To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

Notice

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

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

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

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

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

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

User's Manual

32

H8SX/1651 Group

Hardware Manual

Renesas 32-Bit CISC Microcomputer

H8SX Family / H8SX/1600 Series

H8SX/1651C R5S61651C

All information contained in these materials, including products and product specifications, represents information on the product at the time of publication and is subject to change by Renesas Electronics Corp. without notice. Please review the latest information published by Renesas Electronics Corp. through various means, including the Renesas Electronics Corp. website (http://www.renesas.com).

Renesas Electronics
www.renesas.com

Rev.2.00 2007.06

document, please confirm the latest product information with a Renesas sales office. Also, please pay
and careful attention to additional and different information to be disclosed by Renesas such as that dis
through our website. (http://www.renesas.com)

5. Renesas has used reasonable care in compiling the information included in this document, but Renesa
assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the infor
included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the informati
light of the total system before deciding about the applicability of such information to the intended appli
Renesas makes no representations, warranties or guaranties regarding the suitability of its products fo
particular application and specifically disclaims any liability arising out of the application and use of the
information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas
products are not designed, manufactured or tested for applications or otherwise in systems the failure o
malfunction of which may cause a direct threat to human life or create a risk of human injury or which r
especially high quality and reliability such as safety systems, or equipment or systems for transportatio
traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea commu
transmission. If you are considering the use of our products for such purposes, please contact a Renes
sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth ab
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed
(1) artificial life support devices or systems
(2) surgical implantations
(3) healthcare intervention (e.g., excision, administration of medication, etc.)
(4) any other purposes that pose a direct threat to human life
Renesas shall have no liability for damages arising out of the uses set forth in the above and purchase
elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless R
Technology Corp., its affiliated companies and their officers, directors, and employees against any and
damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with re
to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation
characteristics, installation and other product characteristics. Renesas shall have no liability for malfunc
damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have spe
characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use
conditions. Please be sure to implement safety measures to guard against the possibility of physical inj
injury or damage caused by fire in the event of the failure of a Renesas product, such as safety design
hardware and software including but not limited to redundancy, fire control and malfunction prevention,
appropriate treatment for aging degradation or any other applicable measures. Among others, since th
evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final prod
system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesa
products are attached or affixed, the risk of accident such as swallowing by infants and small children is
high. You should implement safety measures so that Renesas products may not be easily detached fro
products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior writ
approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in
document, Renesas semiconductor products, or if you have any other inquiries.

vicinity of LSI, an associated shoot-through current flows internally, and malfunct
due to the false recognition of the pin state as an input signal become possible. U
pins should be handled as described under Handling of Unused Pins in the manu

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 number
   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 te
   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/1651 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/1651 Group Hardware Manual	This r
Software Manual	Detailed descriptions of the CPU and instruction set	H8SX Software Manual	REJ0
Application Note	Examples of applications and sample programs	The latest versions are available fr web site.	
Renesas Technical Update	Preliminary report on the specifications of a product, document, etc.		

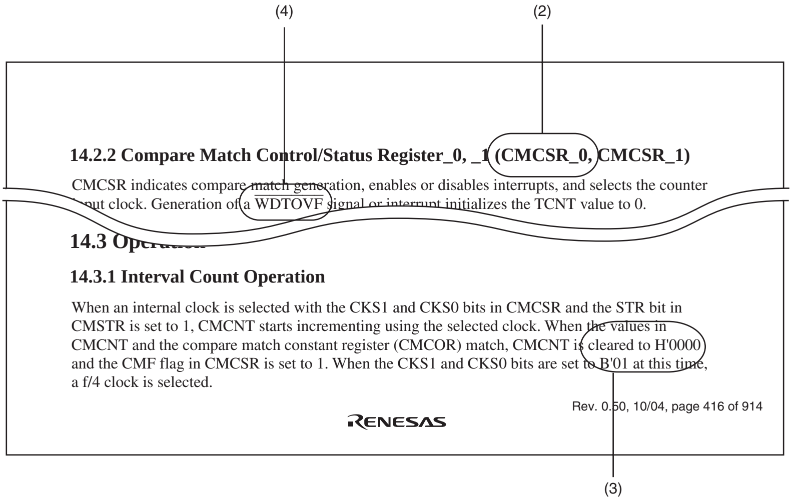
Binary numbers are given as B'nnnn (B' may be omitted if the number is obviously binary), hexadecimal numbers are given as H'nnnn or 0xnnnn, and decimal numbers are given as nnnn.

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



14.2.2 Compare Match Control/Status Register_0, _1 (CMCSR_0, CMCSR_1)

CMCSR indicates compare match generation, enables or disables interrupts, and selects the counter input clock. Generation of a $\overline{\text{WDTOVF}}$ signal or interrupt initializes the TCNT value to 0.

14.3 Operation

14.3.1 Interval Count Operation

When an internal clock is selected with the CKS1 and CKS0 bits in CMCSR and the STR bit in CMSTR is set to 1, CMCNT starts incrementing using the selected clock. When the values in CMCNT and the compare match constant register (CMCOR) match, CMCNT is cleared to H'0000 and the CMF flag in CMCSR is set to 1. When the CKS1 and CKS0 bits are set to B'01 at this time, a f/4 clock is selected.

Rev. 0.50, 10/04, page 416 of 914

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 14	– –	0 0	R R	Reserved 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 conten 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

All trademarks and registered trademarks are the property of their respective owners.

Section 2 CPU

Section 3 MCU Operating Modes ....

Section 4 Exception Handling..

Section 8 Data Transfer Controller (DTC)..

Section 13 Watchdog Timer (WDT)...

Section 14 Serial Communication Interface (SCI) ....

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
kept down dynamically by an on-chip power-management function.

1.1.1 Applications

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

	Upward compatibility for H8/300, H8/300H, and H8S Cl 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 $I\phi$ = 50 MHz and V_{CC} = 3.0 to 3.6 V) • On-chip multiplier (16 × 16 → 32 bits) • Supports multiply-and-accumulate instructions (16 × 16 + 32 → 32 bits)
Operating mode	• Advanced mode
MCU operating mode	Mode 4: On-chip ROM disabled external extended mode, 1 bus (selected by driving the MD0 pin low) Mode 5: On-chip ROM disabled external extended mode, 8 (selected by driving the MD0 pin high) • Low power consumption state (transition driven by the instruction)

	controller (DMAC)	• Three activation methods (auto-request, on-chip modu interrupt, external request) • Three transfer modes (normal transfer, repeat transfe transfer) • Dual or single address mode selectable • Extended repeat-area function
	Data transfer controller (DTC)	• Allows DMA transfer over 55 channels (number of DT activation sources) • Activated by interrupt sources (chain transfer enabled • Three transfer modes (normal transfer, repeat transfe 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 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 e — 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 am internal CPU and DTC, and external bus masters)

	generator (CPG)	• Separate clock signals are provided for each of functio modules (detailed below) and each is independently sp (multi-clock function) — System-intended data transfer modules, i.e. the CP in synchronization with the system clock (Iϕ): 8 to 5 — Internal peripheral functions run in synchronization peripheral module clock (Pϕ): 8 to 35 MHz — Modules in the external space are supplied with the bus clock (Bϕ): 8 to 50 MHz • Includes a PLL frequency multiplication circuit and freq divider, so the operating frequency is selectable • Five low-power-consumption modes: Sleep mode, mod mode, all-module-clock-stop mode, software standby m and hardware standby mode
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 m clock (Pϕ) at 35-MHz operation) • Two operating modes: single mode and scan mode • Three ways to start A/D conversion: software, timer (TF trigger, and external trigger

	pulse unit (TPU)	• Select from among eight counter-input clocks for each • Up to 16 pulse inputs and outputs • Counter clear operation, simultaneous writing to multi counters (TCNT), simultaneous clearing by compare r input capture possible, simultaneous input/output for r possible by counter synchronous operation, and up to PWM output possible by combination with synchronou operation • Buffered operation, cascaded operation (32 bits × two channels), and phase counting mode (two-phase enco input) settable for each channel • Input capture function supported • Output compare function (by the output of compare m waveform) supported
	Program-mable pulse generator (PPG)	• 16-bit pulse output • Four output groups, non-overlapping mode, and inver can be set • Selectable output trigger signals; the PPG can operat conjunction with the data transfer controller (DTC) and controller (DMAC)
Watchdog timer	Watchdog timer (WDT)	• 8 bits × one channel (selectable from eight counter inp • Switchable between watchdog timer mode and interva mode

	• 11 pull-up resistors • 11 open drains
Package	• 120-pin thin QFP package (package code: FP-120B, pa dimensions: 14 × 14 mm, pin pitch: 0.40 mm) • Lead- (Pb-) free versions available
Operating frequency/ Power supply voltage	• Operating frequency: 8 to 50 MHz • Power supply voltage: Vcc = 3.0 to 3.6 V, Avcc = 3.0 to • Supply current: — 30 mA (typ.) (Vcc = 3.3 V, Avcc = 3.3 V, Iϕ = Pϕ = B 35 MHz) — 45 mA (typ.) (Vcc = 3.3 V, Avcc = 3.3 V, Iϕ = Bϕ = 5 Pϕ = 25 MHz)
Operating peripheral temperature (°C)	• –20 to +75°C (regular specifications) • –40 to +85°C (wide-range specifications)

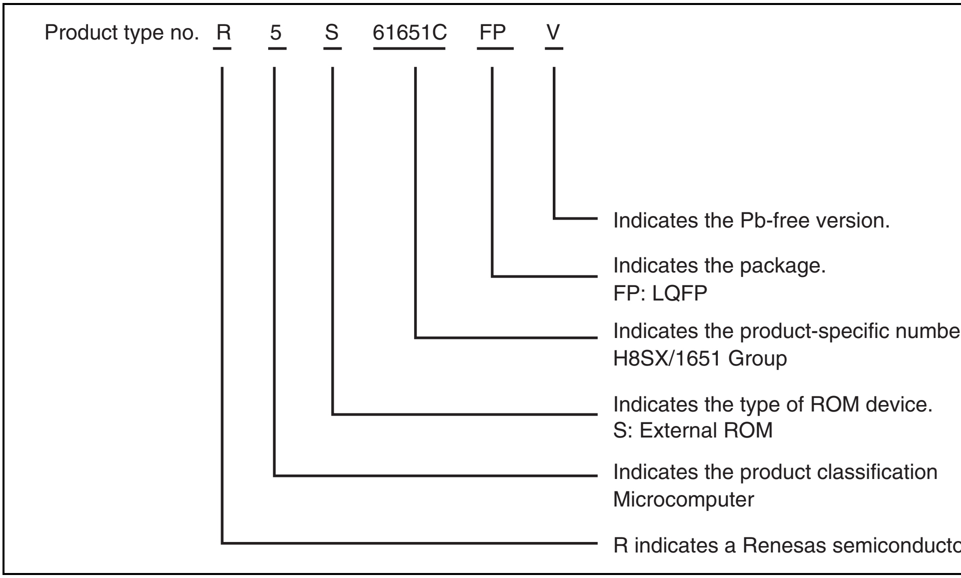

Product type no.
R
5
S
61651C
FP
V
Indicates the Pb-free version.
Indicates the package.
FP: LQFP
Indicates the product-specific numbe
H8SX/1651 Group
Indicates the type of ROM device.
S: External ROM
Indicates the product classification
Microcomputer
R indicates a Renesas semiconducto

Figure 1.1 How to Read the Product Name Code

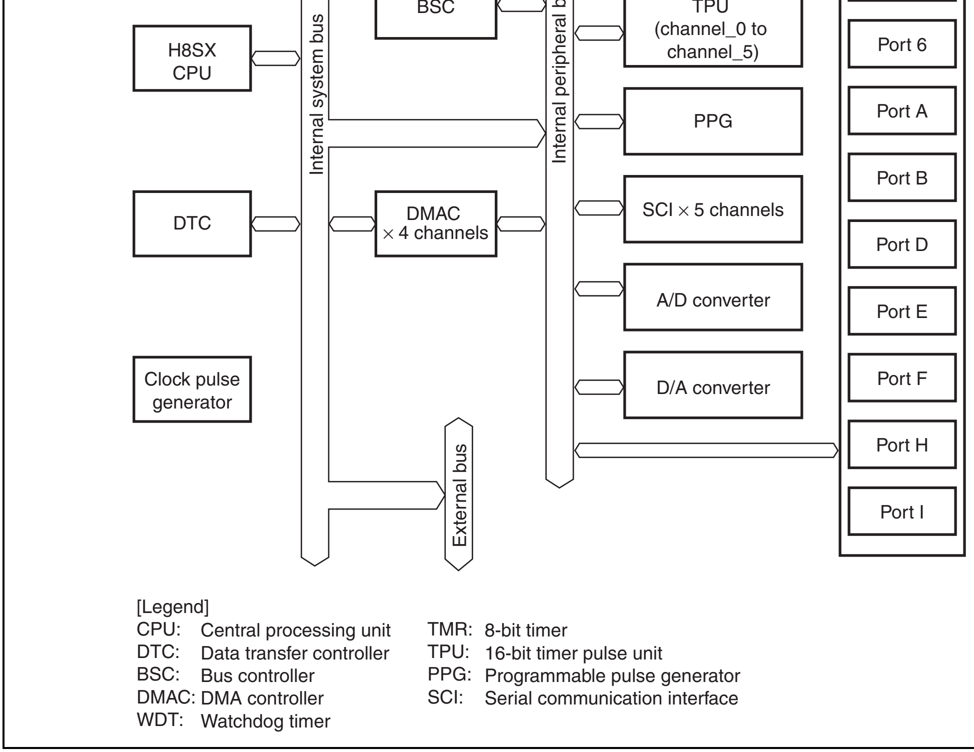
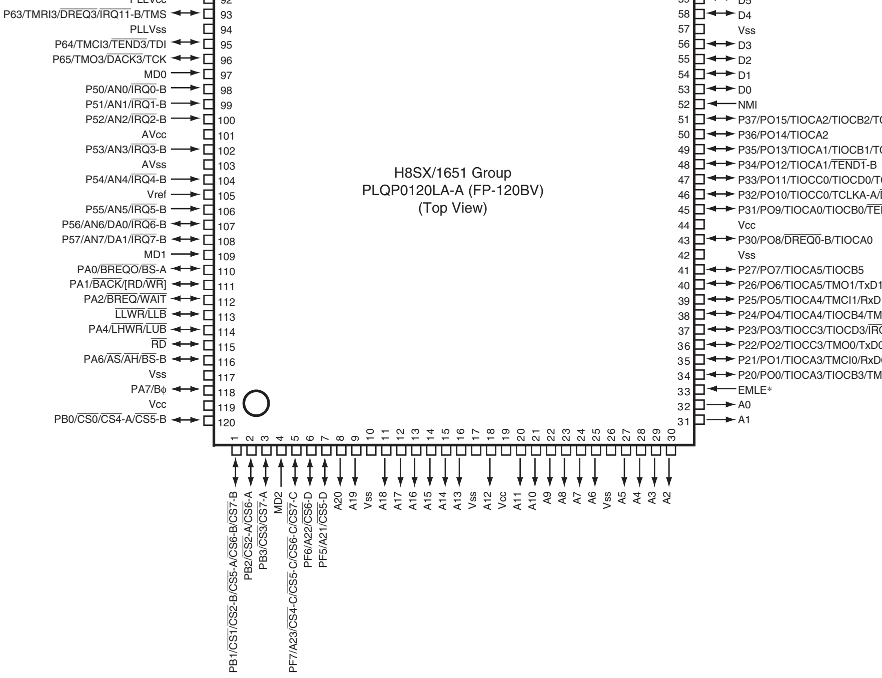


Figure 1.2 Block Diagram



Note: * This is an on-chip emulator enable pin. Drive this pin low for the connection in normal operating mode. When this pin is driven high, the on-chip emulation function is enabled and pins P62, P63, P64, P65, and WDTOVF are dedicated for the on-chip emulator. For details on a connection example with the E10A, refer to the E10A Emulator User's Manual.

Figure 1.3 Pin Assignments

	PLLV_{CC}	—	Power supply pin for the PLL circuit.
	PLLV_{SS}	—	Ground pin for the PLL circuit.
Clock	XTAL	Input	Pins for a crystal resonator. An external clock sign input through the EXTAL pin. For an example of th connection, see section 18, Clock Pulse Generato
	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 lev these pins must not be changed during operation.
System control	$\overline{\text{RES}}$	Input	Reset signal input pin. This LSI enters the reset st this signal goes low.
	$\overline{\text{STBY}}$	Input	This LSI enters hardware standby mode when this goes low.
	$\overline{\text{EMLE}}$	Input	Input pin for the on-chip emulator enable signal. Th 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. The also output addresses when accessing an address multiplexed I/O interface space.
Bus control	$\overline{\text{BREQ}}$	Input	External bus-master modules assert this signal to the bus.
	$\overline{\text{BREQO}}$	Output	Internal bus-master modules assert this signal to r access to the external space via the bus in the ext released state.

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

	$\overline{\text{IRQ2}}$-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 signa
	$\overline{\text{TEND0}}$-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 u input capture inputs, output compare outputs, or P outputs.
	TIOCA1 TIOCB1	Input/ output	Signals for TGRA_1 and TGRB_1. These pins are input capture inputs, output compare outputs, or P outputs.

	TIOCA5 TIOCB5	Input/ output	Signals for TGRA_5 and TGRB_5. These pins ar input capture inputs, output compare outputs, or 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 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 watc 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.

			to the system power supply (0 V).
	Vref	Input	Reference power supply pin for the A/D and D/A c When the A/D and D/A converters are not in use, c this pin to the system power supply.
I/O ports	P17 to P10	Input/ output	8 input/output pins.
	P27 to P20	Input/ output	8 input/output pins.
	P37 to P30	Input/ output	8 input/output pins.
	P57 to P50	Input	8 input/output pins.
	P65 to P60	Input/ output	6 input/output pins.
	PA7, PA6, PA4 PA2 to PA0	Input/ output	5 input/output pins.
	PB3 to PB0	Input/ output	4 input/output pins.
	PF7 to PF5	Input/ output	3 input/output pins.
	PI7 to PI0	Input/ output	8 input/output pins.

Can execute these CPU's object programs

- General-register architecture

  Sixteen 16-bit general registers (also usable as sixteen 8-bit registers or eight 32-bit r

- 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,
  @(d:32,Rn.W), @(d:16,ERn.L), or @(d:32,ERn.L)]

  Register indirect with post-/pre-increment or post-/pre-decrement
  [@+ERn/@−ERn/@ERn+/@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 @(ER

  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/1651 Group.

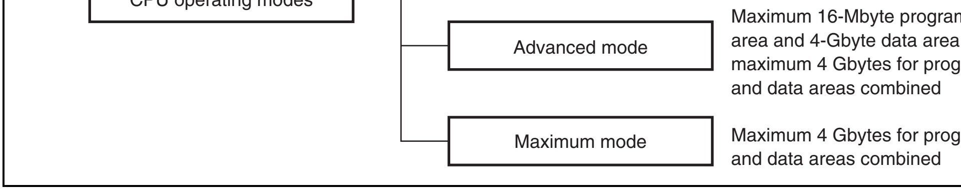


Figure 2.1 CPU Operating Modes

2.2.1 Normal Mode

In normal mode, the exception handling vector table and stack have the same structure a H8/300 CPU.

Note: This LSI does not support this mode.

- Address Space

  A maximum address space of 64 kbytes can be accessed.

- Extended Registers (En)

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

- Instruction Set

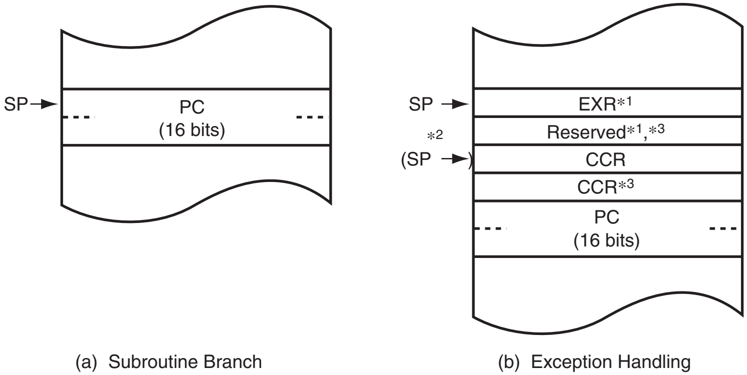
  All instructions and addressing modes can be used. Only the lower 16 bits of effectiv addresses (EA) are valid.

Figure 2.2   Exception Handling Vector Table (Normal Mode)

The memory indirect (@@aa:8) and extended memory indirect (@@vec:7) addressing m used in the JMP and JSR instructions. An 8-bit absolute address included in the instructic specifies a memory location. Execution branches to the address contained in the memory

- Stack Structure

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



(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 2.3   Stack Structure in 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 En is used as a 16-bit register (in other than the J JSR instructions), it can contain any value even when the corresponding general regi is used as an address register. If the general register Rn is referenced in the register i addressing mode with pre-/post-increment or decrement and a carry or borrow occur however, the value in the corresponding extended register 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 Handling Vector Table and Memory Indirect Branch Addresses

  In middle mode, the top area starting at H'000000 is allocated to the exception handl table in 32-bit units. In each 32 bits, the upper eight bits are ignored and one branch stored in the lower 24 bits. The structure of the exception handling vector table is sh 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 branch are shown in figure 2.5. The PC contents are saved or restored in 24

- Instruction Set

  All instructions and addressing modes can be used.

- Exception Handling Vector Table and Memory Indirect Branch Addresses

  In advanced mode, the top area starting at H'00000000 is allocated to the exception ha vector table in 32-bit units. In each 32 bits, the upper eight bits are ignored and one br address is stored in the lower 24 bits. The structure of the exception handling vector ta shown in figure 2.4.

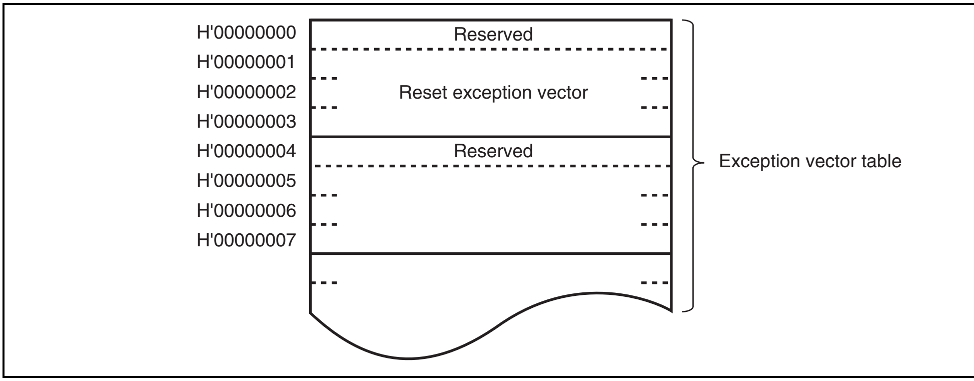


Figure 2.4 Exception Handling Vector Table (Middle and Advanced Modes

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

In advanced mode, an operand is a 32-bit (longword) operand, providing a 32-bit branch 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 in Middle and Advanced Modes

2.2.4 Maximum Mode

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

- Address Space

  A 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 Handling Vector Table and Memory Indirect Branch Addresses

  In maximum mode, the top area starting at H'00000000 is allocated to the exception vector table in 32-bit units. One branch address is stored in 32 bits. The structure of t exception handling vector table is shown in figure 2.6.

Figure 2.6 Exception Handling Vector Table (Maximum Modes)

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

In maximum mode, an operand is a 32-bit (longword) operand, providing a 32-bit branch

- Stack Structure

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

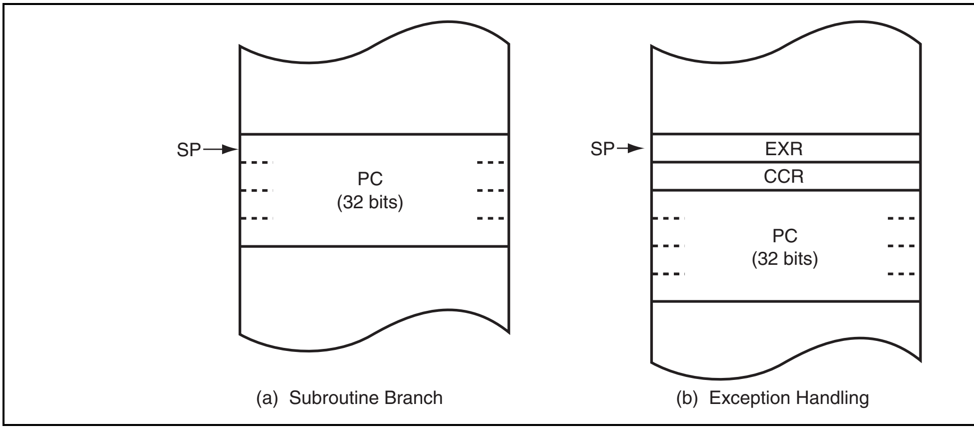


Figure 2.7 Stack Structure in Maximum Mode

2.4 Address Space

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

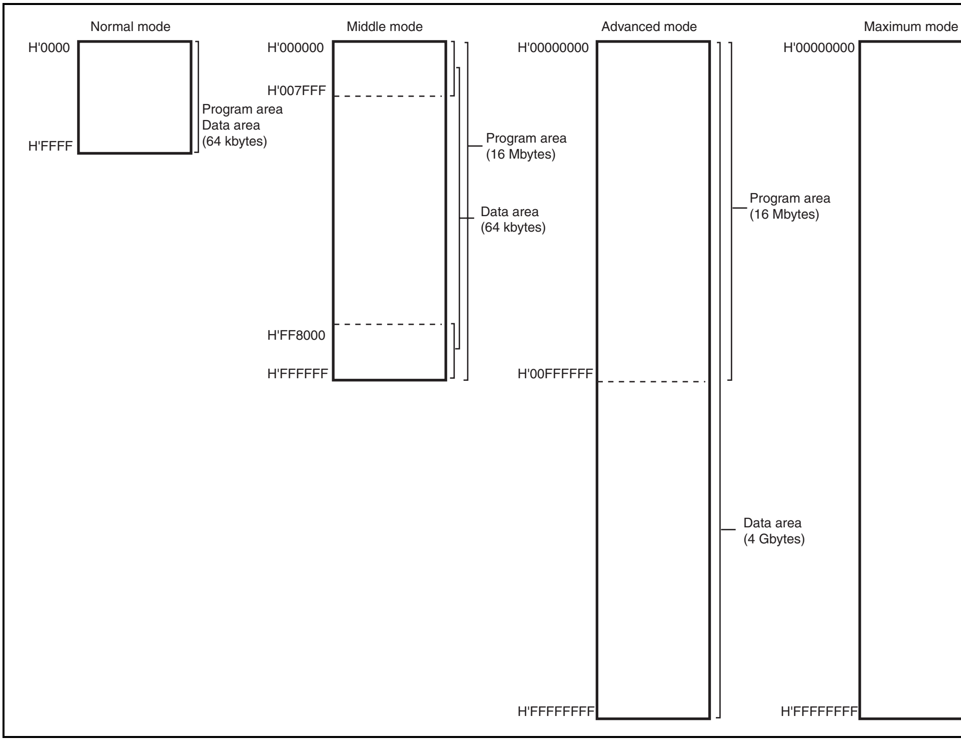
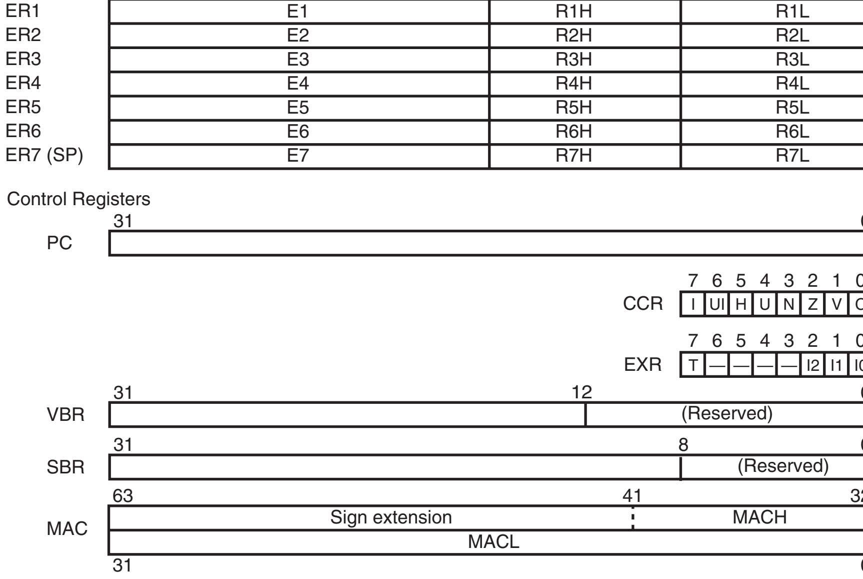


Figure 2.8 Memory Map



[Legend]

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

Figure 2.9 CPU Registers

and R (R0 to R7). These registers are functionally equivalent, providing a maximum of s
bit registers. The E registers (E0 to E7) are also referred to as extended registers.

The R registers are divided into 8-bit general registers designated by the letters RH (R0F
and RL (R0L to R7L). These registers are 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.

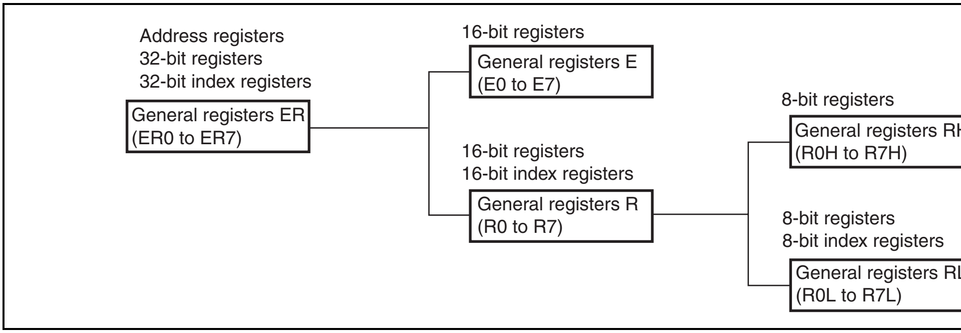


Figure 2.10 Usage of General Registers

General register ER7 has the function of stack pointer (SP) in addition to its general-reg
function, and is used implicitly in exception handling and subroutine calls. Figure 2.11 s
stack.

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 two bytes (one word) or a multiple of two bytes, so the l significant PC bit is ignored. When an instruction is fetched, the least significant PC bit i regarded as 0.

2.5.3 Condition-Code Register (CCR)

CCR is an 8-bit register that contains internal CPU status information, including an inter (I) and user (UI, U) bits and half-carry (H), negative (N), zero (Z), overflow (V), and car flags.

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

Bit	Bit Name	Initial Value	R/W	Description
7	I	1	R/W	Interrupt Mask Bit Masks interrupts when set to 1. This bit is set t start of an exception-handling sequence.

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

7	T	0	R/W	Trace Bit Selects trace mode. When this bit is cleared to 0 instructions are executed in sequence. When thi set to 1, a trace exception is generated each tim instruction is executed.
6 to 3	—	All 1	R/W	Reserved These bits are always read as 1. The write value always be 1.
2	I2	1	R/W	Interrupt Mask Bits
1	I1	1	R/W	These bits designate the interrupt mask level (0
0	I0	1	R/W	

2.5.5 Vector Base Register (VBR)

VBR is a 32-bit register that has the valid upper 20 bits. The lower 12 bits of this register as 0s. This register value is a base address of the vector area for exception handling other reset and a CPU address error (extended memory indirect is also out of the target). The in value is H'00000000.

2.5.6 Short Address Base Register (SBR)

SBR is a 32-bit register that has the valid upper 24 bits. The lower eight bits are read as 0 absolute addressing mode (@aa:8), this register is used as the upper address. The initial v H'FFFFFF00.

bits, MAC and the general registers are not initialized. In particular, the stack pointer (E
initialized. The stack pointer should therefore be initialized using an MOV.L instruction
immediately after a reset.

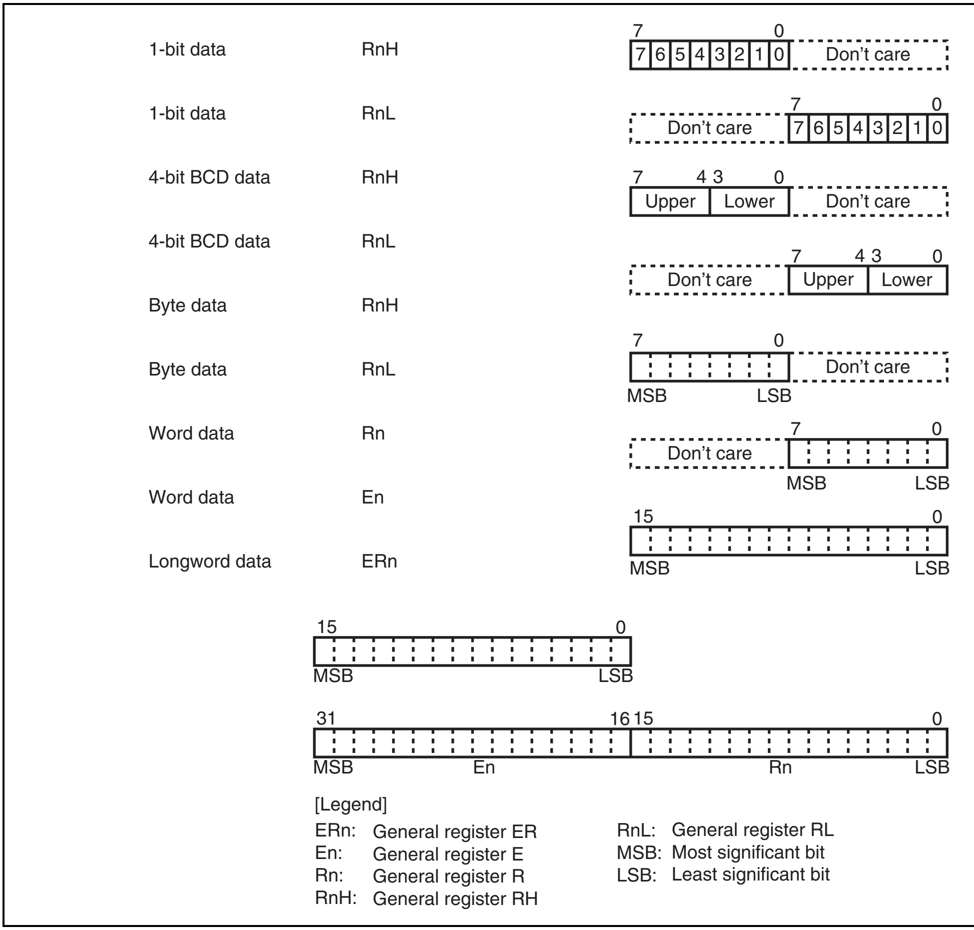


Figure 2.12 General Register Data Formats

the stack manipulation, block transfer instructions, and MAC instruction should be locat
addresses.

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

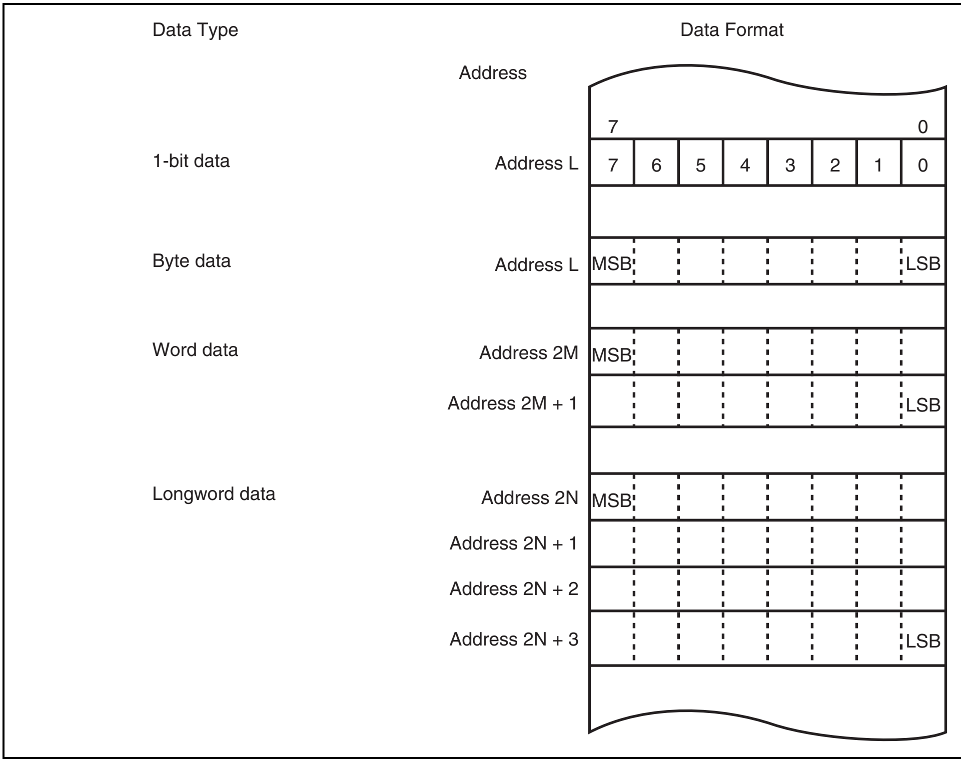


Figure 2.13 Memory Data Formats

Data transfer	MOV	B/W/L
	MOVFPE*6, MOVTPE*6	B
	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	—

	RTS/L	L
	BRA/S	—
System control	TRAPA, RTE, SLEEP, NOP	—
	RTE/L	L*[5]
	LDC, STC, ANDC, ORC, XORC	B/W/
		Total

[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 Rn, @−SP. POP.L ERn and PUSH.L ERn are identical to MOV.L @S and MOV.L ERn, @−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 a general register to be restored

6. Not supported in this LSI

cation	Instruction	Size	#xx	Rn	@ERn	@(d,ERn)	ERn.L)	@+ERn	@aa:8	@aa
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

	DAA, DAS	B		D						
	MULXU, DIVXU	B/W	S:4	SD						
	MULU, DIVU	W/L	S:4	SD						
	MULXS, DIVXS	B/W	S:4	SD						
	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

Bit manipulation	BSET, BCLR, BNOT, BTST, BSET/cc, BCLR/cc	B	D	D	D	D
	BAND, BIAND, BOR, BIOR, BXOR, BIXOR, BLD, BILD, BST, BIST, BSTZ, BISTZ	B	D	D	D	D
	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

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 source or destination operand or both.

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

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

O: Can be used.

Notes:
1. @aa:16 is only 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 transfer
4. Size of data to be added with a displacement
5. @ERn− is only available.
6. When the number of bits to be shifted is 1, 2, 4, 8, or 16
7. When the number of bits to be shifted is specified by 5-bit immediate data or register
8. Size of data to specify a branch condition
9. Byte for immediate or register direct; otherwise, word
10. @ERn+ is only available.
11. @−ERn is only available.
12. Not supported in this LSI

	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: * @(d:8, PC) is only 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 reg 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 general registers. Two, three general registers which have serial register numbers can be spec
STM	L	Rn (register list) → @−SP Saves the contents of general registers on the stack. Two, three, general registers which have serial register numbers can be spec
MOVA	B/W	EA → Rd Zero-extends the contents of a specified general register or mem and adds them with a displacement. The result is stored in a gen register.

Note: * Not supported in this LSI

MOVMD.W	W	Transfers a data block. Transfers word data from a memory location specified by ER5 t memory location specified by ER6. The number of word data to transferred is specified by R4.
MOVMD.L	L	Transfers a data block. Transfers longword data from a memory location specified by E memory location specified by ER6. The number of longword da transferred is specified by R4.
MOVSD.B	B	Transfers a data block with zero data detection. Transfers byte data from a memory location specified by ER5 t memory location specified by ER6. The number of byte data to transferred is specified by R4. When zero data is detected durin the transfer stops and execution branches to a specified addres

		register indirect addressing mode.
INC DEC	B/W/L	Rd ± 1 → Rd, Rd ± 2 → Rd Increments or decrements a general register by 1 or 2. (Byte ope can be incremented or decremented by 1 only.)
ADDS SUBS	L	Rd ± 1 → Rd, Rd ± 2 → Rd, Rd ± 4 → Rd Adds or subtracts the value 1, 2, or 4 to or from data in a general
DAA DAS	B	Rd decimal adjust → Rd Decimal-adjusts an addition or subtraction result in a general reg 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 registers 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 registers 16 bits × 16 bits → 16 bits or 32 bits × 32 bits → 32 bits.
MULU/U	L	Rd × Rs → Rd Performs unsigned multiplication on data in two general registers × 32 bits → upper 32 bits).
MULXS	B/W	Rd × Rs → Rd Performs signed multiplication on data in two general registers: e bits × 8 bits → 16 bits or 16 bits × 16 bits → 32 bits.

		quotient and 16-bit remainder.
DIVU	W/L	Rd ÷ Rs → Rd Performs unsigned division on data in two general registers: eit ÷ 16 bits → 16-bit quotient or 32 bits ÷ 32 bits → 32-bit quotient
DIVXS	B/W	Rd ÷ Rs → Rd Performs signed division on data in two general registers: either 8 bits → 8-bit quotient and 8-bit remainder or 32 bits ÷ 16 bits – quotient and 16-bit remainder.
DIVS	W/L	Rd ÷ Rs → Rd Performs signed division on data in two general registers: either 16 bits → 16-bit quotient or 32 bits ÷ 32 bits → 32-bit quotient.
CMP	B/W/L	(EAd) – #IMM, (EAd) – (EAs) Compares data between immediate data, general registers, and and stores CCR bits according to the result.
NEG	B/W/L	0 – (EAd) → (EAd) Takes the two's complement (arithmetic complement) of the co general register or a memory location.
EXTU	W/L	(EAd) (zero extension) → (EAd) Extends the lower 8 or 16 bits of data in a general register or a location to word or longword size by padding with 0s. The lower eight bits can be extended to word or longword, or lo to longword.

		the MAC.
CLRMAC	—	0 → MAC Clears the MAC to zero.
LDMAC	—	Rs → MAC Loads data from a general register to the MAC.
STMAC	—	MAC → Rd Stores data from the MAC to a general register.

Table 2.7 Logic Operation Instructions

Instruction	Size	Function
AND	B/W/L	(EAd) ∧ #IMM → (EAd), (EAd) ∧ (EAs) → (EAd) Performs a logical AND operation on data between immediate da general registers, and memory.
OR	B/W/L	(EAd) ∨ #IMM → (EAd), (EAd) ∨ (EAs) → (EAd) Performs a logical OR operation on data between immediate dat general registers, and memory.
XOR	B/W/L	(EAd) ⊕ #IMM → (EAd), (EAd) ⊕ (EAs) → (EAd) 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 (logical complement) of the content general register or a memory location.

SHAR		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 flag. 1-bit or 2-bit rotation is possible.

Table 2.9 Bit Manipulation Instructions

Instruction	Size	Function
BSET	B	1 → (<bit-No.> of <EAd>) Sets a specified bit in the contents of a general register or a me location to 1. The bit number is specified by 3-bit immediate dat lower three bits of a general register.
BSET/cc	B	if cc, 1 → (<bit-No.> of <EAd>) If the specified condition is satisfied, this instruction sets a spec a memory location to 1. The bit number can be specified by 3-b immediate data, or by the lower three bits of a general register. status can be specified as a condition.
BCLR	B	0 → (<bit-No.> of <EAd>) Clears a specified bit in the contents of a general register or a m location to 0. The bit number is specified by 3-bit immediate dat lower three bits of a general register.

		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 numbe specified by 3-bit immediate data or the lower three bits of a gen register.
BAND	B	C ∧ (<bit-No.> of <EAd>) → C Logically ANDs the carry flag with a specified bit in the contents o general register or a memory location and stores the result in the flag. The bit number is specified by 3-bit immediate data.
BIAND	B	C ∧ [~ (<bit-No.> of <EAd>)] → C Logically ANDs the carry flag with the inverse of a specified bit in contents of a general register or a memory location and stores th in the carry flag. The bit number is specified by 3-bit immediate data.
BOR	B	C ∨ (<bit-No.> of <EAd>) → C Logically ORs the carry flag with a specified bit in the contents of general register or a memory location and stores the result in the flag. The bit number is specified by 3-bit immediate data.
BIOR	B	C ∨ [~ (<bit-No.> of <EAd>)] → C Logically ORs the carry flag with the inverse of a specified bit in t contents of a general register or a memory location and stores th in the carry flag. The bit number is specified by 3-bit immediate data.
BXOR	B	C ⊕ (<bit-No.> of <EAd>) → C Logically exclusive-ORs the carry flag with a specified bit in the c of a general register or a memory location and stores the result i 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 gene 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 o 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 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 contents of a general register or a memory location. The bit number is specified by 3-bit immediate data.
BISTZ	B	~ Z → (<bit-No.> of <EAd>) Transfers the inverse of the zero flag value to a specified bit in t contents of a memory location. The bit number is specified by 3-bit immediate data.
BFLD	B	(EAs) (bit field) → Rd Transfers a specified bit field in memory location contents to the of a specified general register.
BFST	B	Rd → (EAd) (bit field) Transfers the lower bits of a specified general register to a spec field in memory location contents.

		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 gener registers.

		location to CCR or EXR. Although CCR and EXR are 8-bit regis word-size transfers are performed between them and memory. T eight bits are valid.
	L	Rs → VBR, Rs → SBR Transfers the general register contents to VBR or SBR.
STC	B/W	CCR → (EAd), EXR → (EAd) Transfers CCR or EXR contents to a general register or memor Although CCR and EXR are 8-bit registers, word-size transfers performed between them and memory. The upper eight bits are
	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.

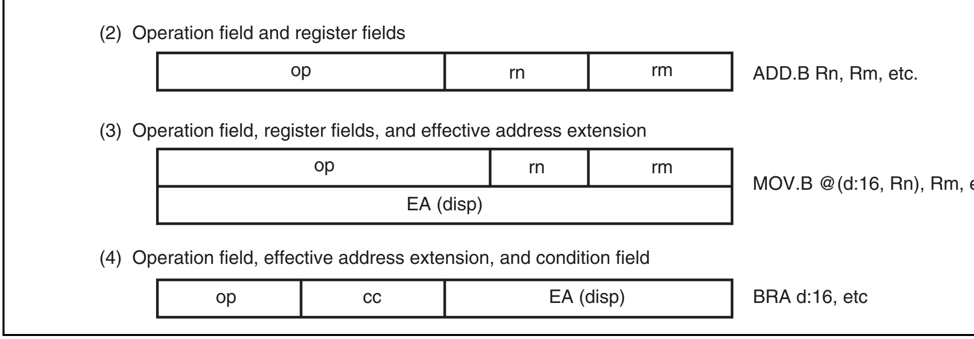

(2) Operation field and register fields
op
rn
rm
ADD.B Rn, Rm, etc.
(3) Operation field, register fields, and effective address extension
op
rn
rm
EA (disp)
MOV.B @(d:16, Rn), Rm, e
(4) Operation field, effective address extension, and condition field
op
cc
EA (disp)
BRA d:16, etc

Figure 2.14 Instruction Formats

- Operation Field

  Indicates the function of the instruction, the addressing mode, and the operation to be out on the operand. The operation field always includes the first four bits of the instru 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

  Eight, 16, or 32 bits specifying immediate data, an absolute address, or a displacemen

- Condition Field

  Specifies the branching 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 in the instruction code.

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

2.8.3 Register Indirect with Displacement—@(d:2, ERn), @(d:16, ERn), or @(d:32, ERn)

The operand value is the contents of a memory location which is pointed to by the sum of
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: w
displacement is 1, 2, or 3 and the operand is byte data, when a displacement is 2, 4, or 6 a
operand is word data, or when a displacement is 4, 8, or 12 and the operand is longword d

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 of
following operation result and a 16- or 32-bit displacement: specified bits of the contents
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 instruction code and the 16-bit displacement is sign-e
when added to ERn. If the operand is byte data, ERn is multiplied by 1. If the operand is
longword data, ERn is multiplied by 2 or 4, respectively.

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

- Register indirect with pre-increment—@+ERn

  The operand value is the contents of a memory location which is pointed to by the fo operation result: the value 1, 2, or 4 is added to the contents of an address register (E is specified by the register field in the instruction code. After that, the sum is stored i address register. The value added is 1 for byte access, 2 for word access, or 4 for lon access.

- Register indirect with post-decrement—@ERn–

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

If the contents of a general register which is also used as an address register is written to using this addressing mode, data to be written is the contents of the general register afte calculating an effective address. If the same general register is specified in an instructio effective addresses are calculated, the contents of the general register after the first calcu an effective address is used in the second calculation of an effective address.

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:2 32-bit (@aa:32) absolute addresses.

To access the data area, the absolute address of eight bits (@aa:8), 16 bits (@aa:16), or 3 (@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 eight bits are all assumed to be 0 (H'00).

	(@aa:32)		H'FFFFFF		
Program area	24 bits (@aa:24)		H'000000 to H'FFFFFF	H'00000000 to H'00FFFF	
	32 bits (@aa:32)			H'00000000 to H'00FFFFFF	H'0000 H'FFF

2.8.7 Immediate—#xx

The operand value is 8-bit (#xx:8), 16-bit (#xx:16), or 32-bit (#xx:32) data included in th instruction code. This addressing mode has short formats in which 3- or 4-bit immediate be used.

When the size of immediate data is less than that of the operand size (byte, word, or long the immediate data is zero-extended.

The ADDS, SUBS, INC, and DEC instructions contain immediate data implicitly. Some manipulation instructions contain 3-bit immediate data in the instruction code, specifyin number. The BFLD and BFST instructions contain 8-bit immediate data in its instructio specifying bit numbers. The TRAPA instruction contains 2-bit immediate data in its inst code, specifying a vector address.

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 specified bits of the contents of an address register (RnL, Rn, or ERn) specified by the reg field in the instruction code is zero-extended to 32-bit data and multiplied by 2.

The PC content to which the displacement is added is the address of the first byte of the n instruction. In advanced mode, only the lower 24 bits of this branch address are valid; the eight bits are all assumed to be 0 (H'00).

2.8.10 Memory Indirect—@@aa:8

This mode is used in the JMP and JSR instructions. The operand value is a branch addres is the content of a memory location pointed to by an 8-bit absolute address in the instructi

The upper bits of an 8-bit absolute address are all assumed to be 0, so the address range to branch address is 0 to 255 (H'0000 to H'00FF in normal mode, H'000000 to H'0000FF in modes). In normal mode, the memory location is pointed to by word-size data and the bra address is 16 bits long. In other modes, the memory location is pointed to by longword-si In middle or advanced mode, the first byte of the longword-size data is assumed to be all

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

(a) Normal Mode (b) Advanced Mode

Figure 2.15 Branch Address Specification in Memory Indirect Mode

2.8.11 Extended Memory Indirect—@@vec:7

This mode is used in the JMP and JSR instructions. The operand value is a branch addre is the contents of a memory location pointed to by the following operation result: the su data in the instruction code and the value of H'80 is multiplied by 2 or 4.

The address range to store a branch address is H'0100 to H'01FF in normal mode and H' H'0003FF in other modes. In assembler notation, an address to store a branch address is

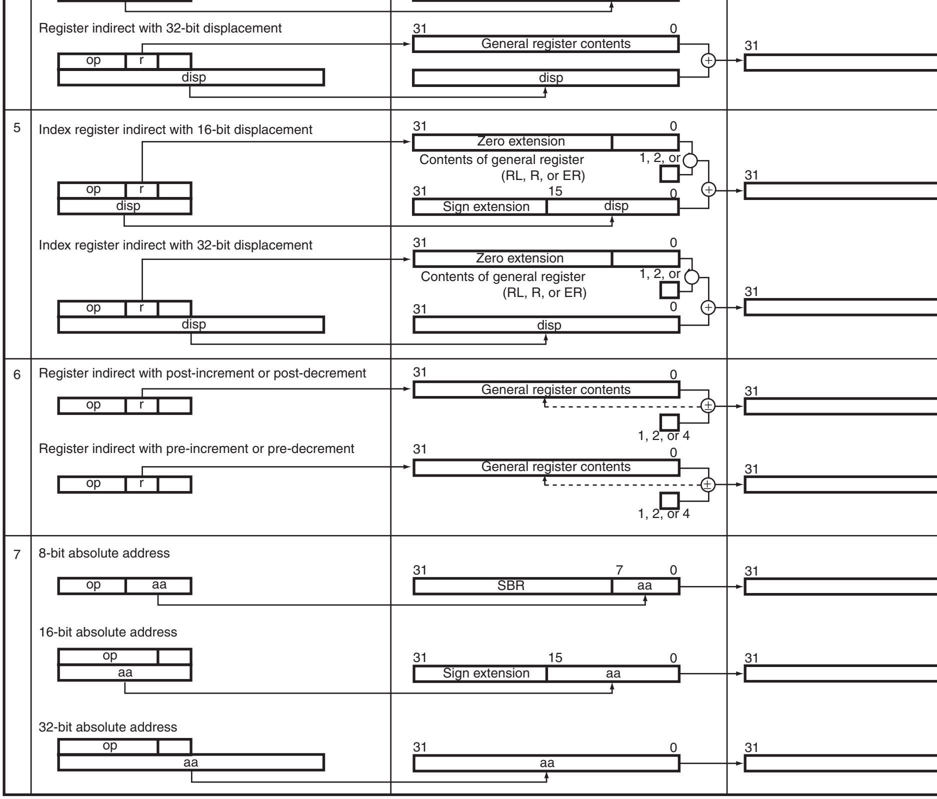
In normal mode, the memory location is pointed to by word-size data and the branch ad bits long. In other modes, the memory location is pointed to by longword-size data. In m advanced mode, the first byte of the longword-size data is assumed to be all 0 (H'00).

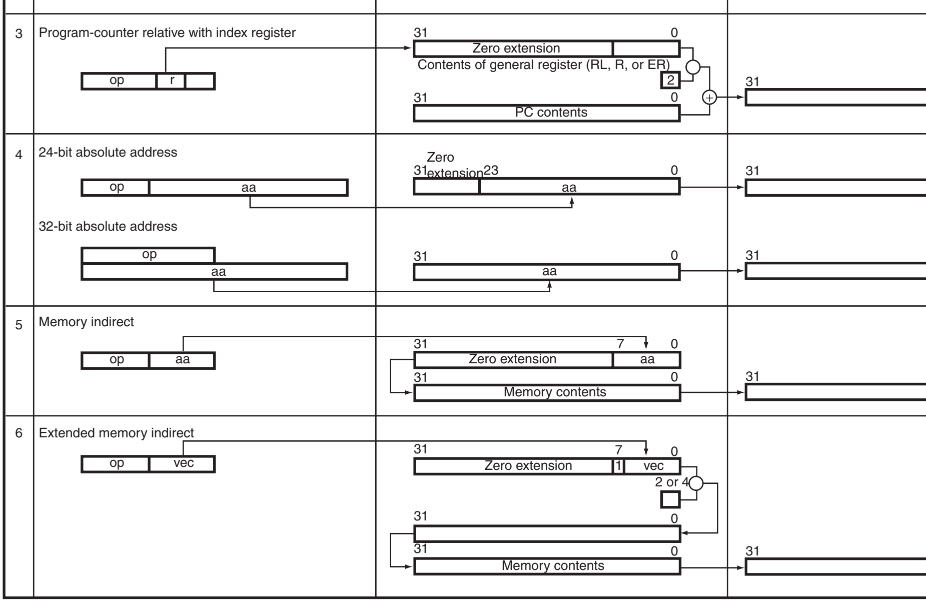
2.8.12 Effective Address Calculation

Tables 2.14 and 2.15 show how effective addresses are calculated in each addressing mo lower bits of the effective address are valid and the upper bits are ignored (zero extende 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-ext the transfer and operation instructions.
- The lower 24 bits of the effective address are valid and the upper eight bits are zero- for the branch instructions.


