and 5, bit 2 is reserved. It is read as 0 and cannot be modified. 0: Interrupt requests (TGIC) by TGFC bit disabl 1: Interrupt requests (TGIC) by TGFC bit enabl
1	TGIEB	0	R/W	TGR Interrupt Enable B Enables/disables interrupt requests (TGIB) by t bit when the TGFB bit in TSR is set to 1. 0: Interrupt requests (TGIB) by TGFB bit disabl 1: Interrupt requests (TGIB) by TGFB bit enable
0	TGIEA	0	R/W	TGR Interrupt Enable A Enables/disables interrupt requests (TGIA) by t bit when the TGFA bit in TSR is set to 1. 0: Interrupt requests (TGIA) by TGFA bit disabl 1: Interrupt requests (TGIA) by TGFA bit enable

Bit	Bit Name	Initial value	R/W	Description
7	TCFD	1	R	Count Direction Flag Status flag that shows the direction in which TCN in channels 1, 2, 4, and 5. In channels 0 and 3, bit 7 is reserved. It is alway 1 and cannot be modified. 0: TCNT counts down 1: TCNT counts up
6	—	1	R	Reserved This is a read-only bit and cannot be modified.
5	TCFU	0	R/(W)*	Underflow Flag Status flag that indicates that a TCNT underflow occurred when channels 1, 2, 4, and 5 are set to counting mode. In channels 0 and 3, bit 5 is reserved. It is alway 0 and cannot be modified. [Setting condition] • When the TCNT value underflows (changes H'0000 to H'FFFF) [Clearing condition] • When a 0 is written to TCFU after reading TC (When the CPU is used to clear this flag by w while the corresponding interrupt is enabled, to read the flag after writing 0 to it.)

				while the corresponding interrupt is enablec to read the flag after writing 0 to it.)
3	TGFD	0	R/(W)*	Input Capture/Output Compare Flag D Status flag that indicates the occurrence of TGF capture or compare match in channels 0 and 3. In channels 1, 2, 4, and 5, bit 3 is reserved. It is read as 0 and cannot be modified. [Setting conditions] • When TCNT = TGRD while TGRD is functio output compare register • When TCNT value is transferred to TGRD b capture signal while TGRD is functioning as capture register [Clearing conditions] • When DTC is activated by a TGID interrupt DISEL bit in MRB of DTC is 0 • When 0 is written to TGFD after reading TG (When the CPU is used to clear this flag by while the corresponding interrupt is enablec to read the flag after writing 0 to it.)

				• When TCNT value is transferred to TGRC by capture signal while TGRC is functioning as i capture register [Clearing conditions] • When DTC is activated by a TGIC interrupt w DISEL bit in MRB of DTC is 0 • When 0 is written to TGFC after reading TGF (When the CPU is used to clear this flag by w while the corresponding interrupt is enabled, to read the flag after writing 0 to it.)
1	TGFB	0	R/(W)*	Input Capture/Output Compare Flag B Status flag that indicates the occurrence of TGR capture or compare match. [Setting conditions] • When TCNT = TGRB while TGRB is function output compare register • When TCNT value is transferred to TGRB by capture signal while TGRB is functioning as i capture register [Clearing conditions] • When DTC is activated by a TGIB interrupt w DISEL bit in MRB of DTC is 0 • When 0 is written to TGFB after reading TGF (When the CPU is used to clear this flag by w while the corresponding interrupt is enabled, to read the flag after writing 0 to it.)

capture register

[Clearing conditions]

- When DTC is activated by a TGIA interrupt
  DISEL bit in MRB of DTC is 0
- When DMAC is activated by a TGIA interrup
  the DTA bit in DMDR of DMAC is 1
- When 0 is written to TGFA after reading TG
  (When the CPU is used to clear this flag by
  while the corresponding interrupt is enablec
  to read the flag after writing 0 to it.)

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

Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Bit	7	6	5	4	3	2	1	
Bit Name								
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

10.3.7 Timer General Register (TGR)

TGR is a 16-bit readable/writable register with a dual function as output compare and inp
capture registers. The TPU has 16 TGR registers, four each for channels 0 and 3 and two
channels 1, 2, 4, and 5. TGRC and TGRD for channels 0 and 3 can also be designated for
operation as buffer registers. The TGR registers cannot be accessed in 8-bit units; they m
always be accessed in 16-bit units. TGR and buffer register combinations during buffer o
are TGRA–TGRC and TGRB–TGRD.

Bit	15	14	13	12	11	10	9	
Bit Name								
Initial Value	1	1	1	1	1	1	1	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Bit	7	6	5	4	3	2	1	
Bit Name								
Initial Value	1	1	1	1	1	1	1	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Bit	Bit Name	Initial value	R/W	Description
7, 6	—	All 0	R/W	Reserved These bits are always read as 0. The write valu always be 0.
5	CST5	0	R/W	Counter Start 5 to 0
4	CST4	0	R/W	These bits select operation or stoppage for TCN
3	CST3	0	R/W	If 0 is written to the CST bit during operation wi
2	CST2	0	R/W	TIOC pin designated for output, the counter sto
1	CST1	0	R/W	TIOC pin output compare output level is retaine is written to when the CST bit is cleared to 0, th
0	CST0	0	R/W	output level will be changed to the set initial out 0: TCNT_5 to TCNT_0 count operation is stopp 1: TCNT_5 to TCNT_0 performs count operatio

Bit	Bit Name	Initial value	R/W	Description
7, 6	—	All 0	R/W	Reserved These bits are always read as 0. The write value always be 0.
5	SYNC5	0	R/W	Timer Synchronization 5 to 0
4	SYNC4	0	R/W	These bits select whether operation is independe synchronized with other channels.
3	SYNC3	0	R/W	
2	SYNC2	0	R/W	When synchronous operation is selected, synchr presetting of multiple channels, and synchronous through counter clearing on another channel are
1	SYNC1	0	R/W	
0	SYNC0	0	R/W	To set synchronous operation, the SYNC bits for two channels must be set to 1. To set synchronc clearing, in addition to the SYNC bit, the TCNT c source must also be set by means of bits CCLR2 CCLR0 in TCR. 0: TCNT_5 to TCNT_0 operate independently (T presetting/clearing is unrelated to other chann 1: TCNT_5 to TCNT_0 perform synchronous ope (TCNT synchronous presetting/synchronous c is possible)

When one of bits CST0 to CST5 is set to 1 in TSTR, the TCNT counter for the correspo
channel starts counting. TCNT can operate as a free-running counter, periodic counter, a

(a) Example of count operation setting procedure

Figure 10.2 shows an example of the count operation setting procedure.

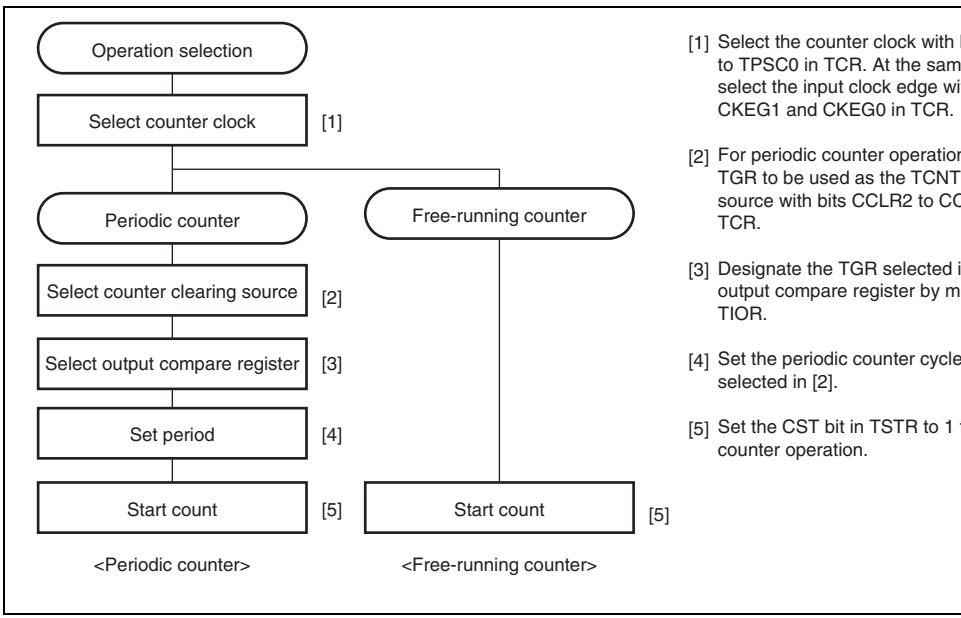


Figure 10.2 Example of Counter Operation Setting Procedure

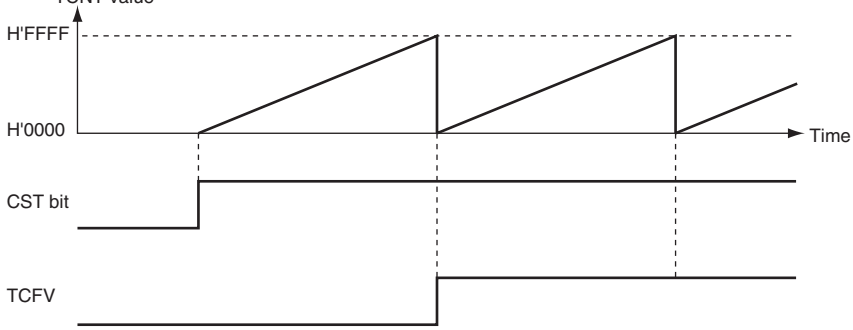


Figure 10.3 Free-Running Counter Operation

When compare match is selected as the TCNT clearing source, the TCNT counter for the
channel performs periodic count operation. The TGR register for setting the period is des
as an output compare register, and counter clearing by compare match is selected by mea
CCLR2 to CCLR0 in TCR. After the settings have been made, TCNT starts count-up ope
a periodic counter when the corresponding bit in TSTR is set to 1. When the count value
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
After a compare match, TCNT starts counting up again from H'0000.

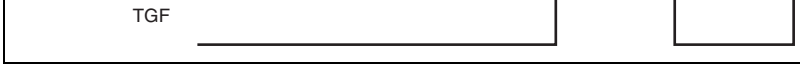


Figure 10.4 Periodic Counter Operation

(2) Waveform Output by Compare Match

The TPU can perform 0, 1, or toggle output from the corresponding output pin using a c
match.

(a) Example of setting procedure for waveform output by compare match

Figure 10.5 shows an example of the setting procedure for waveform output by a compa

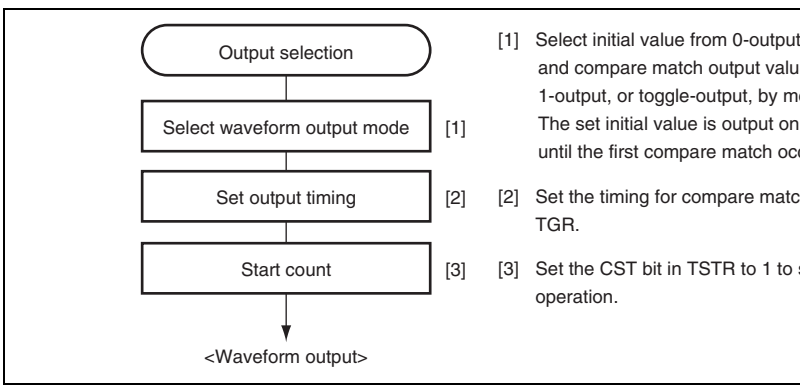


Figure 10.5 Example of Setting Procedure for Waveform Output by Compare

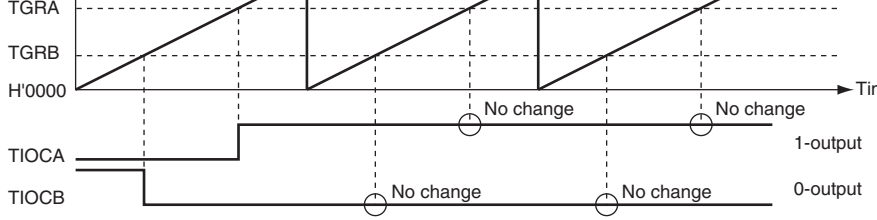


Figure 10.6 Example of 0-Output/1-Output Operation

Figure 10.7 shows an example of toggle output.

In this example, TCNT has been designated as a periodic counter (with counter clearing p
by compare match B), and settings have been made so that output is toggled by both com
match A and compare match B.

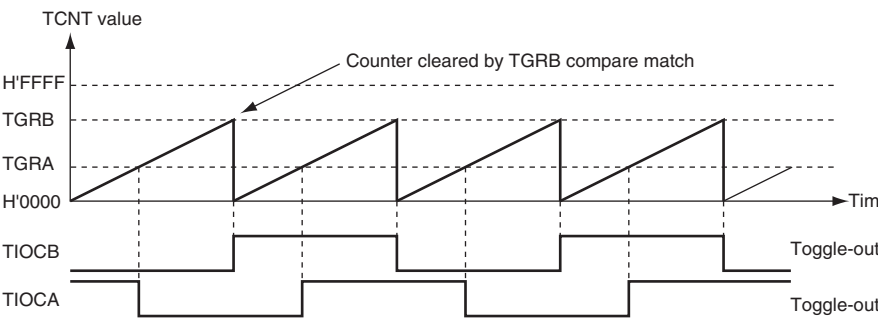


Figure 10.7 Example of Toggle Output Operation

(a) Example of setting procedure for input capture operation

Figure 10.8 shows an example of the setting procedure for input capture operation.

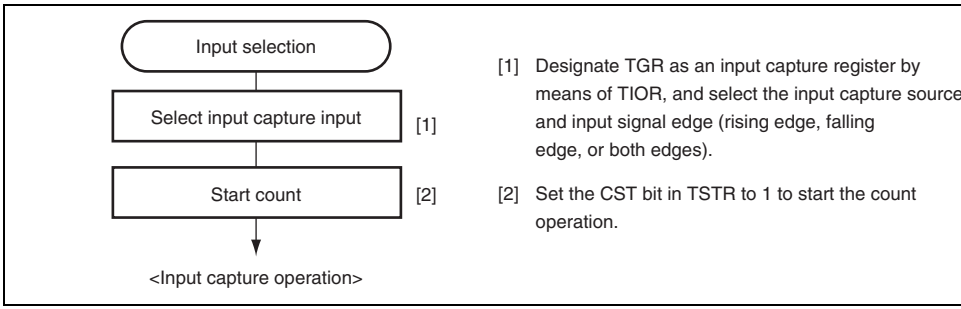


Figure 10.8 Example of Setting Procedure for Input Capture Operation

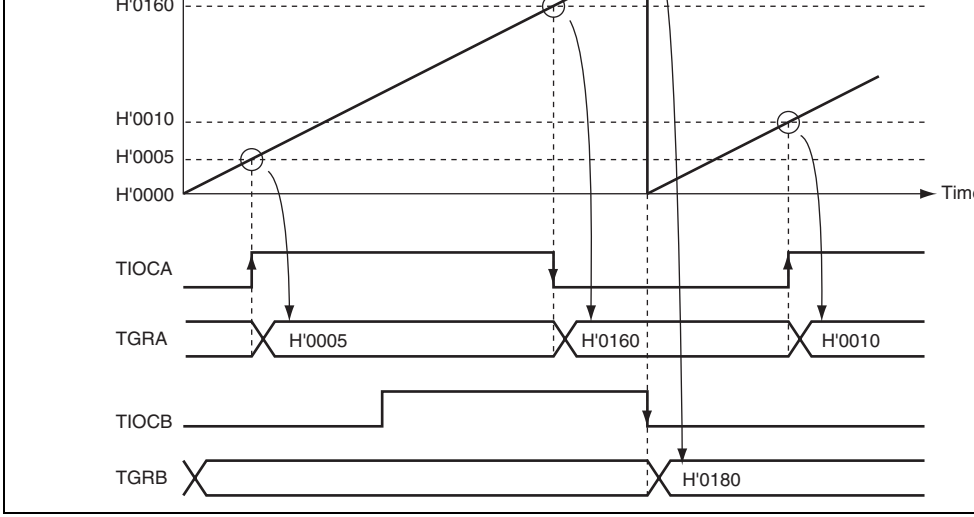


Figure 10.9 Example of Input Capture Operation

Figure 10.10 shows an example of the synchronous operation setting procedure.

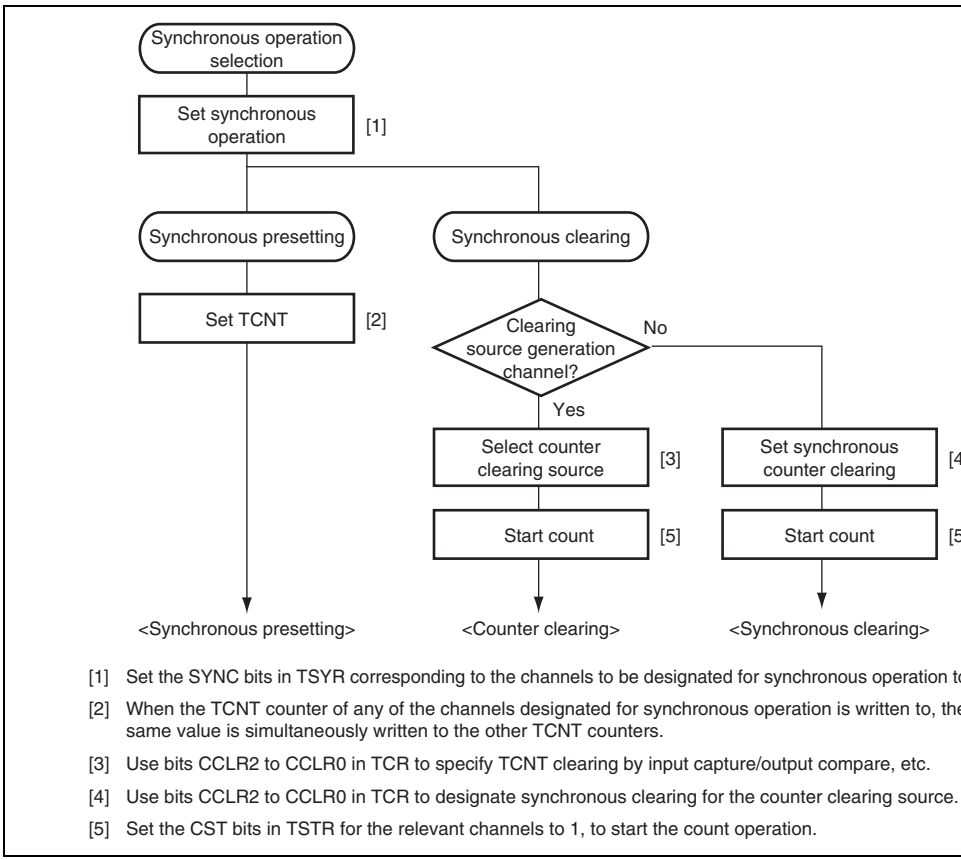


Figure 10.10 Example of Synchronous Operation Setting Procedure

For details on PWM modes, see section 10.4.5, PWM Modes.

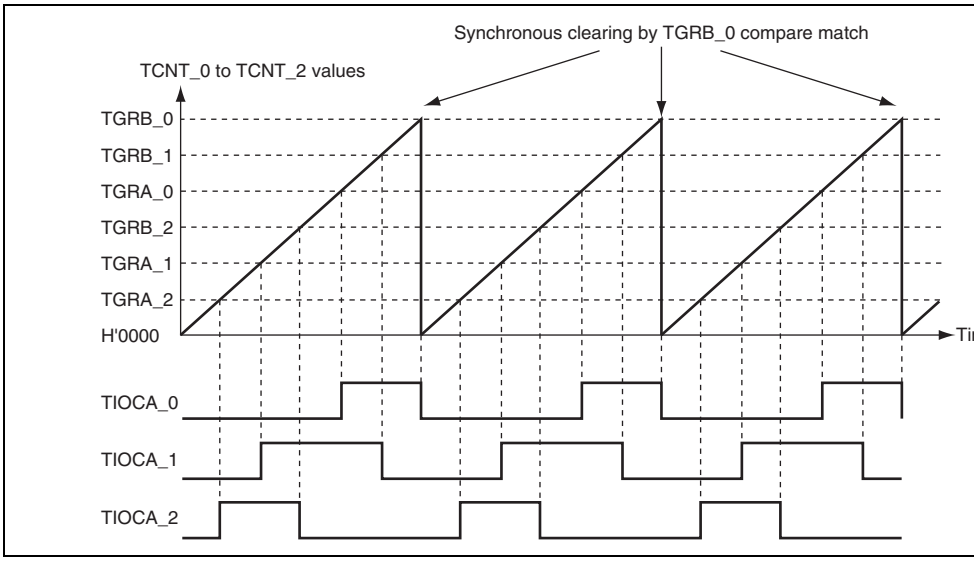


Figure 10.11 Example of Synchronous Operation

Channel	Timer General Register	Buffer Register
0	TGRA_0	TGRC_0
	TGRB_0	TGRD_0
3	TGRA_3	TGRC_3
	TGRB_3	TGRD_3

- When TGR is an output compare register

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

  This operation is illustrated in figure 10.12.

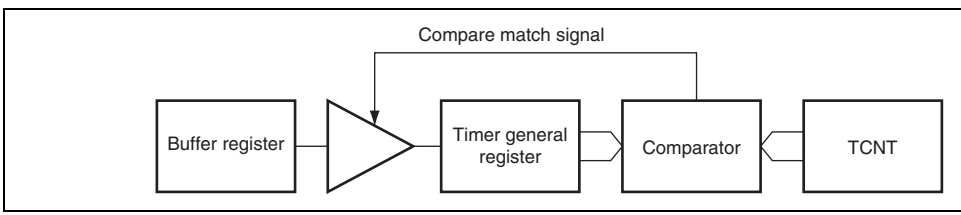
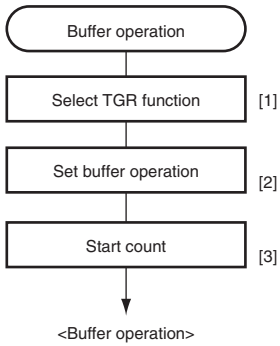


Figure 10.12 Compare Match Buffer Operation

(1) Example of Buffer Operation Setting Procedure

Figure 10.14 shows an example of the buffer operation setting procedure.



[1] Designate TGR as an input capture register or output compare register by means of TIOR.

[2] Designate TGR for buffer operation with bits BFA and BFB in TMDR.

[3] Set the CST bit in TSTR to 1 to start the count operation.

Figure 10.14 Example of Buffer Operation Setting Procedure

For details on PWM modes, see section 10.4.5, PWM Modes.

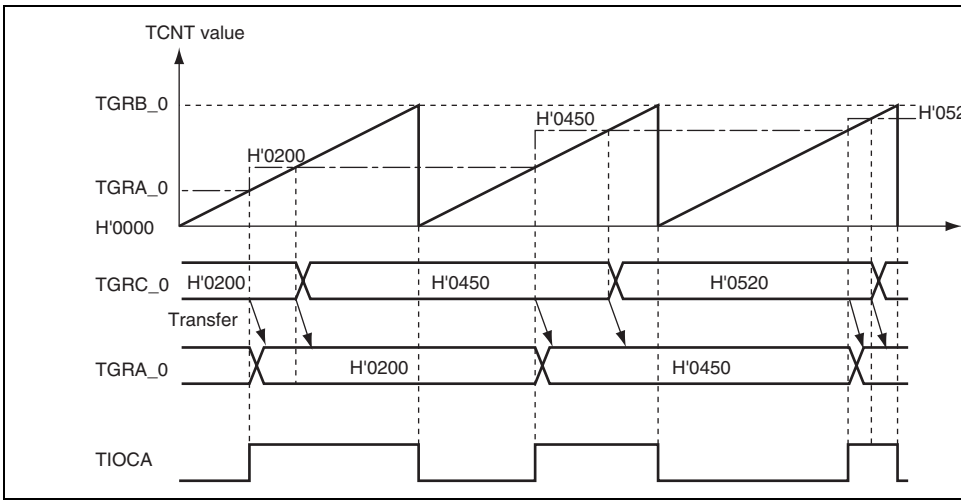


Figure 10.15 Example of Buffer Operation (1)

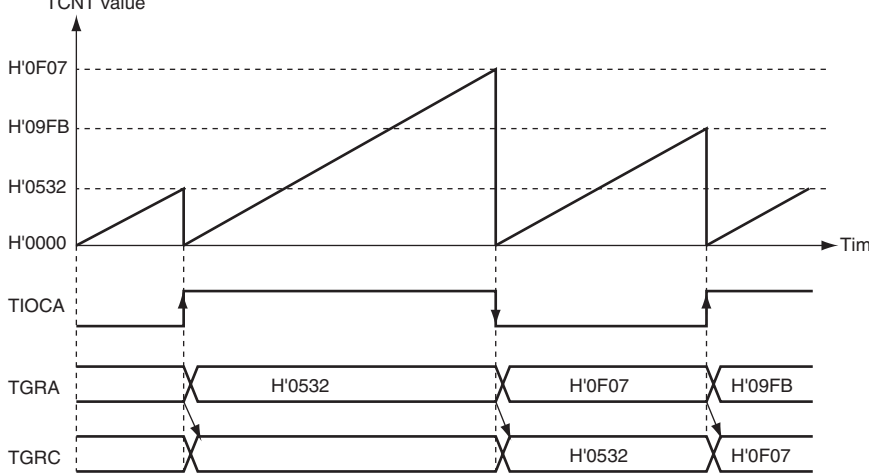


Figure 10.16 Example of Buffer Operation (2)

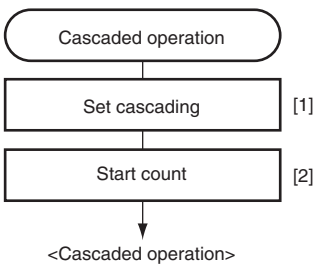
Note: When phase counting mode is set for channel 1 or 4, the counter clock setting is
and the counter operates independently in phase counting mode.

Table 10.30 Cascaded Combinations

Combination	Upper 16 Bits	Lower 16 Bits
Channels 1 and 2	TCNT_1	TCNT_2
Channels 4 and 5	TCNT_4	TCNT_5

(1) Example of Cascaded Operation Setting Procedure

Figure 10.17 shows an example of the setting procedure for cascaded operation.



[1] Set bits TPSC2 to TPSC0 in the channel 1 (channel 4) TCR to B'1111 to select TCNT_2 (TCNT_5) overflow/underflow counting.

[2] Set the CST bit in TSTR for the upper and lower channels to 1 to start the count operation.

Figure 10.17 Example of Cascaded Operation Setting Procedure

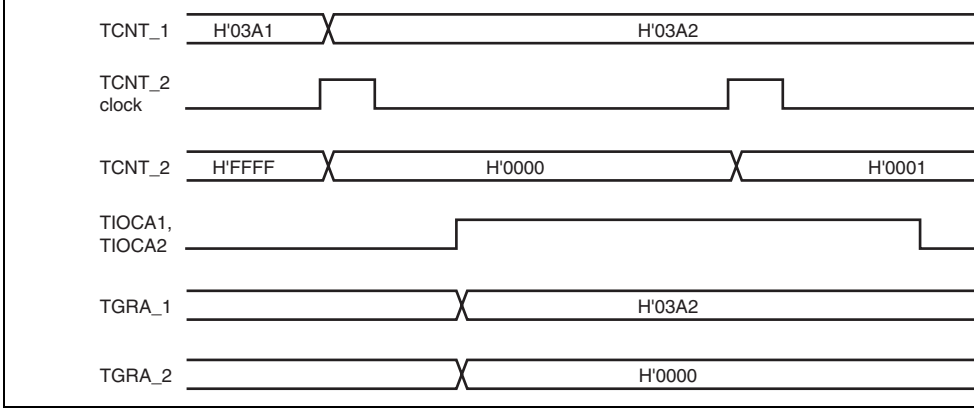


Figure 10.18 Example of Cascaded Operation (1)

Figure 10.19 illustrates the operation when counting upon TCNT_2 overflow/underflow l
set for TCNT_1, and phase counting mode has been designated for channel 2.

TCNT_1 is incremented by TCNT_2 overflow and decremented by TCNT_2 underflow.

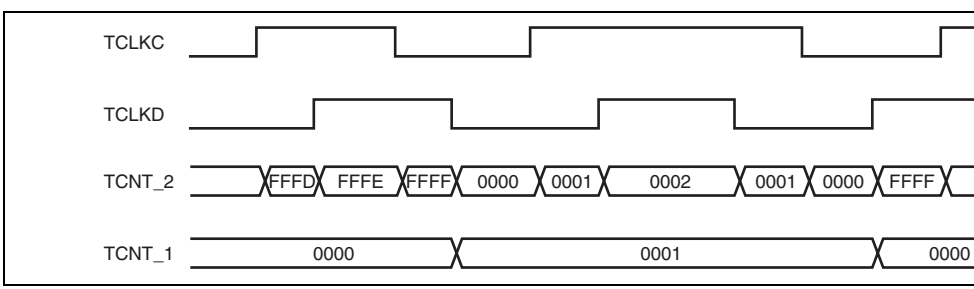


Figure 10.19 Example of Cascaded Operation (2)

There are two PWM modes, as described below.

1. PWM mode 1

   PWM output is generated from the TIOCA and TIOCC pins by pairing TGRA with T
   TGRC with TGRD. The outputs specified by bits IOA3 to IOA0 and IOC3 to IOC0 i
   are output from the TIOCA and TIOCC pins at compare matches A and C, respectiv
   outputs specified by bits IOB3 to IOB0 and IOD3 to IOD0 in TIOR are output at com
   matches B and D, respectively. The initial output value is the value set in TGRA or T
   the set values of paired TGRs are identical, the output value does not change when a
   match occurs.

   In PWM mode 1, a maximum 8-phase PWM output is possible.

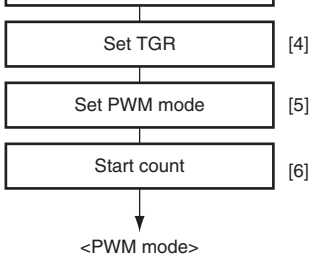
2. PWM mode 2

   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.
   counter clearing by a synchronous register compare match, the output value of each p
   initial value set in TIOR. If the set values of the cycle and duty cycle registers are ide
   output value does not change when a compare match occurs.

   In PWM mode 2, a maximum 15-phase PWM output is possible by combined use wi
   synchronous operation.

1	TGRA_1	TIOCA1	TIOCA1
	TGRB_1		TIOCB1
2	TGRA_2	TIOCA2	TIOCA2
	TGRB_2		TIOCB2
3	TGRA_3	TIOCA3	TIOCA3
	TGRB_3		TIOCB3
	TGRC_3	TIOCC3	TIOCC3
	TGRD_3		TIOCD3
4	TGRA_4	TIOCA4	TIOCA4
	TGRB_4		TIOCB4
5	TGRA_5	TIOCA5	TIOCA5
	TGRB_5		TIOCB5

Note: In PWM mode 2, PWM output is not possible for the TGR register in which the cyc



[4] Set the cycle in TGR selected in [2], and set the duty in the other TGRs.

[5] Select the PWM mode with bits MD3 to MD0 in TMDR.

[6] Set the CST bit in TSTR to 1 to start the count operation.

Figure 10.20 Example of PWM Mode Setting Procedure

(2) Examples of PWM Mode Operation

Figure 10.21 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 tl
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 cycle, and the value set in TGRB regis
duty cycle.

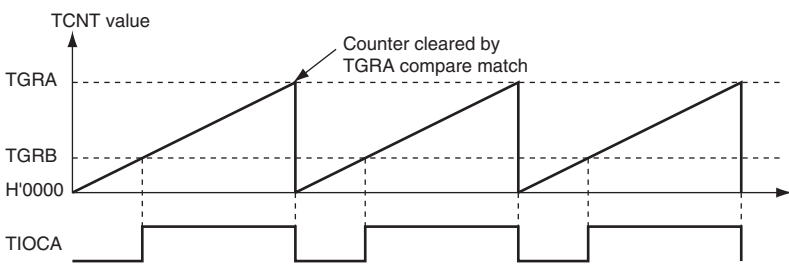


Figure 10.21 Example of PWM Mode Operation (1)

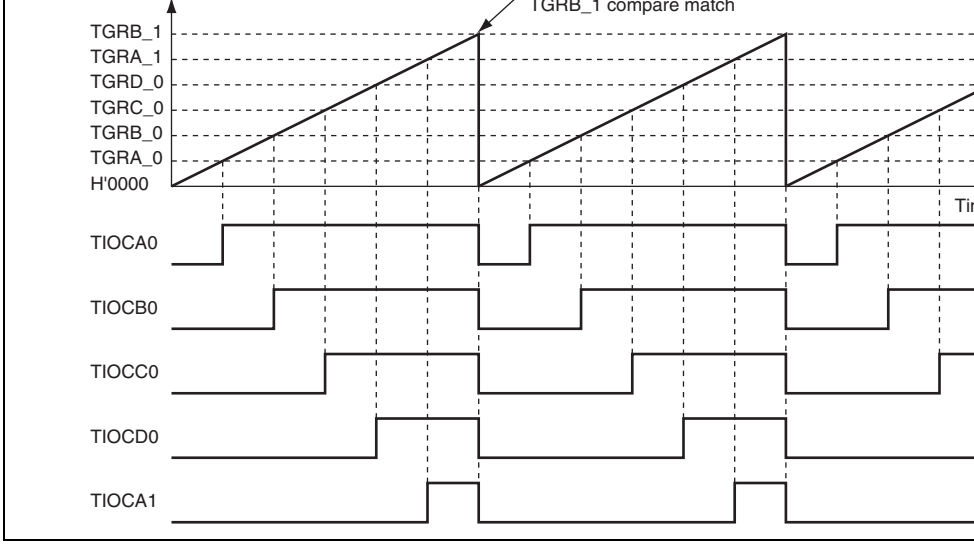


Figure 10.22 Example of PWM Mode Operation (2)

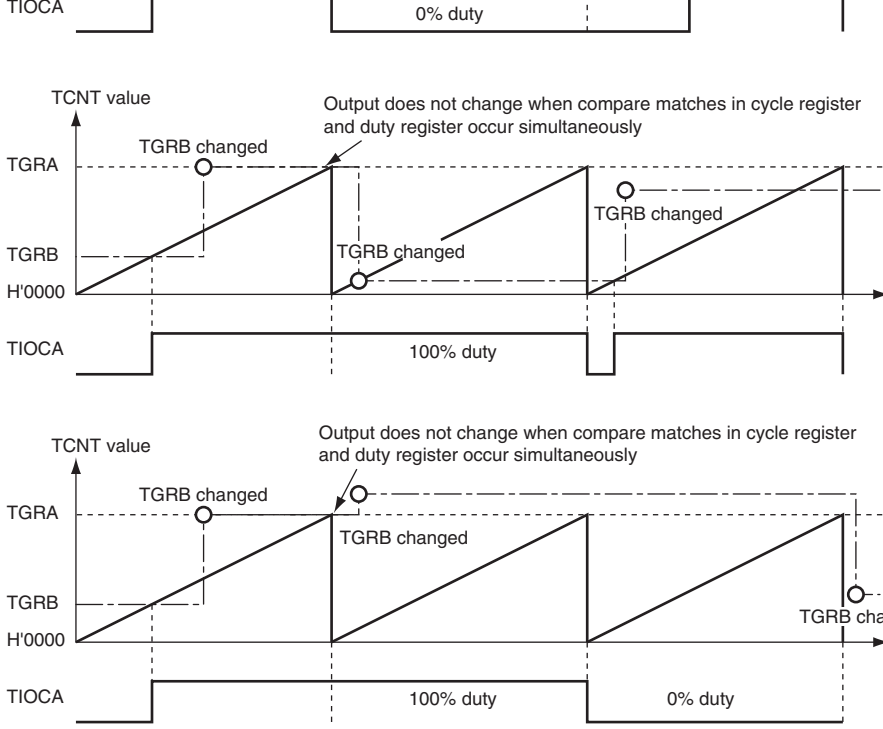


Figure 10.23 Example of PWM Mode Operation (3)

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 ur
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 indi
whether TCNT is counting up or down.

Table 10.32 shows the correspondence between external clock pins and channels.

Table 10.32 Clock Input Pins in Phase Counting Mode

	External Clock Pins	
Channels	**A-Phase**	**B-Phase**
When channel 1 or 5 is set to phase counting mode	TCLKA	TCLKB
When channel 2 or 4 is set to phase counting mode	TCLKC	TCLKD

<Phase counting mode>

Figure 10.24 Example of Phase Counting Mode Setting Procedure

(2) Examples of Phase Counting Mode Operation

In phase counting mode, TCNT counts up or down according to the phase difference bet
external clocks. There are four modes, according to the count conditions.

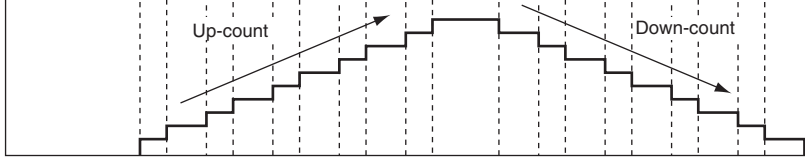


Figure 10.25 Example of Phase Counting Mode 1 Operation

Table 10.33 Up/Down-Count Conditions in Phase Counting Mode 1

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level	↑	Up-count
Low level	↓	
↑	Low level	
↓	High level	
High level	↓	Down-count
Low level	↑	
↑	High level	
↓	Low level	

[Legend]
↑ : Rising edge
↓: Falling edge

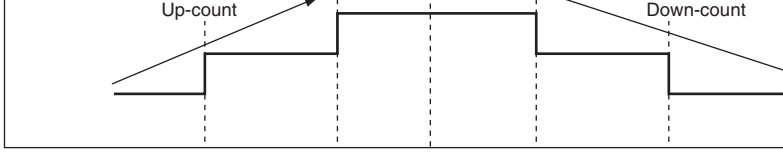


Figure 10.26 Example of Phase Counting Mode 2 Operation

Table 10.34 Up/Down-Count Conditions in Phase Counting Mode 2

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level	↑	Don't care
Low level	↓	Don't care
↑	Low level	Don't care
↓	High level	Up-count
High level	↓	Don't care
Low level	↑	Don't care
↑	High level	Don't care
↓	Low level	Down-count

[Legend]
↑ : Rising edge
↓ : Falling edge

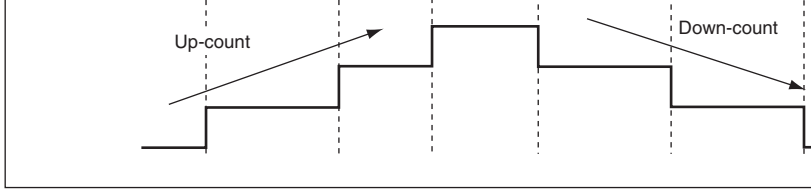


Figure 10.27 Example of Phase Counting Mode 3 Operation

Table 10.35 Up/Down-Count Conditions in Phase Counting Mode 3

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level	↑	Don't care
Low level	↓	Don't care
↑	Low level	Don't care
↓	High level	Up-count
High level	↓	Down-count
Low level	↑	Don't care
↑	High level	Don't care
↓	Low level	Don't care

[Legend]
↑ : Rising edge
↓ : Falling edge

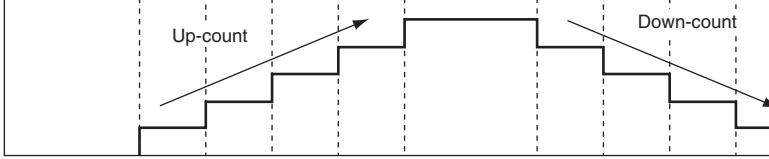


Figure 10.28 Example of Phase Counting Mode 4 Operation

Table 10.36 Up/Down-Count Conditions in Phase Counting Mode 4

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level	↑	Up-count
Low level	↓	
↑	Low level	Don't care
↓	High level	
High level	↓	Down-count
Low level	↑	
↑	High level	Don't care
↓	Low level	

[Legend]
↑ : Rising edge
↓ : Falling edge

in buffer mode. The channel 1 counter input clock is designated as the TGRB_0 input cap
source, and the pulse width of 2-phase encoder 4-multiplication pulses is detected.

TGRA_1 and TGRB_1 for channel 1 are designated for input capture, channel 0 TGRA_
TGRC_0 compare matches are selected as the input capture source, and the up/down-cou
values for the control cycles are stored.

This procedure enables accurate position/speed detection to be achieved.

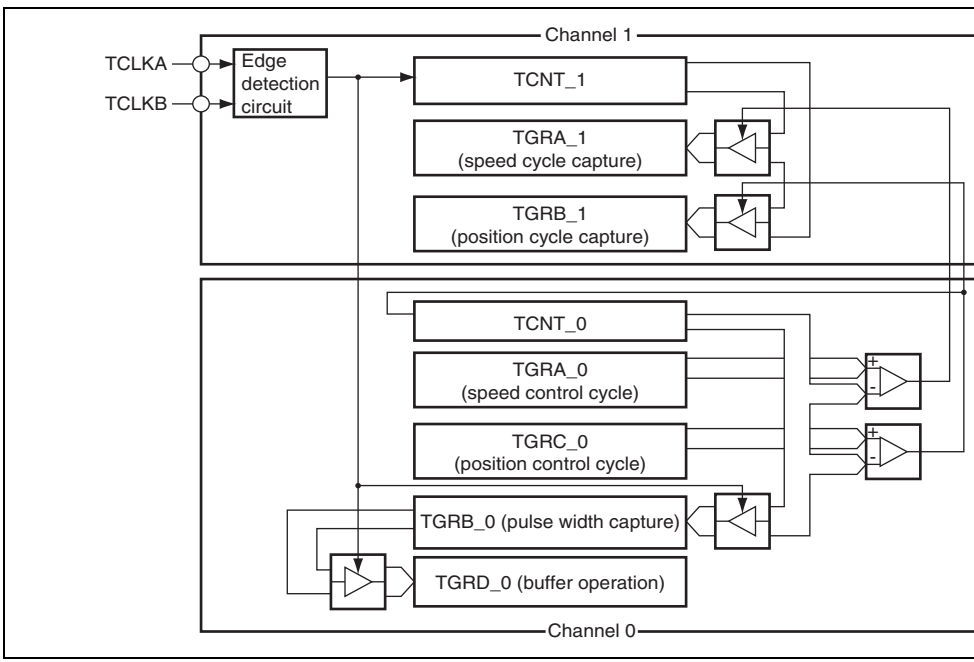


Figure 10.29 Phase Counting Mode Application Example

channel is fixed. For details, see section 5, Interrupt Controller.

Table 10.37 lists the TPU interrupt sources.

Table 10.37 TPU Interrupts

Channel	Name	Interrupt Source	Interrupt Flag	DTC Activation
0	TGI0A	TGRA_0 input capture/compare match	TGFA_0	O
	TGI0B	TGRB_0 input capture/compare match	TGFB_0	O
	TGI0C	TGRC_0 input capture/compare match	TGFC_0	O
	TGI0D	TGRD_0 input capture/compare match	TGFD_0	O
	TCI0V	TCNT_0 overflow	TCFV_0	—
1	TGI1A	TGRA_1 input capture/compare match	TGFA_1	O
	TGI1B	TGRB_1 input capture/compare match	TGFB_1	O
	TCI1V	TCNT_1 overflow	TCFV_1	—
	TCI1U	TCNT_1 underflow	TCFU_1	—
2	TGI2A	TGRA_2 input capture/compare match	TGFA_2	O
	TGI2B	TGRB_2 input capture/compare match	TGFB_2	O
	TCI2V	TCNT_2 overflow	TCFV_2	—
	TCI2U	TCNT_2 underflow	TCFU_2	—
3	TGI3A	TGRA_3 input capture/compare match	TGFA_3	O
	TGI3B	TGRB_3 input capture/compare match	TGFB_3	O
	TGI3C	TGRC_3 input capture/compare match	TGFC_3	O
	TGI3D	TGRD_3 input capture/compare match	TGFD_3	O
	TCI3V	TCNT_3 overflow	TCFV_3	—

TCI5U	TCNT_5 underflow	TCFU_5	—

[Legend]

O : Possible

— : Not possible

Note: This table shows the initial state immediately after a reset. The relative channel pri
levels can be changed by the interrupt controller.

(1) 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
by the occurrence of a TGR input capture/compare match on a channel. The interrupt req
cleared by clearing the TGF flag to 0. The TPU has 16 input capture/compare match inter
four each for channels 0 and 3, and two each for channels 1, 2, 4, and 5.

(2) Overflow Interrupt

An interrupt is requested if the TCIEV bit in TIER is set to 1 when the TCFV flag in TSR
1 by the occurrence of a TCNT overflow on a channel. The interrupt request is cleared by
the TCFV flag to 0. The TPU has six overflow interrupts, one for each channel.

(3) Underflow Interrupt

An interrupt is requested if the TCIEU bit in TIER is set to 1 when the TCFU flag in TSR
1 by the occurrence of a TCNT underflow on a channel. The interrupt request is cleared b
clearing the TCFU flag to 0. The TPU has four underflow interrupts, one each for channe
and 5.

The DMAC can be activated by the TGRA input capture/compare match interrupt for a
For details, see section 7, DMA Controller (DMAC).

A total of six TPU input capture/compare match interrupts can be used as DMAC activa
sources, one for each channel.

10.8 A/D Converter Activation

The TGRA input capture/compare match for each channel can activate the A/D converte

If the TTGE bit in TIER is set to 1 when the TGFA flag in TSR is set to 1 by the occurre
TGRA input capture/compare match on a particular channel, a request to start A/D conv
sent to the A/D converter. If the TPU conversion start trigger has been selected on the A
converter side at this time, A/D conversion is started.

In the TPU, a total of six TGRA input capture/compare match interrupts can be used as
converter conversion start sources, one for each channel.

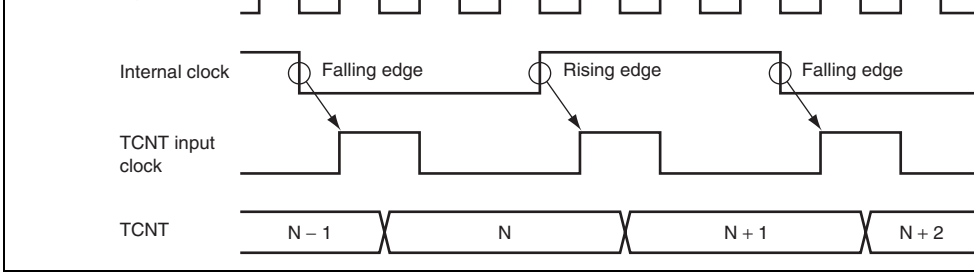


Figure 10.30 Count Timing in Internal Clock Operation

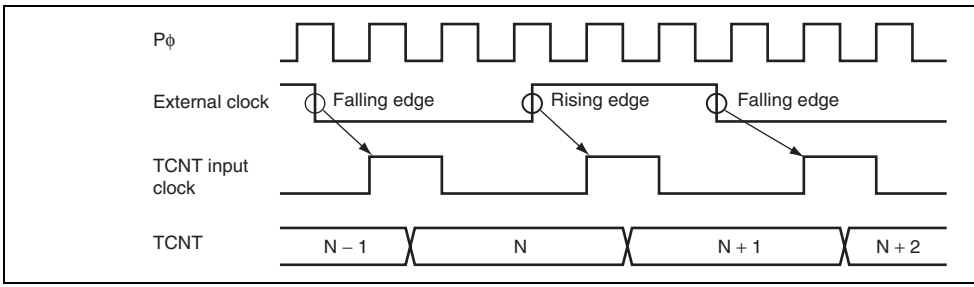


Figure 10.31 Count Timing in External Clock Operation

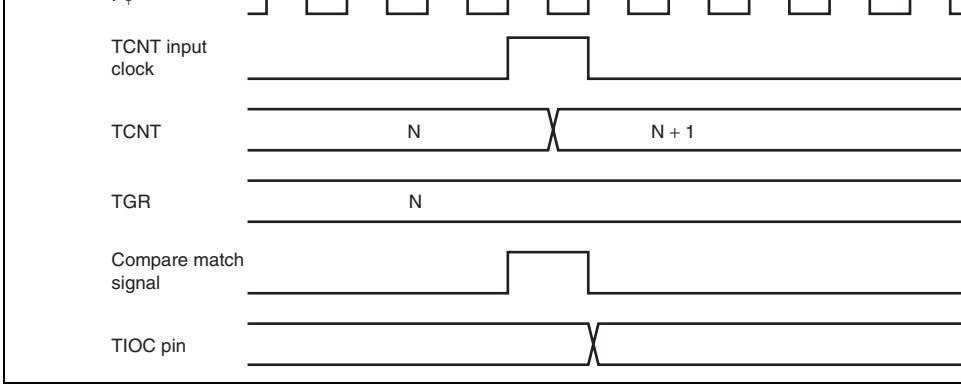


Figure 10.32 Output Compare Output Timing

Input Capture Signal Timing: Figure 10.33 shows input capture signal timing.

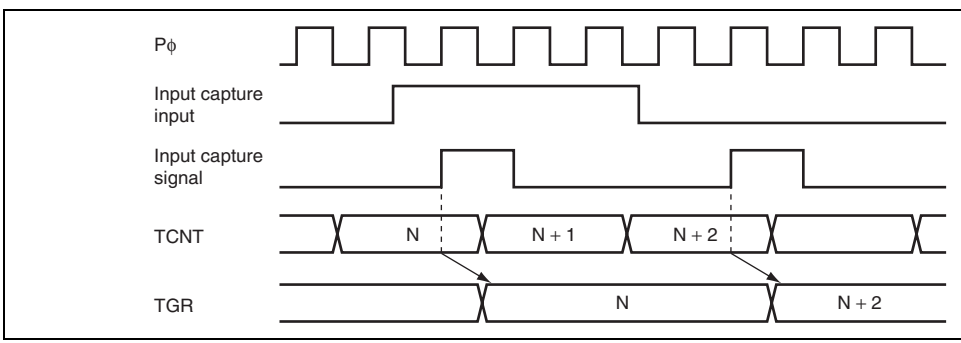


Figure 10.33 Input Capture Input Signal Timing

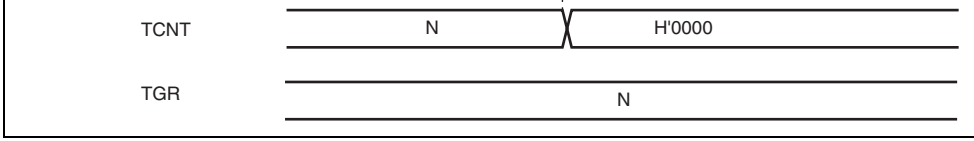


Figure 10.34 Counter Clear Timing (Compare Match)

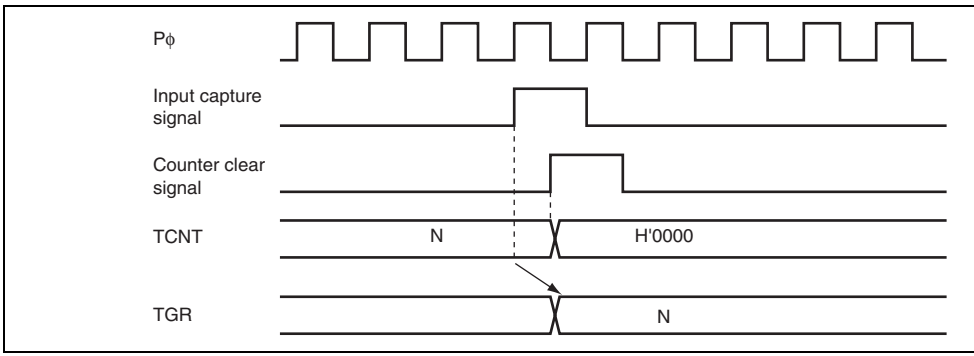


Figure 10.35 Counter Clear Timing (Input Capture)

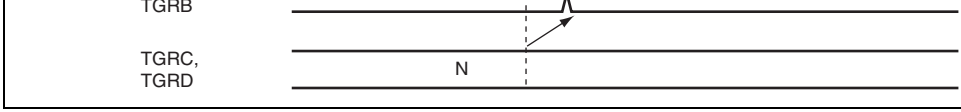


Figure 10.36 Buffer Operation Timing (Compare Match)

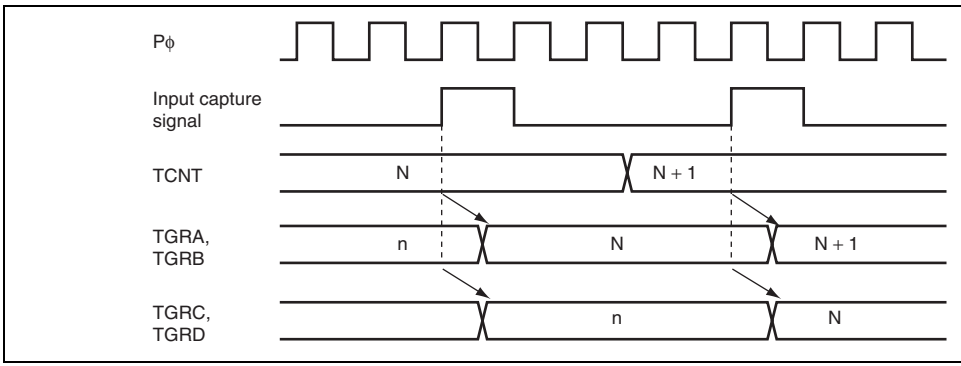


Figure 10.37 Buffer Operation Timing (Input Capture)

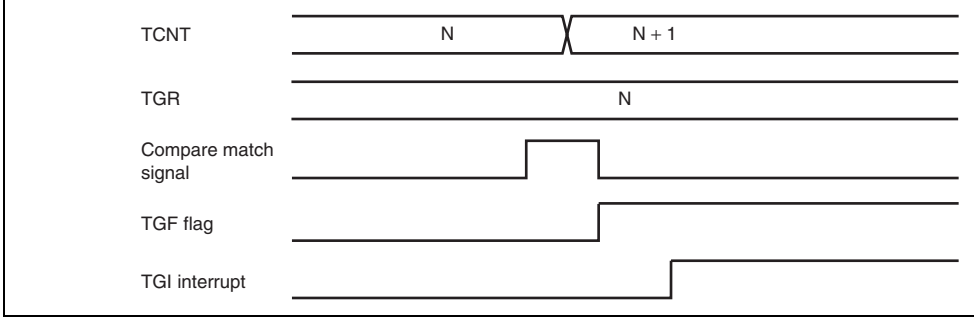


Figure 10.38 TGI Interrupt Timing (Compare Match)

(2) 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 occurr
the TGI interrupt request signal timing.

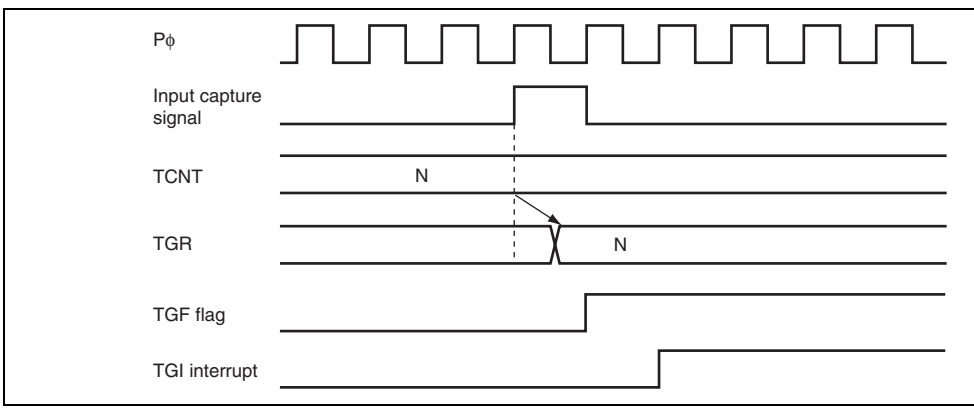


Figure 10.39 TGI Interrupt Timing (Input Capture)

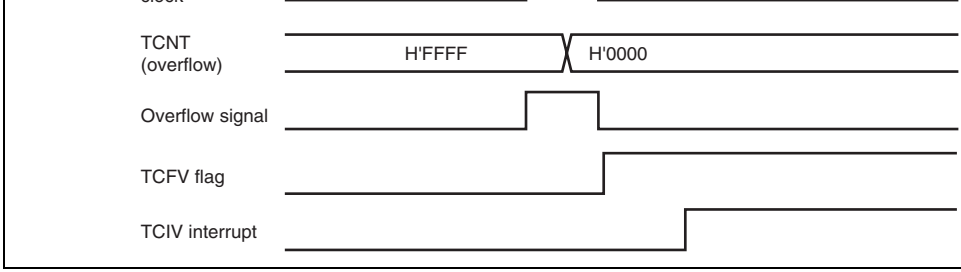


Figure 10.40 TCIV Interrupt Setting Timing

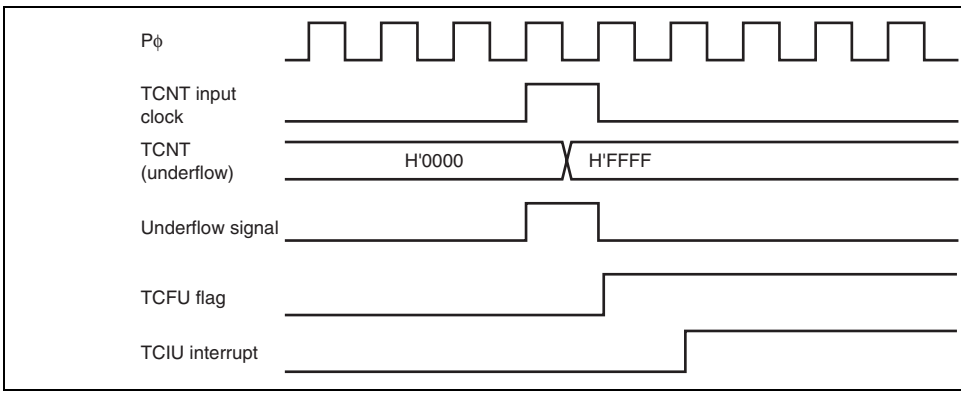


Figure 10.41 TCIU Interrupt Setting Timing

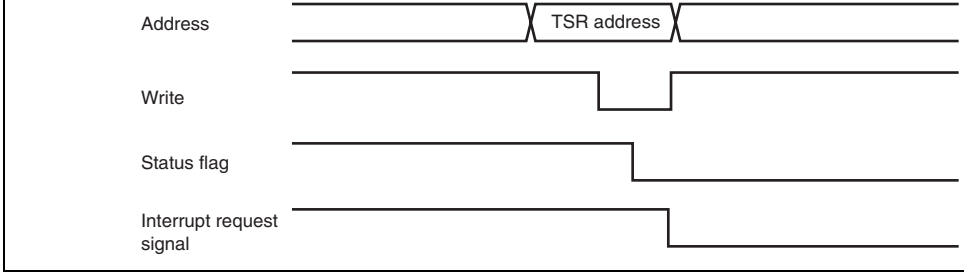


Figure 10.42 Timing for Status Flag Clearing by CPU

The status flag and interrupt request signal are cleared in synchronization with Pϕ after th
DMAC transfer has started, as shown in figure 10.43. If conflict occurs for clearing the st
and interrupt request signal due to activation of multiple DTC or DMAC transfers, it will
to five clock cycles (Pϕ) for clearing them, as shown in figure 10.44. The next transfer re
masked for a longer period of either a period until the current transfer ends or a period fo
clock cycles (Pϕ) from the beginning of the transfer. Note that in the DTC transfer, the st
may be cleared during outputting the destination address.

signal

Figure 10.43 Timing for Status Flag Clearing by DTC or DMAC Activation

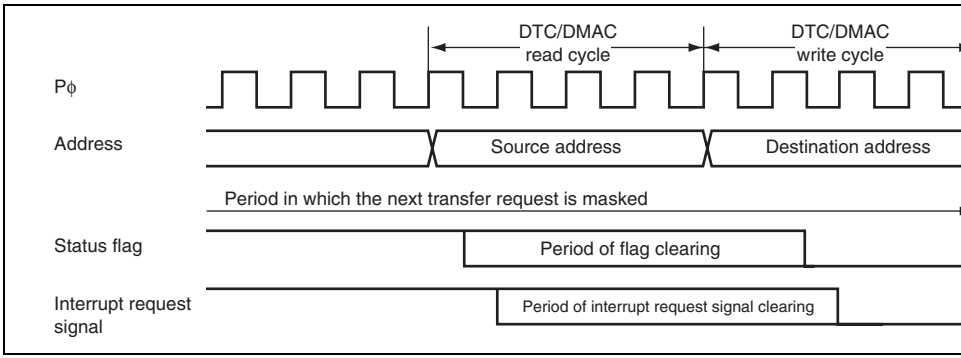


Figure 10.44 Timing for Status Flag Clearing by DTC or DMAC Activation

The input clock pulse width must be at least 1.5 states in the case of single-edge detectio
least 2.5 states in the case of both-edge detection. The TPU will not operate properly wit
narrower pulse width.

In phase counting mode, the phase difference and overlap between the two input clocks n
least 1.5 states, and the pulse width must be at least 2.5 states. Figure 10.45 shows the inp
conditions in phase counting mode.

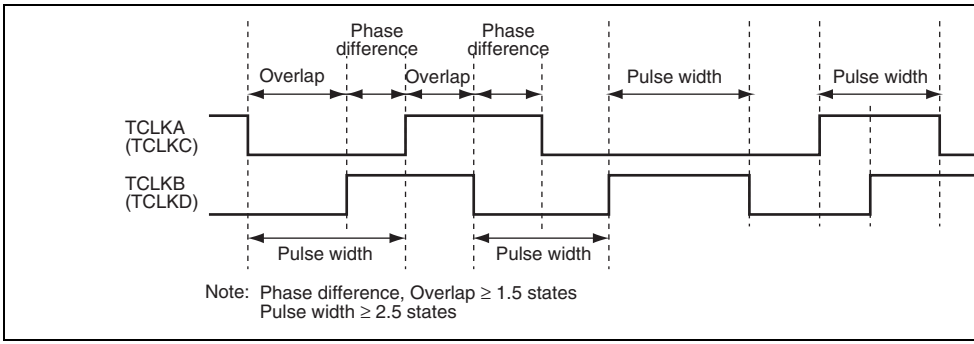


Figure 10.45 Phase Difference, Overlap, and Pulse Width in Phase Counting M

N: TCNT set value

10.10.4 Conflict between TCNT Write and Clear Operations

If the counter clearing signal is generated in the T2 state of a TCNT write cycle, TCNT
takes precedence and the TCNT write is not performed. Figure 10.46 shows the timing i
case.

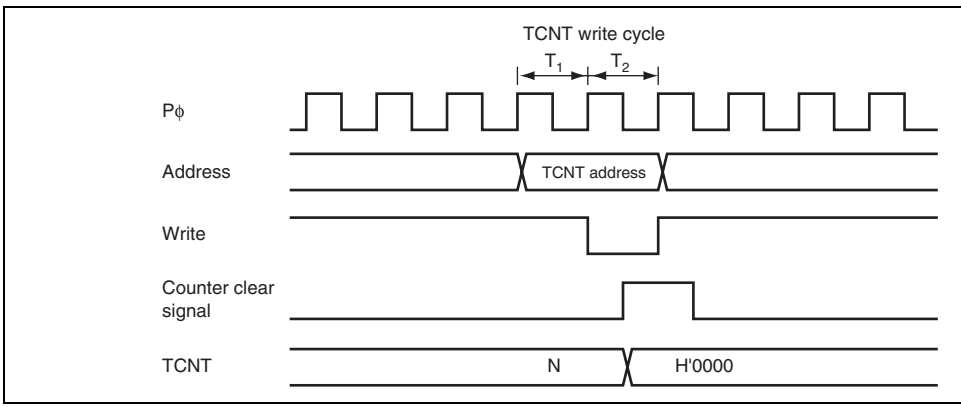


Figure 10.46 Conflict between TCNT Write and Clear Operations

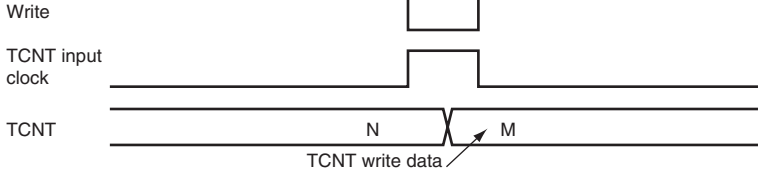


Figure 10.47 Conflict between TCNT Write and Increment Operations

10.10.6 Conflict between TGR Write and Compare Match

If a compare match occurs in the T2 state of a TGR write cycle, the TGR write takes prec
and the compare match signal is disabled. A compare match also does not occur when the
value as before is written.

Figure 10.48 shows the timing in this case.