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



2.8.13 MOVA Instruction

The MOVA Instruction stores the effective address into the general register.

1. Obtains data in the addressing mode of No.2 in table 2.14.
2. By using this data as the index instead of the general register in row No.5 in table 14
   effective address calculation is executed, and the outcome is stored in the general reg

For details, see the H8SX Family Software Manual.

changes from low to high. For details, see 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 n processing flow due to activation of an exception source, such as, a reset, trace, interru trap instruction. The CPU fetches a start address (vector) from the exception handling table and branches to that address. For further details, see section 4, Exception Handli

- 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 req 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 state when a SLEEP instruction is executed or the CPU enters hardware standby mode. For see section 19, Power-Down Modes.

Notes: In any state, when the $\overline{\text{STBY}}$ signal goes low, the hardware standby mode is entered.

* From any state except hardware standby mode, a transition to the reset state occurs whenever the $\overline{\text{RES}}$ 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	Description	On-Chip ROM	Defa
4	1	0	0	Advanced	16 Mbytes	On-chip ROM disabled extended mode	Disabled	16 bi
5	1	0	1				Disabled	8 bits

In this LSI, advanced mode for the CPU operating mode, 16 Mbytes for the address spa eight or 16 bits for the default external bus width are available.

In modes 4 and 5, which are external extended modes, it is possible to access the extern and devices. In external extended mode, the external address space can be designated as 16-bit address space for each area by the bus controller after starting program execution address space is designated for any one area, the bus mode switches to 16 bits. If 8-bit a space is designated for all areas, the bus mode switches to 8 bits.

When MCDR is read, the input levels in pins MD2 to MD 0 are latched. These latches ar by a reset.

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

Note: * Determined by pins MD2 to MD0.

Bit	Bit Name	Initial Value	R/W	Descriptions
15	—	0	R	Reserved These are read-only bits and cannot be modifie
14	—	1	R	
13	—	0	R	
12	—	1	R	
11	—	0	R	
10	MDS2	Undefined*	R	Mode Select 2 to 0 These bits indicate the operating mode selecte mode pins (MD2 to MD0) (see table 3.2). When MDCR is read, the signal levels input on MD2 to MD0 are latched into these bits. These are released by a reset.
9	MDS1	Undefined*	R	
8	MDS0	Undefined*	R	

Note: Determined by pins MD2 to MD0.

Table 3.2 Settings of Bits MSD2 to MSD0

MCU Operating Mode	MD2	MD1	MD0	MDCR		
				MDS2	MDS1	M
4	1	0	0	0	1	0
5	1	0	1	0	0	1

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 value always be 1.
13	MACS	0	R/W	MAC Saturation Operation Control Selects either saturation operation or non-satura 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 shou always be 1.
11	FETCHMD	0	R/W	Instruction Fetch Mode Select The H8SX CPU has two modes for instruction fe bit and 32-bit modes. It is recommended that the should be set according to the bus width of the m which the program is stored*[1]. 0: 32-bit width 1: 16-bit width

				external bus cycle should not be executed. 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 operation 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, see section 2.3, Instruction Fetch.

2. The initial value depends on the startup mode.
In operating modes 4 and 5, which are external extended modes, EXPE = 1.

bus controller, the bus mode switches to 8 bits, and only port H functions as a data bus.

3.3.2 Mode 5

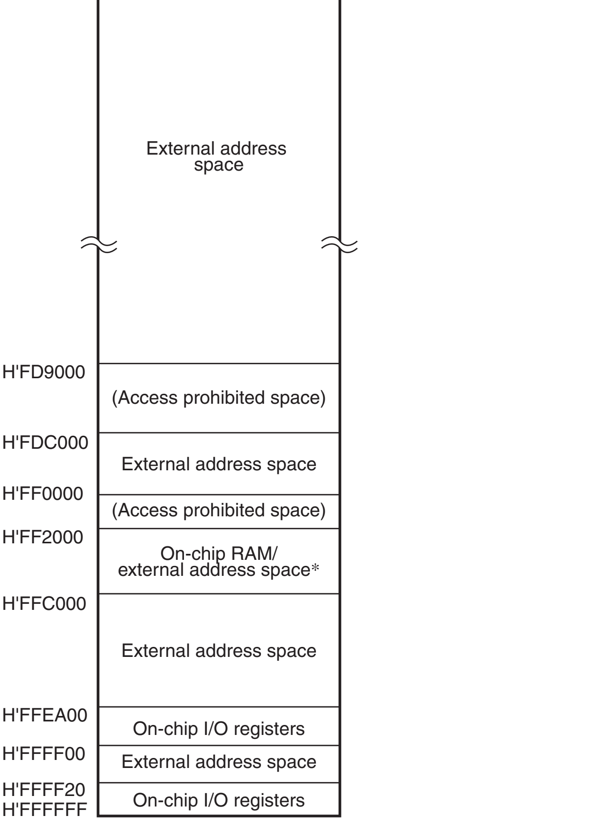
The CPU operating mode is advanced mode in which the address space is 16 Mbytes, and chip ROM is disabled.

The initial bus mode immediately after a reset is 8 bits, with 8-bit access to all areas. Port and F function as an address bus, port H functions as a data bus, and parts of ports A and function as bus control signals. However, if all areas are designated as a 16-bit access spa bus controller, the bus mode switches to 16 bits, and ports H and I function as a data bus.

Port B	PB3 to 1	P*/C	P*/C
	PB0	P/C*	P/C*
Port D		A	A
Port E		A	A
Port F	PF7 to PF5	P*/A	P*/A
	PF4 to PF0	A	A
Port H		D	D
Port I		P/D*	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



Note: * This area is specified as the external address space by clearing the RAME bit in SYSCR to 0.

Figure 3.1 Address Map (Advanced Mode)

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 t 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]
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 are accepted at all times in prog execution state.

section 3, MCU Operating Modes.

Table 4.2 Exception Handling Vector Table

			Vector Table Address Offse	
Exception Source		**Vector Number**	**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 | 17	H'001C to H'001D | H'0022 to H'0023	H'0038 to H'0 | H'0044 to H'0
Sleep interrupt		18	H'0024 to H'0025	H'0048 to H'0

	IRQ2	66	H'0084 to H'0085	H'0108 to H
	IRQ3	67	H'0086 to H'0087	H'010C 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.

A reset has priority over any other exception. When the $\overline{\text{RES}}$ pin goes low, all processing this LSI enters the reset state. To ensure that this LSI is reset, hold the $\overline{\text{RES}}$ pin low for at ms with the $\overline{\text{STBY}}$ pin driven high when the power is turned on. When operation is in pro hold the $\overline{\text{RES}}$ pin low for at least 20 cycles.

The chip can also be reset by overflow of the watchdog timer. For details, see section 13, Watchdog Timer (WDT).

A reset initializes the internal state of the CPU and the registers of the on-chip peripheral The interrupt control mode is 0 immediately after a reset.

4.3.1 Reset Exception Handling

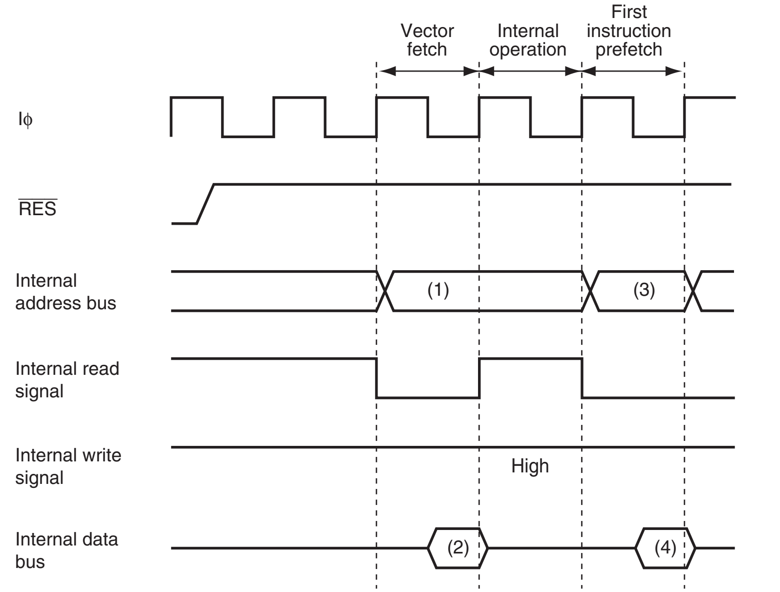
When the $\overline{\text{RES}}$ pin goes high after being held low for the necessary time, this LSI starts r exception handling as follows:

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.

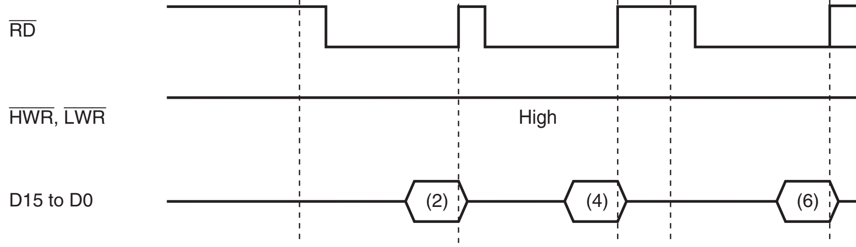
After the reset state is released, MSTPCRA and MSTPCRB are initialized to H'0FFF an
respectively, and all modules except the DTC and DMAC enter module stop mode.

Consequently, on-chip peripheral module registers cannot be read or written to. Register
and writing is enabled when module stop mode is canceled.



(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.

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

Notes: 1. For on-chip peripheral module space, see section 6, Bus Controller (BSC).

2. For the access-prohibited area, see figure 3.1, Address Map (Advanced Mode) 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.

NMI	NMI pin (external input)	1
Sleep interrupt	SLEEP instruction	1
IRQ0 to IRQ11	Pins $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ11}}$ (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)	20

Different vector numbers and vector table offsets are assigned to different interrupt sourc vector number and vector table offset, see table 5.2, Interrupt Sources, Vector Address O 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 inter control modes and can assign interrupts other than NMI or sleep interrupt to eight priority levels to enable multiple-interrupt control. The source to start interrupt exception handlin vector address differ depending on the product. For details, see section 5, Interrupt Contr

4.7.1 Trap Instruction

Trap instruction exception handling starts when a TRAPA instruction is executed. Trap exception handling can be executed at all times in the program execution state. The trap instruction 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 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**	**I2 to I0**	**T**
0	1	—	—	—
2	1	—	—	0

[Legend]

1: Set to 1

0: Cleared to 0

—: Retains the previous value.

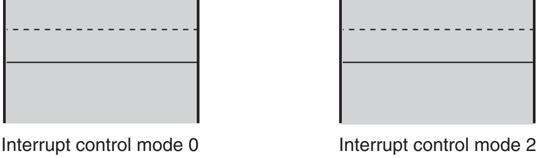
1. The contents of PC, CCR, and EXR are saved in the stack.
2. The interrupt mask bit is updated and the T bit is cleared to 0.
3. An exception handling vector table address corresponding to the occurred exception i generated, the start address of the exception service routine is loaded from the vector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

Table 4.9 shows the state of CCR and EXR after execution of illegal instruction exceptio handling.

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

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

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

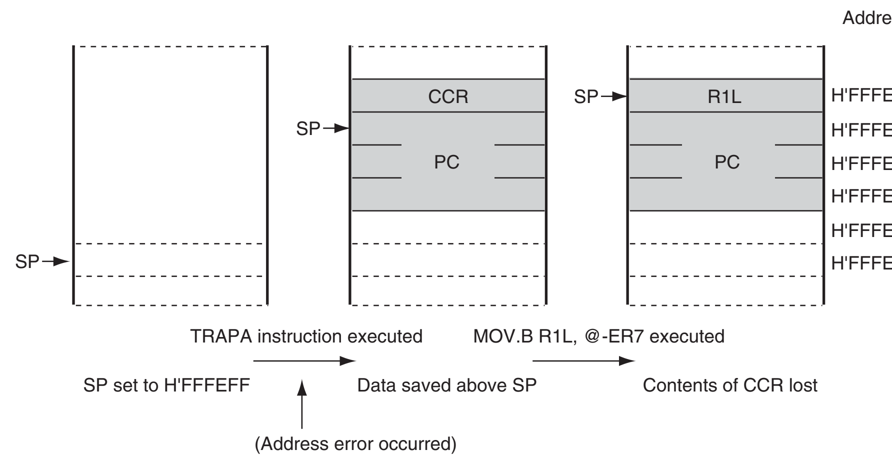


Note: * Ignored on return.

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. F shows an example of operation when the SP value is odd.



[Legend]

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

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

Figure 4.4 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 interrupt

- 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 given
  CPU interrupt exception handling over that of the DTC and DMAC transfer.

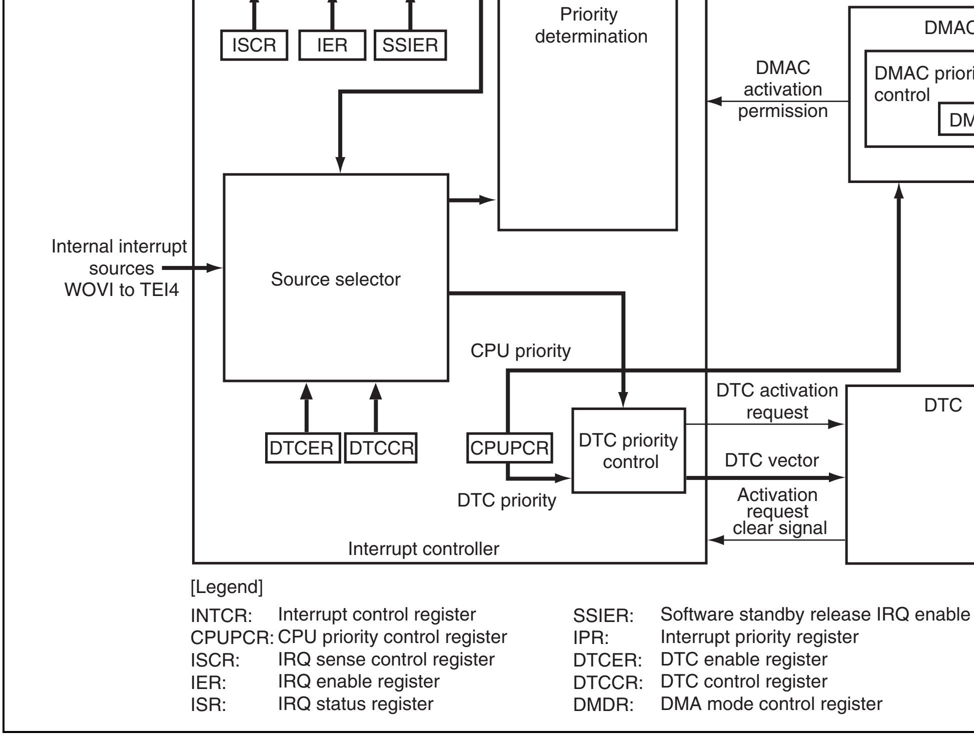

ISCR
IER
SSIER
Priority
determination
DMAC
DMAC activation permission
DMAC priori
control
DM
Internal interrupt sources WOVI to TEI4
Source selector
CPU priority
DTC activation request
DTC
DTCER
DTCCR
CPUPCR
DTC priority control
DTC vector
DTC priority
Activation request clear signal
Interrupt controller
[Legend]
INTCR: Interrupt control register
CPUPCR: CPU priority control register
ISCR: IRQ sense control register
IER: IRQ enable register
ISR: IRQ status register
SSIER: Software standby release IRQ enable
IPR: Interrupt priority register
DTCER: DTC enable register
DTCCR: DTC control register
DMDR: DMA mode control register

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)

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. Set bit to 1 enables the CPU priority control over 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 CPUPCE is set to 1, the CPU priority control fu over the DTC and DMAC becomes valid and th of CPU processing is assigned in accordance w settings of bits CPUP2 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)
0	CPUP0	0	R/(W)*	

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 interru 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
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	IRQ7E	IRQ6E	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E
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 to 12	—	All 0	R/W	Reserved These bits are always read as 0. The write valu always be 0.
11	IRQ11E	0	R/W	IRQ11 Enable The IRQ11 interrupt request is enabled when tl

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

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

				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 $\overline{\text{IRQ11}}$
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 $\overline{\text{IRQ7}}$
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 Ī 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 $\overline{\text{IRQ6}}$
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 Ī 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 $\overline{\text{IRQ5}}$
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 Ī 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 $\overline{\text{IRQ4}}$

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)*

Note: * Only 0 can be written, to clear the flag. The bit manipulation instructions or memory operation instructi 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 i
4	IRQ4F	0	R/(W)*	when falling-, rising-, or both-edge sensin
3	IRQ3F	0	R/(W)*	selected
2	IRQ2F	0	R/(W)*	• When the DTC is activated by an IRQn in
1	IRQ1F	0	R/(W)*	and the DISEL bit in MRB of the DTC is c
0	IRQ0F	0	R/(W)*	

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

Bit	7	6	5	4	3	2	1
Bit Name	SSI7	SSI6	SSI5	SSI4	SSI3	SSI2	SSI1
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 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 These bits select the $\overline{\text{IRQn}}$ pins used to leave s standby mode (n = 11 to 0). 0: IRQn requests are not sampled in software s mode 1: When an IRQn request occurs in software st mode, this LSI leaves software standby moc the oscillation settling time has elapsed
10	SSI10	0	R/W	
9	SSI9	0	R/W	
8	SSI8	0	R/W	
7	SSI7	0	R/W	
6	SSI6	0	R/W	
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 or the settings of the CPU interrupt ma
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 in DTCCR to 1.
- Sets the ERRF bit of DMDR_0 in DMAC to 1.
- The DTE bits of all channels in DMAC are cleared to 0, and transfer is terminated.

(2) IRQn Interrupts

An IRQn interrupt is requested by a signal input on pins $\overline{\text{IRQn}}$ (n = 11 to 0). IRQn intern
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 $\overline{\text{I}}$ 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.
- DTC and DMAC activation can be controlled by the CPU priority control function ov and DMAC.

5.4.3 Sleep Interrupt

A sleep interrupt is generated by executing a SLEEP instruction. The sleep interrupt is no maskable, and is always accepted regardless of the interrupt control mode or the settings CPU interrupt mask bits. The SLPIE bit in SBYCR selects whether the sleep interrupt fun enabled or not.

			Vector Address Offset*			
Classifi-cation	Interrupt Source	Vector Number	Advanced Mode	IPR	Priority	DTC Acti vatio
External pin	NMI	7	H'001C	—	High	—
SLEEP instruc-tion	Sleep interrupt	18	H'0048	—		—
External pin	IRQ0	64	H'0100	IPRA14 to IPRA12		Enab
	IRQ1	65	H'0104	IPRA10 to IPRA8		Enab
	IRQ2	66	H'0108	IPRA6 to IPRA4		Enab
	IRQ3	67	H'010C	IPRA2 to IPRA0		Enab
	IRQ4	68	H'0110	IPRB14 to IPRB12		Enab
	IRQ5	69	H'0114	IPRB10 to IPRB8		Enab
	IRQ6	70	H'0118	IPRB6 to IPRB4		Enab
	IRQ7	71	H'011C	IPRB2 to IPRB0		Enab
	IRQ8	72	H'0120	IPRC14 to IPRC12		Enab
	IRQ9	73	H'0124	IPRC10 to IPRC8		Enab
	IRQ10	74	H'0128	IPRC6 to IPRC4		Enab
	IRQ11	75	H'012C	IPRC2 to IPRC0	Low	Enab

WDT	WOVI	81	H'0144	IPRE10 to IPRE8		—
—	Reserved for system use	82	H'0148	—		—
		83	H'014C			—
		84	H'015C			—
		85	H'0154			—
A/D	ADI	86	H'0158	IPRF10 to IPRF8		Enable
—	Reserved for system use	87	H'015C	—		—
TPU_0	TGI0A	88	H'0160	IPRF6 to IPRF4		Enable
	TGI0B	89	H'0164			Enable
	TGI0C	90	H'0168			Enable
	TGI0D	91	H'016C			Enable
	TCI0V	92	H'0170			—
TPU_1	TGI1A	93	H'0174	IPRF2 to IPRF0		Enable
	TGI1B	94	H'0178			Enable
	TCI1V	95	H'017C			—
	TCI1U	96	H'0180			—
TPU_2	TGI2A	97	H'0184	IPRG14 to IPRG12		Enable
	TGI2B	98	H'0188			Enable
	TCI2V	99	H'018C			—
	TCI2U	100	H'0190		Low	—

TPU_4	TGI4A	106	H'01A8	IPRG6 to IPRG4		Enab
	TGI4B	107	H'01AC			Enab
	TCI4V	108	H'01B0			—
	TCI4U	109	H'01B4			—
TPU_5	TGI5A	110	H'01B8	IPRG2 to IPRG0		Enab
	TGI5B	111	H'01BC			Enab
	TCI5V	112	H'01C0			—
	TCI5U	113	H'01C4			—
—	Reserved for system use	114	H'01C8	—		—
		115	H'01CC			—
TMR_0	CMI0A	116	H'01D0	IPRH14 to IPRH12		Enab
	CMI0B	117	H'01D4			Enab
	OV0I	118	H'01D8			—
TMR_1	CMI1A	119	H'01DC	IPRH10 to IPRH8		Enab
	CMI1B	120	H'01E0			Enab
	OV1I	121	H'01E4			—
TMR_2	CMI2A	122	H'01E8	IPRH6 to IPRH4		Enab
	CMI2B	123	H'01EC			Enab
	OV2I	124	H'01F0			—
TMR_3	CMI3A	125	H'01F4	IPRH2 to IPRH0		Enab
	CMI3B	126	H'01F8			Enab
	OV3I	127	H'01FC		Low	—

		133	H'0214			—
		134	H'0218			—
		135	H'021C			—
DMAC	DMEEND0	136	H'0220	IPRK14 to IPRK12		Enable
	DMEEND1	137	H'0224			Enable
	DMEEND2	138	H'0228			Enable
	DMEEND3	139	H'022C			Enable
—	Reserved for system use	140	H'0230	—		—
		141	H'0234			—
		142	H'0238			—
		143	H'023C			—
SCI_0	ERI0	144	H'0240	IPRK6 to IPRK4		—
	RXI0	145	H'0244			Enable
	TXI0	146	H'0248			Enable
	TEI0	147	H'024C			—
SCI_1	ERI1	148	H'0250	IPRK2 to IPRK0		—
	RXI1	149	H'0254			Enable
	TXI1	150	H'0258			Enable
	TEI1	151	H'025C		Low	—

	RXI3	157	H'0274			Enab
	TXI3	158	H'0278			Enab
	TEI3	159	H'027C			—
SCI_4	ERI4	160	H'0280	IPRL6 to IPRL4		—
	RXI4	161	H'0284			Enab
	TXI4	162	H'0288			Enab
	TEI4	163	H'028C			—
—	Reserved for system use	164	H'0290	—		—
		|	|			|
		255	H'03FC		Low	—

Note: * Lower 16 bits of the start address.

0	Default	I	The priority levels of the interrupt sour fixed default settings. The interrupts except for NMI and slee interrupt is masked by the I bit.
2	IPR	I2 to I0	Eight priority levels can be set for inter sources except for NMI and sleep inte IPR. 8-level interrupt mask control is perfor bits I2 to I0.

5.6.1 Interrupt Control Mode 0

In interrupt control mode 0, interrupt requests except for NMI and sleep interrupt are mas the I bit in CCR of the CPU. Figure 5.3 shows a flowchart of the interrupt acceptance ope this case.

1. If an interrupt request occurs when the corresponding interrupt enable bit is set to 1, th interrupt request is sent to the interrupt controller.
2. If the I bit in CCR is set to 1, NMI and sleep interrupt is accepted, and other interrupt 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 wi highest priority, sends the request to the CPU, and holds other interrupt requests pend
4. When the CPU accepts the interrupt request, it starts interrupt exception handling afte 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 are 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 and sleep inter

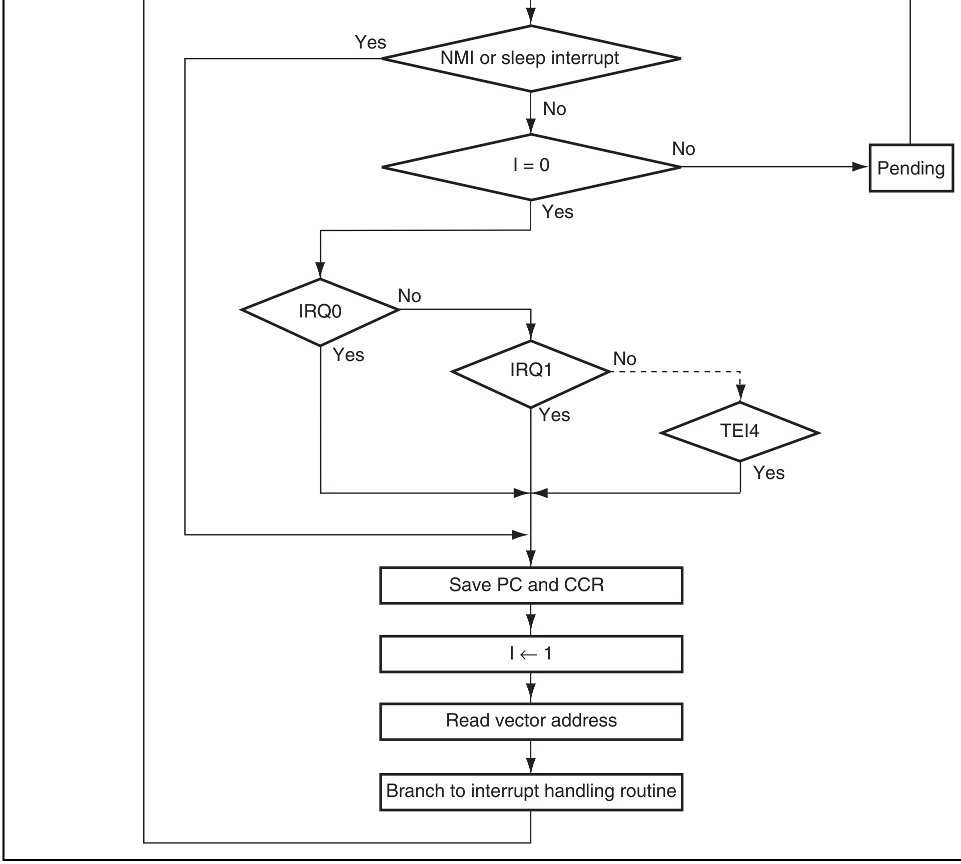


Figure 5.3 Flowchart of Procedure Up to Interrupt Acceptance in Interrupt Control Mode 0

highest priority according to the IPR setting, and holds other interrupt requests pendin multiple interrupt requests have 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 mask in EXR. When the interrupt request does not have priority over the mask level set, it i pending, and only an interrupt request with a priority over the interrupt mask level is a
4. When the CPU accepts an interrupt request, it starts interrupt exception handling after execution of the current instruction has been completed.
5. The PC, CCR, and EXR contents are saved to the stack area during interrupt exceptio handling. The PC saved on the stack is the address of the first instruction to be execut returning from the interrupt handling routine.
6. The T bit in EXR is cleared to 0. The interrupt mask level is rewritten with the priority accepted interrupt. If the accepted interrupt is NMI or sleep interrupt, the interrupt ma is set to H'7.
7. The CPU generates a vector address for the accepted interrupt and starts execution of interrupt handling routine at the address indicated by the contents of the vector addres vector table.

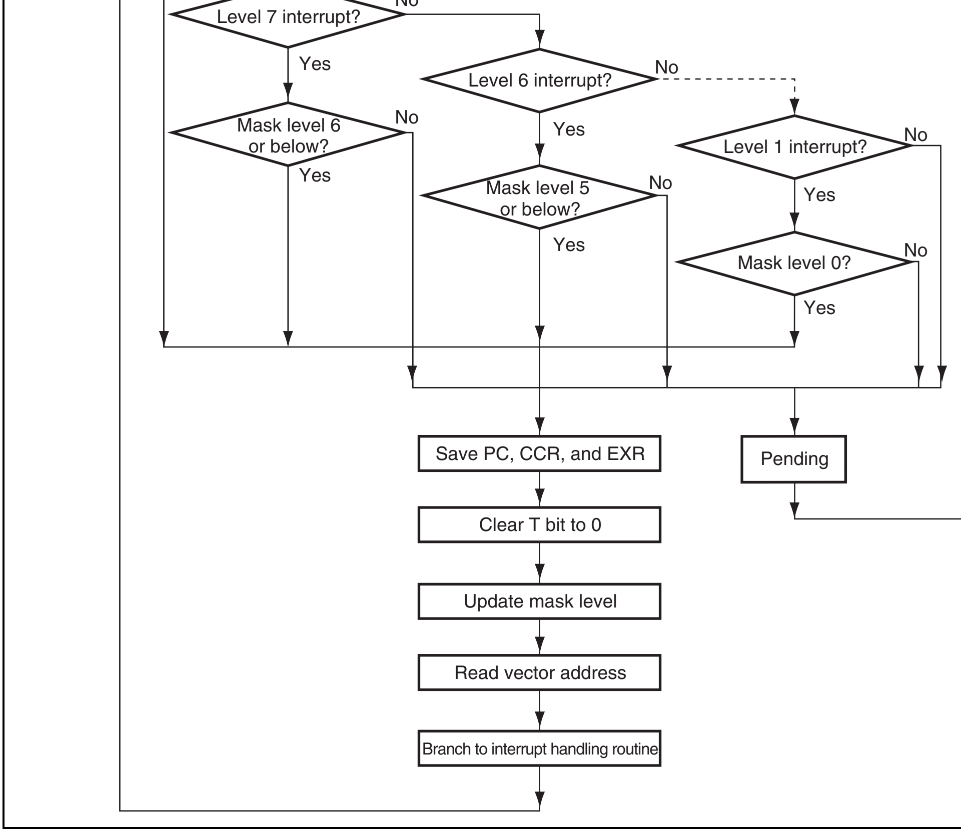
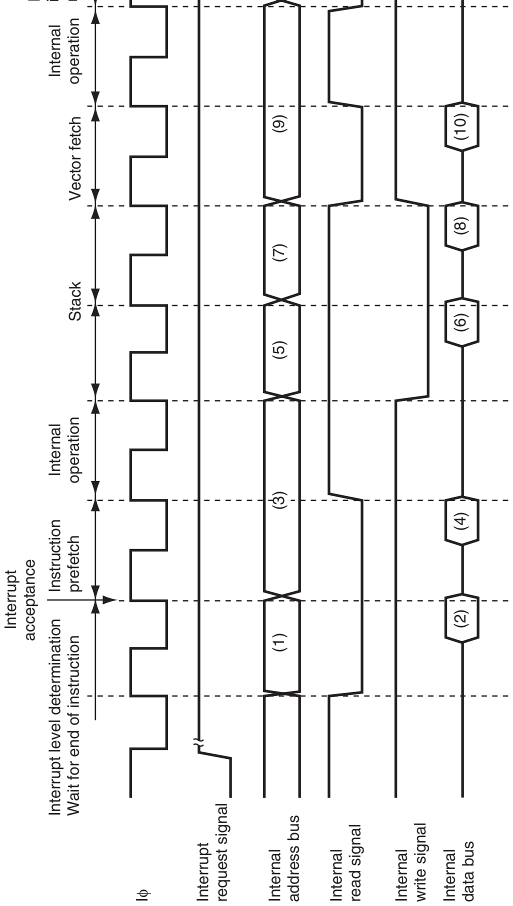


Figure 5.4 Flowchart of Procedure Up to Interrupt Acceptance in Interrupt Control Mode 2



(1) Instruction prefetch address (Not executed. This is the contents of the saved PC, the return address.)
(2) (4) Instruction code (Not executed.)
(3) Instruction prefetch address (Not executed.)
(5) SP – 2
(7) SP – 4

(6) (8) Saved PC and saved CCR
(9) Vector address
(10) Start address of interrupt handling routine (vector add
(11) Start address of Interrupt handling routine ((11) = (10)
(12) First instruction of interrupt handling routine

Figure 5.5 Interrupt Exception Handling

Execution State	Interrupt Control Mode 0	Interrupt Control Mode 2	Interrupt Control Mode 0	Interrupt Control Mode 2	Interrupt Control Mode 0
Interrupt priority determination*1	3				
Number of states until executing instruction ends*2	1 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	$2 \cdot S_K$	$2 \cdot S_K$
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

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 handlin

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 setti 4n + 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 options 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 tab section 7, DMA Controller (DMAC), and section 8, Data Transfer Controller (DTC).

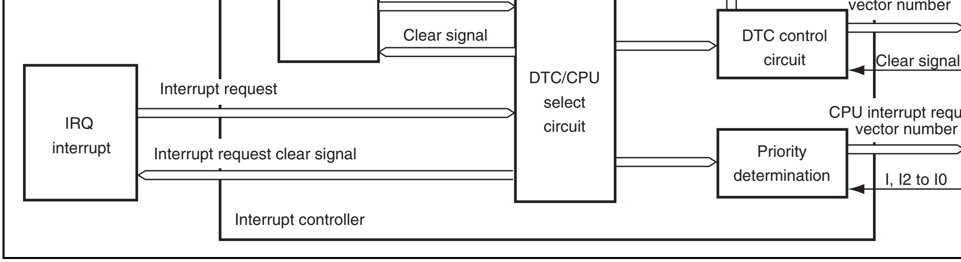


Figure 5.6 Block Diagram of DTC, DMAC, and Interrupt Controller

(1) Selection of Interrupt Sources

The activation source for each DMAC channel is selected by DMRSR. The selected acti source is input to the DMAC through the select circuit. When transfer by an on-chip mo interrupt is enabled (DTF1 = 1, DTF0 = 0, and DTE = 1 in DMDR) and the DTA bit in set to 1, the interrupt source selected for the DMAC activation source is controlled by th 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 source 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 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 tran counter indicates 0, an interrupt request is made to the CPU by clearing the DTCE bit to DTC data transfer.

Sources, DTC Vector Addresses, and Corresponding DTCEs.

(3) Operation Order

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

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

Table 5.6 Interrupt Source Selection and Clear Control

DMAC Setting	DTC Setting		Interrupt Source Selection/Clear C		
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 prio can automatically be assigned to the CPU on an interrupt occurrence, it is possible to ex 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 pri of the DTC is assigned by bits DTCP2 to DTCP0 in CPUPCR. The priority level of the 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 according to the respective priority levels.

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

For the DMAC, the priority level can be specified for each channel. The DMAC activati 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 cancelled (CPUPCE = 1 and value of bits CPUP2 to CPUP0 is greater than that of bits D DMAP0). If different priority levels are specified for channels, the channels of the highe levels continue transfer and the activation sources for the channels of lower priority leve 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	Rewriting 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*

Note: * The CPU priority is automatically updated.

		B'100	B'000	B'000	Masked	Mas
		B'100	B'000	B'011	Masked	Mas
		B'100	B'111	B'101	Enabled	Enal
		B'000	B'111	B'101	Enabled	Enal
2	0	Any	Any	Any	Enabled	Enal
	1	B'000	B'000	B'000	Enabled	Enal
		B'000	B'011	B'101	Enabled	Enal
		B'011	B'011	B'101	Enabled	Enal
		B'100	B'011	B'101	Masked	Enal
		B'101	B'011	B'101	Masked	Enal
		B'110	B'011	B'101	Masked	Mas
		B'111	B'011	B'101	Masked	Mas
		B'101	B'011	B'101	Masked	Enal
		B'101	B'110	B'101	Enabled	Enal

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

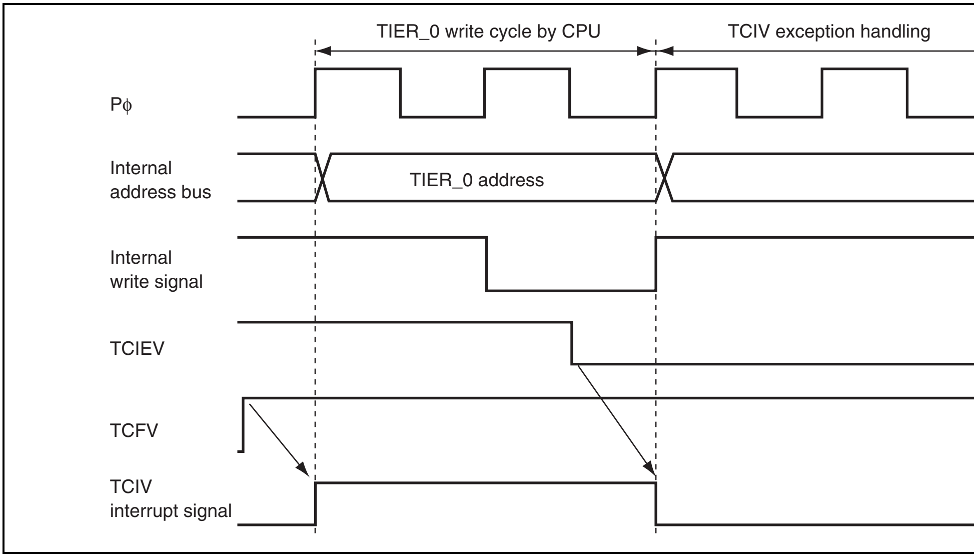


Figure 5.7 Conflict between Interrupt Generation and Disabling

If an interrupt is generated immediately before rewriting the DTC enable bit, both DTC a
and CPU interrupt exception handling are executed. To rewrite the DTC enable bit, execu
while the corresponding interrupt request is not generated.

The interrupt controller disables interrupt acceptance for a 3-state period after the CPU l
updated the mask level with an LDC, ANDC, ORC, or XORC instruction, and for a peri
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, in
exception handling starts at the end of the individual transfer cycle. The PC value saved
stack in this case is the address of the next instruction. Therefore, if an interrupt is gener
during execution of an EEPMOV.W instruction, the following coding should be used.

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

5.8.5 Interrupts during Execution of MOVMD and MOVSD Instructions

With the MOVMD or MOVSD instruction, if an interrupt request is issued during the tr
interrupt exception handling starts at the end of the individual transfer cycle. The PC val
on the stack in this case is the address of the MOVMD or MOVSD instruction. The tran
remaining data is resumed after returning from the interrupt handling routine.

- Manages external address space in area units

  Manages the external address space divided into eight areas.

  Chip select signals ($\overline{\text{CS0}}$ to $\overline{\text{CS7}}$) 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 $\overline{\text{WAIT}}$ pin.

  Extension cycles can be inserted while $\overline{\text{CSn}}$ is asserted for each area (n = 0 to 7).

  The negation timing of the read strobe signal ($\overline{\text{RD}}$) 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, DTC, an external bus master
- Multi-clock function

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

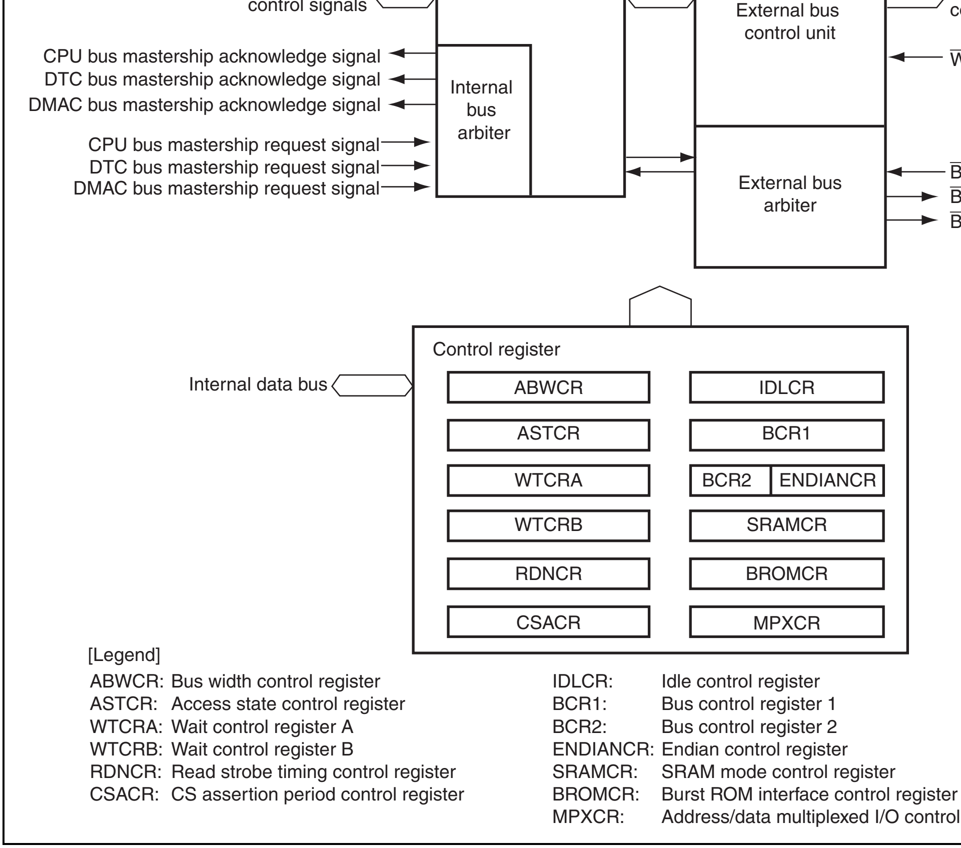

control signals
CPU bus mastership acknowledge signal
DTC bus mastership acknowledge signal
DMAC bus mastership acknowledge signal
CPU bus mastership request signal
DTC bus mastership request signal
DMAC bus mastership request signal
Internal bus arbiter
External bus control unit
External bus arbiter
Internal data bus
Control register
ABWCR
IDLCR
ASTCR
BCR1
WTCRA
BCR2
ENDIANCR
WTCRB
SRAMCR
RDNCR
BROMCR
CSACR
MPXCR
[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)

Initial value								
R/W	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 These bits select whether the corresponding ar designated as 8-bit access space or 16-bit acce ABWHn ABWLn (n = 7 to 0) × 0: Setting prohibited 0 1: Area n is designated as 16- space 1 1: Area n is designated as 8-b space*[2]
14	ABWH6	1	R/W	
13	ABWH5	1	R/W	
12	ABWH4	1	R/W	
11	ABWH3	1	R/W	
10	ABWH2	1	R/W	
9	ABWH1	1	R/W	
8	ABWL0	1/0	R/W	
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 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 designated as 2-state access space or 3-state ac space. Wait cycle insertion is enabled or disable same time.
13	AST5	1	R/W	
12	AST4	1	R/W	
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 disable
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	W62	1	R/W	Area 6 Wait Control 2 to 0
9	W61	1	R/W	These bits select the number of program wait cy when accessing area 6 while bit AST6 in ASTCR
8	W60	1	R/W	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	W42	1	R/W	Area 4 Wait Control 2 to 0
1	W41	1	R/W	These bits select the number of program wait c when accessing area 4 while bit AST4 in ASTC
0	W40	1	R/W	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 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
8	W20	1	R/W	
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	W02 W01 W00	1 1 1	R/W R/W R/W	Area 0 Wait Control 2 to 0 These bits select the number of program wait c 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 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 corresponding area read access.
13	RDN5	0	R/W	
12	RDN4	0	R/W	As shown in figure 6.2, the read strobe for an are which the RDNn bit is set to 1 is negated one ha earlier than that for an area for which the RDNn cleared to 0. The read data setup and hold time given one half-cycle earlier.
11	RDN3	0	R/W	
10	RDN2	0	R/W	
9	RDN1	0	R/W	
8	RDN0	0	R/W	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 and the same operation when RDNn = 0 is performed.

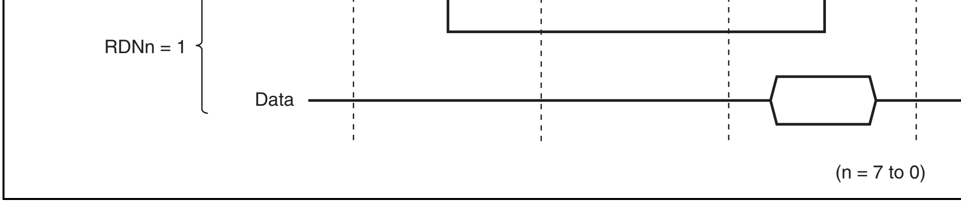


Figure 6.2 Read Strobe Negation Timing (Example of 3-State Access Spac

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

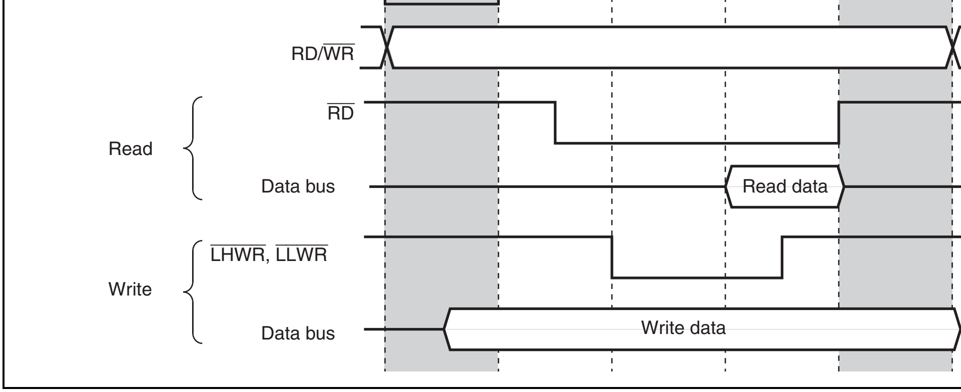


Figure 6.3 $\overline{\text{CS}}$ and Address Assertion Period Extension (Example of Basic Bus Interface, 3-State Access Space, and RDNn = 0)

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	IDLS3	1	R/W	Idle Cycle Insertion 3 Inserts an idle cycle between the bus cycles whe DMAC single address transfer (write cycle) is fol external access. 0: No idle cycle is inserted 1: An idle cycle is inserted
14	IDLS2	1	R/W	Idle Cycle Insertion 2 Inserts an idle cycle between the bus cycles whe external write cycle is followed by external read c 0: No idle cycle is inserted 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

8	IDLCA0	1	R/W	Specifies the number of idle cycles to be inserte 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
7	IDLSEL7	0	R/W	Idle Cycle Number Select Specifies the number of idle cycles to be inserte each area for the idle insertion condition specifi IDLS1 and IDLS0. 0: Number of idle cycles to be inserted for area specified by IDLCA1 and IDLCA0. 1: Number of idle cycles to be inserted for area specified by IDLCB1 and IDLCB0. (n = 7 to 0)
6	IDLSEL6	0	R/W	
5	IDLSEL5	0	R/W	
4	IDLSEL4	0	R/W	
3	IDLSEL3	0	R/W	
2	IDLSEL2	0	R/W	
1	IDLSEL1	0	R/W	
0	IDLSEL0	0	R/W	

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 used ports 1: External bus release enabled* To set this bit to 1, the ICR bit of the correspondi should be specified to 1. For details, see section Ports.
14	BREQOE	0	R/W	$\overline{\text{BREQO}}$ Pin Enable Controls outputting the bus request signal ($\overline{\text{BREQ}}$ the external bus master in the external bus relea when an internal bus master performs an externa address space access. 0: $\overline{\text{BREQO}}$ output disabled $\overline{\text{BREQO}}$ pin can be used as I/O port 1: $\overline{\text{BREQO}}$ output enabled
13, 12	—	All 0	R	Reserved These are read-only bits and cannot be modified

				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 To set this bit to 1, the ICR bit of the correspon should be specified to 1. For details, see sectio Ports.
7	DKC	0	R/W	$\overline{\text{DACK}}$ Control Selects the timing of DMAC transfer acknowled assertion. 0: $\overline{\text{DACK}}$ signal is asserted at the Bϕ falling edg 1: $\overline{\text{DACK}}$ signal is asserted at the Bϕ rising edg
6	—	0	R/W	Reserved This bit is always read as 0. The write value sh always be 0.
5 to 0	—	All 0	R	Reserved These are read-only bits and cannot be modifie

7, 6	—	All 0	R	Reserved These are read-only bits and cannot be modified
5	—	0	R/W	Reserved This bit is always read as 0. The write value shou 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 the 1: Executes the bus cycles alternatively when a mastership request conflicts with a DMAC or D mastership request
3, 2	—	All 0	R	Reserved These are read-only bits and cannot be modified
1	—	1	R/W	Reserved This bit is always read as 1. The write value shou always be 1.
0	PWDBE	0	R/W	Peripheral Module Write Data Buffer Enable Specifies whether or not to use the write data bu 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 endi
4	LE4	0	R/W	1: Data format of area n is specified as little enc
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 modifie

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 a
13	BCSEL5	0	R/W	When setting the area n bit to 1, the bus interfac selection bits for the corresponding area in BRO MPXCR should be cleared to 0.
12	BCSEL4	0	R/W	
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 modified

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 Selects the area 0 bus interface. When setting t 1, clear the BCSEL0 bit 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 modifie

				Selects the area 1 bus interface. When setting th 1, clear the BCSEL1 bit in SRAMCR to 0. 0: Basic bus interface or byte control SRAM inte 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 cy
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 modified
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 t 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	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 correspondin
13	MPXE5	0	R/W	When setting the area n bit to 1, clear the BCS SRAMCR to 0.
12	MPXE4	0	R/W	0: Area n is specified as a basic interface or a b control SRAM interface.
11	MPXE3	0	R/W	1: Area n is specified as an address/data multip interface (n = 7 to 3)
10 to 1	—	All 0	R	Reserved These are read-only bits and cannot be modifie
0	ADDEX	0	R/W	Address Output Cycle Extension Specifies whether a wait cycle is inserted for th output cycle of address/data multiplexed I/O int 0: No wait cycle is inserted for the address outp 1: One wait cycle is inserted for the address ou

registers of peripheral modules such as SCI and timer.

- External access cycle

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

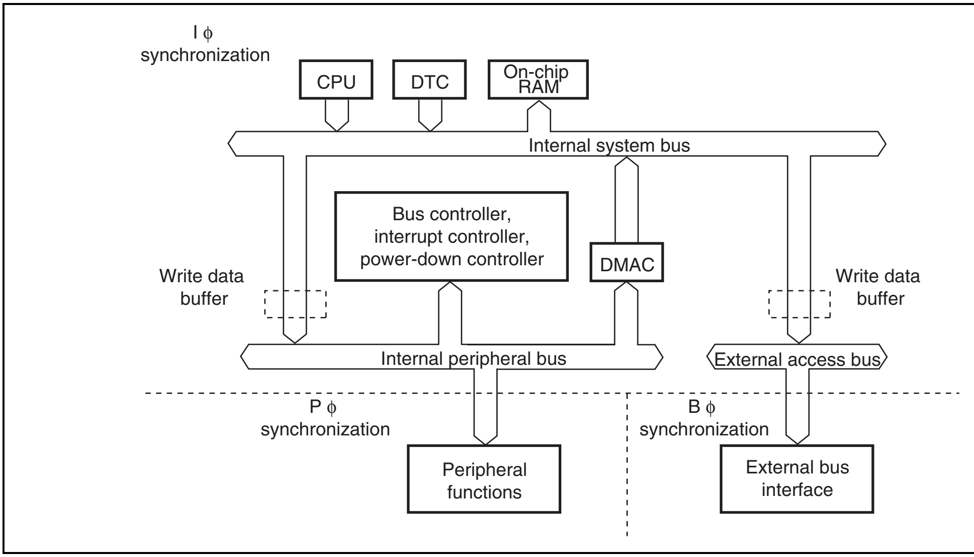


Figure 6.4 Internal Bus Configuration

	Bus controller CPU DMAC DTC 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

The frequency of each synchronization clock (Iϕ, Pϕ, and Bϕ) is specified by the system control register (SCKCR) independently. For further details, see section 18, Clock Pulse Generator.

There will be cases when Pϕ and Bϕ are equal to Iϕ and when Pϕ and Bϕ are different fr according to the SCKCR specifications. In any case, access cycles for internal periphera and external space is performed synchronously with Pϕ and Bϕ, respectively.

For example, in an external address access where the frequency rate of Iϕ and Bϕ is n : 1 operation is performed in synchronization with Bϕ. In this case, external 2-state access s cycles and external 3-state access space is 3n cycles (no wait cycles is inserted) if the nu access cycles is counted based on Iϕ.

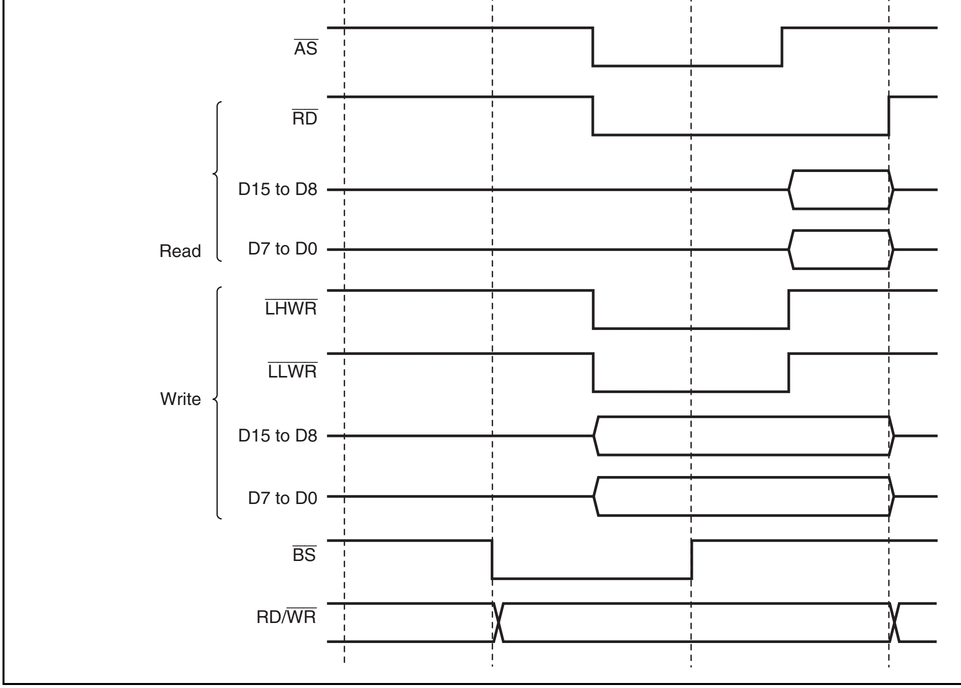


Figure 6.5 System Clock: External Bus Clock = 4:1, External 2-State Acce

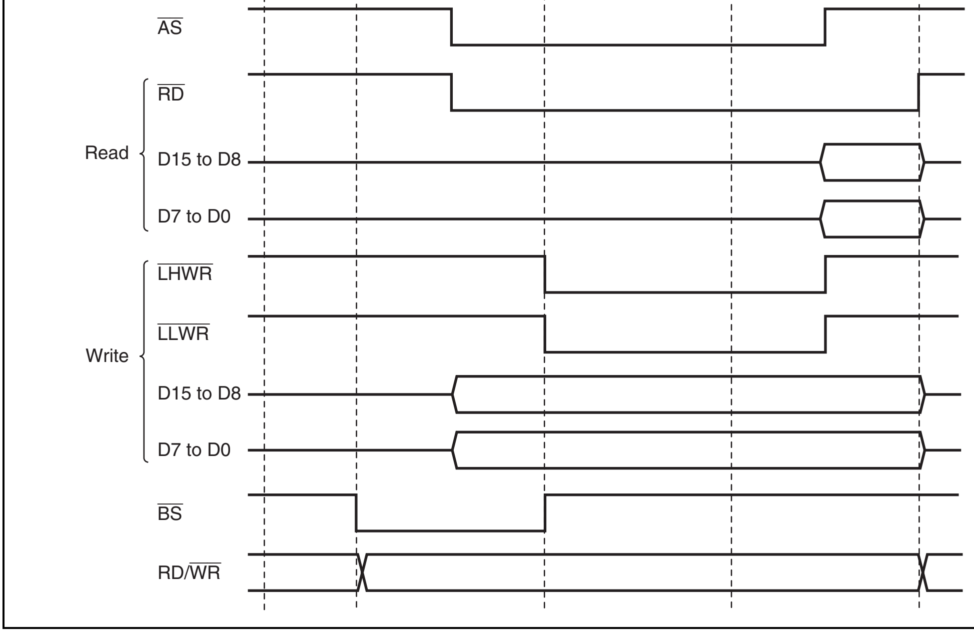


Figure 6.6 System Clock: External Bus Clock = 2:1, External 3-State Acces

Bus cycle start	$\overline{\text{BS}}$	Output	Signal indicating that the bus cyc started
Address strobe/address hold	$\overline{\text{AS}}$/$\overline{\text{AH}}$	Output	• Strobe signal indicating that t bus, byte control SRAM, or b space is accessed and addre on address bus is enabled • Signal to hold the address du access to the address/data m I/O interface
Read strobe	$\overline{\text{RD}}$	Output	Strobe signal indicating that the byte control SRAM, burst ROM, address/data multiplexed I/O spa being read
Read/write	RD/$\overline{\text{WR}}$	Output	• Signal indicating the input or direction • Write enable signal of the SR access to the byte control SR
Low-high write/lower-upper byte select	$\overline{\text{LHWR}}$/$\overline{\text{LUB}}$	Output	• Strobe signal indicating that t bus, burst ROM, or address/ multiplexed I/O space is writt the upper byte (D15 to D8) of is enabled • Strobe signal indicating that t control SRAM space is acces the upper byte (D15 to D8) of is enabled

Chip select 0	CS0	Output	Strobe signal indicating that area selected
Chip select 1	$\overline{\text{CS1}}$	Output	Strobe signal indicating that area selected
Chip select 2	$\overline{\text{CS2}}$	Output	Strobe signal indicating that area selected
Chip select 3	$\overline{\text{CS3}}$	Output	Strobe signal indicating that area selected
Chip select 4	$\overline{\text{CS4}}$	Output	Strobe signal indicating that area selected
Chip select 5	$\overline{\text{CS5}}$	Output	Strobe signal indicating that area selected
Chip select 6	$\overline{\text{CS6}}$	Output	Strobe signal indicating that area selected
Chip select 7	$\overline{\text{CS7}}$	Output	Strobe signal indicating that area selected
Wait	$\overline{\text{WAIT}}$	Input	Wait request signal when accessi external address space.
Bus request	$\overline{\text{BREQ}}$	Input	Request signal for release of bus external bus master
Bus request acknowledge	$\overline{\text{BACK}}$	Output	Acknowledge signal indicating tha been released to external bus ma
Bus request output	$\overline{\text{BREQO}}$	Output	External bus request signal used internal bus master accesses exte address space in the external-bus state
Data transfer acknowledge 3 (DMAC_3)	$\overline{\text{DACK3}}$	Output	Data transfer acknowledge signal DMAC_3 single address transfer

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

[Legend]

O: Used as a bus control signal

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

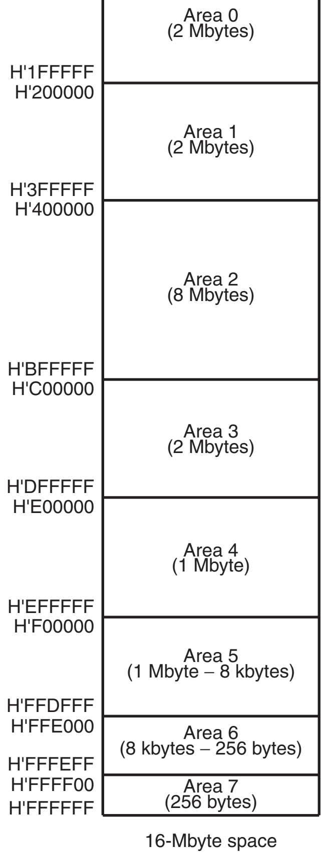


Figure 6.7 Address Space Area Division

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

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

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

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

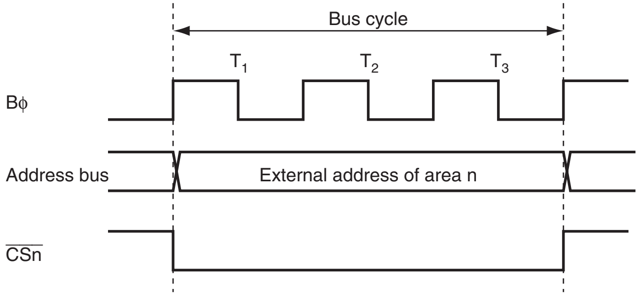


Figure 6.8 $\overline{\text{CSn}}$ Signal Output Timing (n = 0 to 7)

Figure 6.9 Timing When $\overline{\text{CS}}$ 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 cycle strobe assert/negate timings can be set for each area in the external address space. The bu and the number of access cycles for both on-chip memory and internal I/O registers are fi are not affected by the external bus settings.

(1) Type of External Bus Interface

Four types of external bus interfaces are provided and can be selected in area units. Table shows each interface name, description, and area name to be set for each interface. Table shows the areas that can be specified for each interface. The initial state of each area is a 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 sp
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 multiple space

(2) Bus Width

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

(3) Endian Format

Though the endian format of this LSI is big endian, data can be converted into little endi 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 L 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 b

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 $\overline{\text{CS}}$ extension cycles (0, 1, 2)
[+ number of external wait cycles by the $\overline{\text{WAIT}}$ pin]

2. Byte Control SRAM Interface

   The number of access cycles in the byte control SRAM interface is the same as that ir basic 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 $\overline{\text{CS}}$ extension cycles (0, 1, 2)
   [+ number of external wait cycles by the $\overline{\text{WAIT}}$ pin]

3. Burst ROM Interface

   The number of access cycles at full access in the burst ROM interface is the same as t basic bus interface. The number of access cycles in the burst access can be specified a 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 $\overline{\text{CS}}$ extension cycles (0, 1)
   [+number of external wait cycles by the $\overline{\text{WAIT}}$ 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	+T1	+T2				+Tt		
		[0,1]	[1]	[1]				[0,1]		
	=	Th	+T1	+T2	+Tpw	+Ttw	+T3	+Tt		
		[0,1]	[1]	[1]	[0 to 7]	[n]	[1]	[0,1]		
Byte control SRAM interface	=	Th	+T1	+T2				+Tt		
		[0,1]	[1]	[1]				[0,1]		
	=	Th	+T1	+T2	+Tpw	+Ttw	+T3	+Tt		
		[0,1]	[1]	[1]	[0 to 7]	[n]	[1]	[0,1]		
Burst ROM interface	=	Th	+T1	+T2					+Tb	
		[0,1]	[1]	[1]					[(1 to 8) × m]	[(2 to
	=	Th	+T1	+T2	+Tpw	+Ttw	+T3		+Tb	
		[0,1]	[1]	[1]	[0 to 7]	[n]	[1]		[(1 to 8) × m]	[(3 to 11 +
Address/data multiplexed I/O	= Tma	+Th	+T1	+T2				+Tt		
interface	[2,3]	[0,1]	[1]	[1]				[0,1]		
	= Tma	+Th	+T1	+T2	+Tpw	+Ttw	+T3	+Tt		
	[2,3]	[0,1]	[1]	[1]	[0 to 7]	[n]	[1]	[0,1]		

[Legend]

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

(5) Strobe Assert/Negate Timings

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

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

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 enal extended mode, the space excluding on-chip ROM* is external address space.

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

Either of the basic bus interface, byte control SRAM, or burst ROM interface can be selec area 1 by bit BSRM1 in BROMCR and bit BCSEL1 in SRAMCR. Table 6.8 shows the ex 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 $\overline{\text{CS2}}$ signal can be output.

Either the basic bus interface or byte control SRAM interface can be selected for area 2 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 $\overline{\text{CS3}}$ signal can be output.

Either of the basic bus interface, byte control SRAM interface, or address/data multiplex 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 $\overline{\text{CS4}}$ signal can be output.

Either of the basic bus interface, byte control SRAM interface, or address/data multiplexe 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 i 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 SRAMC**
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-c register area is external address space.

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

Either of the basic bus interface, byte control SRAM interface, or address/data multiple interface can be selected for area 6 by the MPXE6 bit in MPXCR and the BCSEL6 bit i 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 intern register area is external address space.

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

Either of the basic bus interface, byte control SRAM interface, or address/data multiplexe interface can be selected for area 7 by the MPXE7 bit in MPXCR and the BCSEL7 bit in 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 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 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.

Data Size	Access Address	Access Count	Bus Cycle	Data Size	Strobe s LHWR/LUB RD Data b D15 D8 D
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 (Bi

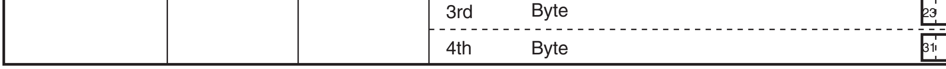


Figure 6.11 Access Sizes and Data Alignment Control for 8-Bit Access Spac
(Little Endian)

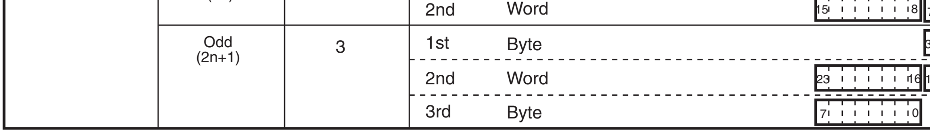
(2) 16-Bit Access Space

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

Figures 6.12 and 6.13 illustrate data alignment control for the 16-bit access space. Figure
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.

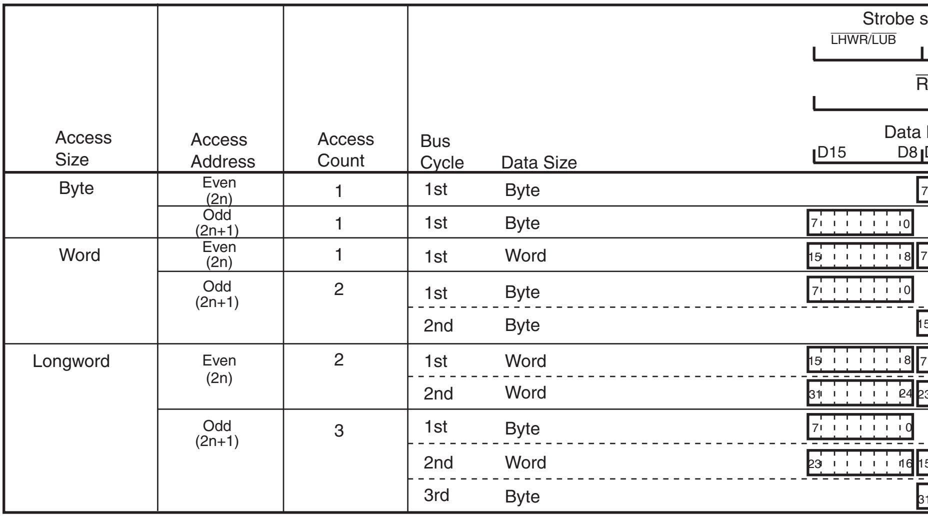
In big endian, byte access for an even address is performed by using the upper byte data b
byte access for an odd address is performed by using the lower byte data bus.

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



Access Size	Access Address	Access Count	Bus Cycle	Data Size	D15 D8	
	(2n)		2nd	Word	15 8	7
	Odd (2n+1)	3	1st	Byte		3
			2nd	Word	23 16	1
			3rd	Byte	7 0	

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



Access Size	Access Address	Access Count	Bus Cycle	Data Size	Strobe s LHWR/LUB Data D15 D8	R 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 Sp
(Little Endian)

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

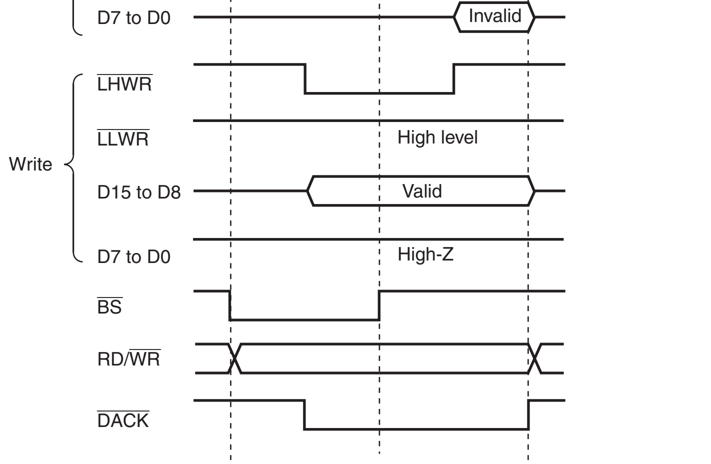
6.6.2 I/O Pins Used for Basic Bus Interface

Table 6.15 shows the pins used for basic bus interface.

Table 6.15 I/O Pins for Basic Bus Interface

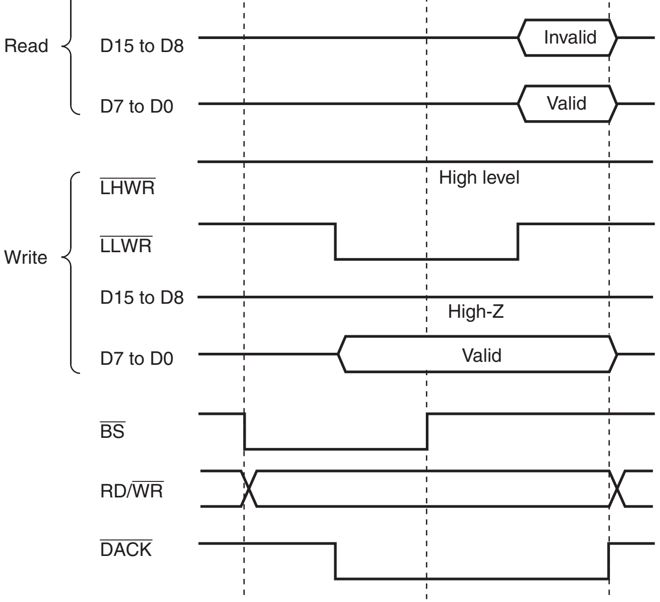
Name	Symbol	I/O	Function
Bus cycle start	$\overline{BS}$	Output	Signal indicating that the bus cycle has start
Address strobe	$\overline{AS}$*	Output	Strobe signal indicating that an address outp address bus is valid during access
Read strobe	$\overline{RD}$	Output	Strobe signal indicating the read access
Read/write	RD/$\overline{WR}$	Output	Signal indicating the data bus input or outpu 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 sele
Wait	$\overline{WAIT}$	Input	Wait request signal used when an external a space is accessed

Note: * When the address/data multiplexed interface is selected, this pin only functions $\overline{AH}$ output and does not function as the $\overline{AS}$ output.



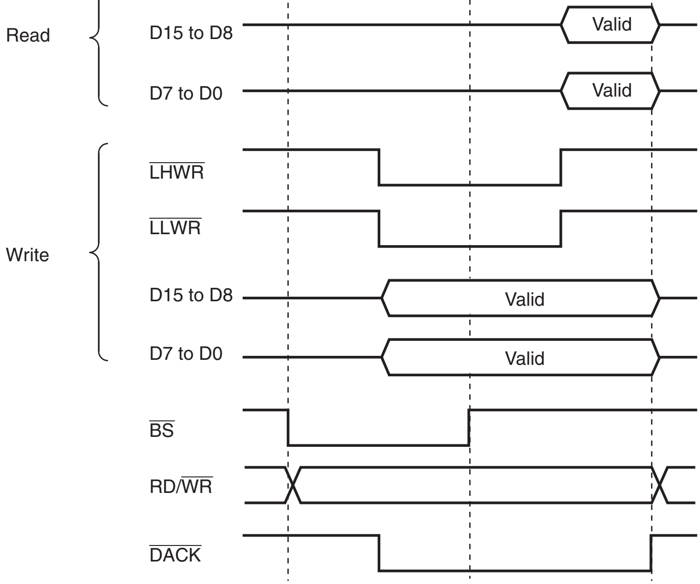
Notes: 1. n = 0 to 7
2. When RDNn = 0
3. When DKC = 0

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



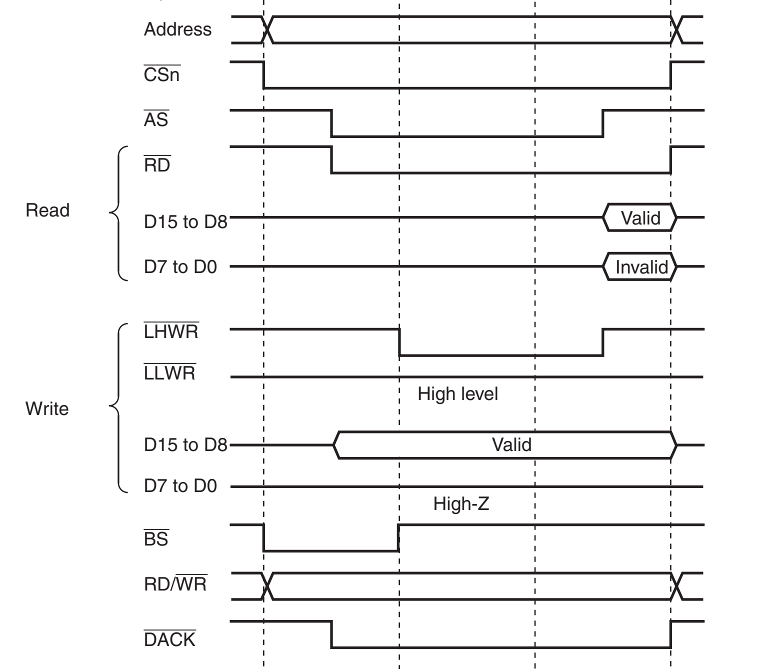
Notes: 1. n = 0 to 7
2. When RDNn = 0
3. When DKC = 0

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



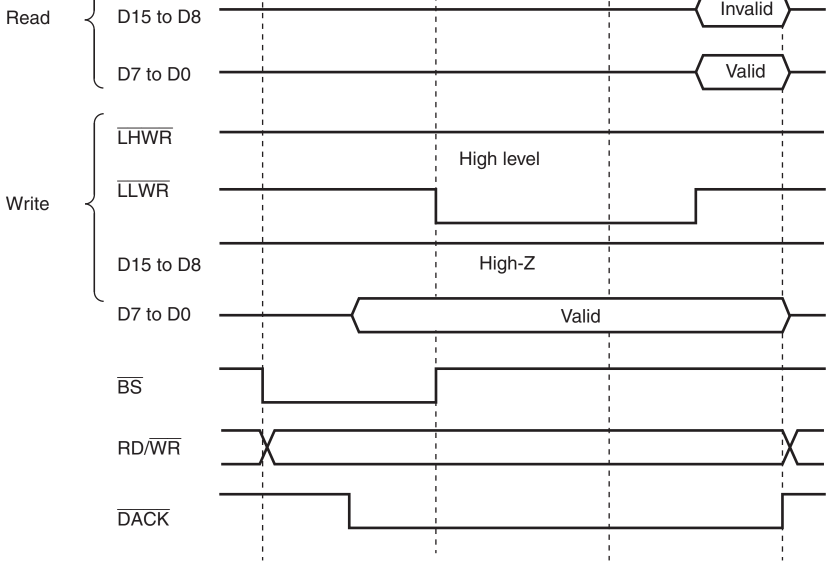
Notes: 1. n = 0 to 7
2. When RDNn = 0
3. When DKC = 0

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



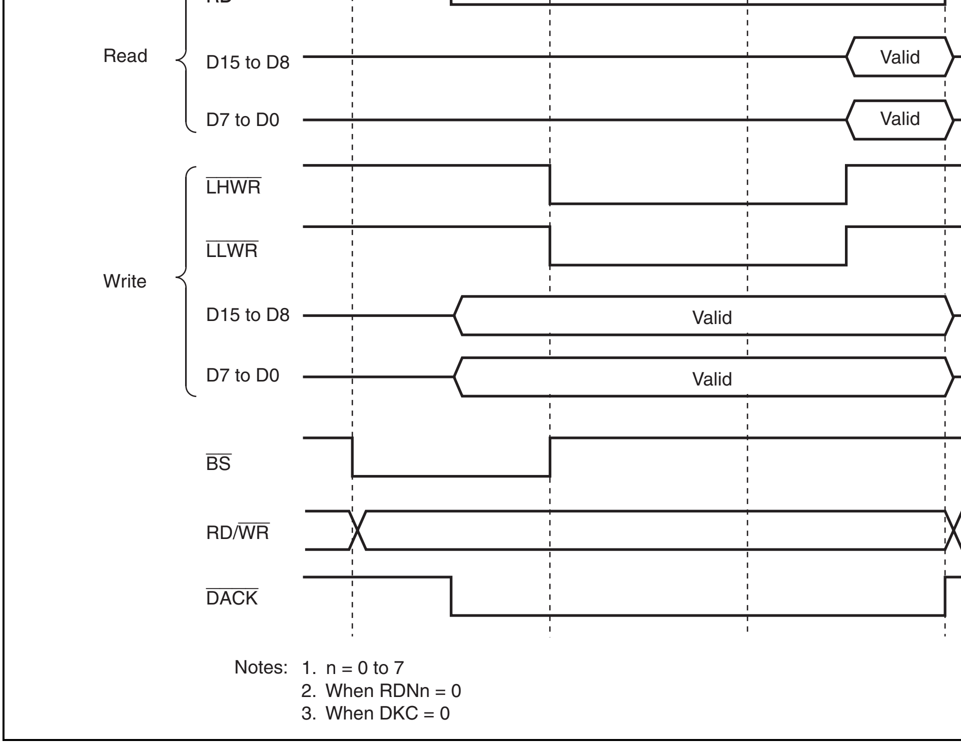
Notes: 1. n = 0 to 7
2. When RDNn = 0
3. When DKC = 0

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)



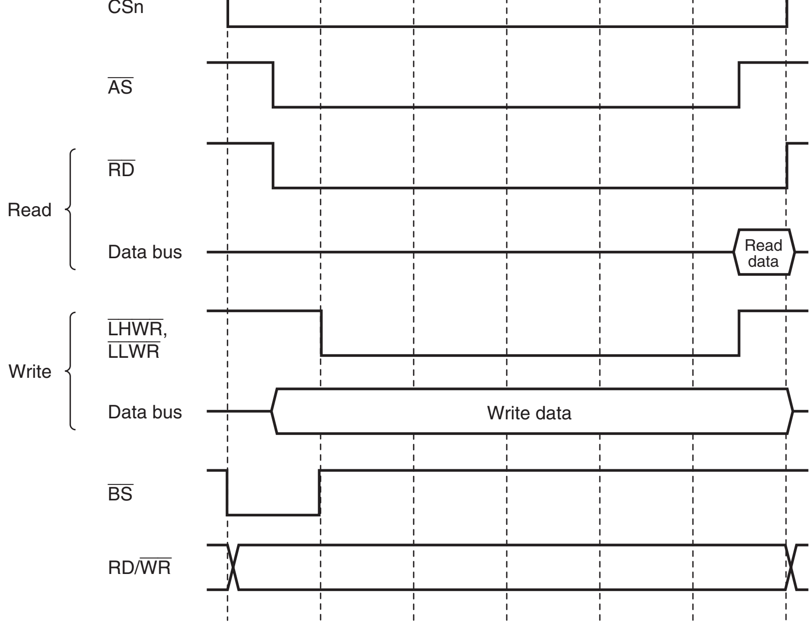
Notes: 1. n = 0 to 7
2. When RDNn = 0
3. When DKC = 0

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

(2) Pin Wait Insertion

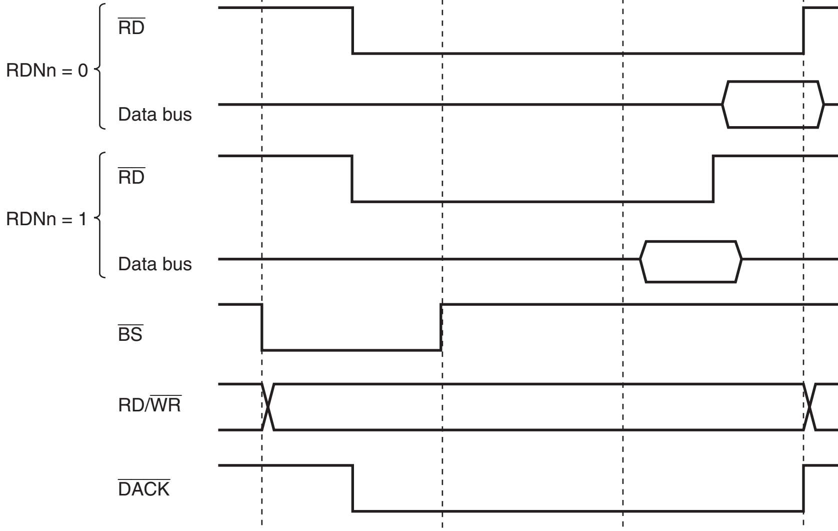
For 3-state access space, when the WAITE bit in BCR1 is set to 1 and the ICR bit for the
corresponding pin is set to 1, wait input by means of the $\overline{\text{WAIT}}$ pin is enabled. When the
address space is accessed in this state, a program wait (T_{PW}) is first inserted according to t
WTCRA and WTCRB settings. If the $\overline{\text{WAIT}}$ pin is low at the falling edge of Bϕ in the las
Tpw cycle, another Ttw cycle is inserted until the $\overline{\text{WAIT}}$ pin is brought high. The pin wai
insertion is effective when the Tw cycles are inserted to seven cycles or more, or when th
of Tw cycles to be inserted is changed according to the external devices. The WAITE bit
common to all areas. For details on 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 WAIT pin sampling.
2. n = 0 to 7
3. RDNn = 0

Figure 6.20 Example of Wait Cycle Insertion Timing

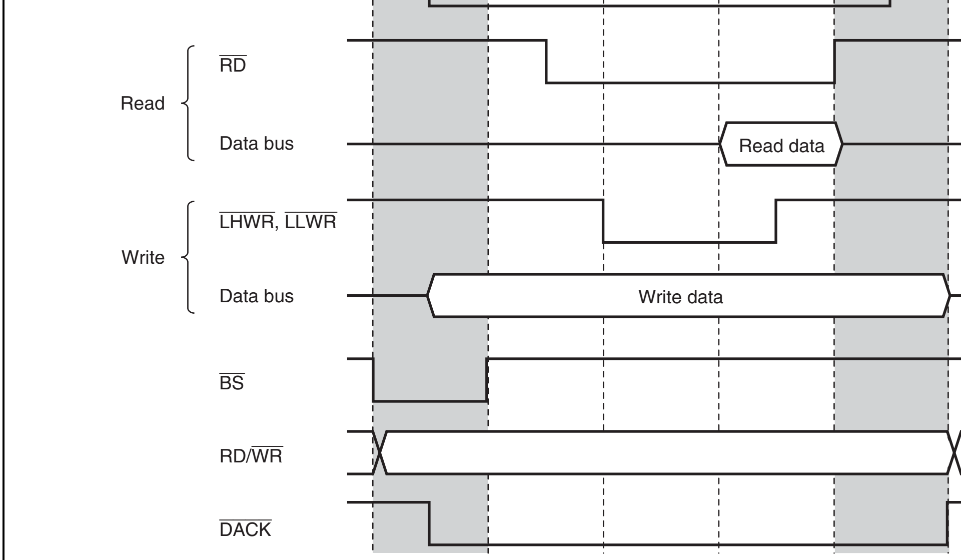


Notes: 1. n = 0 to 7
2. When DKC = 0

Figure 6.21 Example of Read Strobe Timing

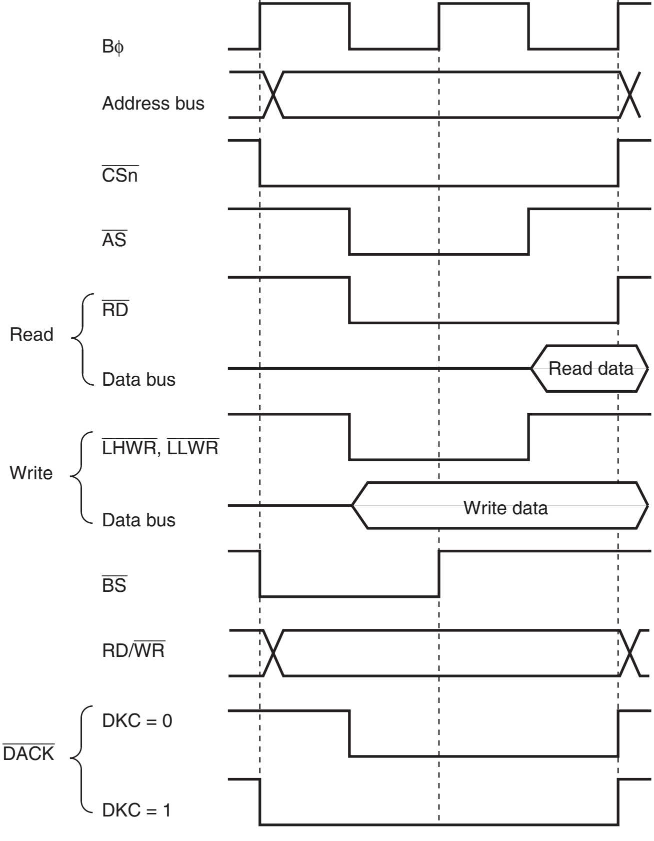
3-state access space.

Both extension cycle Th inserted before the basic bus cycle and extension cycle Tt inserte the basic bus cycle, or only one of these, can be specified for individual areas. Insertion o insertion can be specified for the Th cycle with the upper eight bits (CSXH7 to CSXH0) i 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 Exten



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

Figure 6.23 $\overline{\text{DACK}}$ 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 specifie control SRAM interface by setting bits BCSELn (n = 0 to 7) in SRAMCR. For the area as burst ROM interface or address/data multiplexed I/O interface, the SRAMCR setting and byte control SRAM interface cannot be used.

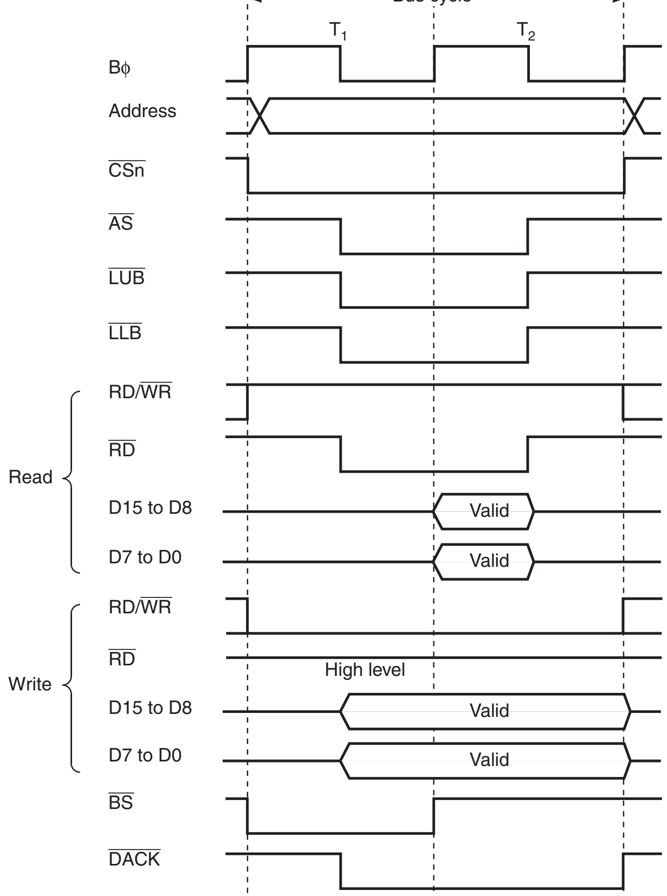
6.7.2 Data Bus

The bus width of the byte control SRAM space can be specified as 16-bit byte control S space according to bits ABWHn and ABWLn (n = 0 to 7) in ABWCR. The area specifie 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.

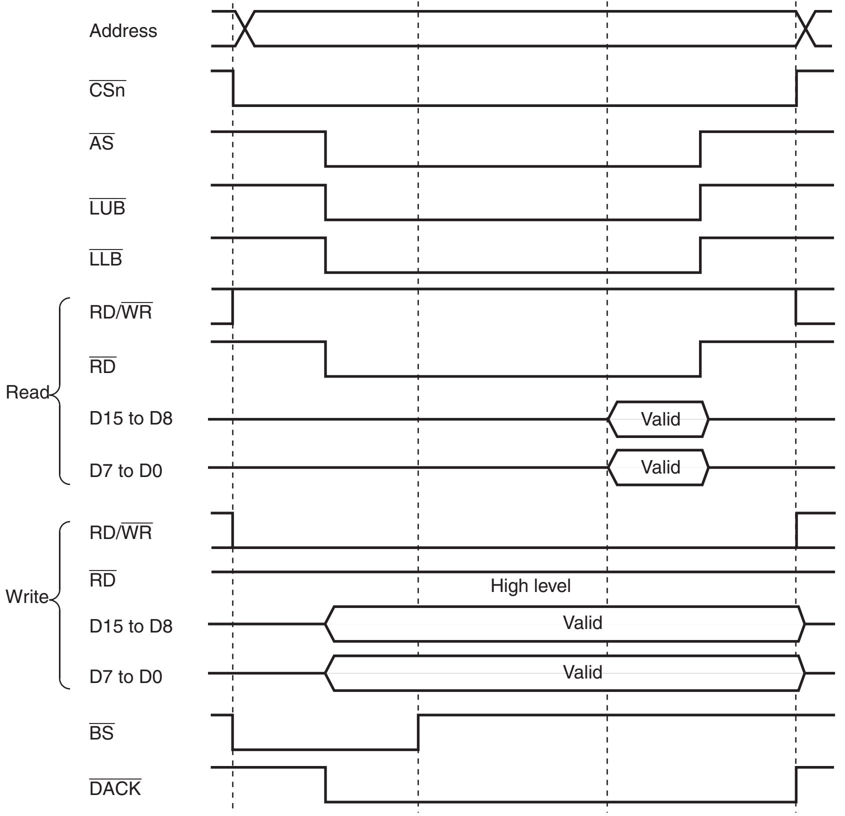
Access size and data alignment are the same as the basic bus interface. For details, see s 6.5.6, Endian and Data Alignment.

AS/AH	AS	Address strobe	Output	Strobe signal indicating that the output on the address bus is va a basic bus interface space or l control SRAM space is accesse
$\overline{\text{CSn}}$	$\overline{\text{CSn}}$	Chip select	Output	Strobe signal indicating that are selected
$\overline{\text{RD}}$	$\overline{\text{RD}}$	Read strobe	Output	Output enable for the SRAM wh byte control SRAM space is ac
RD/$\overline{\text{WR}}$	RD/$\overline{\text{WR}}$	Read/write	Output	Write enable signal for the SRA the byte control SRAM space is accessed
$\overline{\text{LHWR}}$/$\overline{\text{LUB}}$	$\overline{\text{LUB}}$	Lower-upper byte select	Output	Upper byte select when the 16- control SRAM space is accesse
$\overline{\text{LLWR}}$/$\overline{\text{LLB}}$	$\overline{\text{LLB}}$	Lower-lower byte select	Output	Lower byte select when the 16- control SRAM space is accesse
$\overline{\text{WAIT}}$	$\overline{\text{WAIT}}$	Wait	Input	Wait request signal used when external address space is acce
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



Note: n = 0 to 7

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

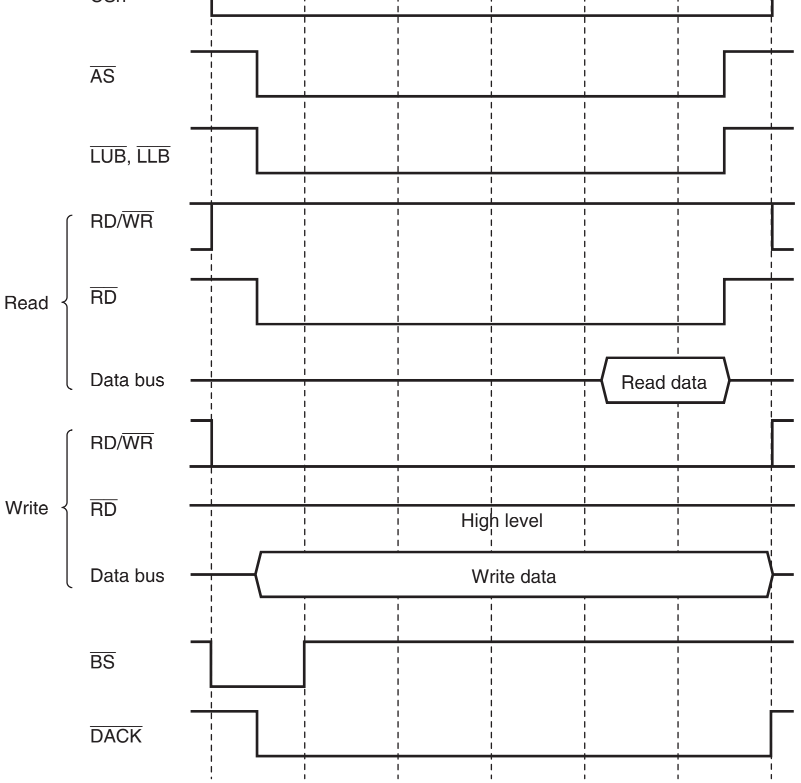


Note: n = 0 to 7

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 DI 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, see section 9, I/O Ports.

Figure 6.26 shows an example of wait cycle insertion timing.



Notes: 1. Upward arrows indicate the timing of 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 afte
cycle in the same way as the basic bus interface. For details, see section 6.6.6, Extensior
Select ($\overline{\text{CS}}$) Assertion Period.

6.7.8 $\overline{\text{DACK}}$ Signal Output Timing

For DMAC single address transfers, the $\overline{\text{DACK}}$ signal assert timing can be modified by
DKC bit in BCR1.

Figure 6.27 shows the $\overline{\text{DACK}}$ signal output timing. Setting the DKC bit to 1 asserts the
signal a half cycle earlier.

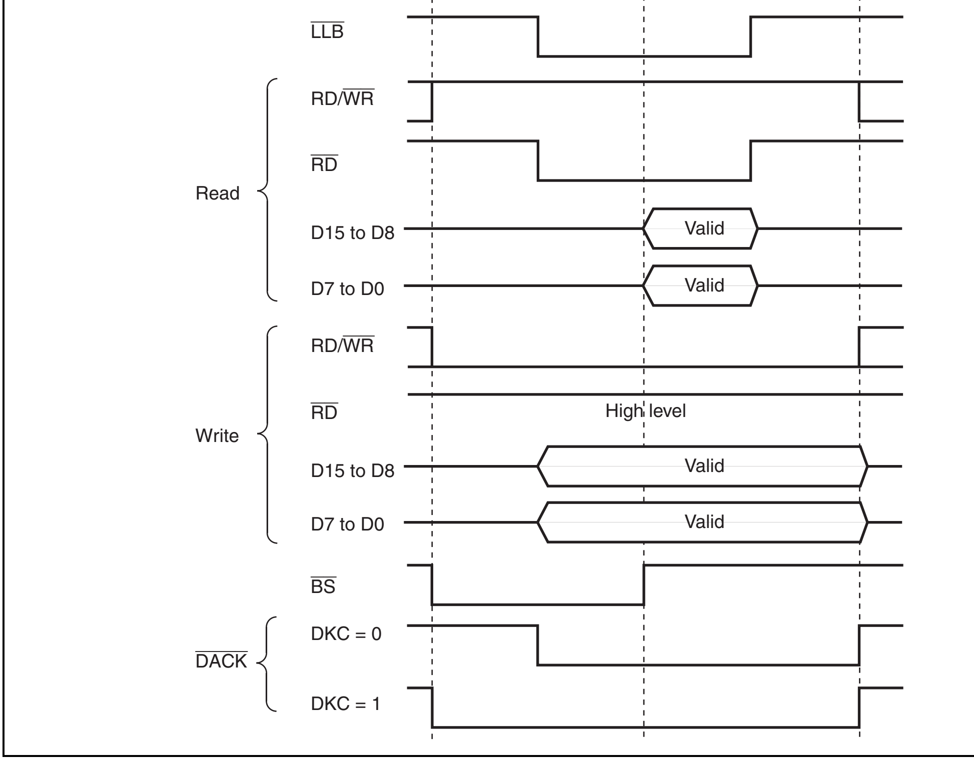


Figure 6.27 $\overline{\text{DACK}}$ Signal Output Timing

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

In the burst ROM interface, burst access covers only CPU read accesses. Other accesses covered by 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 a 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 or16-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 (D0) is valid.

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

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

6.8.4 Basic Timing

The number of access cycles in the initial cycle (full access) on the burst ROM interface i determined by the basic bus interface settings in ABWCR, ASTCR, WTCRA, WTCRB, a CSXHn in CSACR (n = 0 to 7). When area 0 or area 1 designated as burst ROM space is the CPU, the settings in RDNCR and bits CSXTn in CSACR (n = 0 to 7) are ignored.

From one to eight cycles can be selected for the burst cycle, according to the settings of b BSTS02 to BSTS00 and BSTS12 to BSTS10 in BROMCR. Wait cycles cannot be inserte addition, 4-word, 8-word, 16-word, or 32-word consecutive burst access can be performe according to the settings of BSTS01, BSTS00, BSTS11, and BSTS10 bits in BROMCR.

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

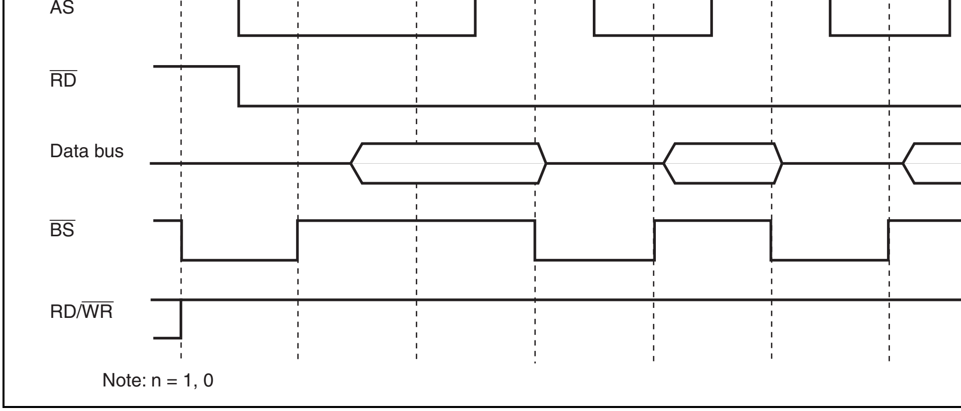
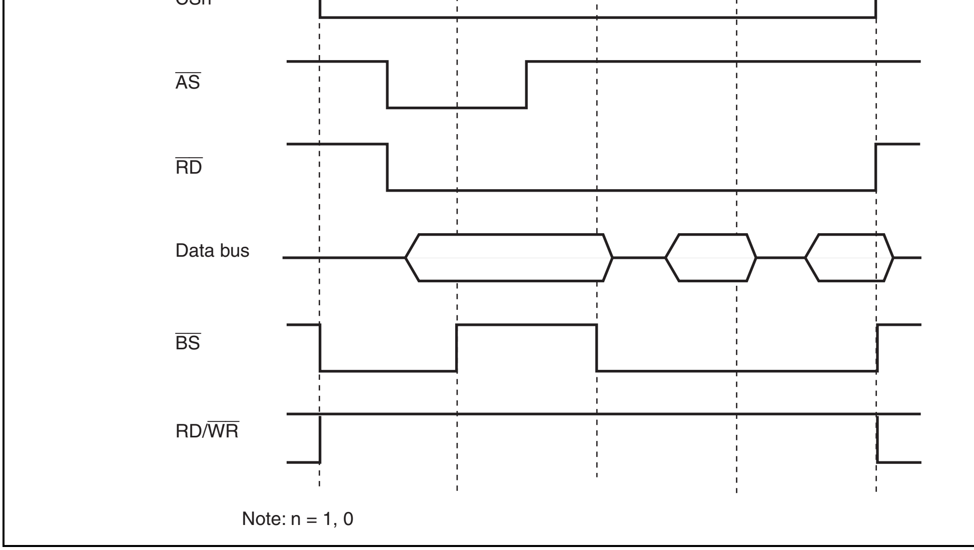


Figure 6.28 Example of Burst ROM Access Timing (ASTn = 1, Two Burst Cy



Note: n = 1, 0

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

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

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

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

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

In read accesses by the CPU, the burst ROM space is equivalent to the basic bus interfac Accordingly, extension cycles can be inserted before and after the burst access cycles.

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

6.9.2 Address/Data Multiplex

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

Table 6.18 Address/Data Multiplex

		Data Pins													
Bus Width	Cycle	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

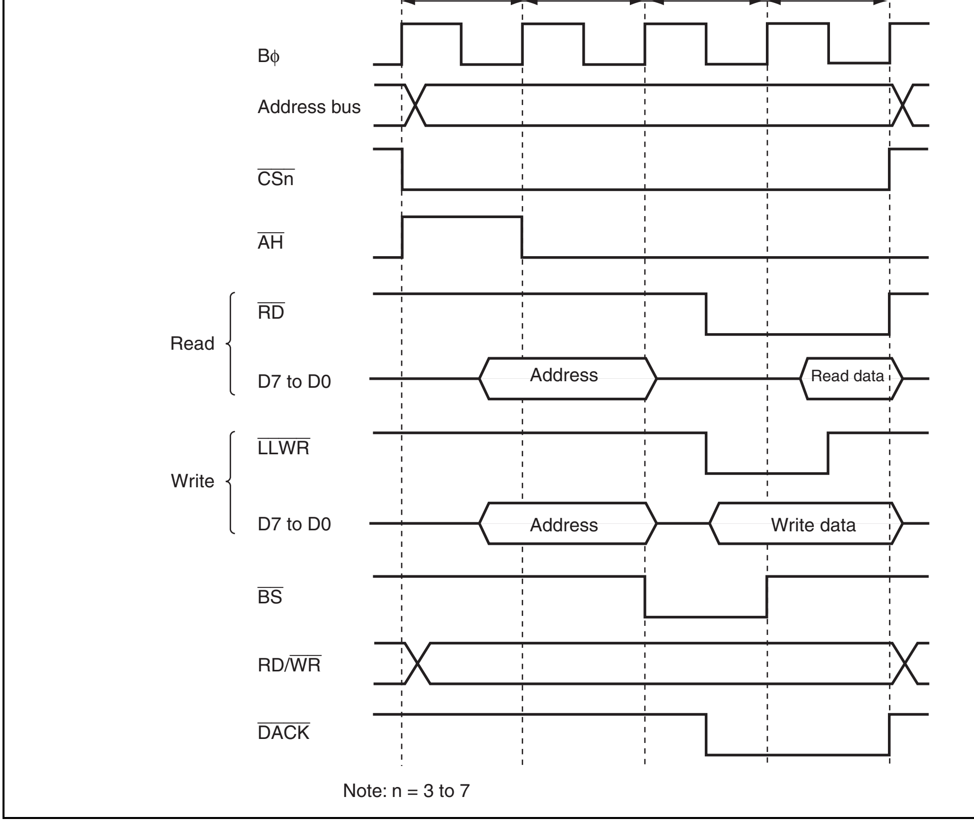
The bus width of the address/data multiplexed I/O space can be specified for either 8-bit space or 16-bit access space by the ABWHn and ABWLn bits (n = 3 to 7) in ABWCR.

For the 8-bit access space, D7 to D0 are valid for both address and data. For 16-bit access D15 to D0 are valid for both address and data. If the address/data multiplexed I/O space i 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

				address/data multiplexed I/O space
AS/AH	AH*	Address hold	Output	Signal to hold an address when the addres multiplexed I/O space is specified
RD	RD	Read strobe	Output	Signal indicating that the address/data mult space is being read
LHWR/LUB	LHWR	Low-high write	Output	Strobe signal indicating that the upper byte D8) is valid when the address/data multiple space is written
LLWR/LLB	LLWR	Low-low write	Output	Strobe signal indicating that the lower bytes is valid when the address/data multiplexed is written
D15 to D0	D15 to D0	Address/data	Input/ output	Address and data multiplexed pins for the address/data multiplexed I/O space. Only D7 to D0 are valid when the 8-bit spac specified. D15 to D0 are valid when the 16- specified.
A23 to A0	A23 to A0	Address	Output	Address output pin
WAIT	WAIT	Wait	Input	Wait request signal used when the externa space is accessed
BS	BS	Bus cycle start	Output	Signal to indicate the bus cycle start
RD/WR	RD/WR	Read/write	Output	Signal indicating the data bus input or outp

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



Note: n = 3 to 7

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

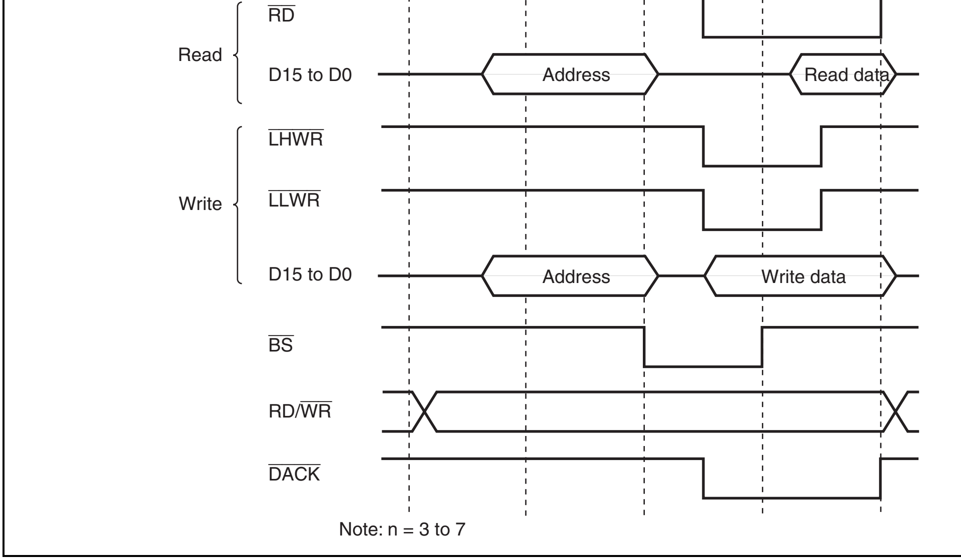


Figure 6.31 16-Bit Access Space Access Timing (ABWHn = 0, ABWLn =

6.9.6 Address Cycle Control

An extension cycle (Tmaw) can be inserted between Tma1 and Tma2 cycles to extend th signal output period by setting the ADDEX bit in MPXCR. By inserting the Tmaw cycl address setup for $\overline{\text{AH}}$ and the $\overline{\text{AH}}$ minimum pulse width can be assured.

Figure 6.32 shows the access timing when the address cycle is three cycles.

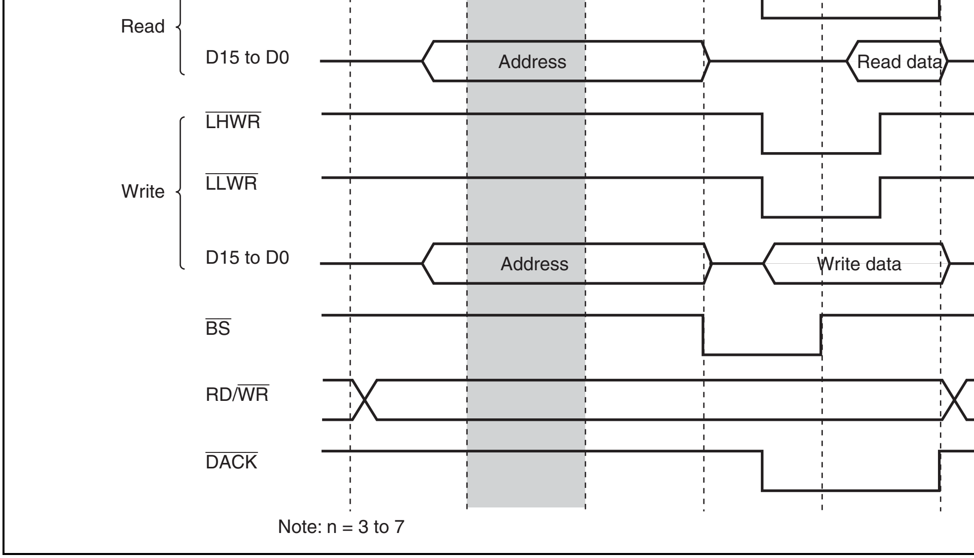


Figure 6.32 Access Timing of 3 Address Cycles (ADDEX = 1)

6.9.7 Wait Control

In the data cycle of the address/data multiplexed I/O interface, program wait insertion anc insertion by the $\overline{\text{WAIT}}$ pin are enabled in the same way as in the basic bus interface. For c see section 6.6.4, Wait Control.

Wait control settings do not affect the address cycles.