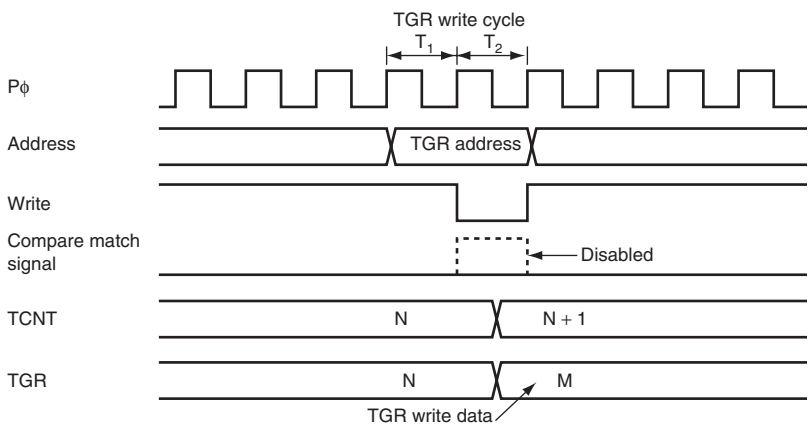


Figure 10.48 Conflict between TGR Write and Compare Match

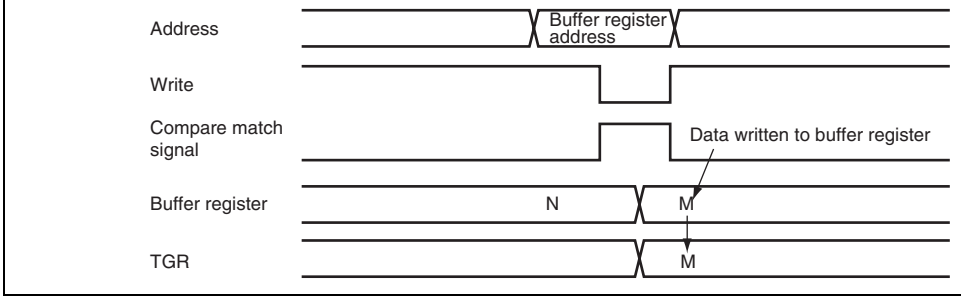


Figure 10.49 Conflict between Buffer Register Write and Compare Matc

10.10.8 Conflict between TGR Read and Input Capture

If the input capture signal is generated in the T1 state of a TGR read cycle, the data that will be the data after input capture transfer.

Figure 10.50 shows the timing in this case.

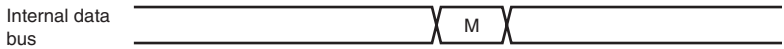


Figure 10.50 Conflict between TGR Read and Input Capture

10.10.9 Conflict between TGR Write and Input Capture

If the input capture signal is generated in the T2 state of a TGR write cycle, the input cap
operation takes precedence and the write to TGR is not performed.

Figure 10.51 shows the timing in this case.

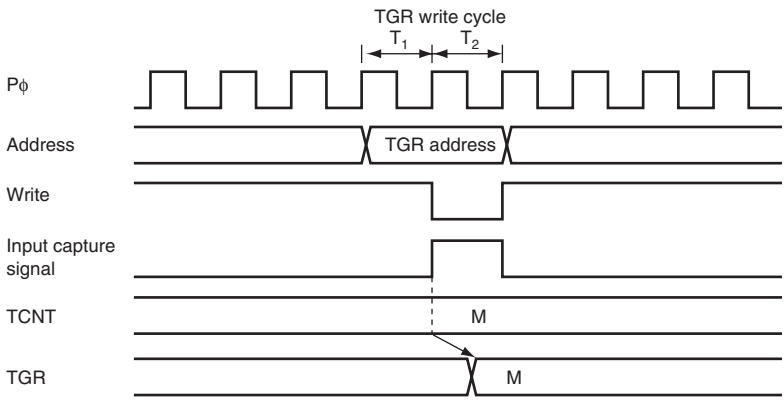


Figure 10.51 Conflict between TGR Write and Input Capture

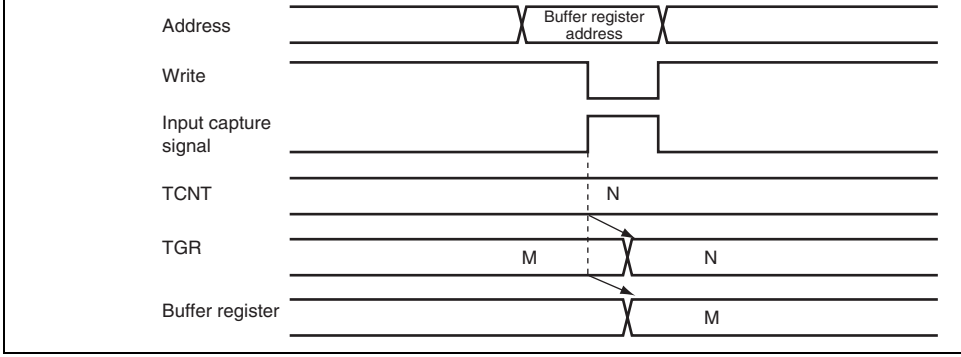


Figure 10.52 Conflict between Buffer Register Write and Input Capture

10.10.11 Conflict between Overflow/Underflow and Counter Clearing

If overflow/underflow and counter clearing occur simultaneously, the TCFV/TCFU flag
not set and TCNT clearing takes precedence.

Figure 10.53 shows the operation timing when a TGR compare match is specified as the
source, and H'FFFF is set in TGR.

Figure 10.53 Conflict between Overflow and Counter Clearing

10.10.12 Conflict between TCNT Write and Overflow/Underflow

If an overflow/underflow occurs due to increment/decrement in the T2 state of a TCNT w
cycle, the TCNT write takes precedence and the TCFV/TCFU flag in TSR is not set.

Figure 10.54 shows the operation timing when there is conflict between TCNT write and
overflow.

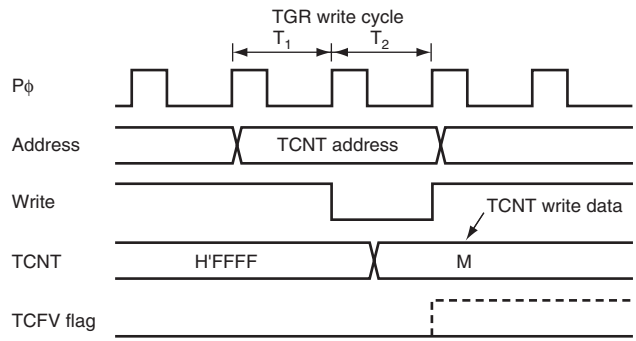


Figure 10.54 Conflict between TCNT Write and Overflow

clear the CPU interrupt source or the DMAC or DTC activation source. Interrupts shoul
be disabled before entering the module stop state.

- Four output groups
- Selectable output trigger signals
- Non-overlapping mode
- Can operate together with the data transfer controller (DTC) and DMA controller (D
- Inverted output can be set
- Module stop state specifiable

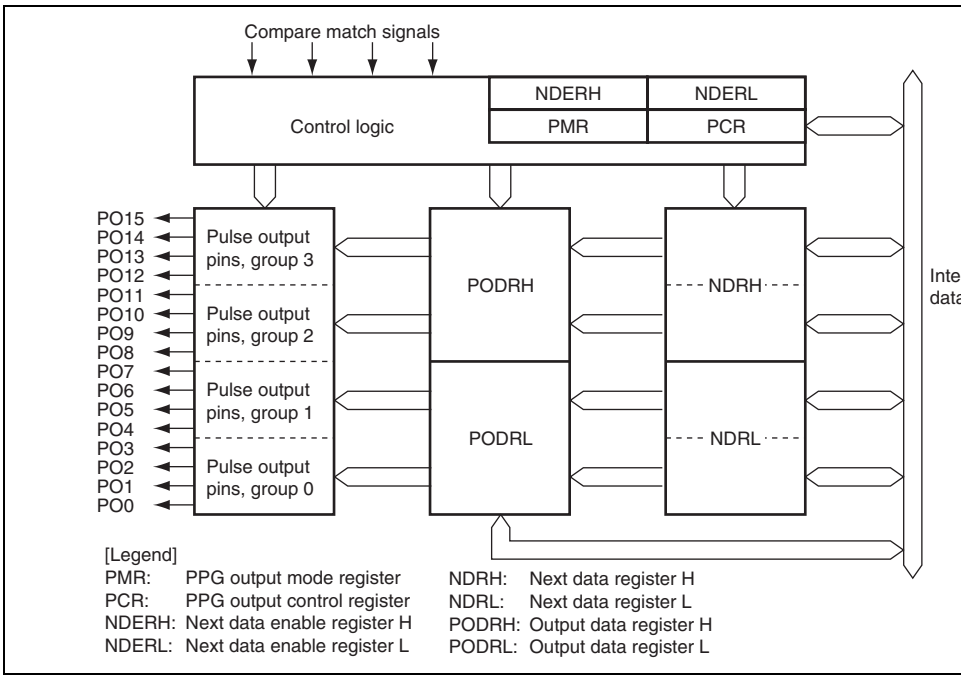


Figure 11.1 Block Diagram of PPG

PO12	Output	
PO11	Output	Group 2 pulse output
PO10	Output	
PO9	Output	
PO8	Output	
PO7	Output	Group 1 pulse output
PO6	Output	
PO5	Output	
PO4	Output	
PO3	Output	Group 0 pulse output
PO2	Output	
PO1	Output	
PO0	Output	

- PPG output control register (PCR)
- PPG output mode register (PMR)

11.3.1 Next Data Enable Registers H, L (NDERH, NDERL)

NDERH and NDERL enable/disable pulse output on a bit-by-bit basis.

- NDERH

Bit	7	6	5	4	3	2	1	
Bit Name	NDER15	NDER14	NDER13	NDER12	NDER11	NDER10	NDER9	
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

- NDERL

Bit	7	6	5	4	3	2	1	
Bit Name	NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1	
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

0	NDER8	0	R/W	

- NDERL

Bit	Bit Name	Initial Value	R/W	Description
7	NDER7	0	R/W	Next Data Enable 7 to 0
6	NDER6	0	R/W	When a bit is set to 1, the value in the correspon
5	NDER5	0	R/W	NDRL bit is transferred to the PODRL bit by the
4	NDER4	0	R/W	output trigger. Values are not transferred from N
3	NDER3	0	R/W	PODRL for cleared bits.
2	NDER2	0	R/W	
1	NDER1	0	R/W	
0	NDER0	0	R/W	

- PODRL

Bit	7	6	5	4	3	2	1
Bit Name	POD7	POD6	POD5	POD4	POD3	POD2	POD1
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- PODRH

Bit	Bit Name	Initial Value	R/W	Description
7	POD15	0	R/W	Output Data Register 15 to 8
6	POD14	0	R/W	For bits which have been set to pulse output by
5	POD13	0	R/W	the output trigger transfers NDRH values to this
4	POD12	0	R/W	during PPG operation. While NDERH is set to 1
3	POD11	0	R/W	cannot write to this register. While NDERH is cl
2	POD10	0	R/W	initial output value of the pulse can be set.
1	POD9	0	R/W	
0	POD8	0	R/W	

0	POD0	0	R/W

11.3.3 Next Data Registers H, L (NDRH, NDRL)

NDRH and NDRL store the next data for pulse output. The NDR addresses differ depend
whether pulse output groups have the same output trigger or different output triggers.

- NDRH

Bit	7	6	5	4	3	2	1
Bit Name	NDR15	NDR14	NDR13	NDR12	NDR11	NDR10	NDR9
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- NDRL

Bit	7	6	5	4	3	2	1
Bit Name	NDR7	NDR6	NDR5	NDR4	NDR3	NDR2	NDR1
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

2	NDR10	0	R/W	
1	NDR9	0	R/W	
0	NDR8	0	R/W	

If pulse output groups 2 and 3 have different output triggers, the upper four bits and
bits are mapped to different addresses as shown below.

Bit	Bit Name	Initial Value	R/W	Description
7	NDR15	0	R/W	Next Data Register 15 to 12
6	NDR14	0	R/W	The register contents are transferred to the corresponding PODRH bits by the output trigge with PCR.
5	NDR13	0	R/W	
4	NDR12	0	R/W	
3 to 0	—	All 1	—	Reserved These bits are always read as 1 and cannot be

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	—	Reserved These bits are always read as 1 and cannot be
3	NDR11	0	R/W	Next Data Register 11 to 8
2	NDR10	0	R/W	The register contents are transferred to the corresponding PODRH bits by the output trigge with PCR.
1	NDR9	0	R/W	
0	NDR8	0	R/W	

2	NDR2	0	R/W	
1	NDR1	0	R/W	
0	NDR0	0	R/W	

If pulse output groups 0 and 1 have different output triggers, the upper four bits and lo
bits are mapped to different addresses as shown below.

Bit	Bit Name	Initial Value	R/W	Description
7	NDR7	0	R/W	Next Data Register 7 to 4
6	NDR6	0	R/W	The register contents are transferred to the corresponding PODRL bits by the output trigger with PCR.
5	NDR5	0	R/W	
4	NDR4	0	R/W	
3 to 0	—	All 1	—	Reserved These bits are always read as 1 and cannot be r

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	—	Reserved These bits are always read as 1 and cannot be r
3	NDR3	0	R/W	Next Data Register 3 to 0
2	NDR2	0	R/W	The register contents are transferred to the corresponding PODRL bits by the output trigger with PCR.
1	NDR1	0	R/W	
0	NDR0	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
7	G3CMS1	1	R/W	Group 3 Compare Match Select 1 and 0
6	G3CMS0	1	R/W	These bits select output trigger of pulse output 00: Compare match in TPU channel 0 01: Compare match in TPU channel 1 10: Compare match in TPU channel 2 11: Compare match in TPU channel 3
5	G2CMS1	1	R/W	Group 2 Compare Match Select 1 and 0
4	G2CMS0	1	R/W	These bits select output trigger of pulse output 00: Compare match in TPU channel 0 01: Compare match in TPU channel 1 10: Compare match in TPU channel 2 11: Compare match in TPU channel 3
3	G1CMS1	1	R/W	Group 1 Compare Match Select 1 and 0
2	G1CMS0	1	R/W	These bits select output trigger of pulse output 00: Compare match in TPU channel 0 01: Compare match in TPU channel 1 10: Compare match in TPU channel 2 11: Compare match in TPU channel 3
1	G0CMS1	1	R/W	Group 0 Compare Match Select 1 and 0
0	G0CMS0	1	R/W	These bits select output trigger of pulse output 00: Compare match in TPU channel 0 01: Compare match in TPU channel 1 10: Compare match in TPU channel 2 11: Compare match in TPU channel 3

Initial Value	1	1	1	1	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	G3INV	1	R/W	Group 3 Inversion Selects direct output or inverted output for pulse group 3. 0: Inverted output 1: Direct output
6	G2INV	1	R/W	Group 2 Inversion Selects direct output or inverted output for pulse group 2. 0: Inverted output 1: Direct output
5	G1INV	1	R/W	Group 1 Inversion Selects direct output or inverted output for pulse group 1. 0: Inverted output 1: Direct output
4	G0INV	1	R/W	Group 0 Inversion Selects direct output or inverted output for pulse group 0. 0: Inverted output 1: Direct output

				Selects normal or non-overlapping operation fo output group 2. 0: Normal operation (output values updated at c match A in the selected TPU channel) 1: Non-overlapping operation (output values up compare match A or B in the selected TPU c
1	G1NOV	0	R/W	Group 1 Non-Overlap Selects normal or non-overlapping operation fo output group 1. 0: Normal operation (output values updated at c match A in the selected TPU channel) 1: Non-overlapping operation (output values up compare match A or B in the selected TPU c
0	G0NOV	0	R/W	Group 0 Non-Overlap Selects normal or non-overlapping operation fo output group 0. 0: Normal operation (output values updated at c match A in the selected TPU channel) 1: Non-overlapping operation (output values up compare match A or B in the selected TPU c

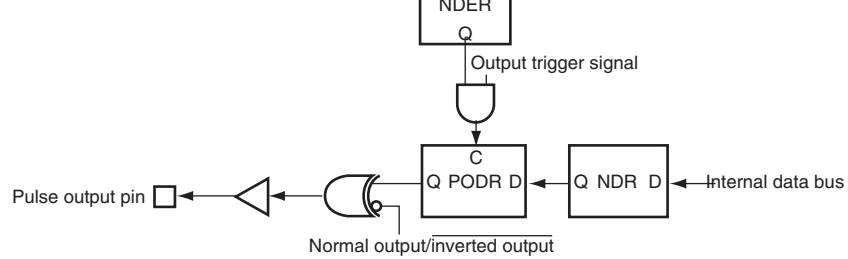


Figure 11.2 Schematic Diagram of PPG

11.4.1 Output Timing

If pulse output is enabled, the NDR contents are transferred to PODR and output when th specified compare match event occurs. Figure 11.3 shows the timing of these operations f case of normal output in groups 2 and 3, triggered by compare match A.

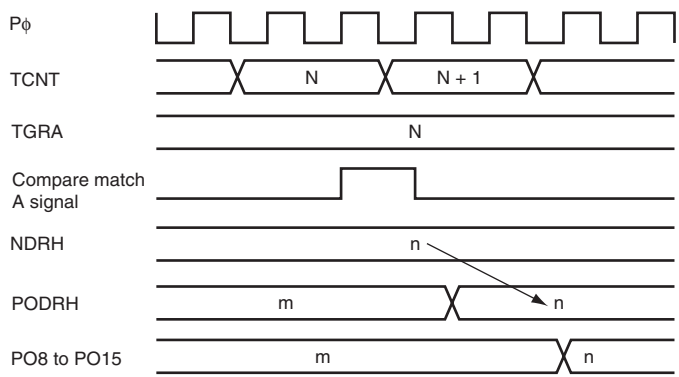
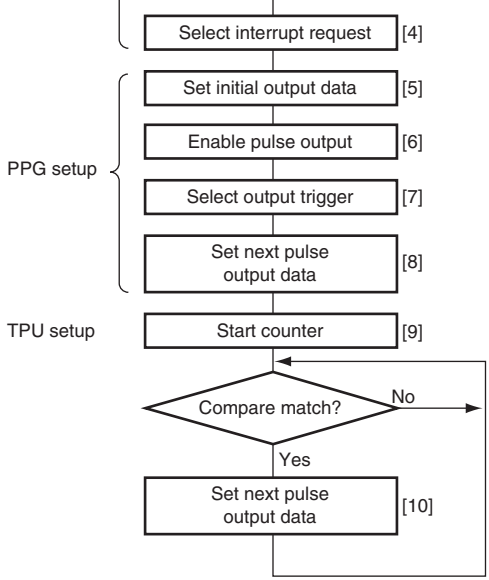


Figure 11.3 Timing of Transfer and Output of NDR Contents (Example)



[4] Enable the TGIA interrupt in TIER. Th
DTC or DMAC can also be set up to
transfer data to NDR.

[5] Set the initial output values in PODR.

[6] Set the bits in NDER for the pins to be
used for pulse output to 1.

[7] Select the TPU compare match event
be used as the output trigger in PCR.

[8] Set the next pulse output values in ND

[9] Set the CST bit in TSTR to 1 to start t
TCNT counter.

[10] At each TGIA interrupt, set the next
output values in NDR.

Figure 11.4 Setup Procedure for Normal Pulse Output (Example)

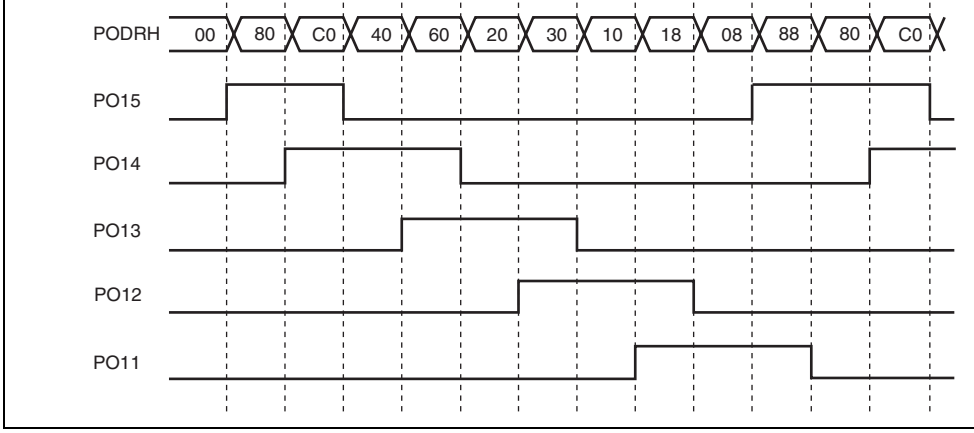


Figure 11.5 Normal Pulse Output Example (5-Phase Pulse Output)

1. Set up TGRA in TPU which is used as the output trigger to be an output compare regi
   a cycle in TGRA so the counter will be cleared by compare match A. Set the TGIEA
   TIER to 1 to enable the compare match/input capture A (TGIA) interrupt.
2. Write H'F8 to NDERH, and set bits G3CMS1, G3CMS0, G2CMS1, and G2CMS0 in
   select compare match in the TPU channel set up in the previous step to be the output
   Write output data H'80 in NDRH.
3. The timer counter in the TPU channel starts. When compare match A occurs, the ND
   contents are transferred to PODRH and output. The TGIA interrupt handling routine
   next output data (H'C0) in NDRH.
4. 5-phase pulse output (one or two phases active at a time) can be obtained subsequentl
   writing H'40, H'60, H'20, H'30, H'10, H'18, H'08, H'88... at successive TGIA interrupt
   If the DTC or DMAC is set for activation by the TGIA interrupt, pulse output can be
   without imposing a load on the CPU.

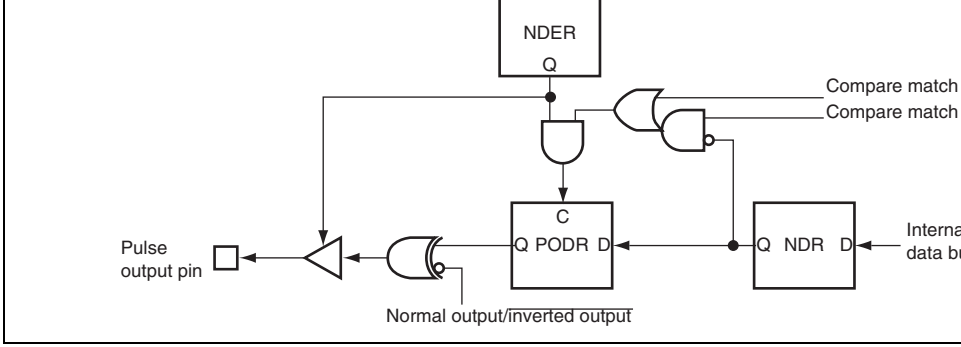


Figure 11.6 Non-Overlapping Pulse Output

Therefore, 0 data can be transferred ahead of 1 data by making compare match B occur
compare match A.

The NDR contents should not be altered during the interval from compare match B to co
match A (the non-overlapping margin).

This can be accomplished by having the TGIA interrupt handling routine write the next
NDR, or by having the TGIA interrupt activate the DTC or DMAC. Note, however, that
data must be written before the next compare match B occurs.

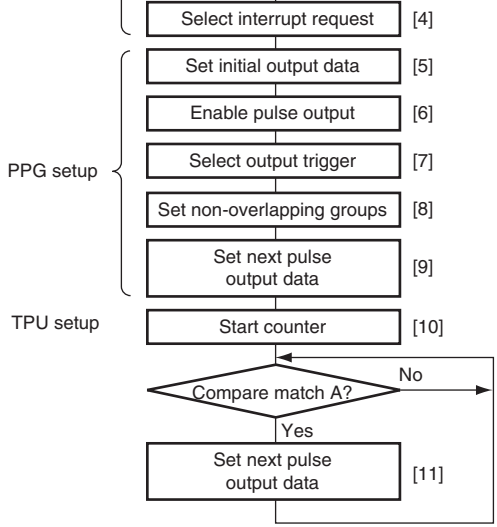
Do not write to NDR here

here

Do not write to NDR here

here

Figure 11.7 Non-Overlapping Operation and NDR Write Timing



counter clear source with bits CCLR1 and CCLR0.

[4] Enable the TGIA interrupt in TIER. T DTC or DMAC can also be set up to transfer data to NDR.

[5] Set the initial output values in PODR.

[6] Set the bits in NDER for the pins to b used for pulse output to 1.

[7] Select the TPU compare match even be used as the pulse output trigger in PCR.

[8] In PMR, select the groups that will operate in non-overlapping mode.

[9] Set the next pulse output values in N

[10] Set the CST bit in TSTR to 1 to start TCNT counter.

[11] At each TGIA interrupt, set the next output values in NDR.

Figure 11.8 Setup Procedure for Non-Overlapping Pulse Output (Exampl

11.4.6 Example of Non-Overlapping Pulse Output (Example of 4-Phase Comple Non-Overlapping Pulse Output)

Figure 11.9 shows an example in which pulse output is used for 4-phase complementary overlapping pulse output.

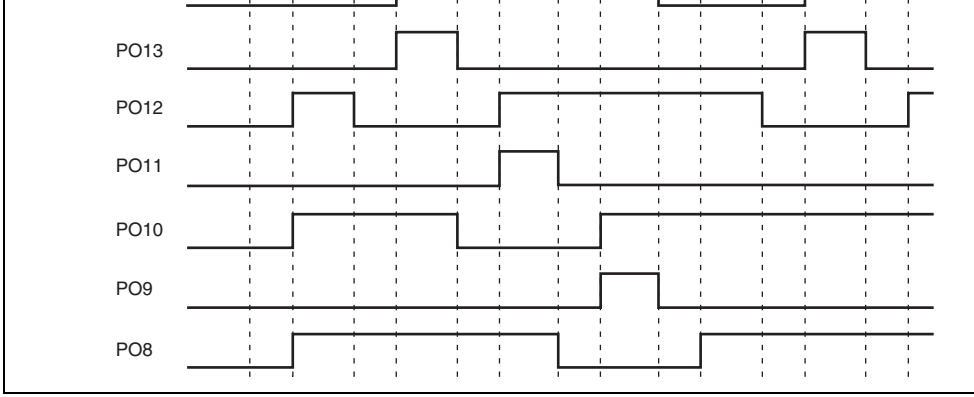


Figure 11.9 Non-Overlapping Pulse Output Example (4-Phase Complementa

1. Set up the TPU channel to be used as the output trigger channel so that TGRA and TG output compare registers. Set the cycle in TGRB and the non-overlapping margin in T and set the counter to be cleared by compare match B. Set the TGIEA bit in TIER to 1 enable the TGIA interrupt.
2. Write H'FF to NDERH, and set bits G3CMS1, G3CMS0, G2CMS1, and G2CMS0 in select compare match in the TPU channel set up in the previous step to be the output t

   Set bits G3NOV and G2NOV in PMR to 1 to select non-overlapping pulse output.

   Write output data H'95 to NDRH.
3. The timer counter in the TPU channel starts. When a compare match with TGRB occu outputs change from 1 to 0. When a compare match with TGRA occurs, outputs chang to 1 (the change from 0 to 1 is delayed by the value set in TGRA).

   The TGIA interrupt handling routine writes the next output data (H'65) to NDRH.

the settings of figure 11.9.

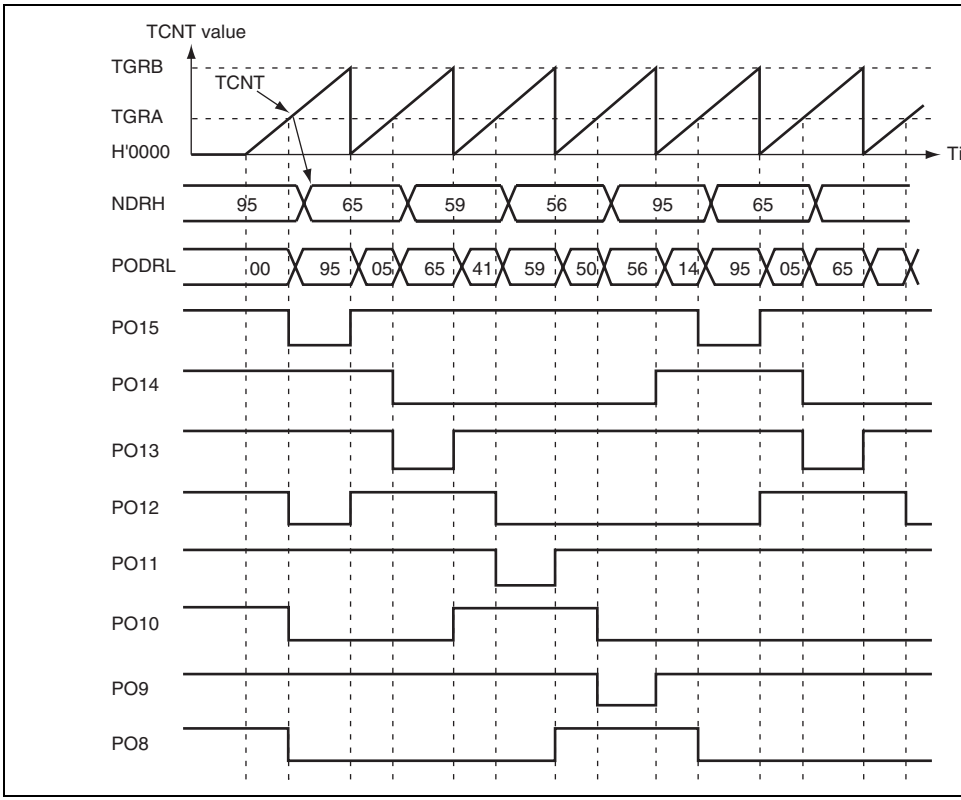


Figure 11.10 Inverted Pulse Output (Example)

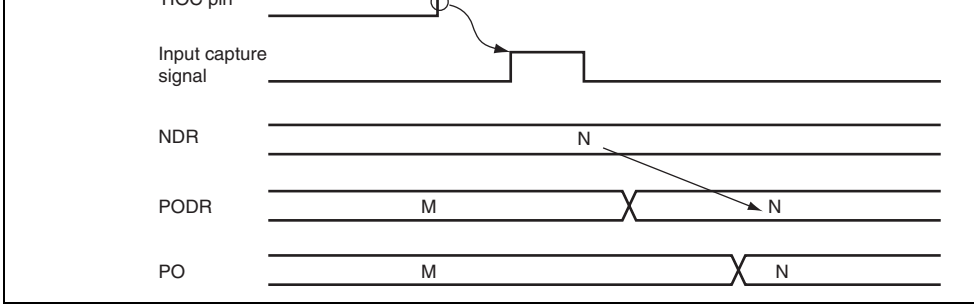


Figure 11.11 Pulse Output Triggered by Input Capture (Example)

11.5 Usage Notes

11.5.1 Module Stop State Setting

PPG operation can be disabled or enabled using the module stop control register. The init
is for PPG operation to be halted. Register access is enabled by clearing the module stop
details, refer to section 20, Power-Down Modes.

11.5.2 Operation of Pulse Output Pins

Pins PO0 to PO15 are also used for other peripheral functions such as the TPU. When ou
another peripheral function is enabled, the corresponding pins cannot be used for pulse ou
Note, however, that data transfer from NDR bits to PODR bits takes place, regardless of t
of the pins.

Pin functions should be changed only under conditions in which the output trigger event
occur.

12.1 Features

- Selection of seven clock sources

  The counters can be driven by one of six internal clock signals (Pϕ/2, Pϕ/8, Pϕ/32, P
  Pϕ/1024, or Pϕ/8192) or an external clock input.
- Selection of three ways to clear the counters

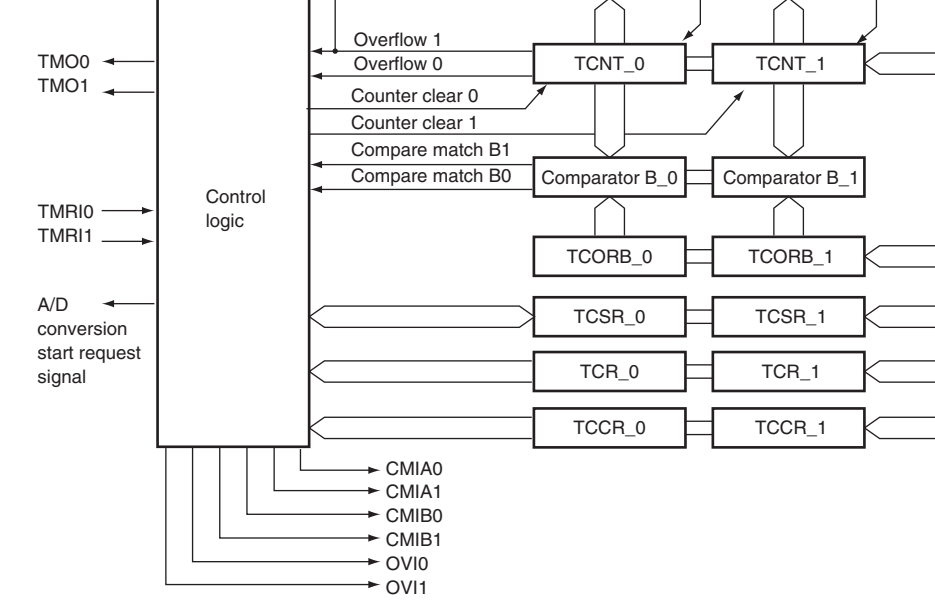
  The counters can be cleared on compare match A or B, or by an external reset signal
- Timer output control by a combination of two compare match signals

  The timer output signal in each channel is controlled by a combination of two indepe
  compare match signals, enabling the timer to output pulses with a desired duty cycle
  output.
- Cascading of two channels (TMR_0 and TMR_1)

  Operation as a 16-bit timer is possible, using TMR_0 for the upper 8 bits and TMR_
  lower 8 bits (16-bit count mode).

  TMR_1 can be used to count TMR_0 compare matches (compare match count mode
- Three interrupt sources

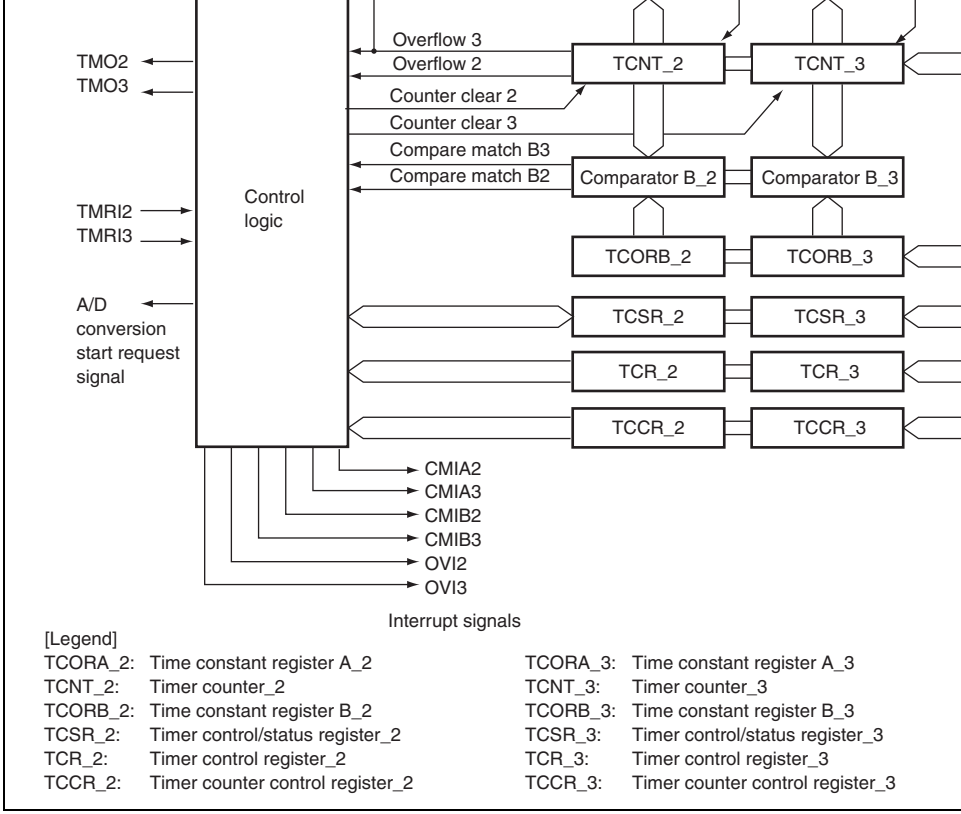
  Compare match A, compare match B, and overflow interrupts can be requested indep
- Generation of trigger to start A/D converter conversion
- Module stop state specifiable



[Legend]

TCORA_0:	Time constant register A_0	TCORA_1:	Time constant register A_1
TCNT_0:	Timer counter_0	TCNT_1:	Timer counter_1
TCORB_0:	Time constant register B_0	TCORB_1:	Time constant register B_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
TCCR_0:	Timer counter control register_0	TCCR_1:	Timer counter control register_1

Figure 12.1 Block Diagram of 8-Bit Timer Module (Unit 0)



[Legend]

TCORA_2:	Time constant register A_2	TCORA_3:	Time constant register A_3
TCNT_2:	Timer counter_2	TCNT_3:	Timer counter_3
TCORB_2:	Time constant register B_2	TCORB_3:	Time constant register B_3
TCSR_2:	Timer control/status register_2	TCSR_3:	Timer control/status register_3
TCR_2:	Timer control register_2	TCR_3:	Timer control register_3
TCCR_2:	Timer counter control register_2	TCCR_3:	Timer counter control register_3

Figure 12.2 Block Diagram of 8-Bit Timer Module (Unit 1)

	1	Timer output pin	TMO1	Output	Outputs compare match
		Timer clock input pin	TMCI1	Input	Inputs external clock for co
		Timer reset input pin	TMRI1	Input	Inputs external reset to cou
1	2	Timer output pin	TMO2	Output	Outputs compare match
		Timer clock input pin	TMCI2	Input	Inputs external clock for co
		Timer reset input pin	TMRI2	Input	Inputs external reset to cou
	3	Timer output pin	TMO3	Output	Outputs compare match
		Timer clock input pin	TMCI3	Input	Inputs external clock for co
		Timer reset input pin	TMRI3	Input	Inputs external reset to cou

12.3 Register Descriptions

The TMR has the following registers.

Unit 0:

- Channel 0
  - Timer counter_0 (TCNT_0)
  - Time constant register A_0 (TCORA_0)
  - Time constant register B_0 (TCORB_0)
  - Timer control register_0 (TCR_0)
  - Timer counter control register_0 (TCCR_0)
  - Timer control/status register_0 (TCSR_0)

- Channel 2
  - Timer counter_2 (TCNT_2)
  - Time constant register A_2 (TCORA_2)
  - Time constant register B_2 (TCORB_2)
  - Timer control register_2 (TCR_2)
  - Timer counter control register_2 (TCCR_2)
  - Timer control/status register_2 (TCSR_2)

- Channel 3
  - Timer counter_3 (TCNT_3)
  - Time constant register A_3 (TCORA_3)
  - Time constant register B_3 (TCORB_3)
  - Timer control register_3 (TCR_3)
  - Timer counter control register_3 (TCCR_3)
  - Timer control/status register_3 (TCSR_3)

Bit Name															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/

12.3.2 Time Constant Register A (TCORA)

TCORA is an 8-bit readable/writable register. TCORA_0 and TCORA_1 comprise a sing register so they can be accessed together by a word transfer instruction. The value in TCO continually compared with the value in TCNT. When a match is detected, the correspond CMFA flag in TCSR is set to 1. Note however that comparison is disabled during the T2 TCORA write cycle. The timer output from the TMO pin can be freely controlled by this match signal (compare match A) and the settings of bits OS1 and OS0 in TCSR. TCORA initialized to H'FF.

	TCORA_0								TCORA_1						
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1
Bit Name															
Initial Value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/

Bit Name															
Initial Value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

12.3.4 Timer Control Register (TCR)

TCR selects the TCNT clock source and the condition for clearing TCNT, and enables/d
interrupt requests.

Bit	7	6	5	4	3	2	1	
Bit Name	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Bit	Bit Name	Initial Value	R/W	Description
7	CMIEB	0	R/W	Compare Match Interrupt Enable B Selects whether CMFB interrupt requests (CM enabled or disabled when the CMFB flag in T to 1. 0: CMFB interrupt requests (CMIB) are disabl 1: CMFB interrupt requests (CMIB) are enabl

				enabled or disabled when the OVF flag in TCS to 1. 0: OVF interrupt requests (OVI) are disabled 1: OVF interrupt requests (OVI) are enabled
4	CCLR1	0	R/W	Counter Clear 1 and 0*
3	CCLR0	0	R/W	These bits select the method by which TCNT is 00: Clearing is disabled 01: Cleared by compare match A 10: Cleared by compare match B 11: Cleared at rising edge (TMRIS in TCCR is to 0) of the external reset input or when the reset input is high (TMRIS in TCCR is set t
2	CKS2	0	R/W	Clock Select 2 to 0*
1	CKS1	0	R/W	These bits select the clock input to TCNT and condition. See table 12.2.
0	CKS0	0	R/W	

Note: * To use an external reset or external clock, the DDR and ICR bits in the corresp pin should be set to 0 and 1, respectively. For details, see section 9, I/O Ports.

Bit	Bit Name	Value	R/W	Description
7 to 4	—	0	R/W	Reserved These bits are always read as 0. The write va always be 0.
3	TMRIS	0	R/W	Timer Reset Input Select Selects an external reset input when the CCL CCLR0 bits in TCR are B'11. 0: Cleared at rising edge of the external reset 1: Cleared when the external reset is high
2	—	0	R/W	Reserved This bit is always read as 0. The write value s always be 0
1	ICKS1	0	R/W	Internal Clock Select 1 and 0
0	ICKS0	0	R/W	These bits in combination with bits CKS2 to C TCR select the internal clock. See table 12.2.

	0	1	0	0	0	Uses internal clock. Counts at rising edge of P
				0	1	Uses internal clock. Counts at rising edge of P
				1	0	Uses internal clock. Counts at falling edge of F
				1	1	Uses internal clock. Counts at falling edge of F
	0	1	1	0	0	Uses internal clock. Counts at rising edge of P
				0	1	Uses internal clock. Counts at rising edge of P
				1	0	Uses internal clock. Counts at falling edge of F
				1	1	Uses internal clock. Counts at falling edge of F
	1	0	0	—	—	Counts at TCNT_1 overflow signal*[1].
TMR_1	0	0	0	—	—	Clock input prohibited.
	0	0	1	0	0	Uses internal clock. Counts at rising edge of P
				0	1	Uses internal clock. Counts at rising edge of P
				1	0	Uses internal clock. Counts at falling edge of F
				1	1	Uses internal clock. Counts at falling edge of F
	0	1	0	0	0	Uses internal clock. Counts at rising edge of P
				0	1	Uses internal clock. Counts at rising edge of P
				1	0	Uses internal clock. Counts at falling edge of F
				1	1	Uses internal clock. Counts at falling edge of F
	0	1	1	0	0	Uses internal clock. Counts at rising edge of P
				0	1	Uses internal clock. Counts at rising edge of P
				1	0	Uses internal clock. Counts at falling edge of F
				1	1	Uses internal clock. Counts at falling edge of F
	1	0	0	—	—	Counts at TCNT_0 compare match A*[1].

2. To use the external clock, the DDR and ICR bits in the corresponding pin shou
to 0 and 1, respectively. For details, see section 9, I/O Ports.

12.3.6 Timer Control/Status Register (TCSR)

TCSR displays status flags, and controls compare match output.

• TCSR_0

Bit	7	6	5	4	3	2	1
Bit Name	CMFB	CMFA	OVF	ADTE	OS3	OS2	OS1
Initial Value	0	0	0	0	0	0	0
R/W	R/(W)*	R/(W)*	R/(W)*	R/W	R/W	R/W	R/W

• TCSR_1

Bit	7	6	5	4	3	2	1
Bit Name	CMFB	CMFA	OVF	—	OS3	OS2	OS1
Initial Value	0	0	0	1	0	0	0
R/W	R/(W)*	R/(W)*	R/(W)*	R	R/W	R/W	R/W

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

				sure to read the flag after writing 0 to it.) • When the DTC is activated by a CMIB inter while the DISEL bit in MRB of the DTC is 0
6	CMFA	0	R/(W)*[1]	Compare Match Flag A [Setting condition] • When TCNT matches TCORA [Clearing conditions] • When writing 0 after reading CMFA = 1 (When the CPU is used to clear this flag by while the corresponding interrupt is enablec sure to read the flag after writing 0 to it.) • When the DTC is activated by a CMIA inter while the DISEL bit in MRB in the DTC is 0
5	OVF	0	R/(W)*[1]	Timer Overflow Flag [Setting condition] • When TCNT overflows from H'FF to H'00 [Clearing condition] • When writing 0 after reading OVF = 1 (When the CPU is used to clear this flag by while the corresponding interrupt is enablec sure to read the flag after writing 0 to it.)

2	OS2	0	R/W	These bits select a method of TMO pin outpu compare match B of TCORB and TCNT occu 00: No change when compare match B occur 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 (toggle output)
1	OS1	0	R/W	Output Select 1 and 0*[2]
0	OS0	0	R/W	These bits select a method of TMO pin outpu compare match A of TCORA and TCNT occu 00: No change when compare match A occur 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 (toggle output)

Notes: 1. Only 0 can be written to bits 7 to 5, to clear these flags.
2. Timer output is disabled when bits OS3 to OS0 are all 0. Timer output is 0 un compare match occurs after resetting.

				sure to read the flag after writing 0 to it.) • When the DTC is activated by a CMIB inter while the DISEL bit in MRB of the DTC is 0
6	CMFA	0	R/(W)*[1]	Compare Match Flag A [Setting condition] • When TCNT matches TCORA [Clearing conditions] • When writing 0 after reading CMFA = 1 (When the CPU is used to clear this flag by while the corresponding interrupt is enablec sure to read the flag after writing 0 to it.) • When the DTC is activated by a CMIA inter while the DISEL bit in MRB of the DTC is 0
5	OVF	0	R/(W)*[1]	Timer Overflow Flag [Setting condition] • When TCNT overflows from H'FF to H'00 [Clearing condition] • Cleared by reading OVF when OVF = 1, th 0 to OVF (When the CPU is used to clear this flag by while the corresponding interrupt is enablec sure to read the flag after writing 0 to it.)
4	—	1	R	Reserved This is a read-only bit and cannot be modified.

1	OS1	0	R/W	Output Select 1 and 0*[2]
0	OS0	0	R/W	These bits select a method of TMO pin outpu compare match A of TCORA and TCNT occu 00: No change when compare match A occur 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 (toggle output)

Notes: 1. Only 0 can be written to bits 7 to 5, to clear these flags.
2. Timer output is disabled when bits OS3 to OS0 are all 0. Timer output is 0 un
compare match occurs after resetting.

12.4 Operation

12.4.1 Pulse Output

Figure 12.3 shows an example of the 8-bit timer being used to generate a pulse output w
desired duty cycle. The control bits are set as follows:

1. In TCR, clear bit CCLR1 to 0 and set bit CCLR0 to 1 so that TCNT is cleared at a T
compare match.
2. In TCSR, set bits OS3 to OS0 to B'0110, causing the output to change to 1 at a TCO
compare match and to 0 at a TCORB compare match.

With these settings, the 8-bit timer provides pulses output at a cycle determined by TCO
pulse width determined by TCORB. No software intervention is required. The output lev
8-bit timer holds 0 until the first compare match occurs after a reset.

12.4.2 Reset Input

Figure 12.4 shows an example of the 8-bit timer being used to generate a pulse which is o
after a desired delay time from a TMRI input. The control bits are set as follows:

1. Set both bits CCLR1 and CCLR0 in TCR to 1 and set the TMRIS bit in TCCR to 1 so
   TCNT is cleared at the high level input of the TMRI signal.
2. In TCSR, set bits OS3 to OS0 to B'0110, causing the output to change to 1 at a TCOR
   compare match and to 0 at a TCORB compare match.

With these settings, the 8-bit timer provides pulses output at a desired delay time from a T
input determined by TCORA and with a pulse width determined by TCORB and TCORA

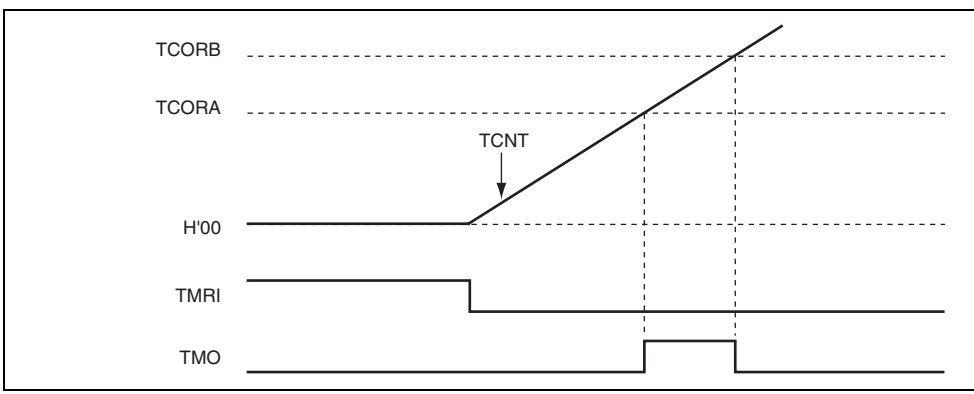


Figure 12.4 Example of Reset Input

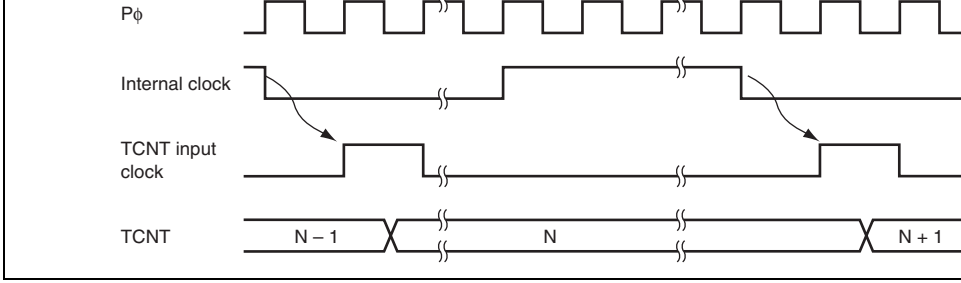


Figure 12.5 Count Timing for Internal Clock Input (Falling Edge)

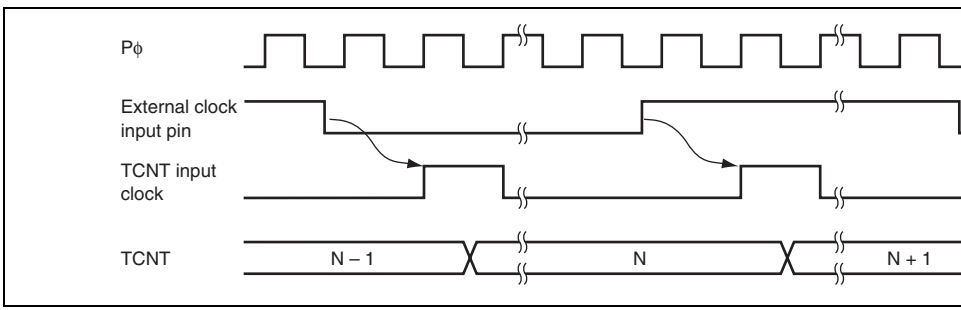


Figure 12.6 Count Timing for External Clock Input (Both Edges)

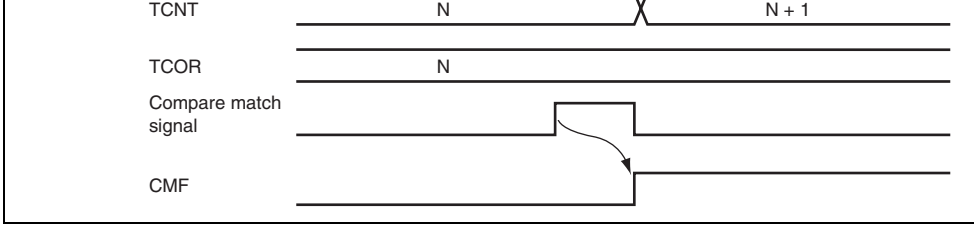


Figure 12.7 Timing of CMF Setting at Compare Match

12.5.3 Timing of Timer Output at Compare Match

When a compare match signal is generated, the timer output changes as specified by bits OS0 in TCSR. Figure 12.8 shows the timing when the timer output is toggled by the com match A signal.

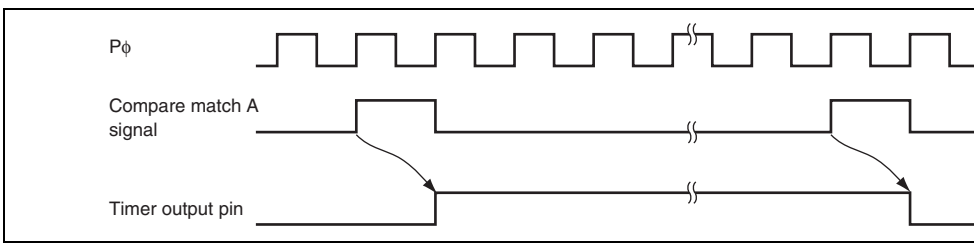


Figure 12.8 Timing of Toggled Timer Output at Compare Match A

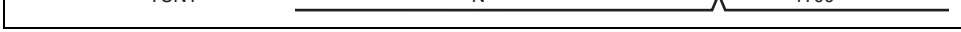


Figure 12.9 Timing of Counter Clear by Compare Match

12.5.5 Timing of TCNT External Reset

TCNT is cleared at the rising edge or high level of an external reset input, depending on settings of bits CCLR1 and CCLR0 in TCR. The clear pulse width must be at least 2 sta Figures 12.10 and 12.11 show the timing of this operation.

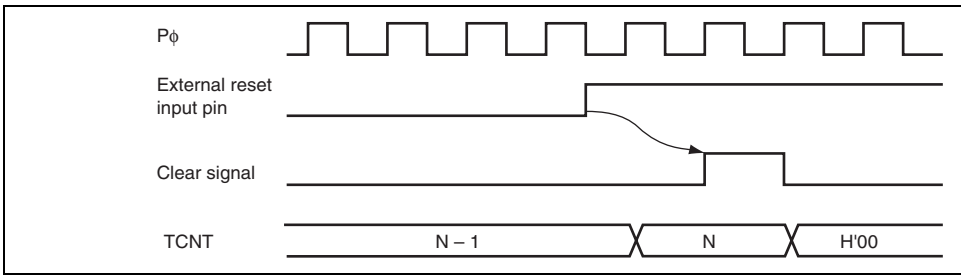


Figure 12.10 Timing of Clearance by External Reset (Rising Edge)

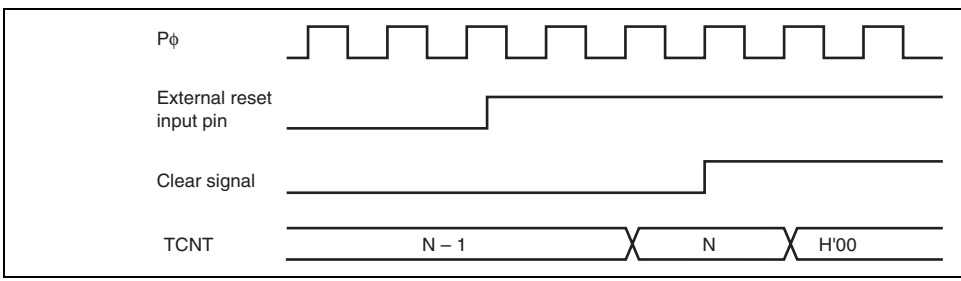


Figure 12.11 Timing of Clearance by External Reset (High Level)

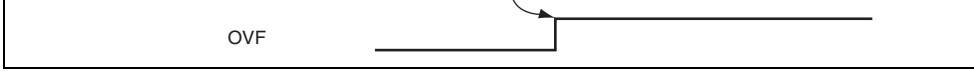


Figure 12.12 Timing of OVF Setting

with channel 0 occupying the upper 8 bits and channel 1 occupying the lower 8 bits.

(1) 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

(2) Counter Clear Specification

- If the CCLR1 and CCLR0 bits in TCR_0 have been set for counter clear at compare 16-bit counter (TCNT_0 and TCNT_1 together) is cleared when a 16-bit compare m occurs. The 16-bit counter (TCNT0 and TCNT1 together) is cleared even if counter the TMRI0 pin has been set.
- The settings of the CCLR1 and CCLR0 bits in TCR_1 are ignored. The lower 8 bits cleared independently.

(3) Pin Output

- Control of output from the TMO0 pin by bits OS3 to OS0 in TCSR_0 is in accordan 16-bit compare match conditions.
- Control of output from the TMO1 pin by bits OS3 to OS0 in TCSR_1 is in accordan lower 8-bit compare match conditions.

12.6.2 Compare Match Count Mode

When bits CKS2 to CKS0 in TCR_1 are set to B'100, TCNT_1 counts compare match A channel 0. Channels 0 and 1 are controlled independently. Conditions such as setting of flag, generation of interrupts, output from the TMO pin, and counter clear are in accorda the settings for each channel.

Table 12.3 8-Bit Timer (TMR_0 or TMR_1) Interrupt Sources

Name	Interrupt Source	Interrupt Flag	DTC Activation	Pri
CMIA0	TCORA_0 compare match	CMFA	Possible (VNUM = 2'b00)	Hig
CMIB0	TCORB_0 compare match	CMFB	Possible (VNUM = 2'b01)	
OVI0	TCNT_0 overflow	OVF	Not possible	Lov
CMIA1	TCORA_1 compare match	CMFA	Possible (VNUM = 2'b10)	Hig
CMIB1	TCORB_1 compare match	CMFB	Possible (VNUM = 2'b11)	
OVI1	TCNT_1 overflow	OVF	Not possible	Lov

Note: VNUM is an internal signal.

12.7.2 A/D Converter Activation

The A/D converter can be activated only by TMR_0 compare match A.

If the ADTE bit in TCSR_0 is set to 1 when the CMFA flag in TCSR_0 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
converter. If the 8-bit timer conversion start trigger has been selected on the A/D convert
this time, A/D conversion is started.

f: Counter frequency
ϕ: Operating frequency
N: TCOR value

12.8.2 Conflict between TCNT Write and Clear

If a counter clear signal is generated during the T_2 state of a TCNT write cycle, the clear priority and the write is not performed as shown in figure 12.13.

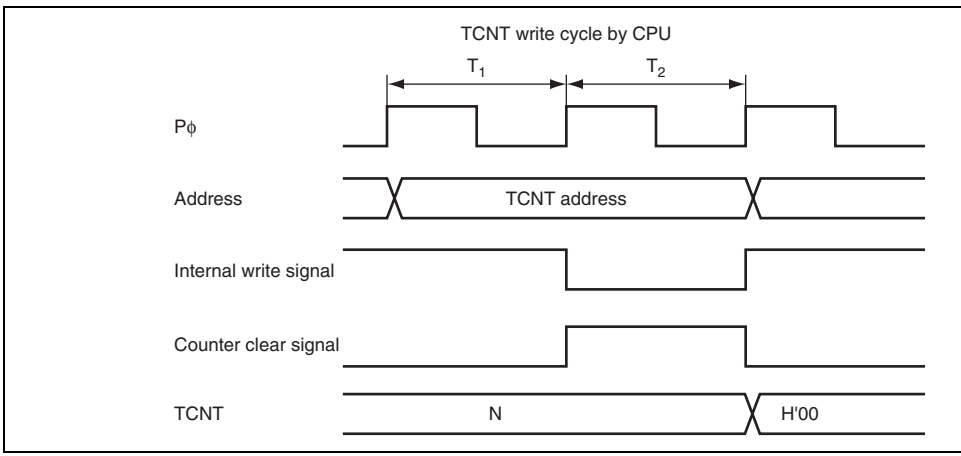


Figure 12.13 Conflict between TCNT Write and Clear

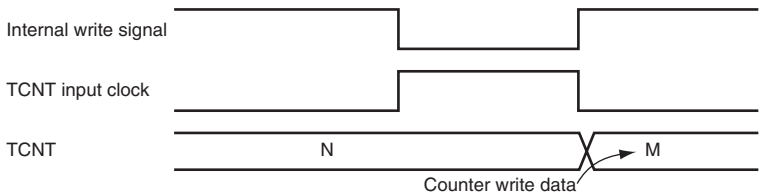


Figure 12.14 Conflict between TCNT Write and Increment

12.8.4 Conflict between TCOR Write and Compare Match

If a compare match event occurs during the T_2 state of a TCOR write cycle, the TCOR wr
priority and the compare match signal is inhibited as shown in figure 12.15.

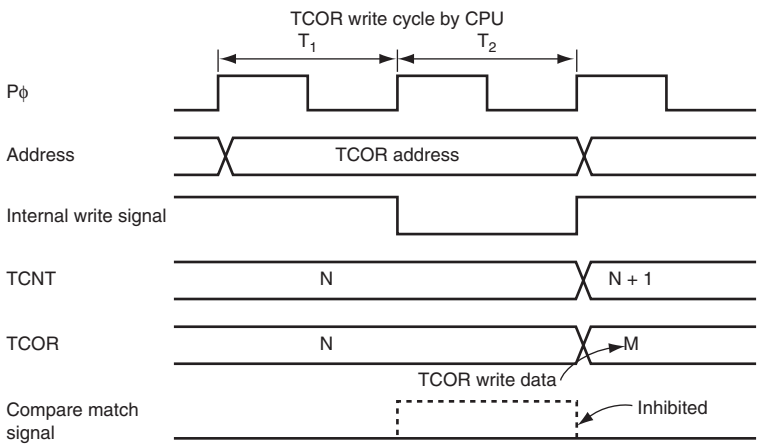


Figure 12.15 Conflict between TCOR Write and Compare Match

0-output	
No change	Lo

12.8.6 Switching of Internal Clocks and TCNT Operation

TCNT may be incremented erroneously depending on when the internal clock is switche
12.5 shows the relationship between the timing at which the internal clock is switched (b
to bits CKS1 and CKS0) and the TCNT operation.

When the TCNT clock is generated from an internal clock, the rising or falling edge of t
clock pulse are always monitored. Table 12.5 assumes that the falling edge is selected. I
signal levels of the clocks before and after switching change from high to low as shown
the change is considered as the falling edge. Therefore, a TCNT clock pulse is generated
TCNT is incremented. This is similar to when the rising edge is selected.

The erroneous incrementation of TCNT can also happen when switching between rising
falling edges of the internal clock, and when switching between internal and external clo

		TCNT N N + 1 CKS bits changed
2	Switching from low to high*2	Clock before switchover Clock after switchover TCNT input clock TCNT N N + 1 N + 2 CKS bits changed
3	Switching from high to low*3	Clock before switchover Clock after switchover *4 TCNT input clock TCNT N N + 1 N + 2 CKS bits changed

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 because the change of the signal levels is considered as a falling
TCNT is incremented.

12.8.7 Mode Setting with Cascaded Connection

If 16-bit counter mode and compare match count mode are specified at the same time, in
for TCNT_0 and TCNT_1 are not generated, and the counter stops. Do not specify 16-b
mode and compare match count mode simultaneously.

12.8.8 Module Stop Function Setting

Operation of the TMR can be disabled or enabled using the module stop control register
initial setting is for operation of the TMR to be halted. Register access is enabled by clea
module stop state. For details, refer to section 20, Power-Down Modes.

12.8.9 Interrupts in Module Stop State

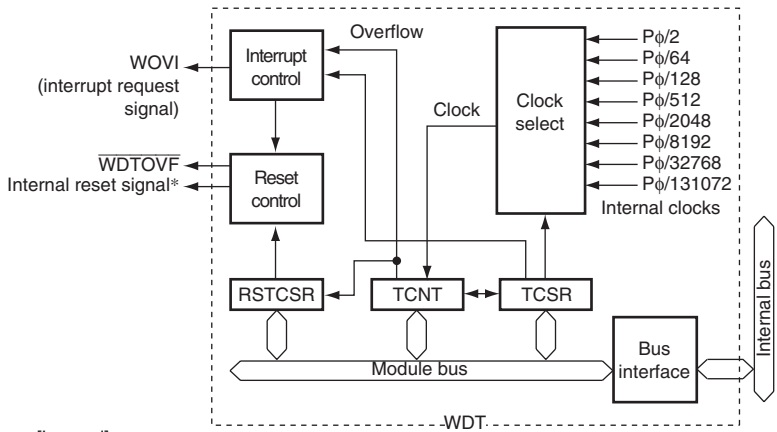
If the module stop state is entered when an interrupt has been requested, it will not be po
clear the CPU interrupt source or the DTC activation source. Interrupts should therefore
disabled before entering the module stop state.

13.1 Features

- Selectable from eight counter input clocks
- Switchable between watchdog timer mode and interval timer mode
  - — In watchdog timer mode

    If the counter overflows, the WDT outputs $\overline{\text{WDTOVF}}$. It is possible to select whe
    not the entire LSI is reset at the same time.
  - — In interval timer mode

    If the counter overflows, the WDT generates an interval timer interrupt (WOVI).