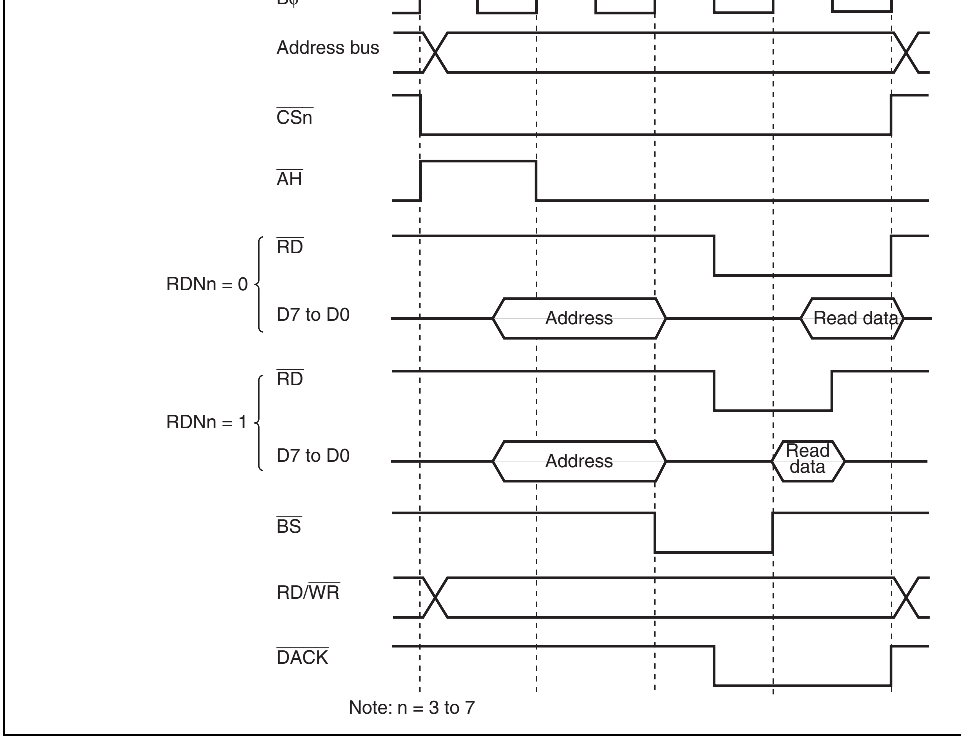


Figure 6.31 Read Strobe Timing

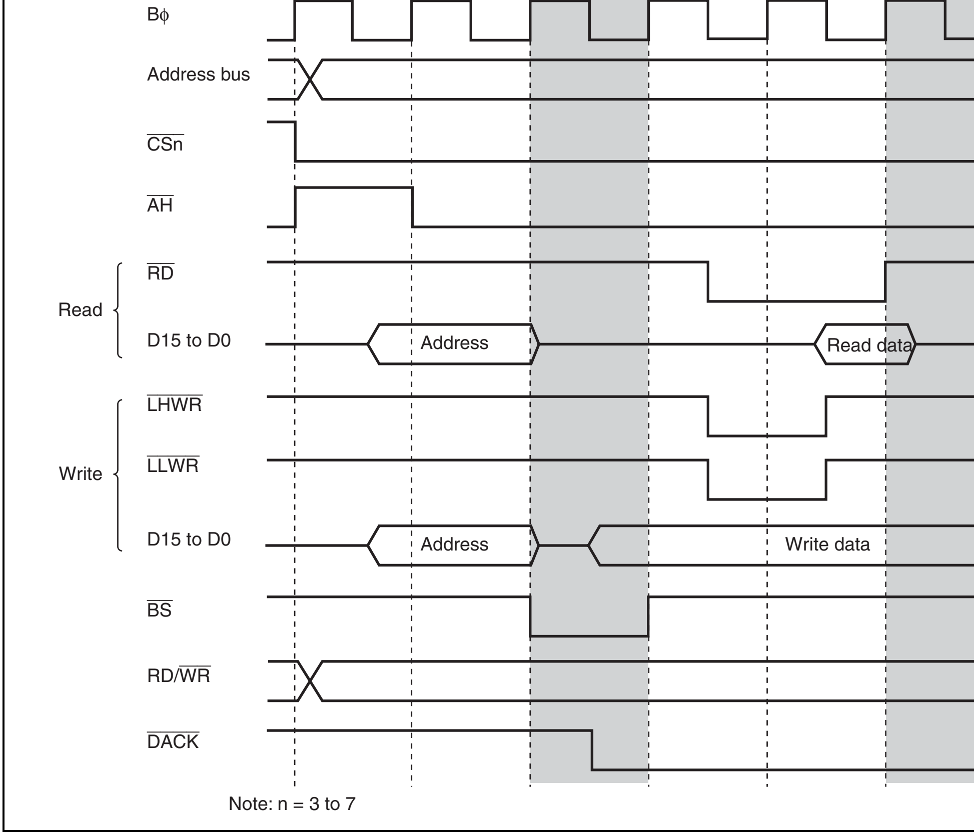


Figure 6.34 Chip Select ($\overline{\text{CS}}$) Assertion Period Extension Timing in Data Cy

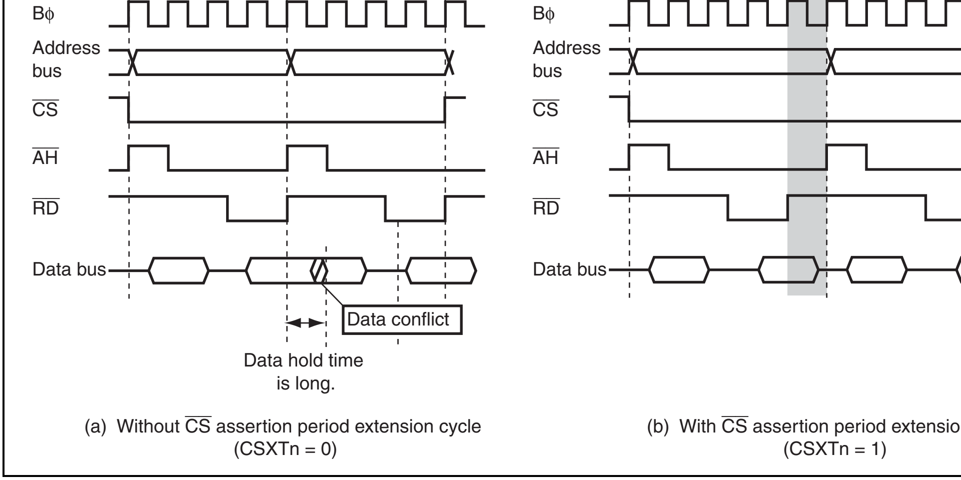


Figure 6.35 Consecutive Read Accesses to Same Area (Address/Data Multiplexed I/O Space)

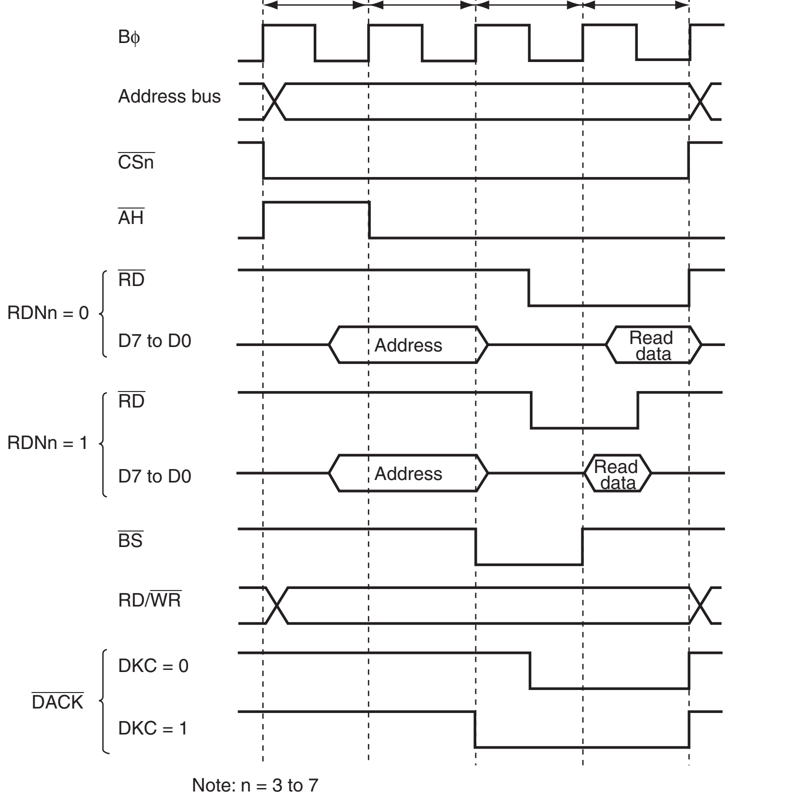


Figure 6.36 DACK Signal Output Timing

and write and previously accessed area.

1. When read cycles of different areas in the external address space occur consecutively
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 the bits IDLCA1 and IDLCA0 in setting B specified by the bits IDLCB1 and IDLCB0 in IDLCR: Setting A can be selecte one to four cycles, and setting B can be selected from one or two to four cycles. Setting be specified for each area by setting the bits IDLSEL7 to IDLSEL0 in IDLCR. Note tha IDLSEL7 to IDLSEL0 correspond to the previously accessed area of the consecutive ac

The number of idle cycles to be inserted under conditions 3 and 4, which is a condition t 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.

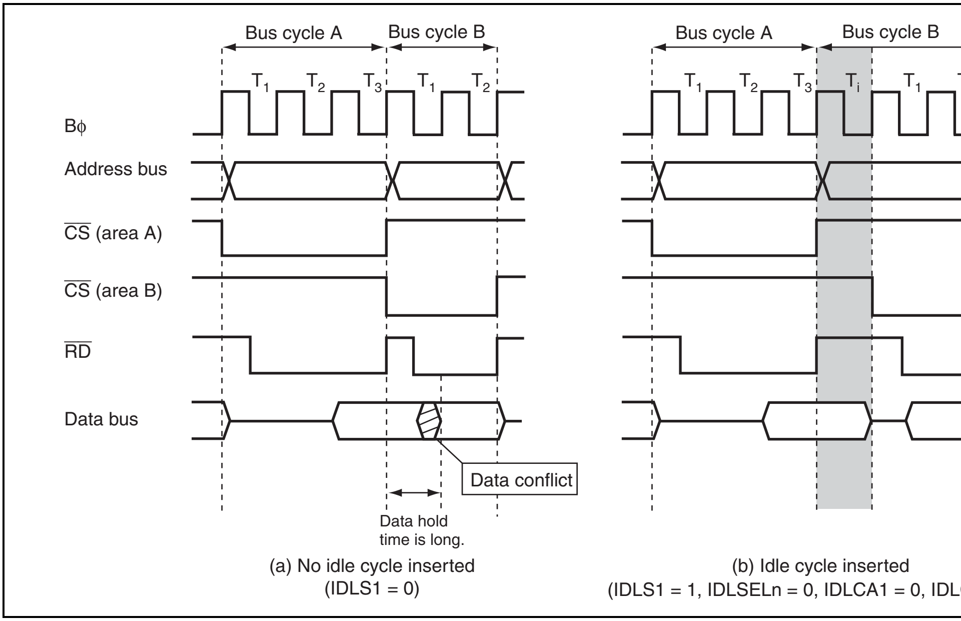


Figure 6.37 Example of Idle Cycle Operation (Consecutive Reads in Different

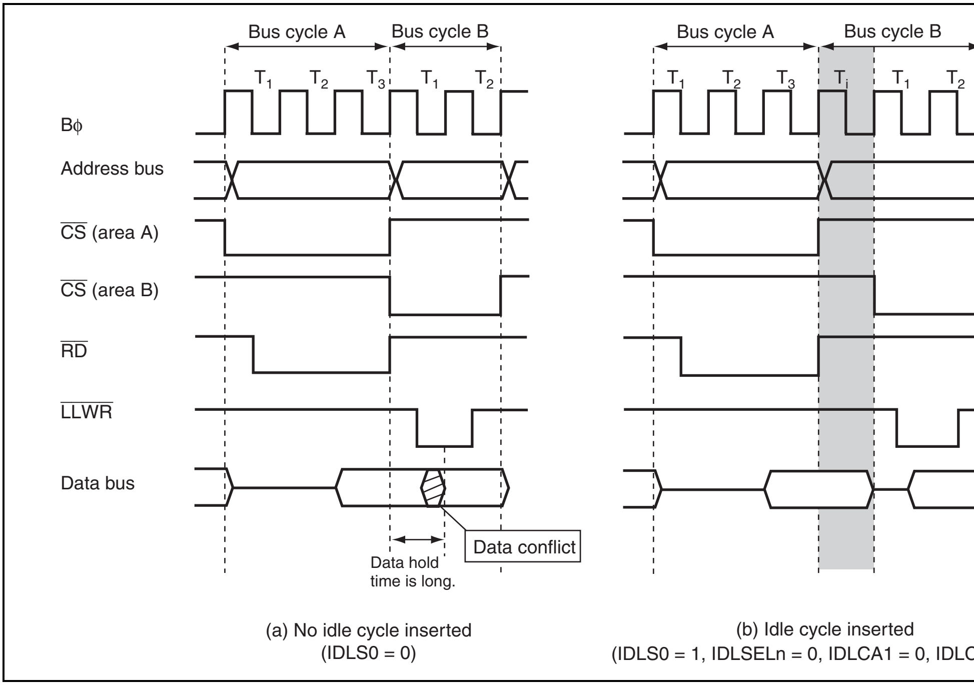
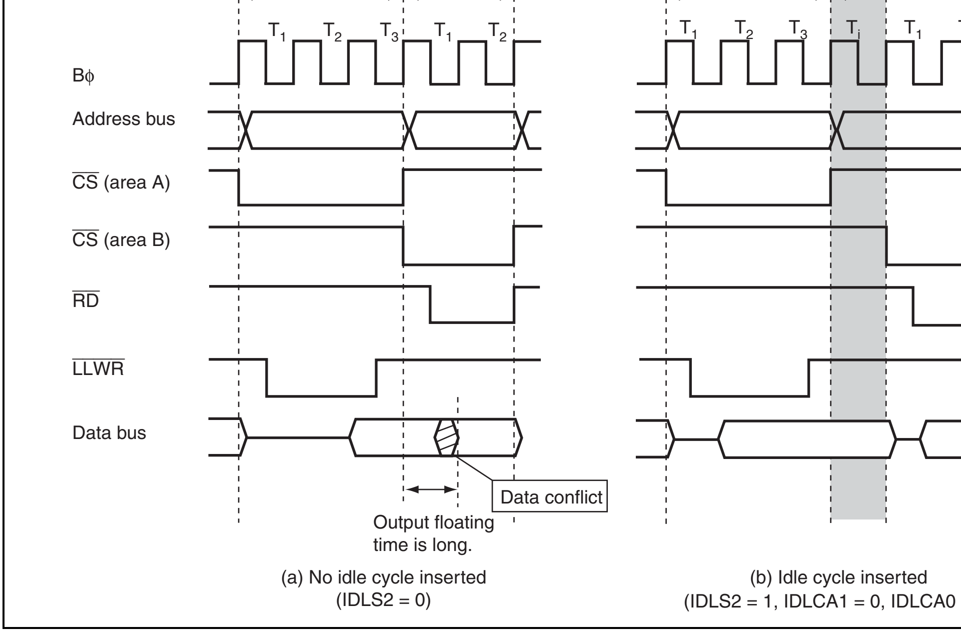


Figure 6.38 Example of Idle Cycle Operation (Write after Read)



(a) No idle cycle inserted
(IDLS2 = 0)

(b) Idle cycle inserted
(IDLS2 = 1, IDLCA1 = 0, IDLCA0

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

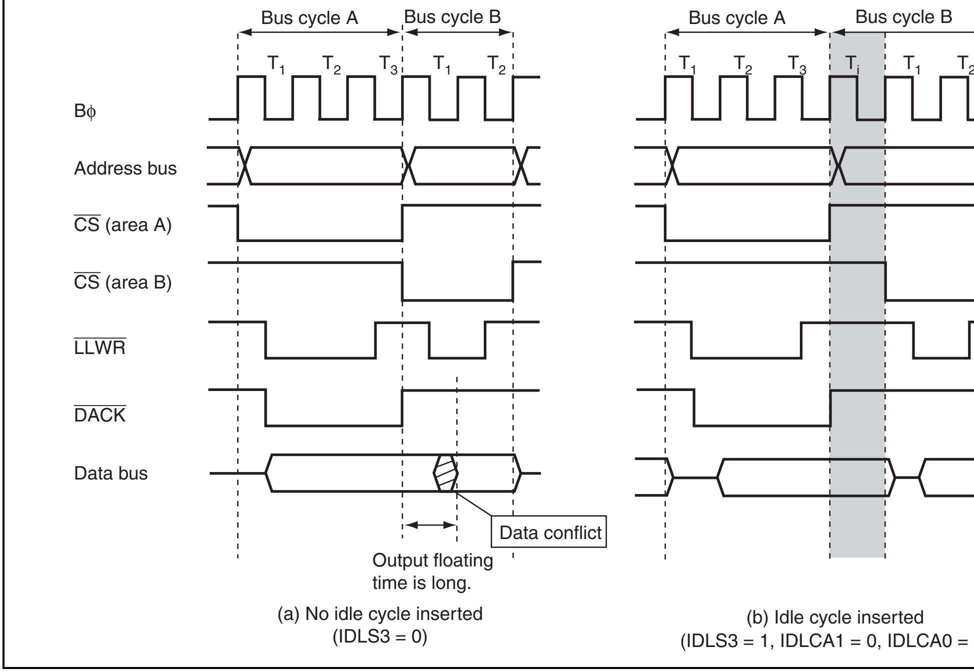


Figure 6.40 Example of Idle Cycle Operation (Write after Single Address Transfe

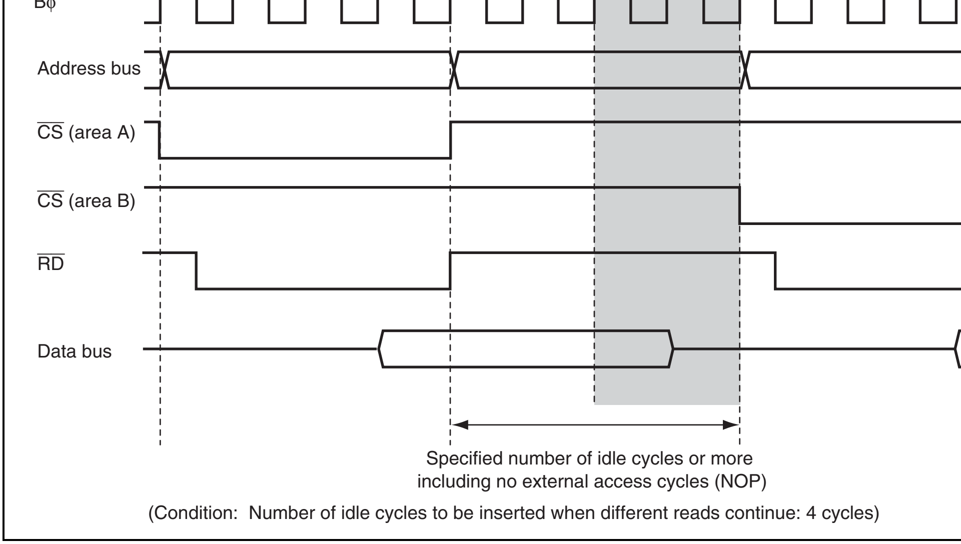
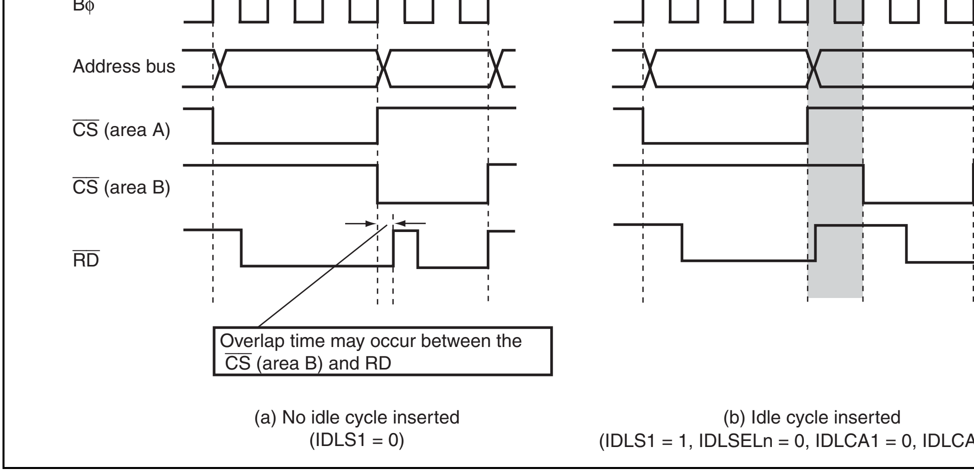

Bφ
Address bus
$\overline{CS}$ (area A)
$\overline{CS}$ (area B)
$\overline{RD}$
Data bus
Specified number of idle cycles or more
including no external access cycles (NOP)
(Condition: Number of idle cycles to be inserted when different reads continue: 4 cycles)

Figure 6.41 Idle Cycle Insertion Example


Bφ
Address bus
$\overline{\text{CS}}$ (area A)
$\overline{\text{CS}}$ (area B)
$\overline{\text{RD}}$
Overlap time may occur between the $\overline{\text{CS}}$ (area B) and RD

(a) No idle cycle inserted
(IDLS1 = 0)

(b) Idle cycle inserted
(IDLS1 = 1, IDLSELn = 0, IDLCA1 = 0, IDLCA

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

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

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

In external extended mode, when the BRLE bit in BCR1 is set to 1, and the ICR bit for t corresponding pin is 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 addres temporarily defers initiation of the bus cycle, and waits for the bus request from the exte master to be canceled.

In the external bus released state, when write access to SCKCR is granted to set the cloc frequency, the current setting for the clock frequency is deferred until the bus request of external bus master is canceled. For details of the SCKCR, see section 18, Clock Pulse C

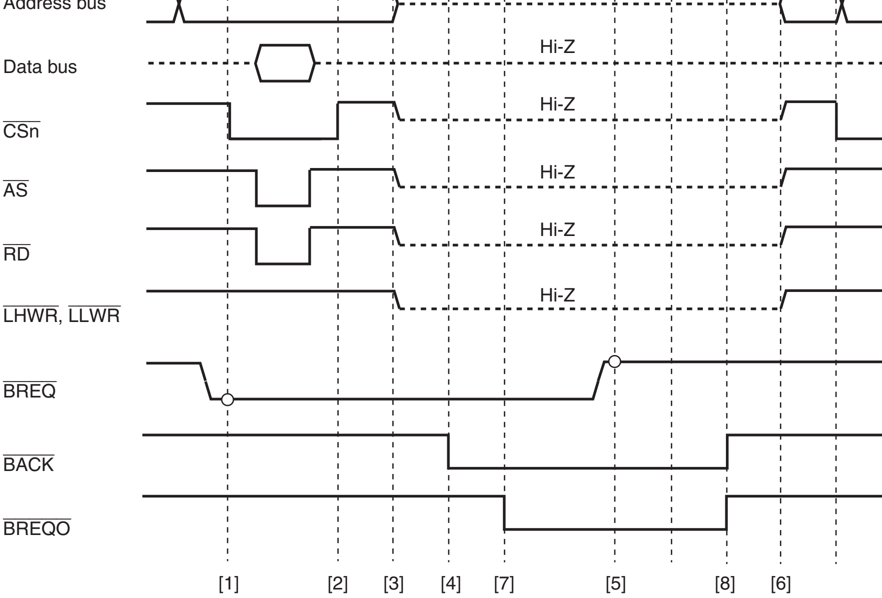
If the BREQOE bit in BCR1is set to 1, the $\overline{\text{BREQO}}$ pin can be driven low when any of t following requests are issued, to request cancellation of the bus request externally.

- 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 or module-clock-stop mode
- When write access to SCKCR is granted to set 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
$\overline{\text{AS}}$	High impedance
$\overline{\text{AH}}$	High impedance
RD/$\overline{\text{WR}}$	High impedance
$\overline{\text{RD}}$	High impedance
$\overline{\text{LUB}}$, $\overline{\text{LLB}}$	High impedance
$\overline{\text{LHWR}}$, $\overline{\text{LLWR}}$	High impedance
$\overline{\text{DACKn}}$ (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

Access Space	Access	Number of Access C
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.

The number of access cycles to the registers for on-chip peripheral modules is shown in t

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.

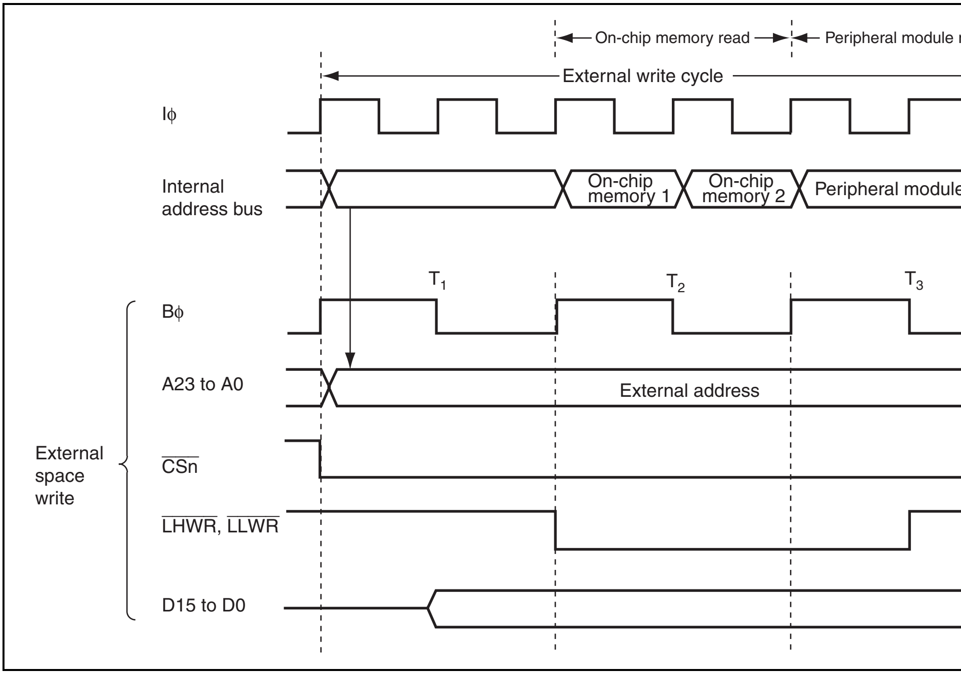


Figure 6.44 Example of Timing when Write Data Buffer Function is Use

two cycles. However, from the next cycle onward an internal memory or an external acce internal I/O register write are executed in parallel rather than waiting until it ends.

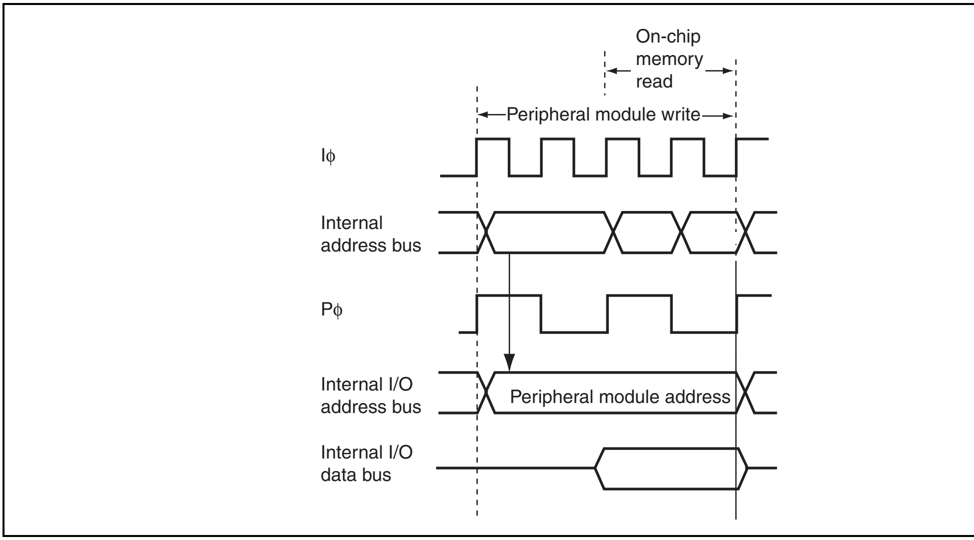


Figure 6.45 Example of Timing when Peripheral Module Write Data Buffer Function is Used

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 DMA

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 IBCCS bit in BCR2 is set to 1, the DTC transfers the bu CPU.

generated. When the DMAC accesses an external bus space, the DMAC first takes contr
bus from the internal bus arbiter and then requests a bus to the external bus arbiter.

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 IBCCS 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 and the ICR bit of the corresponding pin are set to 1, a bus request is sent to the b

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

the external ROM, specify the registers before external accesses other than the instruction
from the external ROM are generated.

(2) External Bus Release Function and All-Module-Clock-Stop Mode

In this LSI, if the ACSE bit in MSTPCRA is set to 1, and then a SLEEP instruction is exe
with the setting for all peripheral module clocks to be stopped (MSTPCRA, MSTPCRB =
H'FFFFFFFF) or for operation of the 8-bit timer module alone (MSTPCRA, MSTPCRB =
to 0]FFFFFF), and a transition is made to the sleep state, the all-module-clock-stop mode
entered in which the clock is also stopped for the bus controller and I/O ports. For details
section 19, 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 RAM, 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.

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

  Auto request: Activated by the CPU
  (cycle stealing or burst access can be selected)

  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 add
  returns to the transfer start address

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

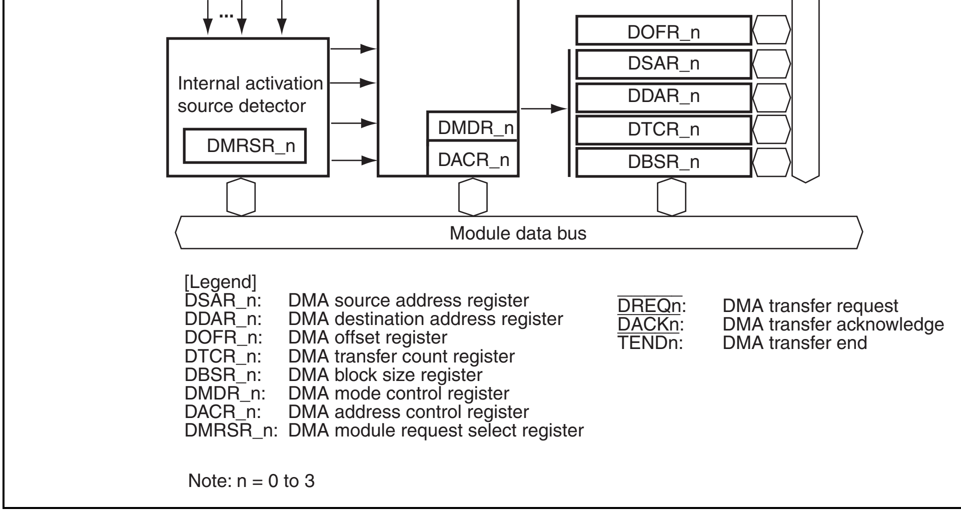
  Block transfer mode: One block data is transferred at a single transfer request

  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
DDAR_n:	DMA destination address register
DOFR_n:	DMA offset register
DTCR_n:	DMA transfer count register
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

$\overline{\text{DREQ}}$n:	DMA transfer request
$\overline{\text{DACK}}$n:	DMA transfer acknowledge
$\overline{\text{TEND}}$n:	DMA transfer end

Note: n = 0 to 3

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)

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

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

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

Although DTCR 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

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	BKSZH23	BKSZH22	BKSZH21	BKSZH20	BKSZH19	BKSZH18	BKSZH17
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	BKSZ15	BKSZ14	BKSZ13	BKSZ12	BKSZ11	BKSZ10	BKSZ9
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	BKSZ7	BKSZ6	BKSZ5	BKSZ4	BKSZ3	BKSZ2	BKSZ1
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
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 s 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 wh DMA is in operation. The value is decremente every time data is transferred. When the rema becomes 0, the value of the BKSZH bits is loa 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.

R/W	R	R	R	R	R	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.

transfer.

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

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

Operating modes and transfer methods must no
changed while this bit is set to 1.

0: Disables a data transfer

1: Enables a data transfer (DMA is in operation

[Clearing conditions]

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

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

- When an address error or an NMI interrupt requested
- In the reset state or hardware standby mode

				1: Disables TEND signal output
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 $\overline{\text{DREQ}}$ signal used in external request mode is When a block transfer is performed in externa mode, clear this bit to 0. 0: Low level detection 1: Falling edge detection (the first transfer aft transfer enabled is detected on a low level
26	NRD	0	R/W	Next Request Delay Selects the accepting timing of the next transf 0: Starts accepting the next transfer request completion of the current transfer 1: Starts accepting the next transfer request after completion of the current transfer
25, 24	—	All 0	R	Reserved These bits are always read as 0 and cannot b modified.
23	ACT	0	R	Active State Indicates the operating state for the channel. 0: Waiting for a transfer request or a transfer 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 b 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 inte been generated in DMAC module stop mode, tl not set to 1.
18	—	0	R	Reserved This bit is always read as 0 and cannot be mod
17	ESIF	0	R/(W)*	Transfer Escape Interrupt Flag Indicates that a transfer escape end interrupt ha requested. A transfer escape end means that a is terminated before the transfer counter reache 0: A transfer escape end interrupt has not bee requested 1: A transfer escape end interrupt has been re [Clearing conditions] • When setting the DTE bit to 1 • When clearing to 0 before reading ESIF = 1 [Setting conditions] • When a transfer size error interrupt is reque • When a repeat size end interrupt is request • When a transfer end interrupt by an extende 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 14	DTSZ1 DTSZ0	0 0	R/W R/W	Data Access Size 1 and 0 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 12	MDS1 MDS0	0 0	R/W R/W	Transfer Mode Select 1 and 0 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 total size set in DTCR is less than the data acces • In block transfer mode, the total transfer siz DTCR is less than the block size 0: Disables a transfer size error interrupt reques 1: Enables a transfer size error interrupt reques
10	—	0	R	Reserved This bit is always read as 0 and cannot be mod
9	ESIE	0	R/W	Transfer Escape Interrupt Enable Enables/disables a transfer escape end interru 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 reques 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 reque 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 reques 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 interrupt source. This bit enables or disables t source flag selected by DMRSR. 0: To clear the source in DMA transfer is disab Since the on-chip module interrupt source i cleared in DMA transfer, it should be cleare CPU or DTC transfer. 1: To clear the source in DMA transfer is enab Since the on-chip module interrupt source i in DMA transfer, it does not require an inter the CPU or DTC transfer.
4, 3	—	All 0	R	Reserved These bits are always read as 0 and cannot b 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.

Bit Name	—	—	SAT1	SAT0	—	—	DAT1
Initial Value	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R/W

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

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

Bit	Bit Name	Initial Value	R/W	Description
31	AMS	0	R/W	Address Mode Select Selects address mode from single or dual add mode. In single address mode, the $\overline{\text{DACK}}$ pin according to the DACKE bit. 0: Dual address mode 1: Single address mode

26	RPTIE	0	R/W	Repeat Size End Interrupt Enable Enables/disables a repeat size end interrupt rec In repeat transfer mode, when the next transfer requested after completion of a 1-repeat-size d transfer while this bit is set to 1, the DTE bit in I cleared to 0. At this time, the ESIF bit in DMDR 1 to indicate that a repeat size end interrupt is requested. Even when the repeat area is not sp (ARS1 = 1 and ARS0 = 0), a repeat size end in after a 1-block data transfer can be requested. In addition, in block transfer mode, when the ne transfer is requested after 1-block data transfer this bit is set to 1, the DTE bit in DMDR is clear At this time, the ESIF bit in DMDR is set to 1 to 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 c transfer mode. 00: Specify the block area or repeat area on the address 01: Specify the block area or repeat area on the destination address 10: Do not specify the block area or repeat area 11: Setting prohibited

				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
19, 18	—	All 0	R	Reserved These bits are always read as 0 and cannot b modified.
17	DAT1	0	R/W	Destination Address Update Mode 1 and 0
16	DAT0	0	R/W	Select the update method of the destination a (DDAR). When DDAR is not specified as the t destination in single address mode, this bit is i 00: Destination address is fixed 01: Destination address is updated by adding 10: Destination address is updated by adding according to the data access size 11: Destination address is updated by subtrac or 4 according to the data access size

				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 specified is ignored. 0: Disables an interrupt request for an extended 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.

				becomes the start address and the end addre 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 C 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 resumed 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

Mbytes in units of byte and a power of 2.

When the lower address is overflowed from the extended repeat area by address update, the a becomes the start address and the end address 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.

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 add
11001	32 Mbytes specified as extended repeat area by the lower 25 bits of the add
11010	64 Mbytes specified as extended repeat area by the lower 26 bits of the add
11011	128 Mbytes specified as extended repeat area by the lower 27 bits of the ac
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 Re
Address Mode	**Transfer mode**	**Activation Source**	**Common Function**	**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/ $\overline{\text{DACK}}$

When the auto request setting is selected as the activation source, the cycle stealing or bu can be selected. When the total transfer size is not specified (DTCR = H'00000000), the t counter is stopped and the transfer is continued without the limitation of the transfer coun 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 $\overline{\text{TENI}}$
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.

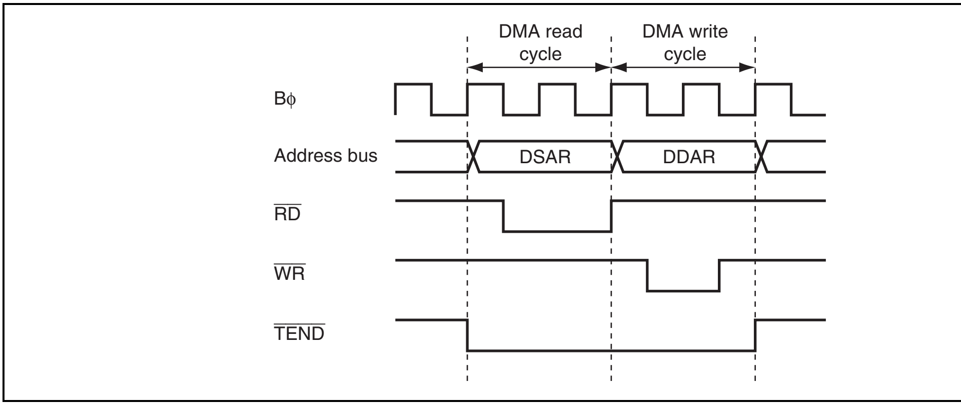


Figure 7.2 Example of Signal Timing in Dual Address Mode

Figure 7.3 Operations 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 perfor 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

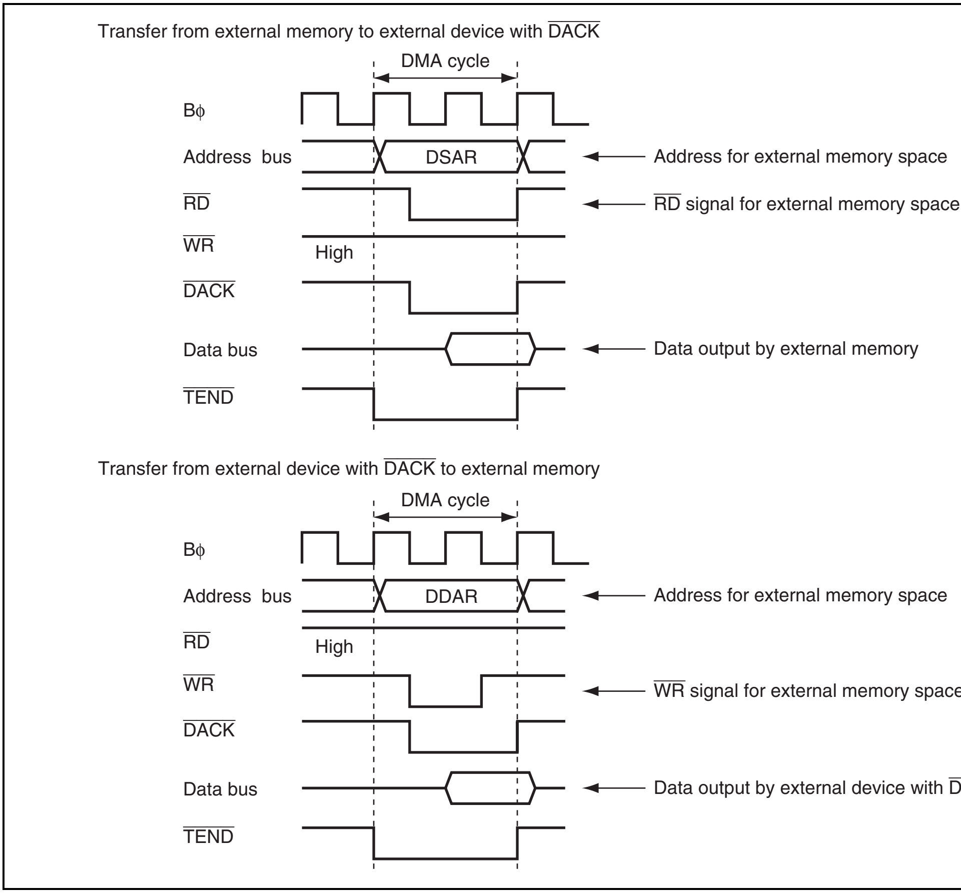

Transfer from external memory to external device with DACK
DMA cycle
Bϕ
Address bus
DSAR
Address for external memory space
RD
RD signal for external memory space
WR
High
DACK
Data bus
Data output by external memory
TEND
Transfer from external device with DACK to external memory
DMA cycle
Bϕ
Address bus
DDAR
Address for external memory space
RD
High
WR
WR signal for external memory space
DACK
Data bus
Data output by external device with D
TEND

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 ev
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.

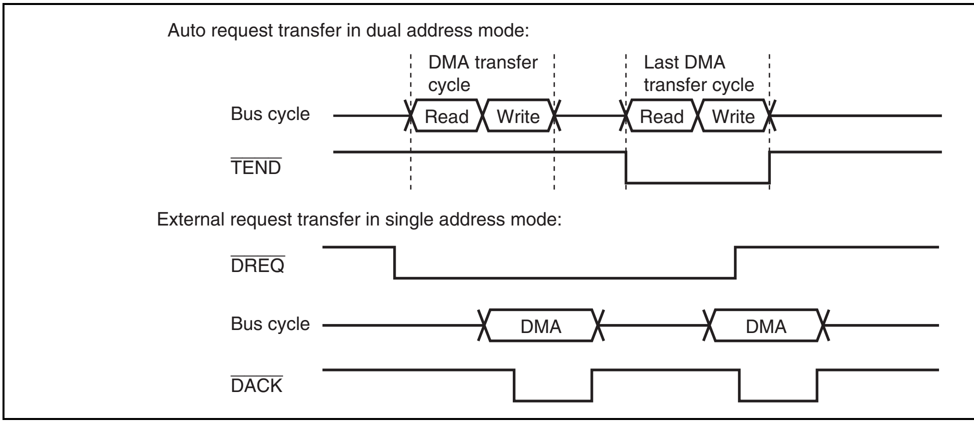


Figure 7.7 Example of Signal Timing in Normal Transfer Mode

Figure 7.8 Operations 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

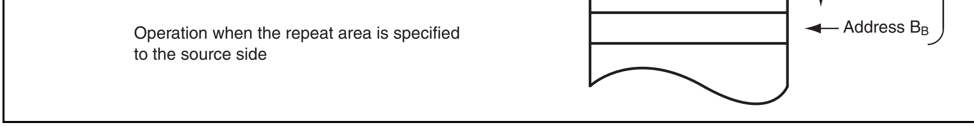

Operation when the repeat area is specified to the source side
Address B_B

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 i up to 65536 × data access size.

While one block of data is being transferred, transfer requests from other channels are su 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 a in DACR. The address specified as the block area returns to the transfer start address wh 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.

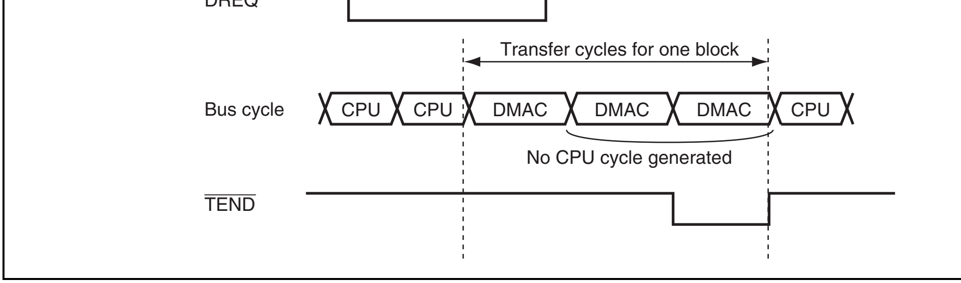


Figure 7.10 Operations in Block Transfer Mode

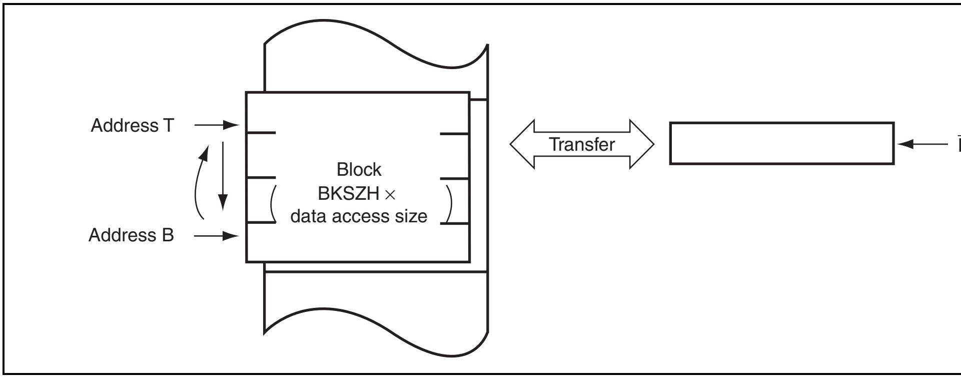


Figure 7.11 Operation in Single Address Mode in Block Transfer Mode (Block Area Specified)

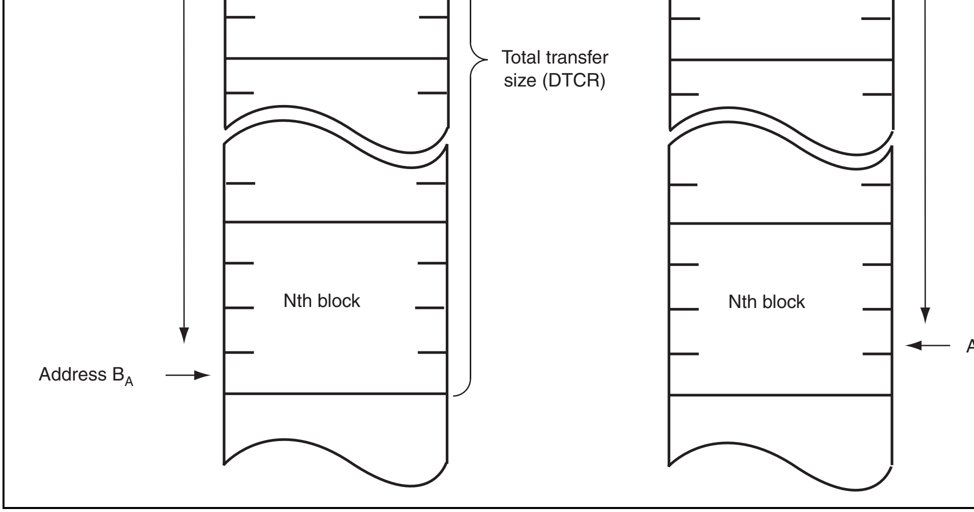


Figure 7.12 Operation in Dual Address Mode in Block Transfer Mode (Block Area Not Specified)

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 transf 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.

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
RXI3 (receive data full interrupt for SCI channel 3)	SCI_3	15
TXI3 (transmit data empty interrupt for SCI channel 3)	SCI_3	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 in internal space is performed, select an activation source from the auto request a 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 s 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

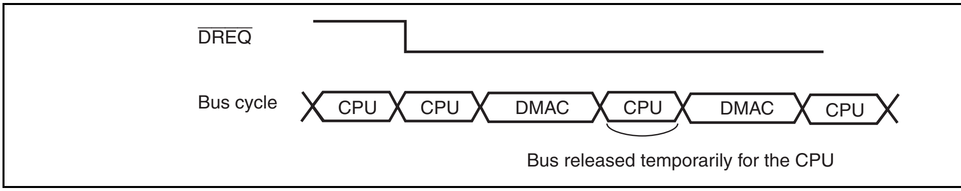


Figure 7.13 Example of Timing in Cycle Stealing Mode

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.

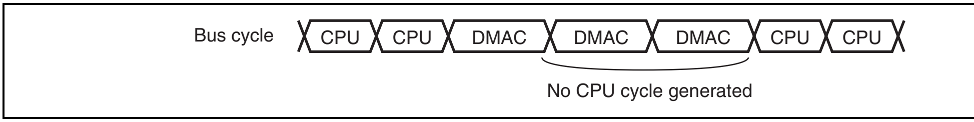


Figure 7.14 Example of Timing in Burst Mode

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 a example, to use a ring buffer as the transfer target, the contents of the address register sh return to the start address of the buffer every time the contents reach the end address of t (overflow on the ring buffer address). This operation can automatically be performed usi extended repeat area function of the DMAC.

The extended repeat areas can be specified independently to the source address register (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 DAR DACR. The extended repeat area sizes for each side can be specified independently.

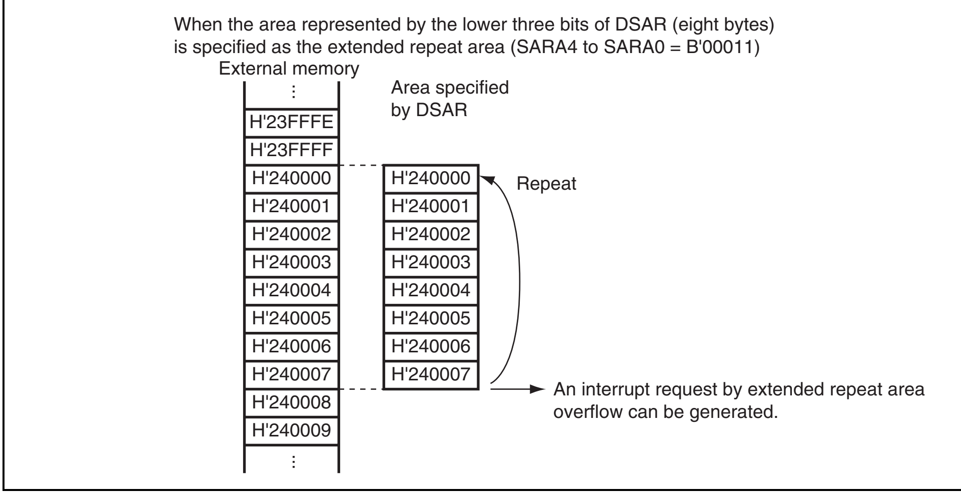

When the area represented by the lower three bits of DSAR (eight bytes)
is specified as the extended repeat area (SARA4 to SARA0 = B'00011)
External memory
Area specified
by DSAR
H'23FFFE
H'23FFFF
H'240000
H'240001
H'240002
H'240003
H'240004
H'240005
H'240006
H'240007
H'240008
H'240009
H'240000
H'240001
H'240002
H'240003
H'240004
H'240005
H'240006
H'240007
Repeat
An interrupt request by extended repeat area
overflow can be generated.

Figure 7.15 Example of Extended Repeat Area Operation

repeat area (SARA4 to SARA0 = 3) and the block size in block transfer mode is specified to 5 (bits 23 to 16 in DTCR = 5).

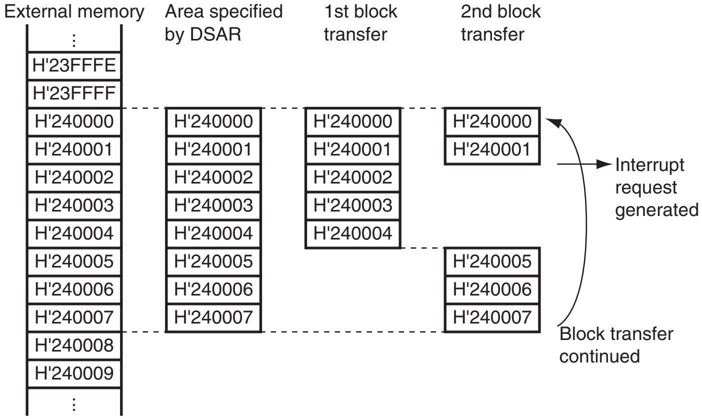


Figure 7.16 Example of Extended Repeat Area Function in Block Transfer M

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.

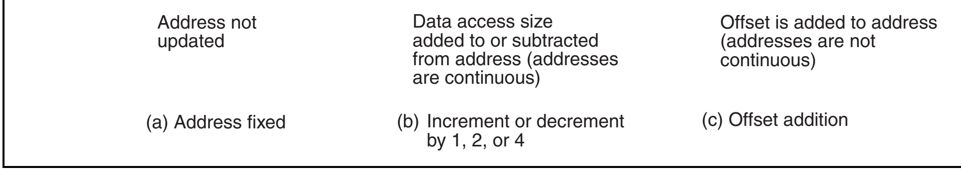

Address not updated
Data access size added to or subtracted from address (addresses are continuous)
Offset is added to address (addresses are not continuous)
(a) Address fixed
(b) Increment or decrement by 1, 2, or 4
(c) Offset addition

Figure 7.17 Address Update Method

In item (a), Address fixed, the transfer source or destination address is not updated indic same address.

In item (b), Increment or decrement by 1, 2, or 4, the transfer source or destination addre incremented or decremented by the value according to the data access size at each transf word, or longword can be specified as the data access size. The value of 1 for byte, 2 for 4 for longword is used for updating the address. This operation realizes the data transfer consecutive areas.

In item (c), Offset addition, the address update does not depend on the data access size. specified by DOFR is added to the address every time the DMAC transfers data of the d 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.

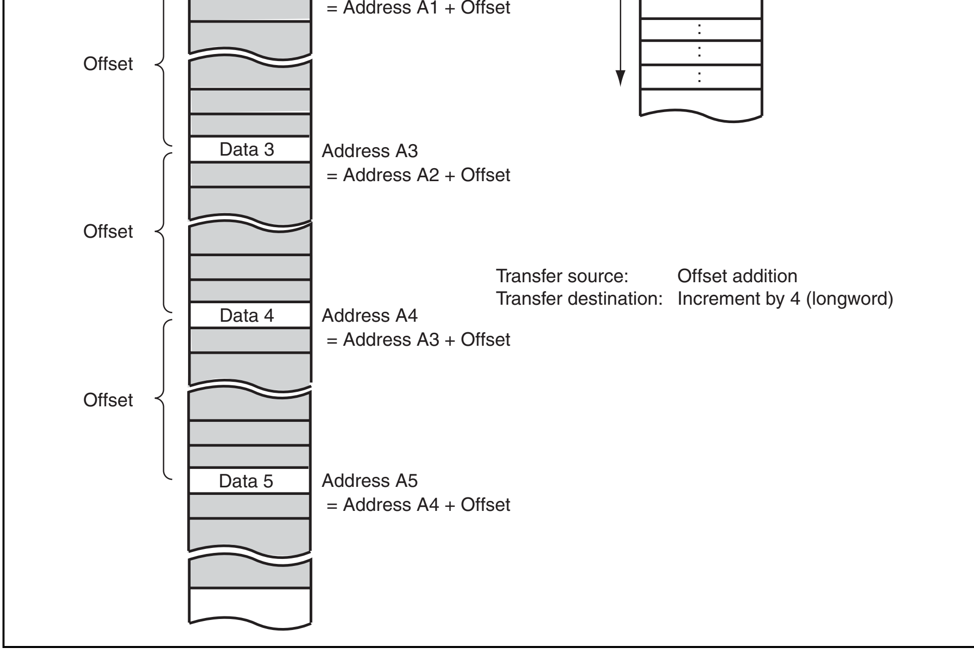


Figure 7.18 Operation of Offset Addition

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.

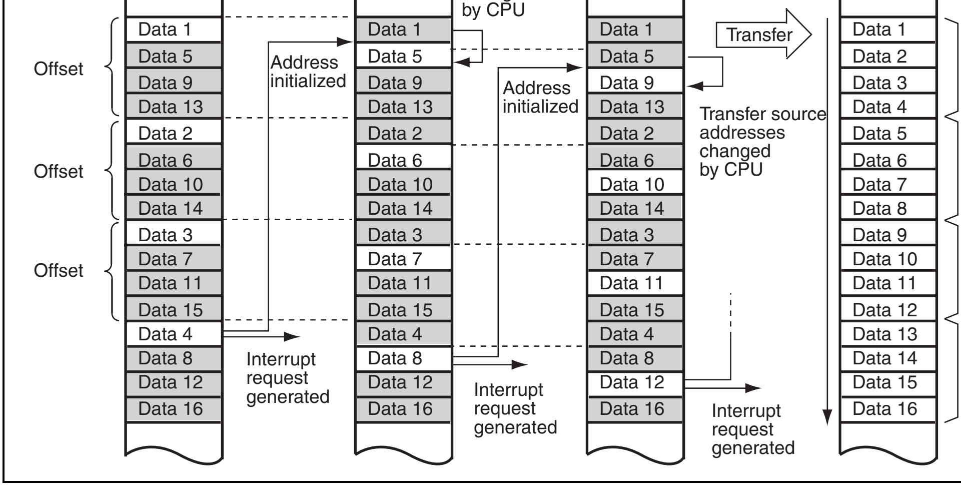


Figure 7.19 XY Conversion Operation Using Offset Addition in Repeat Transfe

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.

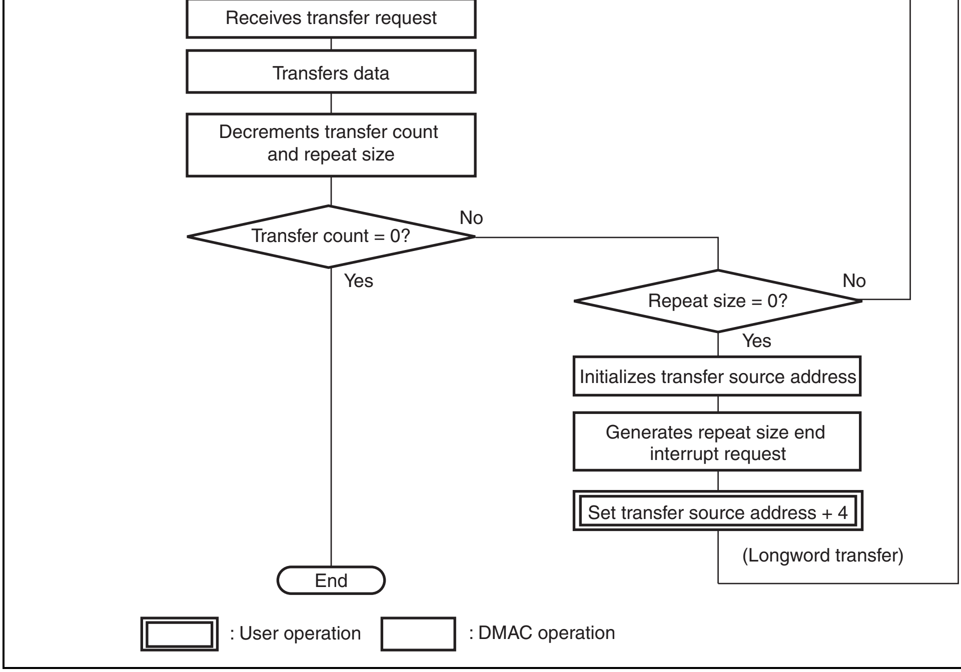

Receives transfer request
Transfers data
Decrements transfer count and repeat size
Transfer count = 0?
No
Yes
Repeat size = 0?
No
Yes
Initializes transfer source address
Generates repeat size end interrupt request
Set transfer source address + 4
(Longword transfer)
End
: User operation
: DMAC operation

Figure 7.20 XY Conversion Flowchart Using Offset Addition in Repeat Transfe

7.5.7 Register during DMA Transfer

The DMAC registers are updated by a DMA transfer. The value to be updated differs acc
the other settings and transfer state. The registers to be updated are DSAR, DDAR, DTC
BKSZH and BKSZ in DBSR, and the DTE, ACT, ERRF, ESIF, and DTIF bits in DMDR

(1) DMA Source Address Register

When the transfer source address set in DSAR is accessed, the contents of DSAR are out
then are updated to the next address.

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 w
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 acce

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. W
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 bou
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 ac
for example, +1 or +2 for byte or word data. After one word or one longword of data is re
address when the read cycle is started is incremented or decremented by the value accord
bits SAT1 and SAT0.

(2) DMA Destination Address Register

When the transfer destination address set in DDAR is accessed, the contents of DDAR a
and then are updated to the next address.

The increment or decrement can be specified by bits DAT1 and DAT0 in DACR. When
and DAT0 = B'00, the address is fixed. When DAT1 and DAT0 = B'01, the address is ad
the offset. When DAT1 and DAT0 = B'10, the address is incremented. When DAT1 and
B'11, the address is decremented. The incrementing or decrementing size depends on the
access size.

The data access size is specified by bits DTSZ1 and DTSZ0 in DMDR. When DTSZ1 a
= 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
decremented by 2. When DTSZ1 and DTSZ0 = B'10, the data access size is longword a
address is incremented or decremented by 4. Even if the access data size of the destinati
is word or longword, when the destination address is not aligned with the word or longw
boundary, the write bus cycle is divided into byte and word cycles. While one word or o
longword of data is being written, the incrementing or decrementing size is changing ac
the actual data access size, for example, +1 or +2 for byte or word data. After the one w
longword of data is written, the address when the write cycle is started is incremented o
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 compl
the block or repeat area is specified to the destination address side, the destination addre
to the transfer start address and is not affected by the address update.

transferred, DTCR is decremented by 1. When word data is transferred, DTCR is decrem
2. When longword data is transferred, DTCR is decremented by 4. However, when DTCR
contents of DTCR are not changed since the number of transfers is not counted.

While data is being transferred, all the bits of DTCR may be changed. DTCR must be acc
longwords. If the upper word and lower word are read separately, incorrect data may be r
since the contents of DTCR during the transfer may be updated regardless of the access b
CPU. Moreover, DTCR for the channel being transferred must not be written to.

When a conflict occurs between the address update by DMA transfer and write access by
the CPU has priority. When a conflict occurs between change from 1, 2, or 4 to 0 in DTC
write access by the CPU (other than 0), the CPU has priority in writing to DTCR. Howev
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 BK
bits 15 to 0 in DBSR function as BKSZ. The BKSZH bits (16 bits) store the block size an
size and its value is not changed. The BKSZ bits (16 bits) function as a counter for the bl
and repeat size and its value is decremented every transfer by 1. When the BKSZ value is
change from 1 to 0 by a DMA transfer, 0 is not stored but the BKSZH value is loaded int
BKSZ bits.

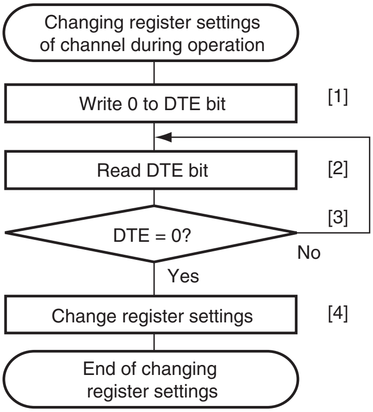
Since the upper 16 bits of DBSR are not updated, DBSR can be accessed in words.

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 pr (except for the DTE bit). When changing the register settings after writing 0 to the DTE 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.


Changing register settings of channel during operation
Write 0 to DTE bit
[1]
Read DTE bit
[2]
DTE = 0?
[3]
No
Yes
Change register settings
[4]
End of changing register settings

[1] Write 0 to the DTE bit in DMDR.

[2] Read the DTE bit.

[3] Confirm that DTE = 0. DTE = 1 indicates that DMA is transferring.

[4] Write the desired values to the registers.

Figure 7.21 Procedure for Changing Register Setting For Channel being Tran

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 D
transfer.

(7) ERRF Bit in DMDR

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

(8) ESIF Bit in DMDR

When an interrupt by an transfer size error, a repeat size end, or an extended repeat area o
is requested, the ESIF bit in DMDR is set to 1. When both the ESIF and ESIE bits are set
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 after
cycle of the interrupt source is completed.

The ESIF bit is automatically cleared to 0 and a transfer request is cleared if the transfer i
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	Priority
Channel 0	High
Channel 1	
Channel 2	
Channel 3	Low

The channel having highest priority other than the channel being transferred is selected v transfer is requested from other channels. The selected channel starts the transfer after th 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.

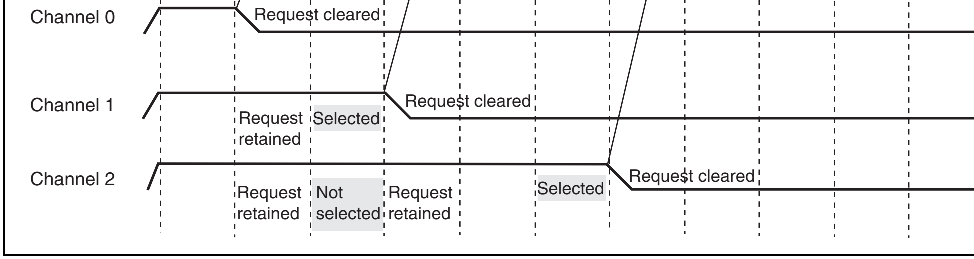


Figure 7.22 Example of Timing for Channel Priority

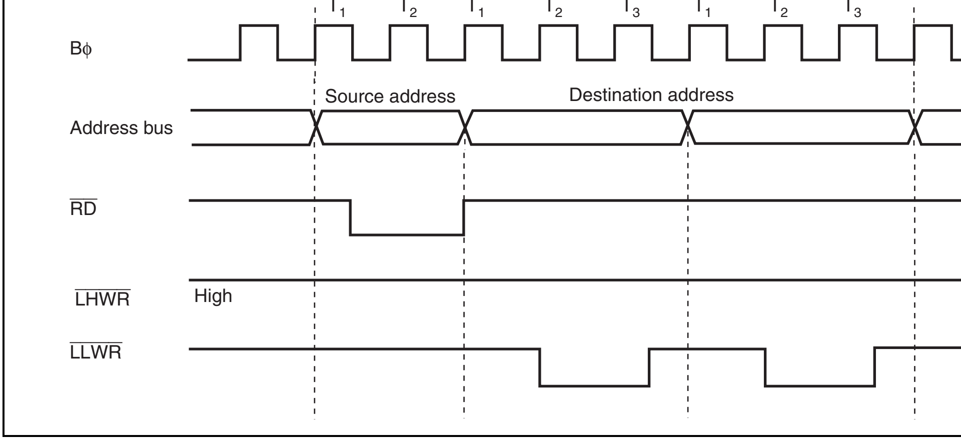


Figure 7.23 Example of Bus Timing of DMA Transfer

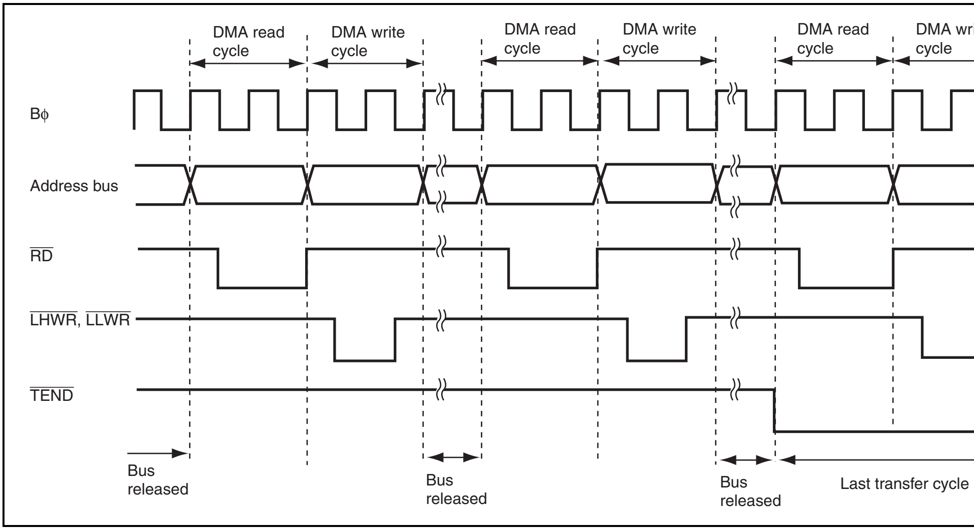


Figure 7.24 Example of Transfer in Normal Transfer Mode by Cycle Stealin

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 tr destination (DDAR) is not aligned with a longword boundary.

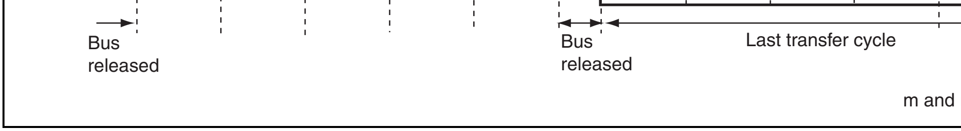


Figure 7.25 Example of Transfer in Normal Transfer Mode by Cycle Steal
(Transfer Source DSAR = Odd Address and Source Address Increment)

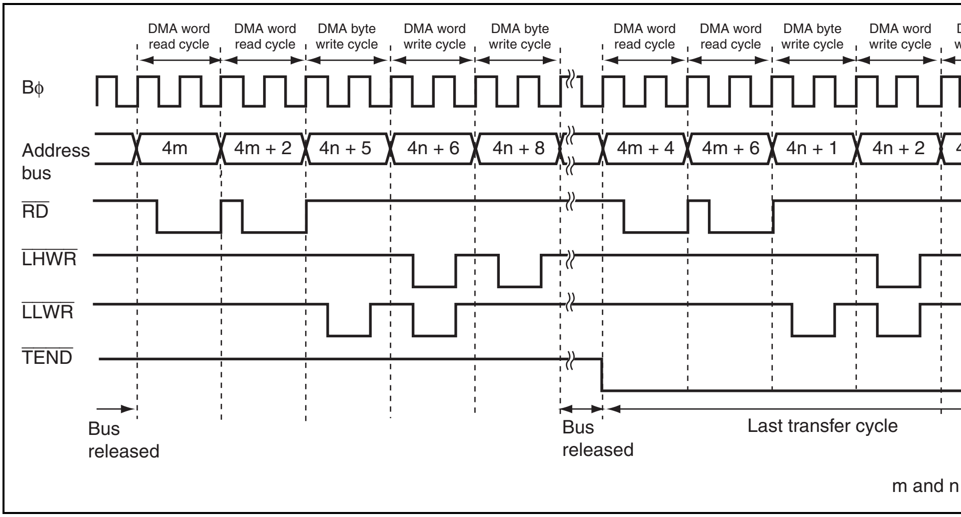


Figure 7.26 Example of Transfer in Normal Transfer Mode by Cycle Steal
(Transfer Destination DDAR = Odd Address and Destination Address Decren

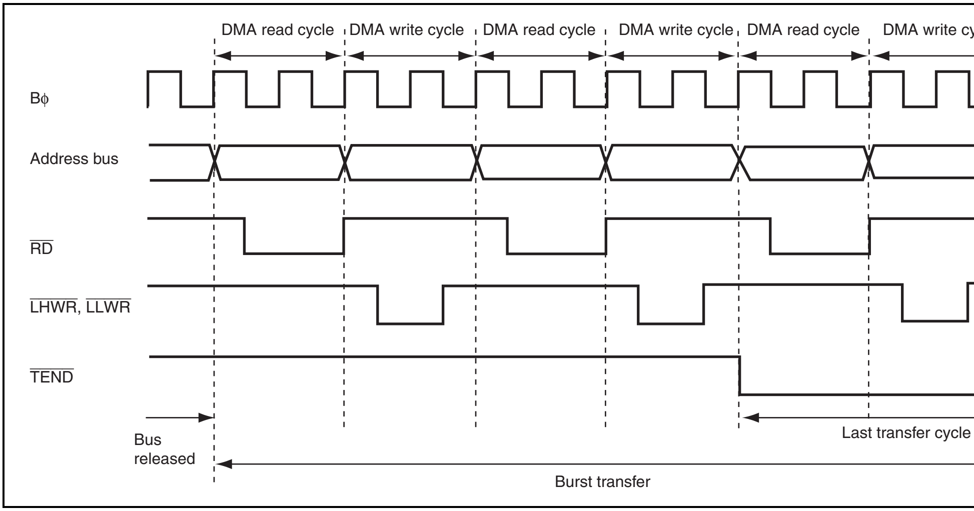


Figure 7.27 Example of Transfer in Normal Transfer Mode by Burst Acces

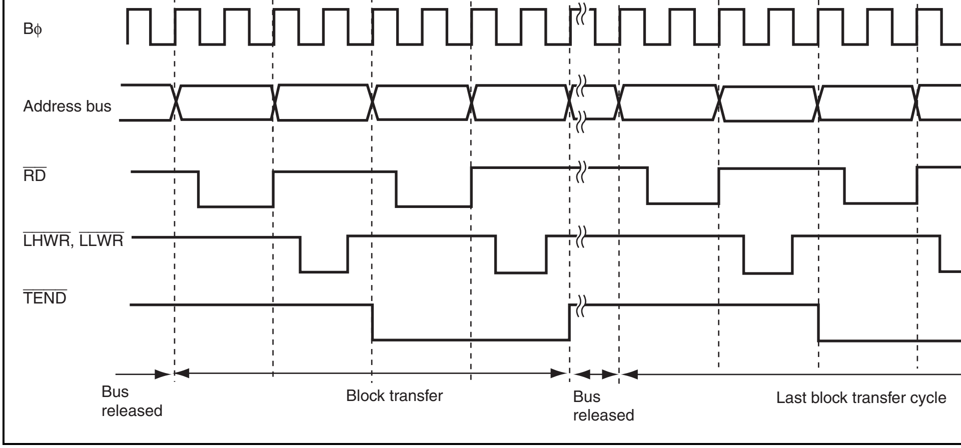
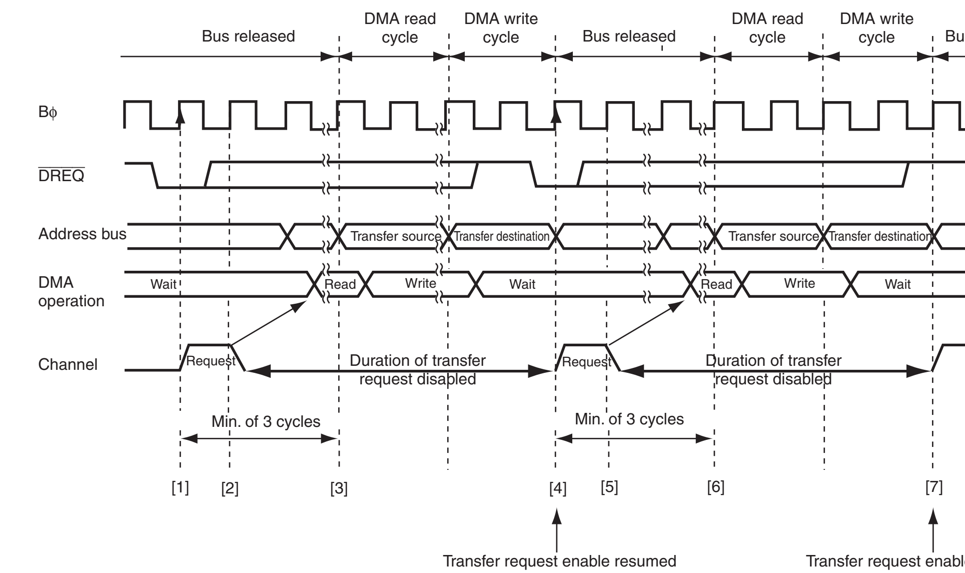


Figure 7.28 Example of Transfer in Block Transfer Mode

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



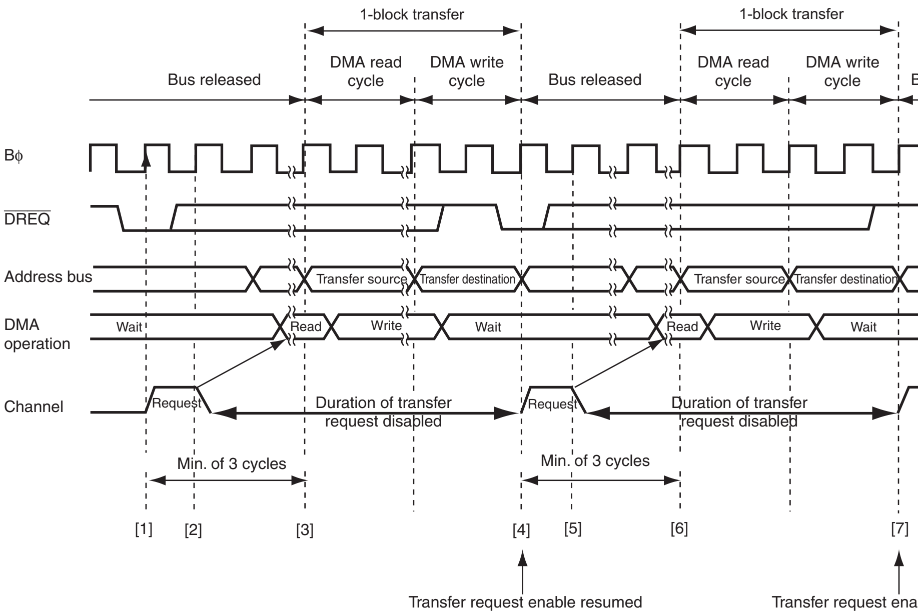
[1] After DMA transfer request is enabled, a low level of the $\overline{DREQ}$ signal is detected at the rising edge of the Bφ signal and a
request is held.

[2][5] The DMAC is activated and the transfer request is cleared.

[3][6] A DMA cycle is started and sampling the $\overline{DREQ}$ signal at the rising edge of the Bφ signal is started to detect a high level o
$\overline{DREQ}$ signal.

[4][7] When a high level of the $\overline{DREQ}$ signal has been detected, transfer request enable is resumed after completion of the write
(A low level of the $\overline{DREQ}$ 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 $\overline{DREQ}$ 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.

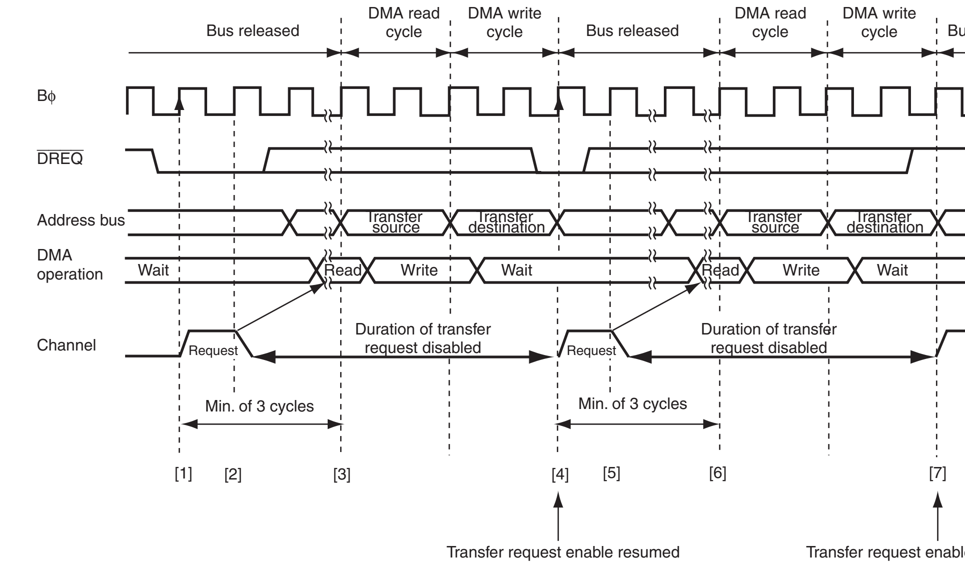
[2][5] The DMAC is activated and the transfer request is cleared.

[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.



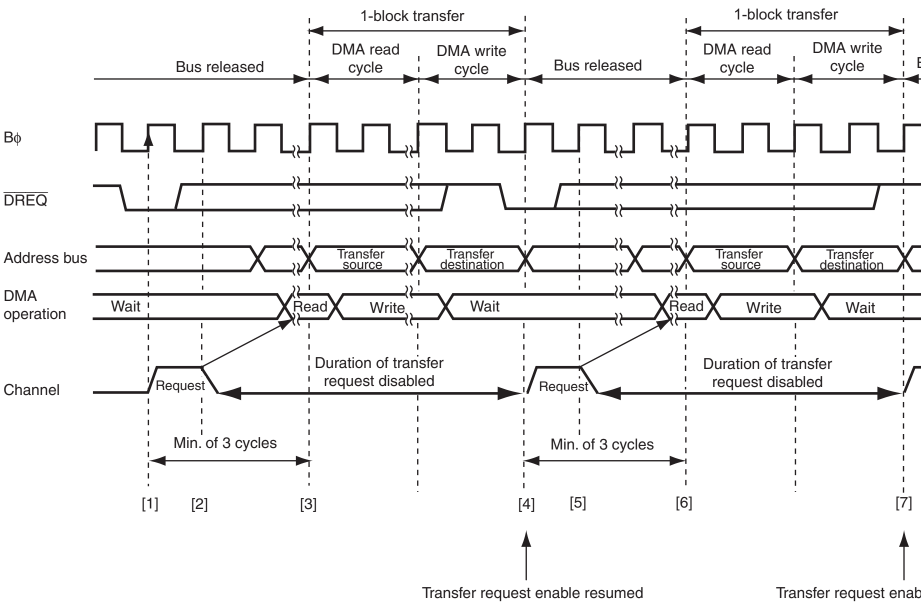
[1] After DMA transfer request is enabled, a low level of the $\overline{\text{DREQ}}$ signal is detected at the rising edge of the Bφ signal and a request is held.

[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 a tran request is held.

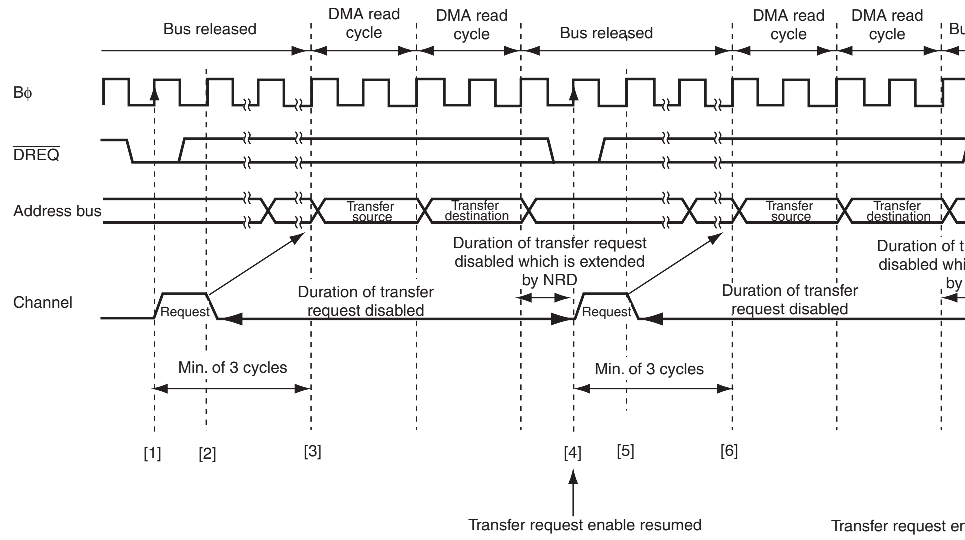
[2][5] The DMAC is activated and the transfer request is cleared.

[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 same a

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 wr
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

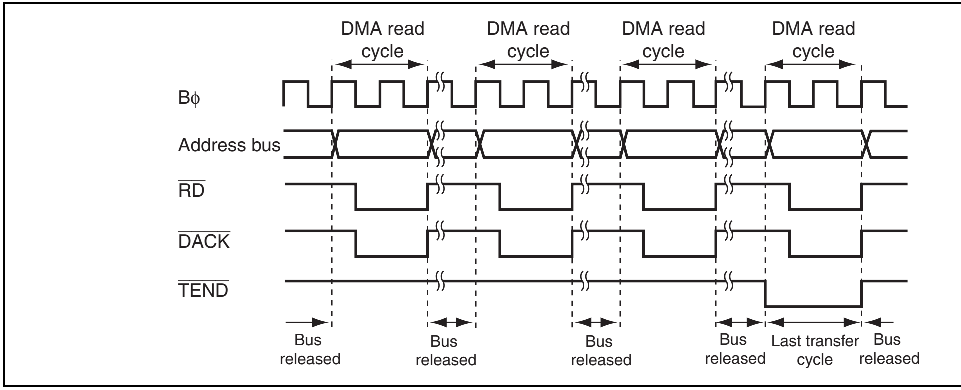


Figure 7.34   Example of Transfer in Single Address Mode (Byte Read)

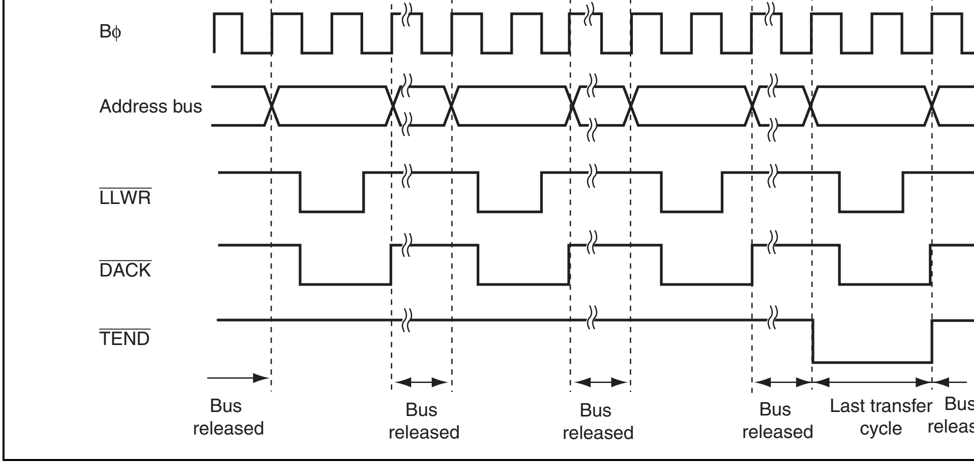


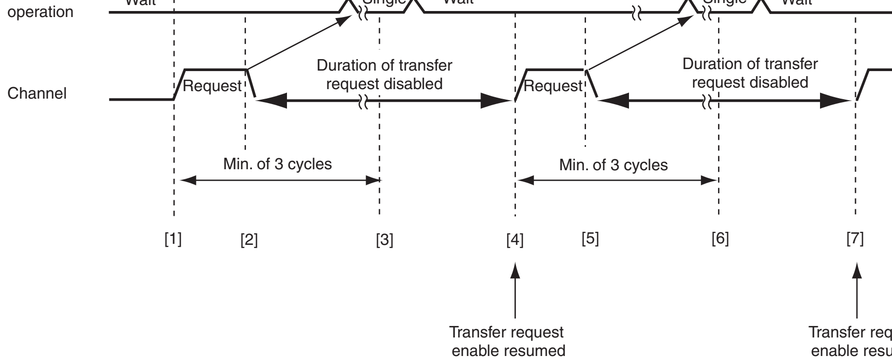
Figure 7.35 Example of Transfer in Single Address Mode (Byte Write)

(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.

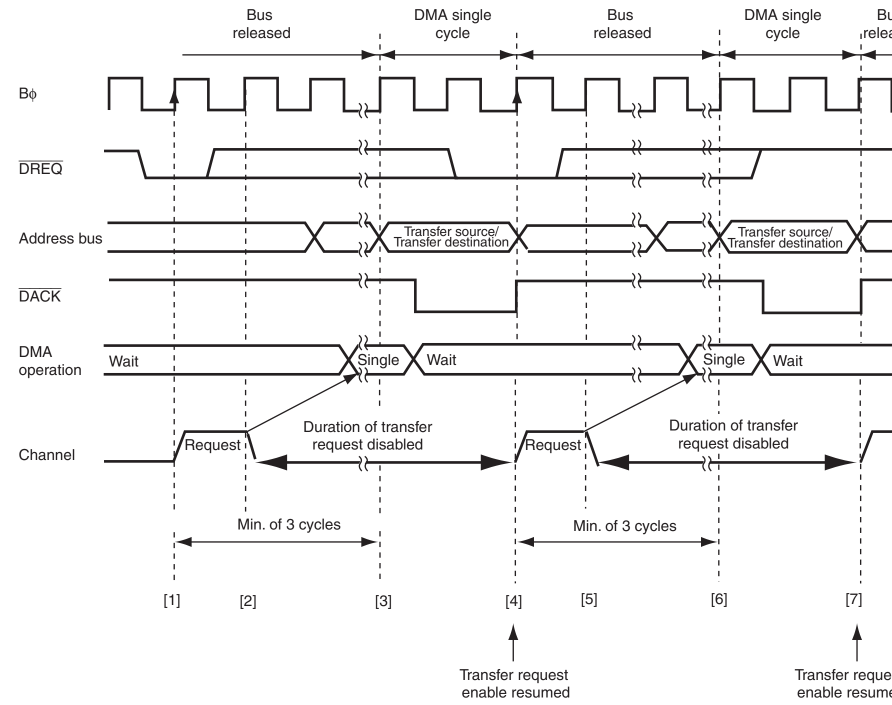
[2][5] The DMAC is activated and the transfer request is cleared.

[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.


Bus released
DMA single cycle
Bus released
DMA single cycle
Bu relea
Bϕ
DREQ
Address bus
Transfer source/ Transfer destination
Transfer source/ Transfer destination
DACK
DMA operation
Wait
Single
Wait
Single
Wait
Channel
Request
Duration of transfer request disabled
Request
Duration of transfer request disabled
Min. of 3 cycles
Min. of 3 cycles
[1]
[2]
[3]
[4]
[5]
[6]
[7]
Transfer request enable resumed
Transfer reque enable resume

[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 transf request is held.

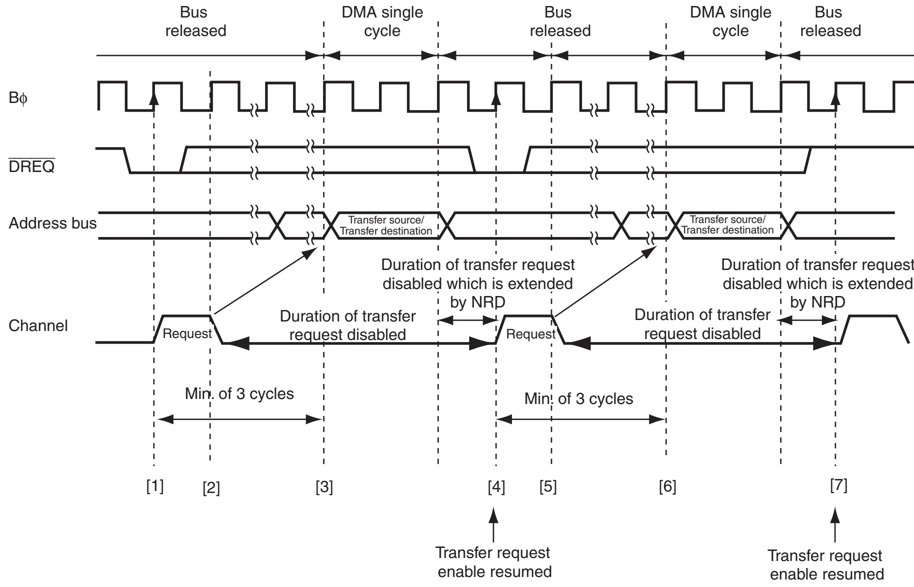
[2][5] The DMAC is activated and the transfer request is cleared.

[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 $\overline{\text{DREQ}}$ signal is detected at the rising edge of the Bϕ signal and a transfer request is held. This is the same as

Figure 7.37 Example of Transfer in Single Address Mode Activated by $\overline{\text{DREQ}}$ 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.


Bus released
DMA single cycle
Bus released
DMA single cycle
Bus released
Bφ
DREQ
Address bus
Transfer source/ Transfer destination
Transfer source/ Transfer destination
Channel
Request
Request
Duration of transfer request disabled
Duration of transfer request disabled
Duration of transfer request disabled which is extended by NRD
Duration of transfer request disabled which is extended by NRD
Min. of 3 cycles
Min. of 3 cycles
[1]
[2]
[3]
[4]
[5]
[6]
[7]
Transfer request enable resumed
Transfer request enable resumed

[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 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 sa

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 ESIF bit in DMDR is set to 1. Under this condition, setting the DTE bit to 1 resumes the

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 th
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 a
ERRF bit in DMDR_0 is set to 1. When an NMI interrupt is requested during a DMA tr
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 priority
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 stopp 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 chanr
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 wh
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
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
transfer by clearing an interrupt.

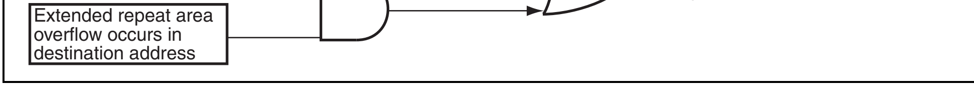
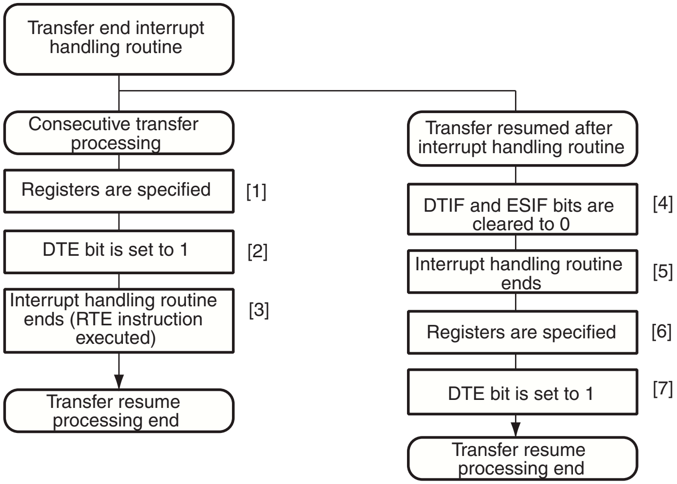


Figure 7.39 Interrupt and Interrupt Sources


Transfer end interrupt handling routine
Consecutive transfer processing
Registers are specified
[1]
DTE bit is set to 1
[2]
Interrupt handling routine ends (RTE instruction executed)
[3]
Transfer resume processing end
Transfer resumed after interrupt handling routine
DTIF and ESIF bits are cleared to 0
[4]
Interrupt handling routine ends
[5]
Registers are specified
[6]
DTE bit is set to 1
[7]
Transfer resume processing end

[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 operation

   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

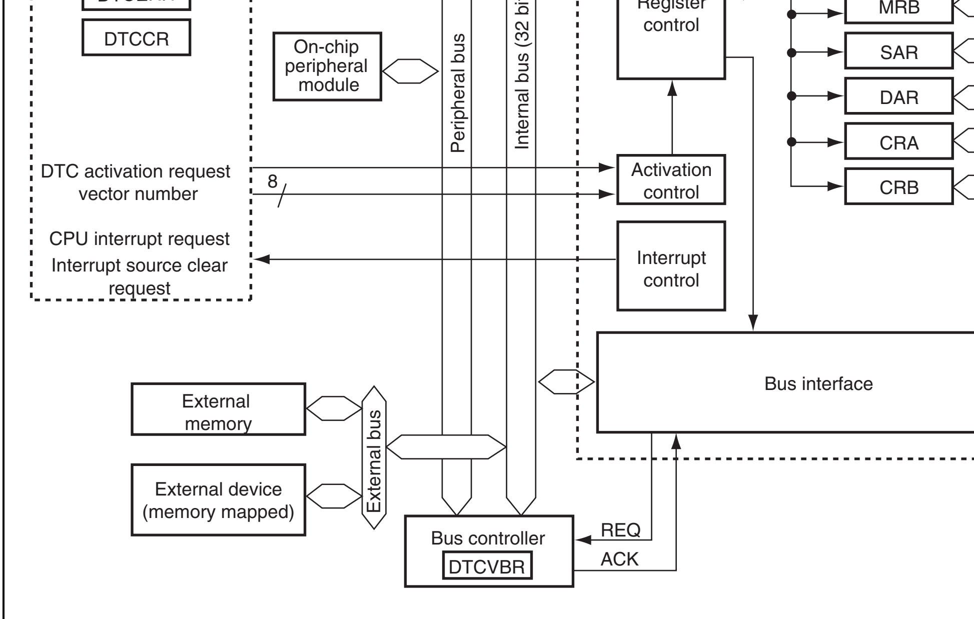
  Transfer source and destination addresses can be selected from increment/decrement
- Short address mode or full address mode selectable
  - Short address mode

    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 mode specifiable



[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
R/W							

Bit	Bit Name	Initial Value	R/W	Description
7	CHNE	Undefined	—	DTC Chain Transfer Enable Specifies the chain transfer. For details, see 8.5 Transfer. The chain transfer condition is selecte 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 follo 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
5	DISEL	Undefined	—	DTC Interrupt Select When this bit is set to 1, a CPU interrupt reques generated every time after a data transfer ends this bit is set to 0, a CPU interrupt request is on generated when the specified number of data tr ends.

				(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 are valid and bits 31 to 24 are ignored. At this time, the upper eight bits are filled with th 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.

eight bits (CRAL). CRAH holds the number of transfers while CRAL functions as an 8-b transfer counter (1 to 256). CRAL is decremented by 1 every time data is transferred, and contents of CRAH are sent to CRAL when the count reaches H'00. The transfer count is 1 CRAH = CRAL = H'01, 255 when CRAH = CRAL = H'FF, and 256 when CRAH = CRA H'00.

In block transfer mode, CRA is divided into two parts: the upper eight bits (CRAH) and t eight bits (CRAL). CRAH holds the block size while CRAL functions as an 8-bit block-s counter (1 to 256 for byte, word, or longword). CRAL is decremented by 1 every time a b (word or longword) data is transferred, and the contents of CRAH are sent to CRAL whe 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 writte CRAL when CRAL is 0. Accordingly, chain trans not occur when CRAL is 0. If this bit is set to 1, th 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

[Clearing condition]

- When writing 0 after reading 1

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
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

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.

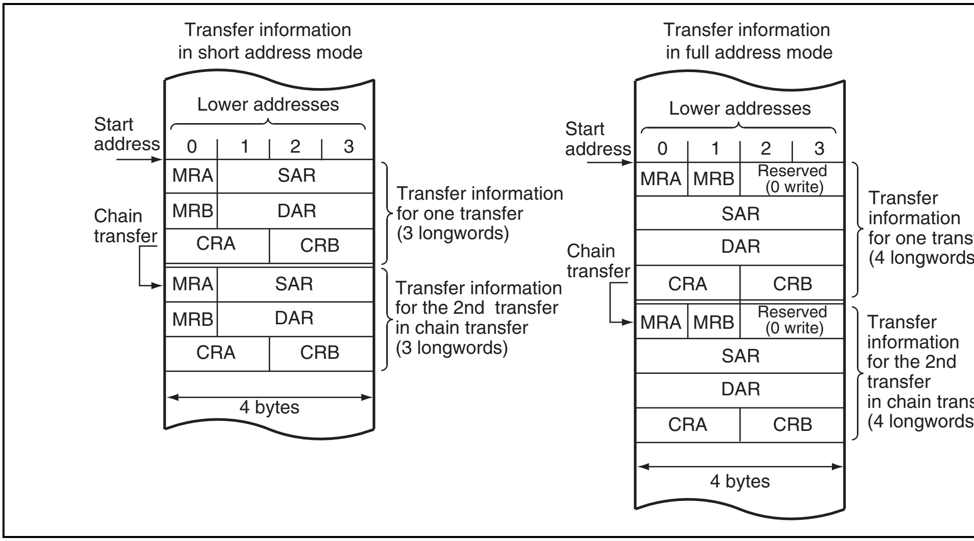

Transfer information in short address mode
Lower addresses
Start address
0
1
2
3
MRA
SAR
MRB
DAR
CRA
CRB
Chain transfer
MRA
SAR
MRB
DAR
CRA
CRB
4 bytes
Transfer information for one transfer (3 longwords)
Transfer information for the 2nd transfer in chain transfer (3 longwords)
Transfer information in full address mode
Lower addresses
Start address
0
1
2
3
MRA
MRB
Reserved (0 write)
SAR
DAR
CRA
CRB
Chain transfer
MRA
MRB
Reserved (0 write)
SAR
DAR
CRA
CRB
4 bytes
Transfer information for one transf (4 longwords
Transfer information for the 2nd transfer in chain trans (4 longwords

Figure 8.2 Transfer Information on Data Area

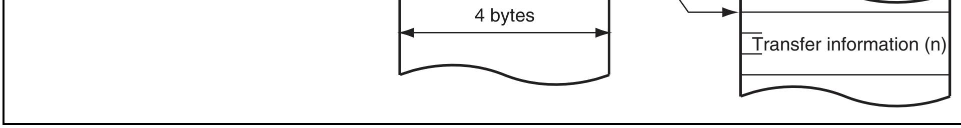


Figure 8.3 Correspondence between DTC Vector Address and Transfer Infor

	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_3	RXI3	157	H'674	DTCEE15	
	TXI3	158	H'678	DTCEE14	
SCI_4	RXI4	161	H'684	DTCEE13	
	TXI4	162	H'688	DTCEE12	L

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).

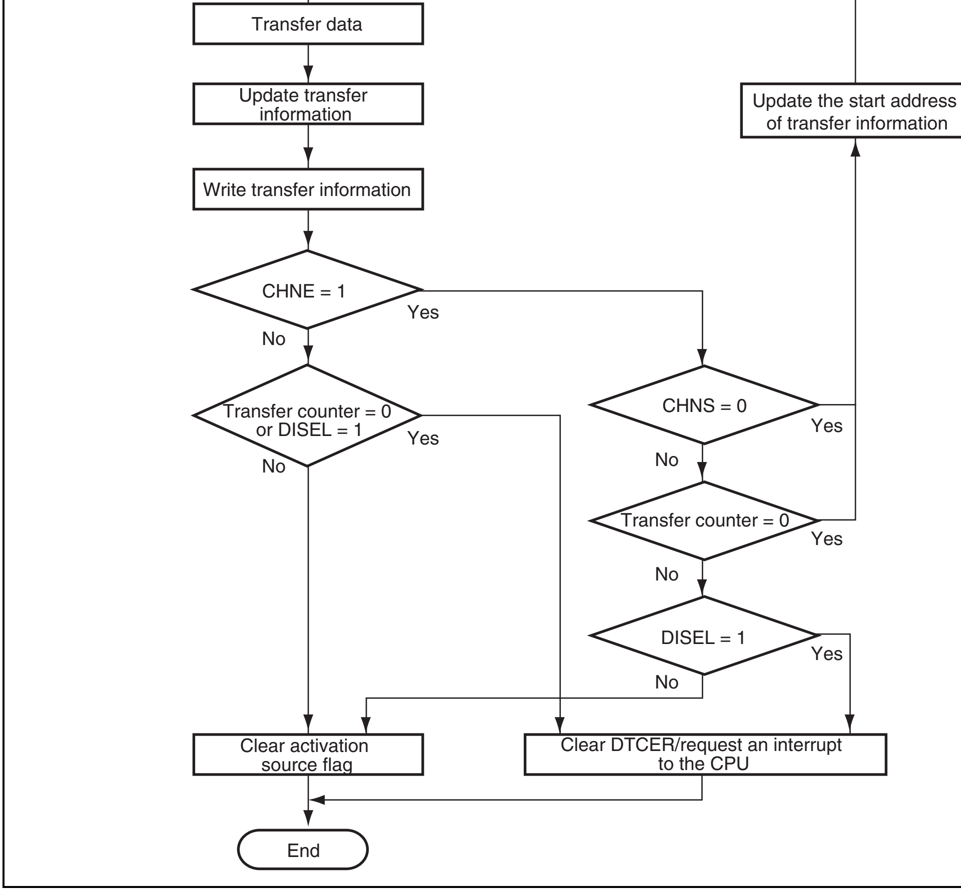

Transfer data
Update transfer information
Write transfer information
CHNE = 1
Yes
No
Transfer counter = 0 or DISEL = 1
Yes
No
CHNS = 0
Yes
No
Transfer counter = 0
Yes
No
DISEL = 1
Yes
No
Update the start address of transfer information
Clear activation source flag
Clear DTCER/request an interrupt to the CPU
End

Figure 8.4 Flowchart of DTC Operation

1	1	0	Not 0	—	—		—	Ends at 1st transf
1	1	—	0*[2]	0	—	0	Not 0	Ends at 2nd trans
				0	—	0	0*[2]	Ends at 2nd trans
				0	—	1		Interrupt request
1	1	1	Not 0	—	—	—	—	Ends at 1st transf
								Interrupt request

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

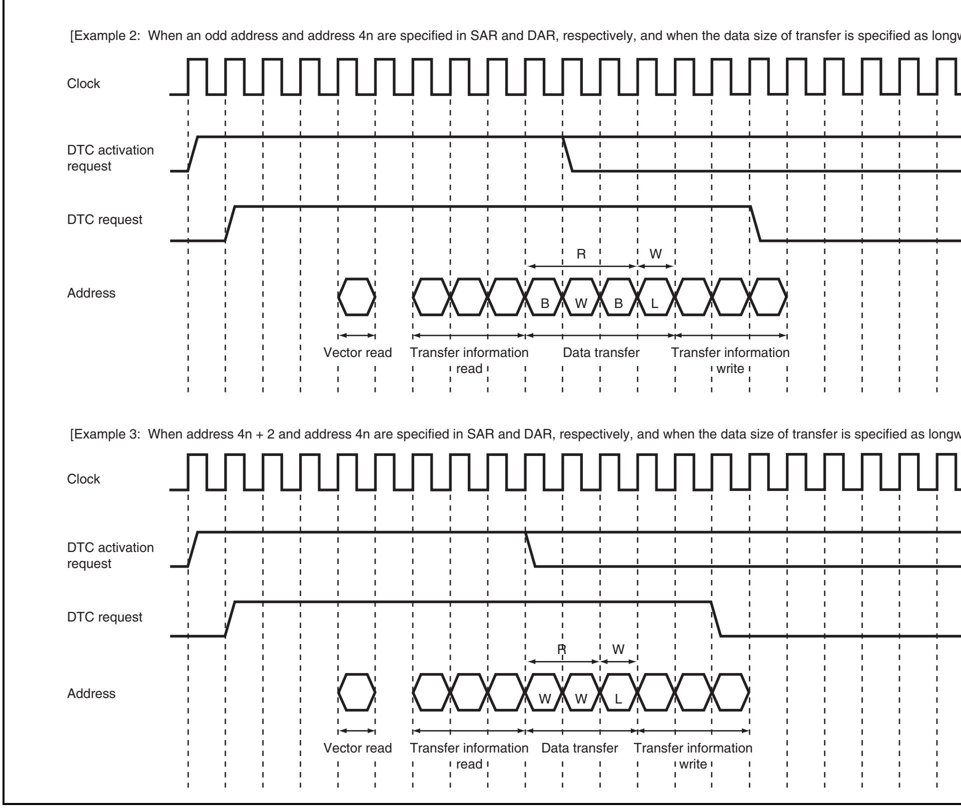
8.5.1 Bus Cycle Division

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 divisio 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 (L**
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)

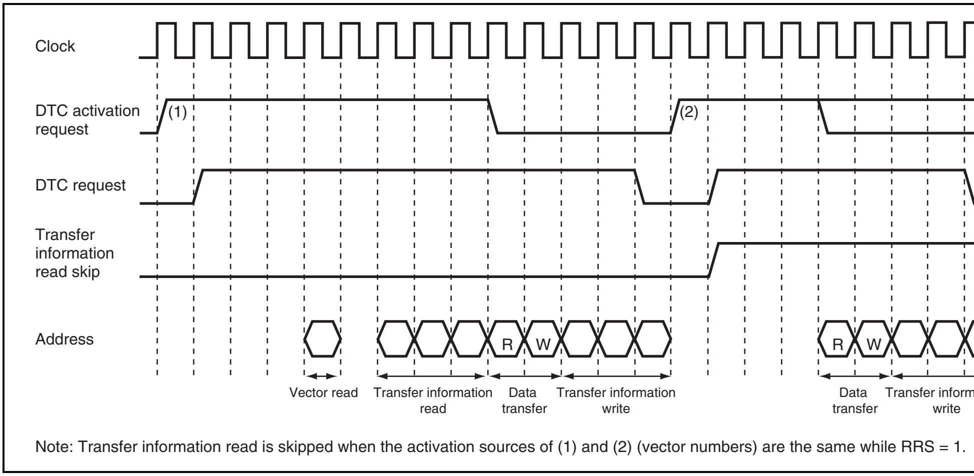


[Example 2: When an odd address and address 4n are specified in SAR and DAR, respectively, and when the data size of transfer is specified as longw

[Example 3: When address 4n + 2 and address 4n are specified in SAR and DAR, respectively, and when the data size of transfer is specified as longw

Figure 8.5 Bus Cycle Division Example

cleared to 0, the stored vector number is deleted, and the updated vector table and transfer 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 address
specified as incremented, decremented, or fixed. When the specified number of transfer
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.

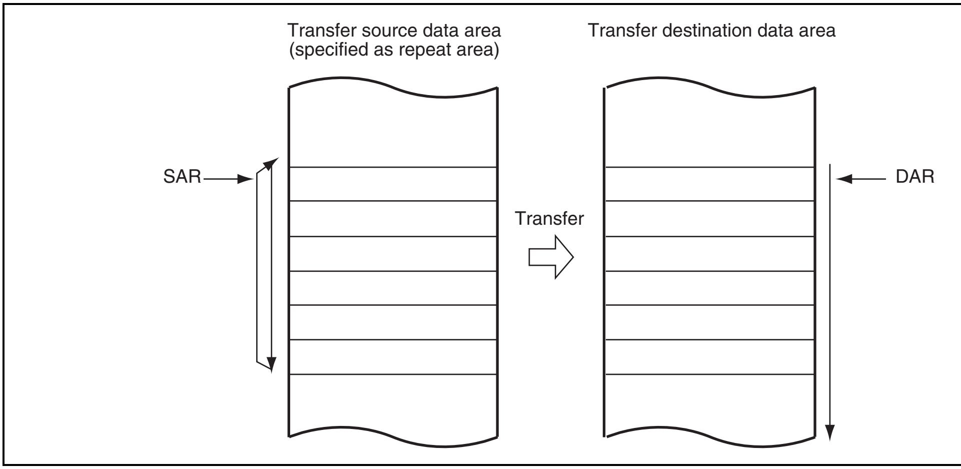

Transfer source data area
(specified as repeat area)
Transfer destination data area
SAR
Transfer
DAR

Figure 8.8 Memory Map in Repeat Transfer Mode (When Transfer Source is Specified as Repeat Area)

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.

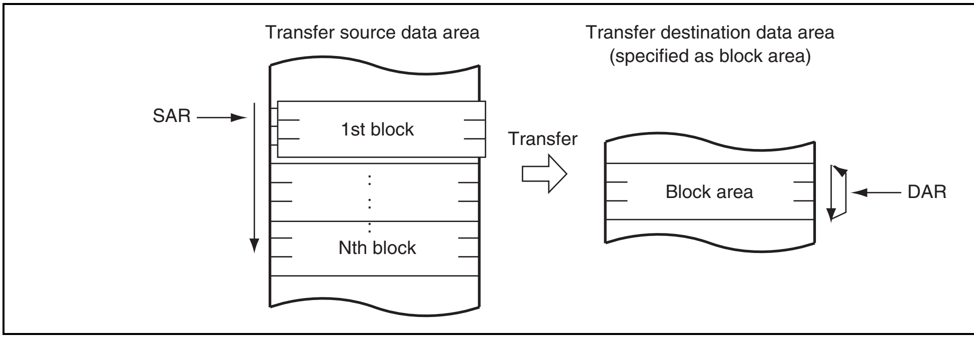

Transfer source data area
Transfer destination data area
(specified as block area)
SAR
1st block
Transfer
Block area
DAR
Nth block

Figure 8.9 Memory Map in Block Transfer Mode
(When Transfer Destination is Specified as Block Area)

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.