[Legend]
TCSR: Timer control/status register
TCNT: Timer counter
RSTCSR: Reset control/status register

Note: * An internal reset signal can be generated by the RSTCSR setting.

Figure 13.1 Block Diagram of WDT

13.3 Register Descriptions

The WDT has the following three registers. To prevent accidental overwriting, TCSR, TC
RSTCSR have to be written to in a method different from normal registers. For details, se
13.6.1, Notes on Register Access.

- Timer counter (TCNT)
- Timer control/status register (TCSR)
- Reset control/status register (RSTCSR)

13.3.1 Timer Counter (TCNT)

TCNT is an 8-bit readable/writable up-counter. TCNT is initialized to H'00 when the TM
TCSR is cleared to 0.

Bit	7	6	5	4	3	2	1	
Bit Name								
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Bit	Bit Name	Initial Value	R/W	Description
7	OVF	0	R/(W)*	Overflow Flag Indicates that TCNT has overflowed in interval mode. Only 0 can be written to this bit, to clear [Setting condition] When TCNT overflows in interval timer mode (c from H'FF to H'00) • When internal reset request generation is se watchdog timer mode, OVF is cleared autor by the internal reset. [Clearing condition] • Cleared by reading TCSR when OVF = 1, th 0 to OVF (When the CPU is used to clear this flag by while the corresponding interrupt is enablec to read the flag after writing 0 to it.)
6	WT/$\overline{\text{IT}}$	0	R/W	Timer Mode Select Selects whether the WDT is used as a watchdo interval timer. 0: Interval timer mode When TCNT overflows, an interval timer inte (WOVI) is requested. 1: Watchdog timer mode When TCNT overflows, the $\overline{\text{WDTOVF}}$ signal

0	CKS0	0	R/W	cycle for Pϕ = 20 MHz is indicated in parenthese 000: Clock Pϕ/2 (cycle: 25.6 μs) 001: Clock Pϕ/64 (cycle: 819.2 μs) 010: Clock Pϕ/128 (cycle: 1.6 ms) 011: Clock Pϕ/512 (cycle: 6.6 ms) 100: Clock Pϕ/2048 (cycle: 26.2 ms) 101: Clock Pϕ/8192 (cycle: 104.9 ms) 110: Clock Pϕ/32768 (cycle: 419.4 ms) 111: Clock Pϕ/131072 (cycle: 1.68 s)

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

13.3.3 Reset Control/Status Register (RSTCSR)

RSTCSR controls the generation of the internal reset signal when TCNT overflows, and s
the type of internal reset signal. RSTCSR is initialized to H'1F by a reset signal from the
but not by the WDT internal reset signal caused by WDT overflows.

Bit	7	6	5	4	3	2	1	
Bit Name	WOVF	RSTE	—	—	—	—	—	
Initial Value	0	0	0	1	1	1	1	
R/W	R/(W)*	R/W	R/W	R	R	R	R	

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

				• Reading RSTCSR when WOVF = 1, and th 0 to WOVF
6	RSTE	0	R/W	Reset Enable Specifies whether or not this LSI is internally re TCNT overflows during watchdog timer operati 0: LSI is not reset even if TCNT overflows (Tho LSI is not reset, TCNT and TCSR in WDT ar 1: LSI is reset if TCNT overflows
5	—	0	R/W	Reserved Although this bit is readable/writable, reading fr writing to this bit does not affect operation.
4 to 0	—	All 1	R	Reserved These are read-only bits and cannot be modifie

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

to reset the LSI internally in watchdog timer mode.

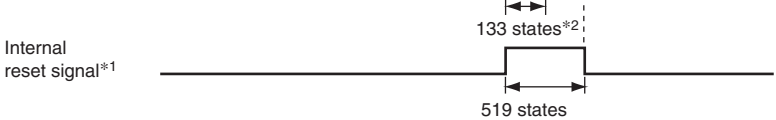
If TCNT overflows when the RSTE bit in RSTCSR is set to 1, a signal that resets this LS internally is generated at the same time as the $\overline{\text{WDTOVF}}$ signal. If a reset caused by a sig to the $\overline{\text{RES}}$ pin occurs at the same time as a reset caused by a WDT overflow, the $\overline{\text{RES}}$ pi has priority and the WOVF bit in RSTCSR is cleared to 0.

The $\overline{\text{WDTOVF}}$ signal is output for 133 cycles of Pϕ when RSTE = 1 in RSTCSR, and for cycles of Pϕ when RSTE = 0 in RSTCSR. The internal reset signal is output for 519 cycl

When RSTE = 1, an internal reset signal is generated. Since the system clock control regi (SCKCR) is initialized, the multiplication ratio of Pϕ becomes the initial value.

When RSTE = 0, an internal reset signal is not generated. Neither SCKCR nor the multip ratio of Pϕ is changed.

When TCNT overflows in watchdog timer mode, the WOVF bit in RSTCSR is set to 1. I overflows when the RSTE bit in RSTCSR is set to 1, an internal reset signal is generated entire LSI.



Notes: 1. If TCNT overflows when the RSTE bit is set to 1, an internal reset signal is generated.
2. 130 states when the RSTE bit is cleared to 0.

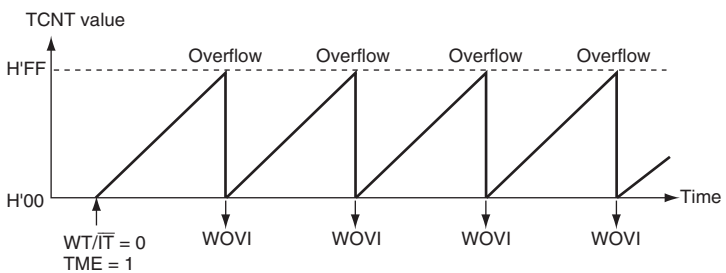
Figure 13.2 Operation in Watchdog Timer Mode

13.4.2 Interval Timer Mode

To use the WDT as an interval timer, set the WT/$\overline{\text{IT}}$ bit to 0 and the TME bit to 1 in TCS

When the WDT is used as an interval timer, an interval timer interrupt (WOVI) is gener
time the TCNT overflows. Therefore, an interrupt can be generated at intervals.

When the TCNT overflows in interval timer mode, an interval timer interrupt (WOVI) is
at the same time the OVF bit in the TCSR is set to 1.



[Legend]
WOVI: Interval timer interrupt request

Figure 13.3 Operation in Interval Timer Mode

13.6 Usage Notes

13.6.1 Notes on Register Access

The watchdog timer's TCNT, TCSR, and RSTCSR registers differ from other registers in
more difficult to write to. The procedures for writing to and reading these registers are gi
below.

(1) Writing to TCNT, TCSR, and RSTCSR

TCNT and TCSR must be written to by a word transfer instruction. They cannot be writte
byte transfer instruction.

For writing, TCNT and TCSR are assigned to the same address. Accordingly, perform da
transfer as shown in figure 13.4. The transfer instruction writes the lower byte data to TC
TCSR.

To write to RSTCSR, execute a word transfer instruction for address H'FFA6. A byte tra
instruction cannot be used to write to RSTCSR.

The method of writing 0 to the WOVF bit in RSTCSR differs from that of writing to the
in RSTCSR. Perform data transfer as shown in figure 13.4.

At data transfer, the transfer instruction clears the WOVF bit to 0, but has no effect on th
bit. To write to the RSTE bit, perform data transfer as shown in figure 13.4. In this case, t
transfer instruction writes the value in bit 6 of the lower byte to the RSTE bit, but has no
the WOVF bit.

Figure 13.4 Writing to TCNT, TCSR, and RSTCSR

(2) Reading from TCNT, TCSR, and RSTCSR

These registers can be read from in the same way as other registers. For reading, TCSR
to address H'FFA4, TCNT to address H'FFA5, and RSTCSR to address H'FFA7.

13.6.2 Conflict between Timer Counter (TCNT) Write and Increment

If a TCNT clock pulse is generated during the T2 cycle of a TCNT write cycle, the write
priority and the timer counter is not incremented. Figure 13.5 shows this operation.

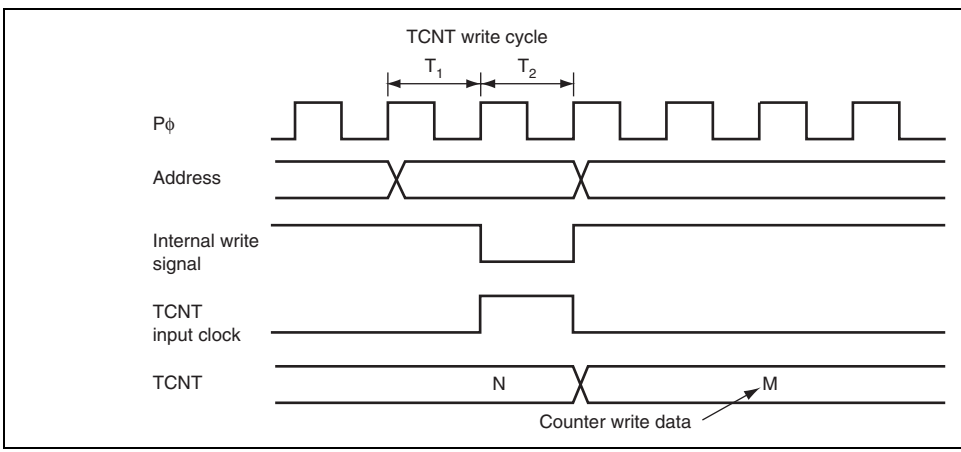


Figure 13.5 Conflict between TCNT Write and Increment

clearing the TME bit to 0) before switching the timer mode.

13.6.5 Internal Reset in Watchdog Timer Mode

This LSI is not reset internally if TCNT overflows while the RSTE bit is cleared to 0 duri
watchdog timer mode operation, but TCNT and TCSR of the WDT are reset.

TCNT, TCSR, and RSTCR cannot be written to while the $\overline{\text{WDTOVF}}$ signal is low. Also
a read of the WOVF flag is not recognized during this period. To clear the WOVF flag, th
read TCSR after the $\overline{\text{WDTOVF}}$ signal goes high, then write 0 to the WOVF flag.

13.6.6 System Reset by $\overline{\text{WDTOVF}}$ Signal

If the $\overline{\text{WDTOVF}}$ signal is input to the $\overline{\text{RES}}$ pin, this LSI will not be initialized correctly. N
sure that the $\overline{\text{WDTOVF}}$ signal is not input logically to the $\overline{\text{RES}}$ pin. To reset the entire sys
means of the $\overline{\text{WDTOVF}}$ signal, use a circuit like that shown in figure 13.6.

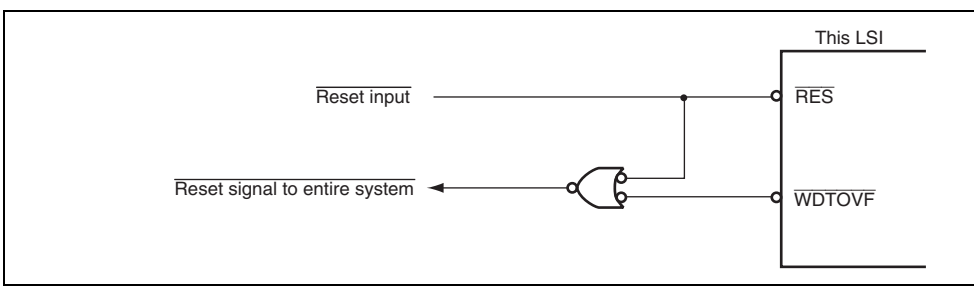


Figure 13.6 Circuit for System Reset by $\overline{\text{WDTOVF}}$ Signal (Example)

14.1 Features

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

  The transmitter and receiver are mutually independent, enabling transmission and re
  be executed simultaneously. Double-buffering is used in both the transmitter and the
  enabling continuous transmission and continuous reception of serial data.
- On-chip baud rate generator allows any bit rate to be selected

  The external clock can be selected as a transfer clock source (except for the smart ca
  interface).
- Choice of LSB-first or MSB-first transfer (except in the case of asynchronous mode
- Four interrupt sources

  The interrupt sources are transmit-end, transmit-data-empty, receive-data-full, and re
  error. The transmit-data-empty and receive-data-full interrupt sources can activate th
  or DTC.
- Module stop state specifiable

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 case
  framing error

- An error signal can be automatically transmitted on detection of a parity error during
- Data can be automatically re-transmitted on receiving an error signal during transmiss
- Both direct convention and inverse convention are supported

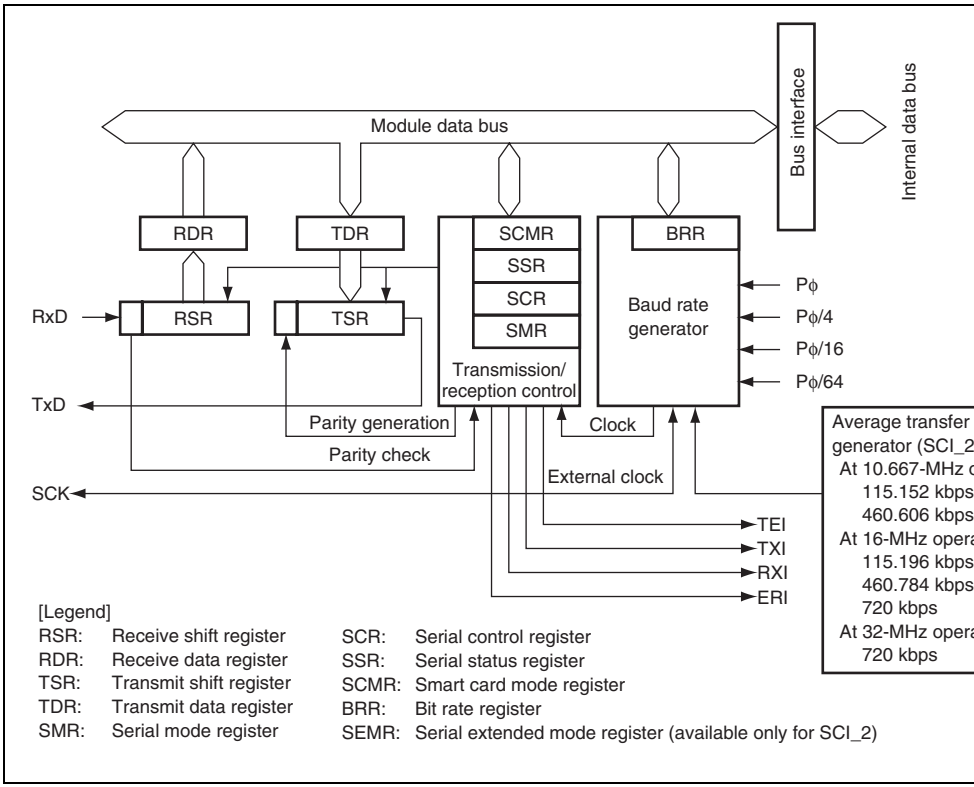


Figure 14.1 Block Diagram of SCI

1	SCK1	I/O	Channel 1 clock input/output
	RxD1	Input	Channel 1 receive data input
	TxD1	Output	Channel 1 transmit data output
2	SCK2	I/O	Channel 2 clock input/output
	RxD2	Input	Channel 2 receive data input
	TxD2	Output	Channel 2 transmit data output
4	SCK4	I/O	Channel 4 clock input/output
	RxD4	Input	Channel 4 receive data input
	TxD4	Output	Channel 4 transmit data output

Note: * Pin names SCK, RxD, and TxD are used in the text for all channels, omitting channel designation.

- Receive data register_0 (RDR_0)
- Transmit data register_0 (TDR_0)
- Serial mode register_0 (SMR_0)
- Serial control register_0 (SCR_0)
- Serial status register_0 (SSR_0)
- Smart card mode register_0 (SCMR_0)
- Bit rate register_0 (BRR_0)

Channel 1:

- Receive shift register_1 (RSR_1)
- Transmit shift register_1 (TSR_1)
- Receive data register_1 (RDR_1)
- Transmit data register_1 (TDR_1)
- Serial mode register_1 (SMR_1)
- Serial control register_1 (SCR_1)
- Serial status register_1 (SSR_1)
- Smart card mode register_1 (SCMR_1)
- Bit rate register_1 (BRR_1)

- Bit rate register_2 (BRR_2)
- Serial extended mode register_2 (SEMR_2) (SCI_2 only)

Channel 4:

- Receive shift register_4 (RSR_4)
- Transmit shift register_4 (TSR_4)
- Receive data register_4 (RDR_4)
- Transmit data register_4 (TDR_4)
- Serial mode register_4 (SMR_4)
- Serial control register_4 (SCR_4)
- Serial status register_4 (SSR_4)
- Smart card mode register_4 (SCMR_4)
- Bit rate register_4 (BRR_4)

14.3.1 Receive Shift Register (RSR)

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

Initial Value	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R

14.3.3 Transmit Data Register (TDR)

TDR is an 8-bit register that stores transmit data. When the SCI detects that TSR is empty transfers the transmit data written in TDR to TSR and starts transmission. The double-buf structures of TDR and TSR enables continuous serial transmission. If the next transmit da already been written to TDR when one frame of data is transmitted, the SCI transfers the data to TSR to continue transmission. Although TDR can be read from or written to by th all times, to achieve reliable serial transmission, write transmit data to TDR for only once confirming that the TDRE bit in SSR is set to 1.

Bit	7	6	5	4	3	2	1	
Bit Name								
Initial Value	1	1	1	1	1	1	1	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

14.3.4 Transmit Shift Register (TSR)

TSR is a shift register that transmits serial data. To perform serial data transmission, the S automatically transfers transmit data from TDR to TSR, then sends the data to the TxD pi cannot be directly accessed by the CPU.

- When SMIF in SCMR = 1

Bit	7	6	5	4	3	2	1
Bit Name	GM	BLK	PE	O/$\overline{E}$	BCP1	BCP0	CKS1
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Functions in Normal Serial Communication Interface Mode (When SMIF in SC

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 (valid only in asynchronous m 0: Selects 8 bits as the data length. 1: Selects 7 bits as the data length. LSB-first is the MSB (bit 7) in TDR is not transmitted in transmission. In clocked synchronous mode, a fixed data leng bits is used.
5	PE	0	R/W	Parity Enable (valid only in asynchronous mode When this bit is set to 1, the parity bit is added t data before transmission, and the parity bit is cl reception. For a multiprocessor format, parity b and checking are not performed regardless of t setting.

				In reception, only the first stop bit is checked. If t second stop bit is 0, it is treated as the start bit o transmit frame.
2	MP	0	R/W	Multiprocessor Mode (valid only in asynchronous When this bit is set to 1, the multiprocessor func enabled. The PE bit and O/$\overline{E}$ bit settings are inva multiprocessor mode.
1	CKS1	0	R/W	Clock Select 1, 0
0	CKS0	0	R/W	These bits select the clock source for the baud r generator. 00: Pϕ clock (n = 0) 01: Pϕ/4 clock (n = 1) 10: Pϕ/16 clock (n = 2) 11: Pϕ/64 clock (n = 3) For the relation between the settings of these bit baud rate, see section 14.3.9, Bit Rate Register is the decimal display of the value of n in BRR (s section 14.3.9, Bit Rate Register (BRR)).

Bit Functions in Smart Card Interface Mode (When SMIF in SCMR = 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. mode, the TEND set timing is put forward to 11.0 the start and the clock output control function is appended. For details, see sections 14.7.6, Data Transmission (Except in Block Transfer Mode) a 14.7.8, Clock Output Control.

				1: Selects odd parity For details on the usage of this bit in smart card mode, see section 14.7.2, Data Format (Except 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 for 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 14.7.4, Receive Data Sampling Timing and Reception Margin. S is described in section 14.3.9, 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: Pϕ clock (n = 0) 01: Pϕ/4 clock (n = 1) 10: Pϕ/16 clock (n = 2) 11: Pϕ/64 clock (n = 3) For the relation between the settings of these bits and baud rate, see section 14.3.9, Bit Rate Register (BRR). (n is the decimal display of the value of n in BRR (see section 14.3.9, Bit Rate Register (BRR)).

Note: etu (Elementary Time Unit): 1-bit transfer time

Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- When SMIF in SCMR = 1

Bit	7	6	5	4	3	2	1	
Bit Name	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Bit Functions in Normal Serial Communication Interface Mode (When SMIF in SC

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable When this bit is set to 1, a TXI interrupt request i enabled. A TXI interrupt request can be cancelled by read from the TDRE flag and then clearing the flag to 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 re are enabled. RXI and ERI interrupt requests can be cancelled reading 1 from the RDRF, FER, PER, or ORER then clearing the flag to 0, or by clearing the RIE

				When this bit is set to 1, reception is enabled. U condition, serial reception is started by detectin bit in asynchronous mode or the synchronous c in clocked synchronous mode. Note that SMR s set prior to setting the RE bit to 1 in order to de the reception format. Even if reception is halted by clearing this bit to RDRF, FER, PER, and ORER flags are not affe the previous value is retained.
3	MPIE	0	R/W	Multiprocessor Interrupt Enable (valid only whe bit in SMR is 1 in asynchronous mode) When this bit is set to 1, receive data in which t multiprocessor bit is 0 is skipped, and setting of RDRF, FER, and ORER status flags in SSR is On receiving data in which the multiprocessor b bit is automatically cleared and normal receptio resumed. For details, see section 14.5, Multipro Communication Function. When receive data including MPB = 0 in SSR is received, transfer of the received data from RS detection of reception errors, and the settings o FER, and ORER flags in SSR are not performe receive data including MPB = 1 is received, the in SSR is set to 1, the MPIE bit is automatically 0, and RXI and ERI interrupt requests (in the ca the TIE and RIE bits in SCR are set to 1) and s the FER and ORER flags are enabled.

00: On-chip baud rate generator

(SCK pin functions as I/O port.)

01: On-chip baud rate generator

(Outputs a clock with the same frequency as rate from the SCK pin.)

1X: External clock

(Inputs a clock with a frequency 16 times the 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.)

Note: X: Don't care

				When this bit is set to 1, RXI and ERI interrupt are enabled. RXI and ERI interrupt requests can be cancelle reading 1 from the RDRF, FER, PER, or ORER then clearing the flag to 0, or by clearing the RI
5	TE	0	R/W	Transmit Enable When this bit is set to 1, transmission is enable this condition, serial transmission is started by transmit data to TDR, and clearing the TDRE fl to 0. Note that SMR should be set prior to settin bit to 1 in order to designate the transmission fo If transmission is halted by clearing this bit to 0, TDRE flag in SSR is fixed 1.
4	RE	0	R/W	Receive Enable When this bit is set to 1, reception is enabled. U condition, serial reception is started by detectin bit in asynchronous mode or the synchronous c in clocked synchronous mode. Note that SMR s set prior to setting the RE bit to 1 in order to de the reception format. Even if reception is halted by clearing this bit to RDRF, FER, PER, and ORER flags are not affe the previous value is retained.
3	MPIE	0	R/W	Multiprocessor Interrupt Enable (valid only whe bit in SMR is 1 in asynchronous mode) Write 0 to this bit in smart card interface mode.
2	TEIE	0	R/W	Transmit End Interrupt Enable Write 0 to this bit in smart card interface mode.

- When GM in SMR = 1

00: Output fixed low

01: Clock output

10: Output fixed high

11: Clock output

14.3.7 Serial Status Register (SSR)

SSR is a register containing status flags of the SCI and multiprocessor bits for transfer. T
RDRF, ORER, PER, and FER can only be cleared. Some bits in SSR have different funct
normal mode and smart card interface mode.

- When SMIF in SCMR = 0

Bit	7	6	5	4	3	2	1	
Bit Name	TDRE	RDRF	ORER	FRE	PER	TEND	MPB	
Initial Value	1	0	0	0	0	1	0	
R/W	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	

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

- When SMIF in SCMR = 1

Bit	7	6	5	4	3	2	1	
Bit Name	TDRE	RDRF	ORER	ERS	PER	TEND	MPB	
Initial Value	1	0	0	0	0	1	0	
R/W	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	

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

				• When 0 is written to TDRE after reading TD (When the CPU is used to clear this flag by while the corresponding interrupt is enablec to read the flag after writing 0 to it.) • When a TXI interrupt request is issued allov DMAC or DTC to write data to TDR
6	RDRF	0	R/(W)*	Receive Data Register Full Indicates whether receive data is stored in RDF [Setting condition] • When serial reception ends normally and re is transferred from RSR to RDR [Clearing conditions] • When 0 is written to RDRF after reading RD (When the CPU is used to clear this flag by while the corresponding interrupt is enablec to read the flag after writing 0 to it.) • When an RXI interrupt request is issued allo DMAC or DTC to read data from RDR The RDRF flag is not affected and retains its pr value when the RE bit in SCR is cleared to 0. Note that when the next serial reception is com while the RDRF flag is being set to 1, an overru occurs and the received data is lost.

is set to 1, subsequent serial reception canno
performed. Note that, in clocked synchronous
serial transmission also cannot continue.

[Clearing condition]

- When 0 is written to ORER after reading ORE
(When the CPU is used to clear this flag by w
while the corresponding interrupt is enabled,
to read the flag after writing 0 to it.)

  Even when the RE bit in SCR is cleared, the
flag is not affected and retains its previous va

is transferred to RDR, however, the RDRF f
set. In addition, when the FER flag is being
the subsequent serial reception cannot be p
In clocked synchronous mode, serial transm
also cannot continue.

[Clearing condition]

- When 0 is written to FER after reading FER
(When the CPU is used to clear this flag by
while the corresponding interrupt is enablec
to read the flag after writing 0 to it.)

  Even when the RE bit in SCR is cleared, the
is not affected and retains its previous value

				subsequent serial reception cannot be perfor clocked synchronous mode, serial transmissi cannot continue. [Clearing condition] • When 0 is written to PER after reading PER * (When the CPU is used to clear this flag by w while the corresponding interrupt is enabled, to read the flag after writing 0 to it.) Even when the RE bit in SCR is cleared, the is not affected and retains its previous value.
2	TEND	1	R	Transmit End [Setting conditions] • When the TE bit in SCR is 0 • When TDRE = 1 at transmission of the last b transmit character [Clearing conditions] • When 0 is written to TDRE after reading TDR • When a TXI interrupt request is issued allowi DMAC or DTC to write data to TDR
1	MPB	0	R	Multiprocessor Bit Stores the multiprocessor bit value in the receive When the RE bit in SCR is cleared to 0 its previo is retained.
0	MPBT	0	R/W	Multiprocessor Bit Transfer Sets the multiprocessor bit value to be added to transmit frame.

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

				• When 0 is written to TDRE after reading TD (When the CPU is used to clear this flag by while the corresponding interrupt is enablec to read the flag after writing 0 to it.) • When a TXI interrupt request is issued allov DMAC or DTC to write data to TDR
6	RDRF	0	R/(W)*	Receive Data Register Full Indicates whether receive data is stored in RDF [Setting condition] • When serial reception ends normally and re is transferred from RSR to RDR [Clearing conditions] • When 0 is written to RDRF after reading RD (When the CPU is used to clear this flag by while the corresponding interrupt is enablec to read the flag after writing 0 to it.) • When an RXI interrupt request is issued allo DMAC or DTC to read data from RDR The RDRF flag is not affected and retains its pr value even when the RE bit in SCR is cleared t Note that when the next reception is completed RDRF flag is being set to 1, an overrun error oc the received data is lost.

				is set to 1, subsequent serial reception canno performed. Note that, in clocked synchronous serial transmission also cannot continue. [Clearing condition] • When 0 is written to ORER after reading ORE (When the CPU is used to clear this flag by w while the corresponding interrupt is enabled, to read the flag after writing 0 to it.) Even when the RE bit in SCR is cleared, the flag is not affected and retains its previous va
4	ERS	0	R/(W)*	Error Signal Status [Setting condition] • When a low error signal is sampled [Clearing condition] • When 0 is written to ERS after reading ERS =

subsequent serial reception cannot be perfo
clocked synchronous mode, serial transmiss
cannot continue.

[Clearing condition]

- When 0 is written to PER after reading PER
(When the CPU is used to clear this flag by
while the corresponding interrupt is enablec
to read the flag after writing 0 to it.)

  Even when the RE bit in SCR is cleared, the
is not affected and retains its previous value

				When GM = 0 and BLK = 0, 2.5 etu after tran start When GM = 0 and BLK = 1, 1.5 etu after tran start When GM = 1 and BLK = 0, 1.0 etu after tran start When GM = 1 and BLK = 1, 1.0 etu after tran start [Clearing conditions] • When 0 is written to TEND after reading TEN • When a TXI interrupt request is issued allowi DMAC or DTC to write the next data to TDR
1	MPB	0	R	Multiprocessor Bit 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.

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

Bit	Bit Name	Value	R/W	Description
7 to 4	—	All 1	R	Reserved These are read-only bits and cannot be modifie
3	SDIR	0	R/W	Smart Card Data Transfer Direction Selects the serial/parallel conversion format. 0: Transfer with LSB-first 1: Transfer with MSB-first This bit is valid only when the 8-bit data format transmission/reception; when the 7-bit data form used, data is always transmitted/received with
2	SINV	0	R/W	Smart Card Data Invert Inverts the transmit/receive data logic level. Thi not affect the logic level of the parity bit. To inve parity bit, invert the O/$\overline{E}$ bit in SMR. 0: TDR contents are transmitted as they are. R data is stored as it is in RDR. 1: TDR contents are inverted before being trans Receive data is stored in inverted form in RD
1	—	1	R	Reserved This is a read-only bit and cannot be modified.
0	SMIF	0	R/W	Smart Card Interface Mode Select When this bit is set to 1, smart card interface m selected. 0: Normal asynchronous or clocked synchronou 1: Smart card interface mode

Asynchronous mode	$N = \frac{P\phi \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	Error (%) = $\{ \frac{P\phi \times 10^6}{B \times 64 \times 2^{2n-1} \times (N+1)} -$
Clocked synchronous mode	$N = \frac{P\phi \times 10^6}{8 \times 2^{2n-1} \times B} - 1$	
Smart card interface mode	$N = \frac{P\phi \times 10^6}{S \times 2^{2n+1} \times B} - 1$	Error (%) = $\{ \frac{P\phi \times 10^6}{B \times S \times 2^{2n+1} \times (N+1)} -$

[Legend]

B: Bit rate (bit/s)

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

Pϕ: Operating frequency (MHz)

n and S: Determined by the SMR settings shown in the following table.

SMR Setting			SMR Setting		
CKS1	**CKS0**	**n**	**BCP1**	**BCP0**	**S**
0	0	0	0	0	32
0	1	1	0	1	64
1	0	2	1	0	37
1	1	3	1	1	25

Table 14.3 shows sample N settings in BRR in normal asynchronous mode. Table 14.4 sh
maximum bit rate settable for each operating frequency. Tables 14.6 and 14.8 show samp
settings in BRR in clocked synchronous mode and smart card interface mode, respectivel
smart card interface mode, the number of basic clock cycles S in a 1-bit data transfer time
selected. For details, see section 14.7.4, Receive Data Sampling Timing and Reception M
Tables 14.5 and 14.7 show the maximum bit rates with external clock input.

1200	0	207	0.16	0	255	0.00	1	64	0.16	1	77
2400	0	103	0.16	0	127	0.00	0	129	0.16	0	155
4800	0	51	0.16	0	63	0.00	0	64	0.16	0	77
9600	0	25	0.16	0	31	0.00	0	32	–1.36	0	38
19200	0	12	0.16	0	15	0.00	0	15	1.73	0	19
31250	0	7	0.00	0	9	–1.70	0	9	0.00	0	11
38400	—	—	—	0	7	0.00	0	7	1.73	0	9

	Operating Frequency Pϕ (MHz)										
	12.288			14			14.7456			1	
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N
110	2	217	0.08	2	248	–0.17	3	64	0.70	3	70
150	2	159	0.00	2	181	0.16	2	191	0.00	2	207
300	2	79	0.00	2	90	0.16	2	95	0.00	2	103
600	1	159	0.00	1	181	0.16	1	191	0.00	1	207
1200	1	79	0.00	1	90	0.16	1	95	0.00	1	103
2400	0	159	0.00	0	181	0.16	0	191	0.00	0	207
4800	0	79	0.00	0	90	0.16	0	95	0.00	0	103
9600	0	39	0.00	0	45	–0.93	0	47	0.00	0	51
19200	0	19	0.00	0	22	–0.93	0	23	0.00	0	25
31250	0	11	2.40	0	13	0.00	0	14	–1.70	0	15
38400	0	9	0.00	—	—	—	0	11	0.00	0	12

1200	1	111	0.00	1	116	0.16	1	127	0.00	1	129
2400	0	223	0.00	0	233	0.16	0	255	0.00	1	64
4800	0	111	0.00	0	116	0.16	0	127	0.00	0	129
9600	0	55	0.00	0	58	–0.69	0	63	0.00	0	64
19200	0	27	0.00	0	28	1.02	0	31	0.00	0	32
31250	0	16	1.20	0	17	0.00	0	19	–1.70	0	19
38400	0	13	0.00	0	14	–2.34	0	15	0.00	0	15

	Operating Frequency Pϕ (MHz)										
	25			30			33			35	
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N
110	3	110	–0.02	3	132	0.13	3	145	0.33	3	154
150	3	80	0.47	3	97	–0.35	3	106	0.39	3	113
300	2	162	–0.15	2	194	0.16	2	214	–0.07	2	227
600	2	80	0.47	2	97	–0.35	2	106	0.39	2	113
1200	1	162	–0.15	1	194	0.16	1	214	–0.07	1	227
2400	1	80	0.47	1	97	–0.35	1	106	0.39	1	113
4800	0	162	–0.15	0	194	0.16	0	214	–0.07	0	227
9600	0	80	0.47	0	97	–0.35	0	106	0.39	0	113
19200	0	40	–0.76	0	48	–0.35	0	53	–0.54	0	56
31250	0	24	0.00	0	29	0	0	32	0	0	34
38400	0	19	1.73	0	23	1.73	0	26	–0.54	0	28

14	437500	0	0	30	937500	0
14.7456	460800	0	0	33	1031250	0
16	500000	0	0	35	1093750	0

Table 14.5 Maximum Bit Rate with External Clock Input (Asynchronous Mode)

Pϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bit/s)	Pϕ (MHz)	External Input Clock (MHz)	Max Rate
8	2.0000	125000	17.2032	4.3008	2688
9.8304	2.4576	153600	18	4.5000	2812
10	2.5000	156250	19.6608	4.9152	3072
12	3.0000	187500	20	5.0000	3125
12.288	3.0720	192000	25	6.2500	3906
14	3.5000	218750	30	7.5000	4687
14.7456	3.6864	230400	33	8.2500	5156
16	4.0000	250000	35	8.7500	5468

5 k	1	99	1	124	1	199	1	2
10 k	0	199	0	249	1	99	1	1
25 k	0	79	0	99	0	159	0	1
50 k	0	39	0	49	0	79	0	9
100 k	0	19	0	24	0	39	0	4
250 k	0	7	0	9	0	15	0	1
500 k	0	3	0	4	0	7	0	9
1 M	0	1			0	3	0	4
2.5 M			0	0*			0	1
5 M							0	0

10 k	1	155	1	187	1	205	1
25 k	0	249	1	74	1	82	1
50 k	0	124	0	149	0	164	0
100 k	0	62	0	74	0	82	0
250 k	0	24	0	29	0	32	0
500 k	—	—	0	14	—	—	—
1 M	—	—	—	—	—	—	—
2.5 M	—	—	0	2	—	—	—
5 M	—	—	—	—	—	—	—

[Legend]

Space : Setting prohibited.

— : Can be set, but there will be error.

* : Continuous transmission or reception is not possible.

Table 14.7 Maximum Bit Rate with External Clock Input (Clocked Synchronous

Pϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bit/s)	Pϕ (MHz)	External Input Clock (MHz)	Maximu Rate (bi
8	1.3333	1333333.3	20	3.3333	333333
10	1.6667	1666666.7	25	4.1667	416666
12	2.0000	2000000.0	30	5.0000	500000
14	2.3333	2333333.3	33	5.5000	550000
16	2.6667	2666666.7	35	5.8336	583362
18	3.0000	3000000.0			

(bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	E
9600	0	1	0.00	0	1	12.01	0	2	15.99	0	2	6

Bit Rate	Operating Frequency Pϕ (MHz)											
	25.00			30.00			33.00			35.0		
(bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	E
9600	0	3	12.49	0	3	5.01	0	4	7.59	0	4	1

Table 14.9 Maximum Bit Rate for Each Operating Frequency (Smart Card Interfa Mode, S = 372)

Pϕ (MHz)	Maximum Bit Rate (bit/s)	n	N	Pϕ (MHz)	Maximum Bit Rate (bit/s)	n
7.1424	9600	0	0	18.00	24194	0
10.00	13441	0	0	20.00	26882	0
10.7136	14400	0	0	25.00	33602	0
13.00	17473	0	0	30.00	40323	0
14.2848	19200	0	0	33.00	44355	0
16.00	21505	0	0	35.00	47043	0

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R/W	Reserved This bit is always read as 0. The write value s always be 0.
6 to 4	—	All 0	R	Reserved These are read-only bits and cannot be modif
3	ABCS	0	R/W	Asynchronous Mode Basic Clock Select (valid asynchronous mode) Selects the basic clock for a 1-bit period. 0: The basic clock has a frequency 16 times t rate 1: The basic clock has a frequency 8 times th rate

basic clock with a frequency 16 times the
rate)

010: 460.606 kbps of average transfer rate spe
Pϕ = 10.667 MHz is selected (operated us
basic clock with a frequency 8 times the t
rate)

011: 720 kbps of average transfer rate specific
32 MHz is selected (operated using the ba
with a frequency 16 times the transfer rate

100: Setting prohibited

101: 115.196 kbps of average transfer rate spe
Pϕ = 16 MHz is selected (operated using
clock with a frequency 16 times the transf

110: 460.784 kbps of average transfer rate spe
Pϕ = 16 MHz is selected (operated using
clock with a frequency 16 times the transf

111: 720 kbps of average transfer rate specific
16 MHz is selected (operated using the ba
with a frequency 8 times the transfer rate)

The average transfer rate only supports operat
frequencies of 10.667 MHz, 16 MHz, and 32 M

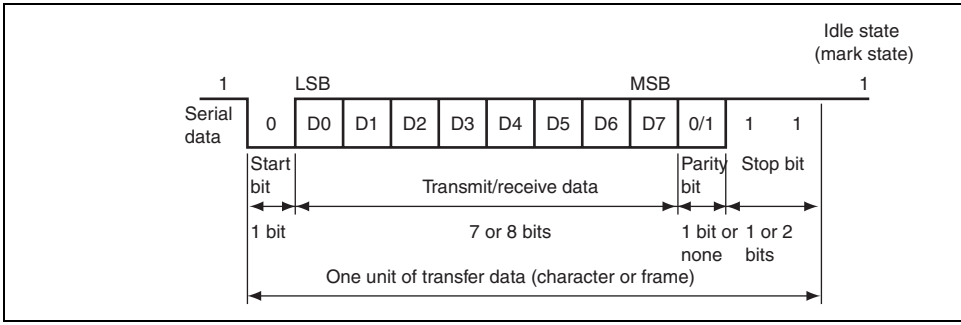


Figure 14.2 Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, Two Stop Bits)

0	0	0	0	S 8-bit data STOP
0	0	0	1	S 8-bit data STOP ST
0	1	0	0	S 8-bit data P ST
0	1	0	1	S 8-bit data P ST
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 ST
0	—	1	0	S 8-bit data MPB ST
0	—	1	1	S 8-bit data MPB ST
1	—	1	0	S 7-bit data MPB STOP
1	—	1	1	S 7-bit data MPB STOP ST

[Legend]
S: Start bit
STOP: Stop bit
P: Parity bit
MPB: Multiprocessor bit

N: Ratio of bit rate to clock (N = 16)
D: Duty cycle of clock (D = 0.5 to 1.0)
L: Frame length (L = 9 to 12)
F: Absolute value of clock frequency deviation

Assuming values of F = 0 and D = 0.5 in formula (1), the reception margin is determine
formula below.

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

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

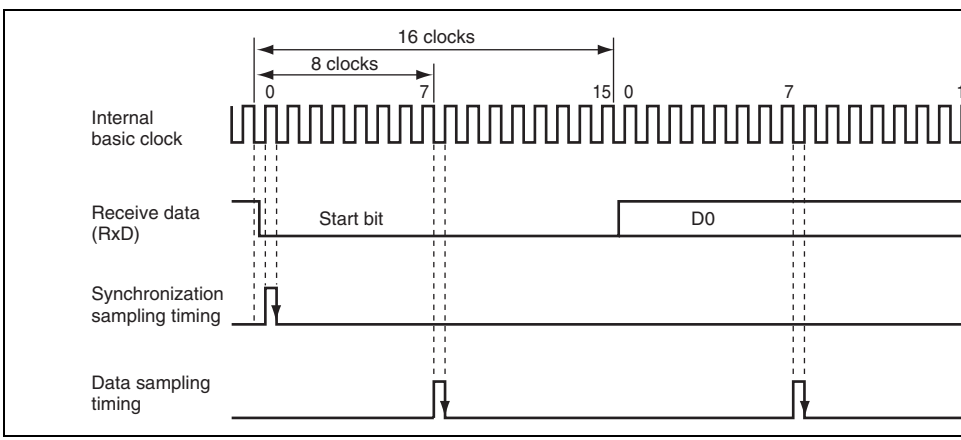


Figure 14.3 Receive Data Sampling Timing in Asynchronous Mode

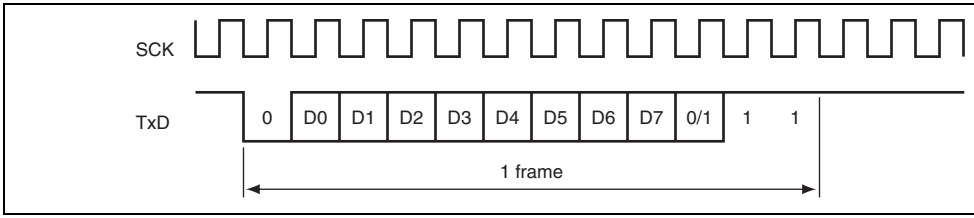
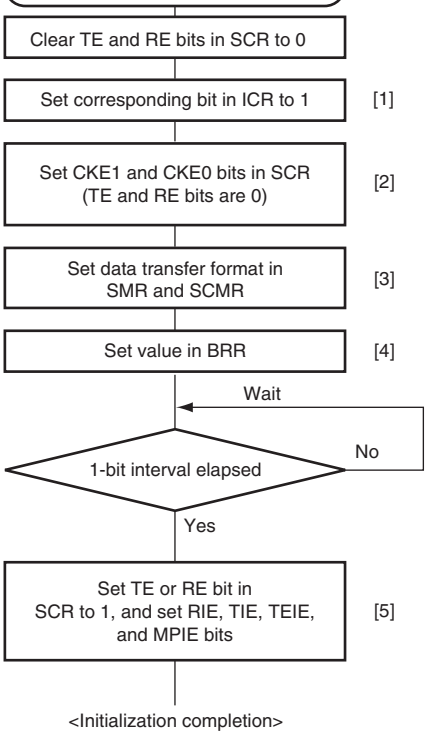


Figure 14.4 Phase Relation between Output Clock and Transmit Data (Asynchronous Mode)



[1] Set the bit in ICR for the corresponding pin when receiving data or using an external clock.

[2] Set the clock selection in SCR. Be sure to clear bits RIE, TIE, TEIE, and MPIE, and bits TE and RE, to 0.

When the clock output is selected in asynchronous mode, the clock is output immediately after SCR settings are made.

[3] Set the data transfer format in SMR and SCMR.

[4] Write a value corresponding to the bit rate to BRR. This step is not necessary if an external clock is used.

[5] Wait at least one bit interval, then set th 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.

Figure 14.5 Sample SCI Initialization Flowchart

transmission of the current transmit data has finished, continuous transmission can be

3. Data is sent from the TxD pin in the following order: start bit, transmit data, parity bit 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 next transmit data is transferred from TDR to TSR, the stop 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 t state is entered in which 1 is output. If the TEIE bit in SCR is set to 1 at this time, a T interrupt request is generated.

Figure 14.7 shows a sample flowchart for transmission in asynchronous mode.

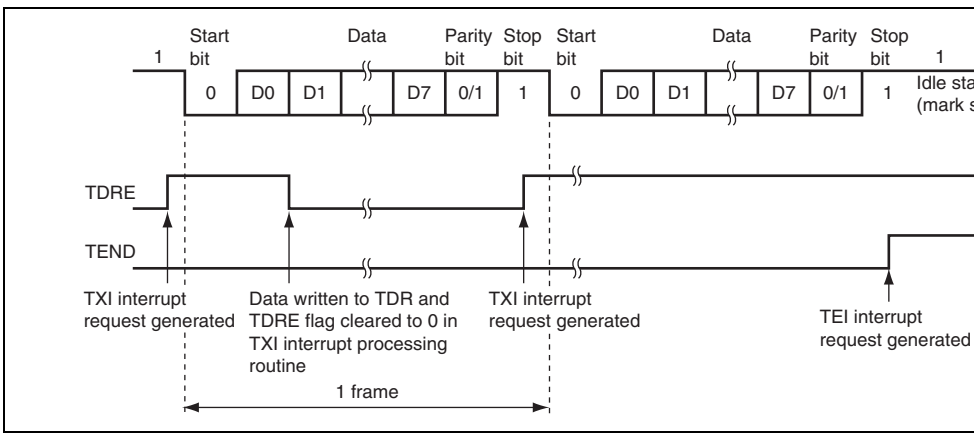
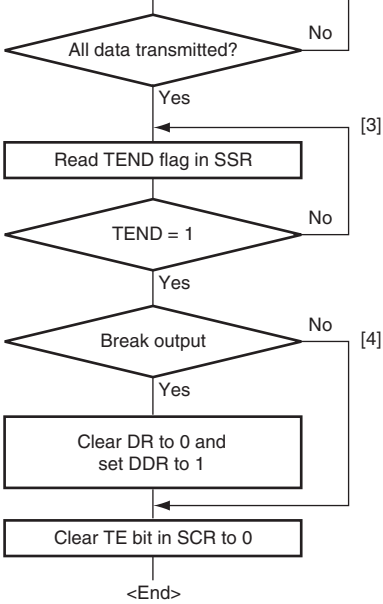


Figure 14.6 Example of Operation for Transmission in Asynchronous Mod (Example with 8-Bit Data, Parity, One Stop Bit)



To continue serial transmission, read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and clear the TDRE flag to 0. However, the TDRE flag is checked and cleared automatically when the DMAC or DTC is initiated by a transmit data empty interrupt (TXI) request and writes data to TDR.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, set DDR for the port corresponding to the TxD pin to 1, clear DR to 0, then clear the TE bit in SCR to 0.

Figure 14.7 Sample Serial Transmission Flowchart

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 transferr RDR. If the RIE bit in SCR is set to 1 at this time, an ERI interrupt request is generate
4. If a framing error (when the stop bit is 0) is detected, the FER bit in SSR is set to 1 an data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an ERI interru request is generated.
5. If reception finishes 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 requ generated. Because the RXI interrupt processing routine reads the receive data transfe RDR before reception of the next receive data has finished, continuous reception can enabled.

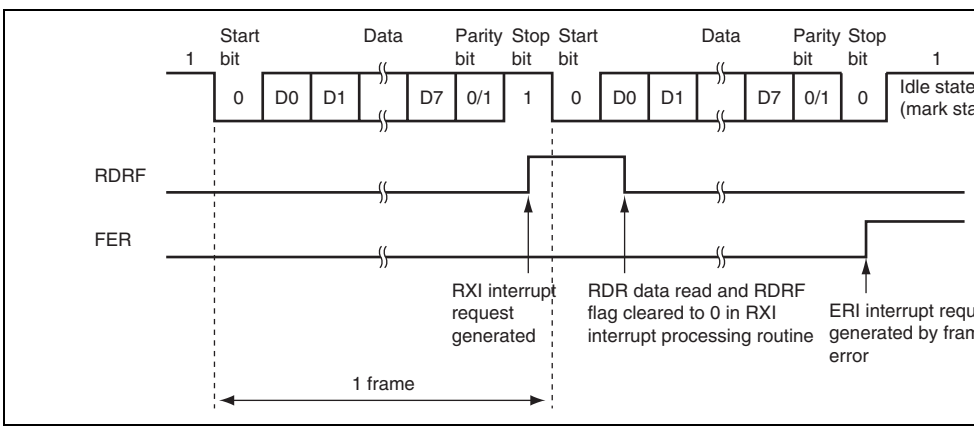
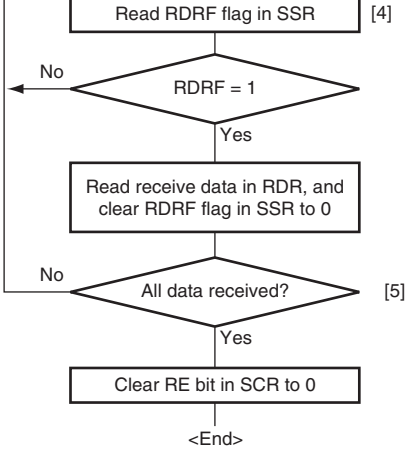


Figure 14.8 Example of SCI Operation for Reception (Example with 8-Bit Data, Parity, One Stop Bit)

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

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



value of the input port corresponding to the RxD pin.

[4] SCI state check and receive data read: Read SSR and check that RDRF = 1, then read the receive data in RDR and clear the RDRF flag to 0. Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial reception continuation procedure: To continue serial reception, before the stop bit for the current frame is received, read the RDRF flag and RDR, and clear the RDRF flag to 0. However, the RDRF flag is cleared automatically when the DMAC or DTC is initiated by an RXI interrupt and reads data from RDR.

Figure 14.9 Sample Serial Reception Flowchart (1)

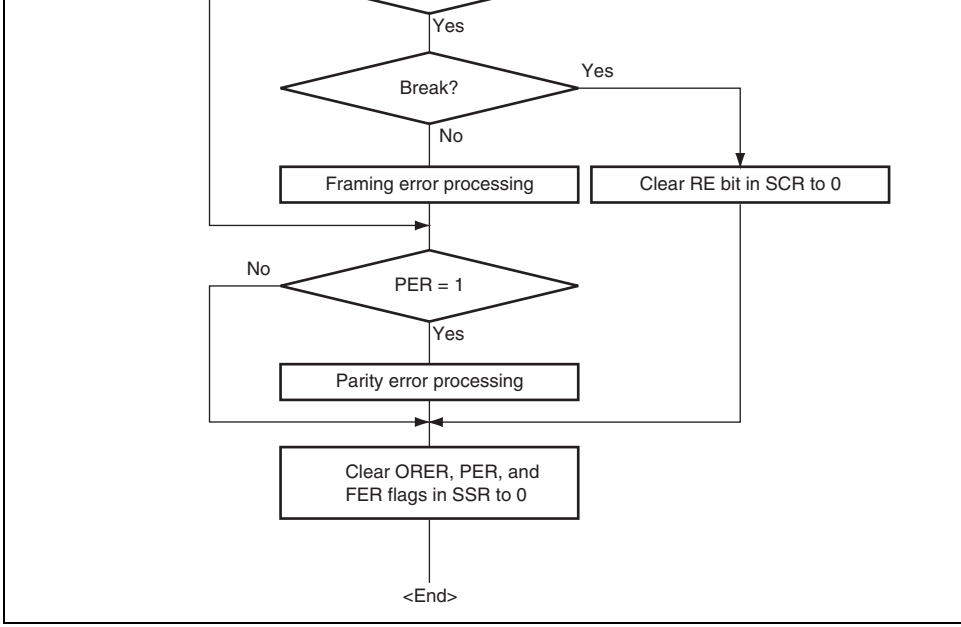


Figure 14.9 Sample Serial Reception Flowchart (2)

transmitting station first sends data which includes the ID code of the receiving station an multiprocessor bit set to 1. It then transmits transmit data added with a multiprocessor bit to 0. The receiving station skips data until data with a 1 multiprocessor bit is sent. When a 1 multiprocessor bit is received, the receiving station compares that data with its own ID station whose ID matches then receives the data sent next. Stations whose ID does not ma 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 t transfer of receive data from RSR to RDR, error flag detection, and setting the SSR status RDRF, FER, and ORER in SSR to 1 are prohibited until data with a 1 multiprocessor bit received. On reception of a receive character with a 1 multiprocessor bit, the MPB bit in S set to 1 and the MPIE bit is automatically cleared, thus normal reception is resumed. If th in SCR is set to 1 at this time, an RXI interrupt is generated.

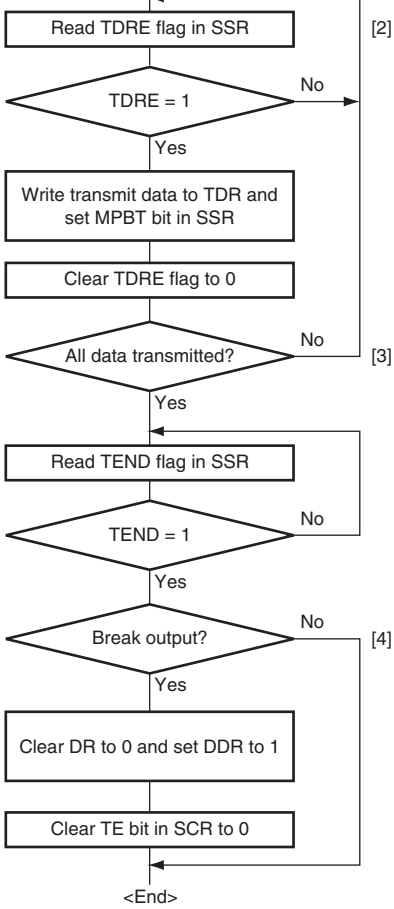
When the multiprocessor format is selected, the parity bit setting is invalid. All other bit s are the same as those in normal asynchronous mode. The clock used for multiprocessor communication is the same as that in normal asynchronous mode.

ID transmission cycle =
receiving station
specification

Data transmission cycle =
Data transmission to
receiving station specified by ID

[Legend]
MPB: Multiprocessor bit

Figure 14.10 Example of Communication Using Multiprocessor Format (Transmission of Data H'AA to Receiving Station A)



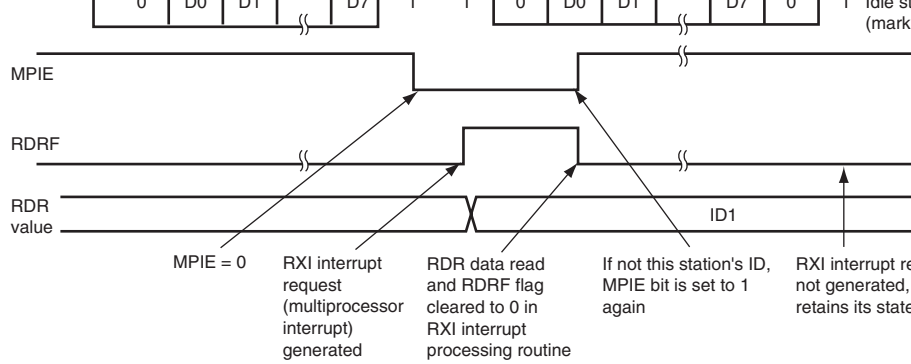
to 1, a 1 is output for one frame, and transmission is enabled.

[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.

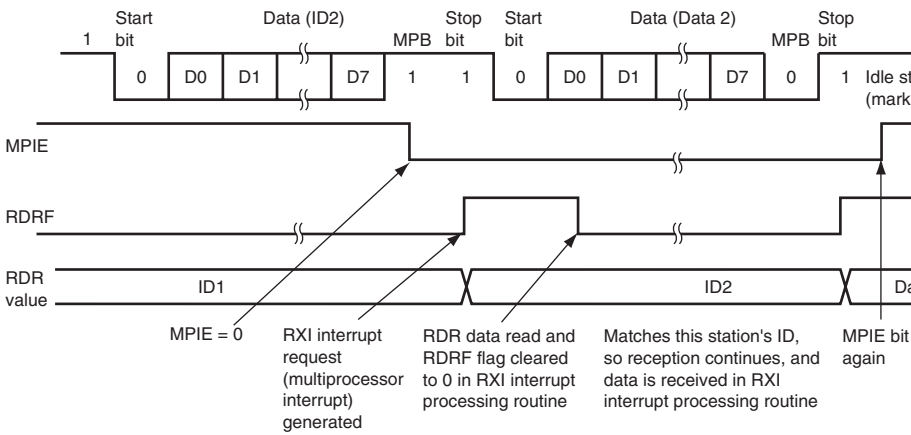
[3] Serial transmission continuation procedure:
To continue serial transmission, be sure to read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and then clear the TDRE flag to 0. However, the TDRE flag is checked and cleared automatically when the DMAC or DTC is initiated by a transmit data empty interrupt (TXI) request and writes data to TDR.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, set DDR for the port to 1, clear DR to 0, and then clear the TE bit in SCR to 0.

Figure 14.11 Sample Multiprocessor Serial Transmission Flowchart

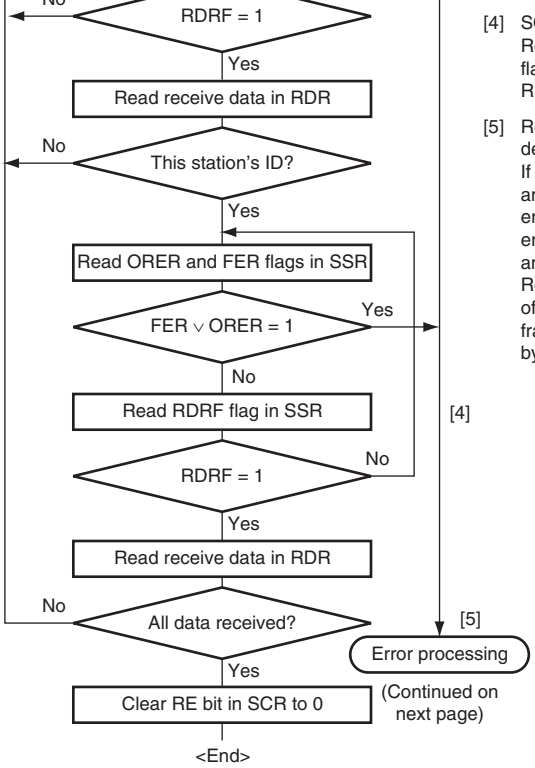


(a) Data does not match station's ID



(b) Data matches station's ID

Figure 14.12 Example of SCI Operation for Reception (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)



[4] SCI state check and data reception:
Read SSR and check that the RDRF flag is set to 1, then read the data in RDR.

[5] Receive error processing and break detection:
If a receive error occurs, read the ORER and FER flags in SSR to identify the error. After performing the appropriate error processing, ensure that the ORER and FER flags are both cleared to 0. Reception cannot be resumed if either of these flags is set to 1. In the case of a framing error, a break can be detected by reading the RxD pin value.

Figure 14.13 Sample Multiprocessor Serial Reception Flowchart (1)

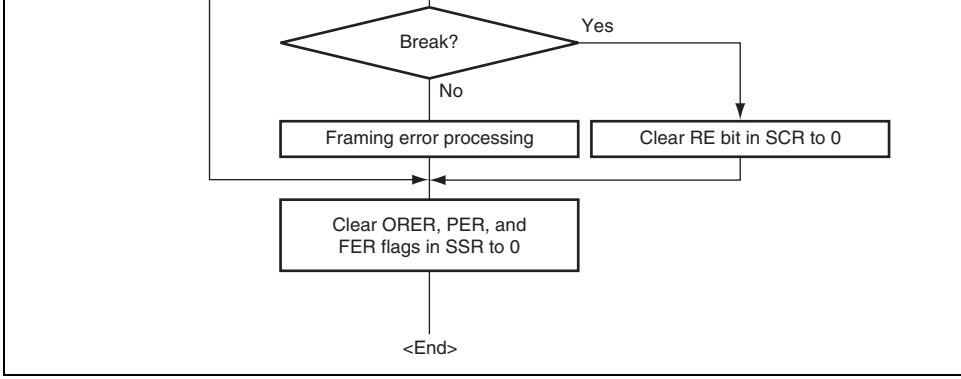
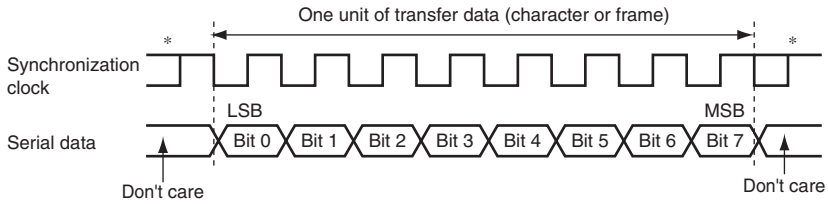


Figure 14.13 Sample Multiprocessor Serial Reception Flowchart (2)

transmission or the previous receive data can be read during reception, enabling continuo
transfer.

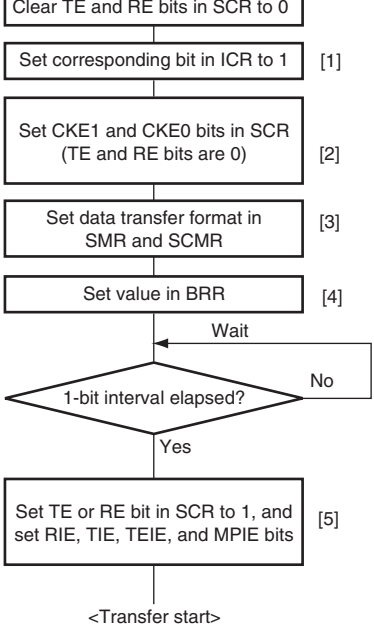


Note: * Holds a high level except during continuous transfer.

Figure 14.14 Data Format in Clocked Synchronous Communication (LSB-Fi

14.6.1 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 the and CKE0 bits in SCR. When the SCI is operated on an internal clock, the synchronizatio is output from the SCK pin. Eight synchronization clock pulses are output in the transfer character, and when no transfer is performed the clock is fixed high. Note that in the case reception only, the synchronization clock is output until an overrun error occurs or until t is cleared to 0.



pin when receiving data or using an external clock.

[2] Set the clock selection in SCR. Be sure to clear bits RIE, TIE, TEIE, and MPIE, and bits TE and RE, to 0.

[3] Set the data transfer format in SMR and SCMR.

[4] Write a value corresponding to the bit rate to BRR. This step is not necessary if an external clock is used.

[5] 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 14.15 Sample SCI Initialization Flowchart

transmission of the current transmit data has finished, continuous transmission can be

3. 8-bit data is sent from the TxD pin synchronized with the output clock when clock ou mode has been specified and synchronized with the input clock when use of an extern has been specified.
4. The SCI checks the TDRE flag at the timing for sending the last bit.
5. If the TDRE flag is cleared to 0, the next transmit data is transferred from TDR to TS 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 TxD pin retains output state of the last bit. If the TEIE bit in SCR is set to 1 at this time, a TEI interru is generated. The SCK pin is fixed high.

Figure 14.17 shows a sample flowchart for serial data transmission. Even if the TDRE fla cleared to 0, transmission will not start while a receive error flag (ORER, FER, or PER) i Make sure to clear the receive error flags to 0 before starting transmission. Note that clea RE bit to 0 does not clear the receive error flags.

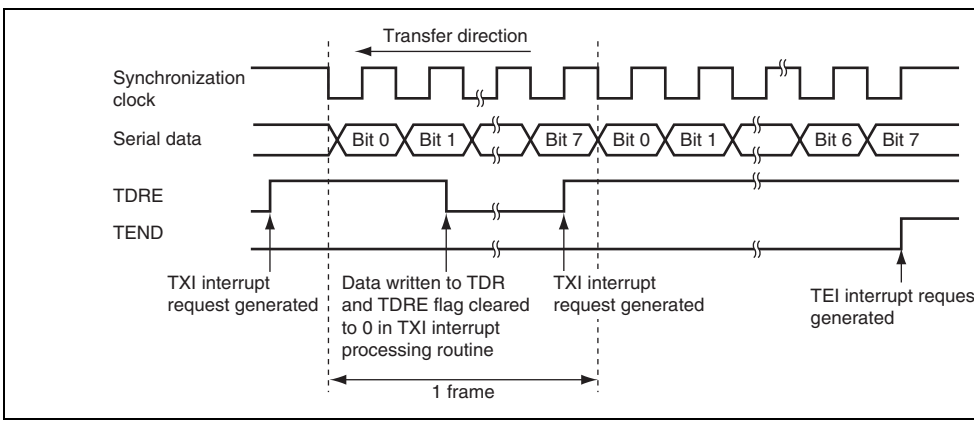
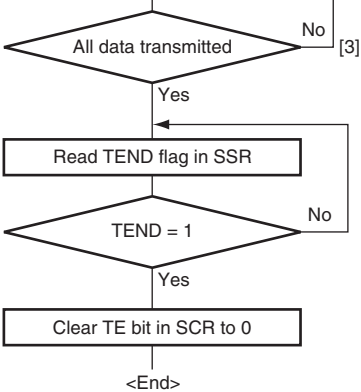


Figure 14.16 Example of Operation for Transmission in Clocked Synchronous



TDRE flag to 0. However, the TDRE flag is checked and cleared automatically when the DMAC or DTC is initiated by a transmit data empty interrupt (TXI) request and writes data to TDR.