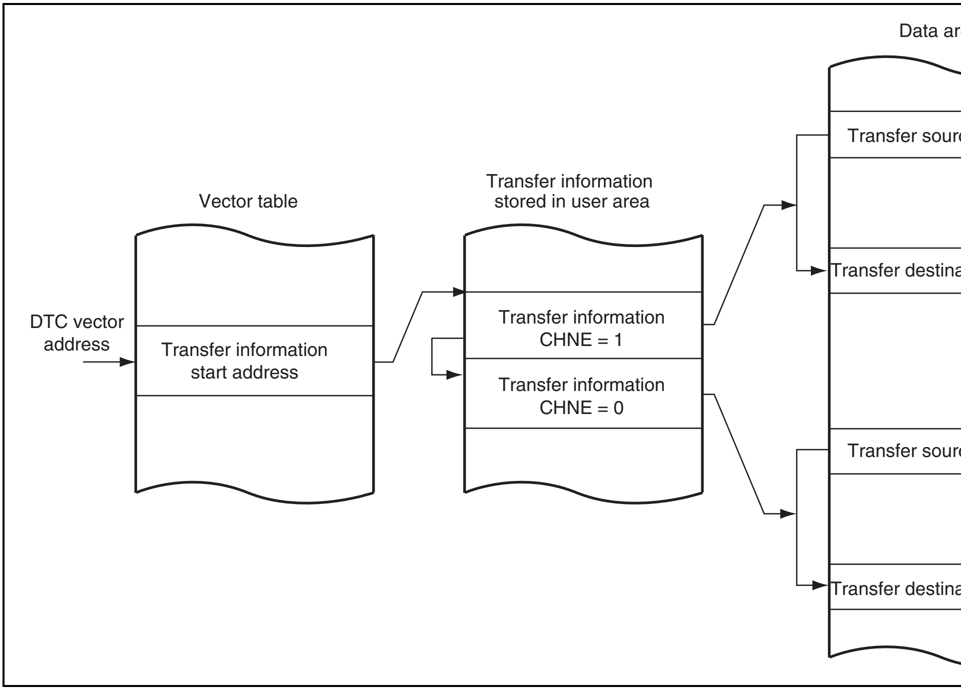

Vector table
Transfer information stored in user area
Data ar
DTC vector address
Transfer information start address
Transfer information CHNE = 1
Transfer information CHNE = 0
Transfer sourc
Transfer destina
Transfer sourc
Transfer destina

Figure 8.10 Operation of Chain Transfer

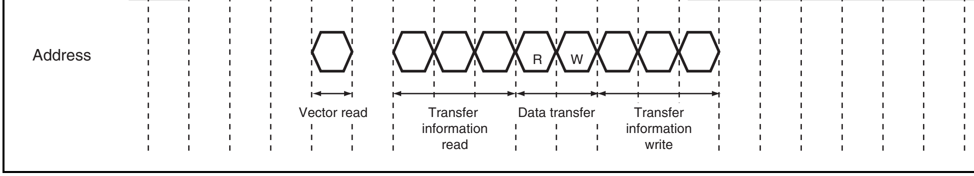


Figure 8.11 DTC Operation Timing
(Example of Short Address Mode in Normal Transfer Mode or Repeat Transfer

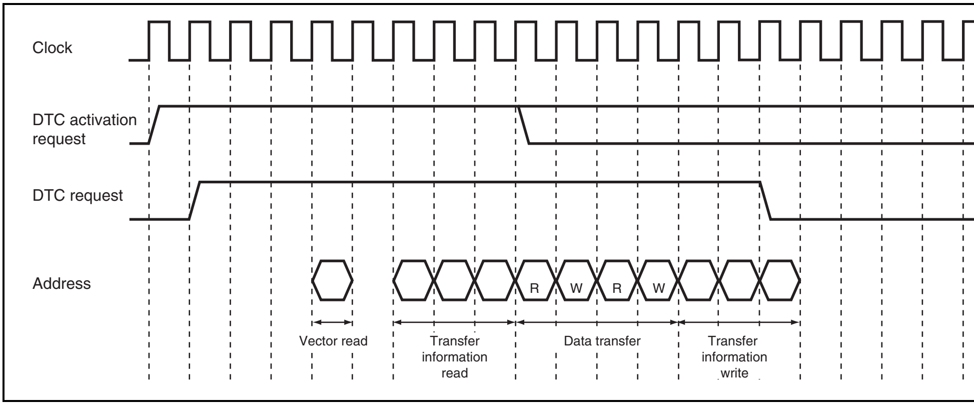


Figure 8.12 DTC Operation Timing
(Example of Short Address Mode in Block Transfer Mode with Block Size of

Figure 8.13 DTC Operation Timing (Example of Short Address Mode in Chain T

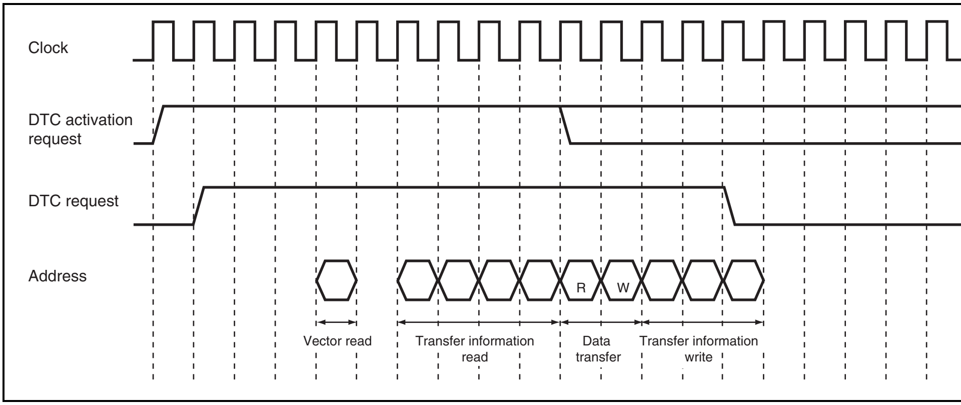


Figure 8.14 DTC Operation Timing
(Example of Full Address Mode in Normal Transfer Mode or Repeat Transfer

Normal	1	0*	4*	3*	0*	3*	2*	1*	3*	2*	1	3*	2*	1	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	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	1

[Legend]

P: Block size (CRAH and CRAL value)

Notes: 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 addres register

7. When a word is transferred while an odd address is specified in the address re 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 mean
of all transfers activated by one activation event (the number in which the CHNE bit is s
plus 1).

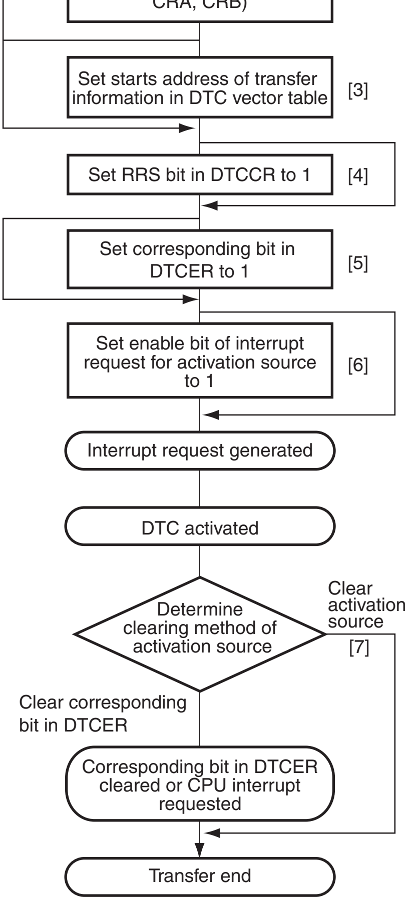
$$\text{Number of execution cycles} = I \bullet S_I + \Sigma\,(J \bullet S_J + K \bullet S_K + L \bullet S_L + M \bullet S_M) + N \bullet$$

8.5.10 DTC Bus Release Timing

The DTC requests the bus mastership to the bus arbiter when an activation request occu
DTC releases the bus after a vector read, transfer information read, a single data transfe
transfer information writeback. The DTC does not release the bus during transfer inform
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
level specified by bits CPUP2 to CPUP0 in CPUPCR and the DTC priority level specifi
DTCP2 to DTCP0. For details, see section 5, Interrupt Controller.


CRA, CRB)
Set starts address of transfer information in DTC vector table
[3]
Set RRS bit in DTCCR to 1
[4]
Set corresponding bit in DTCER to 1
[5]
Set enable bit of interrupt request for activation source to 1
[6]
Interrupt request generated
DTC activated
Determine clearing method of activation source
Clear activation source
[7]
Clear corresponding bit in DTCER
Corresponding bit in DTCER cleared or CPU interrupt requested
Transfer end

information in the data area. For details on setting trans
information, see section 8.2, Register Descriptions. For
on location of transfer information, see section 8.4, Loc
Transfer Information and DTC Vector Table.

[3] Set the start address of the transfer information in the D
vector table. For details on setting DTC vector table, se
8.4, Location of Transfer Information and DTC Vector Ta

[4] Setting the RRS bit to 1 performs a read skip of second
later transfer information when the DTC is activated co
tively by the same interrupt source. Setting the RRS bit
always allowed. However, the value set during transfer
valid from the next transfer.

[5] Set the bit in DTCER corresponding to the DTC activat
interrupt source to 1. For the correspondence of interru
DTCER, refer to table 8.1. The bit in DTCER may be se
the second or later transfer. In this case, setting the bit
needed.

[6] Set the enable bits for the interrupt sources to be used
activation sources to 1. The DTC is activated when an i
used as an activation source is generated. For details o
settings of the interrupt enable bits, see the correspond
descriptions of the corresponding module.

[7] After the end of one data transfer, the DTC clears the a
source flag or clears the corresponding bit in DTCER a
requests an interrupt to the CPU. The operation after tr
depends on the transfer information. For details, see se
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

2. Set the start address of the transfer information for an RXI interrupt at the DTC vect

3. Set the corresponding bit in DTCER to 1.

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

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

6. When CRA becomes 0 after the 128 data transfers have ended, the RDRF flag is hel DTCE bit is cleared to 0, and an RXI interrupt request is sent to the CPU. Terminati 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 (DTS
MRB to chain transfer mode (CHNE = 1, CHNS = 0, DISEL = 0). Set the data table s
address in SAR, the NDRH address in DAR, and the data table size in CRAH and CR
CRB can be set to any value.

2. Perform settings for transfer to the TPU's TGR. Set MRA to source address incremen
(SM1 = 1, SM0 = 0), fixed destination address (DM1 = DM0 = 0), normal mode (MD
= 0), and word size (Sz1 = 0, Sz0 = 1). Set the data table start address in SAR, the TG
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 the
interrupt with TIER.
7. Set the initial output value in PODR, and the next output value in NDR. Set bits in DI
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 NDI
set value of the next output trigger period is transferred to TGRA. The activation sour
flag is cleared.
10. When the specified number of transfers are completed (the TPU transfer CRA value i
TGFA flag is held at 1, the DTCE bit is cleared to 0, and a TGIA interrupt request is s
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 t data transfer in a separate area (in ROM, etc.). For example, if the input buffer is co 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 a setting the transfer destination address for the first data transfer. Use the upper eight DAR in the first transfer information area as the transfer destination. Set CHNE = D If the above input buffer is specified as H'200000 to H'21FFFF, set the transfer coun

4. Execute the first data transfer 65536 times by means of interrupts. When the transfer for the first data transfer reaches 0, the second data transfer is started. Set the upper e of the transfer source address for the first data transfer to H'21. The lower 16 bits of transfer destination address of the first data transfer and the transfer counter are H'00

5. Next, execute the first data transfer the 65536 times specified for the first data transf means of interrupts. When the transfer counter for the first data transfer reaches 0, th data transfer is started. Set the upper eight bits of the transfer source address for the f transfer to H'20. The lower 16 bits of the transfer destination address of the first data and the transfer counter are H'0000.

6. Steps 4 and 5 are repeated endlessly. As repeat mode is specified for the second data no interrupt request is sent to the CPU.

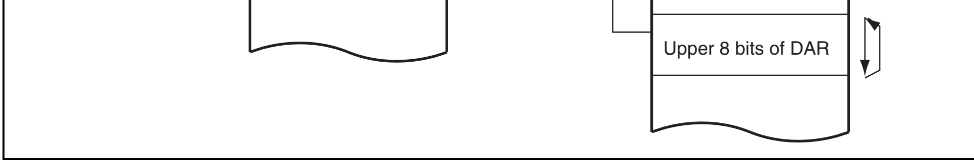


Figure 8.16 Chain Transfer when Counter = 0

8.8 Interrupt Sources

An interrupt request is issued to the CPU when the DTC finishes the specified number of transfers or a data transfer for which the DISEL bit was set to 1. In the case of interrupt ac the interrupt set as the activation source is generated. These interrupts to the CPU are sub 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 tran
counter value and DISEL bit setting. Accordingly, even if the DTC transfer counter valu
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
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 w
last of the chain of data transfers is executed. At this time, SCI and A/D converter
interrupt/activation sources, are cleared when the DTC reads or writes to the relevant reg

Therefore, when the DTC is activated by an interrupt or activation source, if a read/writ
relevant register is not included in the last chained data transfer, the interrupt or activatio
will be retained.

When IBCCS = 1 and the DMAC is used, clear the IBCCS bit to 0 and then set to 1 agai
modifying the DTC transfer information in the CPU exception handling routine initiated
transfer end interrupt.

8.9.8 Endian

The DTC supports the big-endian and little-endian format. However, use the same endian
for writing and reading the transfer information.

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 TPU inputs			$\overline{\text{IRQ5}}$-A/ TCLKB-B		TCLKB-B
	4	P14	$\overline{\text{DREQ1}}$-A/ $\overline{\text{IRQ4}}$-A/ TCLKA-B	TxD3	$\overline{\text{IRQ4}}$-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

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	P33/ TIOCD0/ TIOCC0/ TCKB-A	
		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	P30/ TIOCA0	
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	TMCI2/ $\overline{\text{IRQ8}}$-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	—	—	$\overline{\text{RD}}$		
		4	PA4	—	$\overline{\text{LHWR}}$/$\overline{\text{LUB}}$		
		3	—	—	$\overline{\text{LLWR}}$/$\overline{\text{LLB}}$		
		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	—	$\overline{\text{CS2}}$-A/ $\overline{\text{CS6}}$-A		
		1	PB1	—	$\overline{\text{CS1}}$/ $\overline{\text{CS2}}$-B/ $\overline{\text{CS5}}$-A/ $\overline{\text{CS6}}$-B/ $\overline{\text{CS7}}$-B		
		0	PB0	—	$\overline{\text{CS0}}$/$\overline{\text{CS4}}$-A/ $\overline{\text{CS5}}$-B		
Port D	Address outputs	7	—	—	A7	—	O
		6	—	—	A6		
		5	—	—	A5		
		4	—	—	A4		
		3	—	—	A3		
		2	—	—	A2		
		1	—	—	A1		
		0	—	—	A0		
Port E	Address outputs	7	—	—	A15	—	O
		6	—	—	A14		
		5	—	—	A13		
		4	—	—	A12		
		3	—	—	A11		
		2	—	—	A10		
		1	—	—	A9		
		0	—	—	A8		

		3	—	—	A19		
		2	—	—	A18		
		1	—	—	A17		
		0	—	—	A16		
Port H	General I/O port also functioning as bi-directional data bus	7	D7*[2]	—	—	—	O
		6	D6*[2]	—	—		
		5	D5*[2]	—	—		
		4	D4*[2]	—	—		
		3	D3*[2]	—	—		
		2	D2*[2]	—	—		
		1	D1*[2]	—	—		
		0	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/

3. When enabling the $\overline{\text{CS}}$ output, turn the input pull-up MOS function off before e

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**
Port A	H'80
Other ports	H'00

9.1.2 Data Register (PnDR) (n = 1 to 3, 6, A, B, D to F, H, and I)

DR is an 8-bit readable/writable register that stores the output data of the pins to be usec
general output port.

The initial value of DR is H'00.

Bit	7	6	5	4	3	2	1
Bit Name	Pn7DR	Pn6DR	Pn5DR	Pn4DR	Pn3DR	Pn2DR	Pn1DR
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.

Bit Name	Pn7	Pn6	Pn5	Pn4	Pn3	Pn2	Pn1	
Initial Value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	U
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.

9.1.4 Input Buffer Control Register (PnICR) (n = 1 to 3, 5, 6, A, B, D to F, H, an

ICR is an 8-bit readable/writable register that controls the port input buffers.

For bits in ICR set to 1, the input buffers of the corresponding pins are valid. For bits in I cleared to 0, the input buffers of the corresponding pins are invalid and the input signals a high.

When the pin functions as an input for the peripheral modules, the corresponding bits sho set to 1. The initial value should be written to a bit whose corresponding pin is not used a or is used as an analog input/output pin.

If the bits in ICR have been cleared to 0, the pin state is not reflected to the peripheral mo

When PORT is read, the pin status is always read regardless of the ICR value.

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.

9.1.5 Pull-Up MOS Control Register (PnPCR) (n = D to F, H, and I)

PCR is an 8-bit readable/writable register that controls on/off of the port input pull-up M

If a bit in PCR is set to 1 while the pin is in input state, the input pull-up MOS correspon
the bit in PCR is turned on. Table 9.4 shows the input pull-up MOS status.

The initial value of PCR is H'00.

Bit	7	6	5	4	3	2	1
Bit Name	Pn7PCR	Pn6PCR	Pn5PCR	Pn4PCR	Pn3PCR	Pn2PCR	Pn1PCR
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

			Hardware	Software	O
Port	**Pin State**	**Reset**	**Standby Mode**	**Standby Mode**	**O**
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 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.

Note: * When enabling the $\overline{\text{CS}}$ output, clear PCR to 0 before enabling it.

9.1.6 Open-Drain Control Register (PnODR) (n = 2 and F)

ODR is an 8-bit readable/writable register that selects the open-drain output function.

If a bit in ODR is set to 1, the pin corresponding to that bit in ODR functions as an NMO drain output. If a bit in ODR is cleared to 0, the pin corresponding to that bit in ODR func a CMOS output.

The initial value of ODR is H'00.

Bit	7	6	5	4	3	2	1
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

9.2.1 Port 1

(1) P17/$\overline{\text{IRQ7}}$-A/TCLKD-B

The pin function is switched as shown below according to the P17DDR bit setting.

		Setting
		I/O Port
Module Name	Pin Function	P17DDR
I/O port	P17 output	1
	P17 input (initial setting)	0

(2) P16/SCK3/$\overline{\text{DACK1}}$-A/$\overline{\text{IRQ6}}$-A/TCLKC-B

The pin function is switched as shown below according to the combination of the DMA register settings and P16DDR bit setting.

		Setting		
		DMAC	SCI	I/O Port
Module Name	Pin Function	$\overline{\text{DACK1A}}$_OE	SCK3_OE	P16DDR
DMAC	$\overline{\text{DACK1}}$-A output	1	—	—
SCI	SCK3 output	0	1	—
I/O port	P16 output	0	0	1
	P16 input (initial setting)	0	0	0

	P15 input (initial setting)	0	0

(4) P14/TxD3/$\overline{\text{DREQ1}}$-A/$\overline{\text{IRQ4}}$-A/TCLKA-B

The pin function is switched as shown below according to the combination of the SCI reg setting and P14DDR bit setting.

		Setting	
		SCI	I/O Port
Module Name	**Pin Function**	**TxD3_OE**	**P14DDR**
SCI	TxD3 output	1	—
I/O port	P14 output	0	1
	P14 input (initial setting)	0	0

(5) P13/$\overline{\text{ADTRG0}}$/$\overline{\text{IRQ3}}$-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 DMA 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 ar
register settings and P27DDR bit setting.

		Setting		
		TPU	**PPG**	**I/O Port**
Module Name	**Pin Function**	**TIOCB5_OE**	**PO7_OE**	**P27DDR**
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	1
	P26 input (initial setting)	0	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 a
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 an
register settings and P23DDR bit setting.

		Setting		
		TPU	PPG	I/O Port
Module Name	Pin Function	TIOCD3_OE	PO3_OE	P23DDR
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	1
	P22 input (initial setting)	0	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 a
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 an register settings and P37DDR bit setting.

		Setting		
		TPU	PPG	I/O Port
Module Name	Pin Function	TIOCB2_OE	PO15_OE	P37DDR
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 an register settings and P33DDR bit setting.

		Setting		
		TPU	PPG	I/O Port
Module Name	Pin Function	TIOCD0_OE	PO11_OE	P33DDR
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 DMA
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 DMA
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 DMAC and SCI register settings and P62DDR bit setting.

		Setting			
		DMAC	TMR	SCI	I/O P
Module Name	**Pin Function**	$\overline{\text{DACK2}}$_OE	TMO2_OE	SCK4_OE	P62D
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

	P61 input (initial setting)	0	0

(6) P60/TMRI2/TxD4/$\overline{\text{DREQ2}}$/$\overline{\text{IRQ8}}$-B

The pin function is switched as shown below according to the combination of the SCI re setting and P60DDR bit setting.

		Setting	
		SCI	I/O Port
Module Name	Pin Function	TxD4_OE	P60DDR
SCI	TxD4 output	1	—
I/O port	P60 output	0	1
	P60 input (initial setting)	0	0

	(initial setting)	
	PA7 input	0

Note: * The type of ϕ to be output switches according to the POSEL1 bit in SCKCR. Fo
see section 18.1.1, System Clock Control Register (SCKCR).

(2) PA6/$\overline{\text{AS}}$/$\overline{\text{AH}}$/$\overline{\text{BS}}$-B

The pin function is switched as shown below according to the combination of bus control
register, port function control register (PFCR), and the PA6DDR bit settings.

		Setting			
		Bus Controller			I/O Po
Module Name	**Pin Function**	**$\overline{\text{AH}}$_OE**	**$\overline{\text{BS}}$-B_OE**	**$\overline{\text{AS}}$_OE**	**PA6D**
Bus controller	$\overline{\text{AH}}$ output	1	—	—	—
	$\overline{\text{BS}}$-B output	0	1	—	—
	$\overline{\text{AS}}$ output (initial setting)	0	0	1	—
I/O port	PA6 output	0	0	0	1
	PA6 input	0	0	0	0

The pin function is switched as shown below according to the combination of bus contro
register, port function control register (PFCR), and the PA4DDR bit settings.

		Setting		
		Bus Controller		I/O Port
Module Name	Pin Function	$\overline{\text{LUB}}$_OE*	$\overline{\text{LHWR}}$_OE*	PA4DDR
Bus controller	$\overline{\text{LUB}}$ output	1	—	—
	$\overline{\text{LHWR}}$ output (initial setting)	—	1	—
I/O port	PA4 output	0	0	1
	PA4 input	0	0	0

Note: * When the byte control SRAM space is accessed while the byte control SRAM
specified or while LHWROE =1, this pin functions as the $\overline{\text{LUB}}$ output; otherwis
$\overline{\text{LHWR}}$ output.

(5) PA3/$\overline{\text{LLWR}}$/$\overline{\text{LLB}}$

The pin function is switched as shown below according to the bus controller register set

		Setting		
		Bus Controller		I/O Port
Module Name	Pin Function	$\overline{\text{LLB}}$_OE*	$\overline{\text{LLWR}}$_OE*	PA3DDR
Bus controller	$\overline{\text{LLB}}$ output	1	—	—
	$\overline{\text{LLWR}}$ output (initial setting)	—	1	—

Note: * If the byte control SRAM space is accessed, this pin functions as the $\overline{\text{LLB}}$ out
otherwise, the $\overline{\text{LLWR}}$.

I/O port	PA2 output	0	0	1
	PA2 input (initial setting)	0	0	0

(7) PA1/$\overline{\text{BACK}}$/(RD/$\overline{\text{WR}}$)

The pin function is switched as shown below according to the combination of bus control register, port function control register (PFCR), and the PA1DDR bit settings.

		Setting			
		Bus Controller			I/O Port
Module Name	Pin Function	$\overline{\text{BACK}}$_OE	Byte control SRAM Selection	(RD/$\overline{\text{WR}}$)_OE	PA1D
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

I/O port	PA0 output	0	0	1
	PA0 input (initial setting)	0	0	0

9.2.7 Port B

(1) PB3/$\overline{\text{CS3}}$/$\overline{\text{CS7}}$-A

The pin function is switched as shown below according to the combination of port funct
register (PFCR) and the PB3DDR bit settings.

		Setting		
		I/O Port		
Module Name	**Pin Function**	**$\overline{\text{CS3}}$_OE**	**$\overline{\text{CS7A}}$_OE**	**PB3DDR**
Bus controller	$\overline{\text{CS3}}$ output	1	—	—
	$\overline{\text{CS7}}$-A output	—	1	—
I/O port	PB3 output	0	0	1
	PB3 input (initial setting)	0	0	0

I/O port	PB2 output	0	0	1
	PB2 input (initial setting)	0	0	0

(3) PB1/$\overline{\text{CS1}}$/$\overline{\text{CS2}}$-B/$\overline{\text{CS5}}$-A/$\overline{\text{CS6}}$-B/$\overline{\text{CS7}}$-B

The pin function is switched as shown below according to the combination of port functio
register (PFCR) and the PB1DDR bit settings.

		Setting				
		I/O Port				
Module Name	**Pin Function**	**$\overline{\text{CS1}}$_OE**	**$\overline{\text{CS2B}}$_OE**	**$\overline{\text{CS5A}}$_OE**	**$\overline{\text{CS6B}}$_OE**	**$\overline{\text{CS7B}}$_OE**
Bus controller	$\overline{\text{CS1}}$ output	1	—	—	—	—
	$\overline{\text{CS2}}$-B output	—	1	—	—	—
	$\overline{\text{CS5}}$-A output	—	—	1	—	—
	$\overline{\text{CS6}}$-B output	—	—	—	1	—
	$\overline{\text{CS7}}$-B output	—	—	—	—	1
I/O port	PB1 output	0	0	0	0	0
	PB1 input (initial setting)	0	0	0	0	0

	$\overline{CS5}$-B output	—	—	1	—
I/O port	PB0 output	0	0	0	1
	PB0 input	0	0	0	0

9.2.8 Port D

(1) PD7/A7, PD6/A6, PD5/A5, PD4/A4, PD3/A3, PD2/A2, PD1/A1, PD0/A0

The pin function is always address output.

		Setting
		I/O Port
Module Name	Pin Function	PDnDDR
Bus controller	Address output	—

[Legend]
n = 0 to 7

[Legend]

n = 0 to 7

9.2.10 Port F

(1) PF7/A23/$\overline{\text{CS4}}$-C/$\overline{\text{CS5}}$-C/$\overline{\text{CS6}}$-C/$\overline{\text{CS7}}$-C

The pin function is switched as shown below according to the combination of port functio register (PFCR) and the PF7DDR bit settings.

		Setting				
		I/O Port				
Module Name	**Pin Function**	**A23_OE**	**$\overline{\text{CS4}}$-C output**	**$\overline{\text{CS5}}$-C output**	**$\overline{\text{CS6}}$-C output**	**$\overline{\text{CS7}}$-C output**
Bus controller	A23 output	1	—	—	—	—
	$\overline{\text{CS4}}$-C output	0	1	—	—	—
	$\overline{\text{CS5}}$-C output	0	—	1	—	—
	$\overline{\text{CS6}}$-C output	0	—	—	1	—
	$\overline{\text{CS7}}$-C output	0	—	—	—	1
I/O port	PF7 output	0	0	0	0	0
	PF7 input (initial setting)	0	0	0	0	0

I/O port	PF6 output	0	0	1
	PF6 input (initial setting)	0	0	0

(3) PF5/A21/$\overline{\text{CS5}}$-D

The pin function is switched as shown below according to the combination of port funct register (PFCR) and the PF5DDR bit settings.

		Setting		
		I/O Port		
Module Name	**Pin Function**	**A21_OE**	**$\overline{\text{CS5D}}$_OE**	**PF5DDR**
Bus controller	A21 output	1	—	—
	$\overline{\text{CS5}}$-D output	0	1	—
I/O port	PF5 output	0	0	1
	PF5 input (initial setting)	0	0	0

(4) PF4/A20

The pin function is always address output.

		Setting
		I/O Port
Module Name	**Pin Function**	**PF4DDR**
Bus controller	A20 output	—

The pin function is always address output.

		Setting
		I/O Port
Module Name	Pin Function	PF2DDR
Bus controller	A18 output	—

(7) PF1/A17

The pin function is always address output.

		Setting
		I/O Port
Module Name	Pin Function	PF1DDR
Bus controller	A17 output	—

(8) PF0/A16

The pin function is always address output.

		Setting
		I/O Port
Module Name	Pin Function	PF0DDR
Bus controller	A16 output	—

(initial setting)

9.2.12 Port I

(1) PI7/D15, PI6/D14, PI5/D13, PI4/D12, PI3/D11, PI2/D10, PI1/D9, PI0/D8

The pin function is switched as shown below according to the combination of operating mode, and the PInDDR bit settings.

		Setting	
		Bus Controller	I/O Port
Module Name	Pin Function	16-Bit Bus Mode	PInDDR
Bus controller	Data I/O (mode 4 initial setting)	1	—
I/O port	PIn output	0	1
	PIn input (mode 5 initial setting)	0	0

[Legend]

n = 0 to 7

					while SMR_3.C/A = 0, SCR_3.CKE [1, 0] = 01
					while SMR_3.C/A = 1, SCR_3.CKE 1 = 0

	5	$\overline{\text{TEND1A}}$_OE	$\overline{\text{TEND1}}$	PFCR7.DMAS1[A,B] = 00	DMDR.TENDE = 1
	4	TxD3_OE	TxD3		SCR.TE = 1
	2	$\overline{\text{DACK0A}}$_OE	$\overline{\text{DACK0}}$	PFCR7.DMAS0[A,B] = 00	DACR.AMS = 1, DMDR.DACKE = 1
		SCK2_OE	SCK2		When SCMR.SMIF = 1: SCR.TE = 1 or SCR.RE = 1 while SMR.GM = 0 SCR.CKE [1, 0] = 01 or while SMR.GM = 1 When SCMR.SMIF = 0: SCR.TE = 1 or SCR.RE = 1 while SMR.C/A = 0, SCR.CKE [1, 0] = 01 or while SMR.C/A = 1, SCR.CKE 1 = 0
	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.IOB[1,0] = 0
		PO7_OE	PO7		NDERL.NDER7 = 1
	6	TIOCA5_OE	TIOCA5		TPU.TIOR5.IOA3 = 0, TPU.TIOR5.IOA[1,0] = 0
		TMO1_OE	TMO1		TCSR.OS3,2 = 01/10/11 or TCSR.OS[1,0] = 01
		TxD1_OE	TxD1		SCR.TE = 1
		PO6_OE	PO6		NDERL.NDER6 = 1

			while SMR.C/A = 0, SCR.CKE [1, 0] = 01 or
			while SMR.C/A = 1, SCR.CKE 1 = 0

	PO4_OE	PO4	NDERL.NDER4 = 1
3	TIOCD3_OE	TIOCD3	TPU.TMDR.BFB = 0, TPU.TIORL3.IOD3 = 0, TPU.TIORL3.IOD[1,0] = 01/10/11
	PO3_OE	PO3	NDERL.NDER3 = 1
2	TIOCC3_OE	TIOCC3	TPU.TMDR.BFA = 0, TPU.TIORL3.IOC3 = 0, TPU.TIORL3.IOD[1,0] = 01/10/11
	TMO0_OE	TMO0	TCSR.OS[3,2] = 01/10/11 or TCSR.OS[1,0] =
	TxD0_OE	TxD0	SCR.TE = 1
	PO2_OE	PO2	NDERL.NDER2 = 1
1	TIOCA3_OE	TIOCA3	TPU.TIORH3.IOA3 = 0, TPU.TIORH3.IOA[1,(
	PO1_OE	PO1	NDERL.NDER1 = 1
0	TIOCB3_OE	TIOCB3	TPU.TIORH3.IOB3 = 0, TPU.TIORH3.IOB[1,(

	SCK0_OE	SCK0	When SCMR.SMIF = 1:
			SCR.TE = 1 or SCR.RE = 1 while
			SMR.GM = 0, SCR.CKE [1, 0] = 01 or
			while SMR.GM = 1
			When SCMR.SMIF = 0:
			SCR.TE = 1 or SCR.RE = 1 while
			SMR.C/A = 0, SCR.CKE [1, 0] = 01 or while S
			SCR.CKE 1 = 0

	PO0_OE	PO0	NDERL.NDER0 = 1

	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.TIOR1.IOA[1,0] = 0
	PO12_OE	PO12		NDERH.NDER12 = 1
3	TIOCD0_OE	TIOCD0		TPU.TMDR.BFB = 0, TPU.TIORL0.IOD3 = 0, TPU.TIORL0.IOD[1,0] = 01/10/11
	PO11_OE	PO11		NDERH.NDER11 = 1
2	$\overline{\text{DACK0B}}$_OE	$\overline{\text{DACK0}}$	PFCR7.DMAS0[A,B] = 01	DACR.AMS = 1, DMDR.DACKE = 1
	TIOCC0_OE	TIOCC0		TPU.TMDR.BFA = 0, TPU.TIORL0.IOC3 = 0, TPU.TIORL0.IOD[1,0] = 01/10/11
	PO10_OE	PO10		NDERH.NDER10 = 1
1	$\overline{\text{TEND0B}}$_OE	$\overline{\text{TEND0}}$	PFCR7.DMAS0[A,B] = 01	DMDR.TENDE = 1
	TIOCB0_OE	TIOCB0		TPU.TIORH0.IOB3 = 0, 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

		SCK4_OE	SCK4		When SCMR.SMIF = 1: SCR.TE = 1 or SCR.RE = 1 while SMR.GM = 0, SCR.CKE [1, 0] = 01 or while SMR.GM = 1 When SCMR.SMIF = 0: SCR.TE = 1 or SCR.RE = 1 while SMR.C/A = 0, SCR.CKE [1, 0] = 01 or while SMR.C/A = 1, SCR.CKE 1 = 0
	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.POSEL1 = 0
	6	$\overline{\text{AH}}$_OE	$\overline{\text{AH}}$		MPXCR.MPXEn (n = 7 to 3) = 1
		$\overline{\text{BS-B}}$_OE	$\overline{\text{BS}}$	PFCR2.BSS = 1	PFCR2.BSE = 1
		$\overline{\text{AS}}$_OE	$\overline{\text{AS}}$		PFCR2.ASOE = 1
	5	$\overline{\text{RD}}$_OE	$\overline{\text{RD}}$		
	4	$\overline{\text{LUB}}$_OE	$\overline{\text{LUB}}$		PFCR6.LHWROE = 1 or SRAMCR.BCSELn =
		$\overline{\text{LHWR}}$_OE	$\overline{\text{LHWR}}$		PFCR6.LHWROE = 1
	3	$\overline{\text{LLB}}$_OE	$\overline{\text{LLB}}$		SRAMCR.BCSELn = 1
		$\overline{\text{LLWR}}$_OE	$\overline{\text{LLWR}}$		SRAMCR.BCSELn = 0
	1	$\overline{\text{BACK}}$_OE	$\overline{\text{BACK}}$		BCR1.BRLE = 1
		(RD/$\overline{\text{WR}}$)_OE	RD/$\overline{\text{WR}}$		PFCR2.REWRE = 1 or SRAMCR.BCSELn =
	0	$\overline{\text{BS-A}}$_OE	$\overline{\text{BS}}$	PFCR2.BSS = 0	PFCR2.BSE = 1
		$\overline{\text{BREQO}}$_OE	$\overline{\text{BREQO}}$		BCR1.BRLE = 1, BCR1.BREQOE = 1

		$\overline{\text{CS6B}}$_OE	$\overline{\text{CS6}}$	PFCR1.CS6S[A,B] = 01	PFCR0.CS6E = 1
		$\overline{\text{CS7B}}$_OE	$\overline{\text{CS7}}$	PFCR1.CS7S[A,B] = 01	PFCR0.CS7E = 1
	0	$\overline{\text{CS0}}$_OE	$\overline{\text{CS0}}$		PFCR0.CS0E = 1
		$\overline{\text{CS4A}}$_OE	$\overline{\text{CS4}}$	PFCR1.CS4S[A,B] = 00 PFCR0.CS4E = 1	PFCR0.CS4E = 1
		$\overline{\text{CS5B}}$_OE	$\overline{\text{CS5}}$	PFCR1.CS5S[A,B] = 01	PFCR0.CS5E = 1
PD	7	A7_OE	A7		
	6	A6_OE	A6		
	5	A5_OE	A5		
	4	A4_OE	A4		
	3	A3_OE	A3		
	2	A2_OE	A2		
	1	A1_OE	A1		
	0	A0_OE	A0		
PE	7	A15_OE	A15		
	6	A14_OE	A14		
	5	A13_OE	A13		
	4	A12_OE	A12		
	3	A11_OE	A11		
	2	A10_OE	A10		
	1	A9_OE	A9		
	0	A8_OE	A8		

		CS3D_OE	CS3	PFCR1.CS3S[A,B] = 11	PFCR0.CS3E = 1
	4	A20_OE	A20		
	3	A19_OE	A19		
	2	A18_OE	A18		
	1	A17_OE	A17		
	0	A16_OE	A16		
PH	7	D7_E	D7		
	6	D6_E	D6		
	5	D5_E	D5		
	4	D4_E	D4		
	3	D3_E	D3		
	2	D2_E	D2		
	1	D1_E	D1		
	0	D0_E	D0		
PI	7	D15_E	D15	ABWCR.ABW[H,L]n = 01	
	6	D14_E	D14	ABWCR.ABW[H,L]n = 01	
	5	D13_E	D13	ABWCR.ABW[H,L]n = 01	
	4	D12_E	D12	ABWCR.ABW[H,L]n = 01	
	3	D11_E	D11	ABWCR.ABW[H,L]n = 01	
	2	D10_E	D10	ABWCR.ABW[H,L]n = 01	
	1	D9_E	D9	ABWCR.ABW[H,L]n = 01	
	0	D8_E	D8	ABWCR.ABW[H,L]n = 01	

- 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 $\overline{\text{CS}}$ 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
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	CS7E	0	R/W	CS7 to CS0 Enable These bits enable/disable the corresponding $\overline{\text{C}}$ output. 0: Pin functions as I/O port 1: Pin functions as $\overline{\text{CSn}}$ output pin (n = 7 to 0)
6	CS6E	0	R/W	
5	CS5E	0	R/W	
4	CS4E	0	R/W	
3	CS3E	0	R/W	
2	CS2E	0	R/W	
1	CS1E	0	R/W	
0	CS0E	1	R/W	

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 $\overline{\text{CS7}}$ outpu enabled (CS7E = 1) 00: Specifies pin PB3 as $\overline{\text{CS7}}$-A output 01: Specifies pin PB1 as $\overline{\text{CS7}}$-B output 10: Specifies pin PF7 as $\overline{\text{CS7}}$-C output 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 $\overline{\text{CS6}}$ outpu enabled (CS6E = 1) 00: Specifies pin PB2 as $\overline{\text{CS6}}$-A output 01: Specifies pin PB1 as $\overline{\text{CS6}}$-B output 10: Specifies pin PF7 as $\overline{\text{CS6}}$-C output 11: Specifies pin PF6 as $\overline{\text{CS6}}$-D output
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 $\overline{\text{CS5}}$ outpu enabled (CS5E = 1) 00: Specifies pin PB1 as $\overline{\text{CS5}}$-A output 01: Specifies pin PB0 as $\overline{\text{CS5}}$-B output 10: Specifies pin PF7 as $\overline{\text{CS5}}$-C output 11: Specifies pin PF5 as $\overline{\text{CS5}}$-D output

select bits (n=4 to 7), multiple CS signals are output from the pin. For details, s section 6.5.3, Chip Select Signals.

9.3.3 Port Function Control Register 2 (PFCR2)

PFCR1 selects the $\overline{\text{CS}}$ output pin, enables/disables bus control I/O, and selects the bus co 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 shou always be 0.
6	CS2S*[1]	0	R/W	$\overline{\text{CS2}}$ Output Pin Select Selects the output pin for $\overline{\text{CS2}}$ when $\overline{\text{CS2}}$ output enabled (CS2E = 1) 0: Specifies pin PB2 as $\overline{\text{CS2}}$-A output pin 1: Specifies pin PB1 as $\overline{\text{CS2}}$-B output pin

3	—	0	R/W	Reserved This bit is always read as 0. The write value sh always be 0.
2	RDWRE*[2]	0	R/W	RD/$\overline{\text{WR}}$ Output Enable 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 sh always be 0.

Notes: 1. If multiple $\overline{\text{CS}}$ outputs are specified to a single pin according to the $\overline{\text{CS2}}$ outpu select bit, multiple $\overline{\text{CS}}$ signals are output from the pin. For details, see sectio Chip Select Signals.

2. If an area is specified as a byte control SDRAM space, the pin functions as R output.

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 to 0	—	All 1	R/W	Reserved These bits are always read as 1. The write value always be 1.

7	—	1	R/W	Reserved This bit is always read as 1. The write value sh always be 1.
6	LHWROE	1	R/W	$\overline{\text{LHWR}}$ Output Enable Enables/disables $\overline{\text{LHWR}}$ output (valid in extern extended mode). 0: Specifies pin PA4 as I/O port 1: Specifies pin PA4 as $\overline{\text{LHWR}}$ output pin
5	—	1	R/W	Reserved This bit is always read as 1. The write value sh always be 1.
4	—	0	R	Reserved This is a read-only bit and cannot be modified.
3	TCLKS	0	R/W	TPU External Clock Input Pin Select Selects the TPU external clock input pins. 0: Specifies pins P32, P33, P35, and P37 as ex 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.

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
5	DMAS2A	0	R/W	DMAC Control Pin Select
4	DMAS2B	0	R/W	Selects the I/O port to control DMAC_2.
				00: Setting prohibited
				01: Specifies pins P60 to P62 as DMAC control
				10: Setting prohibited
				11: Setting prohibited
3	DMAS1A	0	R/W	DMAC Control Pin Select
2	DMAS1B	0	R/W	Selects the I/O port to control DMAC_1.
				00: Specifies pins P14 to P16 as DMAC control
				01: Specifies pins P33 to P35 as DMAC control
				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
				01: Specifies pins P30 to P32 as DMAC control
				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 P2 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 P2 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 P2 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 P2 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

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

3. When a pin is used as an output, data to be output from the pin will be latched as the p
   if the input by the ICR setting is enabled. To use the pin as an output, disable the inpu
   function for the pin by setting ICR.

9.4.2 Notes on Port Function Control Register (PFCR) Settings

1. The port function controller controls the I/O ports. To set the input/output to each pin,
   the input/output destination and then enable input/output.
2. When changing the input pin, an edge may be generated if the previous pin level diffe
   the pin level after the change, causing an unintended malfunction. To change the inpu
   follow the procedure below.
   A. Disable the input function by the setting of the peripheral module corresponding t
      to be changed.
   B. Select the input pin by the setting of PFCR.
   C. Enable the input function by the setting of the peripheral module corresponding to
      to be changed.
3. If a pin function has both a selection bit that modifies the input/output destination and
   enable bit that enables the pin function, first specify the input/output destination by th
   selection bit 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 synchrono 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 mode can be set

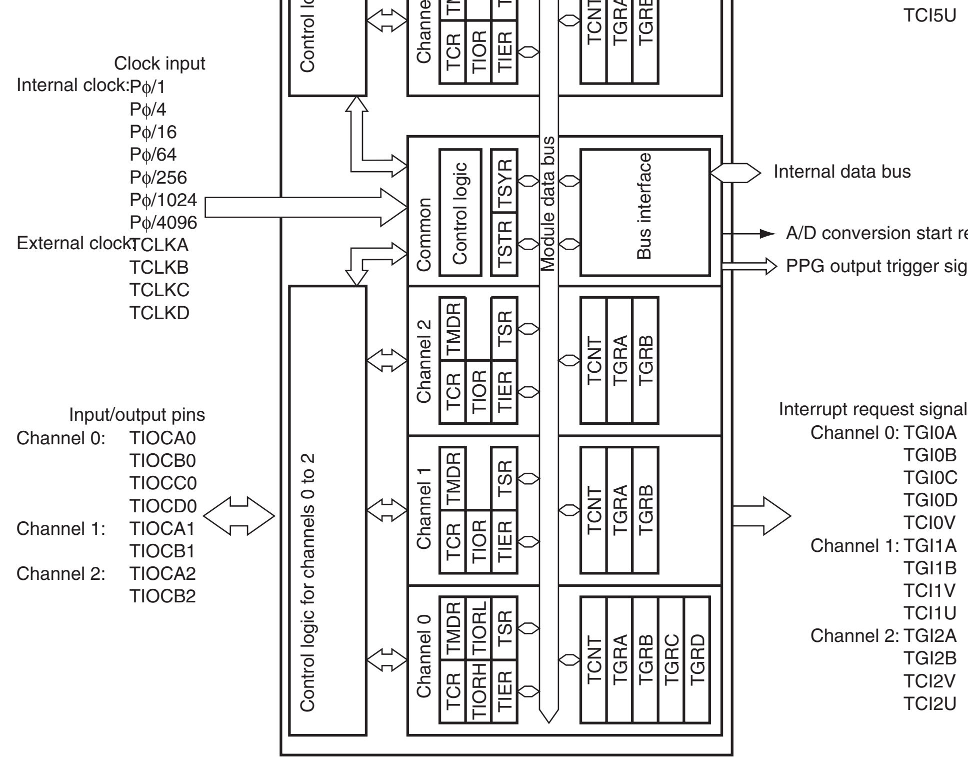
		TGRB_0	TGRB_1	TGRB_2	TGRB_3	TGRB_4	TG
General registers/ buffer registers		TGRC_0 TGRD_0	—	—	TGRC_3 TGRD_3	—	—
I/O pins		TIOCA0 TIOCB0 TIOCC0 TIOCD0	TIOCA1 TIOCB1	TIOCA2 TIOCB2	TIOCA3 TIOCB3 TIOCC3 TIOCD3	TIOCA4 TIOCB4	TI TI
Counter clear function		TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TG ma cap
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	—	—

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 ca C m ca C U

[Legend]

O : Possible

— : Not possible



[Legend]

TSTR:	Timer start register	TIER:	Timer interrupt enable register
TSYR:	Timer synchronous register	TSR:	Timer status register
TCR:	Timer control register	TGR (A, B, C, D):	Timer general registers (A, B, C, D)
TMDR:	Timer mode register	TCNT:	Timer counter
TIOR (H, L):	Timer I/O control registers (H, L)		

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 interrupt enable register_0 (TIER_0)
- Timer status register_0 (TSR_0)
- Timer counter_0 (TCNT_0)
- Timer general register A_0 (TGRA_0)
- Timer general register B_0 (TGRB_0)
- Timer general register C_0 (TGRC_0)
- Timer general register D_0 (TGRD_0)

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 3:

- Timer control register_3 (TCR_3)
- Timer mode register_3 (TMDR_3)
- Timer I/O control register H_3 (TIORH_3)
- Timer I/O control register L_3 (TIORL_3)
- Timer interrupt enable register_3 (TIER_3)
- Timer status register_3 (TSR_3)
- Timer counter_3 (TCNT_3)
- Timer general register A_3 (TGRA_3)
- 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)

Common Registers:

- Timer start register (TSTR)
- Timer synchronous register (TSYR)

Bit	Bit Name	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 so tables 10.3 and 10.4 for details.
5	CCLR0	0	R/W	
4	CKEG1	0	R/W	Clock Edge 1 and 0
3	CKEG0	0	R/W	These bits select the input clock edge. For deta table 10.5. When the input clock is counted usi edges, the input clock period is halved (e.g. Pϕ/ edges = Pϕ/2 rising edge). If phase counting mo used on channels 1, 2, 4, and 5, this setting is i and the phase counting mode setting has priori clock edge selection is valid when the input cloc or slower. This setting is ignored if the input clo or when overflow/underflow of another channel selected.
2	TPSC2	0	R/W	Timer Prescaler 2 to 0
1	TPSC1	0	R/W	These bits select the TCNT counter clock. The source can be selected independently for each See tables 10.6 to 10.11 for details. To select th clock as the clock source, the DDR bit and ICR corresponding pin should be set to 0 and 1, res For details, see section 9, I/O Ports.
0	TPSC0	0	R/W	

			synchronous operation
1	0	0	TCNT clearing disabled
1	0	1	TCNT cleared by TGRC compare matcl capture*2
1	1	0	TCNT cleared by TGRD compare matcl capture*2
1	1	1	TCNT cleared by counter clearing for ar channel performing synchronous clearir 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 becau buffer register setting has priority, and compare match/input capture does not c

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 matcl capture
	0	1	0	TCNT cleared by TGRB compare matcl capture
	0	1	1	TCNT cleared by counter clearing for ar channel performing synchronous clearir 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 b 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 i
	1	0	1	External clock: counts on TCLKB pin i
	1	1	0	External clock: counts on TCLKC pin i
	1	1	1	External clock: counts on TCLKD pin i

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 i
	1	0	1	External clock: counts on TCLKB pin i
	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 in
	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 in
	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 i
	1	0	1	External clock: counts on TCLKC pin i
	1	1	0	Internal clock: counts on Pϕ/256
	1	1	1	External clock: counts on TCLKD pin i

Note: This setting is ignored when channel 5 is in phase counting mode.

Bit	Bit Name	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 TGRB and TGRD are to be used together for b operation. When TGRD is used as a buffer reg TGRD input capture/output compare is not ger In channels 1, 2, 4, and 5, which have no TGR reserved. It is always read as 0 and cannot be 0: TGRB operates normally 1: TGRB and TGRD used together for buffer o
4	BFA	0	R/W	Buffer Operation A Specifies whether TGRA is to normally operate TGRA and TGRC are to be used together for b operation. When TGRC is used as a buffer reg TGRC input capture/output compare is not ger In channels 1, 2, 4, and 5, which have no TGR reserved. It is always read as 0 and cannot be 0: TGRA operates normally 1: TGRA and TGRC used together for buffer o
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. See table 10.12 for details.
0	MD0	0	R/W	

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, each for channels 1, 2, 4, and 5. Care is required since TIOR is affected by the TMDR s

The initial output specified by TIOR is valid when the counter is stopped (the CST bit ir cleared to 0). Note also that, in PWM mode 2, the output at the point at which the count cleared to 0 is specified.

When TGRC or TGRD is designated for buffer operation, this setting is invalid and the operates as a buffer register.

To designate the input capture pin in TIOR, the DDR bit and ICR bit for the correspond should be set to 0 and 1, respectively. For details, see section 9, I/O Ports.

- 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, 10.17 and 10.20.
4	IOB0	0	R/W	
3	IOA3	0	R/W	I/O Control A3 to A0
2	IOA2	0	R/W	Specify the function of TGRA.
1	IOA1	0	R/W	For details, see tables 10.21, 10.23, 10.24, 10.25 and 10.28.
0	IOA0	0	R/W	

- 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/coun Input capture at TCNT_1 count-up/cour

[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 t TCNT_1 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/count Input capture at TCNT_1 count-up/count

[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 th 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 regis 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 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/coun Input capture at TCNT_4 count-up/cour

[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 T 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/count Input capture at TCNT_4 count-up/count

[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 th 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 regis 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 comp match/input capture Input capture at generation of TGRC_3 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/coun Input capture at TCNT_1 count-up/coun

[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/count Input capture at TCNT_1 count-up/count

[Legend]

X: Don't care

Note: * When the BFA bit in TMDR_0 is set to 1 and TGRC_0 is used as a buffer regis 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/coun Input capture at TCNT_4 count-up/coun

[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/count Input capture at TCNT_4 count-up/count

[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 regis 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 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	value	R/W	Description
7	TTGE	0	R/W	A/D Conversion Start Request Enable Enables/disables generation of A/D conversion requests by TGRA input capture/compare matc 0: A/D conversion start request generation disa 1: A/D conversion start request generation enal
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 t flag when the TCFU flag in TSR is set to 1 in ch 2, 4, and 5. In channels 0 and 3, bit 5 is reserved. It is alwa 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 t 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 th bit when the TGFC bit in TSR is set to 1 in chann and 3. In channels 1, 2, 4, and 5, bit 2 is reserved. It is a read as 0 and cannot be modified. 0: Interrupt requests (TGIC) by TGFC bit disable 1: Interrupt requests (TGIC) by TGFC bit enablec
1	TGIEB	0	R/W	TGR Interrupt Enable B Enables/disables interrupt requests (TGIB) by th bit when the TGFB bit in TSR is set to 1. 0: Interrupt requests (TGIB) by TGFB bit disable 1: Interrupt requests (TGIB) by TGFB bit enablec
0	TGIEA	0	R/W	TGR Interrupt Enable A Enables/disables interrupt requests (TGIA) by th bit when the TGFA bit in TSR is set to 1. 0: Interrupt requests (TGIA) by TGFA bit disable 1: Interrupt requests (TGIA) by TGFA bit enablec

Bit	Bit Name	value	R/W	Description
7	TCFD	1	R	Count Direction Flag Status flag that shows the direction in which TC in channels 1, 2, 4, and 5. In channels 0 and 3, bit 7 is reserved. It is alwa 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 underflo occurred when channels 1, 2, 4, and 5 are set t counting mode. In channels 0 and 3, bit 5 is reserved. It is alwa 0 and cannot be modified. [Setting condition] When the TCNT value underflows (changes fro to H'FFFF) [Clearing condition] When a 0 is written to TCFU after reading TCF (When the CPU is used to clear this flag by wri while the corresponding interrupt is enabled, b read the flag after writing 0 to it.)

				(When the CPU is used to clear this flag by writin while the corresponding interrupt is enabled, be 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 TGR 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 functior output compare register • When TCNT value is transferred to TGRD by capture signal while TGRD is functioning as i capture register [Clearing conditions] • When DTC is activated by a TGID interrupt w DISEL bit in MRB of DTC is 0 • When 0 is written to TGFD 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.)

				• When TCNT value is transferred to TGRC b capture signal while TGRC is functioning as capture register [Clearing conditions] • When DTC is activated by a TGIC interrupt DISEL bit in MRB of DTC is 0 • When 0 is written to TGFC 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.)
1	TGFB	0	R/(W)*	Input Capture/Output Compare Flag B Status flag that indicates the occurrence of TGI capture or compare match. [Setting conditions] • When TCNT = TGRB while TGRB is functio output compare register • When TCNT value is transferred to TGRB b capture signal while TGRB is functioning as capture register [Clearing conditions] • When DTC is activated by a TGIB interrupt DISEL bit in MRB of DTC is 0 • When 0 is written to TGFB 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.)

[Clearing conditions]

- When DTC is activated by a TGIA interrupt w
  DISEL bit in MRB of DTC is 0
- When DMAC is activated by a TGIA interrupt
  the DTA bit in DMDR of DMAC is 1
- When 0 is written to TGFA 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.)

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 in
capture registers. The TPU has 16 TGR registers, four each for channels 0 and 3 and tw
channels 1, 2, 4, and 5. TGRC and TGRD for channels 0 and 3 can also be designated f
operation as buffer registers. The TGR registers cannot be accessed in 8-bit units; they r
always be accessed in 16-bit units. TGR and buffer register combinations during buffer
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

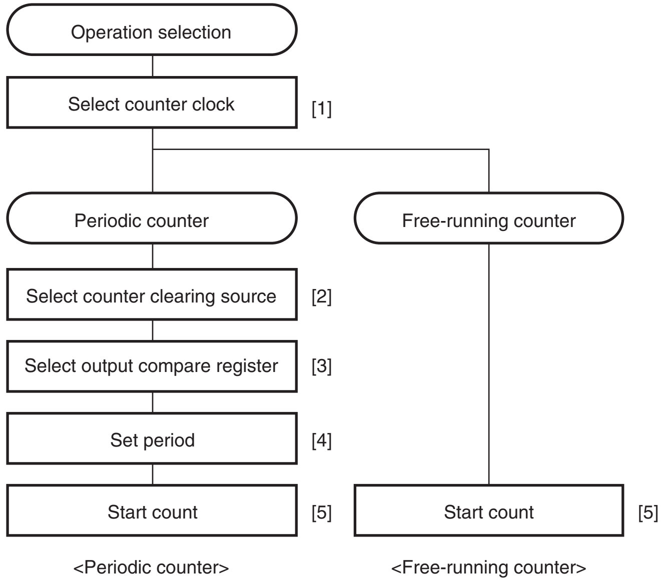
Bit	Bit Name	value	R/W	Description
7, 6	—	All 0	R/W	Reserved These bits are always read as 0. The write value always be 0.
5	CST5	0	R/W	Counter Start 5 to 0
4	CST4	0	R/W	These bits select operation or stoppage for TCN
3	CST3	0	R/W	If 0 is written to the CST bit during operation with TIOC pin designated for output, the counter stop TIOC pin output compare output level is retainec is written to when the CST bit is cleared to 0, the output level will be changed to the set initial outp
2	CST2	0	R/W	
1	CST1	0	R/W	
0	CST0	0	R/W	
				0: TCNT_5 to TCNT_0 count operation is stoppe
				1: TCNT_5 to TCNT_0 performs count operation

Bit	Bit Name	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	SYNC5	0	R/W	Timer Synchronization 5 to 0 These bits select whether operation is indepenc synchronized with other channels. When synchronous operation is selected, synch presetting of multiple channels, and synchronou through counter clearing on another channel ar To set synchronous operation, the SYNC bits fo two channels must be set to 1. To set synchron clearing, in addition to the SYNC bit, the TCNT source must also be set by means of bits CCLF CCLR0 in TCR. 0: TCNT_5 to TCNT_0 operate independently (presetting/clearing is unrelated to other char 1: TCNT_5 to TCNT_0 perform synchronous op (TCNT synchronous presetting/synchronous is possible)
4	SYNC4	0	R/W	
3	SYNC3	0	R/W	
2	SYNC2	0	R/W	
1	SYNC1	0	R/W	
0	SYNC0	0	R/W	

When one of bits CST0 to CST5 is set to 1 in TSTR, the TCNT counter for the correspon
channel starts counting. TCNT can operate as a free-running counter, periodic counter, ar

(a) Example of count operation setting procedure

Figure 10.2 shows an example of the count operation setting procedure.



[1] Select the counter clock with b
to TPSC0 in TCR. At the same
select the input clock edge with
CKEG1 and CKEG0 in TCR.

[2] For periodic counter operation,
TGR to be used as the TCNT c
source with bits CCLR2 to CCL
TCR.

[3] Designate the TGR selected in
output compare register by me
TIOR.

[4] Set the periodic counter cycle i
selected in [2].

[5] Set the CST bit in TSTR to 1 to
counter operation.

Figure 10.2 Example of Counter Operation Setting Procedure

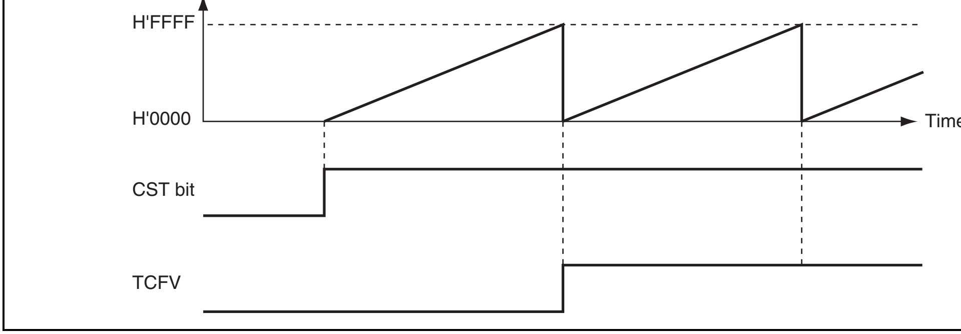


Figure 10.3 Free-Running Counter Operation

When compare match is selected as the TCNT clearing source, the TCNT counter for th
channel performs periodic count operation. The TGR register for setting the period is de
as an output compare register, and counter clearing by compare match is selected by me
CCLR2 to CCLR0 in TCR. After the settings have been made, TCNT starts count-up op
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 a
After a compare match, TCNT starts counting up again from H'0000.

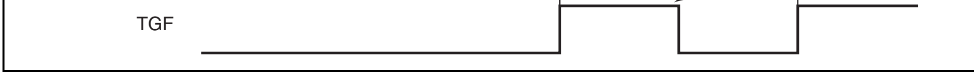


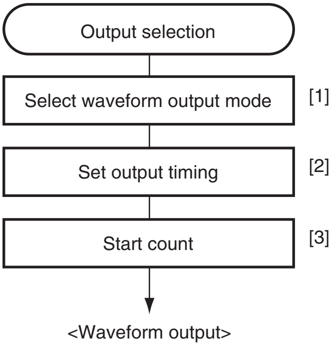
Figure 10.4 Periodic Counter Operation

(2) Waveform Output by Compare Match

The TPU can perform 0, 1, or toggle output from the corresponding output pin using a co 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 compar



[1] Select initial value from 0-output or 1-output, and compare match output value from 0-output, 1-output, or toggle-output, by means of TIOR. The set initial value is output on the TIOC pin until the first compare match occurs.

[2] Set the timing for compare match generation in TGR.

[3] Set the CST bit in TSTR to 1 to start the count operation.

Figure 10.5 Example of Setting Procedure for Waveform Output by Compare M

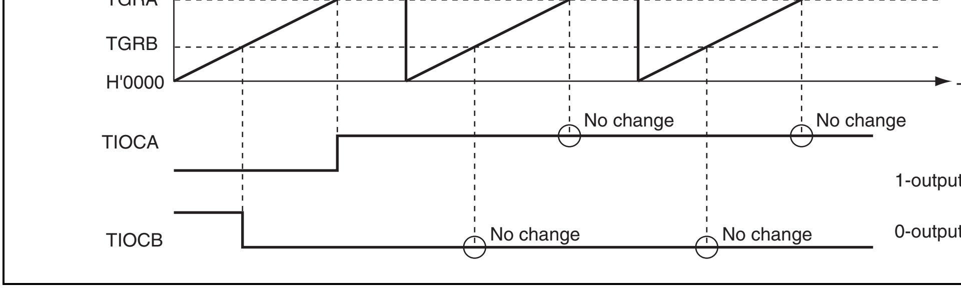


Figure 10.6 Example of 0-Output/1-Output Operation

Figure 10.7 shows an example of toggle output.

In this example, TCNT has been designated as a periodic counter (with counter clearing by compare match B), and settings have been made so that output is toggled by both con match A and compare match B.

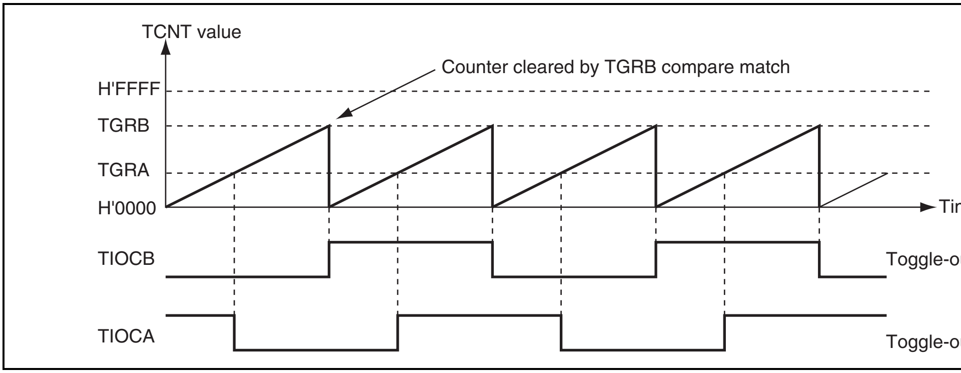
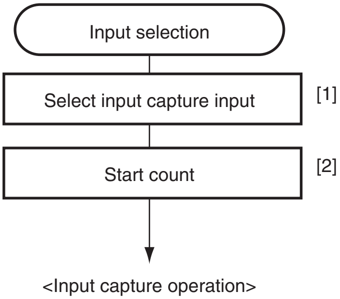


Figure 10.7 Example of Toggle Output Operation

(a) Example of setting procedure for input capture operation

Figure 10.8 shows an example of the setting procedure for input capture operation.



[1] Designate TGR as an input capture register by means of TIOR, and select the input capture source and input signal edge (rising edge, falling edge, or both edges).

[2] Set the CST bit in TSTR to 1 to start the count operation.

Figure 10.8 Example of Setting Procedure for Input Capture Operation

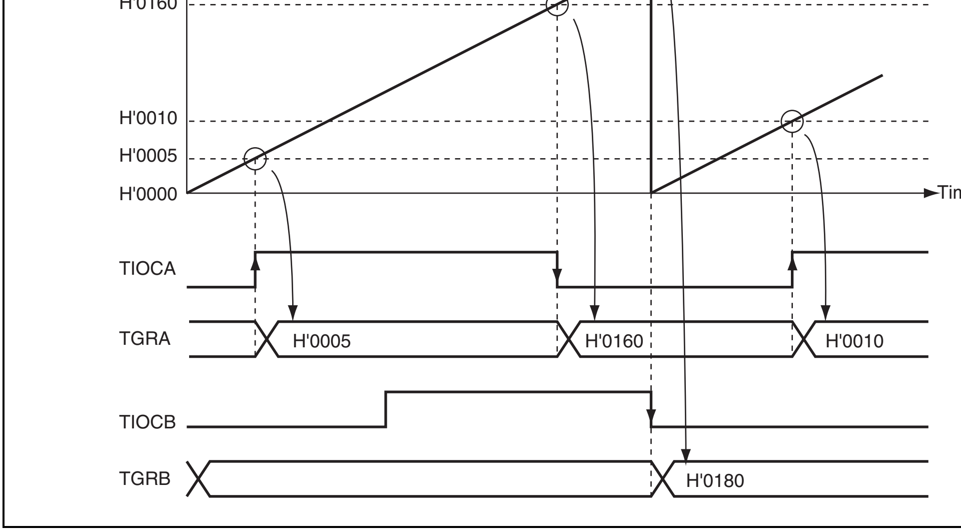
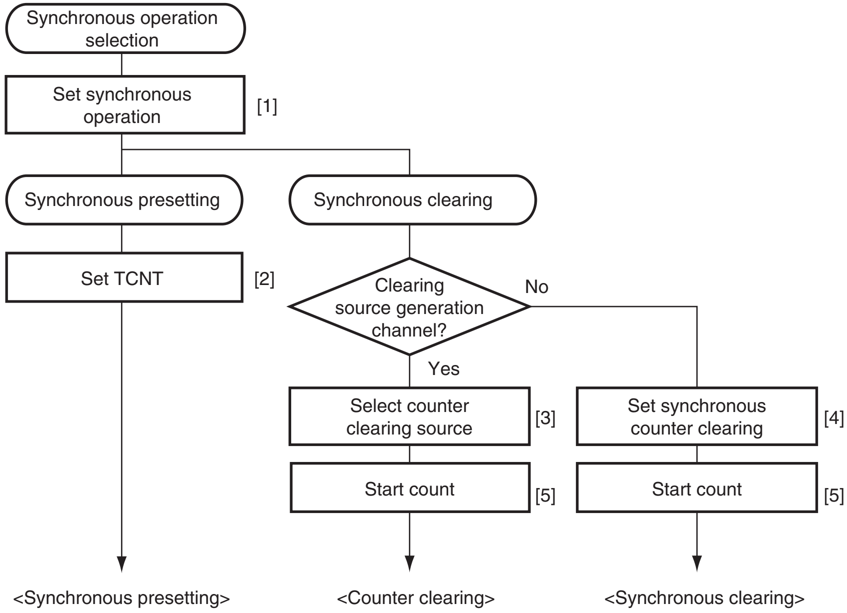


Figure 10.9 Example of Input Capture Operation

Figure 10.10 shows an example of the synchronous operation setting procedure.



[1] Set the SYNC bits in TSYR corresponding to the channels to be designated for synchronous operation to

[2] When the TCNT counter of any of the channels designated for synchronous operation is written to, the same value is simultaneously written to the other TCNT counters.

[3] Use bits CCLR2 to CCLR0 in TCR to specify TCNT clearing by input capture/output compare, etc.

[4] Use bits CCLR2 to CCLR0 in TCR to designate synchronous clearing for the counter clearing source.

[5] Set the CST bits in TSTR for the relevant channels to 1, to start the count operation.

Figure 10.10 Example of Synchronous Operation Setting Procedure

For details on PWM modes, see section 10.4.5, PWM Modes.

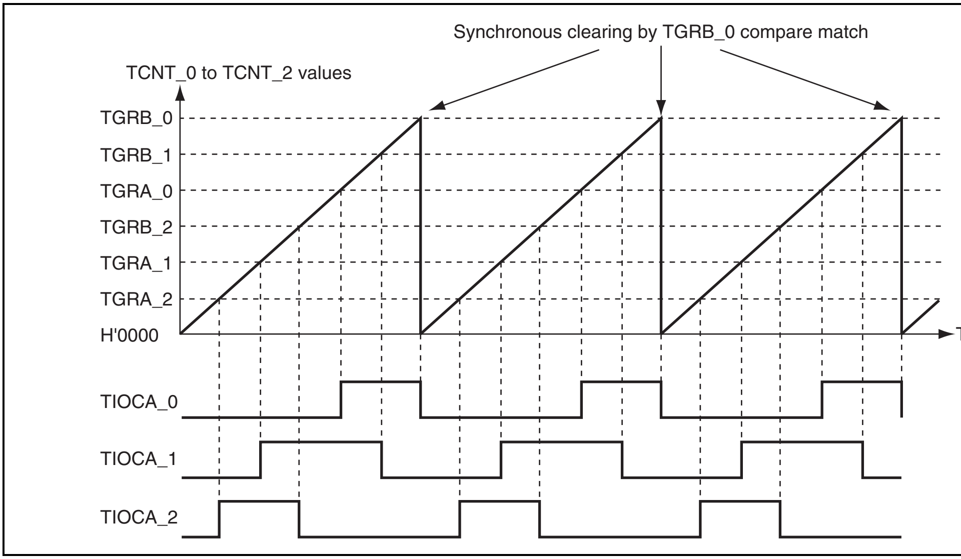


Figure 10.11 Example of Synchronous Operation

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 c transferred to the timer general register.

  This operation is illustrated in figure 10.12.

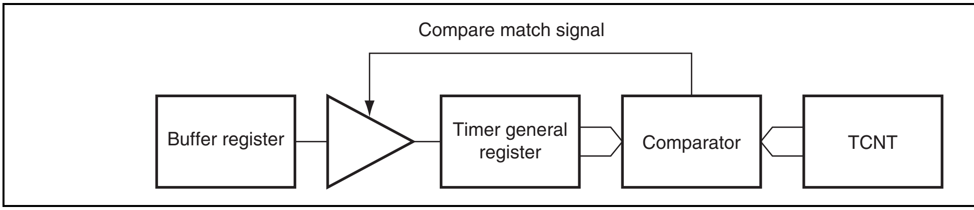
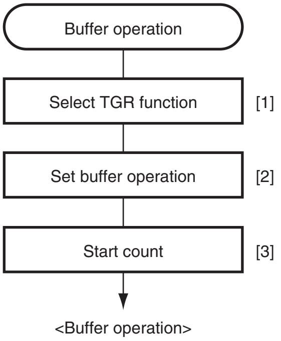


Figure 10.12 Compare Match Buffer Operation

Figure 10.13 Input Capture 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.

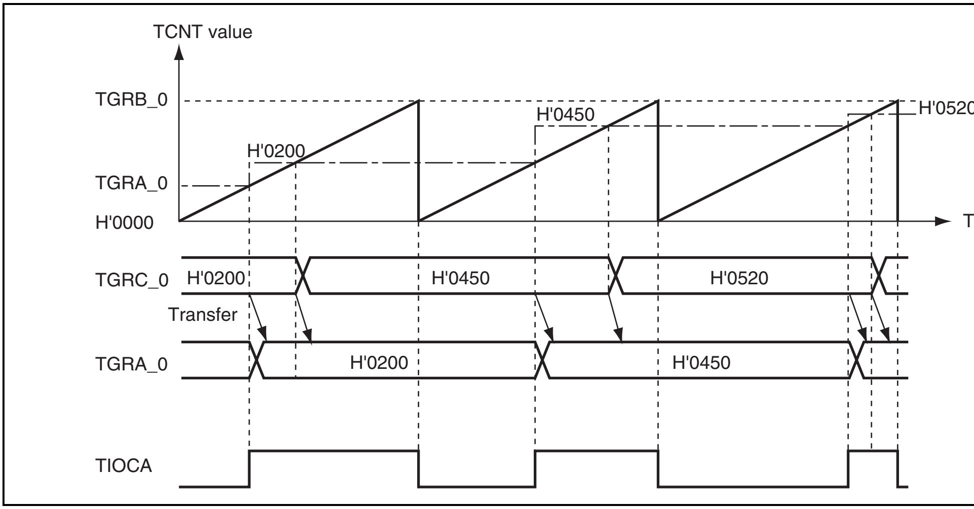


Figure 10.15 Example of Buffer Operation (1)

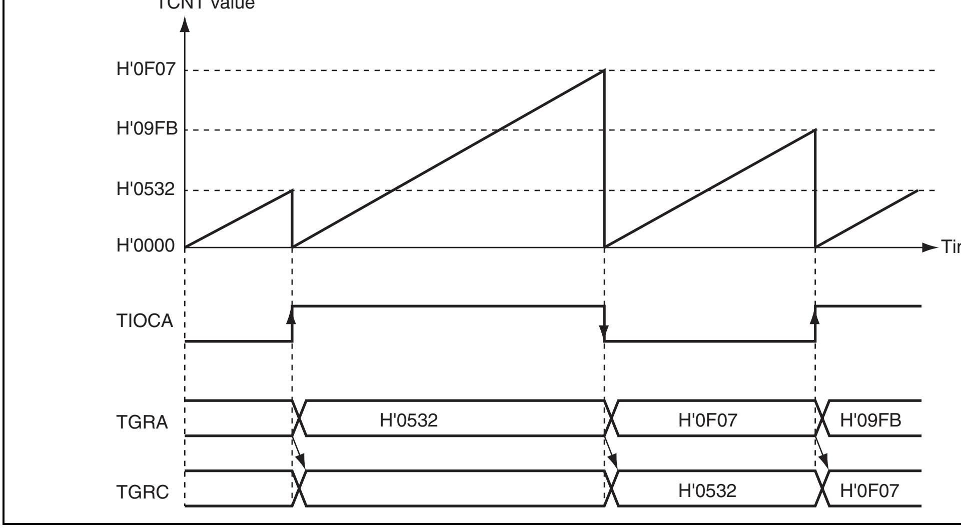


Figure 10.16   Example of Buffer Operation (2)

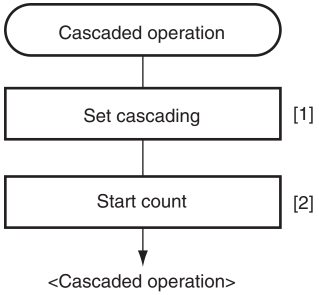
Note: When phase counting mode is set for channel 1 or 4, the counter clock setting is i and the counter operates independently in phase counting mode.

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

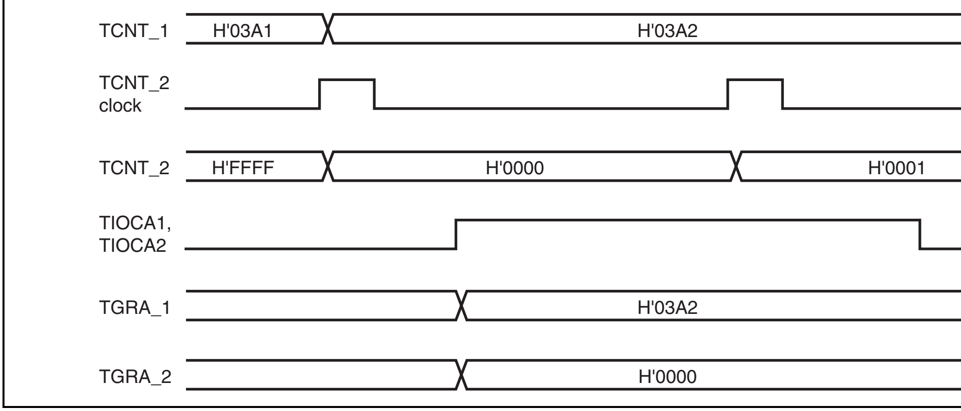


Figure 10.18 Example of Cascaded Operation (1)

Figure 10.19 illustrates the operation when counting upon TCNT_2 overflow/underflow 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

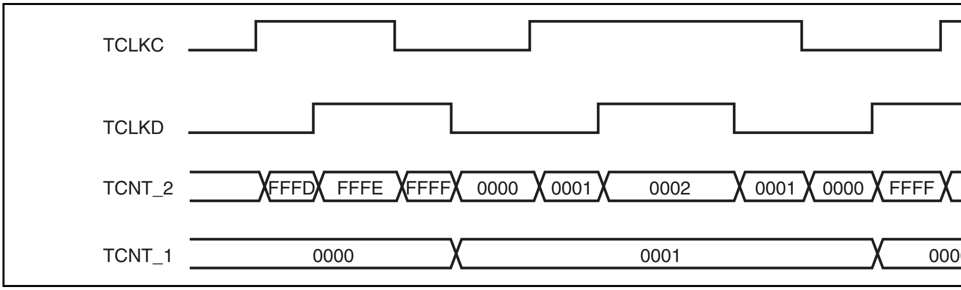


Figure 10.19 Example of Cascaded Operation (2)

There are two PWM modes, as described below.

(a) PWM mode 1

PWM output is generated from the TIOCA and TIOCC pins by pairing TGRA with TGR
TGRC with TGRD. The outputs specified by bits IOA3 to IOA0 and IOC3 to IOC0 in TI
output from the TIOCA and TIOCC pins at compare matches A and C, respectively. The
specified by bits IOB3 to IOB0 and IOD3 to IOD0 in TIOR are output at compare match
D, respectively. The initial output value is the value set in TGRA or TGRC. If the set valu
paired TGRs are identical, the output value does not change when a compare match occu

In PWM mode 1, a maximum 8-phase PWM output is possible.

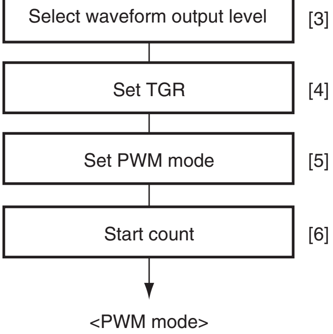
(b) PWM mode 2

PWM output is generated using one TGR as the cycle register and the others as duty cycl
registers. The output specified in TIOR is performed by means of compare matches. Upo
clearing by a cycle register compare match, the output value of each pin is the initial valu
TIOR. If the set values of the cycle and duty cycle registers are identical, the output valu
change when a compare match occurs.

In PWM mode 2, a maximum 15-phase PWM output is possible by combined use with 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 cy



[3] Use TIOR to designate TGR as an output compare register, and select the initial value and output value.

[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

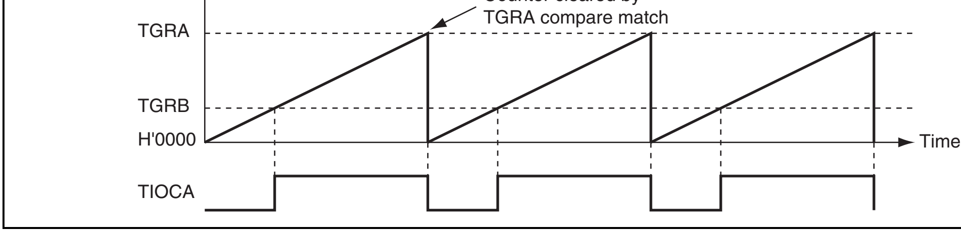


Figure 10.21   Example of PWM Mode Operation (1)

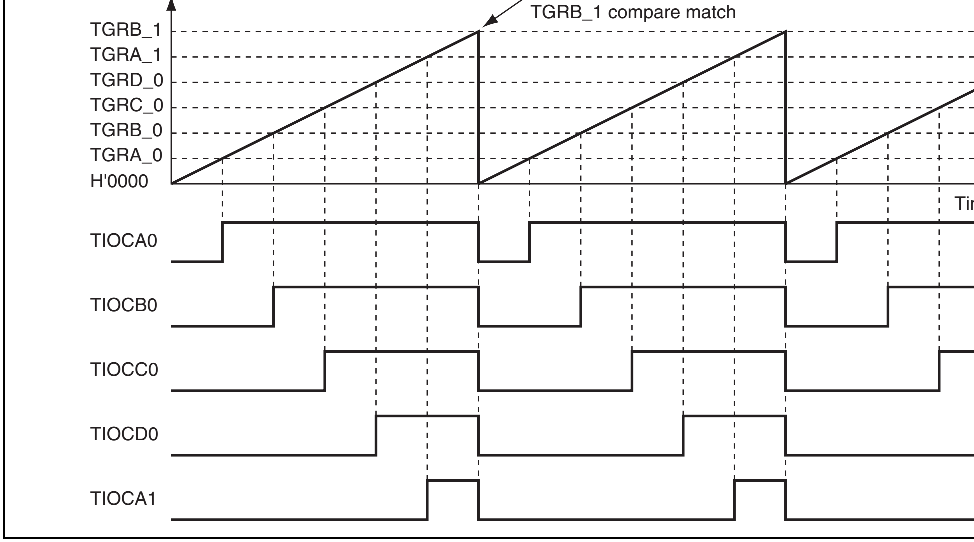


Figure 10.22 Example of PWM Mode Operation (2)

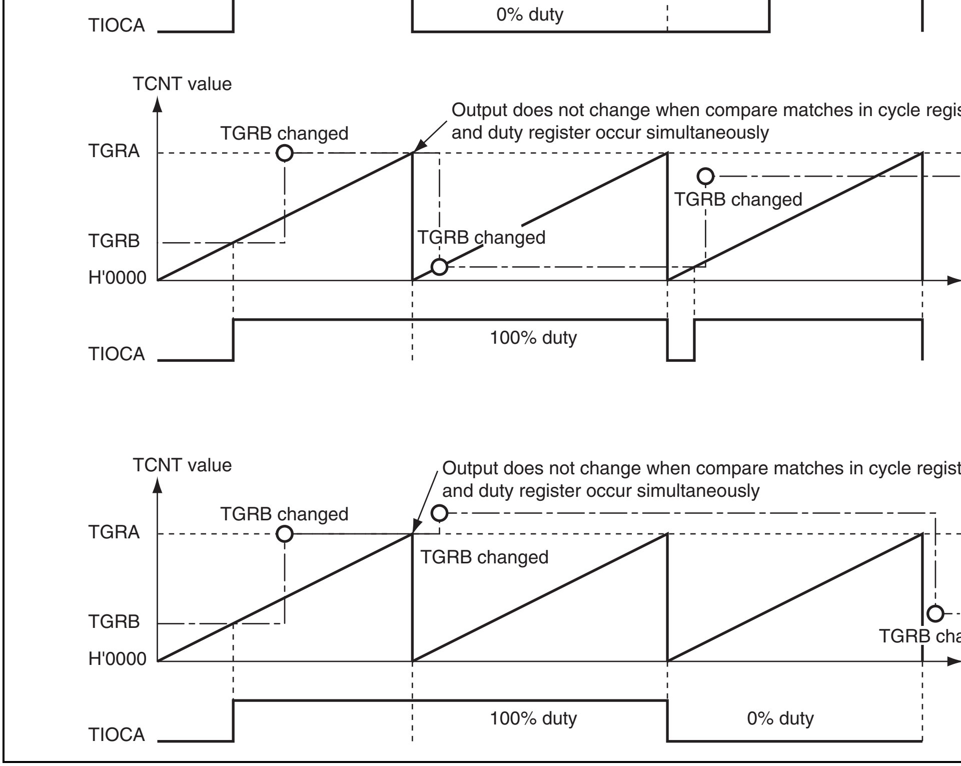

TIOCA
0% duty
TCNT value
Output does not change when compare matches in cycle regis
and duty register occur simultaneously
TGRB changed
TGRA
TGRB changed
TGRB
TGRB changed
H'0000
100% duty
TIOCA
TCNT value
Output does not change when compare matches in cycle regist
and duty register occur simultaneously
TGRB changed
TGRA
TGRB changed
TGRB
TGRB cha
H'0000
100% duty
0% duty
TIOCA

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.

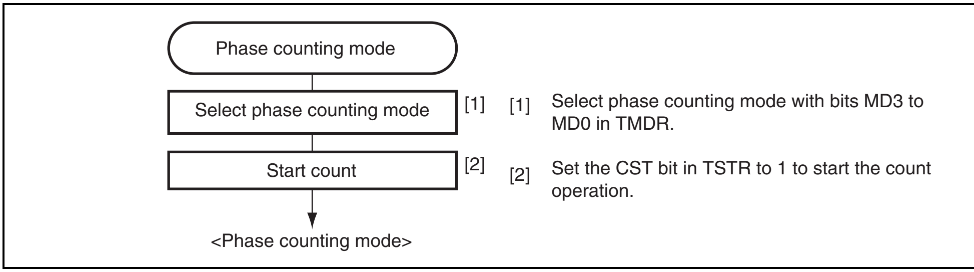
Table 10.32 shows the correspondence between external clock pins and channels.

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

(1) Example of Phase Counting Mode Setting Procedure

Figure 10.24 shows an example of the phase counting mode setting procedure.



[1] Select phase counting mode with bits MD3 to MD0 in TMDR.

[2] Set the CST bit in TSTR to 1 to start the count operation.

Figure 10.24 Example of Phase Counting Mode Setting Procedure

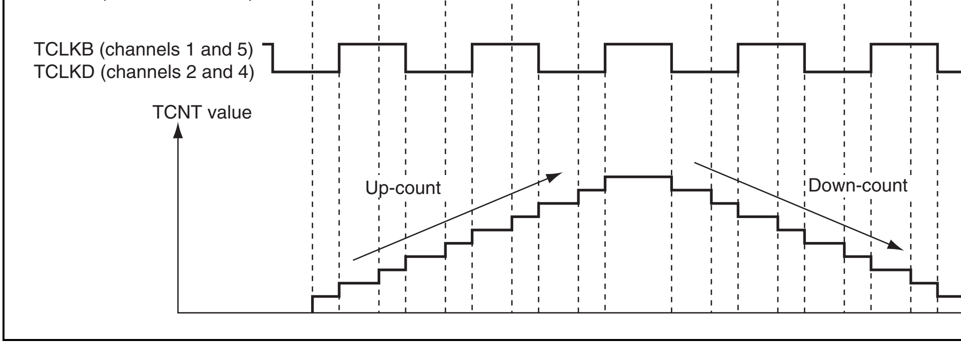


Figure 10.25 Example of Phase Counting Mode 1 Operation

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

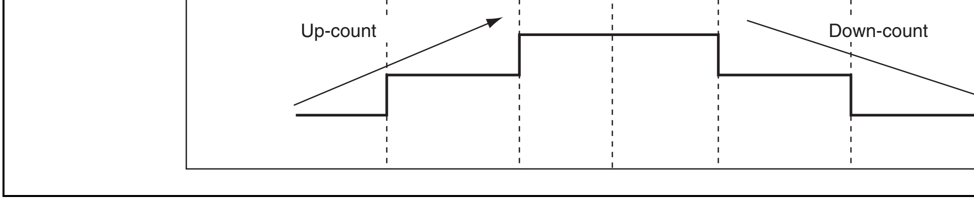


Figure 10.26 Example of Phase Counting Mode 2 Operation

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

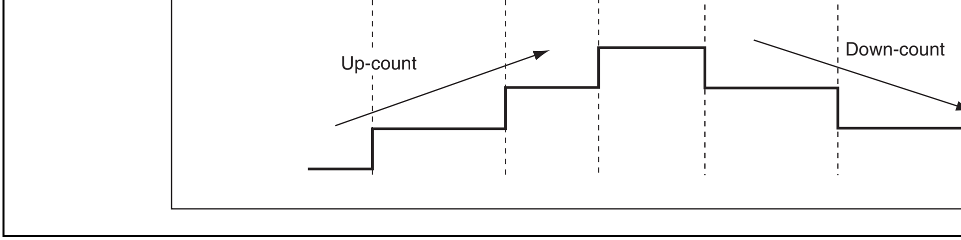


Figure 10.27 Example of Phase Counting Mode 3 Operation

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

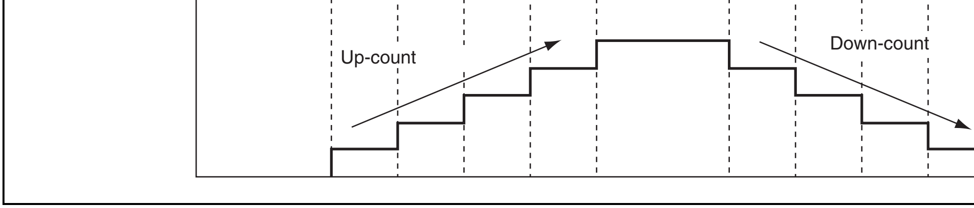


Figure 10.28 Example of Phase Counting Mode 4 Operation

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	

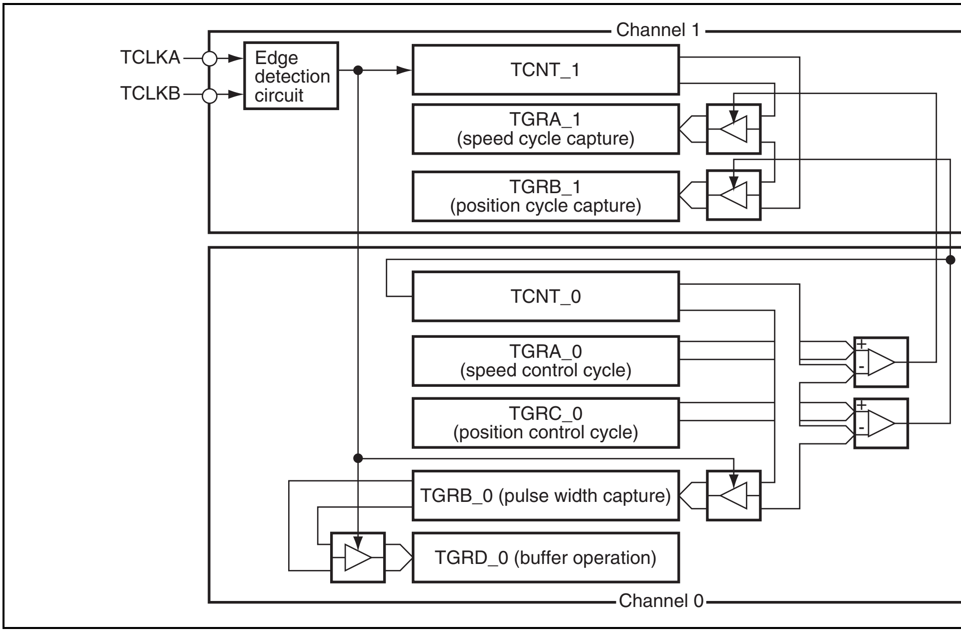
[Legend]

↑:	Rising edge
↓:	Falling edge

position control cycle. TGRB_0 is used for input capture, with TGRB_0 and TGRD_0 o
in buffer mode. The channel 1 counter input clock is designated as the TGRB_0 input ca
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-co
values for the control cycles are stored.

This procedure enables accurate position/speed detection to be achieved.



Channel 1
TCLKA
TCLKB
Edge detection circuit
TCNT_1
TGRA_1 (speed cycle capture)
TGRB_1 (position cycle capture)
TCNT_0
TGRA_0 (speed control cycle)
TGRC_0 (position control cycle)
TGRB_0 (pulse width capture)
TGRD_0 (buffer operation)
+
-
+
-
Channel 0

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	DMAC Activat
0	TGI0A	TGRA_0 input capture/ compare match	TGFA_0	Possible	Possibl
	TGI0B	TGRB_0 input capture/ compare match	TGFB_0	Possible	Not pos
	TGI0C	TGRC_0 input capture/ compare match	TGFC_0	Possible	Not pos
	TGI0D	TGRD_0 input capture/ compare match	TGFD_0	Possible	Not pos
	TCI0V	TCNT_0 overflow	TCFV_0	Not possible	Not pos
1	TGI1A	TGRA_1 input capture/ compare match	TGFA_1	Possible	Possibl
	TGI1B	TGRB_1 input capture/ compare match	TGFB_1	Possible	Not pos
	TCI1V	TCNT_1 overflow	TCFV_1	Not possible	Not pos
	TCI1U	TCNT_1 underflow	TCFU_1	Not possible	Not pos

	TGI3B	TGRB_3 input capture/ compare match	TGFB_3	Possible	Not po
	TGI3C	TGRC_3 input capture/ compare match	TGFC_3	Possible	Not po
	TGI3D	TGRD_3 input capture/ compare match	TGFD_3	Possible	Not po
	TCI3V	TCNT_3 overflow	TCFV_3	Not possible	Not po
4	TGI4A	TGRA_4 input capture/ compare match	TGFA_4	Possible	Possib
	TGI4B	TGRB_4 input capture/ compare match	TGFB_4	Possible	Not po
	TCI4V	TCNT_4 overflow	TCFV_4	Not possible	Not po
	TCI4U	TCNT_4 underflow	TCFU_4	Not possible	Not po
5	TGI5A	TGRA_5 input capture/ compare match	TGFA_5	Possible	Possib
	TGI5B	TGRB_5 input capture/ compare match	TGFB_5	Possible	Not po
	TCI5V	TCNT_5 overflow	TCFV_5	Not possible	Not po
	TCI5U	TCNT_5 underflow	TCFU_5	Not possible	Not po

Note: This table shows the initial state immediately after a reset. The relative channel p levels can be changed by the interrupt controller.

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.

10.6 DTC Activation

The DTC can be activated by the TGR input capture/compare match interrupt for a chann details, see section 8, Data Transfer Controller (DTC).

A total of 16 TPU input capture/compare match interrupts can be used as DTC activation four each for channels 0 and 3, and two each for channels 1, 2, 4, and 5.

10.7 DMAC Activation

The DMAC can be activated by the TGRA input capture/compare match interrupt for a cl For details, see section 7, DMA Controller (DMAC).

In TPU, one in each channel, totally six TGRA input capture/compare match interrupts ca used as DMAC activation sources.

10.9 Operation Timing

10.9.1 Input/Output Timing

(1) TCNT Count Timing

Figure 10.30 shows TCNT count timing in internal clock operation, and figure 10.31 sho
count timing in external clock operation.

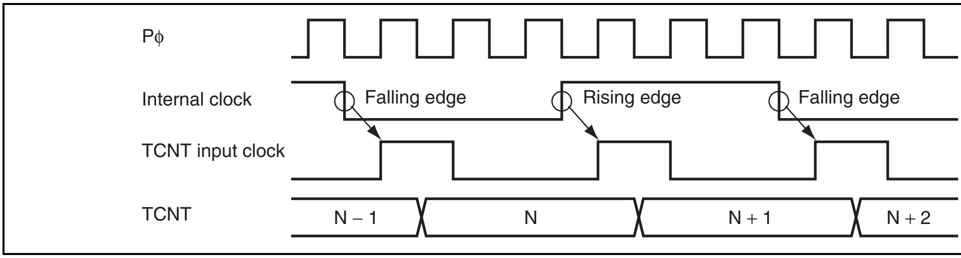


Figure 10.30 Count Timing in Internal Clock Operation

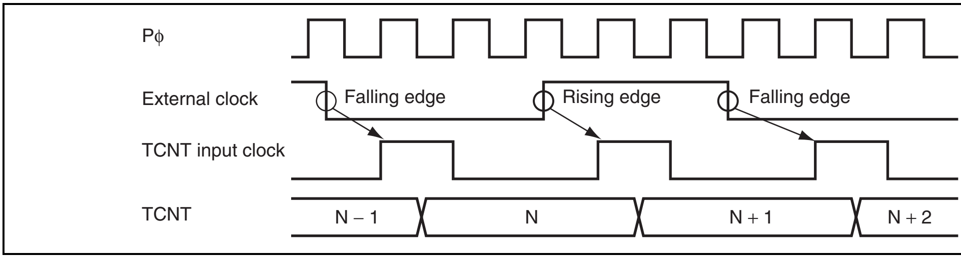


Figure 10.31 Count Timing in External Clock Operation

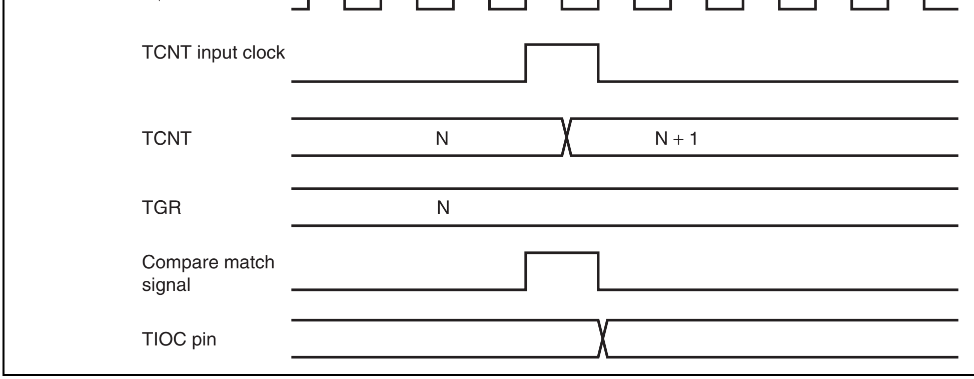


Figure 10.32 Output Compare Output Timing

(3) Input Capture Signal Timing

Figure 10.33 shows input capture signal timing.

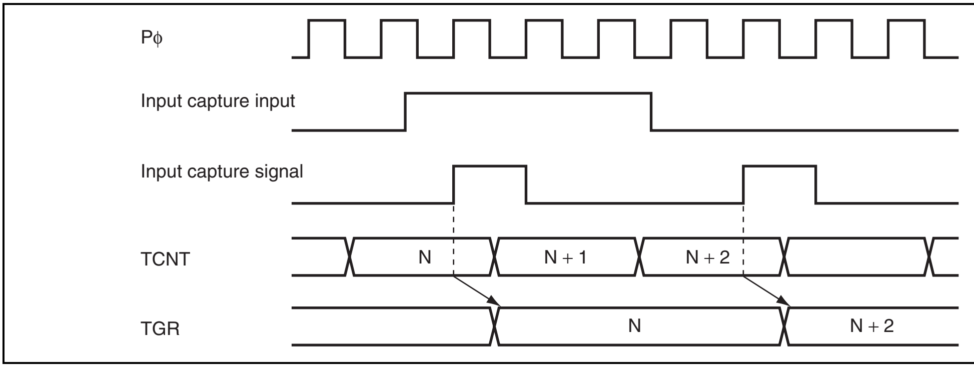


Figure 10.33 Input Capture Input Signal Timing

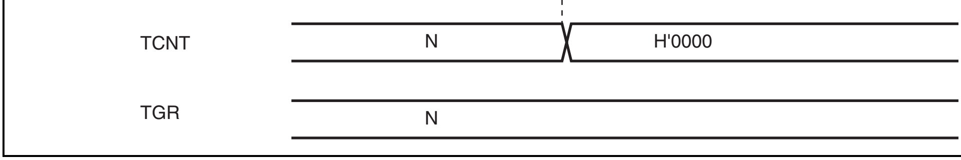


Figure 10.34 Counter Clear Timing (Compare Match)

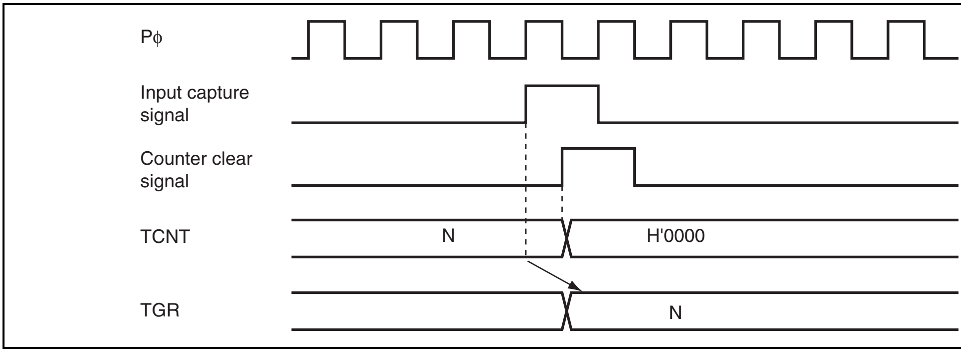


Figure 10.35 Counter Clear Timing (Input Capture)

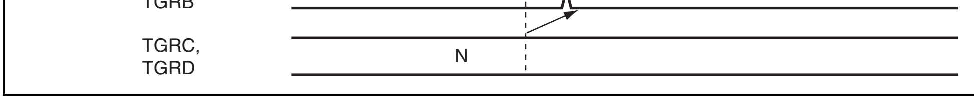


Figure 10.36 Buffer Operation Timing (Compare Match)

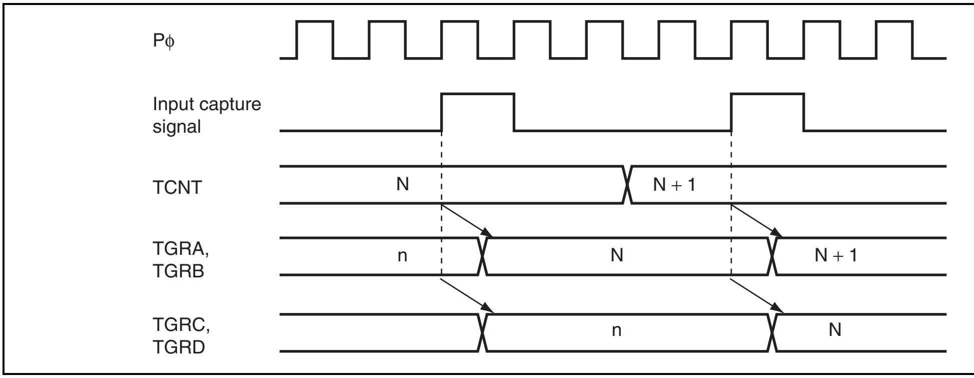


Figure 10.37 Buffer Operation Timing (Input Capture)

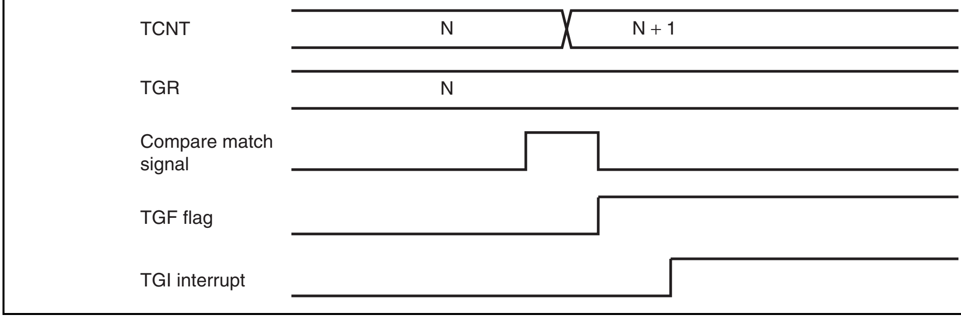


Figure 10.38 TGI Interrupt Timing (Compare Match)

(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 occur the TGI interrupt request signal timing.

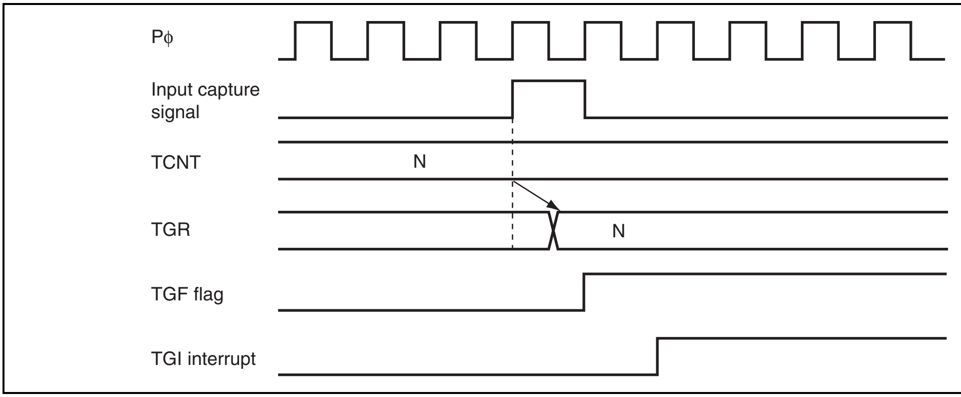


Figure 10.39 TGI Interrupt Timing (Input Capture)

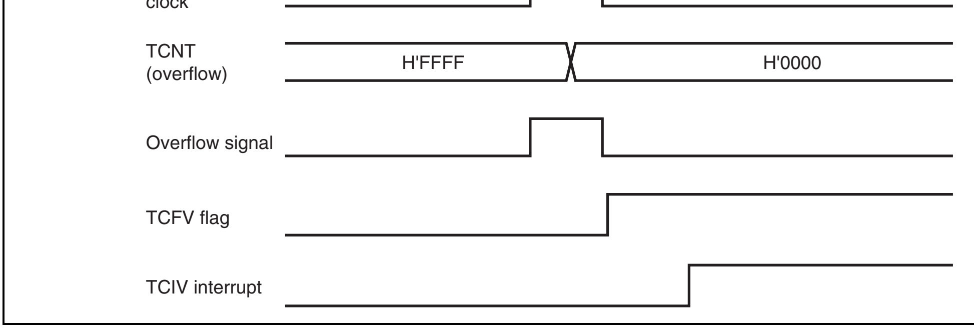


Figure 10.40 TCIV Interrupt Setting Timing

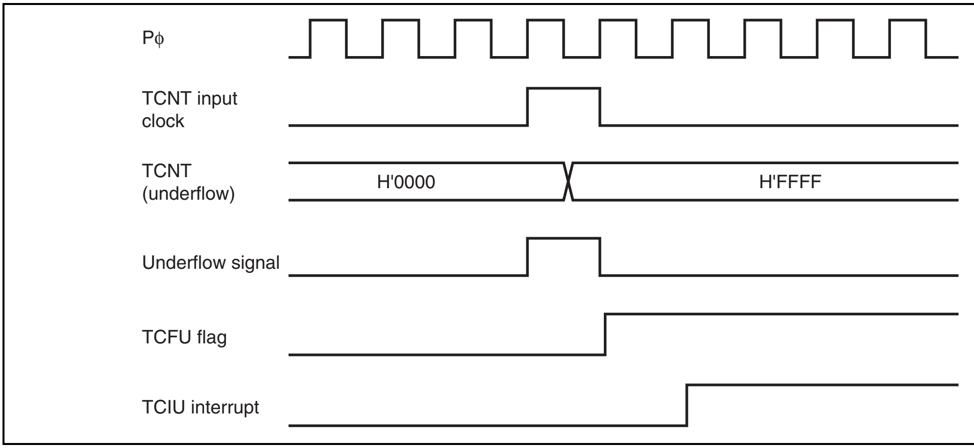


Figure 10.41 TCIU Interrupt Setting Timing

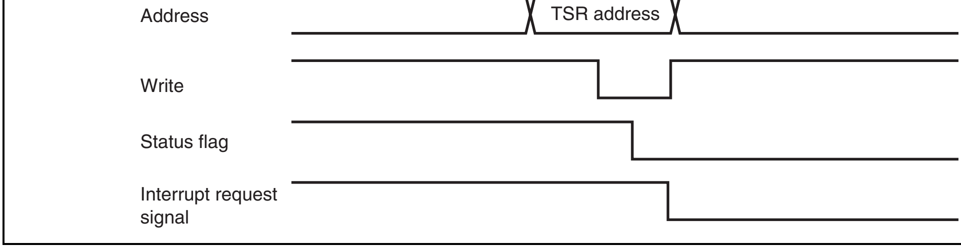


Figure 10.42 Timing for Status Flag Clearing by CPU

The status flag and interrupt request signal are cleared in synchronization with Pϕ after t
DMAC transfer has started, as shown in figure 10.43. If conflict occurs for clearing the s
and interrupt request signal due to activation of multiple DTC or DMAC transfers, it wil
to five clock cycles (Pϕ) for clearing them, as shown in figure 10.44. The next transfer r
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 s
may be cleared during outputting the destination address.

Figure 10.43 Timing for Status Flag Clearing by DTC or DMAC Activation

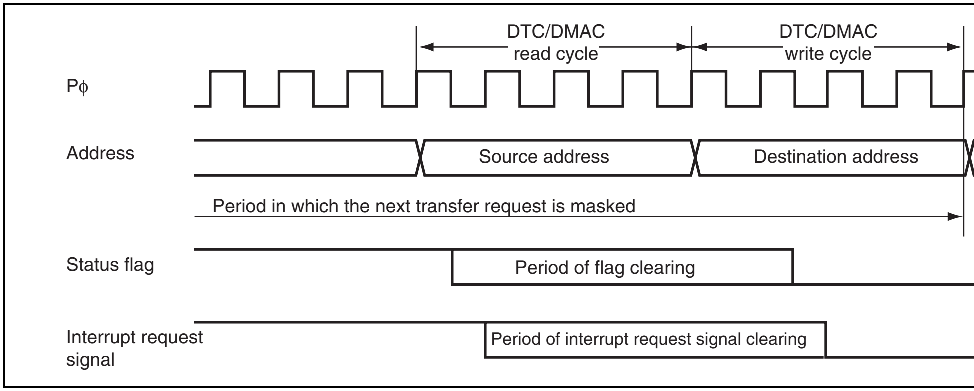
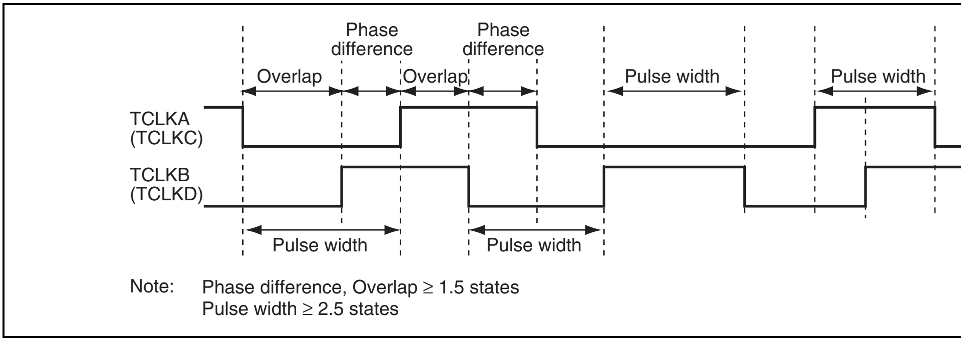


Figure 10.44 Timing for Status Flag Clearing by DTC or DMAC Activation

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 wi narrower pulse width.

In phase counting mode, the phase difference and overlap between the two input clocks least 1.5 states, and the pulse width must be at least 2.5 states. Figure 10.45 shows the in conditions in phase counting mode.



Note: Phase difference, Overlap ≥ 1.5 states
Pulse width ≥ 2.5 states

Figure 10.45 Phase Difference, Overlap, and Pulse Width in Phase Counting

N: TGR 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 cl
takes precedence and the TCNT write is not performed. Figure 10.46 shows the timing in
case.