Figure 14.17 Sample Serial Transmission Flowchart

flag remains to be set to 1.

3. If reception finishes 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 requ generated. Because the RXI interrupt processing routine reads the receive data transfe RDR before reception of the next receive data has finished, continuous reception can enabled.

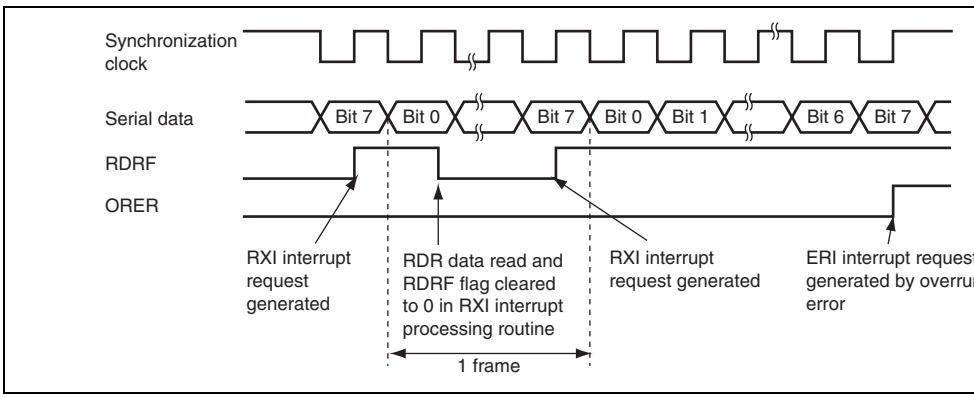
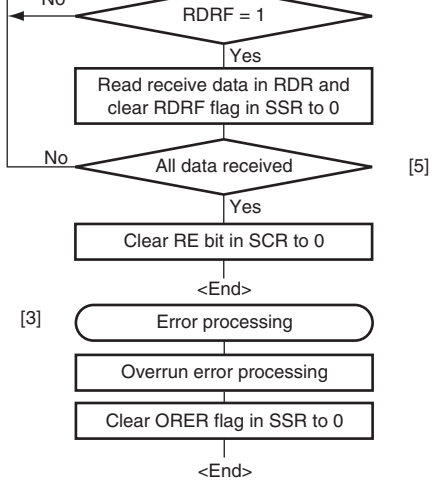


Figure 14.18 Example of Operation for Reception in Clocked Synchronous M

Transfer cannot be resumed while a receive error flag is set to 1. Accordingly, clear the C FER, PER, and RDRF bits to 0 before resuming reception. Figure 14.19 shows a sample for serial data reception.



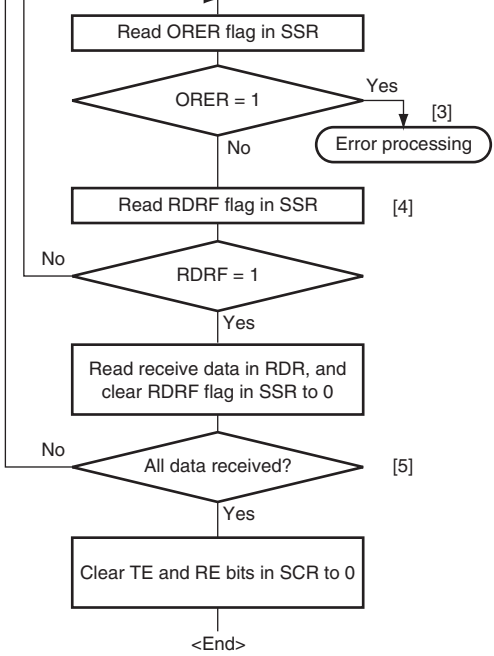
data in RDR and clear the RDRF flag to 0. Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial reception continuation procedure:
To continue serial reception, before the MSB (bit 7) of the current frame is received, reading the RDRF flag, reading RDR, and clearing the RDRF flag to 0 should be finished. However, the RDRF flag is cleared automatically when the DMAC or DTC is initiated by a receive data full interrupt (RXI) and reads data from RDR.

Figure 14.19 Sample Serial Reception Flowchart

14.6.5 Simultaneous Serial Data Transmission and Reception (Clocked Synchro Mode)

Figure 14.20 shows a sample flowchart for simultaneous serial transmit and receive ope After initializing the SCI, the following procedure should be used for simultaneous seria transmit and receive operations. To switch from transmit mode to simultaneous transmit receive mode, after checking that the SCI has finished transmission and the TDRE and T flags are set to 1, clear the TE bit to 0. Then simultaneously set both the TE and RE bits a single instruction. To switch from receive mode to simultaneous transmit and receive after checking that the SCI has finished reception, clear the RE bit to 0. Then after check the RDRF bit and receive error flags (ORER, FER, and PER) are cleared to 0, simultane both the TE and RE bits to 1 with a single instruction.



Read SSR and check that the RDRF flag is
then read the receive data in RDR and clea
RDRF flag to 0. Transition of the RDRF fla
to 1 can also be identified by an RXI interru

[5] Serial transmission/reception continuation
procedure:
To continue serial transmission/ reception,
the MSB (bit 7) of the current frame is rece
finish reading the RDRF flag, reading RDR
clearing the RDRF flag to 0. Also, before th
(bit 7) of the current frame is transmitted, r
from the TDRE flag to confirm that writing i
possible. Then write data to TDR and clea
TDRE flag to 0.
However, the TDRE flag is checked and cl
automatically when the DMAC or DTC is i
by a transmit data empty interrupt (TXI) re
writes data to TDR. Similarly, the RDRF fl
cleared automatically when the DMAC or D
initiated by a receive data full interrupt (RX
reads data from RDR.

Note: When switching from transmit or receive operation to simultaneous transmit and receive operations, first clear the TE bit and RE bit to 0, then set both these bits to 1 simultaneously.

Figure 14.20 Sample Flowchart of Simultaneous Serial Transmission and Rece

TxD and RxD pins and pull up the data transmission line to V_{CC} using a resistor. Setting
and TE bits to 1 with the IC card not connected enables closed transmission/reception al
self diagnosis. To supply the IC card with the clock pulses generated by the SCI, input t
pin output to the CLK pin of the IC card. A reset signal can be supplied via the output p
LSI.

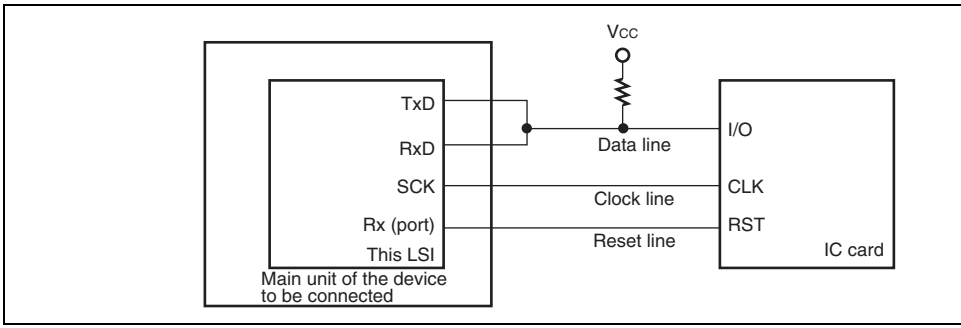


Figure 14.21 Pin Connection for Smart Card Interface

after at least 2 etu.

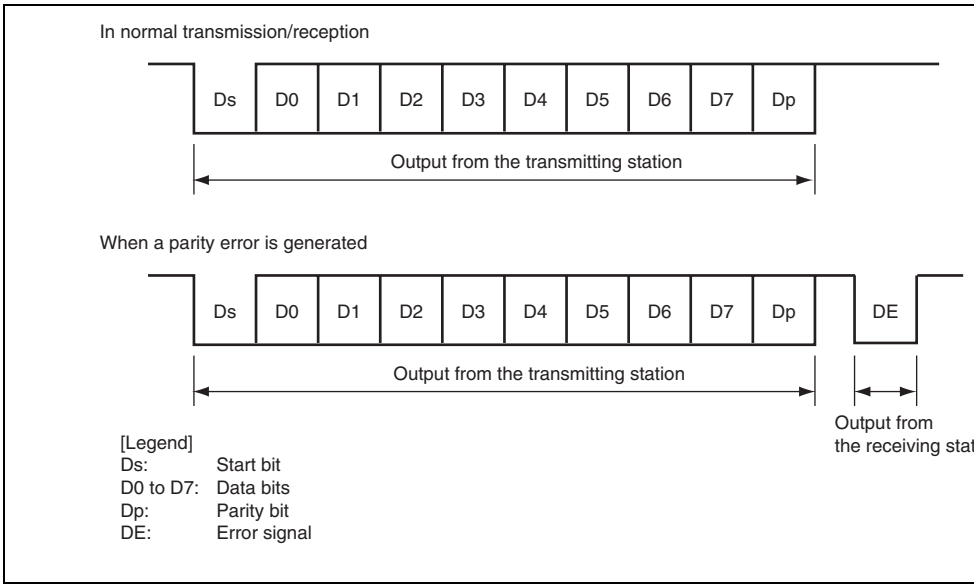


Figure 14.22 Data Formats in Normal Smart Card Interface Mode

For communication with the IC cards of the direct convention and inverse convention typ
follow the procedure below.

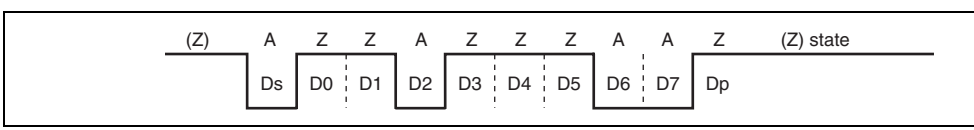


Figure 14.23 Direct Convention (SDIR = SINV = O/$\overline{E}$ = 0)

For the inverse convention type, logic levels 1 and 0 correspond to states A and Z, respe
and data is transferred with MSB-first as the start character, as shown in figure 14.24. Th
data in the start character in the figure is H'3F. When using the inverse convention type,
both the SDIR and SINV bits in SCMR. The parity bit is logic level 0 to produce even p
which is prescribed by the smart card standard, and corresponds to state Z. Since the SN
this LSI only inverts data bits D7 to D0, write 1 to the O/$\overline{E}$ bit in SMR to invert the parit
both transmission and reception.

14.7.3 Block Transfer Mode

Block transfer mode is different from normal smart card interface mode in the following

- Even if a parity error is detected during reception, no error signal is output. Since the
in SSR is set by error detection, clear the PER bit before receiving the parity bit of th
frame.
- During transmission, at least 1 etu is secured as a guard time after the end of the pari
before the start of the next frame.
- Since the same data is not re-transmitted during transmission, the TEND flag is set 1
after transmission start.
- Although the ERS flag in block transfer mode displays the error signal status as in no
smart card interface mode, the flag is always read as 0 because no error signal is tran

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

M: Reception margin (%)
N: Ratio of bit rate to clock (N = 32, 64, 372, 256)
D: Duty cycle of clock (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 marg
determined by the formula below.

$$M = \left(0.5 - \frac{1}{2 \times 372}\right) \times 100\% = 49.866\%$$

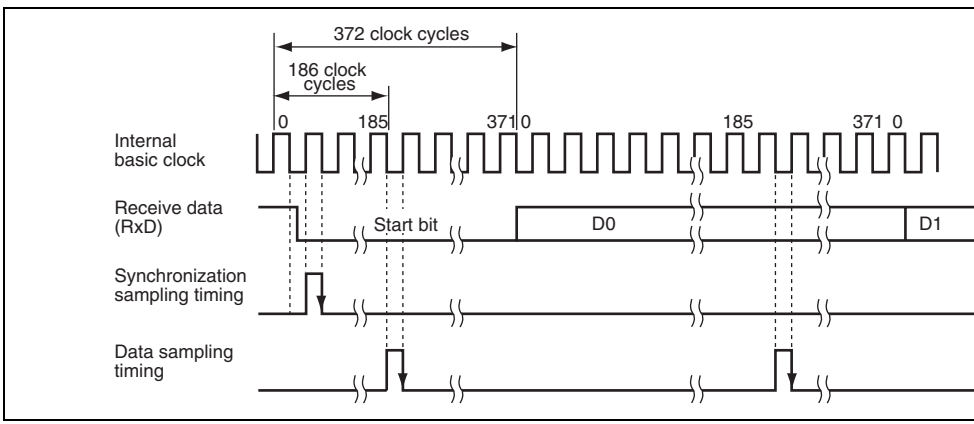


Figure 14.25 Receive Data Sampling Timing in Smart Card Interface Mod (When Clock Frequency is 372 Times the Bit Rate)

5. Set the SMIF, SDIR, and SINV bits in SCMR appropriately. When the DDR corresp
   the TxD pin is cleared to 0, the TxD and RxD pins are changed from port pins to SC
   placing the pins into high impedance state.
6. Set the value corresponding to the bit rate in BRR.
7. Set the CKE1 and CKE0 bits in SCR appropriately. Clear the TIE, RIE, TE, RE, MP
   TEIE bits to 0 simultaneously.

   When the CKE0 bit is set to 1, the SCK pin is allowed to output clock pulses.
8. Set the TIE, RIE, TE, and RE bits in SCR appropriately after waiting for at least a 1-
   interval. Setting the TE and RE bits to 1 simultaneously is prohibited except for self

To switch from reception to transmission, first verify that reception has completed, then
the SCI. At the end of initialization, RE and TE should be set to 0 and 1, respectively. R
completion can be verified by reading the RDRF, PER, or ORER flag. To switch from
transmission to reception, first verify that transmission has completed, then initialize the
the end of initialization, TE and RE should be set to 0 and 1, respectively. Transmission
completion can be verified by reading the TEND flag.

is transferred from TDR to TSR allowing automatic data retransmission.

3. If no error signal is returned from the receiving end, the ERS bit in SSR is not set to 1
4. In this case, one frame of data is determined to have been transmitted including re-tra the TEND bit in SSR is set to 1. Here, a TXI interrupt request is generated if the TIE b SCR is set to 1. Writing transmit data to TDR starts transmission of the next data.

Figure 14.28 shows a sample flowchart for transmission. All the processing steps are automatically performed using a TXI interrupt request to activate the DMAC or DTC. In transmission, the TEND and TDRE flags in SSR are simultaneously set to 1, thus generat TXI interrupt request if the TIE bit in SCR has been set to 1. This activates the DMAC or a TXI request thus allowing transfer of transmit data if the TXI interrupt request is specif source of DMAC or DTC activation beforehand. The TDRE and TEND flags are automat cleared to 0 at data transfer by the DMAC or DTC. If an error occurs, the SCI automatica transmits the same data. During re-transmission, the TEND flag remains as 0, thus not ac the DMAC or DTC. Therefore, the SCI and DMAC or DTC automatically transmit the sp number of bytes, including re-transmission in the case of error occurrence. However, the is not automatically cleared; the ERS flag must be cleared by previously setting the RIE b enable an ERI interrupt request to be generated at error occurrence.

When transmitting/receiving data using the DMAC or DTC, be sure to set and enable the or DTC prior to making SCI settings. For DMAC settings, see section 7, DMA Controller (DMAC), and for DTC settings, see section 8, Data Transfer Controller (DTC).

Figure 14.26 Data Re-Transfer Operation in SCI Transmission Mode

Note that the TEND flag is set in different timings depending on the GM bit setting in S
Figure 14.27 shows the TEND flag set timing.

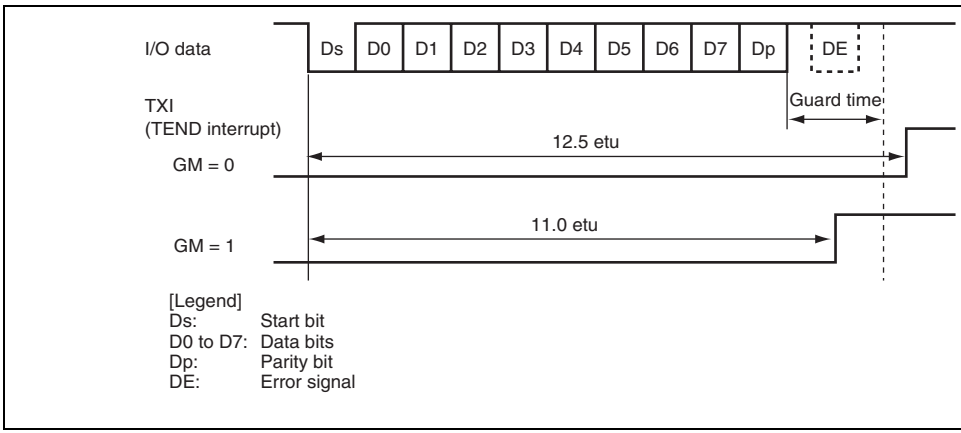


Figure 14.27 TEND Flag Set Timing during Transmission

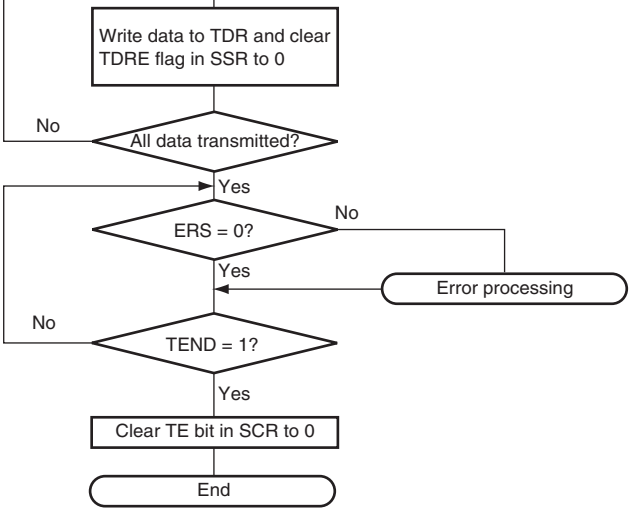


Figure 14.28 Sample Transmission Flowchart

4. In this case, data is determined to have been received successfully, and the RDRF bi
set to 1. Here, an RXI interrupt request is generated if the RIE bit in SCR is set to 1.

Figure 14.30 shows a sample flowchart for reception. All the processing steps are autom
performed using an RXI interrupt request to activate the DMAC or DTC. In reception, s
RIE bit to 1 allows an RXI interrupt request to be generated when the RDRF flag is set t
activates the DMAC or DTC by an RXI request thus allowing transfer of receive data if
interrupt request is specified as a source of DMAC or DTC activation beforehand. The R
is automatically cleared to 0 at data transfer by the DMAC or DTC. If an error occurs du
reception, i.e., either the ORER or PER flag is set to 1, a transmit/receive error interrupt
request is generated and the error flag must be cleared. If an error occurs, the DMAC or
not activated and receive data is skipped, therefore, the number of bytes of receive data
in the DMAC or DTC is transferred. Even if a parity error occurs and the PER bit is set
reception, receive data is transferred to RDR, thus allowing the data to be read.

Note: For operations in block transfer mode, see section 14.4, Operation in Asynchron

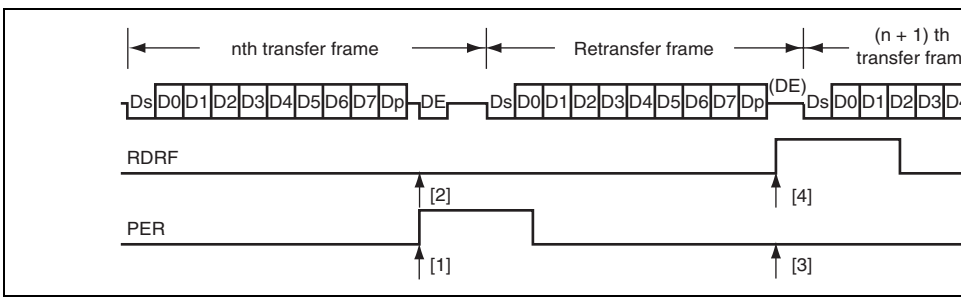


Figure 14.29 Data Re-Transfer Operation in SCI Reception Mode

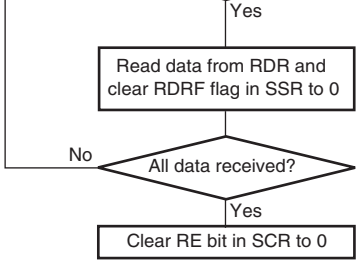


Figure 14.30 Sample Reception Flowchart

14.7.8 Clock Output Control

Clock output can be fixed using the CKE1 and CKE0 bits in SCR when the GM bit in SM
to 1. Specifically, the minimum width of a clock pulse can be specified.

Figure 14.31 shows an example of clock output fixing timing when the CKE0 bit is contr
with GM = 1 and CKE1 = 0.

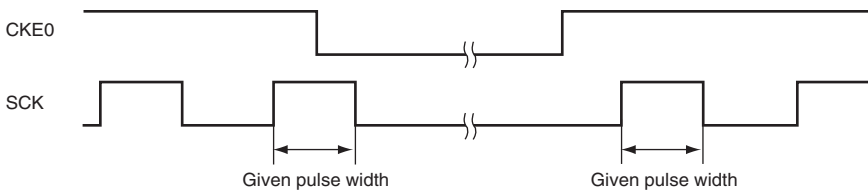


Figure 14.31 Clock Output Fixing Timing

At power-on and transitions to/from software standby mode, use the following procedure
the appropriate clock duty cycle.

1. Set the data register (DR) and data direction register (DDR) corresponding to
pin to the values for the output fixed state in software standby mode.
2. Write 0 to the TE and RE bits in SCR to stop transmission/reception. Simulta
set the CKE1 bit to the value for the output fixed state in software standby m
3. Write 0 to the CKE0 bit in SCR to stop the clock.
4. Wait for one cycle of the serial clock. In the mean time, the clock output is fi
specified level with the duty cycle retained.
5. Make the transition to software standby mode.

— At transition from smart card interface mode to software standby mode

1. Clear software standby mode.
2. Write 1 to the CKE0 bit in SCR to start clock output. A clock signal with the
appropriate duty cycle is then generated.

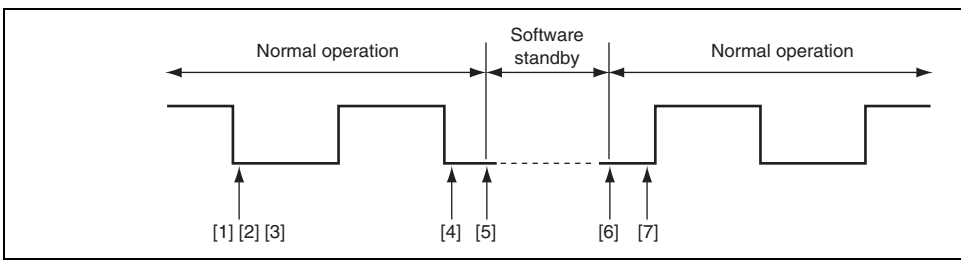


Figure 14.32 Clock Stop and Restart Procedure

DMAC or DTC to allow data transfer. The TDRE flag is automatically cleared to 0 at dat
by the DMAC or DTC.

When the RDRF flag in SSR is set to 1, an RXI interrupt request is generated. When the
PER, or FER flag in SSR is set to 1, an ERI interrupt request is generated. An RXI interru
activate the DMAC or DTC to allow data transfer. The RDRF flag is automatically cleare
data transfer by the DMAC or DTC.

A TEI interrupt is requested when the TEND flag is set to 1 while the TEIE bit is set to 1
interrupt and a TXI interrupt are requested simultaneously, the TXI interrupt has priority
acceptance. However, note that if the TDRE and TEND flags are cleared to 0 simultaneou
the TXI interrupt processing routine, the SCI cannot branch to the TEI interrupt processin
later.

Table 14.12 SCI Interrupt Sources

Name	Interrupt Source	Interrupt Flag	DTC Activation	DMAC Activation
ERI	Receive error	ORER, FER, or PER	Not possible	Not possible
RXI	Receive data full	RDRF	Possible	Possible
TXI	Transmit data empty	TDRE	Possible	Possible
TEI	Transmit end	TEND	Not possible	Not possible

RXI	Receive data full	RDRF	Possible	Possible
TXI	Transmit data empty	TDRE	Possible	Possible

Data transmission/reception using the DMAC or DTC is also possible in smart card inte
mode, similar to in the normal SCI mode. In transmission, the TEND and TDRE flags ir
simultaneously set to 1, thus generating a TXI interrupt. This activates the DMAC or DT
TXI request thus allowing transfer of transmit data if the TXI request is specified as a sc
DMAC or DTC activation beforehand. The TDRE and TEND flags are automatically cl
at data transfer by the DMAC or DTC. If an error occurs, the SCI automatically re-trans
same data. During re-transmission, the TEND flag remains as 0, thus not activating the I
DTC. Therefore, the SCI and DMAC or DTC automatically transmit the specified numb
bytes, including re-transmission in the case of error occurrence. However, the ERS flag
which is set at error occurrence, is not automatically cleared; the ERS flag must be clear
previously setting the RIE bit in SCR to 1 to enable an ERI interrupt request to be gener
error occurrence.

When transmitting/receiving data using the DMAC or DTC, be sure to set and enable th
or DTC prior to making SCI settings. For DMAC settings, see section 7, DMA Controll
(DMAC), and for DTC settings, see section 8, Data Transfer Controller (DTC).

In reception, an RXI interrupt request is generated when the RDRF flag in SSR is set to
activates the DMAC or DTC by an RXI request thus allowing transfer of receive data if
request is specified as a source of DMAC or DTC activation beforehand. The RDRF fla
automatically cleared to 0 at data transfer by the DMAC or DTC. If an error occurs, the
flag is not set but the error flag is set. Therefore, the DMAC or DTC is not activated and
interrupt request is issued to the CPU instead; the error flag must be cleared.

When framing error detection is performed, a break can be detected by reading the RxD p
directly. In a break, the input from the RxD pin becomes all 0s, and so the FER flag is set
PER flag may also be set. Note that, since the SCI continues the receive operation even af
receiving a break, even if the FER flag is cleared to 0, it will be set to 1 again.

14.9.3 Mark State and Break Detection

When the TE bit is 0, the TxD pin is used as an I/O port whose direction (input or output)
level are determined by DR and DDR. This can be used to set the TxD pin to mark state (
level) or send a break during serial data transmission. To maintain the communication lin
state (the state of 1) until TE is set to 1, set both DDR and DR to 1. Since the TE bit is cle
at this point, the TxD pin becomes an I/O port, and 1 is output from the TxD pin. To send
during serial transmission, first set DDR to 1 and DR to 0, and then clear the TE bit to 0.
TE bit is cleared to 0, the transmitter is initialized regardless of the current transmission s
TxD pin becomes an I/O port, and 0 is output from the TxD pin.

14.9.4 Receive Error Flags and Transmit Operations (Clocked Synchronous Mod

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

14.9.6 Restrictions on Using DMAC or DTC

1. When the external clock source is used as a synchronization clock, update TDR by th
or DTC and wait for at least five Pϕ clock cycles before allowing the transmit clock
input. If the transmit clock is input within four clock cycles after TDR modification,
may malfunction (figure 14.33).
2. When using the DMAC or DTC to read RDR, be sure to set the receive end interrupt
the DMAC or DTC activation source.

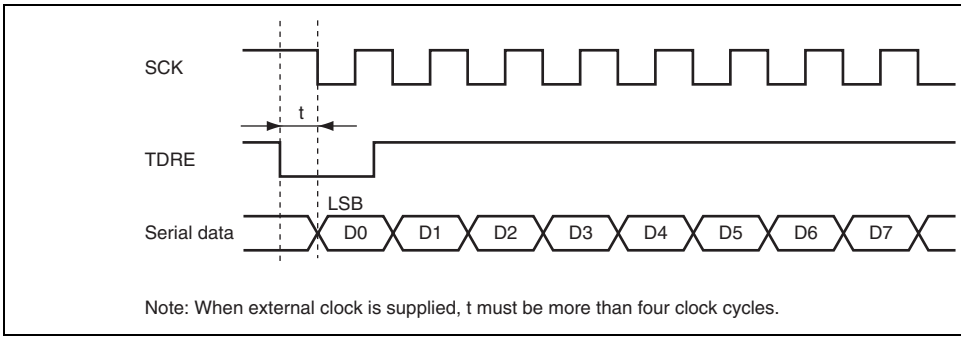


Figure 14.33 Sample Transmission using DTC in Clocked Synchronous Mo

TE bit to 1, read SSR, write to TDR, clear TDRE in this order, and then start transmission transmit data in a different transmission mode, initialize the SCI first.

Figure 14.34 shows a sample flowchart for transition to software standby mode during transmission. Figures 14.35 and 14.36 show the port pin states in transition to software st mode.

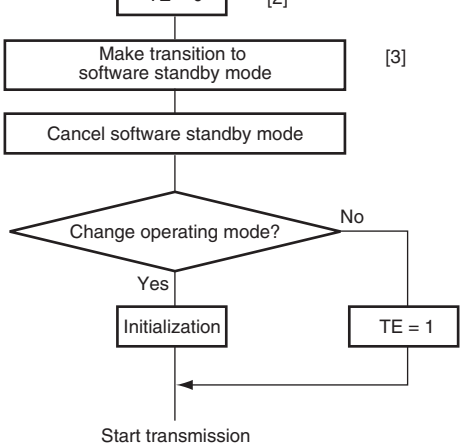
Before specifying the module stop state or making a transition to software standby mode transmission mode using DTC transfer, stop all transmit operations (TE = TIE = TEIE = Setting the TE and TIE bits to 1 after cancellation sets the TXI flag to start transmission u DTC.

(2) Reception

Before specifying the module stop state or making a transition to software standby mode, receive operations (RE = 0). RSR, RDR, and SSR are reset. If transition is made during d reception, the data being received will be invalid.

To receive data in the same reception mode after cancellation of the power-down state, se bit to 1, and then start reception. To receive data in a different reception mode, initialize t first.

Figure 14.37 shows a sample flowchart for transition to software standby mode during re



[2] Clear the TIE and TEIE bits to 0 when they are 1.

[3] Setting of the module stop state is included.

Figure 14.34 Sample Flowchart of Transition to Software Standby Mode in Tran

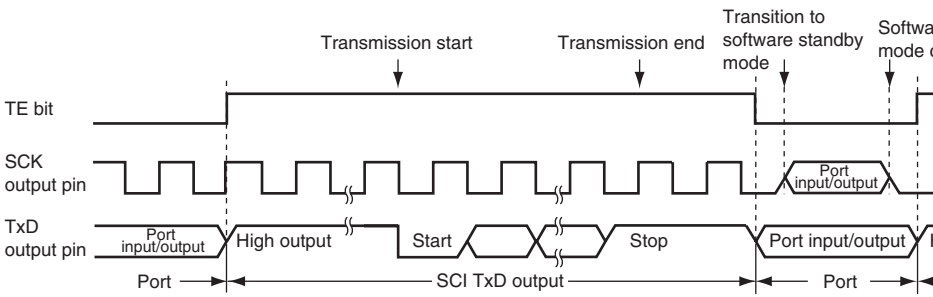


Figure 14.35 Port Pin States during Transition to Software Standby Mod (Internal Clock, Asynchronous Transmission)

Figure 14.36 Port Pin States during Transition to Software Standby Mode (Internal Clock, Clocked Synchronous Transmission)

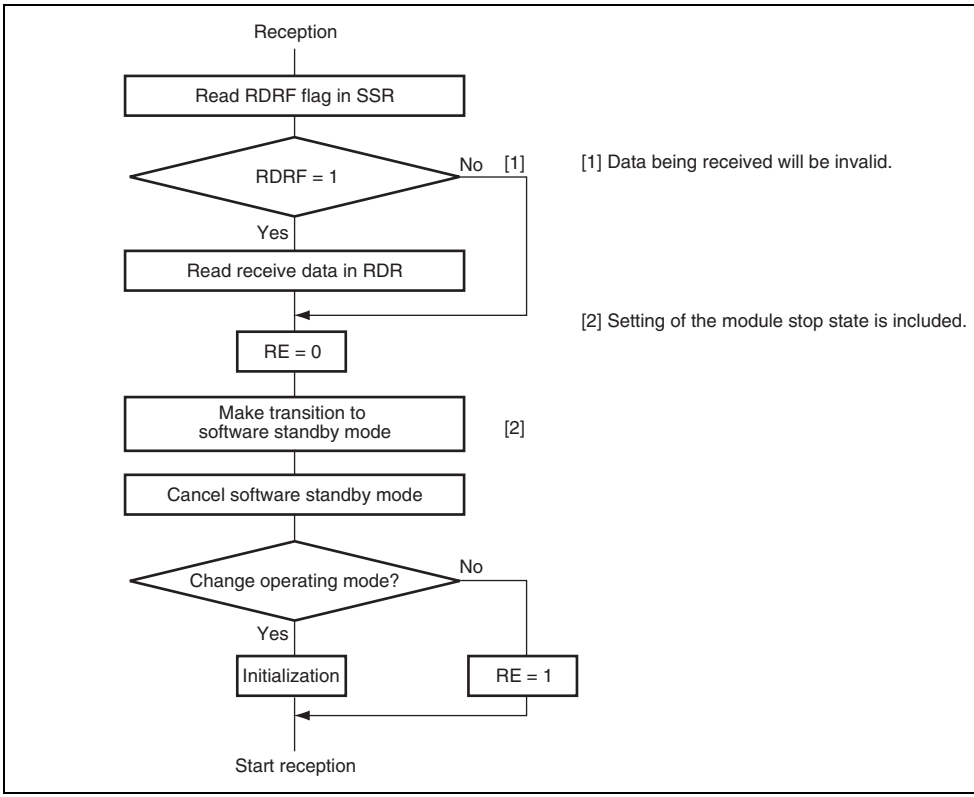


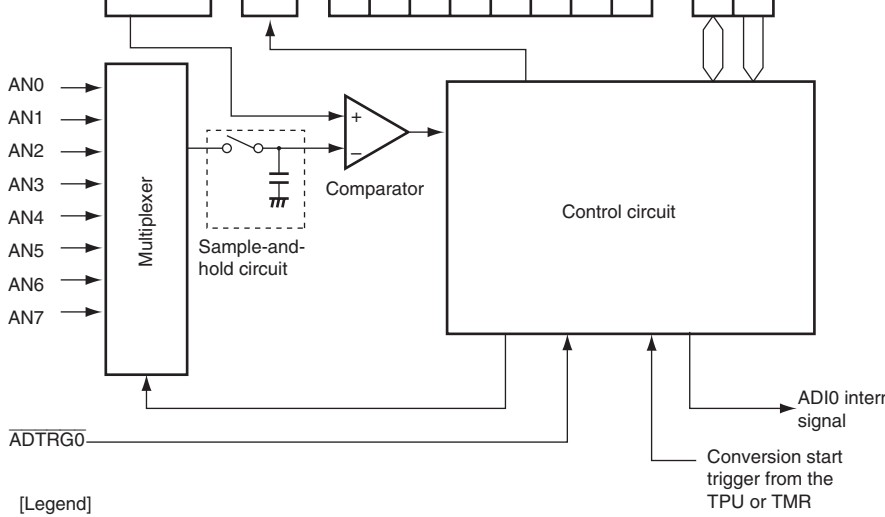
Figure 14.37 Sample Flowchart of Transition to Software Standby Mode in Rec

- Eight input channels
- Conversion time: 7.4 μs per channel (at 35-MHz operation)
- Two kinds of operating modes
  - Single mode: Single-channel A/D conversion
  - Scan mode: Continuous A/D conversion on 1 to 4 channels, or 1 to 8 channels
- Eight data registers

  A/D conversion results are held in a 16-bit data register for each channel
- Sample and hold function
- Three types of conversion start

  Conversion can be started by software, a conversion start trigger by the 16-bit timer (TPU) or 8-bit timer (TMR), or an external trigger signal.
- Interrupt source

  A/D conversion end interrupt (ADI) request can be generated.
- Module stop state specifiable



[Legend]
ADCR: A/D control register
ADCSR: A/D control/status register
ADDRA: A/D data register A
ADDRB: A/D data register B
ADDRC: A/D data register C
ADDRD: A/D data register D
ADDRE: A/D data register E
ADDRF: A/D data register F
ADDRG: A/D data register G
ADDRH: A/D data register H

Figure 15.1 Block Diagram of A/D Converter

Analog input pin 3	AN3	Input	
Analog input pin 4	AN4	Input	
Analog input pin 5	AN5	Input	
Analog input pin 6	AN6	Input	
Analog input pin 7	AN7	Input	
A/D external trigger input pin	$\overline{\text{ADTRG0}}$	Input	External trigger input for starting A/D c
Analog power supply pin	AV_{CC}	Input	Analog block power supply
Analog ground pin	AV_{SS}	Input	Analog block ground
Reference voltage pin	Vref	Input	A/D conversion reference voltage

15.3 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 data register E (ADDRE)
- A/D data register F (ADDRF)
- A/D data register G (ADDRG)
- A/D data register H (ADDRH)
- A/D control/status register (ADCSR)
- A/D control register (ADCR)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
Bit Name											—	—	—	—	—
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 15.2 Analog Input Channels and Corresponding ADDR Registers

Analog Input Channel	A/D Data Register Which Stores Conversion
AN0	ADDRA
AN1	ADDRB
AN2	ADDRC
AN3	ADDRD
AN4	ADDRE
AN5	ADDRF
AN6	ADDRG
AN7	ADDRH

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 conv [Setting conditions] • When A/D conversion ends in single mode • When A/D conversion ends on all specified in scan mode [Clearing conditions] • When 0 is written after reading ADF = 1 (When the CPU is used to clear this flag by while the corresponding interrupt is enablec to read the flag after writing 0 to it.) • When the DMAC or DTC is activated by an interrupt and ADDR is read
6	ADIE	0	R/W	A/D Interrupt Enable When this bit is set to 1, ADI interrupts by ADF enabled.
5	ADST	0	R/W	A/D Start Clearing this bit to 0 stops A/D conversion, and converter enters wait state. Setting this bit to 1 starts A/D conversion. In sir this bit is cleared to 0 automatically when A/D c on the specified channel ends. In scan mode, A conversion continues sequentially on the specif channels until this bit is cleared to 0 by a transit hardware standby mode.
4	—	0	R	Reserved This is a read-only bit and cannot be modified.

		0101: AN5 0110: AN6 0111: AN7 1XXX: Setting prohibited • When SCANE = 1 and SCANS = 0 0000: AN0 0001: AN0 and AN1 0010: AN0 to AN2 0011: AN0 to AN3 0100: AN4 0101: AN4 and AN5 0110: AN4 to AN6 0111: AN4 to AN7 1XXX: Setting prohibited • When SCANE = 1 and SCANS = 1 0000: AN0 0001: AN0 and AN1 0010: AN0 to AN2 0011: AN0 to AN3 0100: AN0 to AN4 0101: AN0 to AN5 0110: AN0 to AN6 0111: AN0 to AN7 1XXX: Setting prohibited

[Legend]
X: Don't care
Note: * Only 0 can be written to this bit, to clear the flag.

Bit	Bit Name	Value	R/W	Description
7	TRGS1	0	R/W	Timer Trigger Select 1 and 0
6	TRGS0	0	R/W	These bits select enabling or disabling of the st conversion by a trigger signal. 00: A/D conversion start by external trigger is d 01: A/D conversion start by external trigger fron enabled 10: A/D conversion start by external trigger fron enabled 11: A/D conversion start by the $\overline{\text{ADTRG0}}$ pin is
5	SCANE	0	R/W	Scan Mode
4	SCANS	0	R/W	These bits select the A/D conversion operating 0X: Single mode 10: Scan mode. A/D conversion is performed continuously for channels 1 to 4. 11: Scan mode. A/D conversion is performed continuously for channels 1 to 8.
3	CKS1	0	R/W	Clock Select 1 and 0
2	CKS0	0	R/W	These bits set the A/D conversion time. Set bits and CKS0 only while A/D conversion is stoppe 0). 00: A/D conversion time = 530 states (max) 01: A/D conversion time = 266 states (max) 10: A/D conversion time = 134 states (max) 11: A/D conversion time = 68 states (max)

15.4 Operation

The A/D converter operates by successive approximation with 10-bit resolution. It has tw operating modes: single mode and scan mode. When changing the operating mode or ana channel, to prevent incorrect operation, first clear the ADST bit in ADCSR to 0 to halt A/ conversion. The ADST bit can be set to 1 at the same time as the operating mode or analo channel is changed.

15.4.1 Single Mode

In single mode, A/D conversion is to be performed only once on the analog input of the s single channel.

1. A/D conversion for the selected channel is started when the ADST bit in ADCSR is s software or an external trigger input.
2. When A/D conversion is completed, the A/D conversion result is transferred to the corresponding A/D data register of the channel.
3. When A/D conversion is completed, the ADF bit in ADCSR is set to 1. If the ADIE b to 1 at this time, an ADI interrupt request is generated.
4. The ADST bit remains set to 1 during A/D conversion, and is automatically cleared to A/D conversion ends. The A/D converter enters wait state. If the ADST bit is cleared during A/D conversion, A/D conversion stops and the A/D converter enters wait state

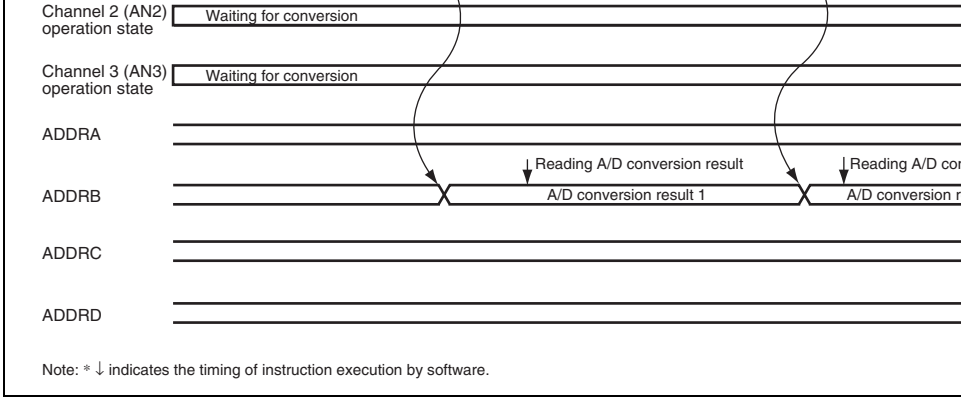
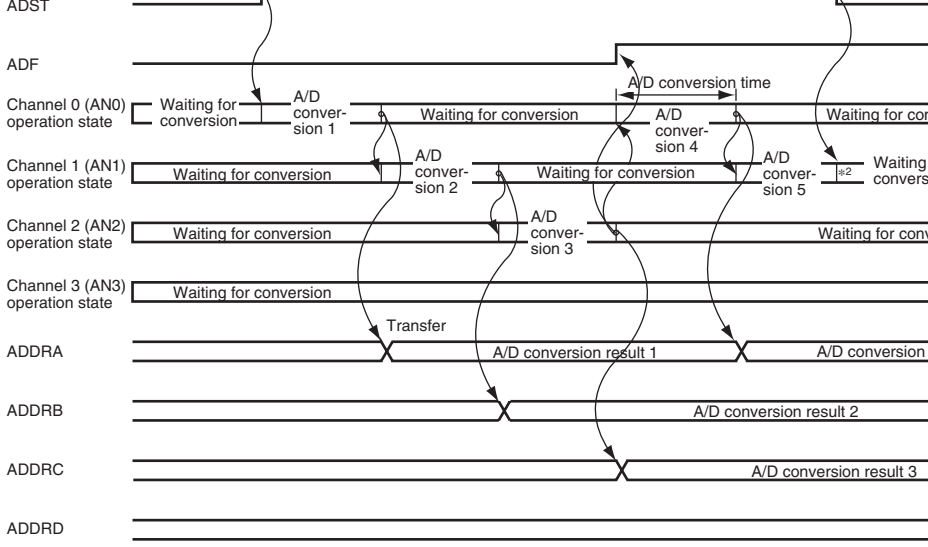


Figure 15.2 Example of A/D Converter Operation (Single Mode, Channel 1 Se

15.4.2 Scan Mode

In scan mode, A/D conversion is to be performed sequentially on the analog inputs of th channels up to four or eight channels.

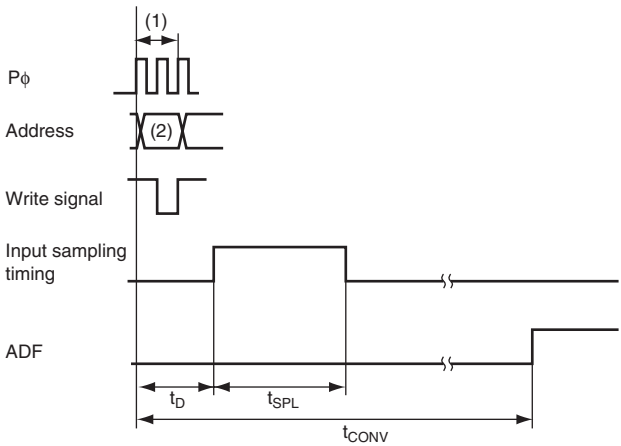
1. When the ADST bit in ADCSR is set to 1 by software, TPU, TMR, or an external tri input, A/D conversion starts on the first channel in the group. Consecutive A/D conv a maximum of four channels (SCANE and SCANS = B'10) or on a maximum of eig channels (SCANE and SCANS = B'11) can be selected. When consecutive A/D conv performed on four channels, A/D conversion starts on AN4 when CH3 and CH2 = B consecutive A/D conversion is performed on eight channels, A/D conversion starts o when CH3 = B'0.
2. When A/D conversion for each channel is completed, the A/D conversion result is se transferred to the corresponding ADDR of each channel.



Notes: 1. ↓ indicates the timing of instruction execution by software.
2. Data being converted is ignored.

Figure 15.3 Example of A/D Conversion (Scan Mode, Three Channels (AN0 to AN2) Selected)

In scan mode, the values given in table 15.3 apply to the first conversion time. The valu
table 15.4 apply to the second and subsequent conversions. In either case, bits CKS1 an
ADCR should be set so that the conversion time is within the ranges indicated by the A/
conversion characteristics.



[Legend]
(1): ADCSR write cycle
(2): ADCSR address
$t_{D:}$ A/D conversion start delay time
t_{SPL}: Input sampling time
t_{CONV}: A/D conversion time

Figure 15.4 A/D Conversion Timing

Note: Values in the table are the number of states.

Table 15.4 A/D Conversion Characteristics (Scan Mode)

CKS1	CKS0	Conversion Time (Number of States)
0	0	512 (Fixed)
	1	256 (Fixed)
1	0	128 (Fixed)
	1	64 (Fixed)

15.4.4 External Trigger Input Timing

A/D conversion can be externally triggered. When the TRGS1 and TRGS0 bits are set to
ADCR, an external trigger is input from the $\overline{\text{ADTRG0}}$ pin. A/D conversion starts when th
bit in ADCSR is set to 1 on the falling edge of the $\overline{\text{ADTRG0}}$ pin. Other operations, in bot
and scan modes, are the same as when the ADST bit has been set to 1 by software. Figure
shows the timing.

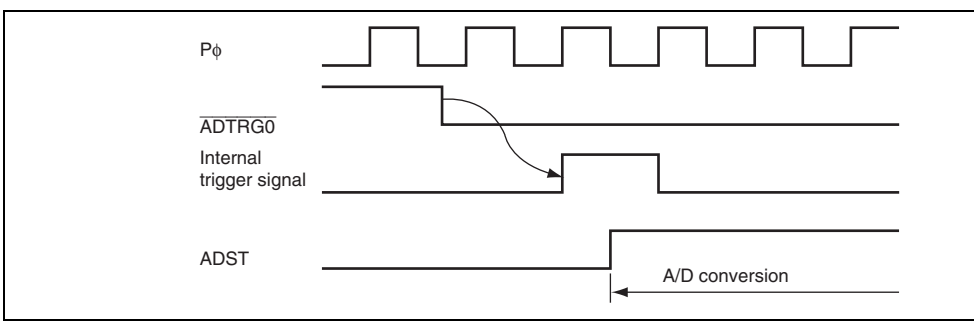


Figure 15.5 External Trigger Input Timing

ADI0	A/D conversion end	ADF	Possible

15.6 A/D Conversion Accuracy Definitions

This LSI's A/D conversion accuracy definitions are given below.

- Resolution

  The number of A/D converter digital output codes.
- Quantization error

  The deviation inherent in the A/D converter, given by 1/2 LSB (see figure 15.6).
- Offset error

  The deviation of the analog input voltage value from the ideal A/D conversion chara
  when the digital output changes from the minimum voltage value B'0000000000 (H'
  B'0000000001 (H'001) (see figure 15.7).
- Full-scale error

  The deviation of the analog input voltage value from the ideal A/D conversion chara
  when the digital output changes from B'1111111110 (H'3FE) to B'1111111111 (H'3
  figure 15.7).
- Nonlinearity error

  The error with respect to the ideal A/D conversion characteristic between the zero vo
  the full-scale voltage. Does not include the offset error, full-scale error, or quantizati
  (see figure 15.7).
- Absolute accuracy

  The deviation between the digital value and the analog input value. Includes the offs
  full-scale error, quantization error, and nonlinearity error.

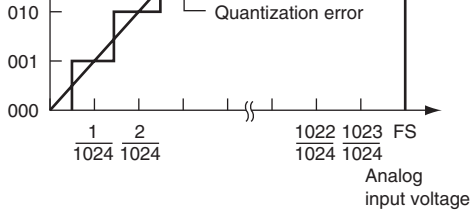


Figure 15.6 A/D Conversion Accuracy Definitions

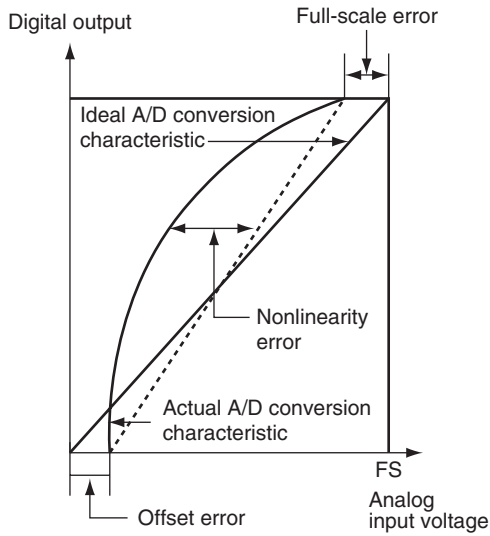


Figure 15.7 A/D Conversion Accuracy Definitions

This LSI's analog input is designed so that the conversion accuracy is guaranteed for an
signal for which the signal source impedance is 10 kΩ or less. This specification is prov
enable the A/D converter's sample-and-hold circuit input capacitance to be charged with
sampling time; if the sensor output impedance exceeds 10 kΩ, charging may be insuffic
may not be possible to guarantee the A/D conversion accuracy. However, if a large capa
provided externally for conversion in single mode, the input load will essentially compri
the internal input resistance of 10 kΩ, and the signal source impedance is ignored. Howe
a low-pass filter effect is obtained in this case, it may not be possible to follow an analog
with a large differential coefficient (e.g., 5 mV/μs or greater) (see figure 15.8). When co
high-speed analog signal or conversion in scan mode, a low-impedance buffer should be

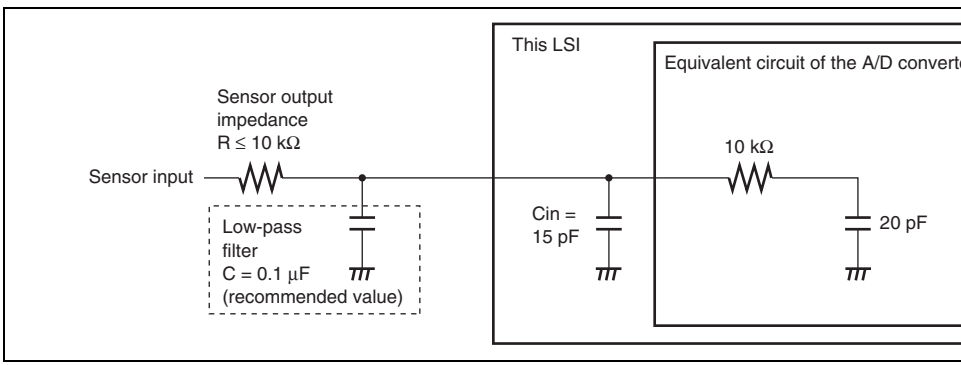


Figure 15.8 Example of Analog Input Circuit

If the conditions shown below are not met, the reliability of the LSI may be adversely aff

- Analog input voltage range

  The voltage applied to analog input pin ANn during A/D conversion should be in the
  AVss ≤ V_{AN} ≤ Vref.
- Relation between AVcc, AVss and Vcc, Vss

  As the relationship between AVcc, AVss and Vcc, Vss, set AVcc = Vcc ± 0.3 V and A
  Vss. If the A/D converter is not used, set AVcc = Vcc and AVss = Vss.
- Vref setting range

  The reference voltage at the Vref pin should be set in the range Vref ≤ AVcc.

15.7.5 Notes on Board Design

In board design, digital circuitry and analog circuitry should be as mutually isolated as po
and layout in which digital circuit signal lines and analog circuit signal lines cross or are
proximity should be avoided as far as possible. Failure to do so may result in incorrect op
of the analog circuitry due to inductance, adversely affecting A/D conversion values.

Digital circuitry must be isolated from the analog input pins (AN0 to AN7), analog refere
power supply (Vref), and analog power supply (AVcc) by the analog ground (AVss). Als
analog ground (AVss) should be connected at one point to a stable ground (Vss) on the b

input pin voltage. Careful consideration is therefore required when deciding the circuit c

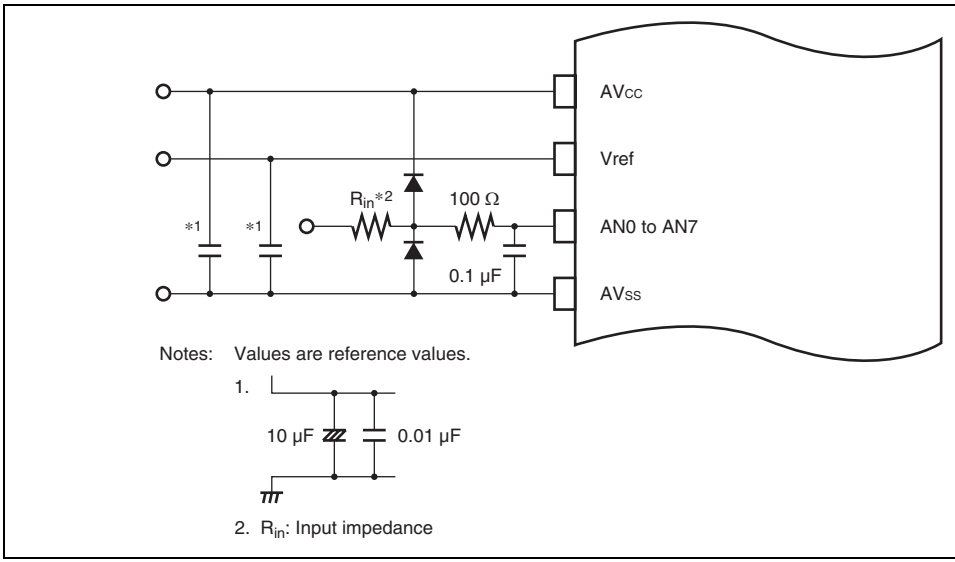


Figure 15.9 Example of Analog Input Protection Circuit

Table 15.6 Analog Pin Specifications

Item	Min	Max	Unit
Analog input capacitance	—	20	pF
Permissible signal source impedance	—	10	kΩ

When this LSI enters software standby mode with A/D conversion enabled, the analog in
retained, and the analog power supply current is equal to as during A/D conversion. If the
power supply current needs to be reduced in software standby mode, clear the ADST, TR
TRGS0 bits all to 0 to disable A/D conversion.

- Module stop state specifiable

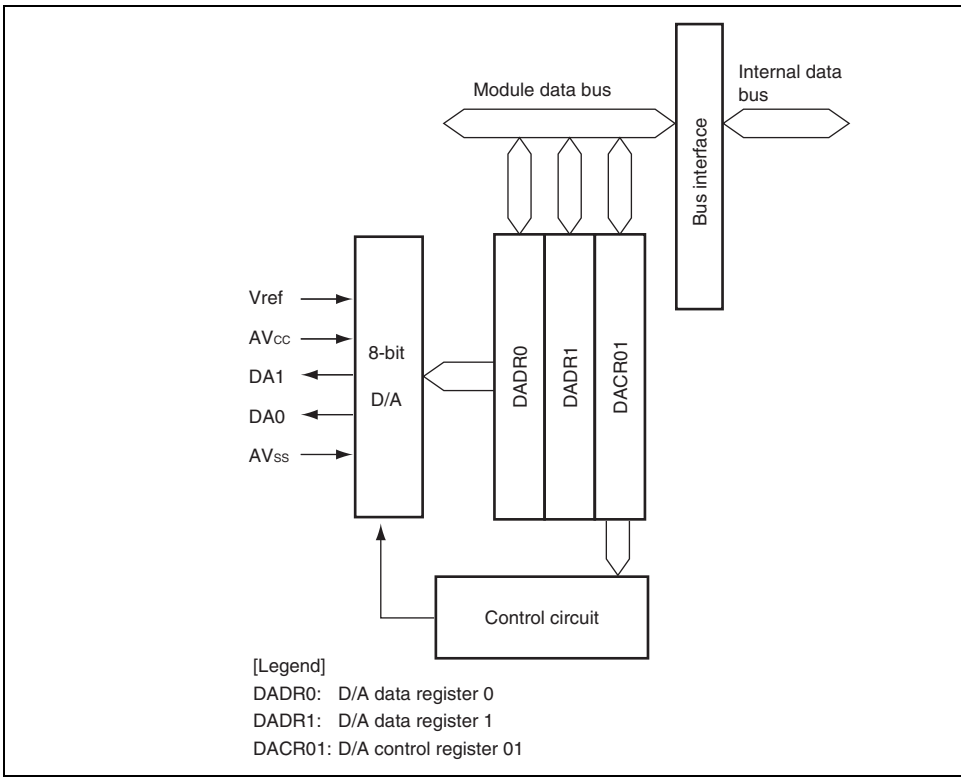


Figure 16.1 Block Diagram of D/A Converter

Analog output pin 0	DA0	Output	Channel 0 analog output
Analog output pin 1	DA1	Output	Channel 1 analog output

16.3 Register Descriptions

The D/A converter has the following registers.

- D/A data register 0 (DADR0)
- D/A data register 1 (DADR1)
- D/A control register 01 (DACR01)

16.3.1 D/A Data Registers 0 and 1 (DADR0 and DADR1)

DADR0 and DADR1 are 8-bit readable/writable registers that store data to which D/A co
is to be performed. Whenever an analog output is enabled, the values in DADR are conve
output to the analog output pins.

Bit	7	6	5	4	3	2	1	
Bit Name								
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Bit	Bit Name	Value	R/W	Description
7	DAOE1	0	R/W	D/A Output Enable 1 Controls D/A conversion and analog output. 0: Analog output of channel 1 (DA1) is disabled 1: D/A conversion of channel 1 is enabled. Ana of channel 1 (DA1) is enabled.
6	DAOE0	0	R/W	D/A Output Enable 0 Controls D/A conversion and analog output. 0: Analog output of channel 0 (DA0) is disabled 1: D/A conversion of channel 0 is enabled. Ana of channel 0 (DA0) is enabled.
5	DAE	0	R/W	D/A Enable Used together with the DAOE0 and DAOE1 bits D/A conversion. When this bit is cleared to 0, D conversion is controlled independently for chan 1. When this bit is set to 1, D/A conversion for c and 1 is controlled together. Output of conversion results is always controlle DAOE0 and DAOE1 bits. For details, see table Control of D/A Conversion.
4 to 0	—	All 1	R	Reserved These are read-only bits and cannot be modifie

			Analog output of channel 0 (DA0) is disabled and an output of channel 1 (DA1) is enabled.
		1	D/A conversion of channels 0 and 1 is enabled. Analog output of channels 0 and 1 (DA0 and DA1) is enabled.
1	0	0	D/A conversion of channels 0 and 1 is enabled. Analog output of channels 0 and 1 (DA0 and DA1) is disabled.
		1	D/A conversion of channels 0 and 1 is enabled. Analog output of channel 0 (DA0) is enabled and an output of channel 1 (DA1) is disabled.
	1	0	D/A conversion of channels 0 and 1 is enabled. Analog output of channel 0 (DA0) is disabled and an output of channel 1 (DA1) is enabled.
		1	D/A conversion of channels 0 and 1 is enabled. Analog output of channels 0 and 1 (DA0 and DA1) is enabled.

from the analog output pin DA0 after the conversion time t_{DCONV} has elapsed. The con
result continues to be output until DADR0 is written to again or the DAOE0 bit is cl
The output value is expressed by the following formula:

$$\text{Contents of DADR}/256 \times V_{ref}$$

3. If DADR0 is written to again, the conversion is immediately started. The conversion
output after the conversion time t_{DCONV} has elapsed.
4. If the DAOE0 bit is cleared to 0, analog output is disabled.

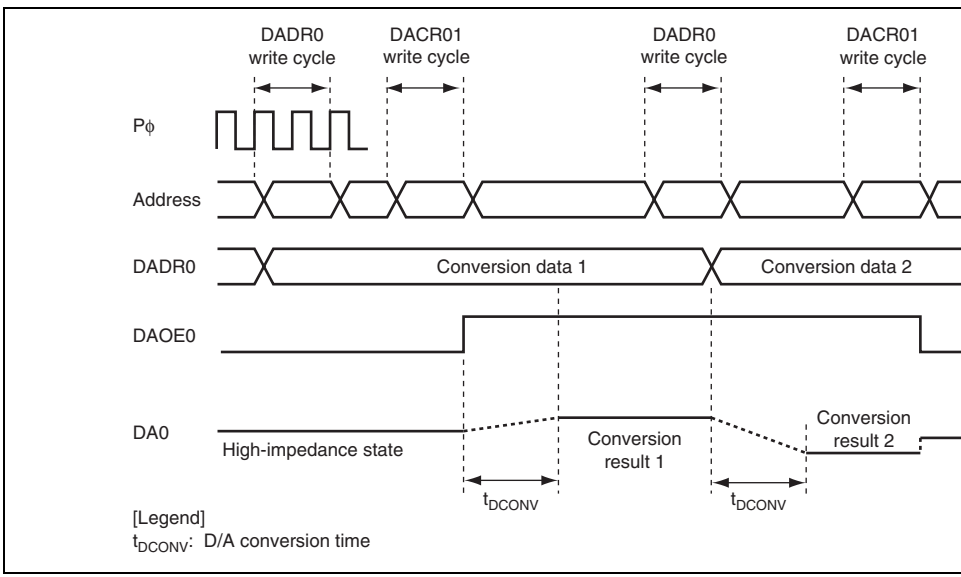


Figure 16.2 Example of D/A Converter Operation

When this LSI enters software standby mode with D/A conversion enabled, the D/A outp
retained, and the analog power supply current is equal to as during D/A conversion. If the
power supply current needs to be reduced in software standby mode, clear the ADST, TR
TRGS0 bits all to 0 to disable D/A conversion.

Flash memory version	H8SX/1657C	24 Kbytes	H'FF6000 to H'
	H8SX/1656C		

- Two memory MATs

  The start addresses of two memory spaces (memory MATs) are allocated to the sam
  The mode setting in the initiation determines which memory MAT is initiated first. T
  memory MATs can be switched by using the bank-switching method after initiation.

  — User MAT initiated at a power-on reset in user mode: 768 Kbytes/512 Kbytes

  — User boot MAT is initiated at a power-on reset in user boot mode: 8 Kbytes
- Programming/erasing interface by the download of on-chip program

  This LSI has a programming/erasing program. After downloading this program to th
  RAM, programming/erasing can be performed by setting the parameters.
- Programming/erasing time

  Programming time: 1 ms (typ) for 128-byte simultaneous programming, 8 μs per byt

  Erasing time: 750 ms (typ) per 1 block (64 Kbytes)
- Number of programming

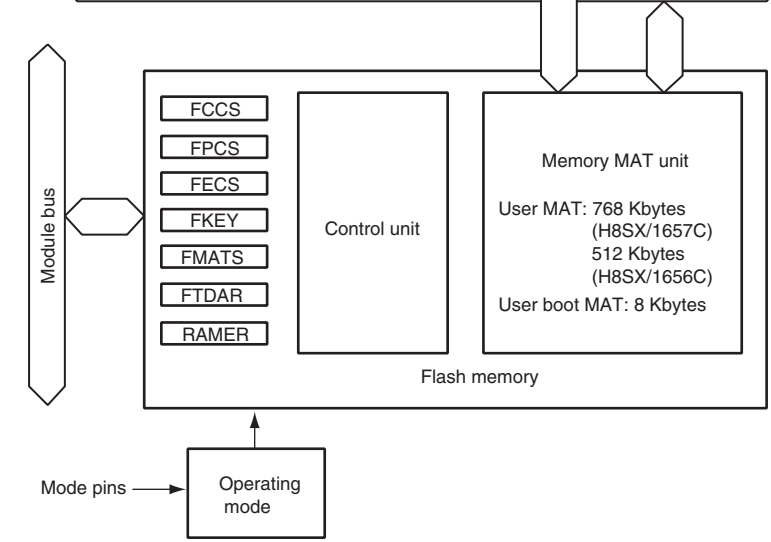
  The number of programming can be up to 100 times at the minimum. (1 to 100 times
  guaranteed.)
- Three on-board programming modes

  Boot mode: Using the on-chip SCI_4, the user MAT and user boot MAT can be
  programmed/erased. In boot mode, the bit rate between the host and this LSI can be
  automatically.

  User program mode: Using a desired interface, the user MAT can be programmed/er

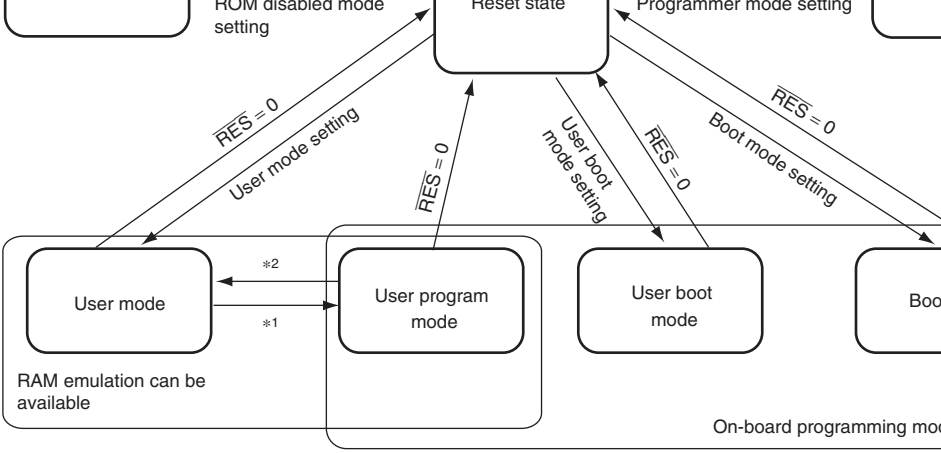
  User boot mode: Using a desired interface, the user boot program can be made and t
  MAT can be programmed/erased.
- Off-board programming mode

  Programmer mode: Using a PROM programmer, the user MAT and user boot MAT
  programmed/erased.



[Legend]
FCCS: Flash code control/status register
FPCS: Flash program code select register
FECS: Flash erase code select register
FKEY: Flash key code register
FMATS: Flash MAT select register
FTDAR: Flash transfer destination address register
RAMER: RAM emulation register

Figure 18.1 Block Diagram of Flash Memory



Notes: * In this LSI, the user program mode is defined as the period from the timing when a prog
concerning programming and erasure is started in user mode to the timing when the pro
completed.
1. Programming and erasure is started.
2. Programing and erasure is completed.

Figure 18.2 Mode Transition of Flash Memory

Programming/ erasing control	Command	Programming/ erasing interface	Programming/ erasing interface	Command
All erasure	O (Automatic)	O	O	O (Autom
Block division erasure	O*[1]	O	O	×
Program data transfer	From host via SCI	From desired device via RAM	From desired device via RAM	Via progra
RAM emulation	×	O	O	×
Reset initiation MAT	Embedded program storage area	User MAT	User boot MAT*[2]	—
Transition to user mode	Changing mode and reset	Completing Programming/ erasure*[3]	Changing mode and reset	—

Notes: 1. All-erasure is performed. After that, the specified block can be erased.

2. First, the reset vector is fetched from the embedded program storage area. Aft flash memory related registers are checked, the reset vector is fetched from th boot MAT.

3. In this LSI, the user programming mode is defined as the period from the timin program concerning programming and erasure is started to the timing when th program is completed. For details on a program concerning programming and see section 18.8.2, User Program Mode.

the size of the 8-kbyte user boot MAT should not be accessed. If an attempt is made, dat
as an undefined value.

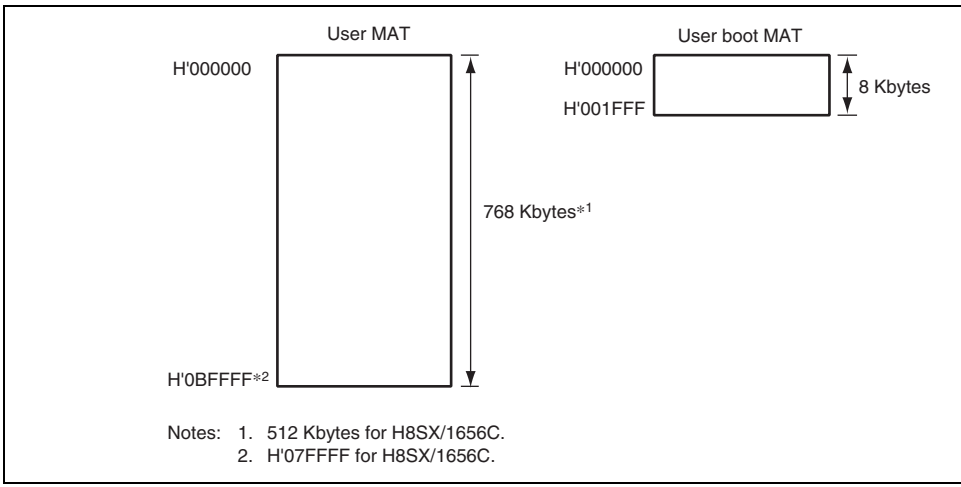


Figure 18.3 Memory MAT Configuration (H8SX/1657C)

EB0 Erase unit: 4 Kbytes	H'000000	H'000001	H'000002	←Programming unit: 128 bytes→	H'0000
	H'000F80	H'000F81	H'000F82	– – – – – – – – – – – –	H'000F
EB1 Erase unit: 4 Kbytes	H'001000	H'001001	H'001002	←Programming unit: 128 bytes→	H'0010
	H'001F80	H'001F81	H'001F82	– – – – – – – – – – – –	H'001F
EB2 Erase unit: 4 Kbytes	H'002000	H'002001	H'002002	←Programming unit: 128 bytes→	H'0020
	H'002F80	H'002F81	H'002F82	– – – – – – – – – – – –	H'002F
EB3 Erase unit: 4 Kbytes	H'003000	H'003001	H'003002	←Programming unit: 128 bytes→	H'0030
	H'003F80	H'003F81	H'003F82	– – – – – – – – – – – –	H'003F
EB4 Erase unit: 4 Kbytes	H'004000	H'004001	H'004002	←Programming unit: 128 bytes→	H'0040
	H'004F80	H'004F81	H'004F82	– – – – – – – – – – – –	H'004F
EB5 Erase unit: 4 Kbytes	H'005000	H'005001	H'005002	←Programming unit: 128 bytes→	H'0050
	H'005F80	H'005F81	H'005F82	– – – – – – – – – – – –	H'005F
EB6 Erase unit: 4 Kbytes	H'006000	H'006001	H'006002	←Programming unit: 128 bytes→	H'0060
	H'006F80	H'006F81	H'006F82	– – – – – – – – – – – –	H'006F
EB7 Erase unit: 4 Kbytes	H'007000	H'007001	H'007002	←Programming unit: 128 bytes→	H'0070
	H'007F80	H'007F81	H'007F82	– – – – – – – – – – – –	H'007F
EB8 Erase unit: 32 Kbytes	H'008000	H'008001	H'008002	←Programming unit: 128 bytes→	H'0080
	H'00FF80	H'00FF81	H'00FF82	– – – – – – – – – – – –	H'00FF
EB9 Erase unit: 64 Kbytes	H'010000	H'010001	H'010002	←Programming unit: 128 bytes→	H'0100
	H'01FF80	H'01FF81	H'01FF82	– – – – – – – – – – – –	H'01FF
EB10	H'020000	H'020001	H'020002	←Programming unit: 128 bytes→	H'0200
	H'0AFF80	H'0AFF81	H'0AFF82	– – – – – – – – – – – –	H'0AFF
EB19 Erase unit: 64 Kbytes	H'0B0000	H'0B0001	H'0B0002	←Programming unit: 128 bytes→	H'0B00
	H'0BFF80	H'0BFF81	H'0BFF82	– – – – – – – – – – – –	H'0BFF

Figure 18.4 User MAT Block Structure of H8SX/1657C (1)

Erase unit: 4 Kbytes	H'000F80	H'000F81	H'000F82	– – – – – – – – – – – –	H'000F
EB1 Erase unit: 4 Kbytes	H'001000	H'001001	H'001002	←Programming unit: 128 bytes→	H'001
	H'001F80	H'001F81	H'001F82	– – – – – – – – – – – –	H'001F
EB2 Erase unit: 4 Kbytes	H'002000	H'002001	H'002002	←Programming unit: 128 bytes→	H'002
	H'002F80	H'002F81	H'002F82	– – – – – – – – – – – –	H'002F
EB3 Erase unit: 4 Kbytes	H'003000	H'003001	H'003002	←Programming unit: 128 bytes→	H'003
	H'003F80	H'003F81	H'003F82	– – – – – – – – – – – –	H'003F
EB4 Erase unit: 4 Kbytes	H'004000	H'004001	H'004002	←Programming unit: 128 bytes→	H'004
	H'004F80	H'004F81	H'004F82	– – – – – – – – – – – –	H'004F
EB5 Erase unit: 4 Kbytes	H'005000	H'005001	H'005002	←Programming unit: 128 bytes→	H'005
	H'005F80	H'005F81	H'005F82	– – – – – – – – – – – –	H'005F
EB6 Erase unit: 4 Kbytes	H'006000	H'006001	H'006002	←Programming unit: 128 bytes→	H'006
	H'006F80	H'006F81	H'006F82	– – – – – – – – – – – –	H'006F
EB7 Erase unit: 4 Kbytes	H'007000	H'007001	H'007002	←Programming unit: 128 bytes→	H'007
	H'007F80	H'007F81	H'007F82	– – – – – – – – – – – –	H'007F
EB8 Erase unit: 32 Kbytes	H'008000	H'008001	H'008002	←Programming unit: 128 bytes→	H'008
	H'00FF80	H'00FF81	H'00FF82	– – – – – – – – – – – –	H'00FF
EB9 Erase unit: 64 Kbytes	H'010000	H'010001	H'010002	←Programming unit: 128 bytes→	H'010
	H'01FF80	H'01FF81	H'01FF82	– – – – – – – – – – – –	H'01FF
EB10	H'020000	H'020001	H'020002	←Programming unit: 128 bytes→	H'020
	H'0AFF80	H'0AFF81	H'0AFF82	– – – – – – – – – – – –	H'0AFF
EB15 Erase unit: 64 Kbytes	H'070000	H'070001	H'070002	←Programming unit: 128 bytes→	H'070
	H'07FF80	H'07FF81	H'07FF82	– – – – – – – – – – – –	H'07FF

Figure 18.4 User MAT Block Structure of H8SX/1656C (2)

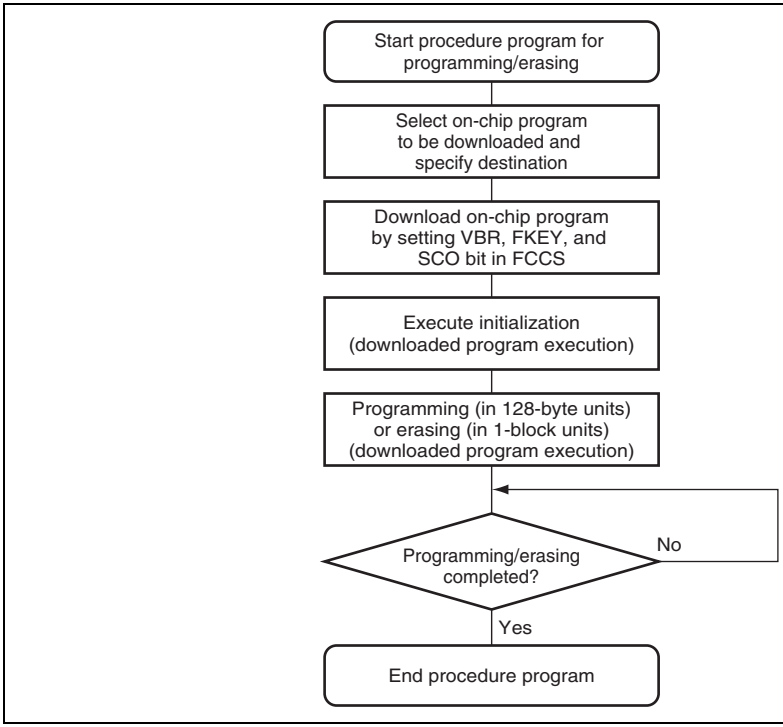


Figure 18.5 Procedure for Creating Procedure Program

(1) Selection of On-Chip Program to be Downloaded

This LSI has programming/erasing programs which can be downloaded to the on-chip RA
on-chip program to be downloaded is selected by the programming/erasing interface regis
start address of the on-chip RAM where an on-chip program is downloaded is specified b
flash transfer destination address register (FTDAR).

(3) Initialization of Programming/Erasing

A pulse with the specified period must be applied when programming or erasing. The sp
pulse width is made by the method in which wait loop is configured by the CPU instruct
Accordingly, the operating frequency of the CPU needs to be set before programming/er
operating frequency of the CPU is set by the programming/erasing interface parameter.

(4) Execution of Programming/Erasing

The start address of the programming destination and the program data are specified in 1
units when programming. The block to be erased is specified with the erase block numb
erase-block units when erasing. Specifications of the start address of the programming d
program data, and erase block number are performed by the programming/erasing interf
parameters, and the on-chip program is initiated. The on-chip program is executed by us
JSR or BSR instruction and executing the subroutine call of the specified address in the
RAM. The execution result is returned to the programming/erasing interface parameter.

The area to be programmed must be erased in advance when programming flash memor
interrupts are disabled during programming/erasing.

(5) When Programming/Erasing is Executed Consecutively

When processing does not end by 128-byte programming or 1-block erasure, consecutiv
programming/erasing can be realized by updating the start address of the programming d
and program data, or the erase block number. Since the downloaded on-chip program is
on-chip RAM even after programming/erasing completes, download and initialization ar
required when the same processing is executed consecutively.

RxD4	Input	Serial receive data input (used in boot mo

18.7 Register Descriptions

The flash memory has the following registers.

Programming/Erasing Interface Registers:

- Flash code control/status register (FCCS)
- Flash program code select register (FPCS)
- Flash erase code select register (FECS)
- Flash key code register (FKEY)
- Flash MAT select register (FMATS)
- Flash transfer destination address register (FTDAR)

Programming/Erasing Interface Parameters:

- Download pass and fail result parameter (DPFR)
- Flash pass and fail result parameter (FPFR)
- Flash program/erase frequency parameter (FPEFEQ)
- Flash multipurpose address area parameter (FMPAR)
- Flash multipurpose data destination area parameter (FMPDR)
- Flash erase block select parameter (FEBS)

- RAM emulation register (RAMER)

There are several operating modes for accessing the flash memory. Respective operating
registers, and parameters are assigned to the user MAT and user boot MAT. The correspo
between operating modes and registers/parameters for use is shown in table 18.3.

Programming/ erasing interface parameters	DPFR	O	—	—	—	—	–
	FPFR	—	O	O	O	—	–
	FPEFEQ	—	O	—	—	—	–
	FMPAR	—	—	O	—	—	–
	FMPDR	—	—	O	—	—	–
	FEBS	—	—	—	O	—	–
RAM emulation	RAMER	—	—	—	—	—	O

Notes: 1. The setting is required when programming or erasing the user MAT in user b
2. The setting may be required according to the combination of initiation mode a target memory MAT.

18.7.1 Programming/Erasing Interface Registers

The programming/erasing interface registers are 8-bit registers that can be accessed only These registers are initialized by a power-on reset.

(1) Flash Code Control/Status Register (FCCS)

FCCS monitors errors during programming/erasing the flash memory and requests the o program to be downloaded to the on-chip RAM.

Bit	7	6	5	4	3	2	1	
Bit Name	—	—	—	FLER	—	—	—	
Initial Value	1	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	

				flash memory, the reset must be released after input period (period of $\overline{RES}$ = 0) of at least 100 0: Flash memory operates normally (Error pro invalid) [Clearing condition] • At a power-on reset 1: An error occurs during programming/erasin memory (Error protection is valid) [Setting conditions] • When an interrupt, such as NMI, occurs dur programming/erasing. • When the flash memory is read during programming/erasing (including a vector rea an instruction fetch). • When the SLEEP instruction is executed du programming/erasing (including software st mode). • When a bus master other than the CPU, su DMAC and DTC, obtains bus mastership du programming/erasing.
3 to 1	—	All 0	R	Reserved These are read-only bits and cannot be modifie

immediately after setting this bit to 1. All interr
be disabled during download. This bit is cleare
when download is completed.

During program download initiated with this bi
particular processing which accompanies ban
switching of the program storage area is exec
Before a download request, initialize the VBR
to H'00000000. After download is completed,
contents can be changed.

0: Download of the programming/erasing pro
not requested.

[Clearing condition]

- When download is completed

1: Download of the programming/erasing pro
requested.

[Setting conditions] (When all of the following
are satisfied)

- Not in RAM emulation mode (the RAMS bi
RAMER is cleared to 0)
- H'A5 is written to FKEY
- Setting of this bit is executed in the on-chi

Note: * This is a write-only bit. This bit is always read as 0.

7 to 1	—	All 0	R	Reserved These are read-only bits and cannot be modifie
0	PPVS	0	R/W	Program Pulse Verify Selects the programming program to be downlo 0: Programming program is not selected. [Clearing condition] When transfer is completed 1: Programming program is selected.

(3) Flash Erase Code Select Register (FECS)

FECS selects the erasing program to be downloaded.

Bit	7	6	5	4	3	2	1	
Bit Name	—	—	—	—	—	—	—	E
Initial Value	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These are read-only bits and cannot be modifie
0	EPVB	0	R/W	Erase Pulse Verify Block Selects the erasing program to be downloaded. 0: Erasing program is not selected. [Clearing condition] When transfer is completed 1: Erasing program is selected.

Bit	Bit Name	Value	R/W	Description
7	K7	0	R/W	Key Code
6	K6	0	R/W	When H'A5 is written to FKEY, writing to the S
5	K5	0	R/W	FCCS is enabled. When a value other than H'
4	K4	0	R/W	written, the SCO bit cannot be set to 1. Theref
3	K3	0	R/W	on-chip program cannot be downloaded to the RAM.
2	K2	0	R/W	Only when H'5A is written can programming/e
1	K1	0	R/W	the flash memory be executed. When a value
0	K0	0	R/W	H'5A is written, even if the programming/erasi program is executed, programming/erasing ca performed.
				H'A5: Writing to the SCO bit is enabled. (The cannot be set to 1 when FKEY is a valu than H'A5.)
				H'5A: Programming/erasing of the flash mem enabled. (When FKEY is a value other H'A5, the software protection state is e
				H'00: Initial value

Bit	Bit Name	Initial Value	R/W	Description
7	MS7	0/1*	R/W	MAT Select
6	MS6	0	R/W	The memory MATs can be switched by writing
5	MS5	0/1*	R/W	to FMATS.
4	MS4	0	R/W	When H'AA is written to FMATS, the user boot
3	MS3	0/1*	R/W	selected. When a value other than H'AA is writt
2	MS2	0	R/W	user MAT is selected. Switch the MATs followin
1	MS1	0/1*	R/W	memory MAT switching procedure in section 18
0	MS0	0	R/W	Switching between User MAT and User Boot M user boot MAT cannot be selected by FMATS i programming mode. The user boot MAT can be selected in boot mode or programmer mode.
				H'AA: The user boot MAT is selected. (The us selected when FMATS is a value other t H'AA.) (Initial value when initiated in user boot
				H'00: The user MAT is selected. (Initial value when initiated in a mode ex user boot mode.)

Note: * This bit is set to 1 in user boot mode, otherwise cleared to 0.

(6) Flash Transfer Destination Address Register (FTDAR)

FTDAR specifies the start address of the on-chip RAM at which to download an on-chip
FTDAR must be set before setting the SCO bit in FCCS to 1.

Bit	7	6	5	4	3	2	1	
Bit Name	TDER	TDA6	TDA5	TDA4	TDA3	TDA2	TDA1	T
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

				H'00 to H'02: 0: The value specified by bits TDA6 to TDA0 the range. 1: The value specified by bits TDA6 to TDA0 between H'03 and H'FF and download has
6	TDA6	0	R/W	Transfer Destination Address
5	TDA5	0	R/W	Specifies the on-chip RAM start address of the download destination. A value between H'00 a and up to 4 Kbytes can be specified as the sta of the on-chip RAM.
4	TDA4	0	R/W	
3	TDA3	0	R/W	
2	TDA2	0	R/W	H'00: H'FF9000 is specified as the sta address.
1	TDA1	0	R/W	
0	TDA0	0	R/W	H'01: H'FFA000 is specified as the sta address. H'02: H'FFB000 is specified as the sta address. H'03 to H'7F: Setting prohibited. (Specifying a value from H'03 to the TDER bit to 1 and stops dow the on-chip program.)

processing result is written in R0. The programming/erasing interface parameters are used
download control, initialization before programming or erasing, programming, and erasin
18.4 shows the usable parameters and target modes. The meaning of the bits in the flash p
fail result parameter (FPFR) varies in initialization, programming, and erasure.

Table 18.4 Parameters and Target Modes

Parameter	Download	Initialization	Programming	Erasure	R/W	Initial Value	Allo
DPFR	O	—	—	—	R/W	Undefined	On-
FPFR	O	O	O	O	R/W	Undefined	R0L
FPEFEQ	—	O	—	—	R/W	Undefined	ER0
FMPAR	—	—	O	—	R/W	Undefined	ER1
FMPDR	—	—	O	—	R/W	Undefined	ER0
FEBS	—	—	—	O	R/W	Undefined	ER0

Note: * A single byte of the start address of the on-chip RAM specified by FTDAR

Download Control: The on-chip program is automatically downloaded by setting the SC
FCCS to 1. The on-chip RAM area to download the on-chip program is the 4-kbyte area s
from the start address specified by FTDAR. Download is set by the programming/erasing
registers, and the download pass and fail result parameter (DPFR) indicates the return val

register ER1. This parameter is called the flash multipurpose address area parameter (FM

The program data is always in 128-byte units. When the program data does not satisfy 1 128-byte program data is prepared by filling the dummy code (H'FF). The boundary of t address of the programming destination on the user MAT is aligned at an address where eight bits (A7 to A0) are H'00 or H'80.

The program data for the user MAT must be prepared in consecutive areas. The program must be in a consecutive space which can be accessed using the MOV.B instruction of t and is not in the flash memory space.

The start address of the area that stores the data to be written in the user MAT must be s general register ER0. This parameter is called the flash multipurpose data destination are parameter (FMPDR).

For details on the programming procedure, see section 18.8.2, User Program Mode.

Erasure: When the flash memory is erased, the erase block number on the user MAT m passed to the erasing program which is downloaded.

The erase block number on the user MAT must be set in general register ER0. This para called the flash erase block select parameter (FEBS).

One block is selected from the block numbers of 0 to 19 as the erase block number.

For details on the erasing procedure, see section 18.8.2, User Program Mode.

7 to 3	—	—	—	Unused These bits return 0.
2	SS	—	R/W	Source Select Error Detect Only one type can be specified for the on-chip p which can be downloaded. When the program t downloaded is not selected, more than two type programs are selected, or a program which is n mapped is selected, an error occurs. 0: Download program selection is normal 1: Download program selection is abnormal
1	FK	—	R/W	Flash Key Register Error Detect Checks the FKEY value (H'A5) and returns the 0: FKEY setting is normal (H'A5) 1: FKEY setting is abnormal (value other than
0	SF	—	R/W	Success/Fail Returns the download result. Reads back the p downloaded to the on-chip RAM and determine whether it has been transferred to the on-chip R 0: Download of the program has ended norma error) 1: Download of the program has ended abnorm (error occurs)

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	—	—	Unused These bits return 0.
1	FQ	—	R/W	Frequency Error Detect Compares the specified CPU operating freque the operating frequencies supported by this LS returns the result. 0: Setting of operating frequency is normal 1: Setting of operating frequency is abnormal
0	SF	—	R/W	Success/Fail Returns the initialization result. 0: Initialization has ended normally (no error) 1: Initialization has ended abnormally (error oc

6	MD	—	R/W	Programming Mode Related Setting Error Dete Detects the error protection state and returns th When the error protection state is entered, this to 1. Whether the error protection state is enter can be confirmed with the FLER bit in FCCS. F conditions to enter the error protection state, se 18.9.3, Error Protection. 0: Normal operation (FLER = 0) 1: Error protection state, and programming ca performed (FLER = 1)
5	EE	—	R/W	Programming Execution Error Detect Writes 1 to this bit when the specified data coul written because the user MAT was not erased. is set to 1, there is a high possibility that the us has been written to partially. In this case, after the error factor, erase the user MAT. If FMATS H'AA and the user boot MAT is selected, an er when programming is performed. In this case, b user MAT and user boot MAT have not been w Programming the user boot MAT should be per boot mode or programmer mode. 0: Programming has ended normally 1: Programming has ended abnormally (progr result is not guaranteed)

				When an address not in the flash memory are specified as the start address of the storage d for the program data, an error occurs. 0: Setting of the start address of the storage destination for the program data is normal 1: Setting of the start address of the storage destination for the program data is abnorm
1	WA	—	R/W	Write Address Error Detect When the following items are specified as the address of the programming destination, an er occurs. • An area other than flash memory • The specified address is not aligned with t byte boundary (lower eight bits of the addre other than H'00 and H'80) 0: Setting of the start address of the program destination is normal 1: Setting of the start address of the program destination is abnormal
0	SF	—	R/W	Success/Fail Returns the programming result. 0: Programming has ended normally (no error 1: Programming has ended abnormally (error

6	MD	—	R/W	Erasure Mode Related Setting Error Detect Detects the error protection state and returns th When the error protection state is entered, this to 1. Whether the error protection state is enter can be confirmed with the FLER bit in FCCS. F conditions to enter the error protection state, se 18.9.3, Error Protection. 0: Normal operation (FLER = 0) 1: Error protection state, and programming ca performed (FLER = 1)
5	EE	—	R/W	Erasure Execution Error Detect Returns 1 when the user MAT could not be era when the flash memory related register settings partially changed. If this bit is set to 1, there is a possibility that the user MAT has been erased p In this case, after removing the error factor, era user MAT. If FMATS is set to H'AA and the use MAT is selected, an error occurs when erasure performed. In this case, both the user MAT and boot MAT have not been erased. Erasing of the boot MAT should be performed in boot mode o programmer mode. 0: Erasure has ended normally 1: Erasure has ended abnormally

				0: Setting of erase block number is normal 1: Setting of erase block number is abnormal
2, 1	—	—	—	Unused These bits return 0.
0	SF	—	R/W	Success/Fail Indicates the erasure result. 0: Erasure has ended normally (no error) 1: Erasure has ended abnormally (error occur

(3) Flash Program/Erase Frequency Parameter (FPEFEQ: General Register ER0

FPEFEQ sets the operating frequency of the CPU. The operating frequency available in
ranges from 8 MHz to 35 MHz.

Bit	31	30	29	28	27	26	25	
Bit Name	—	—	—	—	—	—	—	

Bit	23	22	21	20	19	18	17	
Bit Name	—	—	—	—	—	—	—	

Bit	15	14	13	12	11	10	9	
Bit Name	F15	F14	F13	F12	F11	F10	F9	

Bit	7	6	5	4	3	2	1	
Bit Name	F7	F6	F5	F4	F3	F2	F1	

be shown in a number of two decimal place

2. The value multiplied by 100 is converted to
binary digit and is written to FPEFEQ (gene
register ER0).

For example, when the operating frequency of
is 35.000 MHz, the value is as follows:

1. The number of three decimal places of 35.0
rounded.
2. The formula of 35.00 × 100 = 3500 is conve
the binary digit and B'0000 1101 1010 1100
(H'0DAC) is set to ER0.

Bit	23	22	21	20	19	18	17
Bit Name	MOA23	MOA22	MOA21	MOA20	MOA19	MOA18	MOA17

Bit	15	14	13	12	11	10	9
Bit Name	MOA15	MOA14	MOA13	MOA12	MOA11	MOA10	MOA9

Bit	7	6	5	4	3	2	1
Bit Name	MOA7	MOA6	MOA5	MOA4	MOA3	MOA2	MOA1

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	MOA31 to MOA0	—	R/W	These bits store the start address of the progr destination on the user MAT. Consecutive 128 programming is executed starting from the spe start address of the user MAT. Therefore, the start address of the programming destination 128-byte boundary, and MOA6 to MOA0 are a cleared to 0.

Bit	23	22	21	20	19	18	17	
Bit Name	MOD23	MOD22	MOD21	MOD20	MOD19	MOD18	MOD17	MO

Bit	15	14	13	12	11	10	9	
Bit Name	MOD15	MOD14	MOD13	MOD12	MOD11	MOD10	MOD9	M

Bit	7	6	5	4	3	2	1	
Bit Name	MOD7	MOD6	MOD5	MOD4	MOD3	MOD2	MOD1	M

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	MOD31 to MOD0	—	R/W	These bits store the start address of the area w stores the program data for the user MAT. Con 128-byte data is programmed to the user MAT from the specified start address.

Bit	31	30	29	28	27	26	25	
Bit Name								
Initial Value	—	—	—	—	—	—	—	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Bit	23	22	21	20	19	18	17	
Bit Name								
Initial Value	—	—	—	—	—	—	—	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Bit	15	14	13	12	11	10	9	
Bit Name								
Initial Value	—	—	—	—	—	—	—	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Bit	7	6	5	4	3	2	1	
Bit Name								
Initial Value	—	—	—	—	—	—	—	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Initial Value	0	0	0	0	0	0	0	
R/W	R	R	R	R	R/W	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	0	R	Reserved These are read-only bits and cannot be modifie
3	RAMS	0	R/W	RAM Select Selects the function which emulates the flash m using the on-chip RAM. 0: Disables RAM emulation function 1: Enables RAM emulation function (all blocks o user MAT are protected against programmin erasing)
2	RAM2	0	R/W	Flash Memory Area Select
1	RAM1	0	R/W	These bits select the user MAT area overlaid w on-chip RAM when RAMS = 1. The following ar correspond to the 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)

Mode Setting	MD2	MD1	MD0
User boot mode	0	0	1
Boot mode	0	1	0
User program mode	1	1	0
	1	1	1

18.8.1 Boot Mode

Boot mode executes programming/erasing of the user MAT or user boot MAT by means control command and program data transmitted from the externally connected host via t SCI_4.

In boot mode, the tool for transmitting the control command and program data, and the p data must be prepared in the host. The serial communication mode is set to asynchronou The system configuration in boot mode is shown in figure 18.6. Interrupts are ignored in mode. Configure the user system so that interrupts do not occur.

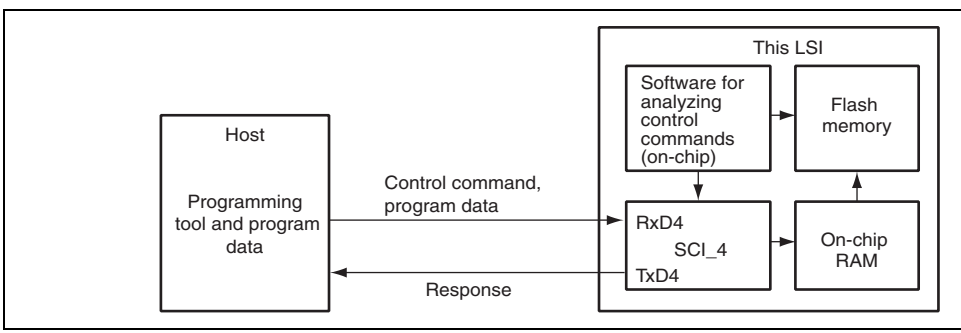


Figure 18.6 System Configuration in Boot Mode

adjustment end sign. When the host receives this bit adjustment end sign normally, it tran
byte of H'55 to this LSI. When reception is not executed normally, initiate boot mode aga
bit rate may not be adjusted within the allowable range depending on the combination of
rate of the host and the system clock frequency of this LSI. Therefore, the transfer bit rate
host and the system clock frequency of this LSI must be as shown in table 18.6.

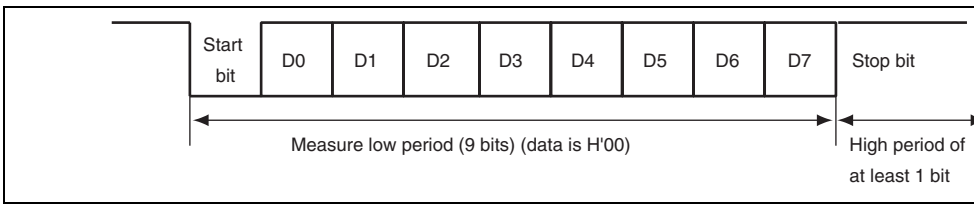


Figure 18.7 Automatic-Bit-Rate Adjustment Operation

Table 18.6 System Clock Frequency for Automatic-Bit-Rate Adjustment

Bit Rate of Host	System Clock Frequency of This LSI
9,600 bps	8 to 18 MHz
19,200 bps	8 to 18 MHz

(2) State Transition Diagram

The state transition after boot mode is initiated is shown in figure 18.8.

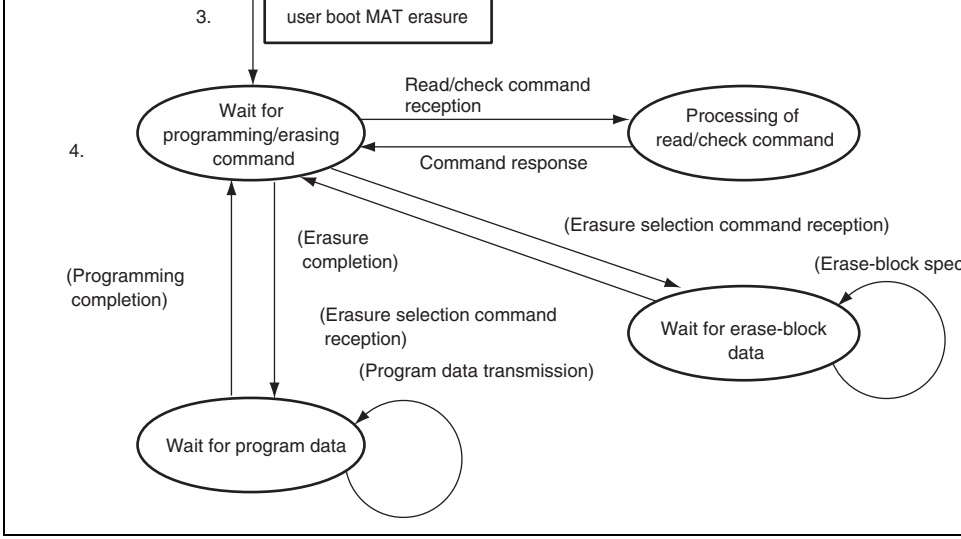


Figure 18.8 Boot Mode State Transition Diagram

which the programming end command is issued, erase the erase block. An example o
erase block is shown in figure 18.9. When the erasure preparation notice is received, t
of waiting for erase block data is entered. The erase block number must be transmitte
erasing command is transmitted. When the erasure is finished, the erase block number
set to H'FF and transmitted. Then the state of waiting for erase block data is returned
state of waiting for programming/erasing command. Erasure must be executed when t
specified block is programmed without a reset start after programming is executed in
mode. When programming can be executed by only one operation, all blocks are eras
entering the state of waiting for programming/erasing command or another command
this case, the erasing operation is not required. The commands other than the
programming/erasing command perform sum check, blank check (erasure check), and
read of the user MAT/user boot MAT and acquisition of current status information.

Memory read of the user MAT/user boot MAT can only read the data programmed after a
MAT/user boot MAT has automatically been erased. No other data can be read.

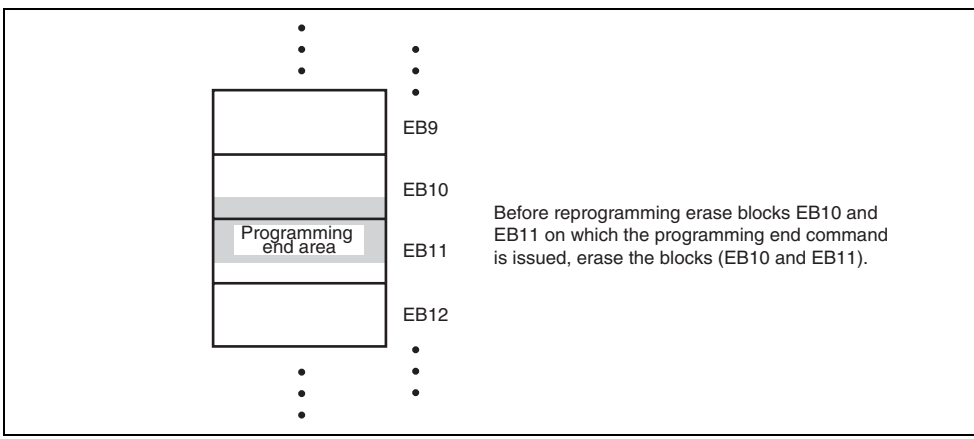
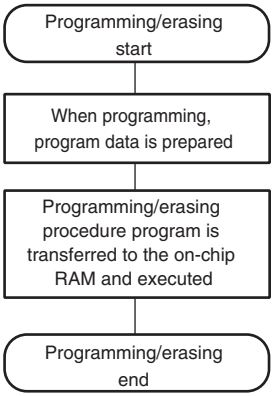


Figure 18.9 Example of Erase Block Including Programmed Area



1. Exit RAM emulation mode beforehand. Download is not allowe in emulation mode.
2. When the program data is adjusted in emulation mode, select the download destination specified by FTDAR carefully. Make sure that the download area does not overlap the emulation area.
3. Programming/erasing is executed only in the on-chip RAM.
4. After programming/erasing is finished, protect the flash memo by the hardware protection.

Figure 18.10 Programming/Erasing Flow

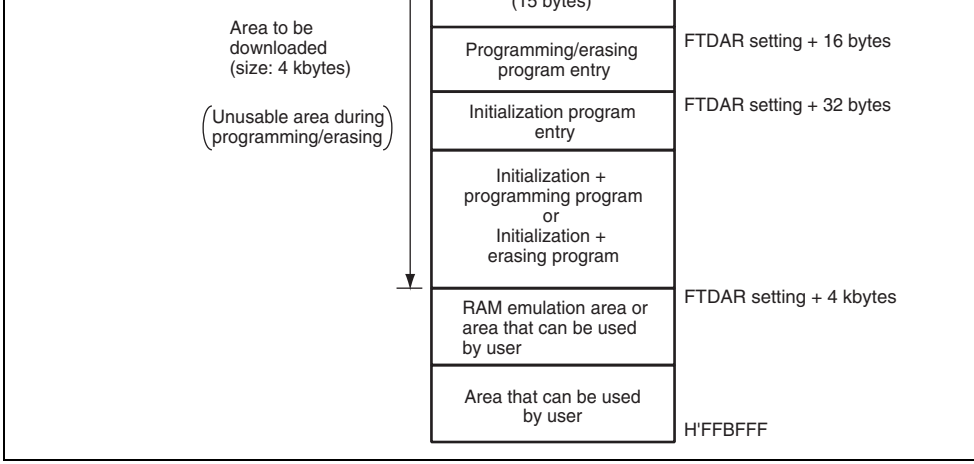


Figure 18.11 RAM Map when Programming/Erasing is Executed

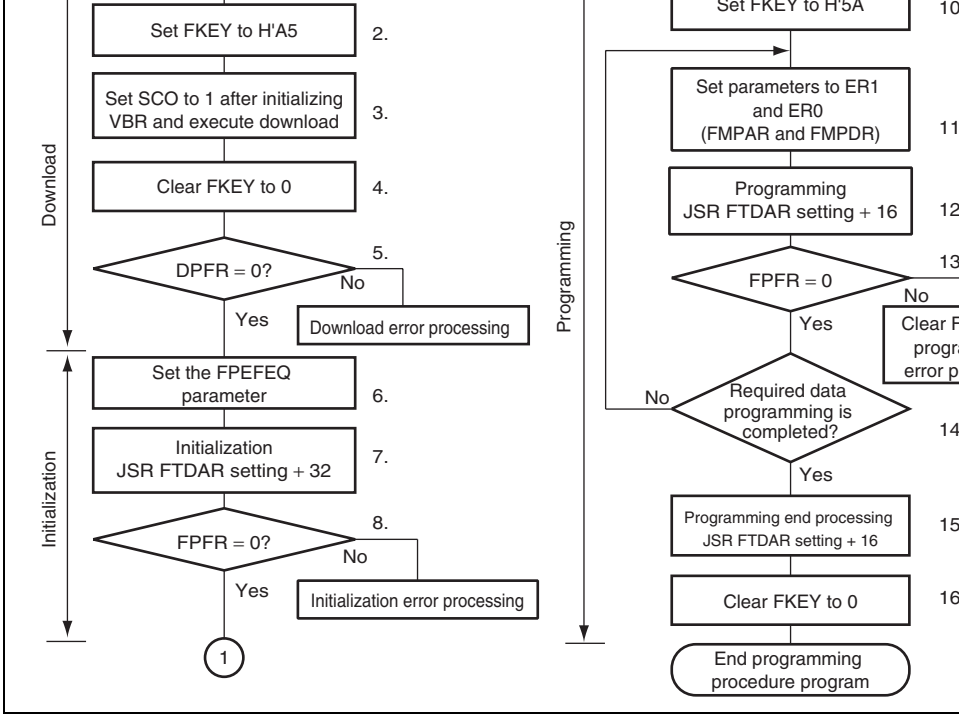


Figure 18.12 Programming Procedure in User Program Mode

H'FF, the program processing time can be shortened.

1. Select the on-chip program to be downloaded and the download destination. When the bit in FPCS is set to 1, the programming program is selected. Several programming/er programs cannot be selected at one time. If several programs are selected, a download returned to the SS bit in the DPFR parameter. The on-chip RAM start address of the d destination is specified by FTDAR.
2. Write H'A5 in FKEY. If H'A5 is not written to FKEY, the SCO bit in FCCS cannot be to request download of the on-chip program.
3. After initializing VBR to H'00000000, set the SCO bit to 1 to execute download. To s SCO bit to 1, all of the following conditions must be satisfied.
   — RAM emulation mode has been canceled.
   — H'A5 is written to FKEY.
   — Setting the SCO bit is executed in the on-chip RAM.

   When the SCO bit is set to 1, download is started automatically. Since the SCO bit is to 0 when the procedure program is resumed, the SCO bit cannot be confirmed to be 1 procedure program. The download result can be confirmed by the return value of the parameter. To prevent incorrect decision, before setting the SCO bit to 1, set one byte on-chip RAM start address specified by FTDAR, which becomes the DPFR paramete value other than the return value (e.g. H'FF). Since particular processing that is accom by bank switching as described below is performed when download is executed, initia VBR contents to H'00000000. Dummy read of FCCS must be performed twice immed after the SCO bit is set to 1.
   — The user-MAT space is switched to the on-chip program storage area.
   — After the program to be downloaded and the on-chip RAM start address specified FTDAR are checked, they are transferred to the on-chip RAM.
   — FPCS, FECS, and the SCO bit in FCCS are cleared to 0.

— If access to the flash memory is requested by the DMAC or DTC during downloa
operation cannot be guaranteed. Make sure that an access request by the DMAC
not generated.

4. FKEY is cleared to H'00 for protection.
5. The download result must be confirmed by the value of the DPFR parameter. Check
of the DPFR parameter (one byte of start address of the download destination specifi
FTDAR). If the value of the DPFR parameter is H'00, download has been performed
If the value is not H'00, the source that caused download to fail can be investigated b
description below.
   — If the value of the DPFR parameter is the same as that before downloading, the s
   the start address of the download destination in FTDAR may be abnormal. In this
   confirm the setting of the TDER bit in FTDAR.
   — If the value of the DPFR parameter is different from that before downloading, ch
   bit or FK bit in the DPFR parameter to confirm the download program selection
   setting, respectively.
6. The operating frequency of the CPU is set in the FPEFEQ parameter for initializatio
settable operating frequency of the FPEFEQ parameter ranges from 8 to 35 MHz. W
frequency is set otherwise, an error is returned to the FPFR parameter of the initializ
program and initialization is not performed. For details on setting the frequency, see
18.7.2 (3), Flash Program/Erase Frequency Parameter (FPEFEQ: General Register E
CPU).

— Since the stack area is used in the initialization program, a stack area of 128 bytes maximum must be allocated in RAM.

— Interrupts can be accepted during execution of the initialization program. Make su program storage area and stack area in the on-chip RAM and register values are n overwritten.

8. The return value in the initialization program, the FPFR parameter is determined.
9. All interrupts and the use of a bus master other than the CPU are disabled during programming/erasing. The specified voltage is applied for the specified time when programming or erasing. If interrupts occur or the bus mastership is moved to other th CPU during programming/erasing, causing a voltage exceeding the specifications to b applied, the flash memory may be damaged. Therefore, interrupts are disabled by sett (I bit) in the condition code register (CCR) to B'1 in interrupt control mode 0 and by s bits 2 to 0 (I2 to I0 bits) in the extend register (EXR) to B'111 in interrupt control mod Accordingly, interrupts other than NMI are held and not executed. Configure the user so that NMI interrupts do not occur. The interrupts that are held must be executed afte programming completes. When the bus mastership is moved to other than the CPU, su the DMAC or DTC, the error protection state is entered. Therefore, make sure the DM not acquire the bus.
10. FKEY must be set to H'5A and the user MAT must be prepared for programming.
11. The parameters required for programming are set. The start address of the programmi destination on the user MAT (FMPAR parameter) is set in general register ER1. The s address of the program data storage area (FMPDR parameter) is set in general registe

— Example of FMPAR parameter setting: When an address other than one in the use area is specified for the start address of the programming destination, even if the programming program is executed, programming is not executed and an error is re the FPFR parameter. Since the program data for one programming operation is 12 the lower eight bits of the address must be H'00 or H'80 to be aligned with the 128 boundary.

— The general registers other than ER0 and ER1 are held in the programming progr
— R0L is a return value of the FPFR parameter.
— Since the stack area is used in the programming program, a stack area of 128 byt maximum must be allocated in RAM.

13. The return value in the programming program, the FPFR parameter is determined.
14. Determine whether programming of the necessary data has finished. If more than 12 data are to be programmed, update the FMPAR and FMPDR parameters in 128-byte repeat steps 11 to 14. Increment the programming destination address by 128 bytes a the programming data pointer correctly. If an address which has already been progra written to again, not only will a programming error occur, but also flash memory wil damaged.
15. Programming end processing is executed.

    The entry point of the programming library is at the address which is 16 bytes after t address of the download destination specified by FTDAR. Call the subroutine by usi following steps.

```
MOV.L    #H'F0F0F0F0,ER0
MOV.L    #H'0F0F0F0F,ER1
MOV.L    #DLTOP+16,ER2   ; Set entry address to ER2
JSR      @ER2            ; Call programming end routine
```

— The general registers other than ER0 and ER1 are held in the programming end p
— R0L is a return value of the FPFR parameter.
— Since the stack area is used in the programming end program, a stack area of 128 the maximum must be allocated in RAM.

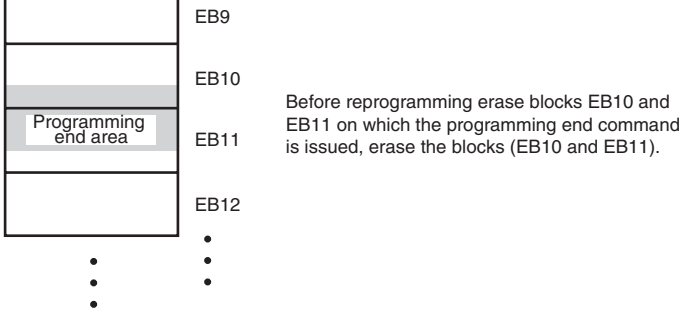


Figure 18.13 Example of Erase Block Including Programmed Area

16. After programming finishes, clear FKEY and specify software protection. If this LSI restarted by a reset immediately after programming has finished, secure the reset inpu (period of $\overline{RES}$ = 0) of at least 100 μs.

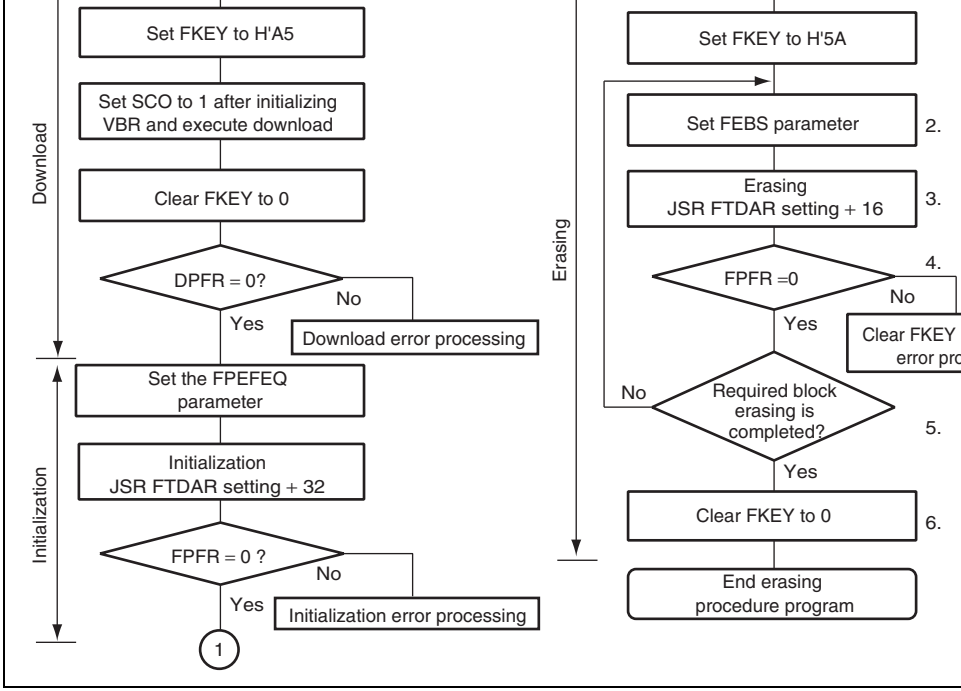


Figure 18.14 Erasing Procedure in User Program Mode

bit in FPCS is set to 1, the programming program is selected. Several programming/er
programs cannot be selected at one time. If several programs are selected, a download
returned to the SS bit in the DPFR parameter. The on-chip RAM start address of the d
destination is specified by FTDAR.

For the procedures to be carried out after setting FKEY, see section 18.8.2 (2), Progra
Procedure in User Program Mode.

2. Set the FEBS parameter necessary for erasure. Set the erase block number (FEBS par
of the user MAT in general register ER0. If a value other than an erase block number
user MAT is set, no block is erased even though the erasing program is executed, and
is returned to the FPFR parameter.
3. Erasure is executed. Similar to as in programming, the entry point of the erasing prog
the address which is 16 bytes after #DLTOP (start address of the download destinatio
specified by FTDAR). Call the subroutine to execute erasure by using the following s

```
MOV.L #DLTOP+16, ER2      ; Set entry address to ER2
JSR   @ER2                ; Call erasing routine
NOP
```

   - The general registers other than ER0 and ER1 are held in the erasing program.
   - R0L is a return value of the FPFR parameter.
   - Since the stack area is used in the erasing program, a stack area of 128 bytes at the
     maximum must be allocated in RAM.
4. The return value in the erasing program, the FPFR parameter is determined.
5. Determine whether erasure of the necessary blocks has finished. If more than one bloc
be erased, update the FEBS parameter and repeat steps 2 to 5.
6. After erasure completes, clear FKEY and specify software protection. If this LSI is re
a power-on reset immediately after erasure has finished, secure the reset input period
of $\overline{RES}$ = 0) of at least 100 μs.

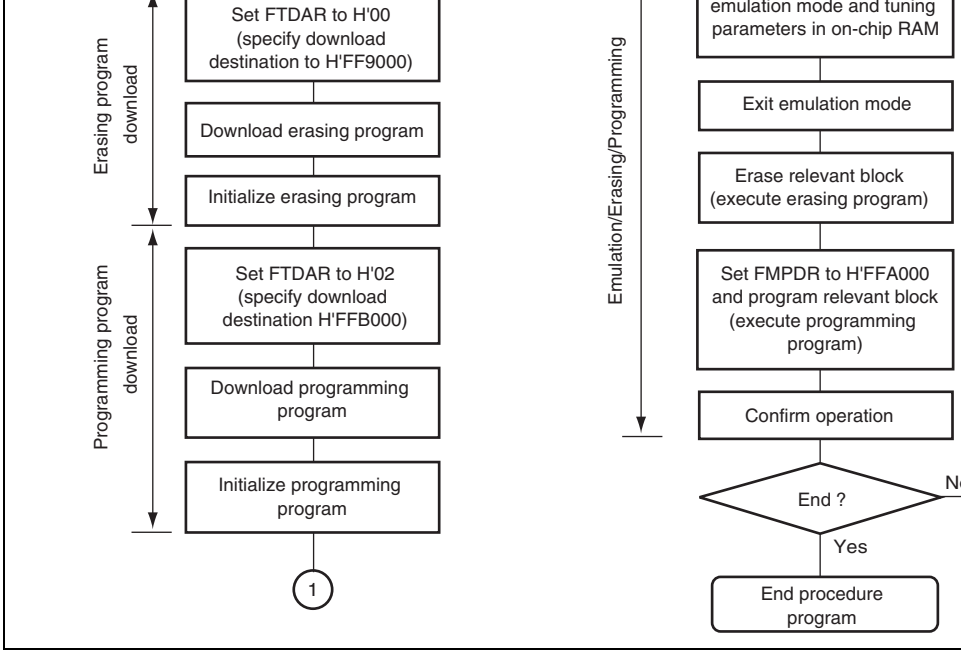


Figure 18.15 Repeating Procedure of Erasing, Programming, and RAM Emulation in User Program Mode

Initialization must be executed for both entry addresses: #DLTOP (start address of do
destination for erasing program) + 32 bytes, and #DLTOP (start address of download
destination for programming program) + 32 bytes.

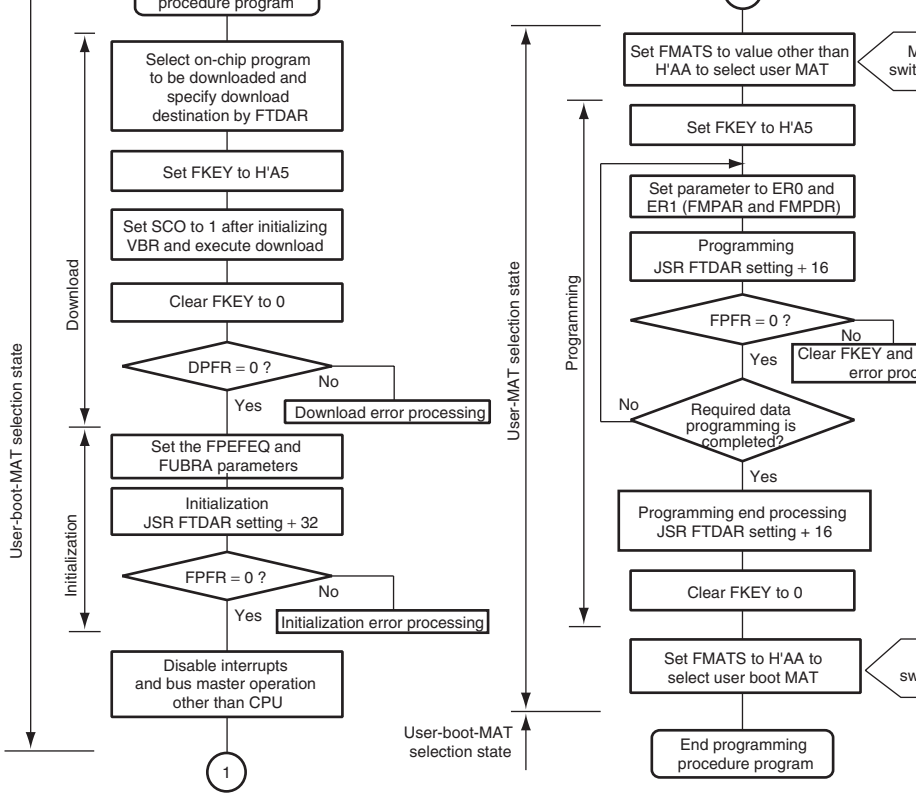
18.8.3 User Boot Mode

Branching to a programming/erasing program prepared by the user enables user boot moc
is a user-arbitrary boot mode to be used.

Only the user MAT can be programmed/erased in user boot mode. Programming/erasing
user boot MAT is only enabled in boot mode or programmer mode.

(1) Initiation in User Boot Mode

When the reset start is executed with the mode pins set to user boot mode, the built-in che
routine runs and checks the user MAT and user boot MAT states. While the check routine
running, NMI and all other interrupts cannot be accepted. Next, processing starts from the
execution start address of the reset vector in the user boot MAT. At this point, the user bo
is selected (FMATS = H'AA) as the execution memory MAT.



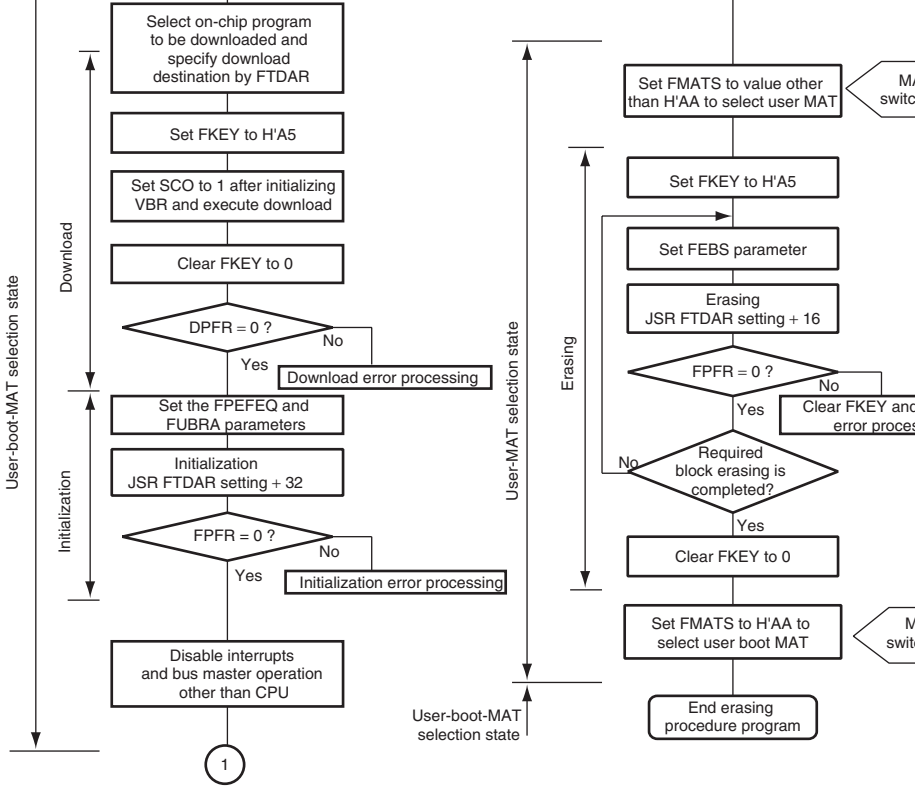
Note: The MAT must be switched by FMATS to perform the programming error processing in the user boot MA

Figure 18.16 Procedure for Programming User MAT in User Boot Mode

description in section 18.11, Switching between User MAT and User Boot MAT.

Except for memory MAT switching, the programming procedure is the same as that in us
program mode.

The area that can be executed in the steps of the procedure program (on-chip RAM, user
and external space) is shown in section 18.8.4, On-Chip Program and Storable Area for P
Data.



Note: The MAT must be switched by FMATS to perform the erasing error processing in the user boot MAT.

Figure 18.17 Procedure for Erasing User MAT in User Boot Mode

Data.

18.8.4 On-Chip Program and Storable Area for Program Data

In the descriptions in this manual, the on-chip programs and program data storage areas a
assumed to be in the on-chip RAM. However, they can be executed from part of the flash
which is not to be programmed or erased as long as the following conditions are satisfied

- The on-chip program is downloaded to and executed in the on-chip RAM specified by
  FTDAR. Therefore, this on-chip RAM area is not available for use.
- Since the on-chip program uses a stack area, allocate 128 bytes at the maximum as a s
  area.
- Download requested by setting the SCO bit in FCCS to 1 should be executed from the
  RAM because it will require switching of the memory MATs.
- In an operating mode in which the external address space is not accessible, such as sin
  mode, the required procedure programs, NMI handling vector table, and NMI handlin
  should be transferred to the on-chip RAM before programming/erasing starts (downlo
  is determined).
- The flash memory is not accessible during programming/erasing. Programming/erasin
  executed by the program downloaded to the on-chip RAM. Therefore, the procedure p
  that initiates operation, the NMI handling vector table, and the NMI handling routine
  stored in the on-chip RAM other than the flash memory.
- After programming/erasing starts, access to the flash memory should be inhibited unti
  is cleared. The reset input state (period of $\overline{RES}$ = 0) must be set to at least 100 µs whe
  operating mode is changed and the reset start executed on completion of programmin
  Transitions to the reset state are inhibited during programming/erasing. When the rese
  is input, a reset input state (period of $\overline{RES}$ = 0) of at least 100 µs is needed before the
  signal is released.

executed are determined by the combination of the processing contents, operating mode structure of the memory MATs, as shown in tables 18.7 to 18.11.

Table 18.7 Executable Memory MAT

	Operating Mode	
Processing Contents	**User Program Mode**	**User Boot Mode***
Programming	See table 18.8	See table 18.10
Erasing	See table 18.9	See table 18.11

Note: * Programming/Erasing is possible to the user MAT.

FCCS (download)			
Operation for clearing FKEY	O	O	O
Decision of download result	O	O	O
Operation for download error	O	O	O
Operation for setting initialization parameter	O	O	O
Execution of initialization	O	×	O
Decision of initialization result	O	O	O
Operation for initialization error	O	O	O
NMI handling routine	O	×	O
Operation for disabling interrupts	O	O	O
Operation for writing H'5A to FKEY	O	O	O
Operation for setting programming parameter	O	×	O
Execution of programming	O	×	O
Decision of programming result	O	×	O
Operation for programming error	O	×	O
Operation for clearing FKEY	O	×	O

Note: * Transferring the program data to the on-chip RAM beforehand enables this are used.

Operation for clearing FKEY	O	O	O
Decision of download result	O	O	O
Operation for download error	O	O	O
Operation for setting initialization parameter	O	O	O
Execution of initialization	O	×	O
Decision of initialization result	O	O	O
Operation for initialization error	O	O	O
NMI handling routine	O	×	O
Operation for disabling interrupts	O	O	O
Operation for writing H'5A to FKEY	O	O	O
Operation for setting erasure parameter	O	×	O
Execution of erasure	O	×	O
Decision of erasure result	O	×	O
Operation for erasure error	O	×	O
Operation for clearing FKEY	O	×	O

FCCS (download)				
Operation for clearing FKEY	O	O		O
Decision of download result	O	O		O
Operation for download error	O	O		O
Operation for setting initialization parameter	O	O		O
Execution of initialization	O	×		O
Decision of initialization result	O	O		O
Operation for initialization error	O	O		O
NMI handling routine	O	×		O
Operation for disabling interrupts	O	O		O
Switching memory MATs by FMATS	O	×	O	
Operation for writing H'5A to FKEY	O	×	O	
Operation for setting programming parameter	O	×	O	
Execution of programming	O	×	O	
Decision of programming result	O	×	O	
Operation for programming error	O	×*2	O	
Operation for clearing FKEY	O	×	O	
Switching memory MATs by FMATS	O	×		O

Notes: 1. Transferring the program data to the on-chip RAM beforehand enables this are used.

2. Switching memory MATs by FMATS by a program in the on-chip RAM enables area to be used.

Operation for clearing FKEY	O	O		O
Decision of download result	O	O		O
Operation for download error	O	O		O
Operation for setting initialization parameter	O	O		O
Execution of initialization	O	×		O
Decision of initialization result	O	O		O
Operation for initialization error	O	O		O
NMI handling routine	O	×		O
Operation for disabling interrupts	O	O		O
Switching memory MATs by FMATS	O	×	O	
Operation for writing H'5A to FKEY	O	×	O	
Operation for setting erasure parameter	O	×	O	
Execution of erasure	O	×	O	
Decision of erasure result	O	×	O	
Operation for erasure error	O	×*	O	
Operation for clearing FKEY	O	×	O	
Switching memory MATs by FMATS	O	×	O	

Note: Switching memory MATs by FMATS by a program in the on-chip RAM enables th
be used.

program is initiated, and the error in programming/erasing is indicated by the FPFR paran

Table 18.12 Hardware Protection

		Function to be Pro	
Item	**Description**	**Download**	**Progra Erasin**
Reset protection	• The programming/erasing interface registers are initialized in the reset state (including a reset by the WDT) and the programming/erasing protection state is entered. • The reset state will not be entered by a reset using the $\overline{\text{RES}}$ pin unless the $\overline{\text{RES}}$ pin is held low until oscillation has settled after a power is initially supplied. In the case of a reset during operation, hold the $\overline{\text{RES}}$ pin low for the $\overline{\text{RES}}$ pulse width given in the AC characteristics. If a reset is input during programming or erasure, data in the flash memory is not guaranteed. In this case, execute erasure and then execute programming again.	O	O

by SCO bit	entered when the SCO bit in FCCS is cleared to 0 to disable download of the programming/erasing programs.		
Protection by FKEY	The programming/erasing protection state is entered because download and programming/erasing are disabled unless the required key code is written in FKEY.	O	O
Emulation protection	The programming/erasing protection state is entered when the RAMS bit in the RAM emulation register (RAMER) is set to 1.	O	O

18.9.3 Error Protection

Error protection is a mechanism for aborting programming or erasure when a CPU runav occurs or operations not according to the programming/erasing procedures are detected programming/erasing of the flash memory. Aborting programming or erasure in such ca prevents damage to the flash memory due to excessive programming or erasing.

If an error occurs during programming/erasing of the flash memory, the FLER bit in FC to 1 and the error protection state is entered.

- When an interrupt request, such as NMI, occurs during programming/erasing.
- When the flash memory is read from during programming/erasing (including a vecto an instruction fetch).
- When a SLEEP instruction is executed (including software-standby mode) during programming/erasing.
- When a bus master other than the CPU, such as the DMAC and DTC, obtains bus m during programming/erasing.

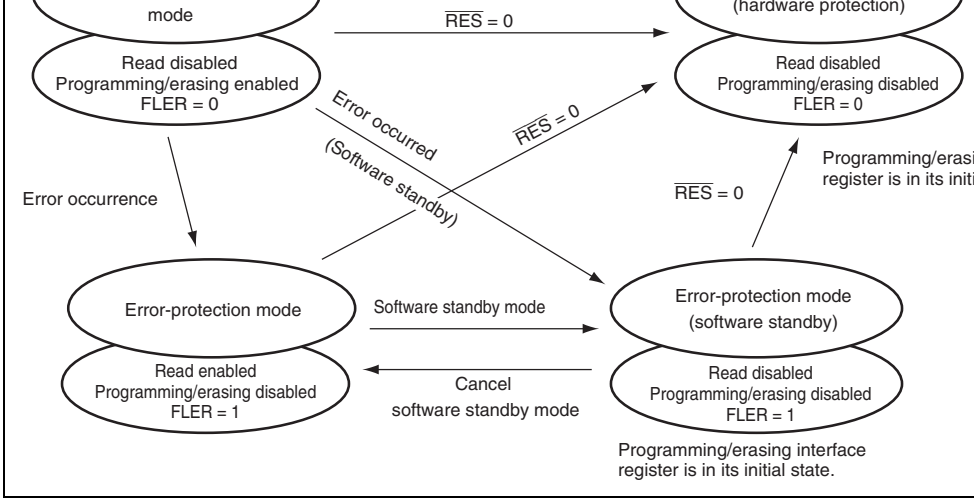


Figure 18.18 Transitions to Error Protection State

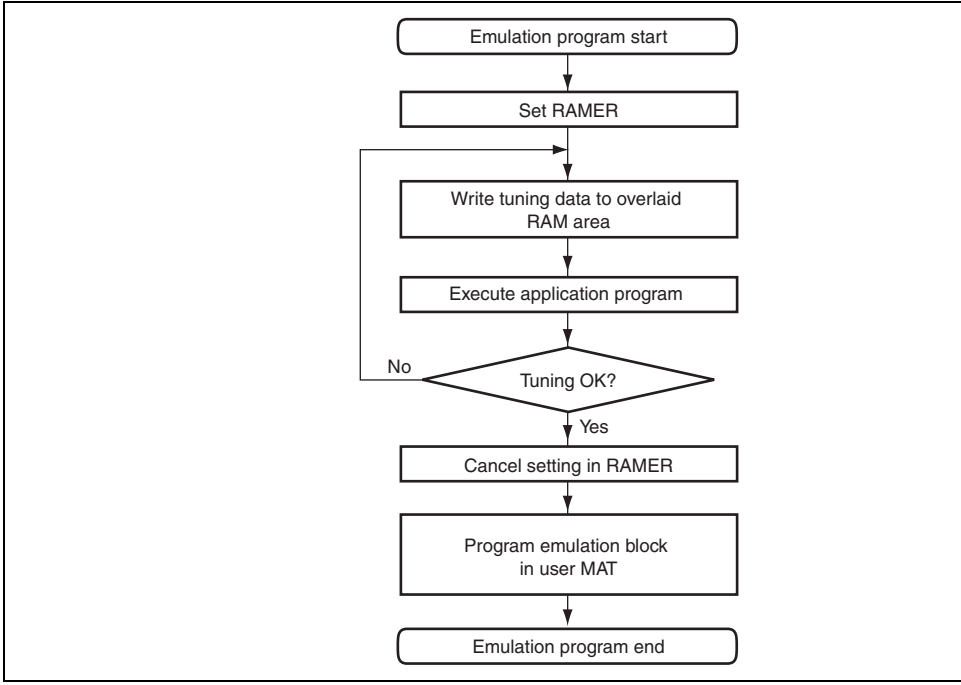
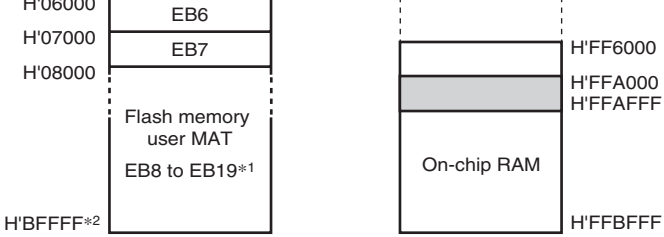


Figure 18.19 RAM Emulation Flow



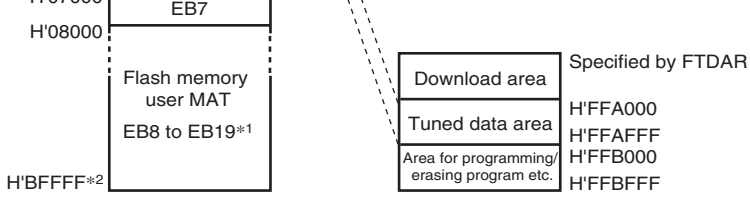
Notes: 1. EB8 to EB15 for H8SX/1656C.
2. H'7FFF for H8SX/1656C.

Figure 18.20 Address Map of Overlaid RAM Area (H8SX/1657C)

The flash memory area that can be emulated is the one area selected by bits RAM2 to RA
RAMER from among the eight blocks, EB0 to EB7, of the user MAT.

To overlay a part of the on-chip RAM with block EB0 for realtime emulation, set the RA
RAMER to 1 and bits RAM2 to RAM0 to B'000.

For programming/erasing the user MAT, the procedure programs including a download p
of the on-chip program must be executed. At this time, the download area should be spec
that the overlaid RAM area is not overwritten by downloading the on-chip program. Sinc
in which the tuned data is stored is overlaid with the download area when FTDAR = H'01
tuned data must be saved in an unused area beforehand.



Notes: 1. EB8 to EB15 for H8SX/1656C.
2. H'7FFFF for H8SX/1656C.

Figure 18.21 Programming Tuned Data (H8SX/1657C)

1. After tuning program data is completed, clear the RAMS bit in RAMER to 0 to canc
overlaid RAM.
2. Transfer the user-created procedure program to the on-chip RAM.
3. Start the procedure program and download the on-chip program to the on-chip RAM
address of the download destination should be specified by FTDAR so that the tuned
does not overlay the download area.
4. When block EB0 of the user MAT has not been erased, the programming program m
downloaded after block EB0 is erased. Specify the tuned data saved in the FMPAR a
FMPDR parameters and then execute programming.

Note: Setting the RAMS bit to 1 makes all the blocks of the user MAT enter the programming/erasing protection state (emulation protection state) regardless of t
of the RAM2 to RAM0 bits. Under this condition, the on-chip program cannot b
downloaded. When data is to be actually programmed and erased, clear the RAM
to 0.

for eight times (this prevents access to the flash memory during memory MAT switch

3. If an interrupt request has occurred during memory MAT switching, there is no guara which memory MAT is accessed. Always mask the maskable interrupts before switch memory MATs. In addition, configure the system so that NMI interrupts do not occur memory MAT switching.
4. After the memory MATs have been switched, take care because the interrupt vector ta also have been switched. If interrupt processing is to be the same before and after mer MAT switching, transfer the interrupt processing routines to the on-chip RAM and sp VBR to place the interrupt vector table in the on-chip RAM.
5. The size of the user MAT is different from that of the user boot MAT. Addresses whi the size of the 8-kbyte user boot MAT should not be accessed. If an attempt is made, read as an undefined value.

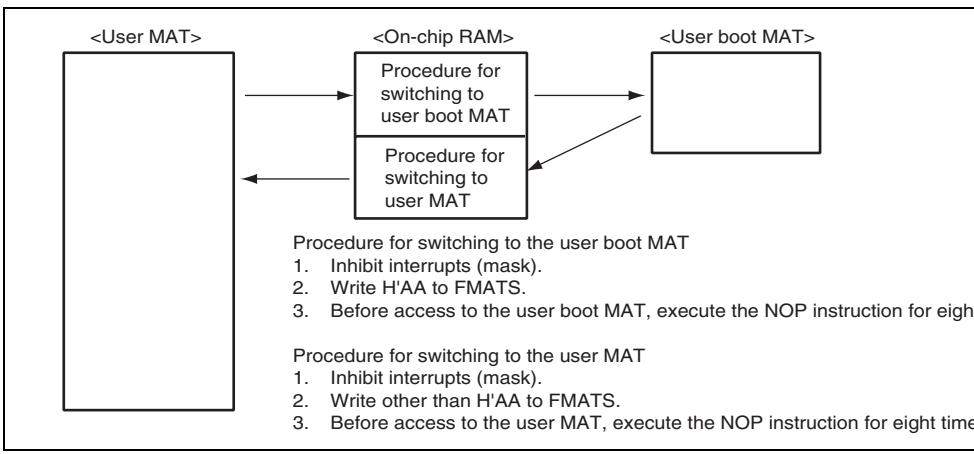


Figure 18.22 Switching between User MAT and User Boot MAT

	H8SX/1656C	512 Kbytes	
User boot MAT	H8SX/1657C	8 Kbytes	FZTATUSBT
	H8SX/1656C		

18.13 Standard Serial Communication Interface Specifications for Mode

The boot program initiated in boot mode performs serial communication using the host a chip SCI_4. The serial communication interface specifications are shown below.

The boot program has three states.

1. Bit-rate-adjustment state

   In this state, the boot program adjusts the bit rate to achieve serial communication w host. Initiating boot mode enables starting of the boot program and entry to the bit-ra adjustment state. The program receives the command from the host to adjust the bit r adjusting the bit rate, the program enters the inquiry/selection state.

2. Inquiry/selection state

   In this state, the boot program responds to inquiry commands from the host. The dev clock mode, and bit rate are selected. After selection of these settings, the program is enter the programming/erasing state by the command for a transition to the programming/erasing state. The program transfers the libraries required for erasure t chip RAM and erases the user MATs and user boot MATs before the transition.

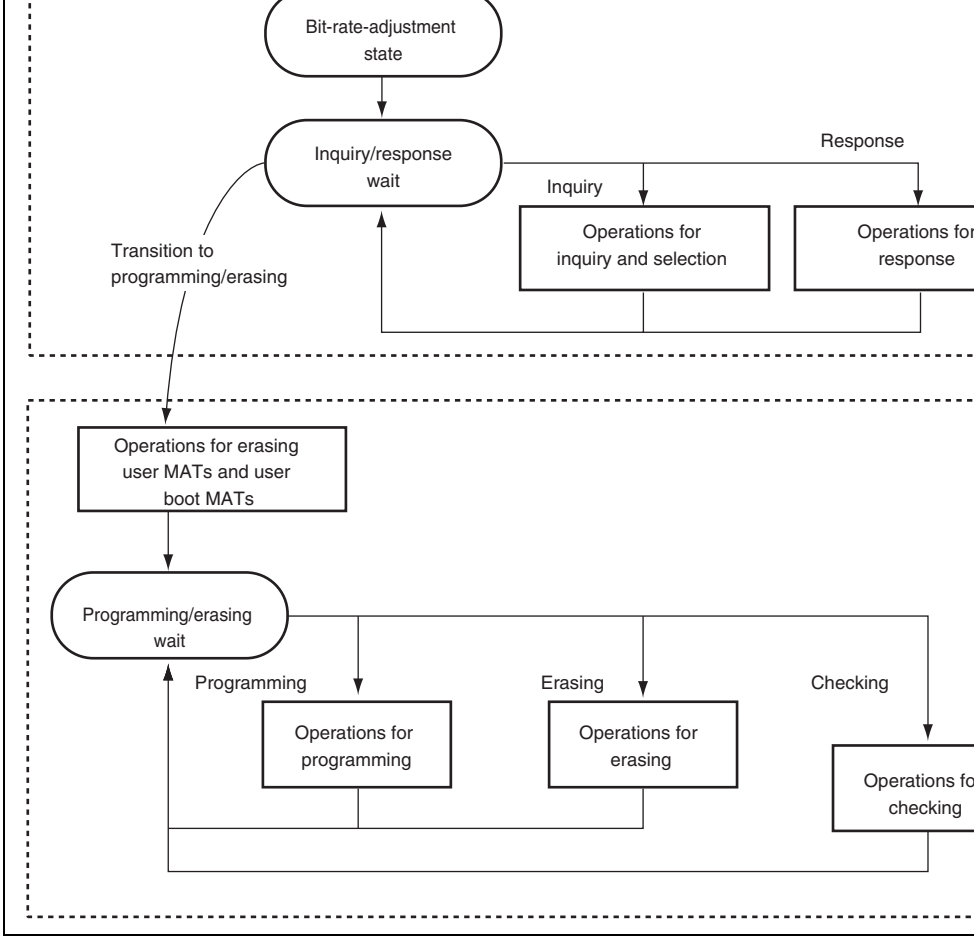


Figure 18.23 Boot Program States

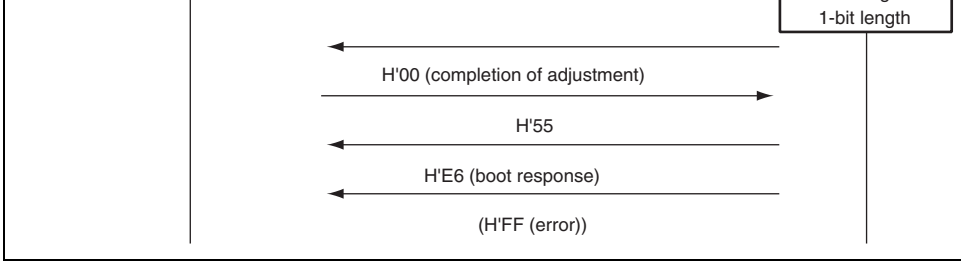


Figure 18.24 Bit-Rate-Adjustment Sequence

(2) Communications Protocol

After adjustment of the bit rate, the protocol for serial communications between the host
boot program is as shown below.

1. One-byte commands and one-byte responses

   These one-byte commands and one-byte responses consist of the inquiries and the A
   successful completion.
2. n-byte commands or n-byte responses

   These commands and responses are comprised of n bytes of data. These are selectio
   responses to inquiries.

   The program data size is not included under this heading because it is determined in
   command.
3. Error response

   The error response is a response to inquiries. It consists of an error response and an e
   and comes two bytes.
4. Programming of 128 bytes

   The size is not specified in commands. The size of n is indicated in response to the
   programming unit inquiry.

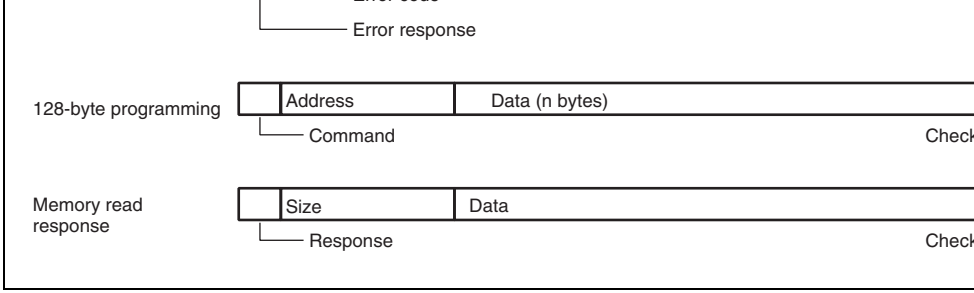


Figure 18.25 Communication Protocol Format

- Command (one byte): Commands including inquiries, selection, programming, erasin checking
- Response (one byte): Response to an inquiry
- Size (one byte): The amount of data for transmission excluding the command, amoun and checksum
- Checksum (one byte): The checksum is calculated so that the total of all values from t command byte to the SUM byte becomes H'00.
- Data (n bytes): Detailed data of a command or response
- Error response (one byte): Error response to a command
- Error code (one byte): Type of the error
- Address (four bytes): Address for programming
- Data (n bytes): Data to be programmed (the size is indicated in the response to the programming unit inquiry.)
- Size (four bytes): Four-byte response to a memory read

H'10	Device selection	Selection of device code
H'21	Clock mode inquiry	Inquiry regarding numbers of clock and values of each mode
H'11	Clock mode selection	Indication of the selected clock moc
H'22	Multiplication ratio inquiry	Inquiry regarding the number of freq multiplied clock types, the number c multiplication ratios, and the values multiple
H'23	Operating clock frequency inquiry	Inquiry regarding the maximum and values of the main clock and periph
H'24	User boot MAT information inquiry	Inquiry regarding the number of use MATs and the start and last addres each MAT
H'25	User MAT information inquiry	Inquiry regarding the a number of u and the start and last addresses of
H'26	Block for erasing information Inquiry	Inquiry regarding the number of blo the start and last addresses of each
H'27	Programming unit inquiry	Inquiry regarding the unit of progran
H'3F	New bit rate selection	Selection of new bit rate
H'40	Transition to programming/erasing state	Erasing of user MAT and user boot entry to programming/erasing state
H'4F	Boot program status inquiry	Inquiry into the operated status of th program

response to the supported device inquiry.

Command

H'20

- Command, H'20, (one byte): Inquiry regarding supported devices

Response

H'30	Size	Number of devices	
Number of characters	Device code		Product name
...			
SUM			

- Response, H'30, (one byte): Response to the supported device inquiry
- Size (one byte): Number of bytes to be transmitted, excluding the command, size, and checksum, that is, the amount of data contributes by the number of devices, characters codes and product names
- Number of devices (one byte): The number of device types supported by the boot pro
- Number of characters (one byte): The number of characters in the device codes and b program's name
- Device code (four bytes): ASCII code of the supporting product
- Product name (n bytes): Type name of the boot program in ASCII-coded characters
- SUM (one byte): Checksum

  The checksum is calculated so that the total number of all values from the command b the SUM byte becomes H'00.

- SUM (one byte): Checksum

Response

H'06

- Response, H'06, (one byte): Response to the device selection command
  ACK will be returned when the device code matches.

Error response

H'90	ERROR

- Error response, H'90, (one byte): Error response to the device selection command
  ERROR : (one byte): Error code
  H'11: Sum check error
  H'21: Device code error, that is, the device code does not match

(c) Clock Mode Inquiry

The boot program will return the supported clock modes in response to the clock mode i

Command

H'21

- Command, H'21, (one byte): Inquiry regarding clock mode

Response

H'31	Size	Mode	···	SUM

- Response, H'31, (one byte): Response to the clock-mode inquiry
- Size (one byte): Amount of data that represents the modes
- Mode (one byte): Values of the supported clock modes (i.e. H'01 means clock mode
- SUM (one byte): Checksum

- SUM (one byte): Checksum

Response

H'06

- Response, H'06, (one byte): Response to the clock mode selection command
  ACK will be returned when the clock mode matches.

Error Response

H'91	ERROR

- Error response, H'91, (one byte): Error response to the clock mode selection comman
- ERROR : (one byte): Error code
  H'11: Checksum error
  H'22: Clock mode error, that is, the clock mode does not match.

Even if the clock mode numbers are H'00 and H'01 by a clock mode inquiry, the clock m
be selected using these respective values.

...						
SUM						

- Response, H'32, (one byte): Response to the multiplication ratio inquiry
- Size (one byte): The amount of data that represents the number of multiplication typ
  multiplication ratios and the multiplication ratios
- Number of multiplication types (one byte): The number of multiplication types to wh
  device can be set.
  (e.g. when there are two multiplied clock types, which are the main and peripheral cl
  number of types will be H'02.)
- Number of multiplication ratios (one byte): The number of multiplication ratios for e
  (e.g. the number of multiplication ratios to which the main clock can be set and the p
  clock can be set.)
- Multiplication ratio (one byte)

  Multiplication ratio: The value of the multiplication ratio (e.g. when the clock-freque
  multiplier is four, the value of multiplication ratio will be H'04.)

  Division ratio: The inverse of the division ratio, i.e. a negative number (e.g. when the
  divided by two, the value of division ratio will be H'FE. H'FE = D'-2)

  The number of multiplication ratios returned is the same as the number of multiplica
  and as many groups of data are returned as there are multiplication types.
- SUM (one byte): Checksum

operating clock frequency	frequency
...	
SUM	

- Response, H'33, (one byte): Response to operating clock frequency inquiry
- Size (one byte): The number of bytes that represents the minimum values, maximum
  and the number of frequencies.
- Number of operating clock frequencies (one byte): The number of supported operatin
  frequency types
  (e.g. when there are two operating clock frequency types, which are the main and peri
  clocks, the number of types will be H'02.)
- Minimum value of operating clock frequency (two bytes): The minimum value of the
  multiplied or divided clock frequency.

  The minimum and maximum values of the operating clock frequency represent the va
  MHz, valid to the hundredths place of MHz, and multiplied by 100. (e.g. when the val
  17.00 MHz, it will be 2000, which is H'07D0.)
- Maximum value (two bytes): Maximum value among the multiplied or divided clock
  frequencies.

  There are as many pairs of minimum and maximum values as there are operating cloc
  frequencies.
- SUM (one byte): Checksum

- Response, H'34, (one byte): Response to user boot MAT information inquiry
- Size (one byte): The number of bytes that represents the number of areas, area-start a
  and area-last address
- Number of Areas (one byte): The number of consecutive user boot MAT areas
  When user boot MAT areas are consecutive, the number of areas returned is H'01.
- Area-start address (four byte): Start address of the area
- Area-last address (four byte): Last address of the area
  There are as many groups of data representing the start and last addresses as there ar
- SUM (one byte): Checksum

(h) User MAT Information Inquiry

The boot program will return the number of user MATs and their addresses.

Command

H'25

- Command, H'25, (one byte): Inquiry regarding user MAT information

Response

H'35	Size	Number of areas	
Start address area			Last address area
...			
SUM			

- Response, H'35, (one byte): Response to the user MAT information inquiry
- Size (one byte): The number of bytes that represents the number of areas, area-start a
  and area-last address
- Number of areas (one byte): The number of consecutive user MAT areas
  When the user MAT areas are consecutive, the number of areas is H'01.
- Area-start address (four bytes): Start address of the area

Response	H'36	Size	Number of blocks
	Block start address		Block last address
	...		
	SUM		

- Response, H'36, (one byte): Response to the number of erased blocks and addresses
- Size (three bytes): The number of bytes that represents the number of blocks, block-st
  addresses, and block-last addresses.
- Number of blocks (one byte): The number of erased blocks
- Block start address (four bytes): Start address of a block
- Block last Address (four bytes): Last address of a block
  There are as many groups of data representing the start and last addresses as there are
- SUM (one byte): Checksum

(j) Programming Unit Inquiry

The boot program will return the programming unit used to program data.

Command	H'27	

- Command, H'27, (one byte): Inquiry regarding programming unit

Response	H'37	Size	Programming unit	SUM

- Response, H'37, (one byte): Response to programming unit inquiry
- Size (one byte): The number of bytes that indicate the programming unit, which is fix
- Programming unit (two bytes): A unit for programming
  This is the unit for reception of programming.
- SUM (one byte): Checksum

- Size (one byte): The amount of data that represents the bit rate, input frequency, nur
  multiplication types, and multiplication ratio
- Bit rate (two bytes): New bit rate
  One hundredth of the value (e.g. when the value is 19200 bps, it will be 192, which i
- Input frequency (two bytes): Frequency of the clock input to the boot program
  This is valid to the hundredths place and represents the value in MHz multiplied by 1
  when the value is 20.00 MHz, it will be 2000, which is H'07D0.)
- Number of multiplication types (one byte): The number of multiplication types to wh
  device can be set.
- Multiplication ratio 1 (one byte): The value of multiplication or division ratios for th
  operating frequency
  Multiplication ratio (one byte): The value of the multiplication ratio (e.g. when the c
  frequency is multiplied by four, the multiplication ratio will be H'04.)
  Division ratio: The inverse of the division ratio, as a negative number (e.g. when the
  frequency is divided by two, the value of division ratio will be H'FE. H'FE = D'-2)
- Multiplication ratio 2 (one byte): The value of multiplication or division ratios for th
  peripheral frequency
  Multiplication ratio (one byte): The value of the multiplication ratio (e.g. when the c
  frequency is multiplied by four, the multiplication ratio will be H'04.)
  (Division ratio: The inverse of the division ratio, as a negative number (E.g. when th
  divided by two, the value of division ratio will be H'FE. H'FE = D'-2)
- SUM (one byte): Checksum

Response | H'06 |

- Response, H'06, (one byte): Response to selection of a new bit rate
  When it is possible to set the bit rate, the response will be ACK.

H'27: Operating frequency error
The frequency is not within the specified range.

(4) Receive Data Check

The methods for checking of receive data are listed below.

1. Input frequency

   The received value of the input frequency is checked to ensure that it is within the ran
   minimum to maximum frequencies which matches the clock modes of the specified d
   When the value is out of this range, an input-frequency error is generated.

2. Multiplication ratio

   The received value of the multiplication ratio or division ratio is checked to ensure tha
   matches the clock modes of the specified device. When the value is out of this range,
   frequency error is generated.

3. Operating frequency error

   Operating frequency is calculated from the received value of the input frequency and
   multiplication or division ratio. The input frequency is input to the LSI and the LSI is
   at the operating frequency. The expression is given below.

   Operating frequency = Input frequency × Multiplication ratio, or
   Operating frequency = Input frequency ÷ Division ratio

   The calculated operating frequency should be checked to ensure that it is within the ra
   minimum to maximum frequencies which are available with the clock modes of the sp
   device. When it is out of this range, an operating frequency error is generated.

When the new bit rate is selectable, the rate will be set in the register after sending ACK response. The host will send an ACK with the new bit rate for confirmation and the boot will response with that rate.

Confirmation | H'06 |

- Confirmation, H'06, (one byte): Confirmation of a new bit rate

Response | H'06 |

- Response, H'06, (one byte): Response to confirmation of a new bit rate

The sequence of new bit-rate selection is shown in figure 18.26.

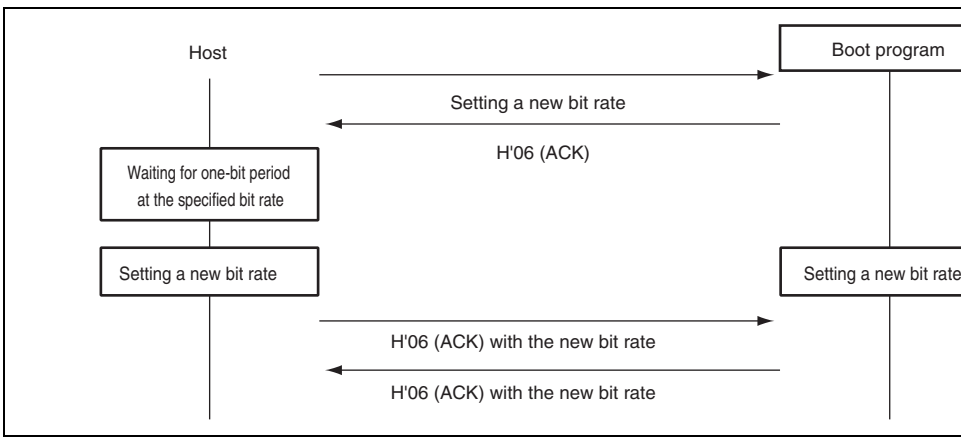


Figure 18.26 New Bit-Rate Selection Sequence

Command | H'40 |

- Command, H'40, (one byte): Transition to programming/erasing state

Response | H'06 |

- Response, H'06, (one byte): Response to transition to programming/erasing state
  The boot program will send ACK when the user MAT and user boot MAT have been
  by the transferred erasing program.

Error Response | H'C0 | H'51 |

- Error response, H'C0, (one byte): Error response for user boot MAT blank check
- Error code, H'51, (one byte): Erasing error
  An error occurred and erasure was not completed.

(6) Command Error

A command error will occur when a command is undefined, the order of commands is inc
or a command is unacceptable. Issuing a clock-mode selection command before a device
or an inquiry command after the transition to programming/erasing state command, are ex

Error Response | H'80 | H'xx |

- Error response, H'80, (one byte): Command error
- Command, H'xx, (one byte): Received command

be made, such as the multiplication-ratio inquiry (H'22) or operating frequency inqui
which are needed for a new bit-rate selection.

6. A new bit rate should be selected with the new bit-rate selection (H'3F) command, ac
to the returned information on multiplication ratios and operating frequencies.
7. After selection of the device and clock mode, the information of the user boot MAT
MAT should be made to inquire about the user boot MATs information inquiry (H'2
MATs information inquiry (H'25), erased block information inquiry (H'26), and prog
unit inquiry (H'27).
8. After making inquiries and selecting a new bit rate, issue the transition to
programming/erasing state command (H'40). The boot program will then enter the
programming/erasing state.

		program
H'43	User MAT programming selection	Transfers the user MAT programmi program
H'50	128-byte programming	Programs 128 bytes of data
H'48	Erasing selection	Transfers the erasing program
H'58	Block erasing	Erases a block of data
H'52	Memory read	Reads the contents of memory
H'4A	User boot MAT sum check	Checks the checksum of the user b
H'4B	User MAT sum check	Checks the checksum of the user M
H'4C	User boot MAT blank check	Checks the blank data of the user b
H'4D	User MAT blank check	Checks the blank data of the user M
H'4C	User boot MAT blank check	Checks whether the contents of the boot MAT are blank
H'4D	User MAT blank check	Checks whether the contents of the MAT are blank
H'4F	Boot program status inquiry	Inquires into the boot program's sta

command represents the data programmed according to the method specified by the
command. When more than 128-byte data is programmed, 128-byte commands shou
repeatedly be executed. Sending a 128-byte programming command with H'FFFFFF
address will stop the programming. On completion of programming, the boot progra
wait for selection of programming or erasing.

Where the sequence of programming operations that is executed includes programm
another method or of another MAT, the procedure must be repeated from the progra
selection command.

The sequence for the programming selection and 128-byte programming commands
in figure 18.27.

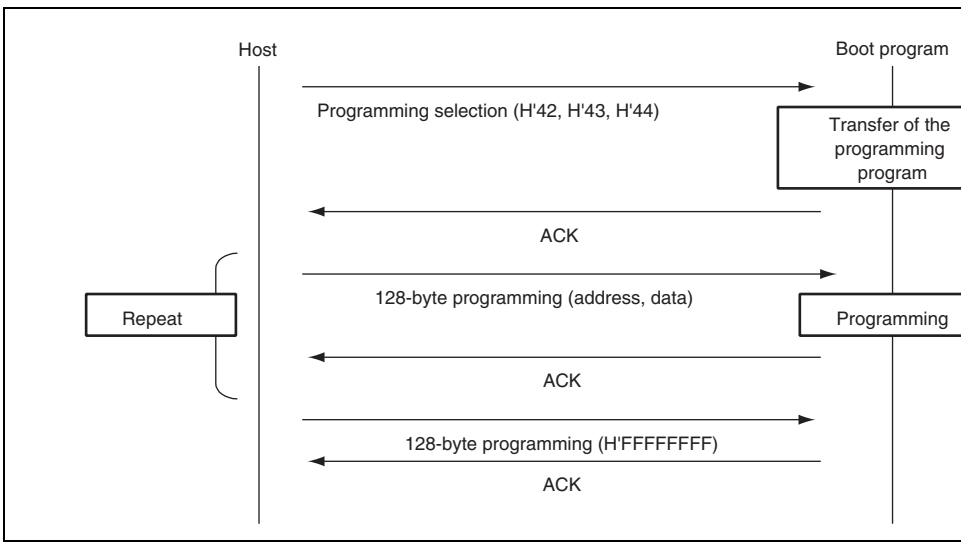


Figure 18.27 Programming Sequence

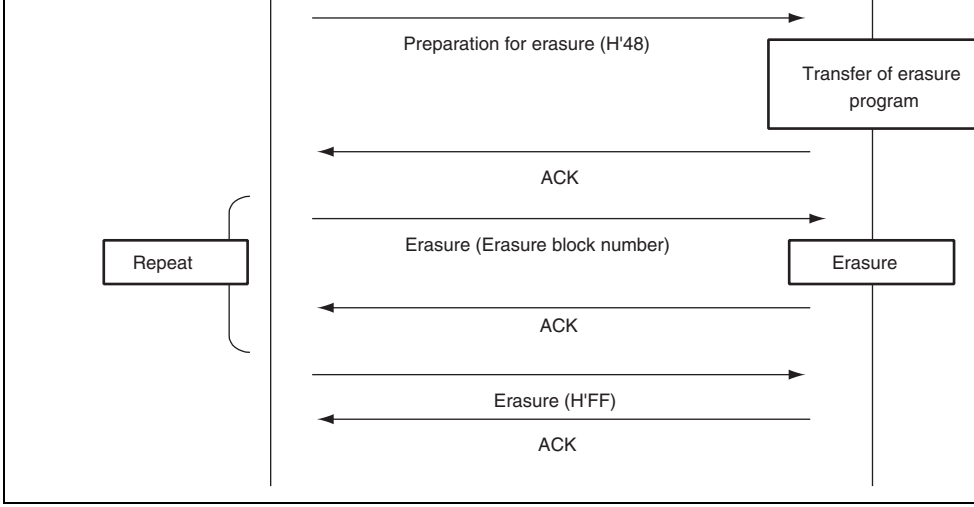


Figure 18.28 Erasure Sequence

Error Response	H'C2	ERROR

- Error response : H'C2 (1 byte): Error response to user boot MAT programming selec
- ERROR : (1 byte): Error code
  H'54: Selection processing error (transfer error occurs and processing is not complet

(b) User MAT Programming Selection

The boot program will transfer a program for user MAT programming selection. The da
programmed to the user MATs by the transferred program for programming.

Command	H'43

- Command, H'43, (one byte): User-program programming selection

Response	H'06

- Response, H'06, (one byte): Response to user-program programming selection
  When the programming program has been transferred, the boot program will return A

Error Response	H'C3	ERROR

- Error response : H'C3 (1 byte): Error response to user boot MAT programming selec
- ERROR : (1 byte): Error code
  H'54: Selection processing error (transfer error occurs and processing is not complet

- Command, H'50, (one byte): 128-byte programming
- Programming Address (four bytes): Start address for programming
  Multiple of the size specified in response to the programming unit inquiry
  (i.e. H'00, H'01, H'00, H'00 : H'01000000)
- Program data (128 bytes): Data to be programmed
  The size is specified in the response to the programming unit inquiry.
- SUM (one byte): Checksum

Response

H'06

- Response, H'06, (one byte): Response to 128-byte programming
  On completion of programming, the boot program will return ACK.

Error Response

H'D0	ERROR

- Error response, H'D0, (one byte): Error response for 128-byte programming
- ERROR: (one byte): Error code

  H'11: Checksum Error

  H'2A: Address error
  The address is not in the specified MAT.

  H'53: Programming error
  A programming error has occurred and programming cannot be continu

The specified address should match the unit for programming of data. For example, when programming is in 128-byte units, the lower eight bits of the address should be H'00 or H When there are less than 128 bytes of data to be programmed, the host should fill the rest H'FF.

Sending the 128-byte programming command with the address of H'FFFFFFFF will stop programming operation. The boot program will interpret this as the end of the programmi wait for selection of programming or erasing.

- Error Response, H'D0, (one byte): Error response for 128-byte programming
- ERROR: (one byte): Error code

  H'11: Checksum error

  H'53: Programming error

  An error has occurred in programming and programming cannot be co

(d) Erasure Selection

The boot program will transfer the erasure program. User MAT data is erased by the tra
erasure program.

Command | H'48 |

- Command, H'48, (one byte): Erasure selection

Response | H'06 |

- Response, H'06, (one byte): Response for erasure selection

  After the erasure program has been transferred, the boot program will return ACK.

Error Response | H'C8 | ERROR |

- Error Response, H'C8, (one byte): Error response to erasure selection
- ERROR: (one byte): Error code

  H'54: Selection processing error (transfer error occurs and processing is not compl

Response

H'06

- Response, H'06, (one byte): Response to Erasure
  After erasure has been completed, the boot program will return ACK.

Error Response

H'D8	ERROR

- Error Response, H'D8, (one byte): Response to Erasure
- ERROR (one byte): Error code

H'11:	Sum check error
H'29:	Block number error Block number is incorrect.
H'51:	Erasure error An error has occurred during erasure.

On receiving block number H'FF, the boot program will stop erasure and wait for a select
command.

Command

H'58	Size	Block number	SUM

- Command, H'58, (one byte): Erasure
- Size, (one byte): The number of bytes that represents the block number
  This is fixed to 1.
- Block number (one byte): H'FF
  Stop code for erasure
- SUM (one byte): Checksum

Response

H'06

- Response, H'06, (one byte): Response to end of erasure (ACK)
  When erasure is to be performed after the block number H'FF has been sent, the proce
  should be executed from the erasure selection command.

An address error occurs when the area setting is incorrect.

- Read address (4 bytes): Start address to be read from
- Read size (4 bytes): Size of data to be read
- SUM (1 byte): Checksum

Response

H'52	Read size						
Data	···						
SUM							

- Response: H'52 (1 byte): Response to memory read
- Read size (4 bytes): Size of data to be read
- Data (n bytes): Data for the read size from the read address
- SUM (1 byte): Checksum

Error Response

H'D2	ERROR

- Error response: H'D2 (1 byte): Error response to memory read
- ERROR: (1 byte): Error code

  H'11: Sum check error

  H'2A: Address error

  The read address is not in the MAT.

  H'2B: Size error

  The read size exceeds the MAT.

This is fixed to 4.

- Checksum of user boot program (four bytes): Checksum of user boot MATs
  The total of the data is obtained in byte units.
- SUM (one byte): Sum check for data being transmitted

(h) User-Program Sum Check

The boot program will return the byte-by-byte total of the contents of the bytes of the use program.

Command	H'4B

- Command, H'4B, (one byte): Sum check for user program

Response	H'5B	Size	Checksum of user program	SUM

- Response, H'5B, (one byte): Response to the sum check of the user program
- Size (one byte): The number of bytes that represents the checksum
  This is fixed to 4.
- Checksum of user boot program (four bytes): Checksum of user MATs
  The total of the data is obtained in byte units.
- SUM (one byte): Sum check for data being transmitted

Error Response | H'CC | H'52 |

- Error Response, H'CC, (one byte): Response to blank check for user boot MAT
- Error Code, H'52, (one byte): Erasure has not been completed.

(j) User MAT Blank Check

The boot program will check whether or not all user MATs are blank and return the resu

Command | H'4D |

- Command, H'4D, (one byte): Blank check for user MATs

Response | H'06 |

- Response, H'06, (one byte): Response to the blank check for user MATs
  If the contents of all user MATs are blank (H'FF), the boot program will return ACK

Error Response | H'CD | H'52 |

- Error Response, H'CD, (one byte): Error response to the blank check of user MATs.
- Error code, H'52, (one byte): Erasure has not been completed.

- Status (one byte): State of the boot program
- ERROR (one byte): Error status

  ERROR = 0 indicates normal operation.
  ERROR = 1 indicates error has occurred.

- SUM (one byte): Sum check

Table 18.17 Status Code

Code	Description
H'11	Device selection wait
H'12	Clock mode selection wait
H'13	Bit rate selection wait
H'1F	Programming/erasing state transition wait (bit rate selection is completed)
H'31	Programming state for erasure
H'3F	Programming/erasing selection wait (erasure is completed)
H'4F	Program data receive wait
H'5F	Erase block specification wait (erasure is completed)

H'26	Multiplication ratio error
H'27	Operating frequency error
H'29	Block number error
H'2A	Address error
H'2B	Data length error
H'51	Erasure error
H'52	Erasure incomplete error
H'53	Programming error
H'54	Selection processing error
H'80	Command error
H'FF	Bit-rate-adjustment confirmation error

3.3-V programming voltage. Use only the specified socket adapter.

5. Do not remove the chip from the PROM programmer nor input a reset signal during programming/erasing in which a high voltage is applied to the flash memory. Doing s damage the flash memory permanently. If a reset is input accidentally, the reset must released after the reset input period of at least 100µs.
6. The flash memory is not accessible until FKEY is cleared after programming/erasing the operating mode is changed and this LSI is restarted by a reset immediately after programming/erasing has finished, secure the reset input period (period of $\overline{\text{RES}}$ = 0) c 100µs. Transition to the reset state during programming/erasing is inhibited. If a reset accidentally, the reset must be released after the reset input period of at least 100µs.
7. At powering on or off the Vcc power supply, fix the $\overline{\text{RES}}$ pin to low and set the flash to hardware protection state. This power on/off timing must also be satisfied at a pow power-on caused by a power failure and other factors.
8. In on-board programming mode or programmer mode, programming of the 128-byte programming-unit block must be performed only once. Perform programming in the s where the programming-unit block is fully erased.
9. When the chip is to be reprogrammed with the programmer after execution of progra erasure in on-board programming mode, it is recommended that automatic programm performed after execution of automatic erasure.
10. To program the flash memory, the program data and program must be allocated to ad which are higher than those of the external interrupt vector table and H'FF must be wr all the system reserved areas in the exception handling vector table.
11. The programming program that includes the initialization routine and the erasing prog includes the initialization routine are each 4 Kbytes or less. Accordingly, when the CP frequency is 35 MHz, the download for each program takes approximately 60 µs at th maximum.

after immediately setting it to 1. Otherwise, download cannot be performed normally
Immediately after executing the instruction to set the SCO bit to 1, dummy read of th
must be executed twice.

15. The contents of general registers ER0 and ER1 are not saved during download of an
program, initialization, programming, or erasure. When needed, save the general reg
before a download request or before execution of initialization, programming, or era
the procedure program.

This LSI supports three types of clocks: a system clock provided to the CPU and bus ma
peripheral module clock provided to the peripheral modules, and an external bus clock p
the external bus. These clocks can be specified independently. Note, however, that the fr
of the peripheral clock and external bus clock are lower than that of the system clock.

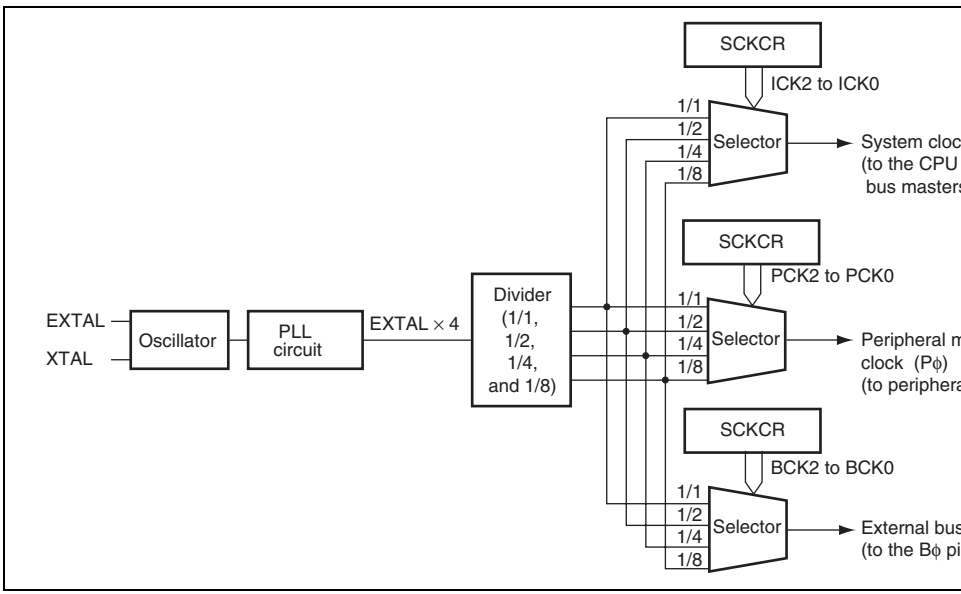


Figure 19.1 Block Diagram of Clock Pulse Generator

Bit	15	14	13	12	11	10	9
Bit Name	PSTOP1	—	—	—	—	ICK2	ICK1
Initial Value	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1
Bit Name	—	PCK2	PCK1	PCK0	—	BCK2	BCK1
Initial Value	0	0	1	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	PSTOP1	0	R/W	B ϕ Clock Output Enable Controls Bϕ output on PA7. Normal operation 0: Bϕ output 1: Fixed high
14 to 11	—	All 0	R/W	Reserved These bits are always read as 0. The write value always be 0.

				The frequencies of the peripheral module clock external bus clock change to the same frequen system clock if the frequency of the system cloc than that of the two clocks.
7	—	0	R/W	Reserved This bit is always read as 0. The write value sh always be 0.
6	PCK2	0	R/W	Peripheral Module Clock (Pϕ) Select
5	PCK1	1	R/W	These bits select the frequency of the periphera
4	PCK0	0	R/W	clock. The ratio to the input clock is as follows: 000: × 4 001: × 2 010: × 1 011: × 1/2 1XX: Setting prohibited The frequency of the peripheral module clock s lower than that of the system clock. Though the can be set so as to make the frequency of the p module clock higher than that of the system clo clocks will have the same frequency in reality.
3	—	0	R/W	Reserved This bit is always read as 0. The write value sh always be 0.

	The frequency of the external bus clock should b than that of the system clock. Though these bits set so as to make the frequency of the external b higher than that of the system clock, the clocks w the same frequency in reality.

Note: X: Don't care

frequency of 8 to 18 MHz should be connected.

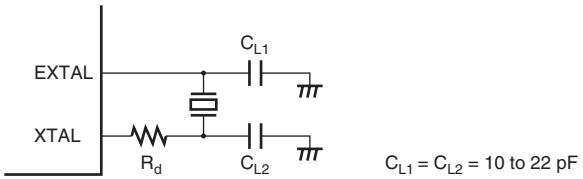


Figure 19.2 Connection of Crystal Resonator (Example)

Table 19.1 Damping Resistance Value

Frequency (MHz)	8	12	18
R_d (Ω)	200	0	0

Figure 19.3 shows an equivalent circuit of the crystal resonator. Use a crystal resonator t
the characteristics shown in table 19.2.

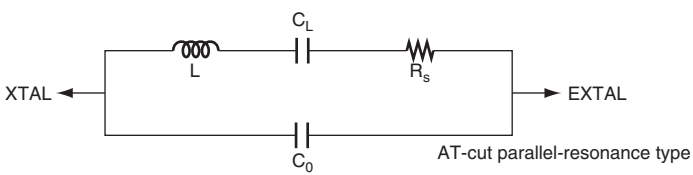


Figure 19.3 Crystal Resonator Equivalent Circuit

input to the XTAL pin, make sure that the external clock is held high in standby mode.

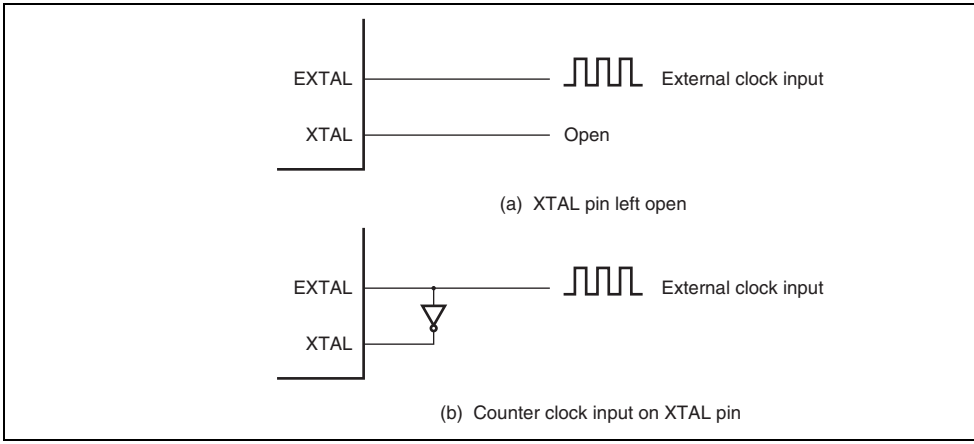


Figure 19.4 External Clock Input (Examples)

For the input conditions of the external clock, refer to table 22.4 in section 22.3.1, Clock
The input external clock should be from 8 to 18 MHz.

19.3 PLL Circuit

The PLL circuit has the function of multiplying the frequency of the clock from the oscill
factor of 4. The frequency multiplication factor is fixed. The phase difference is controlle
the timing of the rising edge of the internal clock is the same as that of the EXTAL pin si

1. The following points should be noted since the frequency of ϕ (Iϕ: system clock, Pϕ: peripheral module clock, Bϕ: external bus clock) supplied to each module changes a to the setting of SCKCR.

   Select a clock division ratio that is within the operation guaranteed range of clock cy t_{cyc} shown in the AC timing of electrical characteristics.

   For example, the following settings are not permitted under the conditions of 8 MHz MHz, 8 MHz ≤ Pϕ ≤ 35 MHz, and 8 MHz ≤ Bϕ ≤ 35 MHz: Iϕ < 8 MHz, 35 MHz < I MHz, 35 MHz < Pϕ, Bϕ < 8 MHz, and 35 MHz < Bϕ.

2. All the on-chip peripheral modules (except for the DMAC and DTC) operate on the Therefore, note that the time processing of modules such as a timer and SCI differs b after changing the clock division ratio.

   In addition, wait time for clearing software standby mode differs by changing the clo division ratio. For details, see section 20.7.3, Setting Oscillation Settling Time after Software Standby Mode.

3. The relationship among the system clock, peripheral module clock, and external bus ≥ Pϕ and Iϕ ≥ Bϕ. In addition, the system clock setting has the highest priority. Acco Pϕ or Bϕ may have the frequency set by bits ICK2 to ICK0 regardless of the settings PCK2 to PCK0 or BCK2 to BCK0.

4. Figure 19.5 shows the clock modification timing. After a value is written to SCKCR waits for the current bus cycle to complete. After the current bus cycle completes, ea frequency will be modified within one cycle (worst case) of the external input clock.

Operating clock
specified in SCKCR

Operating clock changed

Figure 19.5 Clock Modification Timing

19.5.2 Notes on Resonator

Since various characteristics related to the resonator are closely linked to the user's board thorough evaluation is necessary on the user's part, using the resonator connection examp shown in this section as a reference. As the parameters for the resonator will depend on th floating capacitance of the resonator and the mounting circuit, the parameters should be determined in consultation with the resonator manufacturer. The design must ensure that exceeding the maximum rating is not applied to the resonator pin.

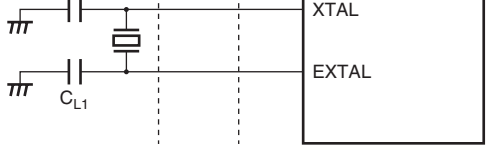
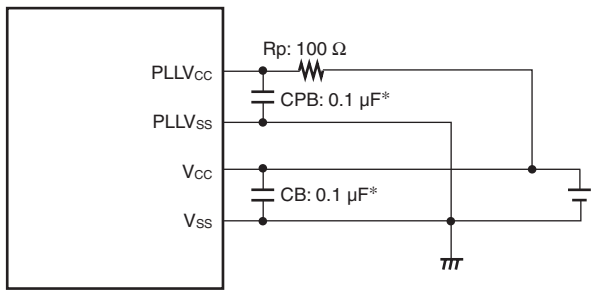


Figure 19.6 Note on Board Design for Oscillation Circuit

Figure 19.7 shows the external circuitry recommended for the PLL circuit. Separate PLL
PLLVss from the other Vcc and Vss lines at the board power supply source, and be sure
bypass capacitors CPB and CB close to the pins.



Note: * CB and CPB are laminated ceramic capacitors.

Figure 19.7 Recommended External Circuitry for PLL Circuit

- Module stop function

  The functions for each peripheral modules can be stopped to make a transition to a p
  down state.
- Transition function to power-down mode

  Transition to a power-down mode is possible to stop the CPU, peripheral modules, a
  oscillator.
- Four power-down modes

  Sleep mode

  All-module-clock-stop mode

  Software standby mode

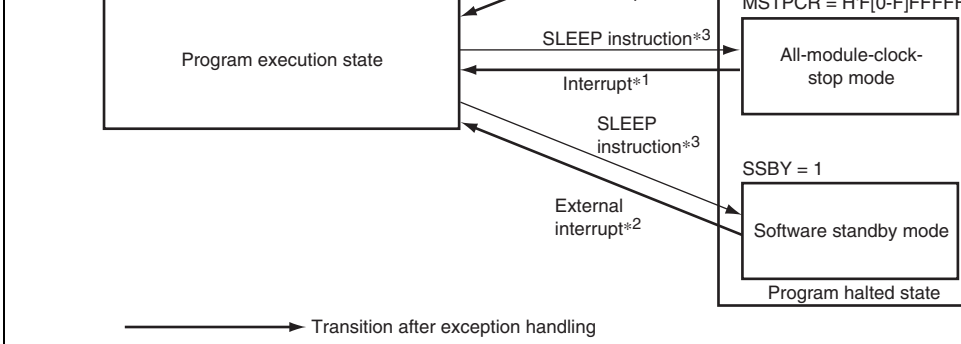
  Hardware standby mode

Table 20.1 shows conditions for making a transition to a power-down mode, states of th
peripheral modules, and clearing method for each mode. After the reset state, since this
operates in normal program execution state, the modules, other than the DMAC and DT
stopped.

8-bit timer	Functions	Functions*4	Halted (retained)	Halted
Other peripheral modules	Functions	Halted*1	Halted*1	Halted*3
I/O port	Functions	Retained	Retained	Hi-Z

Notes: "Halted (retained)" in the table means that the internal register values are retained internal operations are suspended.

1. SCI enters the reset state, and other peripheral modules retain their states.
2. External interrupt and some internal interrupts (8-bit timer and watchdog timer)
3. All peripheral modules enter the reset state.
4. "Functions" or "Halted" is selectable through the setting of bits MSTPA11 to MS MSTPCRA. However, pin output is disabled even when "Functions" is selected



Notes: From any state, a transition to hardware standby mode occurs when $\overline{\text{STBY}}$ is driven low.
From any state except for hardware standby mode, a transition to the reset state occurs when $\overline{\text{RES}}$ is d
1. NMI, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ11}}$, 8-bit timer interrupts, and watchdog timer interrupts.
The 8-bit timer is valid when bits MSTPCRA11 to MSTPCRA8 are all cleared to 0.
2. NMI and $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ11}}$. Note that IRQ is valid only when the corresponding bit in SSIER is set to 1
3. The SLPIE bit is 0.

Figure 20.1 Mode Transitions

20.2 Register Descriptions

The registers related to the power-down modes are shown below. For details on the syst
control register (SCKCR), refer to section 19.1.1, System Clock Control Register (SCK

- Standby control register (SBYCR)
- Module stop control register A (MSTPCRA)
- Module stop control register B (MSTPCRB)
- Module stop control register C (MSTPCRC)

Bit Name	SLPIE	—	—	—	—	—	—
Initial Value	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	SSBY	0	R/W	Software Standby Specifies the transition mode after executing the instruction 0: Shifts to sleep mode after the SLEEP instructi executed 1: Shifts to software standby mode after the SLE instruction is executed This bit remains set to 1 when software standby cleared by using external interrupts and a transit normal operation is made. For clearing, write 0 t When the WDT is used as the watchdog timer, t of this bit is disabled. In this case, a transition is made to sleep mode or all-module-clock-stop mo the SLEEP instruction is executed. This bit should be cleared to 0 when setting the to 1.
14	OPE	1	R/W	Output Port Enable Specifies whether the output of the address bus control signals ($\overline{CS0}$ to $\overline{CS7}$, $\overline{AS}$, $\overline{RD}$, $\overline{HWR}$, and retained or set to the high-impedance state in so standby mode. 0: In software standby mode, address bus and b control signals are high-impedance 1: In software standby mode, address bus and b control signals retain output state

circuit settling time is necessary. Refer to table
the standby time.

While oscillation is being settled, the timer is co
the Pϕ clock frequency. Careful consideration is
in multi-clock mode.

00000: Reserved

00001: Reserved

00010: Reserved

00011: Reserved

00100: Reserved

00101: Standby time = 64 states

00110: Standby time = 512 states

00111: Standby time = 1024 states

01000: Standby time = 2048 states

01001: Standby time = 4096 states

01010: Standby time = 16384 states

01011: Standby time = 32768 states

01100: Standby time = 65536 states

01101: Standby time = 131072 states

01110: Standby time = 262144 states

01111: Standby time = 524288 states

1XXXX: Reserved

				remains set to 1. Writing 0 clears this bit.
6 to 0	—	All 0	R/W	Reserved These bits are always read as 0. The write value always be 0.

Bit	7	6	5	4	3	2	1
Bit Name	MSTPA7	MSTPA6	MSTPA5	MSTPA4	MSTPA3	MSTPA2	MSTPA1
Initial Value	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- MSTPCRB

Bit	15	14	13	12	11	10	9
Bit Name	MSTPB15	MSTPB14	MSTPB13	MSTPB12	MSTPB11	MSTPB10	MSTPB9
Initial Value	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1
Bit Name	MSTPB7	MSTPB6	MSTPB5	MSTPB4	MSTPB3	MSTPB2	MSTPB1
Initial Value	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

				1: All-module-clock-stop mode enabled
14	MSTPA14	0	R/W	Reserved
13	MSTPA13	0	R/W	These bits are always read as 0. The write value always be 0.
12	MSTPA12	0	R/W	Data transfer controller (DTC)
11	MSTPA11	1	R/W	Reserved
10	MSTPA10	1	R/W	These bits are always read as 1. The write value always be 1.
9	MSTPA9	1	R/W	8-bit timer (TMR_3 and TMR_2)
8	MSTPA8	1	R/W	8-bit timer (TMR_1 and TMR_0)
7	MSTPA7	1	R/W	Reserved
6	MSTPA6	1	R/W	These bits are always read as 1. The write value always be 1.
5	MSTPA5	1	R/W	D/A converter (channels 1 and 0)
4	MSTPA4	1	R/W	Reserved This bit is always read as 1. The write value sho always be 1.
3	MSTPA3	1	R/W	A/D converter (unit 0)
2	MSTPA2	1	R/W	Reserved
1	MSTPA1	1	R/W	These bits are always read as 1. The write value always be 1.
0	MSTPA0	1	R/W	16-bit timer pulse unit (TPU channels 5 to 0)

				always be 1.
10	MSTPB10	1	R/W	Serial communication interface_2 (SCI_2)
9	MSTPB9	1	R/W	Serial communication interface_1 (SCI_1)
8	MSTPB8	1	R/W	Serial communication interface_0 (SCI_0)
7	MSTPB7	1	R/W	Reserved
6	MSTPB6	1	R/W	These bits are always read as 1. The write valu
5	MSTPB5	1	R/W	always be 1.
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	

Bit	7	6	5	4	3	2	1	0
Bit Name	MSTPC7	MSTPC6	MSTPC5	MSTPC4	MSTPC3	MSTPC2	MSTPC1	M
Initial Value	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Bit	Bit Name	Initial Value	R/W	Module
15	MSTPC15	1	R/W	Reserved
14	MSTPC14	1	R/W	These bits are always read as 1. The write value always be 1.
13	MSTPC13	1	R/W	
12	MSTPC12	1	R/W	
11	MSTPC11	1	R/W	
10	MSTPC10	1	R/W	
9	MSTPC9	1	R/W	
8	MSTPC8	1	R/W	
7	MSTPC7	0	R/W	Reserved
6	MSTPC6	0	R/W	These bits are always read as 0. The write value always be 0.
5	MSTPC5	0	R/W	
4	MSTPC4	0	R/W	
3	MSTPC3	0	R/W	
2	MSTPC2	0	R/W	On-chip RAM_2 (H'FF6000 to H'FF7FFF)
1	MSTPC1	0	R/W	On-chip RAM_1 (H'FF8000 to H'FF9FFF)
0	MSTPC0	0	R/W	On-chip RAM_0 (H'FFA000 to H'FFBFFF)

20.4 Module Stop Function

Module stop function can be set for individual on-chip peripheral modules.

When the corresponding MSTP bit in MSTPCRA, MSTPCRB, or MSTPCRC is set to 1 operation stops at the end of the bus cycle and a transition is made to the module stop st CPU continues operating independently.

When the corresponding MSTP bit is cleared to 0, the module stop state is cleared and t starts operating at the end of the bus cycle. In the module stop state, the internal states o other than the SCI are retained.

After the reset state is cleared, all modules other than the DMAC, DTC, and on-chip RA the module stop state.

The registers of the module for which the module stop state is selected cannot be read fr written to.

20.5 Sleep Mode

20.5.1 Transition to Sleep Mode

When the SLEEP instruction is executed when the SSBY bit in SBYCR is 0, the CPU e mode. In sleep mode, CPU operation stops but the contents of the CPU's internal registe retained. Other peripheral functions do not stop.

driving the $\overline{\text{RES}}$ pin high makes the CPU start the reset exception processing.

3. Clearing by $\overline{\text{STBY}}$ pin

   When the $\overline{\text{STBY}}$ pin level is driven low, a transition is made to hardware standby mod

4. Clearing by reset caused by watchdog timer overflow

   Sleep mode is exited by an internal reset caused by a watchdog timer overflow.

normal program execution state via the exception handling state. All-module-clock-stop
not cleared if interrupts are disabled, if interrupts other than NMI are masked on the CP
if the relevant interrupt is designated as a DTC activation source.

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

Note: * Operation or halting of the 8-bit timer can be selected by bits MSTPA11 to M
MSTPCRA.

consumption to be significantly reduced.

If the WDT is used as a watchdog timer, it is impossible to make a transition to software
mode. The WDT should be stopped before the SLEEP instruction execution.

20.7.2 Clearing Software Standby Mode

Software standby mode is cleared by an external interrupt (NMI pin, or pins $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ}}$
by means of the $\overline{\text{RES}}$ pin or $\overline{\text{STBY}}$ pin.

1. Clearing by interrupt

   When an NMI or IRQ0 to IRQ11* interrupt request signal is input, clock oscillation s
   after the elapse of the time set in bits STS4 to STS0 in SBYCR, stable clocks are supp
   the entire LSI, software standby mode is cleared, and interrupt exception handling is s

   When clearing software standby mode with an IRQ0 to IRQ11* interrupt, set the
   corresponding enable bit to 1 and ensure that no interrupt with a higher priority than i
   IRQ0 to IRQ11* is generated. Software standby mode cannot be cleared if the interru
   been masked on the CPU side or has been designated as a DTC activation source.

   Note: * By setting the SSIn bit in SSIER to 1, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ11}}$ can be used as a soft
   standby mode clearing source.

2. Clearing by $\overline{\text{RES}}$ pin

   When the $\overline{\text{RES}}$ pin is driven low, clock oscillation is started. At the same time as cloc
   oscillation starts, clocks are supplied to the entire LSI. Note that the $\overline{\text{RES}}$ pin must be
   until clock oscillation settles. When the $\overline{\text{RES}}$ pin goes high, the CPU begins reset exce
   handling.

3. Clearing by $\overline{\text{STBY}}$ pin

   When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

Table 20.2 Oscillation Settling Time Settings

STS4	STS3	STS2	STS1	STS0	Standby Time	Pϕ* [MHz] 35	25	20
0	0	0	0	0	Reserved	—	—	—
				1	Reserved	—	—	—
			1	0	Reserved	—	—	—
				1	Reserved	—	—	—
		1	0	0	Reserved	—	—	—
				1	64	1.8	2.6	3.2
			1	0	512	14.6	20.5	25.6
				1	1024	29.3	41.0	51.2
	1	0	0	0	2048	58.5	81.9	102.4
				1	4096	0.12	0.16	0.20
			1	0	16384	0.47	0.66	0.82
				1	32768	0.94	1.31	1.64
		1	0	0	65536	1.87	2.62	3.28
				1	131072	3.74	5.24	6.55
			1	0	262144	7.49	10.49	13.11
				1	524288	14.98	20.97	26.21
1	0	0	0	0	Reserved	—	—	—

: Recommended time setting when using a crystal resonator.

: Recommended time setting when using an external clock.

Note: * Pϕ is the output from the peripheral module frequency divider.

				1	1024	78.8	102.4	128.0
	1	0	0	0	2048	157.5	204.8	256.0
				1	4096	0.32	0.41	0.51
			1	0	16384	1.26	1.64	2.05
				1	32765	2.52	3.28	4.10
		1	0	0	65536	5.04	6.55	8.19
				1	131072	10.08	13.11	16.38
			1	0	262144	20.16	26.21	32.77
				1	524288	40.33	52.43	65.54
1	0	0	0	0	Reserved	—	—	—

: Recommended time setting when using a crystal resonator.

: Recommended time setting when using an external clock.

Note: * ϕ is the output from the peripheral module frequency divider.

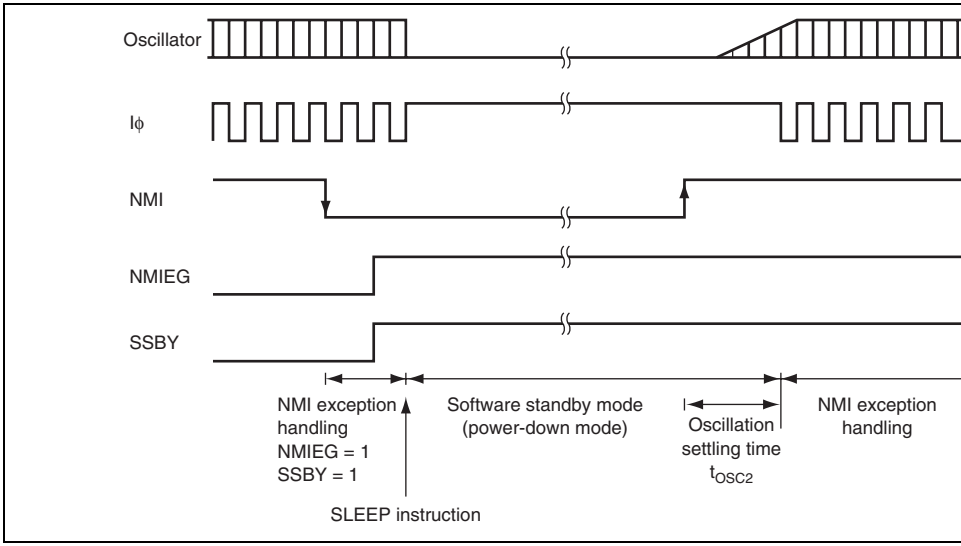


Figure 20.2 Software Standby Mode Application Example

In order to retain on-chip RAM data, the RAME bit in SYSCR should be cleared to 0 bef
driving the $\overline{\text{STBY}}$ pin low. Do not change the state of the mode pins (MD2 to MD0) whil
LSI is in hardware standby mode.

20.8.2 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
pin is driven high while the $\overline{\text{RES}}$ pin is low, the reset state is entered and clock oscillation
started. Ensure that the $\overline{\text{RES}}$ pin is held low until clock oscillation settles (for details on th
oscillation settling time, refer to table 20.2). When the $\overline{\text{RES}}$ pin is subsequently driven hig
transition is made to the program execution state via the reset exception handling state.

20.8.3 Hardware Standby Mode Timing

Figure 20.3 shows an example of hardware standby mode timing.

When the $\overline{\text{STBY}}$ pin is driven low after the $\overline{\text{RES}}$ pin has been driven low, a transition is m
hardware standby mode. Hardware standby mode is cleared by driving the $\overline{\text{STBY}}$ pin high
waiting for the oscillation settling time, then changing the $\overline{\text{RES}}$ pin from low to high.

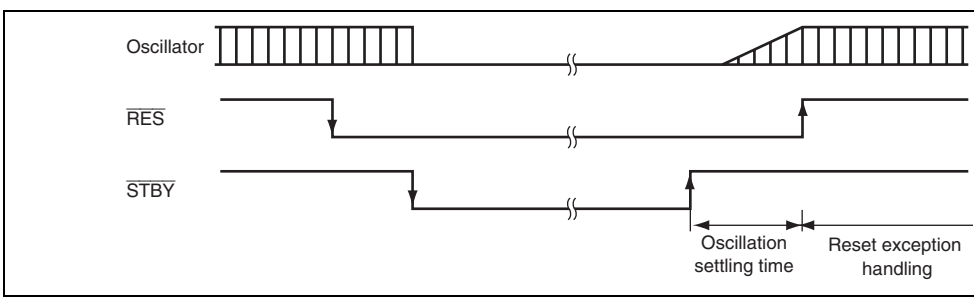


Figure 20.3 Hardware Standby Mode Timing

Timing.

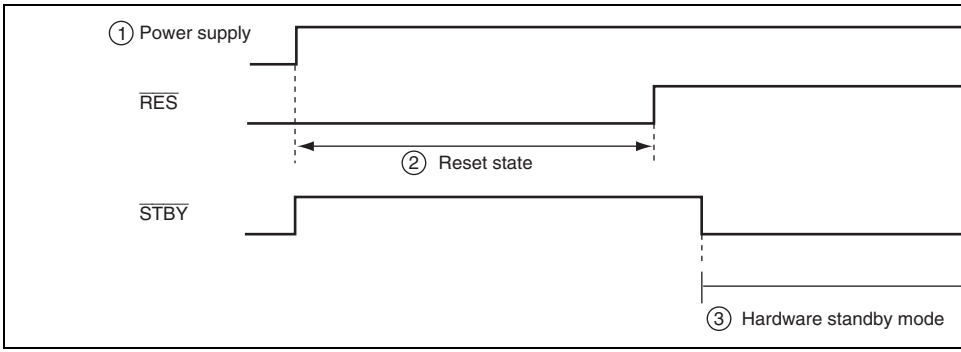


Figure 20.4 Timing Sequence at Power-On

Transitions to the power-down state are inhibited when sleep instruction exception handli
initiated, and the CPU immediately starts sleep instruction exception handling.

When a SLEEP instruction is executed while the SLPIE bit is cleared to 0, a transition is
the power-down state. The power-down state is canceled by a canceling factor interrupt (
20.5).

When a canceling factor interrupt is generated immediately before the execution of a SLE
instruction, exception handling for the interrupt starts. When execution returns from the e
service routine, the SLEEP instruction is executed to enter the power-down state. In this c
power-down state is not canceled until the next canceling factor interrupt is generated (se
20.6).

When the SLPIE bit is set to 1 in the service routine for a canceling factor interrupt so tha
execution of a SLEEP instruction will produce sleep instruction exception handling, the c
of the system is as shown in figure 20.7. Even if a canceling factor interrupt is generated
immediately before the SLEEP instruction is executed, sleep instruction exception handli
initiated by execution of the SLEEP instruction. Therefore, the CPU executes the instruct
follows the SLEEP instruction after sleep instruction exception and exception service rou
without shifting to the power-down state.

When the SLPIE bit is set to 1 to start sleep exception handling, clear the SSBY bit in SB
to 0.

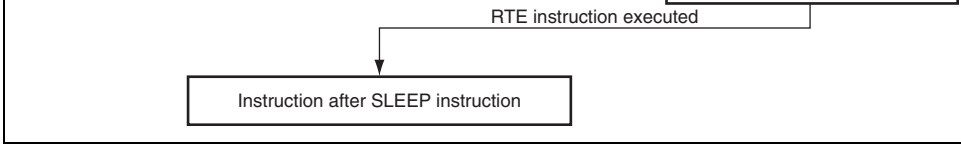


Figure 20.5 When Canceling Factor Interrupt is Generated after SLEEP Instruction Execution

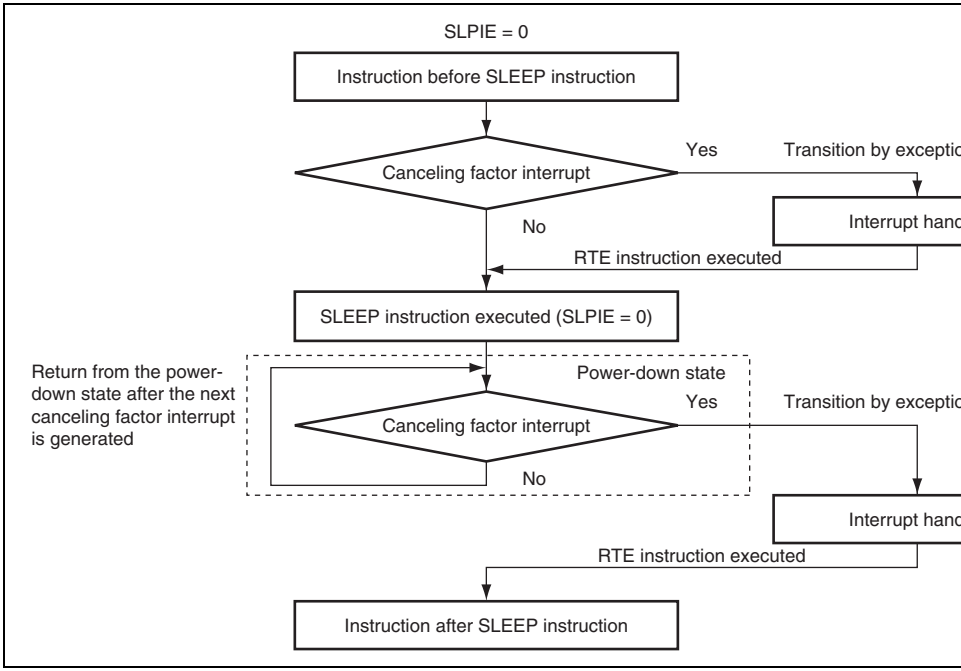


Figure 20.6 When Canceling Factor Interrupt is Generated before SLEEP Instruction Execution (Sleep Instruction Exception Handling Not Initi

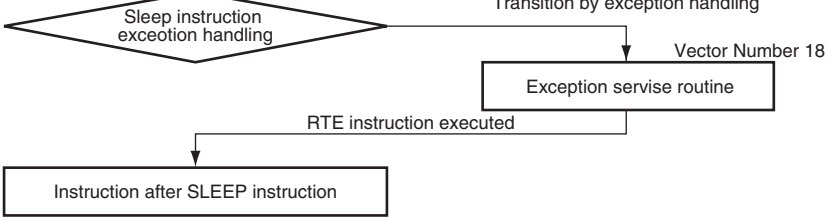


Figure 20.7 When Canceling Factor Interrupt is Generated before SLEEP Instruction Execution (Sleep Instruction Exception Handling Initiate

Table 20.3 Bϕ Pin (PA7) State in Each Processing State

Register Setting Value			Normal Operating State	Sleep Mode	All-Module-Clock-Stop Mode	Software Standby Mode	
DDR	PSTOP1	POSEL1				OPE = 0	OPE = 1
0	X	X	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
1	0	0	Bϕ output	Bϕ output	Bϕ output	High	High
1	0	1	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
1	1	X	High	High	High	High	High

20.11.3 Module Stop of DMAC or DTC

Depending on the operating state of the DMAC and DTC, bits MSTPA13 and MSTPA12 be set to 1, respectively. The module stop state setting for the DMAC or DTC should be c out only when the DMAC or DTC is not activated.

For details, refer to section 7, DMA Controller (DMAC), and section 8, Data Transfer Co (DTC).

20.11.4 On-Chip Peripheral Module Interrupts

Relevant interrupt operations cannot be performed in module stop mode. Consequently, i module stop state is entered when an interrupt has been requested, it will not be possible t the CPU interrupt source or the DMAC /DTC activation source. Interrupts should therefo disabled before entering the module stop state.

20.11.5 Writing to MSTPCRA, MSTPCRB, and MSTPCRC

MSTPCRA, MSTPCRB, and MSTPCRC should only be written to by the CPU.

clock. For details, refer to section 6.5.4, External Bus Interface.

- Among the internal I/O register area, addresses not listed in the list of registers are u
or reserved addresses. Undefined and reserved addresses cannot be accessed. Do not
these addresses; otherwise, the operation when accessing these bits and subsequent c
cannot be guaranteed.

2. Register bits

- Bit configurations of the registers are listed in the same order as the register addresse
- Reserved bits are indicated by — in the bit name column.
- Space in the bit name field indicates that the entire register is allocated to either the c
data.
- For the registers of 16 or 32 bits, the MSB is listed first.

  Byte configuration description order is subject to big endian.

3. Register states in each operating mode

- Register states are listed in the same order as the register addresses.
- For the initialized state of each bit, refer to the register description in the correspond
section.
- The register states shown here are for the basic operating modes. If there is a specific
an on-chip peripheral module, refer to the section on that on-chip peripheral module.

Port D data direction register	PDDDR	8	H'FFB8C	I/O ports	8	2P
Port E data direction register	PEDDR	8	H'FFB8D	I/O ports	8	2P
Port F data direction register	PFDDR	8	H'FFB8E	I/O ports	8	2P
Port 1 input buffer control register	P1ICR	8	H'FFB90	I/O ports	8	2P
Port 2 input buffer control register	P2ICR	8	H'FFB91	I/O ports	8	2P
Port 3 input buffer control register	P3ICR	8	H'FFB92	I/O ports	8	2P
Port 5 input buffer control register	P5ICR	8	H'FFB94	I/O ports	8	2P
Port 6 input buffer control register	P6ICR	8	H'FFB95	I/O ports	8	2P
Port A input buffer control register	PAICR	8	H'FFB99	I/O ports	8	2P
Port B input buffer control register	PBICR	8	H'FFB9A	I/O ports	8	2P
Port D input buffer control register	PDICR	8	H'FFB9C	I/O ports	8	2P
Port E input buffer control register	PEICR	8	H'FFB9D	I/O ports	8	2P
Port F input buffer control register	PFICR	8	H'FFB9E	I/O ports	8	2P
Port H register	PORTH	8	H'FFBA0	I/O ports	8	2P
Port I register	PORTI	8	H'FFBA1	I/O ports	8	2P
Port H data register	PHDR	8	H'FFBA4	I/O ports	8	2P
Port I data register	PIDR	8	H'FFBA5	I/O ports	8	2P
Port H data direction register	PHDDR	8	H'FFBA8	I/O ports	8	2P
Port I data direction register	PIDDR	8	H'FFBA9	I/O ports	8	2P
Port H input buffer control register	PHICR	8	H'FFBAC	I/O ports	8	2P
Port I input buffer control register	PIICR	8	H'FFBAD	I/O ports	8	2P
Port D pull-up MOS control register	PDPCR	8	H'FFBB4	I/O ports	8	2P
Port E pull-up MOS control register	PEPCR	8	H'FFBB5	I/O ports	8	2P
Port F pull-up MOS control register	PFPCR	8	H'FFBB6	I/O ports	8	2P
Port H pull-up MOS control register	PHPCR	8	H'FFBB8	I/O ports	8	2P

Port function control register 6	PFCR6	8	H'FFBC6	I/O ports	8	2
Port function control register 7	PFCR7	8	H'FFBC7	I/O ports	8	2
Port function control register 9	PFCR9	8	H'FFBC9	I/O ports	8	2
Port function control register B	PFCRB	8	H'FFBCB	I/O ports	8	2
Port function control register C	PFCRC	8	H'FFBCC	I/O ports	8	2
Software standby release IRQ enable register	SSIER	16	H'FFBCE	INTC	8	2
DMA source address register_0	DSAR_0	32	H'FFC00	DMAC_0	16	2
DMA destination address register_0	DDAR_0	32	H'FFC04	DMAC_0	16	2
DMA offset register_0	DOFR_0	32	H'FFC08	DMAC_0	16	2
DMA transfer count register_0	DTCR_0	32	H'FFC0C	DMAC_0	16	2
DMA block size register_0	DBSR_0	32	H'FFC10	DMAC_0	16	2
DMA mode control register_0	DMDR_0	32	H'FFC14	DMAC_0	16	2
DMA address control register_0	DACR_0	32	H'FFC18	DMAC_0	16	2
DMA source address register_1	DSAR_1	32	H'FFC20	DMAC_1	16	2
DMA destination address register_1	DDAR_1	32	H'FFC24	DMAC_1	16	2
DMA offset register_1	DOFR_1	32	H'FFC28	DMAC_1	16	2
DMA transfer count register_1	DTCR_1	32	H'FFC2C	DMAC_1	16	2
DMA block size register_1	DBSR_1	32	H'FFC30	DMAC_1	16	2
DMA mode control register_1	DMDR_1	32	H'FFC34	DMAC_1	16	2
DMA address control register_1	DACR_1	32	H'FFC38	DMAC_1	16	2
DMA source address register_2	DSAR_2	32	H'FFC40	DMAC_2	16	2
DMA destination address register_2	DDAR_2	32	H'FFC44	DMAC_2	16	2

register_3						
DMA offset register_3	DOFR_3	32	H'FFC68	DMAC_3	16	2Iϕ
DMA transfer count register_3	DTCR_3	32	H'FFC6C	DMAC_3	16	2Iϕ
DMA block size register_3	DBSR_3	32	H'FFC70	DMAC_3	16	2Iϕ
DMA mode control register_3	DMDR_3	32	H'FFC74	DMAC_3	16	2Iϕ
DMA address control register_3	DACR_3	32	H'FFC78	DMAC_3	16	2Iϕ
DMA module request select register_0	DMRSR_0	8	H'FFD20	DMAC_0	16	2Iϕ
DMA module request select register_1	DMRSR_1	8	H'FFD21	DMAC_1	16	2Iϕ
DMA module request select register_2	DMRSR_2	8	H'FFD22	DMAC_2	16	2Iϕ
DMA module request select register_3	DMRSR_3	8	H'FFD23	DMAC_3	16	2Iϕ
Interrupt priority register A	IPRA	16	H'FFD40	INTC	16	2Iϕ
Interrupt priority register B	IPRB	16	H'FFD42	INTC	16	2Iϕ
Interrupt priority register C	IPRC	16	H'FFD44	INTC	16	2Iϕ
Interrupt priority register E	IPRE	16	H'FFD48	INTC	16	2Iϕ
Interrupt priority register F	IPRF	16	H'FFD4A	INTC	16	2Iϕ
Interrupt priority register G	IPRG	16	H'FFD4C	INTC	16	2Iϕ
Interrupt priority register H	IPRH	16	H'FFD4E	INTC	16	2Iϕ
Interrupt priority register I	IPRI	16	H'FFD50	INTC	16	2Iϕ
Interrupt priority register K	IPRK	16	H'FFD54	INTC	16	2Iϕ
Interrupt priority register L	IPRL	16	H'FFD56	INTC	16	2Iϕ
IRQ sense control register H	ISCRH	16	H'FFD68	INTC	16	2Iϕ
IRQ sense control register L	ISCRL	16	H'FFD6A	INTC	16	2Iϕ

Idle control register	IDLCR	16	H'FFD90	BSC	16	2
Bus control register 1	BCR1	16	H'FFD92	BSC	16	2
Bus control register 2	BCR2	8	H'FFD94	BSC	16	2
Endian control register	ENDIANCR	8	H'FFD95	BSC	16	2
SRAM mode control register	SRAMCR	16	H'FFD98	BSC	16	2
Burst ROM interface control register	BROMCR	16	H'FFD9A	BSC	16	2
Address/data multiplexed I/O control register	MPXCR	16	H'FFD9C	BSC	16	2
RAM emulation register	RAMER	8	H'FFD9E	BSC	16	2
Mode control register	MDCR	16	H'FFDC0	SYSTEM	16	2
System control register	SYSCR	16	H'FFDC2	SYSTEM	16	2
System clock control register	SCKCR	16	H'FFDC4	SYSTEM	16	2
Standby control register	SBYCR	16	H'FFDC6	SYSTEM	16	2
Module stop control register A	MSTPCRA	16	H'FFDC8	SYSTEM	16	2
Module stop control register B	MSTPCRB	16	H'FFDCA	SYSTEM	16	2
Module stop control register C	MSTPCRC	16	H'FFDCC	SYSTEM	16	2
Serial extended mode register_2	SEMR_2	8	H'FFE84	SCI_2	8	2
Serial mode register_4	SMR_4	8	H'FFE90	SCI_4	8	2
Bit rate register_4	BRR_4	8	H'FFE91	SCI_4	8	2
Serial control register_4	SCR_4	8	H'FFE92	SCI_4	8	2
Transmit data register_4	TDR_4	8	H'FFE93	SCI_4	8	2
Serial status register_4	SSR_4	8	H'FFE94	SCI_4	8	2
Receive data register_4	RDR_4	8	H'FFE95	SCI_4	8	2
Smart card mode register_4	SCMR_4	8	H'FFE96	SCI_4	8	2

Timer control register_2	TCR_2	8	H'FFEC0	TMR_2	16	2P
Timer control register_3	TCR_3	8	H'FFEC1	TMR_3	16	2P
Timer control/status register_2	TCSR_2	8	H'FFEC2	TMR_2	16	2P
Timer control/status register_3	TCSR_3	8	H'FFEC3	TMR_3	16	2P
Time constant register A_2	TCORA_2	8	H'FFEC4	TMR_2	16	2P
Time constant register A_3	TCORA_3	8	H'FFEC5	TMR_3	16	2P
Time constant register B_2	TCORB_2	8	H'FFEC6	TMR_2	16	2P
Time constant register B_3	TCORB_3	8	H'FFEC7	TMR_3	16	2P
Timer counter_2	TCNT_2	8	H'FFEC8	TMR_2	16	2P
Timer counter_3	TCNT_3	8	H'FFEC9	TMR_3	16	2P
Timer counter control register_2	TCCR_2	8	H'FFECA	TMR_2	16	2P
Timer counter control register_3	TCCR_3	8	H'FFECB	TMR_3	16	2P
Timer control register_4	TCR_4	8	H'FFEE0	TPU_4	16	2P
Timer mode register_4	TMDR_4	8	H'FFEE1	TPU_4	16	2P
Timer I/O control register_4	TIOR_4	8	H'FFEE2	TPU_4	16	2P
Timer interrupt enable register_4	TIER_4	8	H'FFEE4	TPU_4	16	2P
Timer status register_4	TSR_4	8	H'FFEE5	TPU_4	16	2P
Timer counter_4	TCNT_4	16	H'FFEE6	TPU_4	16	2P
Timer general register A_4	TGRA_4	16	H'FFEE8	TPU_4	16	2P
Timer general register B_4	TGRB_4	16	H'FFEEA	TPU_4	16	2P
Timer control register_5	TCR_5	8	H'FFEF0	TPU_5	16	2P
Timer mode register_5	TMDR_5	8	H'FFEF1	TPU_5	16	2P
Timer I/O control register_5	TIOR_5	8	H'FFEF2	TPU_5	16	2P
Timer interrupt enable register_5	TIER_5	8	H'FFEF4	TPU_5	16	2P
Timer status register_5	TSR_5	8	H'FFEF5	TPU_5	16	2P
Timer counter_5	TCNT_5	16	H'FFEF6	TPU_5	16	2P

DTC control register	DTCCR	8	H'FFF30	INTC	16	2
Interrupt control register	INTCR	8	H'FFF32	INTC	16	2
CPU priority control register	CPUPCR	8	H'FFF33	INTC	16	2
IRQ enable register	IER	16	H'FFF34	INTC	16	2
IRQ status register	ISR	16	H'FFF36	INTC	16	2
Port 1 register	PORT1	8	H'FFF40	I/O ports	8	2
Port 2 register	PORT2	8	H'FFF41	I/O ports	8	2
Port 3 register	PORT3	8	H'FFF42	I/O ports	8	2
Port 5 register	PORT5	8	H'FFF44	I/O ports	8	2
Port 6 register	PORT6	8	H'FFF45	I/O ports	8	2
Port A register	PORTA	8	H'FFF49	I/O ports	8	2
Port B register	PORTB	8	H'FFF4A	I/O ports	8	2
Port D register	PORTD	8	H'FFF4C	I/O ports	8	2
Port E register	PORTE	8	H'FFF4D	I/O ports	8	2
Port F register	PORTF	8	H'FFF4E	I/O ports	8	2
Port 1 data register	P1DR	8	H'FFF50	I/O ports	8	2
Port 2 data register	P2DR	8	H'FFF51	I/O ports	8	2
Port 3 data register	P3DR	8	H'FFF52	I/O ports	8	2
Port 6 data register	P6DR	8	H'FFF55	I/O ports	8	2
Port A data register	PADR	8	H'FFF59	I/O ports	8	2
Port B data register	PBDR	8	H'FFF5A	I/O ports	8	2
Port D data register	PDDR	8	H'FFF5C	I/O ports	8	2
Port E data register	PEDR	8	H'FFF5D	I/O ports	8	2
Port F data register	PFDR	8	H'FFF5E	I/O ports	8	2
Serial mode register_2	SMR_2	8	H'FFF60	SCI_2	8	2
Bit rate register_2	BRR_2	8	H'FFF61	SCI_2	8	2

D/A control register 01	DACR01	8	H'FFF6A	D/A	8	2P
PPG output control register	PCR	8	H'FFF76	PPG	8	2P
PPG output mode register	PMR	8	H'FFF77	PPG	8	2P
Next data enable register H	NDERH	8	H'FFF78	PPG	8	2P
Next data enable register L	NDERL	8	H'FFF79	PPG	8	2P
Output data register H	PODRH	8	H'FFF7A	PPG	8	2P
Output data register L	PODRL	8	H'FFF7B	PPG	8	2P
Next data register H*	NDRH	8	H'FFF7C	PPG	8	2P
Next data register L*	NDRL	8	H'FFF7D	PPG	8	2P
Next data register H*	NDRH	8	H'FFF7E	PPG	8	2P
Next data register L*	NDRL	8	H'FFF7F	PPG	8	2P
Serial mode register_0	SMR_0	8	H'FFF80	SCI_0	8	2P
Bit rate register_0	BRR_0	8	H'FFF81	SCI_0	8	2P
Serial control register_0	SCR_0	8	H'FFF82	SCI_0	8	2P
Transmit data register_0	TDR_0	8	H'FFF83	SCI_0	8	2P
Serial status register_0	SSR_0	8	H'FFF84	SCI_0	8	2P
Receive data register_0	RDR_0	8	H'FFF85	SCI_0	8	2P
Smart card mode register_0	SCMR_0	8	H'FFF86	SCI_0	8	2P
Serial mode register_1	SMR_1	8	H'FFF88	SCI_1	8	2P
Bit rate register_1	BRR_1	8	H'FFF89	SCI_1	8	2P
Serial control register_1	SCR_1	8	H'FFF8A	SCI_1	8	2P
Transmit data register_1	TDR_1	8	H'FFF8B	SCI_1	8	2P
Serial status register_1	SSR_1	8	H'FFF8C	SCI_1	8	2P
Receive data register_1	RDR_1	8	H'FFF8D	SCI_1	8	2P
Smart card mode register_1	SCMR_1	8	H'FFF8E	SCI_1	8	2P

A/D data register H	ADDRH	16	H'FFF9E	A/D	16	2
A/D control/status register	ADCSR	8	H'FFFA0	A/D	16	2
A/D control register	ADCR	8	H'FFFA1	A/D	16	2
Timer control/status register	TCSR	8	H'FFFA4	WDT		2
Timer counter	TCNT	8	H'FFFA5	WDT		2
Reset control/status register	RSTCSR	8	H'FFFA7	WDT		2
Timer control register_0	TCR_0	8	H'FFFB0	TMR_0	16	2
Timer control register_1	TCR_1	8	H'FFFB1	TMR_1	16	2
Timer control/status register_0	TCSR_0	8	H'FFFB2	TMR_0	16	2
Timer control/status register_1	TCSR_1	8	H'FFFB3	TMR_1	16	2
Time constant register A_0	TCORA_0	8	H'FFFB4	TMR_0	16	2
Time constant register A_1	TCORA_1	8	H'FFFB5	TMR_1	16	2
Time constant register B_0	TCORB_0	8	H'FFFB6	TMR_0	16	2
Time constant register B_1	TCORB_1	8	H'FFFB7	TMR_1	16	2
Timer counter_0	TCNT_0	8	H'FFFB8	TMR_0	16	2
Timer counter_1	TCNT_1	8	H'FFFB9	TMR_1	16	2
Timer counter control register_0	TCCR_0	8	H'FFFBA	TMR_0	16	2
Timer counter control register_1	TCCR_1	8	H'FFFBB	TMR_1	16	2
Timer start register	TSTR	8	H'FFFBC	TPU	16	2
Timer synchronous register	TSYR	8	H'FFFBD	TPU	16	2
Timer control register_0	TCR_0	8	H'FFFC0	TPU_0	16	2
Timer mode register_0	TMDR_0	8	H'FFFC1	TPU_0	16	2
Timer I/O control register H_0	TIORH_0	8	H'FFFC2	TPU_0	16	2
Timer I/O control register L_0	TIORL_0	8	H'FFFC3	TPU_0	16	2

Timer control register_1	TCR_1	8	H'FFFD0	TPU_1	16	2P
Timer mode register_1	TMDR_1	8	H'FFFD1	TPU_1	16	2P
Timer I/O control register_1	TIOR_1	8	H'FFFD2	TPU_1	16	2P
Timer interrupt enable register_1	TIER_1	8	H'FFFD4	TPU_1	16	2P
Timer status register_1	TSR_1	8	H'FFFD5	TPU_1	16	2P
Timer counter_1	TCNT_1	16	H'FFFD6	TPU_1	16	2P
Timer general register A_1	TGRA_1	16	H'FFFD8	TPU_1	16	2P
Timer general register B_1	TGRB_1	16	H'FFFDA	TPU_1	16	2P
Timer control register_2	TCR_2	8	H'FFFE0	TPU_2	16	2P
Timer mode register_2	TMDR_2	8	H'FFFE1	TPU_2	16	2P
Timer I/O control register_2	TIOR_2	8	H'FFFE2	TPU_2	16	2P
Timer interrupt enable register_2	TIER_2	8	H'FFFE4	TPU_2	16	2P
Timer status register_2	TSR_2	8	H'FFFE5	TPU_2	16	2P
Timer counter_2	TCNT_2	16	H'FFFE6	TPU_2	16	2P
Timer general register A_2	TGRA_2	16	H'FFFE8	TPU_2	16	2P
Timer general register B_2	TGRB_2	16	H'FFFEA	TPU_2	16	2P
Timer control register_3	TCR_3	8	H'FFFF0	TPU_3	16	2P
Timer mode register_3	TMDR_3	8	H'FFFF1	TPU_3	16	2P
Timer I/O control register H_3	TIORH_3	8	H'FFFF2	TPU_3	16	2P
Timer I/O control register L_3	TIORL_3	8	H'FFFF3	TPU_3	16	2P
Timer interrupt enable register_3	TIER_3	8	H'FFFF4	TPU_3	16	2P
Timer status register_3	TSR_3	8	H'FFFF5	TPU_3	16	2P
Timer counter_3	TCNT_3	16	H'FFFF6	TPU_3	16	2P

0 and 1 by the PCR setting, the NDRL address is H'FFF7D. When different ou
triggers are specified, the NDRL addresses for pulse output groups 0 and 1 ar
and H'FFF7D, respectively.

P6DDR	—	—	P65DDR	P64DDR	P63DDR	P62DDR	P61DDR	P60DDR
PADDR	PA7DDR	PA6DDR	PA5DDR	PA4DDR	PA3DDR	PA2DDR	PA1DDR	PA0DDR
PBDDR	—	—	—	—	PB3DDR	PB2DDR	PB1DDR	PB0DDR
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
P1ICR	P17ICR	P16ICR	P15ICR	P14ICR	P13ICR	P12ICR	P11ICR	P10ICR
P2ICR	P27ICR	P26ICR	P25ICR	P24ICR	P23ICR	P22ICR	P21ICR	P20ICR
P3ICR	P37ICR	P36ICR	P35ICR	P34ICR	P33ICR	P32ICR	P31ICR	P30ICR
P5ICR	P57ICR	P56ICR	P55ICR	P54ICR	P53ICR	P52ICR	P51ICR	P50ICR
P6ICR	—	—	P65ICR	P64ICR	P63ICR	P62ICR	P61ICR	P60ICR
PAICR	PA7ICR	PA6ICR	PA5ICR	PA4ICR	PA3ICR	PA2ICR	PA1ICR	PA0ICR
PBICR	—	—	—	—	PB3ICR	PB2ICR	PB1ICR	PB0ICR
PDICR	PD7ICR	PD6ICR	PD5ICR	PD4ICR	PD3ICR	PD2ICR	PD1ICR	PD0ICR
PEICR	PE7ICR	PE6ICR	PE5ICR	PE4ICR	PE3ICR	PE2ICR	PE1ICR	PE0ICR
PFICR	PF7ICR	PF6ICR	PF5ICR	PF4ICR	PF3ICR	PF2ICR	PF1ICR	PF0ICR
PORTH	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0
PORTI	PI7	PI6	PI5	PI4	PI3	PI2	PI1	PI0
PHDR	PH7DR	PH6DR	PH5DR	PH4DR	PH3DR	PH2DR	PH1DR	PH0DR
PIDR	PI7DR	PI6DR	PI5DR	PI4DR	PI3DR	PI2DR	PI1DR	PI0DR
PHDDR	PH7DDR	PH6DDR	PH5DDR	PH4DDR	PH3DDR	PH2DDR	PH1DDR	PH0DDR
PIDDR	PI7DDR	PI6DDR	PI5DDR	PI4DDR	PI3DDR	PI2DDR	PI1DDR	PI0DDR
PHICR	PH7ICR	PH6ICR	PH5ICR	PH4ICR	PH3ICR	PH2ICR	PH1ICR	PH0ICR
PIICR	PI7ICR	PI6ICR	PI5ICR	PI4ICR	PI3ICR	PI2ICR	PI1ICR	PI0ICR
PDPCR	PD7PCR	PD6PCR	PD5PCR	PD4PCR	PD3PCR	PD2PCR	PD1PCR	PD0PCR
PEPCR	PE7PCR	PE6PCR	PE6PCR	PE4PCR	PE3PCR	PE2PCR	PE1PCR	PE0PCR

PFCR6	—	LHWROE	—	—	TCLKS	—	—	—
PFCR7	DMAS3A	DMAS3B	DMAS2A	DMAS2B	DMAS1A	DMAS1B	DMAS0A	DMAS0B
PFCR9	TPUMS5	TPUMS4	TPUMS3A	TPUMS3B	TPUMS2	TPUMS1	TPUMS0A	TPUMS0
PFCRB	—	—	—	—	ITS11	ITS10	ITS9	ITS8
PFCRC	ITS7	ITS6	ITS5	ITS4	ITS3	ITS2	ITS1	ITS0
SSIER	—	—	—	—	SSI11	SSI10	SSI9	SSI8
	SSI7	SSI6	SSI5	SSI4	SSI3	SSI2	SSI1	SSI0
DSAR_0								
DDAR_0								
DOFR_0								
DTCR_0								
DBSR_0	BKSZH31	BKSZH30	BKSZH29	BKSZH28	BKSZH27	BKSZH26	BKSZH25	BKSZH2
	BKSZH23	BKSZH22	BKSZH21	BKSZH20	BKSZH19	BKSZH18	BKSZH17	BKSZH1
	BKSZH15	BKSZH14	BKSZH13	BKSZH12	BKSZH11	BKSZH10	BKSZH9	BKSZH8
	BKSZH7	BKSZH6	BKSZH5	BKSZH4	BKSZH3	BKSZH2	BKSZH1	BKSZH0

DDAR_1								
DOFR_1								
DTCR_1								
DBSR_1	BKSZH31	BKSZH30	BKSZH29	BKSZH28	BKSZH27	BKSZH26	BKSZH25	BKSZH24
	BKSZH23	BKSZH22	BKSZH21	BKSZH20	BKSZH19	BKSZH18	BKSZH17	BKSZH16
	BKSZH15	BKSZH14	BKSZH13	BKSZH12	BKSZH11	BKSZH10	BKSZH9	BKSZH8
	BKSZH7	BKSZH6	BKSZH5	BKSZH4	BKSZH3	BKSZH2	BKSZH1	BKSZH0
DMDR_1	DTE	DACKE	TENDE	—	DREQS	NRD	—	—
	ACT	—	—	—	ERRF	—	ESIF	DTIF
	DTSZ1	DTSZ0	MDS1	MDS0	TSEIE	—	ESIE	DTIE
	DTF1	DTF0	DTA	—	—	DMAP2	DMAP1	DMAP0
DACR_1	AMS	DIRS	—	—	—	RPTIE	ARS1	ARS0
	—	—	SAT1	SAT0	—	—	DAT1	DAT0
	SARIE	—	—	SARA4	SARA3	SARA2	SARA1	SARA0
	DARIE	—	—	DARA4	DARA3	DARA2	DARA1	DARA0

DSAR_2								
DTCR_2								
DBSR_2	BKSZH31	BKSZH30	BKSZH29	BKSZH28	BKSZH27	BKSZH26	BKSZH25	BKSZH2
	BKSZH23	BKSZH22	BKSZH21	BKSZH20	BKSZH19	BKSZH18	BKSZH17	BKSZH1
	BKSZH15	BKSZH14	BKSZH13	BKSZH12	BKSZH11	BKSZH10	BKSZH9	BKSZH8
	BKSZH7	BKSZH6	BKSZH5	BKSZH4	BKSZH3	BKSZH2	BKSZH1	BKSZH0
DMDR_2	DTE	DACKE	TENDE	—	DREQS	NRD	—	—
	ACT	—	—	—	ERRF	—	ESIF	DTIF
	DTSZ1	DTSZ0	MDS1	MDS0	TSEIE	—	ESIE	DTIE
	DTF1	DTF0	DTA	—	—	DMAP2	DMAP1	DMAP0
DACR_2	AMS	DIRS	—	—	—	RPTIE	ARS1	ARS0
	—	—	SAT1	SAT0	—	—	DAT1	DAT0
	SARIE	—	—	SARA4	SARA3	SARA2	SARA1	SARA0
	DARIE	—	—	DARA4	DARA3	DARA2	DARA1	DARA0
DSAR_3								
DDAR_3								

DBCR_3	BKSZH31	BKSZH30	BKSZH29	BKSZH28	BKSZH27	BKSZH26	BKSZH25	BKSZH24
	BKSZH23	BKSZH22	BKSZH21	BKSZH20	BKSZH19	BKSZH18	BKSZH17	BKSZH16
	BKSZH15	BKSZH14	BKSZH13	BKSZH12	BKSZH11	BKSZH10	BKSZH9	BKSZH8
	BKSZH7	BKSZH6	BKSZH5	BKSZH4	BKSZH3	BKSZH2	BKSZH1	BKSZH0
DMDR_3	DTE	DACKE	TENDE	—	DREQS	NRD	—	—
	ACT	—	—	—	ERRF	—	ESIF	DTIF
	DTSZ1	DTSZ0	MDS1	MDS0	TSEIE	—	ESIE	DTIE
	DTF1	DTF0	DTA	—	—	DMAP2	DMAP1	DMAP0
DACR_3	AMS	DIRS	—	—	—	RPTIE	ARS1	ARS0
	—	—	SAT1	SAT0	—	—	DAT1	DAT0
	SARIE	—	—	SARA4	SARA3	SARA2	SARA1	SARA0
	DARIE	—	—	DARA4	DARA3	DARA2	DARA1	DARA0
DMRSR_0								
DMRSR_1								
DMRSR_2								
DMRSR_3								
IPRA	—	IPRA14	IPRA13	IPRA12	—	IPRA10	IPRA9	IPRA8
	—	IPRA6	IPRA5	IPRA4	—	IPRA2	IPRA1	IPRA0
IPRB	—	IPRB14	IPRB13	IPRB12	—	IPRB10	IPRB9	IPRB8
	—	IPRB6	IPRB5	IPRB4	—	IPRB2	IPRB1	IPRB0
IPRC	—	IPRC14	IPRC13	IPRC12	—	IPRC10	IPRC9	IPRC8
	—	IPRC6	IPRC5	IPRC4	—	IPRC2	IPRC1	IPRC0
IPRE	—	—	—	—	—	IPRE10	IPRE9	IPRE8
	—	—	—	—	—	—	—	—
IPRF	—	—	—	—	—	IPRF10	IPRF9	IPRF8
	—	IPRF6	IPRF5	IPRF4	—	IPRF2	IPRF1	IPRF0

IPR2		IPRL7	IPRL6	IPRL5	IPRL4			
	—	IPRL6	IPRL5	IPRL4	—	—	—	—
ISCRH	—	—	—	—	—	—	—	—
	IRQ11SR	IRQ11SF	IRQ10SR	IRQ10SF	IRQ9SR	IRQ9SF	IRQ8SR	IRQ8SF
ISCRL	IRQ7SR	IRQ7SF	IRQ6SR	IRQ6SF	IRQ5SR	IRQ5SF	IRQ4SR	IRQ4SF
	IRQ3SR	IRQ3SF	IRQ2SR	IRQ2SF	IRQ1SR	IRQ1SF	IRQ0SR	IRQ0SF
DTCVBR								
ABWCR	ABWH7	ABWH6	ABWH5	ABWH4	ABWH3	ABWH2	ABWH1	ABWH0
	ABWL7	ABWL6	ABWL5	ABWL4	ABWL3	ABWL2	ABWL1	ABWL0
ASTCR	AST7	AST6	AST5	AST4	AST3	AST2	AST1	AST0
	—	—	—	—	—	—	—	—
WTCRA	—	W72	W71	W70	—	W62	W61	W60
	—	W52	W51	W50	—	W42	W41	W40
WTCRB	—	W32	W31	W30	—	W22	W21	W20
	—	W12	W11	W10	—	W02	W01	W00
RDNCR	RDN7	RDN6	RDN5	RDN4	RDN3	RDN2	RDN1	RDN0
	—	—	—	—	—	—	—	—
CSACR	CSXH7	CSXH6	CSXH5	CSXH4	CSXH3	CSXH2	CSXH1	CSXH0
	CSXT7	CSXT6	CSXT5	CSXT4	CSXT3	CSXT2	CSXT1	CSXT0
IDLCR	IDLS3	IDLS2	IDLS1	IDLS0	IDLCB1	IDLCB0	IDLCA1	IDLCA0
	IDLSEL7	IDLSEL6	IDLSEL5	IDLSEL4	IDLSEL3	IDLSEL2	IDLSEL1	IDLSEL0
BCR1	BRLE	BREQOE	—	—	—	—	WDBE	WAITE
	DKC	—	—	—	—	—	—	—
BCR2	—	—	—	IBCCS	—	—	—	PWDBE

MDCR					MDS3	MDS2	MDS1	MDS0
	—	—	—	—	—	—	—	—
SYSCR	—	—	MACS	—	FETCHMD	—	EXPE	RAME
	—	—	—	—	—	—	DTCMD	—
SCKCR	PSTOP1	—	—	—	—	ICK2	ICK1	ICK0
	—	PCK2	PCK1	PCK0	—	BCK2	BCK1	BCK0
SBYCR	SSBY	OPE	—	STS4	STS3	STS2	STS1	STS0
	SLPIE	—	—	—	—	—	—	—
MSTPCRA	ACSE	MSTPA14	MSTPA13	MSTPA12	MSTPA11	MSTPA10	MSTPA9	MSTPA8
	MSTPA7	MSTPA6	MSTPA5	MSTPA4	MSTPA3	MSTPA2	MSTPA1	MSTPA0
MSTPCRB	MSTPB15	MSTPB14	MSTPB13	MSTPB12	MSTPB11	MSTPB10	MSTPB9	MSTPB8
	MSTPB7	MSTPB6	MSTPB5	MSTPB4	MSTPB3	MSTPB2	MSTPB1	MSTPB0
MSTPCRC	MSTPC15	MSTPC14	MSTPC13	MSTPC12	MSTPC11	MSTPC10	MSTPC9	MSTPC8
	MSTPC7	MSTPC5	MSTPC5	MSTPC4	MSTPC3	MSTPC2	MSTPC1	MSTPC0
SEMR_2	—	—	—	—	ABCS	ACS2	ACS1	ACS0
SMR_4*	C/$\overline{A}$ (GM)	CHR (BLK)	PE (PE)	O/$\overline{E}$ (O/$\overline{E}$)	STOP (BCP1)	MP (BCP0)	CKS1	CKS0
BRR_4								
SCR_4*	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
TDR_4								
SSR_4*	TDRE	RDRF	ORER	FER (ERS)	PER	TEND	MPB	MPBT
RDR_4								
SCMR_4	—	—	—	—	SDIR	SINV	—	SMIF
FCCS	—	—	—	FLER	—	—	—	SCO
FPCS	—	—	—	—	—	—	—	PPVS
FECS	—	—	—	—	—	—	—	EPVB
FKEY	K7	K6	K5	K4	K3	K2	K1	K0

TCORB_2								
TCORB_3								
TCNT_2								
TCNT_3								
TCCR_2	—	—	—	—	TMRIS	—	ICKS1	ICKS0
TCCR_3	—	—	—	—	TMRIS	—	ICKS1	ICKS0
TCR_4	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
TMDR_4	—	—	—	—	MD3	MD2	MD1	MD0
TIOR_4	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
TIER_4	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
TSR_4	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA
TCNT_4								
TGRA_4								
TGRB_4								
TCR_5	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
TMDR_5	—	—	—	—	MD3	MD2	MD1	MD0
TIOR_5	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
TIER_5	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
TSR_5	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA
TCNT_5								
TGRA_5								
TGRB_5								

DTCERE	DTCERE15	DTCERE14	DTCERE13	DTCERE12	DTCERE11	DTCERE10	DTCERE9	DTCERE8
	—	—	—	—	—	—	—	—
DTCCR	—	—	—	RRS	RCHNE	—	—	ERR
INTCR	—	—	INTM1	INTM0	NMIEG	—	—	—
CPUPCR	CPUPCE	DTCP2	DTCP1	DTCP0	IPSETE	CPUP2	CPUP1	CPUP0
IER	—	—	—	—	IRQ11E	IRQ10E	IRQ9E	IRQ8E
	IRQ7E	IRQ6E	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IRQ0E
ISR	—	—	—	—	IRQ11F	IRQ10F	IRQ9F	IRQ8F
	IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F
PORT1	P17	P16	P15	P14	P13	P12	P11	P10
PORT2	P27	P26	P25	P24	P23	P22	P21	P20
PORT3	P37	P36	P35	P34	P33	P32	P31	P30
PORT5	P57	P56	P55	P54	P53	P52	P51	P50
PORT6	—	—	P65	P64	P63	P62	P61	P60
PORTA	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
PORTB	—	—	—	—	PB3	PB2	PB1	PB0
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
P1DR	P17DR	P16DR	P15DR	P14DR	P13DR	P12DR	P11DR	P10DR
P2DR	P27DR	P26DR	P25DR	P24DR	P23DR	P22DR	P21DR	P20DR
P3DR	P37DR	P36DR	P35DR	P34DR	P33DR	P32DR	P31DR	P30DR
P6DR	—	—	P65DR	P64DR	P63DR	P62DR	P61DR	P60DR
PADR	PA7DR	PA6DR	PA5DR	PA4DR	PA3DR	PA2DR	PA1DR	PA0DR
PBDR	—	—	—	—	PB3DR	PB2DR	PB1DR	PA0DR
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

DADR1								
DACR01	DAOE1	DAOE0	DAE	—	—	—	—	—
PCR	G3CMS1	G3CMS0	G2CMS1	G2CMS0	G1CMS1	G1CMS0	G0CMS1	G0CMS0
PMR	G3INV	G2INV	G1INV	G0INV	G3NOV	G2NOV	G1NOV	G0NOV
NDERH	NDER15	NDER14	NDER13	NDER12	NDER11	NDER10	NDER9	NDER8
NDERL	NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1	NDER0
PODRH	POD15	POD14	POD13	POD12	POD11	POD10	POD9	POD8
PODRL	POD7	POD6	POD5	POD4	POD3	POD2	POD1	POD0
NDRH*2	NDR15	NDR14	NDR13	NDR12	NDR11	NDR10	NDR9	NDR8
NDRL*2	NDR7	NDR6	NDR5	NDR4	NDR3	NDR2	NDR1	NDR0
NDRH*2	—	—	—	—	NDR11	NDR10	NDR9	NDR8
NDRL*2	—	—	—	—	NDR3	NDR2	NDR1	NDR0
SMR_0*1	C/$\overline{A}$ (GM)	CHR (BLK)	PE (PE)	O/$\overline{E}$ (O/$\overline{E}$)	STOP (BCP1)	MP (BCP0)	CKS1	CKS0
BRR_0								
SCR_0*1	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
TDR_0								
SSR_0*1	TDRE	RDRF	ORER	FER (ERS)	PER	TEND	MPB	MPBT
RDR_0								
SCMR_0	—	—	—	—	SDIR	SINV	—	SMIF
SMR_1*1	C/$\overline{A}$ (GM)	CHR (BLK)	PE (PE)	O/$\overline{E}$ (O/$\overline{E}$)	STOP (BCP1)	MP (BCP0)	CKS1	CKS0
BRR_1								
SCR_1*1	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
TDR_1								
SSR_1*1	TDRE	RDRF	ORER	FER (ERS)	PER	TEND	MPB	MPBT
RDR_1								
SCMR_1	—	—	—	—	SDIR	SINV	—	SMIF

ADDRE								
ADDRF								
ADDRG								
ADDRH								
ADCSR	ADF	ADIE	ADST	—	CH3	CH2	CH1	CH0
ADCR	TRGS1	TRGS0	SCANE	SCANS	CKS1	CKS0	—	—
TCSR	OVF	WT/$\overline{\text{IT}}$	TME	—	—	CKS2	CKS1	CKS0
TCNT								
RSTCSR	WOVF	RSTE	—	—	—	—	—	—
TCR_0	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0
TCR_1	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0
TCSR_0	CMFB	CMFA	OVF	ADTE	OS3	OS2	OS1	OS0
TCSR_1	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0
TCORA_0								
TCORA_1								
TCORB_0								
TCORB_1								
TCNT_0								
TCNT_1								
TCCR_0	—	—	—	—	TMRIS	—	ICKS1	ICKS0
TCCR_1	—	—	—	—	TMRIS	—	ICKS1	ICKS0
TSTR	—	—	CST5	CST4	CST3	CST2	CST1	CST0
TSYR	—	—	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0

TGRA_0								
TGRB_0								
TGRC_0								
TGRD_0								
TCR_1	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
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								
TGRA_1								
TGRB_1								
TCR_2	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
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								

TSR_3	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
TCNT_3								
TGRA_3								
TGRB_3								
TGRC_3								
TGRD_3								

Notes: 1. Parts of the bit functions differ in normal mode and the smart card interface.

2. When the same output trigger is specified for pulse output groups 2 and 3 by t setting, the NDRH address is H'FFF7C. When different output triggers are spe NDRH addresses for pulse output groups 2 and 3 are H'FFF7E and H'FFF7C, respectively. Similarly, When the same output trigger is specified for pulse out groups 0 and 1 by the PCR setting, the NDRL address is H'FFF7D. When diffe output triggers are specified, the NDRL addresses for pulse output groups 0 a H'FFF7F and H'FFF7D, respectively.

PEDDR	Initialized	—	—	—	—	Initialized
PFDDR	Initialized	—	—	—	—	Initialized
P1ICR	Initialized	—	—	—	—	Initialized
P2ICR	Initialized	—	—	—	—	Initialized
P3ICR	Initialized	—	—	—	—	Initialized
P5ICR	Initialized	—	—	—	—	Initialized
P6ICR	Initialized	—	—	—	—	Initialized
PAICR	Initialized	—	—	—	—	Initialized
PBICR	Initialized	—	—	—	—	Initialized
PDICR	Initialized	—	—	—	—	Initialized
PEICR	Initialized	—	—	—	—	Initialized
PFICR	Initialized	—	—	—	—	Initialized
PORTH	—	—	—	—	—	—
PORTI	—	—	—	—	—	—
PHDR	Initialized	—	—	—	—	Initialized
PIDR	Initialized	—	—	—	—	Initialized
PHDDR	Initialized	—	—	—	—	Initialized
PIDDR	Initialized	—	—	—	—	Initialized
PHICR	Initialized	—	—	—	—	Initialized
PIICR	Initialized	—	—	—	—	Initialized
PDPCR	Initialized	—	—	—	—	Initialized
PEPCR	Initialized	—	—	—	—	Initialized
PFPCR	Initialized	—	—	—	—	Initialized
PHPCR	Initialized	—	—	—	—	Initialized
PIPCR	Initialized	—	—	—	—	Initialized

PFCR9	Initialized	—	—	—	—	Initialized	
PFCRB	Initialized	—	—	—	—	Initialized	
PFCRC	Initialized	—	—	—	—	Initialized	
SSIER	Initialized	—	—	—	—	Initialized	IN
DSAR_0	Initialized	—	—	—	—	Initialized	DM
DDAR_0	Initialized	—	—	—	—	Initialized	
DOFR_0	Initialized	—	—	—	—	Initialized	
DTCR_0	Initialized	—	—	—	—	Initialized	
DBSR_0	Initialized	—	—	—	—	Initialized	
DMDR_0	Initialized	—	—	—	—	Initialized	
DACR_0	Initialized	—	—	—	—	Initialized	
DSAR_1	Initialized	—	—	—	—	Initialized	DM
DDAR_1	Initialized	—	—	—	—	Initialized	
DOFR_1	Initialized	—	—	—	—	Initialized	
DTCR_1	Initialized	—	—	—	—	Initialized	
DBSR_1	Initialized	—	—	—	—	Initialized	
DMDR_1	Initialized	—	—	—	—	Initialized	
DACR_1	Initialized	—	—	—	—	Initialized	
DSAR_2	Initialized	—	—	—	—	Initialized	DM
DDAR_2	Initialized	—	—	—	—	Initialized	
DOFR_2	Initialized	—	—	—	—	Initialized	
DTCR_2	Initialized	—	—	—	—	Initialized	
DBSR_2	Initialized	—	—	—	—	Initialized	
DMDR_2	Initialized	—	—	—	—	Initialized	
DACR_2	Initialized	—	—	—	—	Initialized	

DMRSR_1	Initialized	—	—	—	—	Initialized	D
DMRSR_2	Initialized	—	—	—	—	Initialized	D
DMRSR_3	Initialized	—	—	—	—	Initialized	D
IPRA	Initialized	—	—	—	—	Initialized	I
IPRB	Initialized	—	—	—	—	Initialized	
IPRC	Initialized	—	—	—	—	Initialized	
IPRE	Initialized	—	—	—	—	Initialized	
IPRF	Initialized	—	—	—	—	Initialized	
IPRG	Initialized	—	—	—	—	Initialized	
IPRH	Initialized	—	—	—	—	Initialized	
IPRI	Initialized	—	—	—	—	Initialized	
IPRK	Initialized	—	—	—	—	Initialized	
IPRL	Initialized	—	—	—	—	Initialized	
ISCRH	Initialized	—	—	—	—	Initialized	
ISCRL	Initialized	—	—	—	—	Initialized	
DTCVBR	Initialized	—	—	—	—	Initialized	B
ABWCR	Initialized	—	—	—	—	Initialized	
ASTCR	Initialized	—	—	—	—	Initialized	
WTCRA	Initialized	—	—	—	—	Initialized	
WTCRB	Initialized	—	—	—	—	Initialized	
RDNCR	Initialized	—	—	—	—	Initialized	
CSACR	Initialized	—	—	—	—	Initialized	
IDLCR	Initialized	—	—	—	—	Initialized	
BCR1	Initialized	—	—	—	—	Initialized	
BCR2	Initialized	—	—	—	—	Initialized	
ENDIANCR	Initialized	—	—	—	—	Initialized	

MSTPCRA	Initialized	—	—	—	—	Initialized	
MSTPCRB	Initialized	—	—	—	—	Initialized	
MSTPCRC	Initialized	—	—	—	—	Initialized	
SEMR_2	Initialized	—	—	—	—	Initialized	SC
SMR_4	Initialized	—	—	—	—	Initialized	SC
BRR_4	Initialized	—	—	—	—	Initialized	
SCR_4	Initialized	—	—	—	—	Initialized	
TDR_4	Initialized	Initialized	—	Initialized	Initialized	Initialized	
SSR_4	Initialized	Initialized	—	Initialized	Initialized	Initialized	
RDR_4	Initialized	Initialized	—	Initialized	Initialized	Initialized	
SCMR_4	Initialized	—	—	—	—	Initialized	
FCCS	Initialized	—	—	—	—	Initialized	FL
FPCS	Initialized	—	—	—	—	Initialized	
FECS	Initialized	—	—	—	—	Initialized	
FKEY	Initialized	—	—	—	—	Initialized	
FMATS	Initialized	—	—	—	—	Initialized	
FTDAR	Initialized	—	—	—	—	Initialized	
TCR_2	Initialized	—	—	—	—	Initialized	TM
TCR_3	Initialized	—	—	—	—	Initialized	TM
TCSR_2	Initialized	—	—	—	—	Initialized	TM
TCSR_3	Initialized	—	—	—	—	Initialized	TM
TCORA_2	Initialized	—	—	—	—	Initialized	TM
TCORA_3	Initialized	—	—	—	—	Initialized	TM
TCORB_2	Initialized	—	—	—	—	Initialized	TM
TCORB_3	Initialized	—	—	—	—	Initialized	TM

TSR_4	Initialized	—	—	—	—	Initialized	
TCNT_4	Initialized	—	—	—	—	Initialized	
TGRA_4	Initialized	—	—	—	—	Initialized	
TGRB_4	Initialized	—	—	—	—	Initialized	
TCR_5	Initialized	—	—	—	—	Initialized	T
TMDR_5	Initialized	—	—	—	—	Initialized	
TIOR_5	Initialized	—	—	—	—	Initialized	
TIER_5	Initialized	—	—	—	—	Initialized	
TSR_5	Initialized	—	—	—	—	Initialized	
TCNT_5	Initialized	—	—	—	—	Initialized	
TGRA_5	Initialized	—	—	—	—	Initialized	
TGRB_5	Initialized	—	—	—	—	Initialized	
DTCERA	Initialized	—	—	—	—	Initialized	I
DTCERB	Initialized	—	—	—	—	Initialized	
DTCERC	Initialized	—	—	—	—	Initialized	
DTCERD	Initialized	—	—	—	—	Initialized	
DTCERE	Initialized	—	—	—	—	Initialized	
DTCCR	Initialized	—	—	—	—	Initialized	
INTCR	Initialized	—	—	—	—	Initialized	
CPUPCR	Initialized	—	—	—	—	Initialized	
IER	Initialized	—	—	—	—	Initialized	
ISR	Initialized	—	—	—	—	Initialized	

PORTE	—	—	—	—	—	—	
PORTF	—	—	—	—	—	—	
P1DR	Initialized	—	—	—	—	Initialized	
P2DR	Initialized	—	—	—	—	Initialized	
P3DR	Initialized	—	—	—	—	Initialized	
P6DR	Initialized	—	—	—	—	Initialized	
PADR	Initialized	—	—	—	—	Initialized	
PBDR	Initialized	—	—	—	—	Initialized	
PDDR	Initialized	—	—	—	—	Initialized	
PEDR	Initialized	—	—	—	—	Initialized	
PFDR	Initialized	—	—	—	—	Initialized	
SMR_2	Initialized	—	—	—	—	Initialized	SC
BRR_2	Initialized	—	—	—	—	Initialized	
SCR_2	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	
DADR0	Initialized	—	—	—	—	Initialized	DA
DADR1	Initialized	—	—	—	—	Initialized	
DACR01	Initialized	—	—	—	—	Initialized	

SMR_0	Initialized	—	—	—	—	Initialized	S
BRR_0	Initialized	—	—	—	—	Initialized	
SCR_0	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	
SMR_1	Initialized	—	—	—	—	Initialized	S
BRR_1	Initialized	—	—	—	—	Initialized	
SCR_1	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	
ADDRA	Initialized	—	—	—	—	Initialized	A
ADDRB	Initialized	—	—	—	—	Initialized	
ADDRC	Initialized	—	—	—	—	Initialized	
ADDRD	Initialized	—	—	—	—	Initialized	
ADDRE	Initialized	—	—	—	—	Initialized	
ADDRF	Initialized	—	—	—	—	Initialized	
ADDRG	Initialized	—	—	—	—	Initialized	
ADDRH	Initialized	—	—	—	—	Initialized	
ADCSR	Initialized	—	—	—	—	Initialized	
ADCR	Initialized	—	—	—	—	Initialized	

TCORA_1	Initialized	—	—	—	—	Initialized	TM
TCORB_0	Initialized	—	—	—	—	Initialized	TM
TCORB_1	Initialized	—	—	—	—	Initialized	TM
TCNT_0	Initialized	—	—	—	—	Initialized	TM
TCNT_1	Initialized	—	—	—	—	Initialized	TM
TCCR_0	Initialized	—	—	—	—	Initialized	TM
TCCR_1	Initialized	—	—	—	—	Initialized	TM
TSTR	Initialized	—	—	—	—	Initialized	TP
TSYR	Initialized	—	—	—	—	Initialized	
TCR_0	Initialized	—	—	—	—	Initialized	TP
TMDR_0	Initialized	—	—	—	—	Initialized	
TIORH_0	Initialized	—	—	—	—	Initialized	
TIORL_0	Initialized	—	—	—	—	Initialized	
TIER_0	Initialized	—	—	—	—	Initialized	
TSR_0	Initialized	—	—	—	—	Initialized	
TCNT_0	Initialized	—	—	—	—	Initialized	
TGRA_0	Initialized	—	—	—	—	Initialized	
TGRB_0	Initialized	—	—	—	—	Initialized	
TGRC_0	Initialized	—	—	—	—	Initialized	
TGRD_0	Initialized	—	—	—	—	Initialized	

TCR_2	Initialized	—	—	—	—	Initialized	T
TMDR_2	Initialized	—	—	—	—	Initialized	
TIOR_2	Initialized	—	—	—	—	Initialized	
TIER_2	Initialized	—	—	—	—	Initialized	
TSR_2	Initialized	—	—	—	—	Initialized	
TCNT_2	Initialized	—	—	—	—	Initialized	
TGRA_2	Initialized	—	—	—	—	Initialized	
TGRB_2	Initialized	—	—	—	—	Initialized	
TCR_3	Initialized	—	—	—	—	Initialized	T
TMDR_3	Initialized	—	—	—	—	Initialized	
TIORH_3	Initialized	—	—	—	—	Initialized	
TIORL_3	Initialized	—	—	—	—	Initialized	
TIER_3	Initialized	—	—	—	—	Initialized	
TSR_3	Initialized	—	—	—	—	Initialized	
TCNT_3	Initialized	—	—	—	—	Initialized	
TGRA_3	Initialized	—	—	—	—	Initialized	
TGRB_3	Initialized	—	—	—	—	Initialized	
TGRC_3	Initialized	—	—	—	—	Initialized	
TGRD_3	Initialized	—	—	—	—	Initialized	

Input voltage (port 5)	V_{in}	–0.3 to AV_{CC} +0.3
Reference power supply voltage	V_{ref}	–0.3 to AV_{CC} +0.3
Analog power supply voltage	AV_{CC}	–0.3 to +4.6
Analog input voltage	V_{AN}	–0.3 to AV_{CC} +0.3
Operating temperature	T_{opr}	Regular specifications: –20 to +75*
		Wide-range specifications: –40 to +85*
Storage temperature	T_{stg}	–55 to +125

Caution: Permanent damage to the LSI may result if absolute maximum ratings are exce

Note: * The operating temperature range during programming/erasing of the flash me
0°C to +75°C for regular specifications and 0°C to +85°C for wide-range spec

Schmitt trigger input voltage	IRQ input pin, TPU input pin, TMR input pin, port 2, port 3	VT^-	$V_{CC} \times 0.2$	—	—	V	
		VT^+	—	—	$V_{CC} \times 0.7$	V	
		$VT^+ - VT^-$	$V_{CC} \times 0.06$	—	—	V	
	Port 5*2	VT^-	$AV_{CC} \times 0.2$	—	—	V	
		VT^+	—	—	$AV_{CC} \times 0.7$	V	
		$VT^+ - VT^-$	$AV_{CC} \times 0.06$	—	—	V	
Input high voltage (except Schmitt trigger input pin)	MD, $\overline{RES}$, $\overline{STBY}$, EMLE, NMI	V_{IH}	$V_{CC} \times 0.9$	—	$V_{CC} + 0.3$	V	
	EXTAL		$V_{CC} \times 0.7$	—	$V_{CC} + 0.3$	V	
	Other input pins		$V_{CC} \times 0.7$	—	$V_{CC} + 0.3$	V	
	Port 5		$AV_{CC} \times 0.7$	—	$AV_{CC} + 0.3$	V	
Input low voltage (except Schmitt trigger input pin)	MD, $\overline{RES}$, $\overline{STBY}$, EMLE	V_{IL}	–0.3	—	$V_{CC} \times 0.1$	V	
	EXTAL, NMI		–0.3	—	$V_{CC} \times 0.2$	V	
	Other pins		–0.3	—	$V_{CC} \times 0.2$	V	
Output high voltage	All output pins	V_{OH}	$V_{CC} - 0.5$	—	—	V	I_{OH} =
			$V_{CC} - 1.0$	—	—		I_{OH} =
Output low voltage	All output pins	V_{OL}	—	—	0.4	V	I_{OL} =
	Port 3		—	—	1.0		I_{OL} =
Input leakage current	$\overline{RES}$	$\lvert I_{in} \rvert$	—	—	10.0	μA	V_{in} = V_{CC} –
	MD, $\overline{STBY}$, EMLE, NMI		—	—	1.0		
	Port 5		—	—	1.0		V_{in} = AV_{CC}

(off state)							
Input pull-up MOS current	Ports D to F, H, I	$-I_p$	10	—	300	μA	V_{CC} 3.6 V_{in} =
Input capacitance	All input pins	C_{in}	—	—	15	pF	V_{in} = f = T_a =
Current consumption *3	Normal operation	I_{CC}*5	—	35 (3.3 V)	45	mA	f = 3
	Sleep mode		—	30 (3.3 V)	37		
	Standby mode*4		—	0.15	0.5		T_a ≤
			—	—	3.0		50°C
	All-module-clock-stop mode*6		—	15	25		
Analog power supply current	During A/D and D/A conversion	AI_{CC}	—	1.0 (3.0 V)	2.0	mA	
	Standby for A/D and D/A conversion		—	1.0	20	μA	
Reference power supply current	During A/D and D/A conversion	AI_{CC}	—	1.5 (3.0 V)	3.0	mA	
	Standby for A/D and D/A conversion		—	0.4	5.0	μA	
RAM standby voltage		V_{RAM}	2.5	—	—	V	
Vcc start voltage*7		$V_{CCSTART}$	—	—	0.8	V	
Vcc rising gradient*7		SV_{CC}	—	—	20	ms/V	

Notes: 1. When the A/D and D/A converters are not used, the AV_{CC}, V_{ref}, and AV_{SS} pins s
be open. Connect the AV_{CC} and V_{ref} pins to V_{CC}, and the AV_{SS} pin to V_{SS}.

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC},
V_{SS} = AV_{SS} = 0 V*,
T_a = –20°C to +75°C (regular specifications),
T_a = –40°C to +85°C (wide-range specifications)

Item		Symbol	Min.	Typ.	Max.
Permissible output low current (per pin)	Output pins except port 3	I_{OL}	—	—	2.0
Permissible output low current (per pin)	Port 3	I_{OL}	—	—	10
Permissible output low current (total)	Total of all output pins	ΣI_{OL}	—	—	80
Permissible output high current (per pin)	All output pins	$-I_{OH}$	—	—	2.0
Permissible output high current (total)	Total of all output pins	$\Sigma -I_{OH}$	—	—	40

Caution: To protect the LSI's reliability, do not exceed the output current values in table

Note: * When the A/D and D/A converters are not used, the AV_{CC}, V_{ref}, and AV_{SS} pins sl
be open. Connect the AV_{CC} and V_{ref} pins to V_{CC}, and the AV_{SS} pin to V_{SS}.

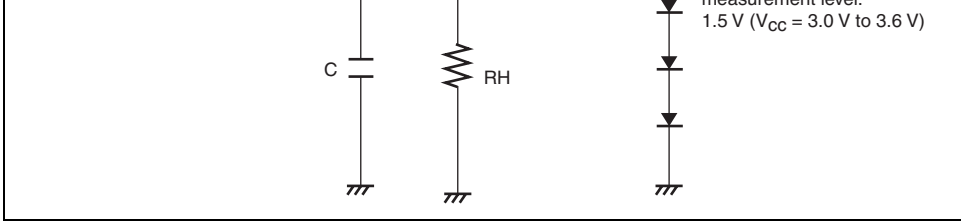


Figure 22.1 Output Load Circuit

Clock cycle time	t_{cyc}	28.0	125	ns	Figure 2
Clock high pulse width	t_{CH}	5	—	ns	
Clock low pulse width	t_{CL}	5	—	ns	
Clock rising time	t_{Cr}	—	5	ns	
Clock falling time	t_{Cf}	—	5	ns	
Oscillation settling time after reset (crystal)	t_{OSC1}	10	—	ms	Figure 2
Oscillation settling time after leaving software standby mode (crystal)	t_{OSC2}	10	—	ms	Figure 2
External clock output delay settling time	t_{DEXT}	1	—	ms	Figure 2
External clock input low pulse width	T_{EXL}	27.7	—	ns	Figure 2
External clock input high pulse width	T_{EXH}	27.7	—	ns	
External clock rising time	T_{EXr}	—	5	ns	
External clock falling time	T_{EXf}	—	5	ns	

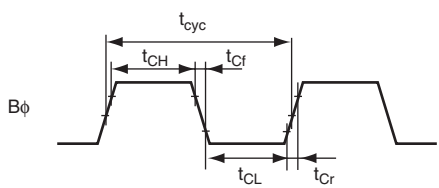


Figure 22.2 External Bus Clock Timing

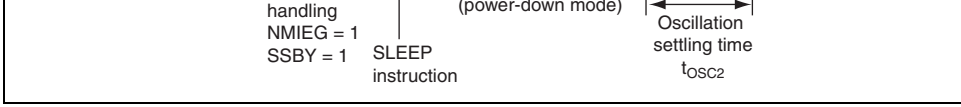


Figure 22.3 Oscillation Settling Timing after Software Standby Mode

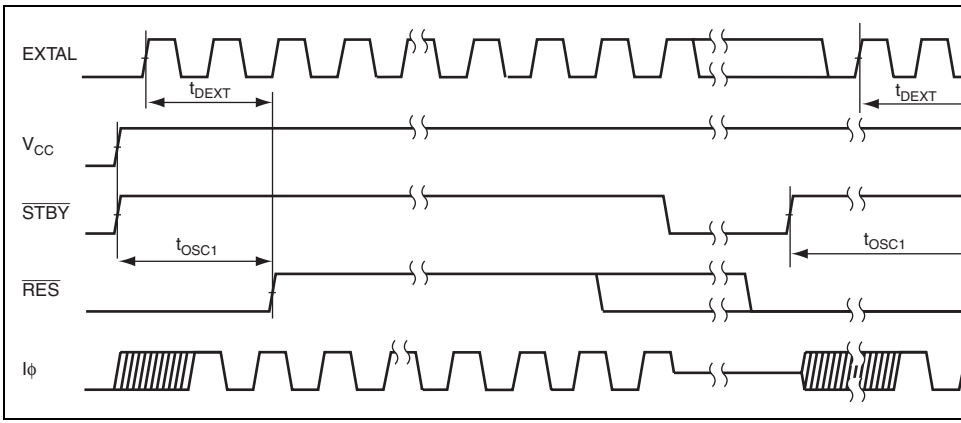


Figure 22.4 Oscillation Settling Timing

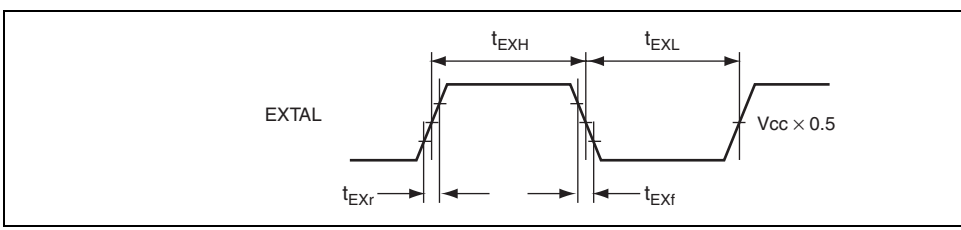


Figure 22.5 External Input Clock Timing

$\overline{RES}$ pulse width	t_{RESW}	20	—	t_{cyc}	
NMI setup time	t_{NMIS}	150	—	ns	Figure 22.
NMI hold time	t_{NMIH}	10	—	ns	
NMI pulse width (after leaving software standby mode)	t_{NMIW}	200	—	ns	
$\overline{IRQ}$ setup time	t_{IRQS}	150	—	ns	
$\overline{IRQ}$ hold time	t_{IRQH}	10	—	ns	
$\overline{IRQ}$ pulse width (after leaving software standby mode)	t_{IRQW}	200	—	ns	

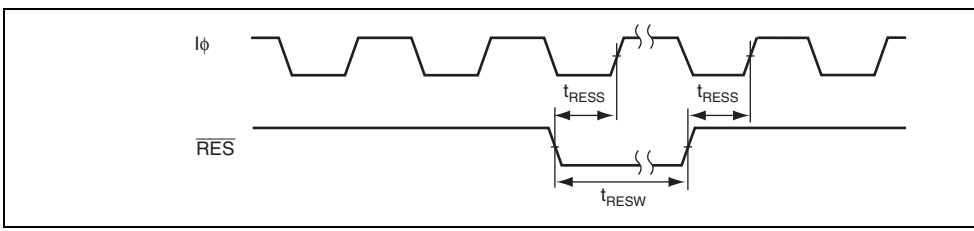


Figure 22.6 Reset Input Timing

IRQ
(level input)

Note: * SSIER must be set to cancel software standby mode.

Figure 22.7 Interrupt Input Timing

22.3.3 Bus Timing

Table 22.6 Bus Timing (1)

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC},
V_{SS} = AV_{SS} = 0 V, Bϕ = 8 MHz to 35 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 Cond
Address delay time	t_{AD}	—	15	ns	Figur 22.20
Address setup time 1	t_{AS1}	$0.5 \times t_{cyc} - 8$	—	ns	
Address setup time 2	t_{AS2}	$1.0 \times t_{cyc} - 8$	—	ns	
Address setup time 3	t_{AS3}	$1.5 \times t_{cyc} - 8$	—	ns	
Address setup time 4	t_{AS4}	$2.0 \times t_{cyc} - 8$	—	ns	
Address hold time 1	t_{AH1}	$0.5 \times t_{cyc} - 8$	—	ns	
Address hold time 2	t_{AH2}	$1.0 \times t_{cyc} - 8$	—	ns	
Address hold time 3	t_{AH3}	$1.5 \times t_{cyc} - 8$	—	ns	
$\overline{CS}$ delay time 1	t_{CSD1}	—	15	ns	
$\overline{AS}$ delay time	t_{ASD}	—	15	ns	

Read data access time 4	t_{AC4}	—	$2.5 \times t_{cyc} - 20$	ns
Read data access time 5	t_{AC5}	—	$1.0 \times t_{cyc} - 20$	ns
Read data access time 6	t_{AC6}	—	$2.0 \times t_{cyc} - 20$	ns
Read data access time (from address) 1	t_{AA1}	—	$1.0 \times t_{cyc} - 20$	ns
Read data access time (from address) 2	t_{AA2}	—	$1.5 \times t_{cyc} - 20$	ns
Read data access time (from address) 3	t_{AA3}	—	$2.0 \times t_{cyc} - 20$	ns
Read data access time (from address) 4	t_{AA4}		$2.5 \times t_{cyc} - 20$	ns
Read data access time (from address) 5	t_{AA5}	—	$3.0 \times t_{cyc} - 20$	ns

$\overline{WR}$ pulse width 2	t_{WSW2}	$1.5 \times t_{cyc} - 13$	—	ns	
Write data delay time	t_{WDD}	—	20	ns	
Write data setup time 1	t_{WDS1}	$0.5 \times t_{cyc} - 13$	—	ns	
Write data setup time 2	t_{WDS2}	$1.0 \times t_{cyc} - 13$	—	ns	
Write data setup time 3	t_{WDS3}	$1.5 \times t_{cyc} - 13$	—	ns	
Write data hold time 1	t_{WDH1}	$0.5 \times t_{cyc} - 8$	—	ns	
Write data hold time 3	t_{WDH3}	$1.5 \times t_{cyc} - 8$	—	ns	
Byte control delay time	t_{UBD}	—	15	ns	Figure 22.14
Byte control pulse width 1	t_{UBW1}	—	$1.0 \times t_{cyc} - 15$	ns	Figure
Byte control pulse width 2	t_{UBW2}	—	$2.0 \times t_{cyc} - 15$	ns	Figure
Multiplexed address delay time 1	t_{MAD1}	—	15	ns	Figure 22.18
Multiplexed address hold time	t_{MAH}	$1.0 \times t_{cyc} - 15$	—	ns	
Multiplexed address setup time 1	t_{MAS1}	$0.5 \times t_{cyc} - 15$	—	ns	
Multiplexed address setup time 2	t_{MAS2}	$1.5 \times t_{cyc} - 15$	—	ns	
Address hold delay time	t_{AHD}	—	15	ns	
Address hold pulse width 1	t_{AHW1}	$1.0 \times t_{cyc} - 15$	—	ns	
Address hold pulse width 2	t_{AHW2}	$2.0 \times t_{cyc} - 15$	—	ns	
$\overline{WAIT}$ setup time	t_{WTS}	15	—	ns	Figure 22.18
$\overline{WAIT}$ hold time	t_{WTH}	5.0	—	ns	
$\overline{BREQ}$ setup time	t_{BREQS}	20	—	ns	Figure
$\overline{BACK}$ delay time	t_{BACD}	—	15	ns	
Bus floating time	t_{BZD}	—	30	ns	
$\overline{BREQO}$ delay time	t_{BRQOD}	—	15	ns	Figure
$\overline{BS}$ delay time	T_{BSD}	1.0	15	ns	Figure 22.9, 22.14
RD/$\overline{WR}$ delay time	T_{RWD}	—	15	ns	

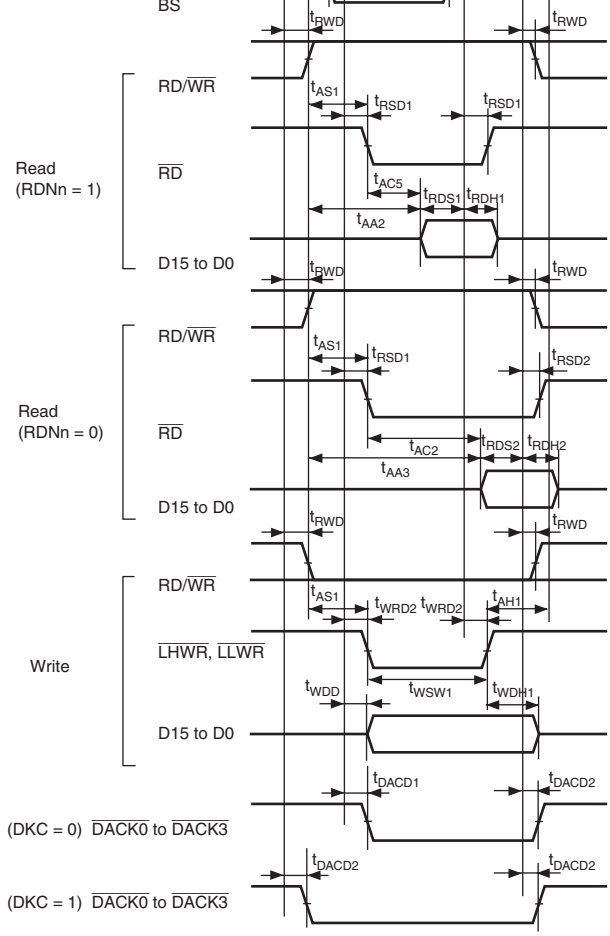


Figure 22.8 Basic Bus Timing: 2-State Access

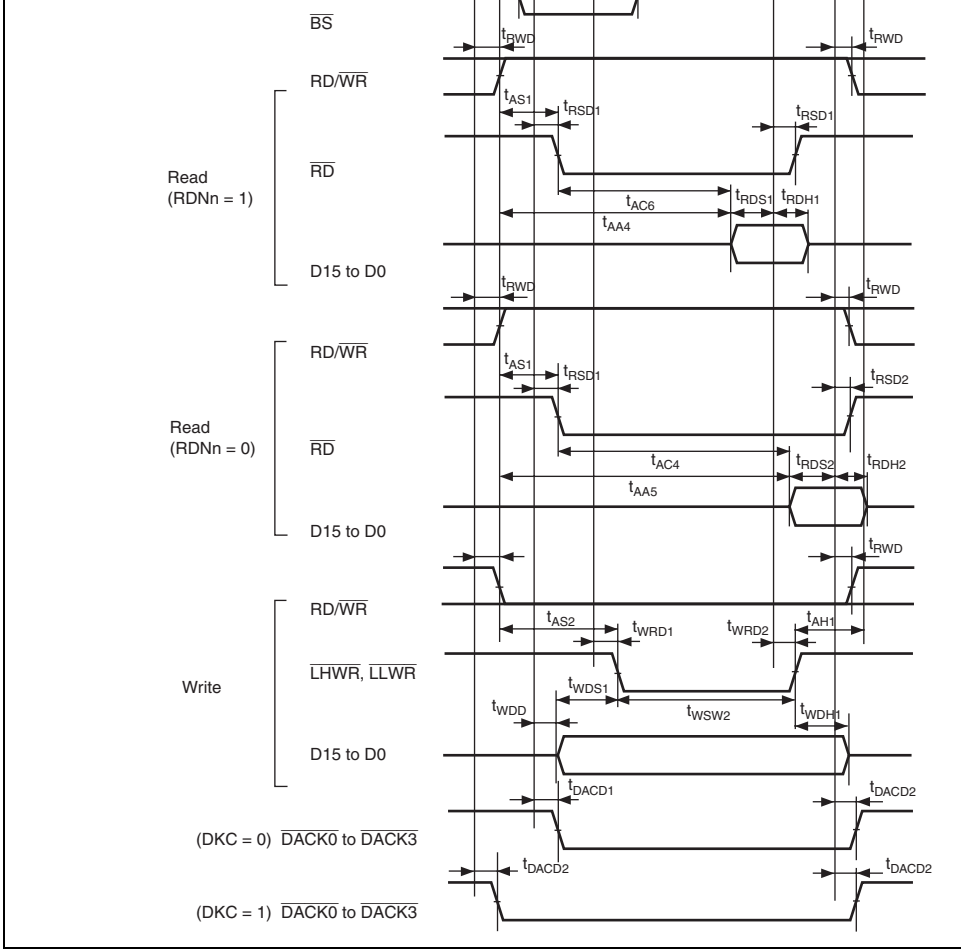


Figure 22.9 Basic Bus Timing: 3-State Access

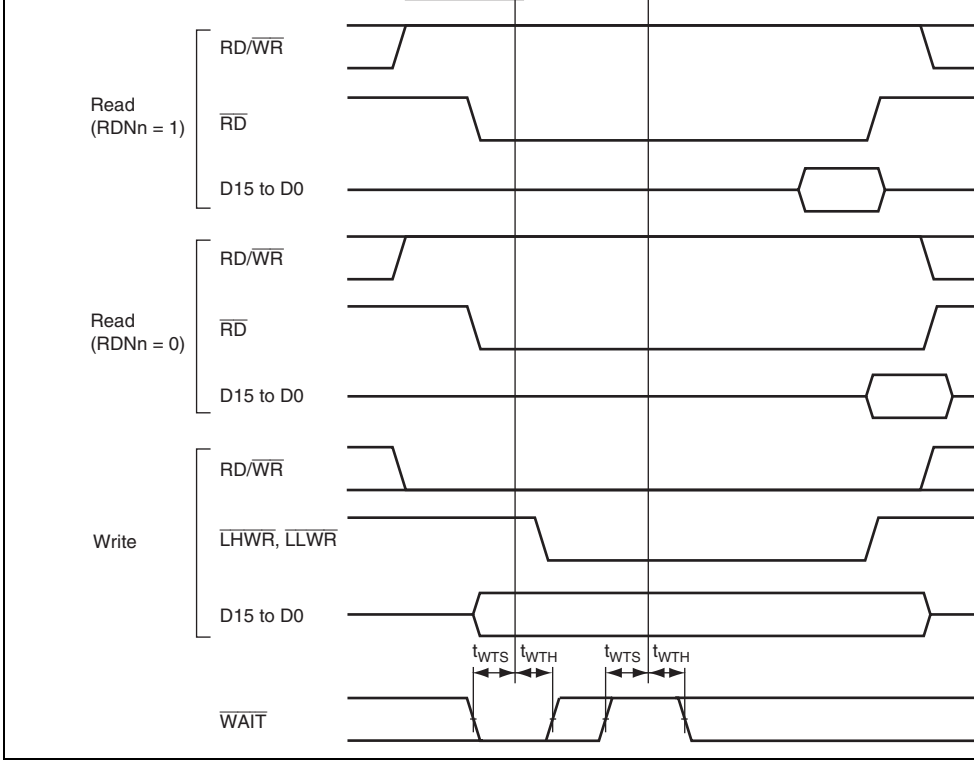


Figure 22.10 Basic Bus Timing: Three-State Access, One Wait

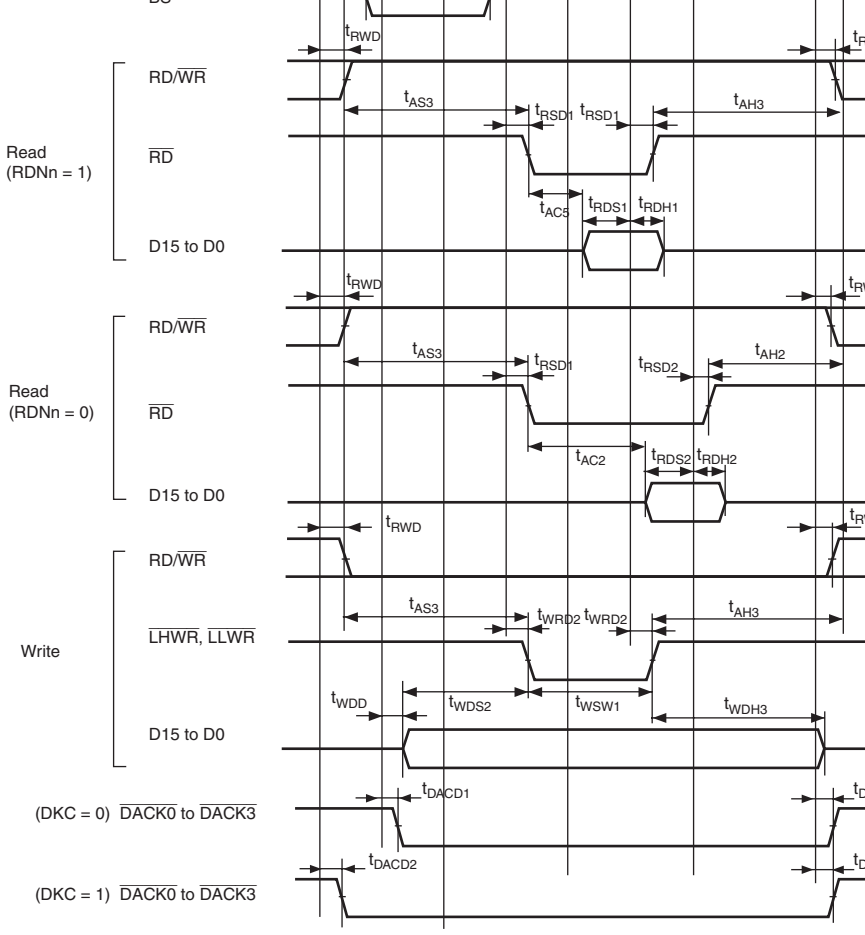


Figure 22.11 Basic Bus Timing: 2-State Access ($\overline{CS}$ Assertion Period Extend

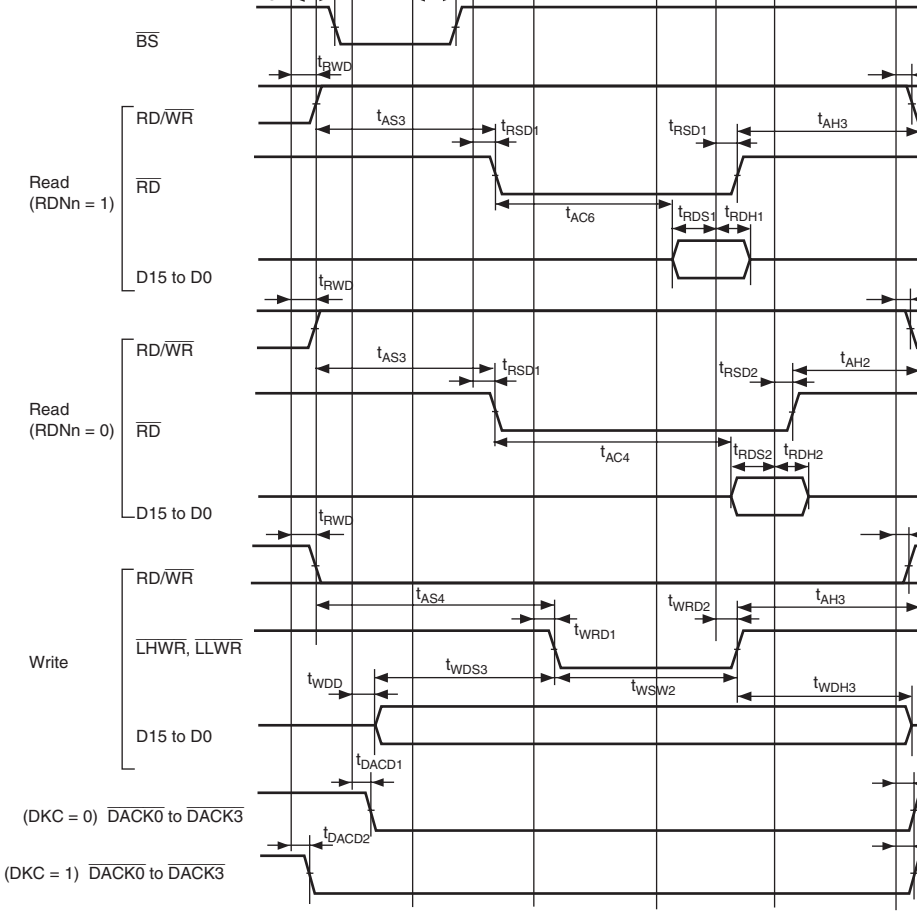


Figure 22.12 Basic Bus Timing: 3-State Access ($\overline{CS}$ Assertion Period Extende

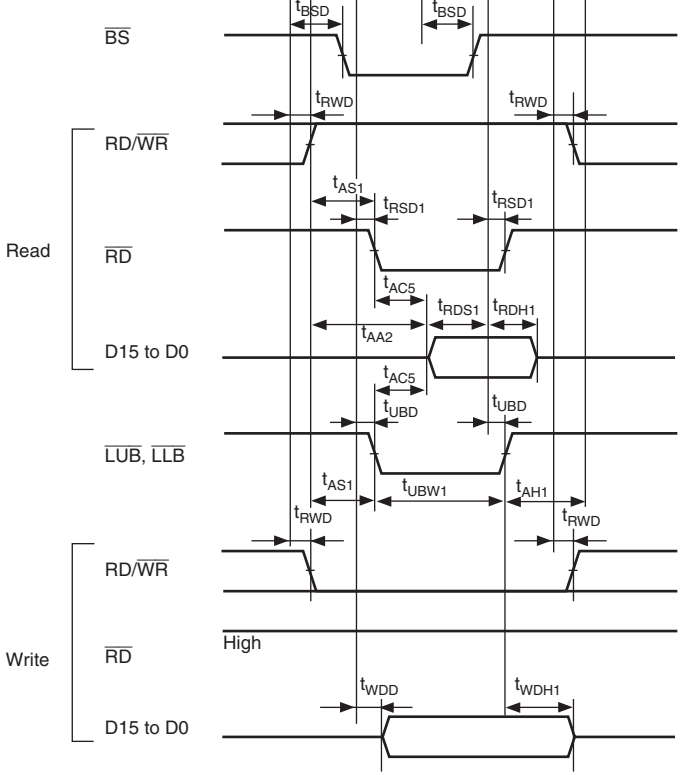


Figure 22.13 Byte Control SRAM: 2-State Read/Write Access

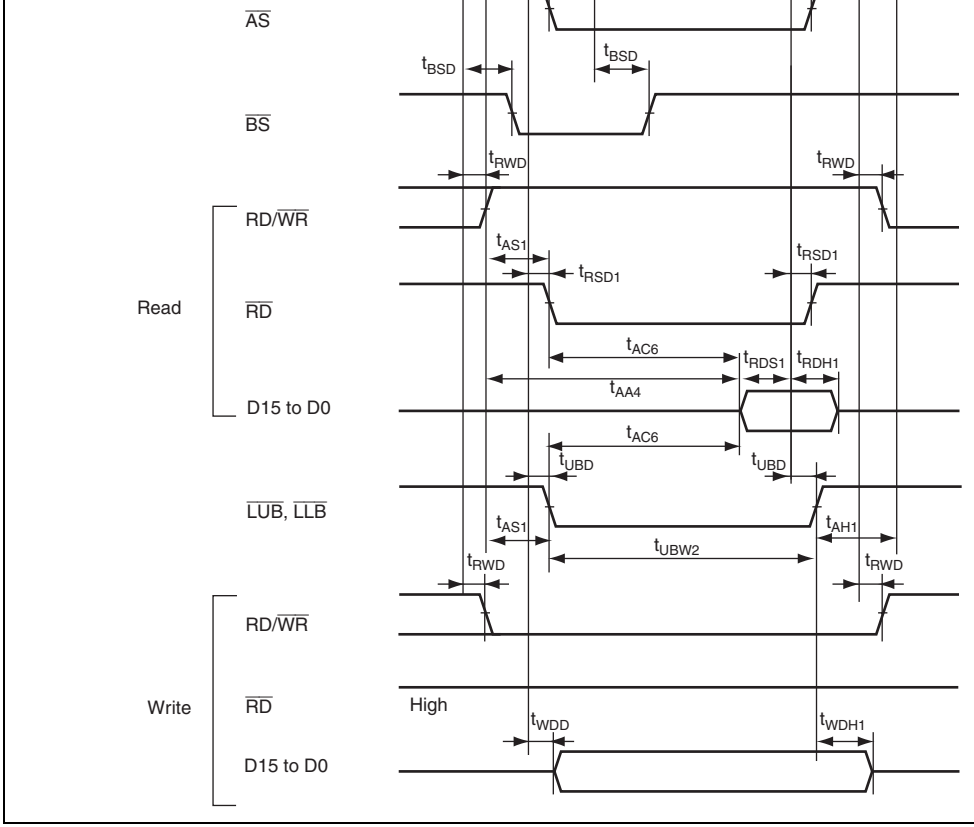


Figure 22.14 Byte Control SRAM: 3-State Read/Write Access

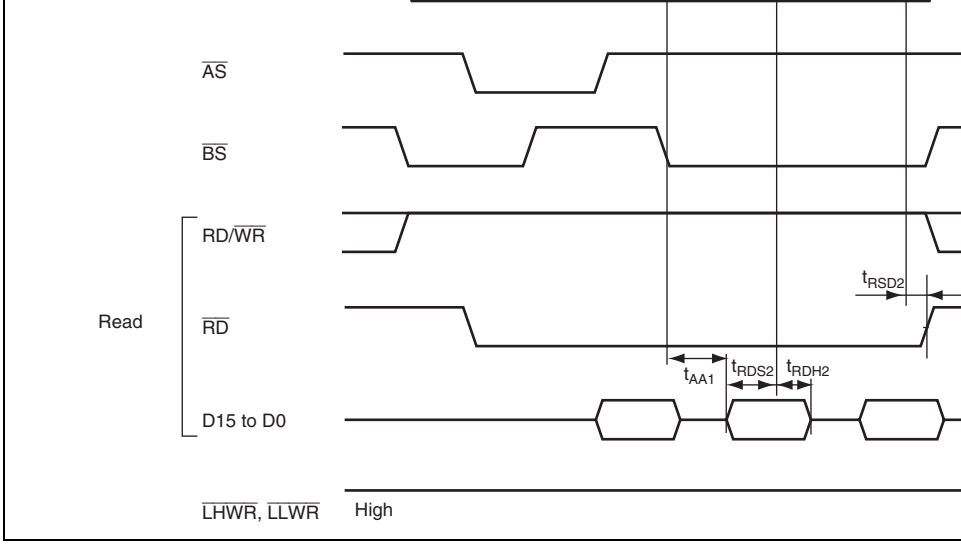


Figure 22.15 Burst ROM Access Timing: 1-State Burst Access

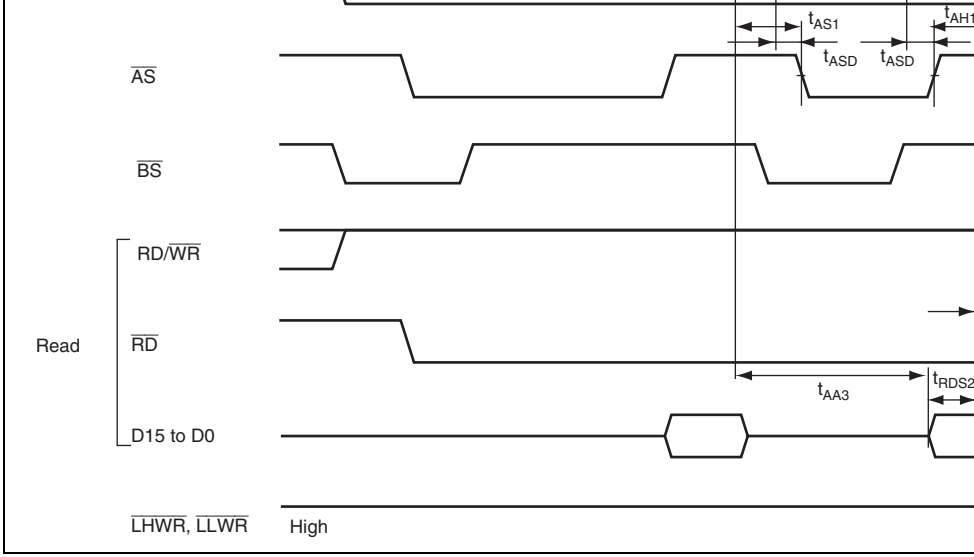


Figure 22.16 Burst ROM Access Timing: 2-State Burst Access

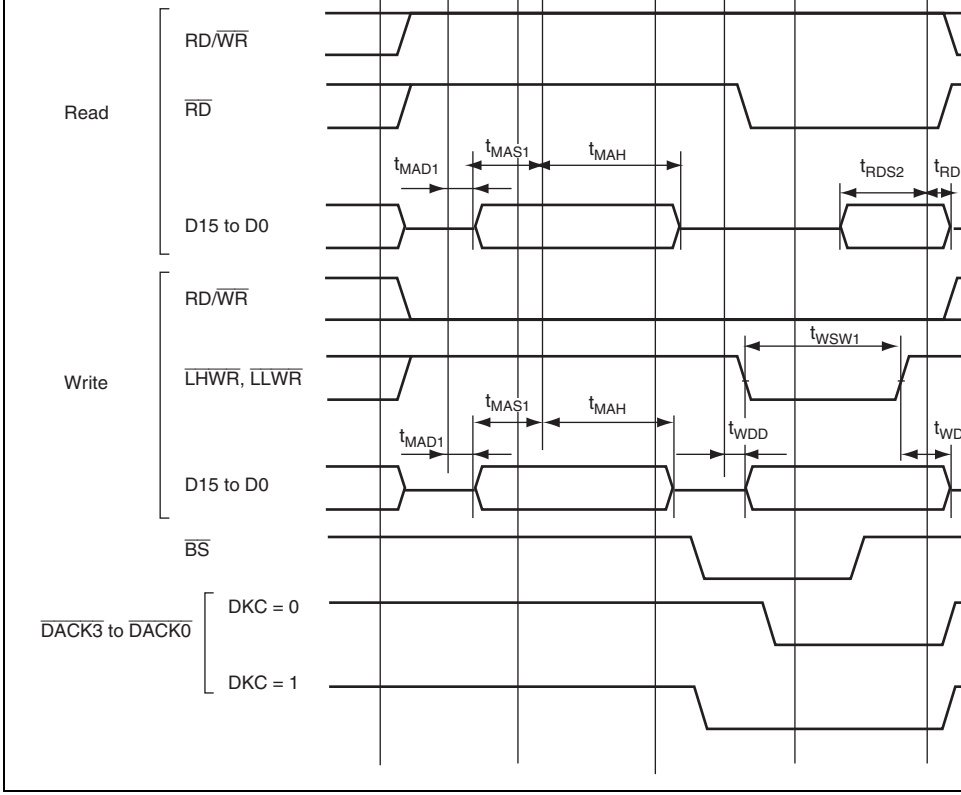


Figure 22.17 Address/Data Multiplexed Access Timing (No Wait) (Basic, 4-State

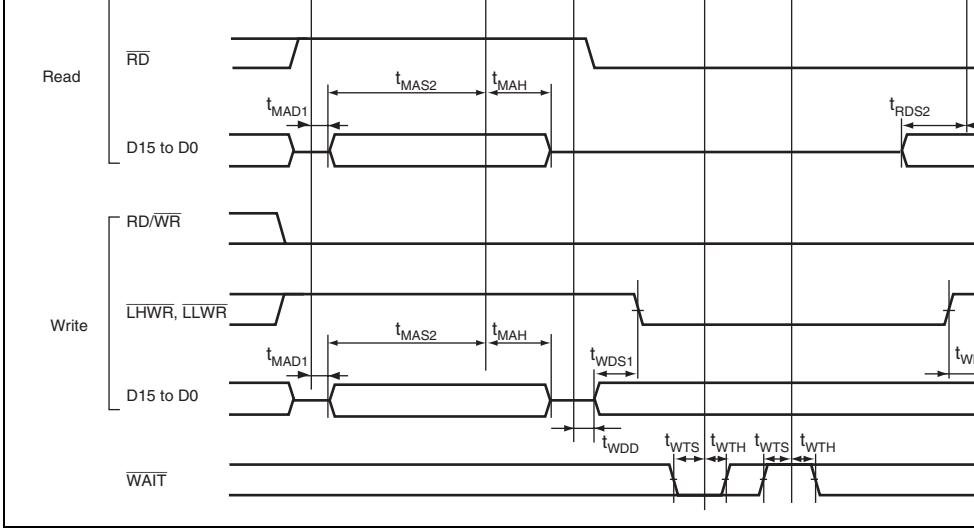


Figure 22.18 Address/Data Multiplexed Access Timing (Wait Control) (Address Cycle Program Wait × 1 + Data Cycle Program Wait × 1 + Data Cycle Pin Wait × 1)

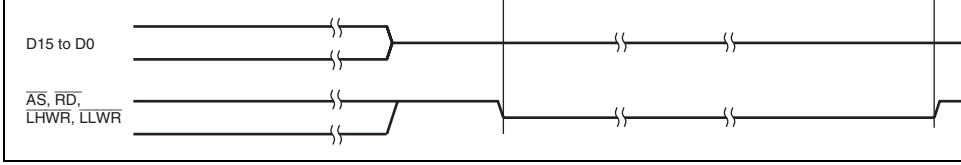


Figure 22.19 External Bus Release Timing

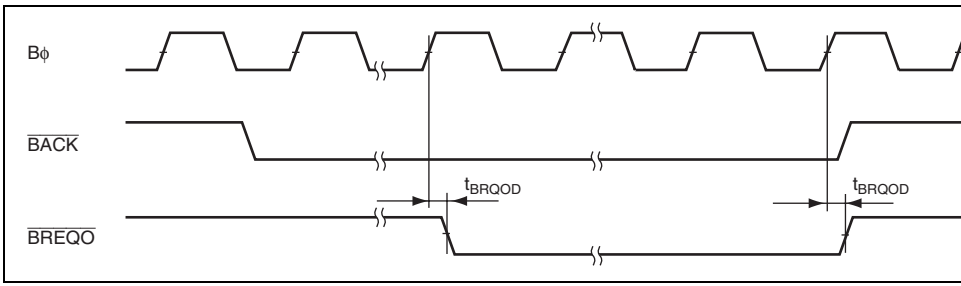


Figure 22.20 External Bus Request Output Timing

$\overline{\text{DREQ}}$ hold time	t_{DRQH}	5	—	ns	
$\overline{\text{TEND}}$ delay time	t_{TED}	—	15	ns	Figure 22.
$\overline{\text{DACK}}$ delay time 1	t_{DACD1}	—	15	ns	Figures 22
$\overline{\text{DACK}}$ delay time 2	t_{DACD2}	—	15	ns	22.24

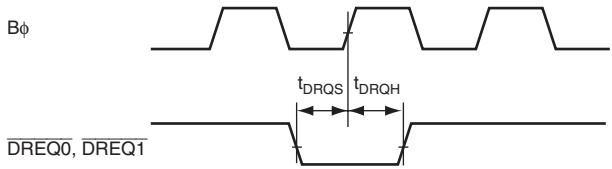


Figure 22.21 DMAC ($\overline{\text{DREQ}}$) Input Timing

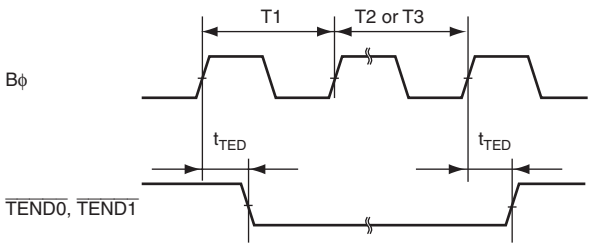


Figure 22.22 DMAC ($\overline{\text{TEND}}$) Output Timing

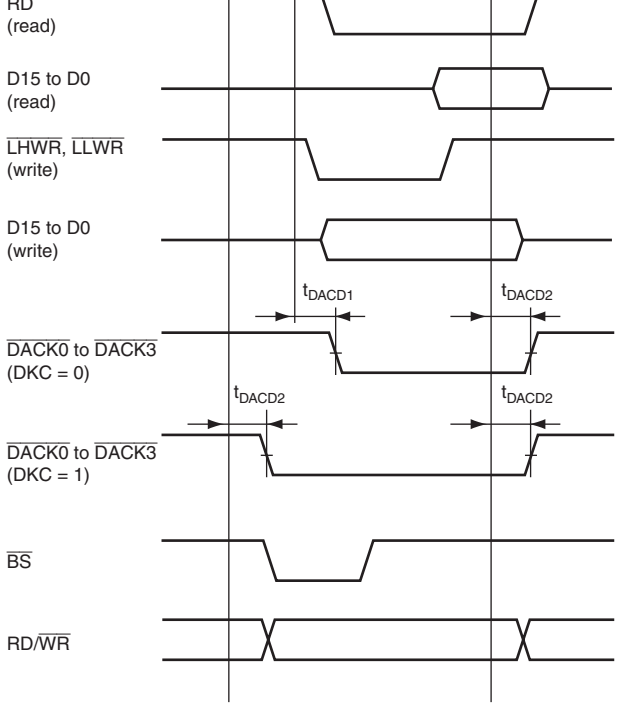


Figure 22.23 DMAC Single-Address Transfer Timing: 2-State Access

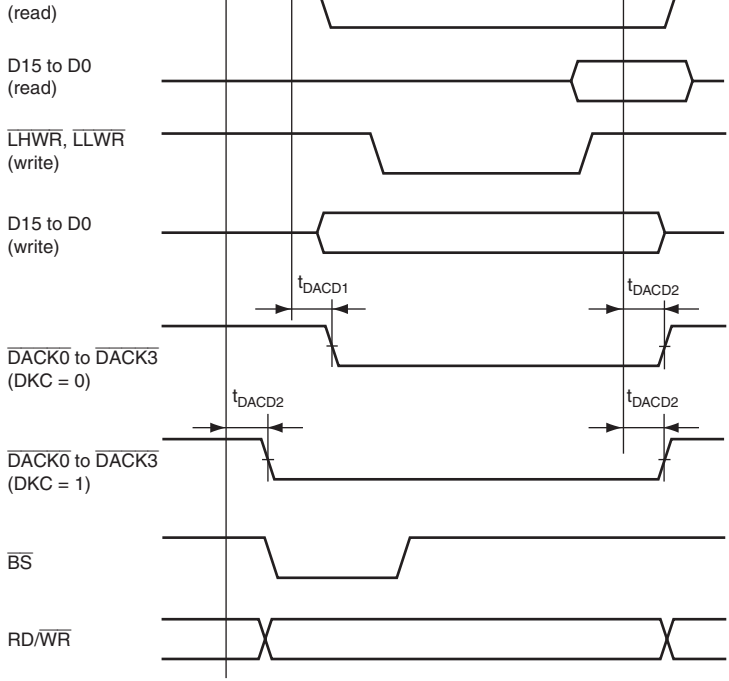


Figure 22.24 DMAC Single-Address Transfer Timing: 3-State Access

	Input data setup time		t_{PRS}	25	—	ns	
	Input data hold time		t_{PRH}	25	—	ns	
TPU	Timer output delay time		t_{TOCD}	—	40	ns	Figure
	Timer input setup time		t_{TICS}	25	—	ns	
	Timer clock input setup time		t_{TCKS}	25	—	ns	Figure
	Timer clock pulse width	Single-edge setting	t_{TCKWH}	1.5	—	t_{cyc}	
		Both-edge setting	t_{TCKWL}	2.5	—	t_{cyc}	
PPG	Pulse output delay time		t_{POD}	—	40	ns	Figure
8-bit timer	Timer output delay time		t_{TMOD}	—	40	ns	Figure
	Timer reset input setup time		t_{TMRS}	25	—	ns	Figure
	Timer clock input setup time		t_{TMCS}	25	—	ns	Figure
	Timer clock pulse width	Single-edge setting	t_{TMCWH}	1.5	—	t_{cyc}	
		Both-edge setting	t_{TMCWL}	2.5	—	t_{cyc}	
WDT	Overflow output delay time		t_{WOVD}	—	40	ns	Figure
SCI	Input clock cycle	Asynchronous	t_{Scyc}	4	—	t_{cyc}	Figure
		Clocked 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	t_{cyc}	

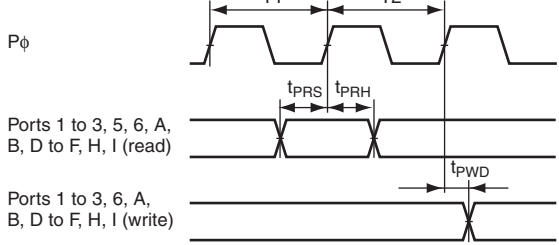
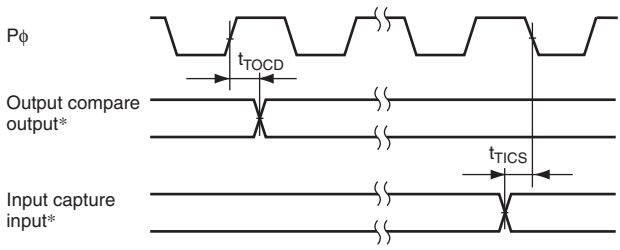


Figure 22.25 I/O Port Input/Output Timing



Note: * TIOCA0 to TIOCA5, TIOCB0 to TIOCB5, TIOCC0, TIOCC3, TIOCD0, TIOCD3

Figure 22.26 TPU Input/Output Timing

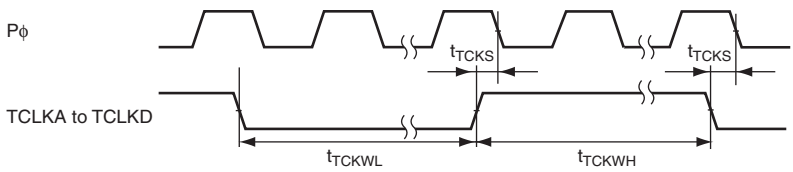


Figure 22.27 TPU Clock Input Timing

Figure 22.29 8-Bit Timer Output Timing

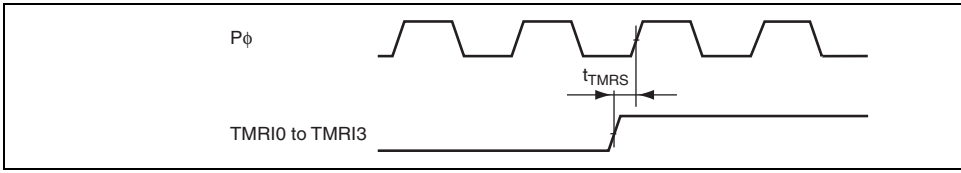


Figure 22.30 8-Bit Timer Reset Input Timing

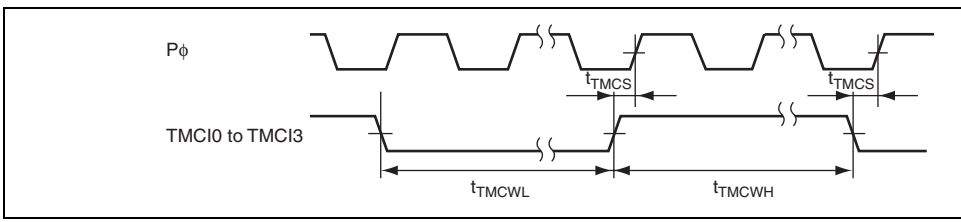


Figure 22.31 8-Bit Timer Clock Input Timing

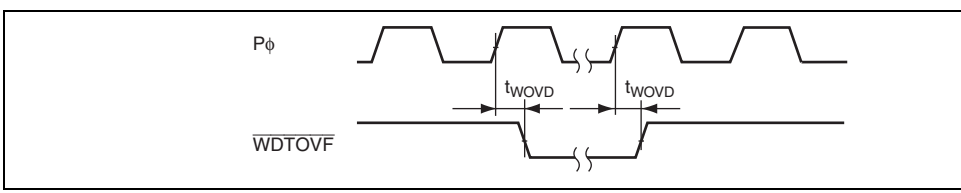


Figure 22.32 WDT Output Timing

Figure 22.34 SCI Input/Output Timing: Clocked Synchronous Mode

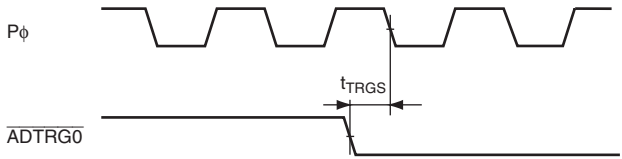


Figure 22.35 A/D Converter External Trigger Input Timing

Conversion time	7.4	—	—	μs
Analog input capacitance	—	—	20	pF
Permissible signal source impedance	—	—	10	kΩ
Nonlinearity error	—	—	±7.5	LS
Offset error	—	—	±7.5	LS
Full-scale error	—	—	±7.5	LS
Quantization error	—	±0.5	—	LS
Absolute accuracy	—	—	±8.0	LS

22.5 D/A Conversion Characteristics

Table 22.10 D/A Conversion Characteristics

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC},
V_{SS} = AV_{SS} = 0 V, Pϕ = 8 MHz to 35 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 Condi
Resolution	8	8	8	Bit	
Conversion time	—	—	10	μs	20-pF capac
Absolute accuracy	—	±2.0	±3.0	LSB	2-MΩ resisti
	—	—	±2.0	LSB	4-MΩ resisti

Item	Symbol	Min.	Typ.	Max.	Unit	Test Cond
Programming time*1, *2, *4	t_P	—	1	10	ms/128 bytes	
Erasure time*1, *2, *4	t_E	—	250	1500	ms/4-Kbyte block	
		—	500	4000	ms/32-Kbyte block	
		—	750	6500	ms/64-Kbyte block	
Programming time (total)*1, *2, *4	Σ_{tP}	—	6	18	s/768 Kbytes	T_a = 2
Erasure time (total)*1, *2, *4	Σ_{tE}	—	10	30	s/768 Kbytes	T_a = 2
Programming, Erasure time (total)*1, *2, *4	Σ_{tPE}	—	16	48	s/768 Kbytes	T_a = 2
Overwrite count	N_{WEC}	100*3	—	—	times	
Data save time*5	T_{DRP}	10	—	—	years	

Notes: 1. Programming time and erase time depend on data in the flash memory.
2. Programming time and erase time do not include time for data transfer.
3. All the characteristics after programming are guaranteed within this value (gua value is from 1 to Min. value).
4. Characteristics when programming is performed within the Min. value

Programming time*1, *2, *4	t_P	—	1	10	ms/128 bytes	
Erasure time*1, *2, *4	t_E	—	250	1500	ms/4-Kbyte block	
		—	500	4000	ms/32-Kbyte block	
		—	750	6500	ms/64-Kbyte block	
Programming time (total)*1, *2, *4	Σ_{tP}	—	4	12	s/512 Kbytes	T_a =
Erasure time (total)*1, *2, *4	Σ_{tE}	—	10	30	s/512 Kbytes	T_a =
Programming, Erasure time (total)*1, *2, *4	Σ_{tPE}	—	14	42	s/512 Kbytes	T_a =
Overwrite count	N_{WEC}	100*3	—	—	times	
Data retention time*4	T_{DRP}	10	—	—	years	

Notes: 1. Programming time and erasure time depend on the data in the flash memory.
2. Programming time and erasure time do not include time for data transfer.
3. All the characteristics after programming are guaranteed within this value (gu
value is from 1 to Min. value).
4. Characteristics when programming is performed within the Min. value

Port 2	All	Hi-Z	Hi-Z	Keep	Keep	Kee
Part3	All	Hi-Z	Hi-Z	Keep	Keep	Kee
R55 to P50	All	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Kee
P56/ AN6/ DA0/ $\overline{\text{IRQ6}}$-B	All	Hi-Z	Hi-Z	[DAOE0 = 1] Keep [DAOE0 = 0] Hi-Z	[DAOE0 = 1] Keep [DAOE0 = 0] Hi-Z	Kee
P57/ AN7/ DA1/ $\overline{\text{IRQ7}}$-B	All	Hi-Z	Hi-Z	[DAOE1 = 1] Keep [DAOE1 = 0] Hi-Z	[DAOE1 = 1] Keep [DAOE1 = 0] Hi-Z	Kee
P65 to P60	All	Hi-Z	Hi-Z	Keep	Keep	Kee
PA0/ $\overline{\text{BREQO}}$/ $\overline{\text{BS}}$-A	All	Hi-Z	Hi-Z	[$\overline{\text{BREQO}}$ output] Hi-Z [$\overline{\text{BS}}$ output] Keep [Other than above] Keep	[$\overline{\text{BREQO}}$ output] Hi-Z [$\overline{\text{BS}}$ output] Hi-Z [Other than above] Keep	[BR out BR [BS Hi-Z [Oth abo Kee

				Keep	Keep	
PA2/ $\overline{\text{BREQ}}$/ $\overline{\text{WAIT}}$	All	Hi-Z	Hi-Z	[$\overline{\text{BREQ}}$ input]	[$\overline{\text{BREQ}}$ input]	[$\overline{\text{BRE}}$
				Hi-Z	Hi-Z	Hi-Z
				[$\overline{\text{WAIT}}$ input]	[$\overline{\text{WAIT}}$ input]	[$\overline{\text{WAI}}$
				Hi-Z	Hi-Z	Hi-Z
				[Other than above]	[Other than above]	
				Keep	Keep	
PA3/ $\overline{\text{LLWR}}$/ $\overline{\text{LLB}}$	Single-chip mode (EXPE = 0)	Hi-Z	Hi-Z	Keep	Keep	Keep
	External extended mode (EXPE = 1)	H	Hi-Z	H	Hi-Z	Hi-Z
PA4/ $\overline{\text{LHWR}}$/ $\overline{\text{LUB}}$	Single-chip mode (EXPE = 0)	Hi-Z	Hi-Z	Keep	Keep	Keep
	External extended mode (EXPE = 1)	H	Hi-Z	[$\overline{\text{LHWR}}$, $\overline{\text{LUB}}$ output]	[$\overline{\text{LHWR}}$, $\overline{\text{LUB}}$ output]	[$\overline{\text{LHW}}$ outpu
				H	Hi-Z	Hi-Z
				[Other than above]	[Other than above]	[Othe abov
				Keep	Keep	Keep
PA5/$\overline{\text{RD}}$	Single-chip mode (EXPE = 0)	Hi-Z	Hi-Z	Keep	Keep	Keep
	External extended mode (EXPE = 1)	H	Hi-Z	H	Hi-Z	Hi-Z

PA7/Bϕ	Single-chip mode (EXPE = 0)	Hi-Z	Hi-Z	[Clock output]	[Clock output]	[Clo
				H	H	Clo
	External extended mode (EXPE = 1)	Clock output	Hi-Z	[Other than above]	[Other than above]	[Oth abo
				Keep	Keep	Kee
PB0/ $\overline{CS0}$/ $\overline{CS4}$/ $\overline{CS5}$-B	Single-chip mode (EXPE = 0)	Hi-Z	Hi-Z	[$\overline{CS}$ output]	[$\overline{CS}$ output]	[$\overline{CS}$
				H	Hi-Z	Hi-Z
	External extended mode (EXPE = 1)	H		[Other than above]	[Other than above]	[Oth abo
				Keep	Keep	Kee
PB1/ $\overline{CS1}$/ $\overline{CS2}$-B/ $\overline{CS5}$-A/ $\overline{CS6}$-B/ $\overline{CS7}$-B	All	Hi-Z	Hi-Z	[$\overline{CS}$ output]	[$\overline{CS}$ output]	[$\overline{CS}$
				H	Hi-Z	Hi-Z
				[Other than above]	[Other than above]	[Oth abo
				Keep	Keep	Kee
PB2/ $\overline{CS2}$-A/ $\overline{CS6}$-A	All	Hi-Z	Hi-Z	[$\overline{CS}$ output]	[$\overline{CS}$ output]	[$\overline{CS}$
				H	Hi-Z	Hi-Z
				[Other than above]	[Other than above]	[Oth abo
				Keep	Keep	Kee
PB3/ $\overline{CS3}$/ $\overline{CS7}$-A	All	Hi-Z	Hi-Z	[$\overline{CS}$ output]	[$\overline{CS}$ output]	[$\overline{CS}$
				H	Hi-Z	Hi-Z
				[Other than above]	[Other than above]	[Oth abo
				Keep	Keep	Kee

	Single-chip mode (EXPE = 0)	Hi-Z	Hi-Z	Keep	Keep	Keep
Port E	External extended mode (EXPE = 1)	L	Hi-Z	Keep	Hi-Z	Hi-Z
	ROM enabled extended mode	Hi-Z	Hi-Z	Keep	[Address output] Hi-Z [Other than above] Keep	[Add outpu Hi-Z [Othe abov Keep
	Single-chip mode (EXPE = 0)	Hi-Z	Hi-Z	Keep	Keep	Keep
PF7 to PF0	External extended mode (EXPE = 1)	L/ Hi-Z*	Hi-Z	Keep	[Address output] Hi-Z [Other than above] Keep	[Add outpu Hi-Z [Othe abov Keep
	Single-chip mode (EXPE = 0)	Hi-Z	Hi-Z	Keep	Keep	Keep
Port H	Single-chip mode (EXPE = 0)	Hi-Z	Hi-Z	Keep	Keep	Keep
	External extended mode (EXPE = 1)	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z

	32-bit bus mode	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z

[Legend]

H: High level

L: Low level

Keep: Input ports become high-impedance, and output ports retain the state.

Hi-Z: High impedance

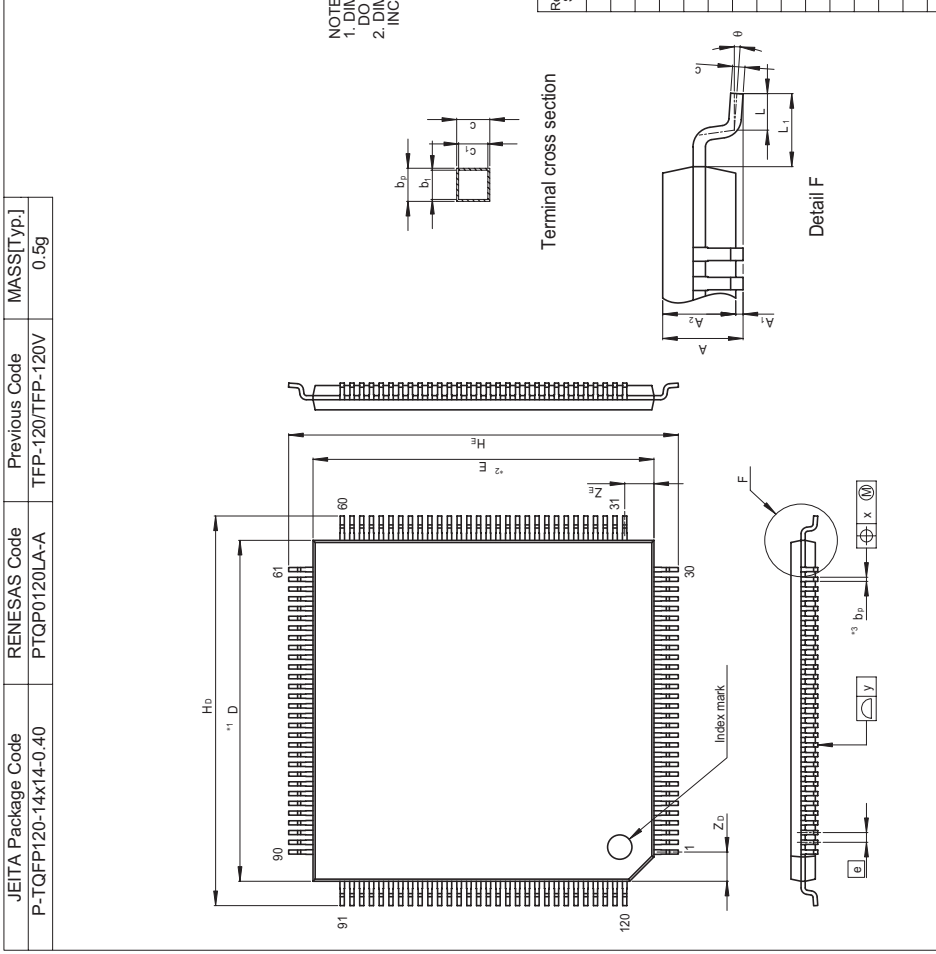


Figure C.1 Package Dimensions (TFP-120V)

MD2, MD1, MD0	(Always used as mode pins)
NMI	• Connect to V_{CC} via a pull-up resistor.
EXTAL	(Always used as a clock pin)
XTAL	• Leave the pin unconnected.
$\overline{\text{WDTOVF}}$	• Leave the pin unconnected.
Port 1 Port 2 Port 3 Port 6 PA2 to PA0 PB3 to PB1 PF7 to PF5	• Connect each pin to V_{CC} via a pull-up resistor or to V_{SS} via a pull-d resistor.
Port 5	• Connect each pin to AV_{CC} via a pull-up resistor or to AV_{SS} via a pu resistor.

	the pin unconnected.	
PB0	• Since this is the $\overline{CS0}$ output in its initial state, leave the pin unconnected.	
Port D Port E PF4 to PF0	• Since this is the address output in its initial state, leave the pin unconnected.	
Port H	(Used as a data bus)	
Port I	(Used as a data bus)	Since this is a general-purpose input port in its initial state, connect each pin to V_{CC} via a pull-up resistor or connect each pin to V_{SS} via a pull-down resistor.
V_{ref}	• Connect to AV_{CC}.	

Notes: 1. Do not change the initial value (input buffer disabled) of PnICR corresponding
unused pin.
2. When changing the pin function from its initial state, use a pull-up or pull-dow
as needed.

		Note
18.8.2 User Program Mode (2) Programming Procedure in User Program Mode	689	Modified 3. After initializing VBR to H'00000000, set th to 1 to execute download. To set the SCO of the following conditions must be satisfie : —The values of general registers other t and ER1 are held during download.
18.13 Standard Serial Communication Interface Specifications for Boot Mode (3) Inquiry and Selection States (e) Multiplication Ratio Inquiry	721	Modified

• Command, H'22, (one byte): Inquiry regarding multiplication ratio

Response	H'32	Size	Number of multiplica-tion types			
	Number of multiplication ratios	Multipli-cation ratio	· · ·			
	...					
	SUM					

• Response, H'32, (one byte): Response to the multiplication ratio inquiry

• Size (one byte): The amount of data that represents the number of multiplication multiplication ratios and the multiplication ratios

• Number of multiplication types (one byte): The number of multiplication types to can be set.

:

• Multiplication ratio (one byte)

Multiplication ratio: The value of the multiplication ratio (e.g. when the clock-freq four, the value of multiplication ratio will be H'04.)

Division ratio: The inverse of the division ratio, i.e. a negative number (e.g. wher divided by two, the value of division ratio will be H'FE. H'FE = D'-2)

The number of multiplication ratios returned is the same as the number of multip as many groups of data are returned as there are multiplication types.

		• Number of multiplication types (one byte): The number of multiplication types to wh can be set.
18.14 Usage Notes	743	Modified 15. The contents of general registers ER0 and not saved during download of an on-chip p initialization, programming, or erasure. Wh needed, save the general registers before download request or before execution of initialization, programming, or erasure usin procedure program.
19.5.1 Notes on Clock Pulse Generator	751	Deleted ~~4. Note that the frequency of ϕ will be changed middle of a bus cycle when setting SCKCR executing the external bus cycle with the wr buffer function.~~

E

F

G

H

I

S

U

Renesas 32-Bit CISC Microcomputer
Hardware Manual
H8SX/1657 Group

Publication Date:	Rev.1.00, Sep. 25, 2006
	Rev.2.00, Jun. 28, 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, J

RENESAS SALES OFFICES

http://www.ren

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,
Tel: <603> 7955-9390, Fax: <603> 7955-9510

H8SX/1657 Group
Hardware Manual

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

REJ09B0341-0200