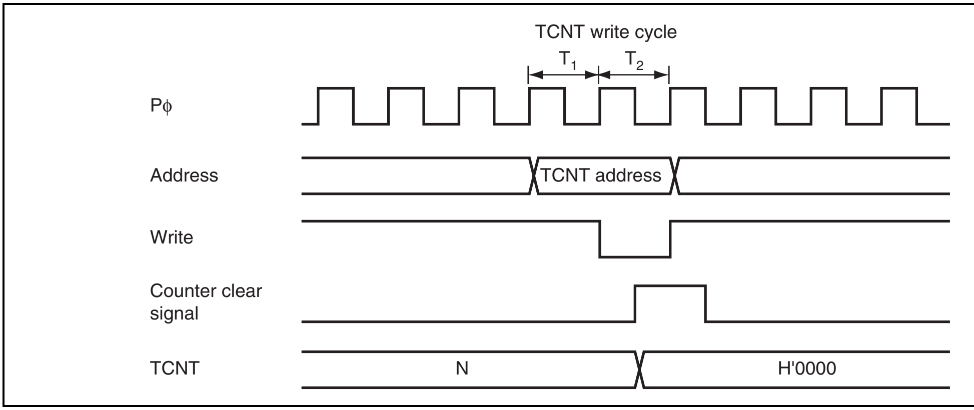


Figure 10.46 Conflict between TCNT Write and Clear Operations

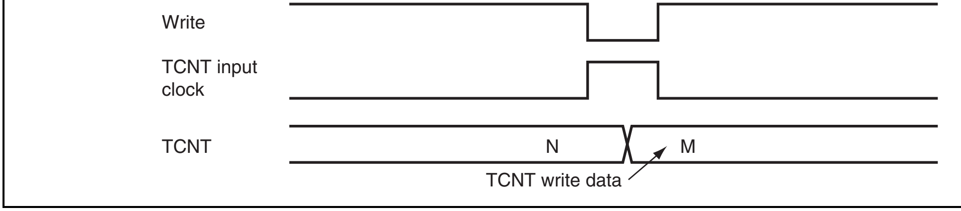


Figure 10.47 Conflict between TCNT Write and Increment Operations

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 pre and the compare match signal is disabled. A compare match also does not occur when th value as before is written.

Figure 10.48 shows the timing in this case.

Figure 10.48 Conflict between TGR Write and Compare Match

10.10.7 Conflict between Buffer Register Write and Compare Match

If a compare match occurs in the T2 state of a TGR write cycle, the data transferred to T
buffer operation will be the write data.

Figure 10.49 shows the timing in this case.

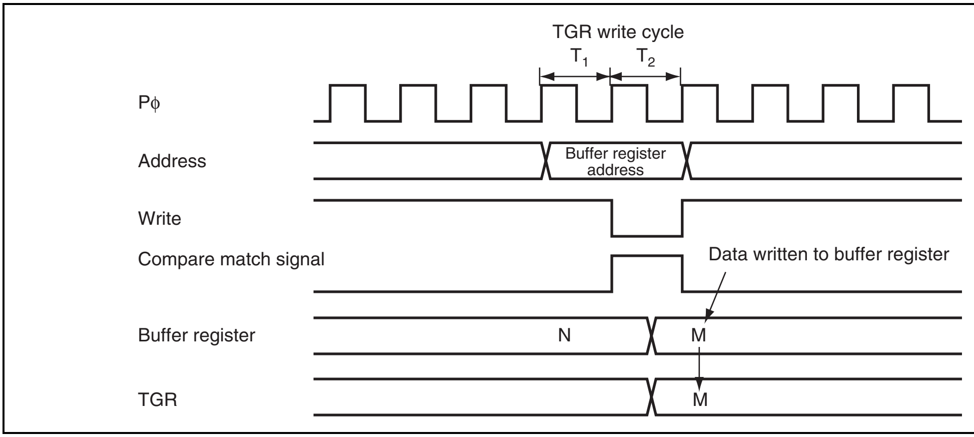


Figure 10.49 Conflict between Buffer Register Write and Compare Match

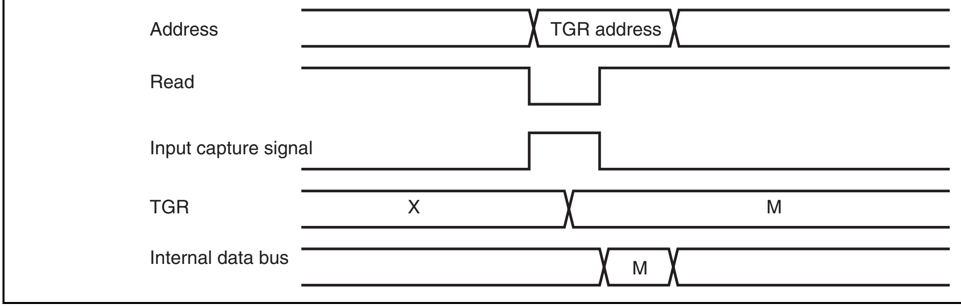


Figure 10.50 Conflict between TGR Read and Input Capture

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 ca 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

10.10.10 Conflict between Buffer Register Write and Input Capture

If the input capture signal is generated in the T2 state of a buffer register write cycle, the operation takes precedence and the write to the buffer register is not performed.

Figure 10.52 shows the timing in this case.

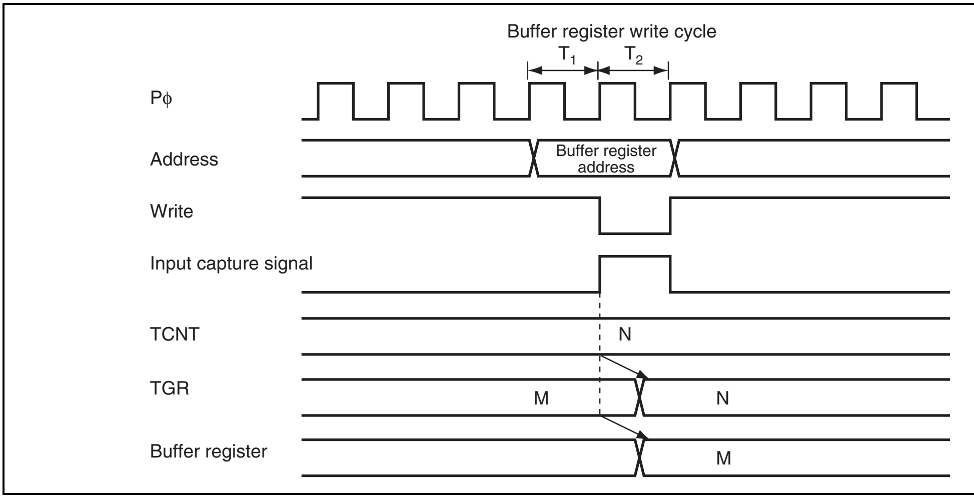


Figure 10.52 Conflict between Buffer Register Write and Input Capture

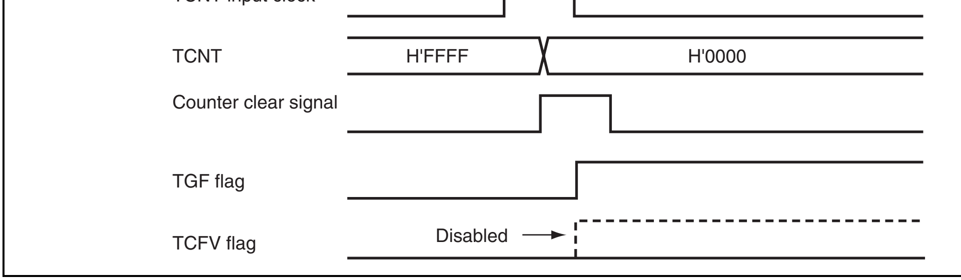


Figure 10.53 Conflict between Overflow and Counter Clearing

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 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

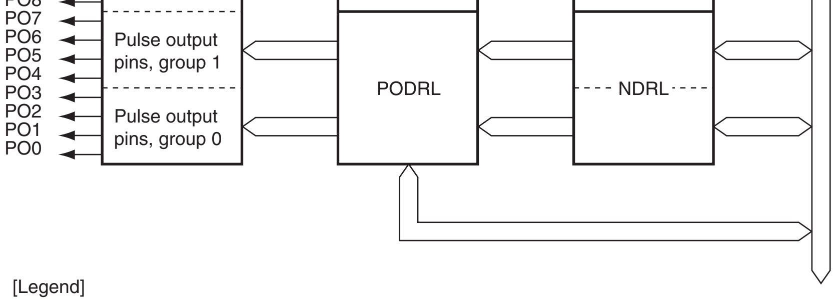
10.10.13 Multiplexing of I/O Pins

In this LSI, the TCLKA input pin is multiplexed with the TIOCC0 I/O pin, the TCLKB in with the TIOCD0 I/O pin, the TCLKC input pin with the TIOCB1 I/O pin, and the TCLK pin with the TIOCB2 I/O pin. When an external clock is input, compare match output sho be performed from a multiplexed pin.

10.10.14 Interrupts and Module Stop Mode

If module stop mode is entered when an interrupt has been requested, it will not be possib clear the CPU interrupt source or the DMAC or DTC activation source. Interrupts should be disabled before entering module stop mode.

- 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 mode can be set



[Legend]

PMR:	PPG output mode register
PCR:	PPG output control register
NDERH:	Next data enable register H
NDERL:	Next data enable register L
NDRH:	Next data register H
NDRL:	Next data register L
PODRH:	Output data register H
PODRL:	Output data register L

Figure 11.1 Block Diagram of PPG

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 correspo NDRL bit is transferred to the PODRL bit by the output trigger. Values are not transferred from N PODRL for cleared bits.
5	NDER5	0	R/W	
4	NDER4	0	R/W	
3	NDER3	0	R/W	
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	POD2
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 the output trigger transfers NDRH values to this during PPG operation. While NDERH is set to 1, cannot write to this register. While NDERH is cle initial output value of the pulse can be set.
5	POD13	0	R/W	
4	POD12	0	R/W	
3	POD11	0	R/W	
2	POD10	0	R/W	
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 depen
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

3	NDR11	0	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 lo 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 The register contents are transferred to the corresponding PODRH bits by the output trigger with PCR.
6	NDR14	0	R/W	
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 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	NDR11	0	R/W	Next Data Register 11 to 8 The register contents are transferred to the corresponding PODRH bits by the output trigger with PCR.
2	NDR10	0	R/W	
1	NDR9	0	R/W	
0	NDR8	0	R/W	

3	NDR3	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 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 trigge 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

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	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 trigge with PCR.
1	NDR1	0	R/W	
0	NDR0	0	R/W	

Bit	Bit Name	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 g
				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 g
				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 g
				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 g
				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

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 puls group 3. 0: Inverted output 1: Direct output
6	G2INV	1	R/W	Group 2 Inversion Selects direct output or inverted output for puls group 2. 0: Inverted output 1: Direct output
5	G1INV	1	R/W	Group 1 Inversion Selects direct output or inverted output for puls group 1. 0: Inverted output 1: Direct output
4	G0INV	1	R/W	Group 0 Inversion Selects direct output or inverted output for puls group 0. 0: Inverted output 1: Direct output

				Selects normal or non-overlapping operation for output group 2. 0: Normal operation (output values updated at co match A in the selected TPU channel) 1: Non-overlapping operation (output values upd compare match A or B in the selected TPU ch
1	G1NOV	0	R/W	Group 1 Non-Overlap Selects normal or non-overlapping operation for output group 1. 0: Normal operation (output values updated at co match A in the selected TPU channel) 1: Non-overlapping operation (output values upd compare match A or B in the selected TPU ch
0	G0NOV	0	R/W	Group 0 Non-Overlap Selects normal or non-overlapping operation for output group 0. 0: Normal operation (output values updated at co match A in the selected TPU channel) 1: Non-overlapping operation (output values upd compare match A or B in the selected TPU ch

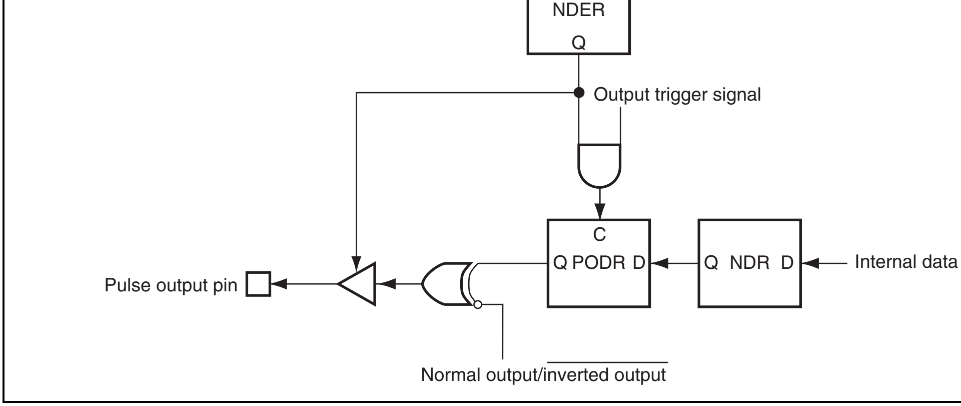


Figure 11.2 Schematic Diagram of PPG

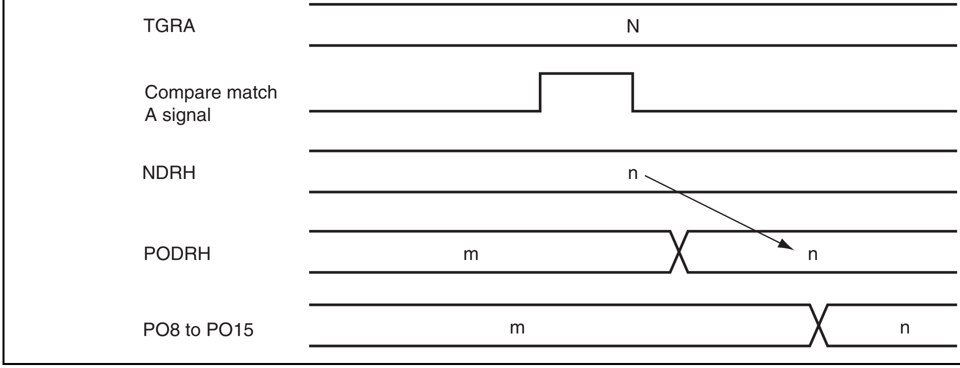
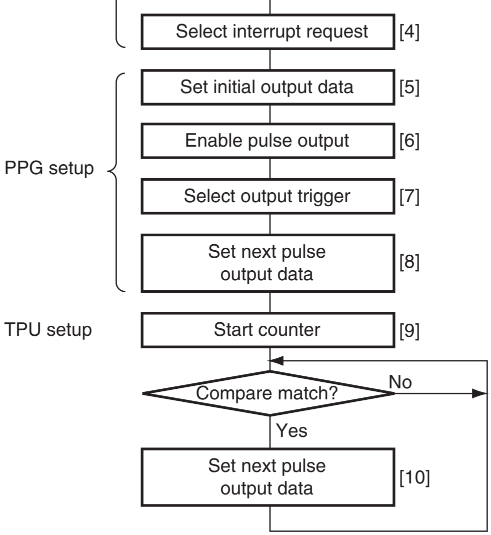


Figure 11.3 Timing of Transfer and Output of NDR Contents (Example)



[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)

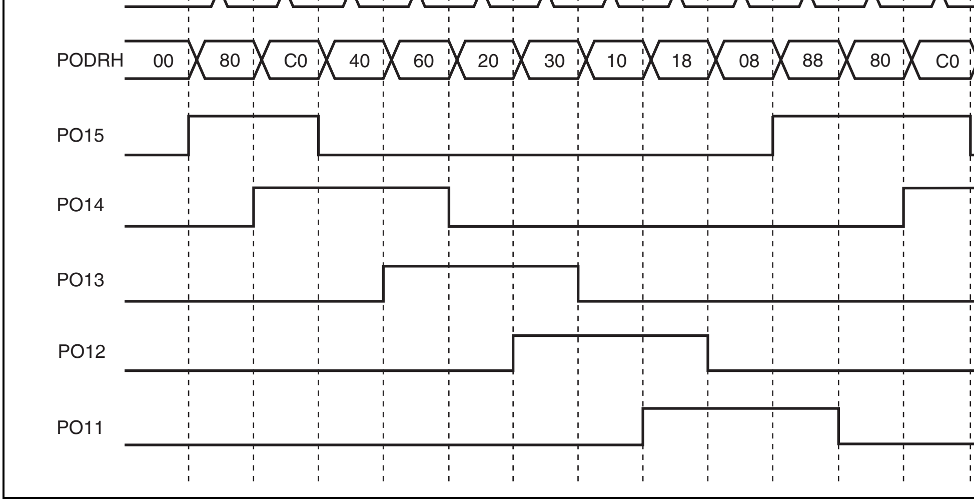


Figure 11.5 Normal Pulse Output Example (5-Phase Pulse Output)

writing H'40, H'60, H'20, H'30, H'10, H'18, H'08, H'88... at successive TGIA interru

5. If the DTC or DMAC is set for activation by the TGIA interrupt, pulse output can be without imposing a load on the CPU.

11.4.4 Non-Overlapping Pulse Output

During non-overlapping operation, transfer from NDR to PODR is performed as follows

- At compare match A, the NDR bits are always transferred to PODR.
- At compare match B, the NDR bits are transferred only if their value is 0. The NDR not transferred if their value is 1.

Figure 11.6 illustrates the non-overlapping pulse output operation.

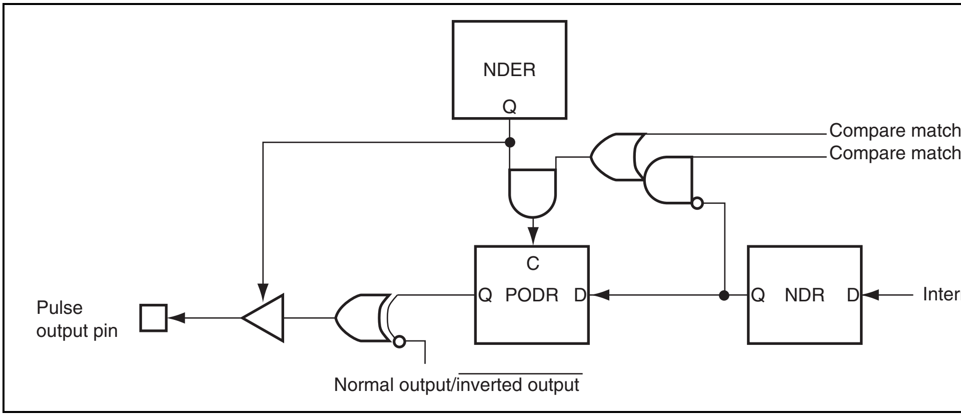


Figure 11.6 Non-Overlapping Pulse Output

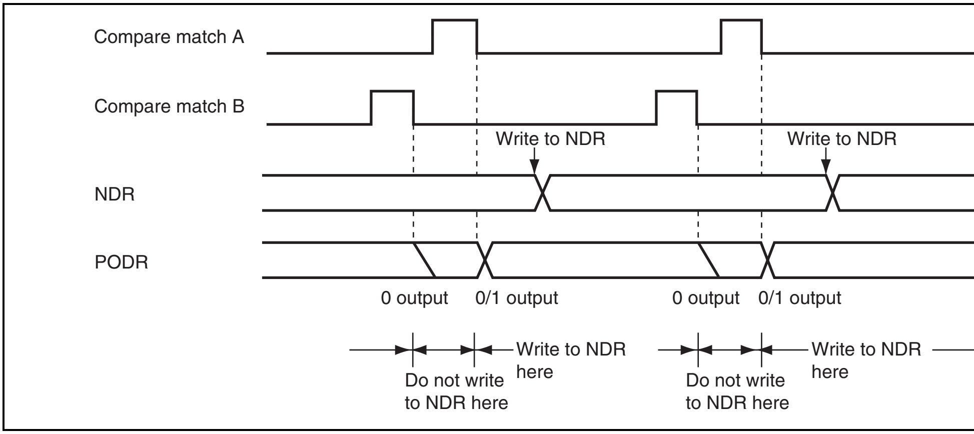
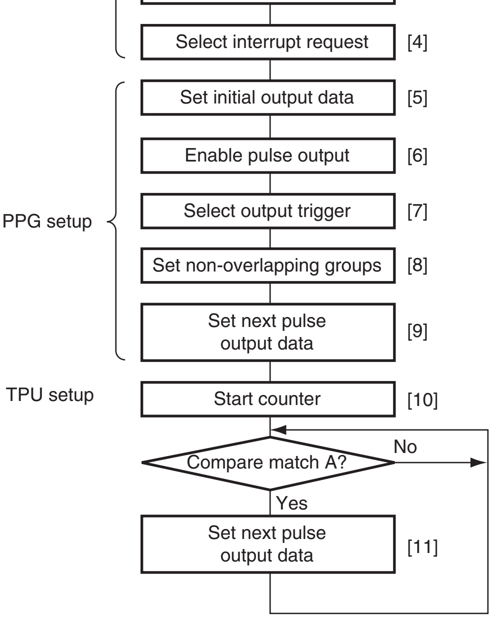


Figure 11.7 Non-Overlapping Operation and NDR Write Timing



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

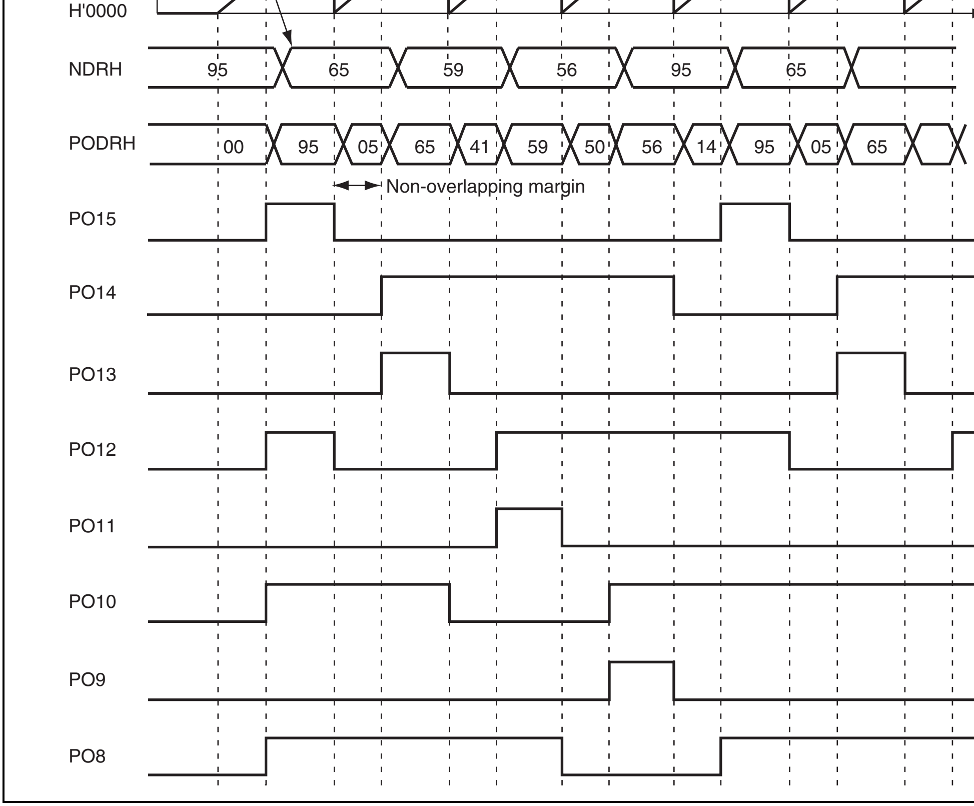


Figure 11.9 Non-Overlapping Pulse Output Example (4-Phase Complementa

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.

4. 4-phase complementary non-overlapping pulse output can be obtained subsequently H'59, H'56, H'95... at successive TGIA interrupts.

   If the DTC or DMAC is set for activation by a TGIA interrupt, pulse can be output v imposing a load on the CPU.

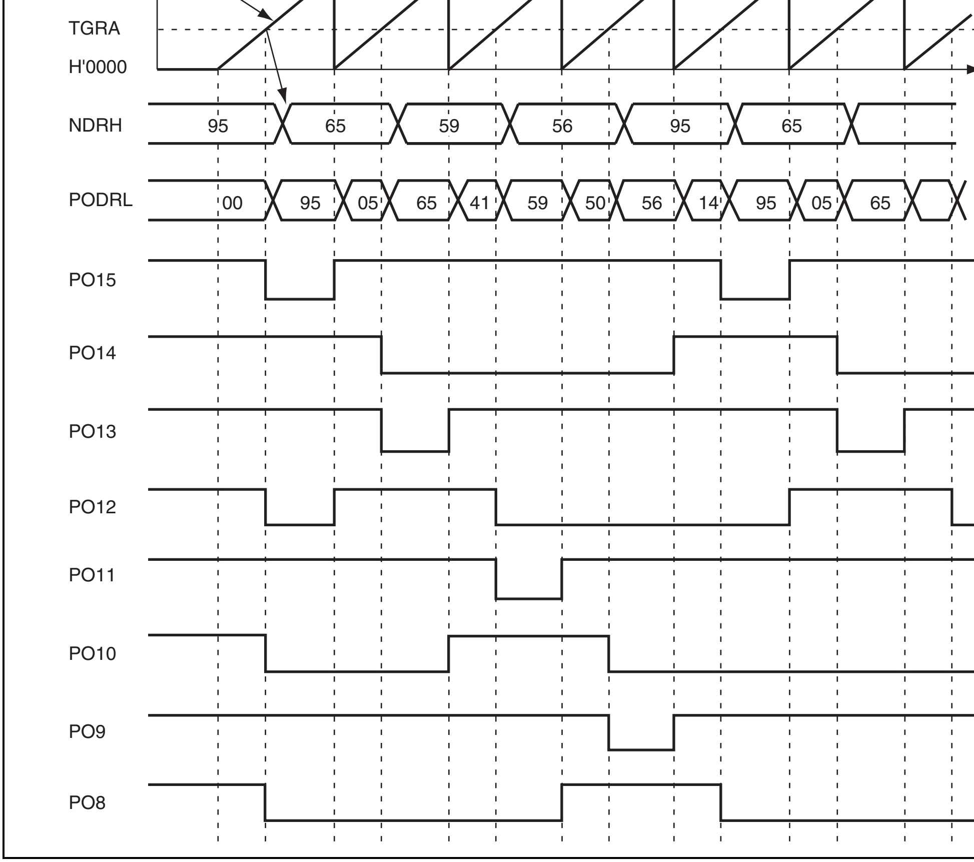


Figure 11.10 Inverted Pulse Output (Example)

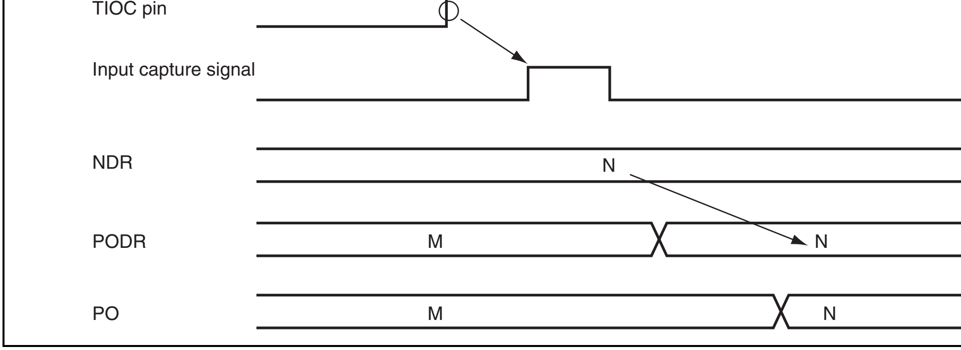


Figure 11.11   Pulse Output Triggered by Input Capture (Example)

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 v occur.

12.1 Features

- Selection of seven clock sources

  The counters can be driven by one of six internal clock signals (Pϕ/2, Pϕ/18, Pϕ/32,
  Pϕ/1024, or Pϕ/8192) or an external clock input.
- Selection of three ways to clear the counters

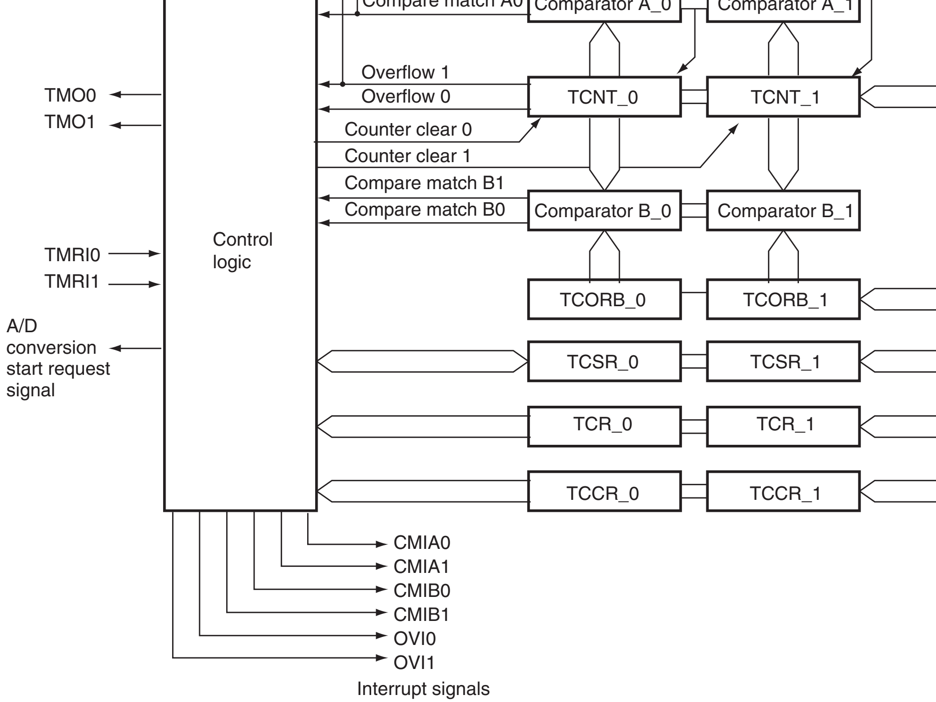
  The counters can be cleared on compare match A or B, or by an external reset signal
- Timer output control by a combination of two compare match signals

  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

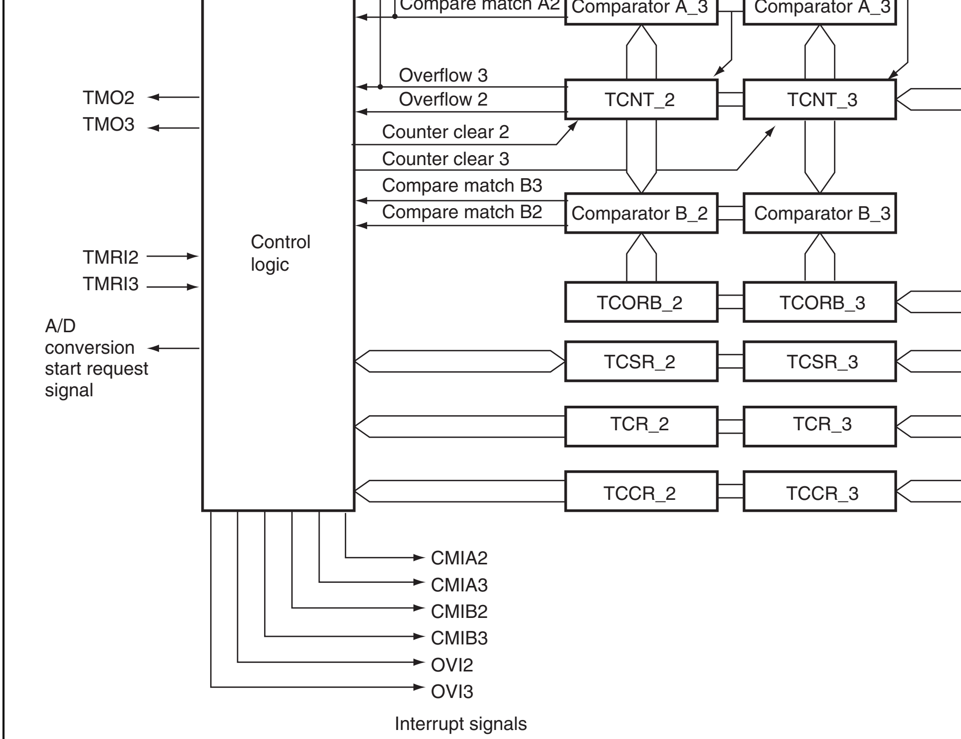
  Compare match A, compare match B, and overflow interrupts can be requested inde
- Generation of trigger to start A/D converter conversion



[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

  - Timer control register_0 (TCR_0)
  - Timer counter control register_0 (TCCR_0)
  - Timer control/status register_0 (TCSR_0)
- Channel 1
  - Timer counter_1 (TCNT_1)
  - Time constant register A_1 (TCORA_1)
  - Time constant register B_1 (TCORB_1)
  - Timer control register_1 (TCR_1)
  - Timer counter control register_1 (TCCR_1)
  - Timer control/status register_1 (TCSR_1)

Unit 1:

- 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	
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	
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

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

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 TC 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 cleared. 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 th external reset input is high (TMRIS in TC to 1)
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.

7 to 4	—	0	R/W	Reserved These bits are always read as 0. The write valu always be 0.
3	TMRIS	0	R/W	Timer Reset Input Select Selects an external reset input when the CCLR 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 sh 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 CK select the internal clock. See table 12.2.

	0	1	0	0	0	Uses internal clock. Counts at rising edge of
				0	1	Uses internal clock. Counts at rising edge of
				1	0	Uses internal clock. Counts at falling edge of
				1	1	Uses internal clock. Counts at falling edge of
	0	1	1	0	0	Uses internal clock. Counts at rising edge of
				0	1	Uses internal clock. Counts at rising edge of
				1	0	Uses internal clock. Counts at falling edge of
				1	1	Uses internal clock. Counts at falling edge of
	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
				0	1	Uses internal clock. Counts at rising edge of
				1	0	Uses internal clock. Counts at falling edge of
				1	1	Uses internal clock. Counts at falling edge of
	0	1	0	0	0	Uses internal clock. Counts at rising edge of
				0	1	Uses internal clock. Counts at rising edge of
				1	0	Uses internal clock. Counts at falling edge of
				1	1	Uses internal clock. Counts at falling edge of
	0	1	1	0	0	Uses internal clock. Counts at rising edge of
				0	1	Uses internal clock. Counts at rising edge of
				1	0	Uses internal clock. Counts at falling edge of
				1	1	Uses internal clock. Counts at falling edge of
	1	0	0	—	—	Counts at TCNT_0 compare match A*[1].

2. To use the external clock, the DDR and ISR bits in the corresponding pin sho
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.

				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 wri while the corresponding interrupt is enabled, be 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 E (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 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, then w to OVF (When the CPU is used to clear this flag by wri while the corresponding interrupt is enabled, be read the flag after writing 0 to it.)

				11: Output is inverted when compare match E (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.

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 TCOR 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.

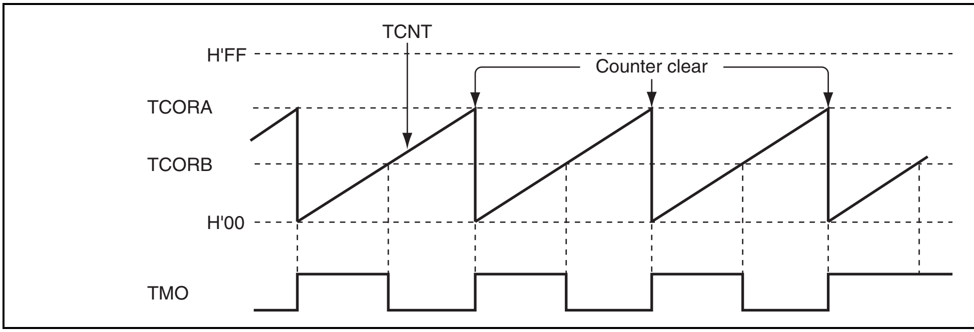


Figure 12.3 Example of Pulse Output

input determined by TCORA and with a pulse width determined by TCORB and TCOR

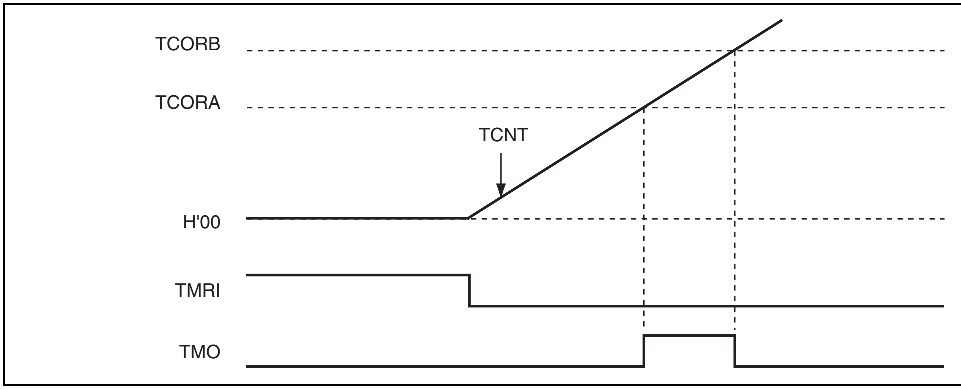


Figure 12.4 Example of Reset Input

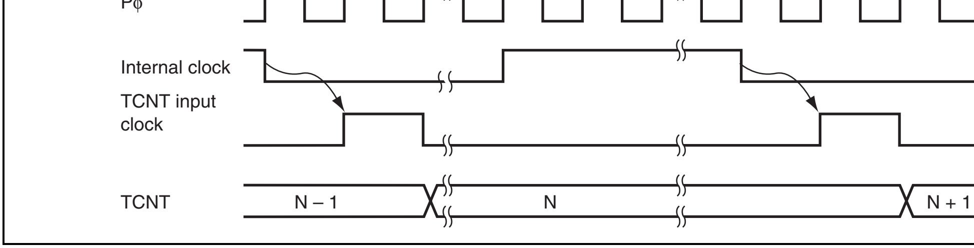


Figure 12.5   Count Timing for Internal Clock Input at Falling Edge

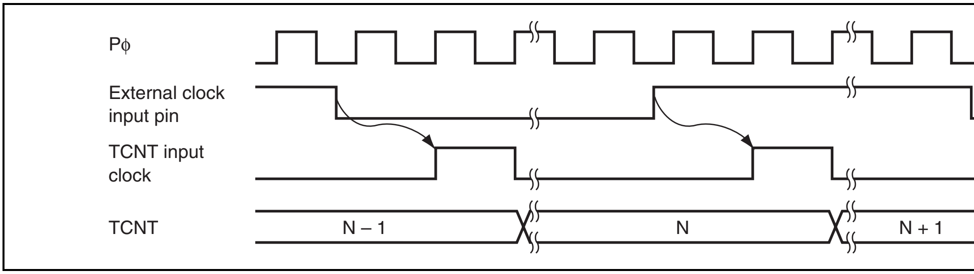


Figure 12.6   Count Timing for External Clock Input at Falling and Rising Ed

12.5.2 Timing of CMFA and CMFB Setting at Compare Match

The CMFA and CMFB flags in TCSR are set to 1 by a compare match signal generated v TCOR and TCNT values match. The compare match signal is generated at the last state ir the match is true, just before the timer counter is updated. Therefore, when the TCOR anc values match, the compare match signal is not generated until the next TCNT clock input 12.7 shows this timing.

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 cor match A signal.

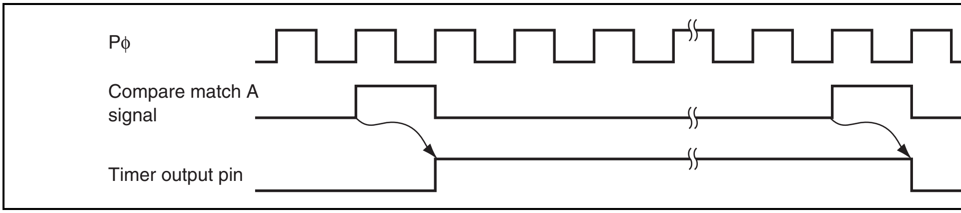


Figure 12.8 Timing of Toggled Timer Output at Compare Match A

12.5.4 Timing of Counter Clear by Compare Match

TCNT is cleared when compare match A or B occurs, depending on the settings of bits C and CCLR0 in TCR. Figure 12.9 shows the timing of this operation.

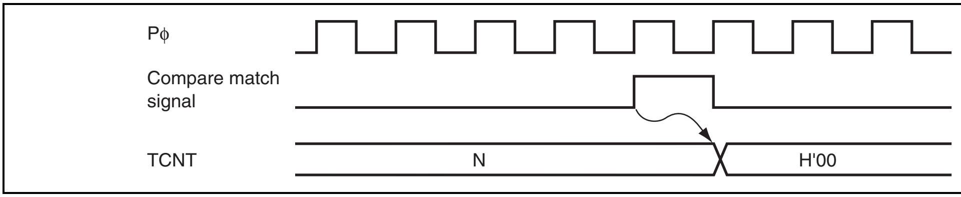


Figure 12.9 Timing of Counter Clear by Compare Match

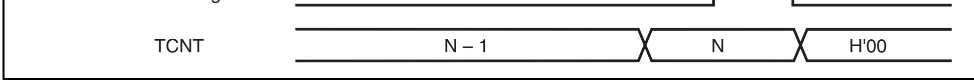


Figure 12.10 Timing of Clearance by External Reset (Rising Edge)

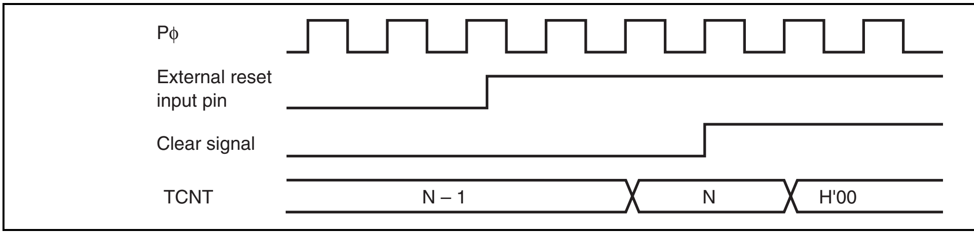


Figure 12.11 Timing of Clearance by External Reset (High Level)

12.5.6 Timing of Overflow Flag (OVF) Setting

The OVF bit in TCSR is set to 1 when TCNT overflows (changes from H'FF to H'00). Fi
12.12 shows the timing of this operation.

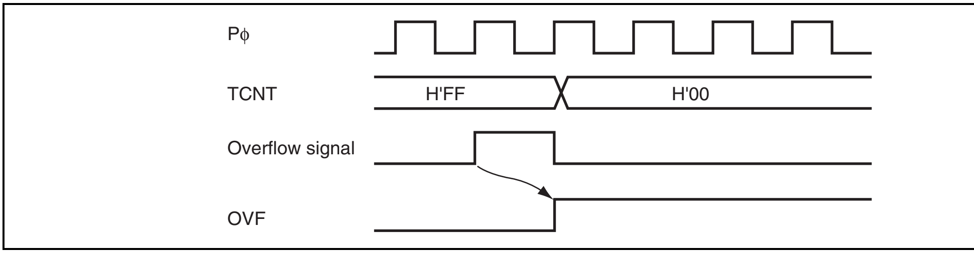


Figure 12.12 Timing of OVF Setting

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	Lo
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	Lo

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 converter. If the 8-bit timer conversion start trigger has been selected on the A/D converte this time, A/D conversion is started.

Pϕ: 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.

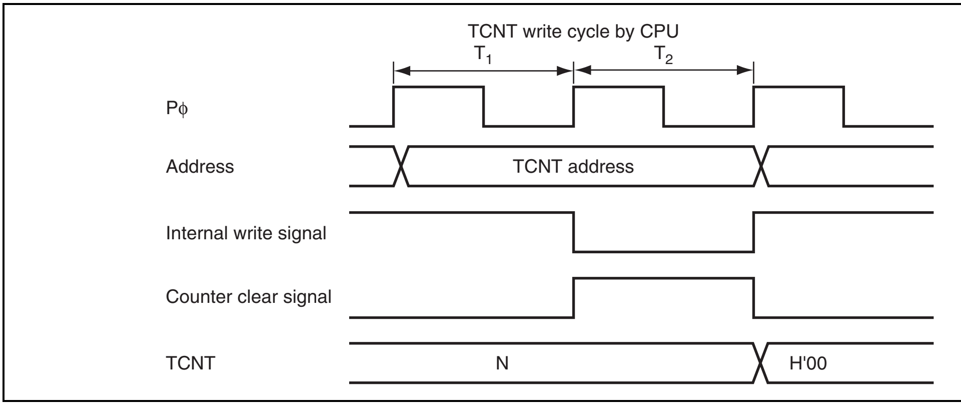


Figure 12.13 Conflict between TCNT Write and Clear

12.8.3 Conflict between TCNT Write and Increment

If a TCNT input clock pulse is generated during the T_2 state of a TCNT write cycle, the
priority and the counter is not incremented as shown in figure 12.14.

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.

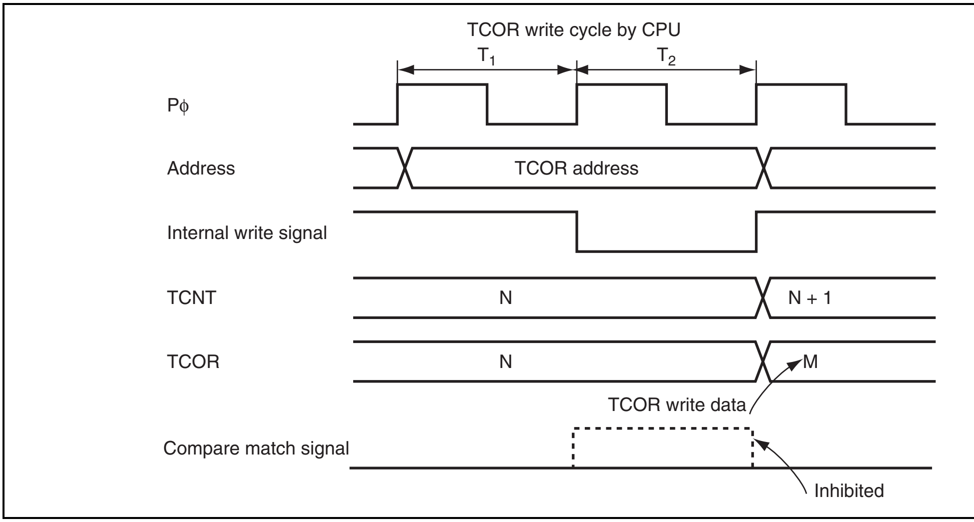


Figure 12.15 Conflict between TCOR Write and Compare Match

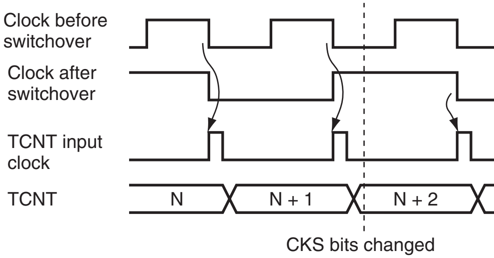
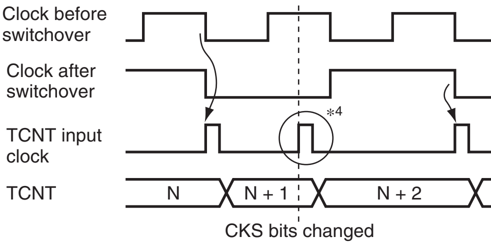
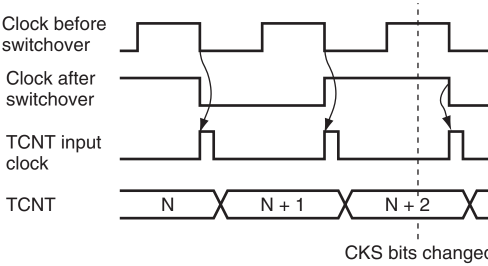
0-output	
No change	Lo

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 generatec
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

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
4	Switching from high to high	 Clock before switchover Clock after switchover 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 e TCNT is incremented.

module stop mode. For details, see section 19, Power-Down Modes.

12.8.9 Interrupts in Module Stop Mode

If module stop mode is entered when an interrupt has been requested, it will not be poss
clear the CPU interrupt source or the DTC activation source. Interrupts should therefore
disabled before entering module stop mode.

13.1 Features

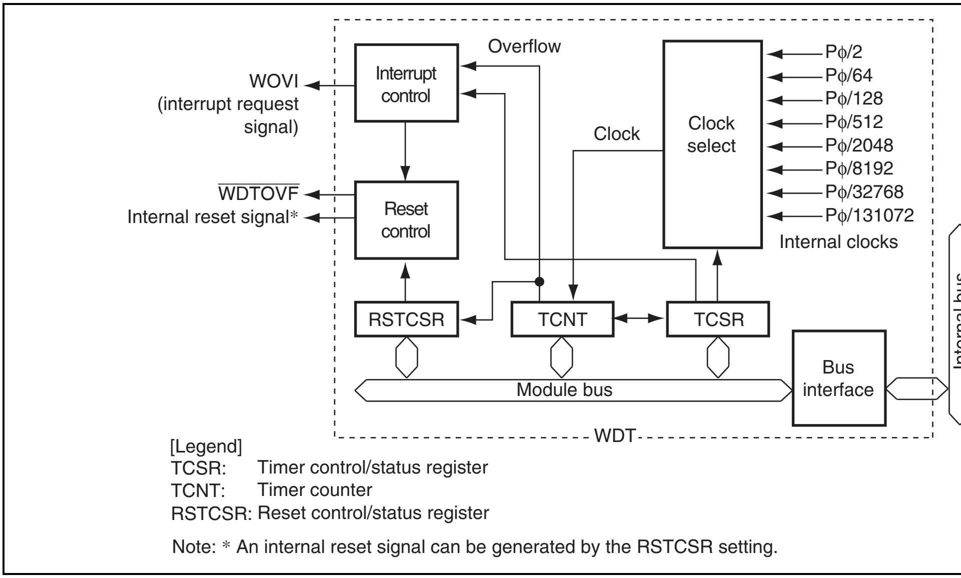
- Selectable from eight counter input clocks
- Switchable between watchdog timer mode and interval timer mode

  — 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 selec watchdog timer mode, OVF is cleared automati the internal reset. [Clearing condition] Cleared by reading TCSR when OVF = 1, then to OVF (When the CPU is used to clear this flag by writ while the corresponding interrupt is enabled, be 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 P φ = 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 then w WOVF (When the CPU is used to clear this flag by writ while the corresponding interrupt is enabled, be read the flag after writing 0 to it.)
6	RSTE	0	R/W	Reset Enable Specifies whether or not this LSI is internally re TCNT overflows during watchdog timer operatio 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.

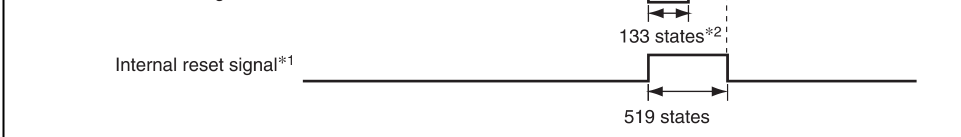
to reset the LSI internally in watchdog timer mode.

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 states with Pϕ when RSTE = 1 in RSTCSR, and f states with Pϕ when RSTE = 0 in RSTCSR. The internal reset signal is output for 519 sta Pϕ.

When the RSTE bit = 1, an internal reset signal is generated. As this signal resets the syst control register (SCKCR), the magnification power of Pϕ to the input clock becomes the value. When the RSTE bit = 0, no internal reset signal is generated. Therefore, the setting SCKCR is retained and the magnification power of Pϕ to the input clock does not change

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.

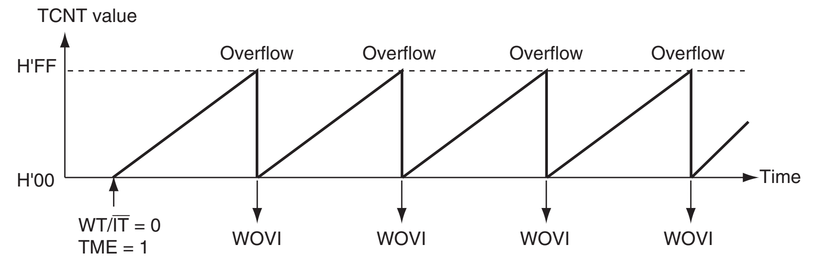
Figure 13.2 Operation in Watchdog Timer Mode

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) i at the same time the OVF bit in the TCSR is set to 1.



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 giv 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 tran 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 the 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.

(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 state of a TCNT write cycle, the write
priority and the timer counter is not incremented. Figure 13.5 shows this operation.

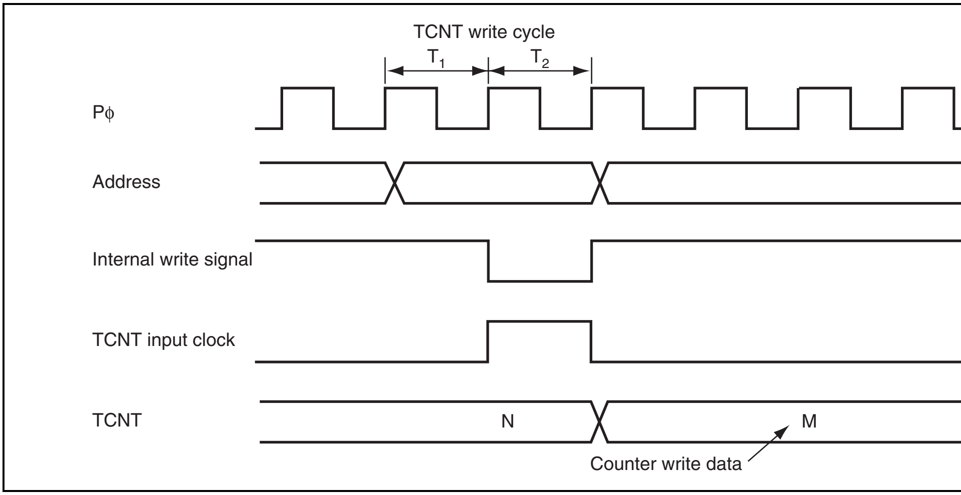


Figure 13.5 Conflict between TCNT Write and Increment

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, and 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. M 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.

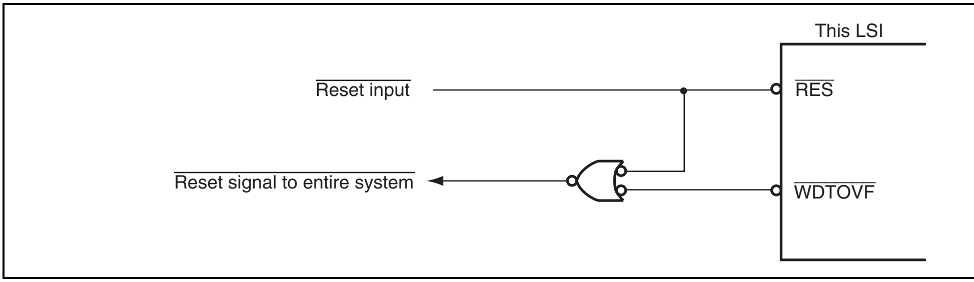

This LSI
$\overline{\text{Reset input}}$
$\overline{\text{RES}}$
$\overline{\text{Reset signal to entire system}}$
$\overline{\text{WDTOVF}}$

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 DMAC.
- Module stop mode can be set

Asynchronous Mode:

- Data length: 7 or 8 bits
- Stop bit length: 1 or 2 bits
- Parity: Even, odd, or none
- Receive error detection: Parity, overrun, and framing errors
- Break detection: Break can be detected by reading the RxD pin level directly in case framing error

- An error signal can be automatically transmitted on detection of a parity error during r
- Data can be automatically re-transmitted on receiving an error signal during transmiss
- Both direct convention and inverse convention are supported

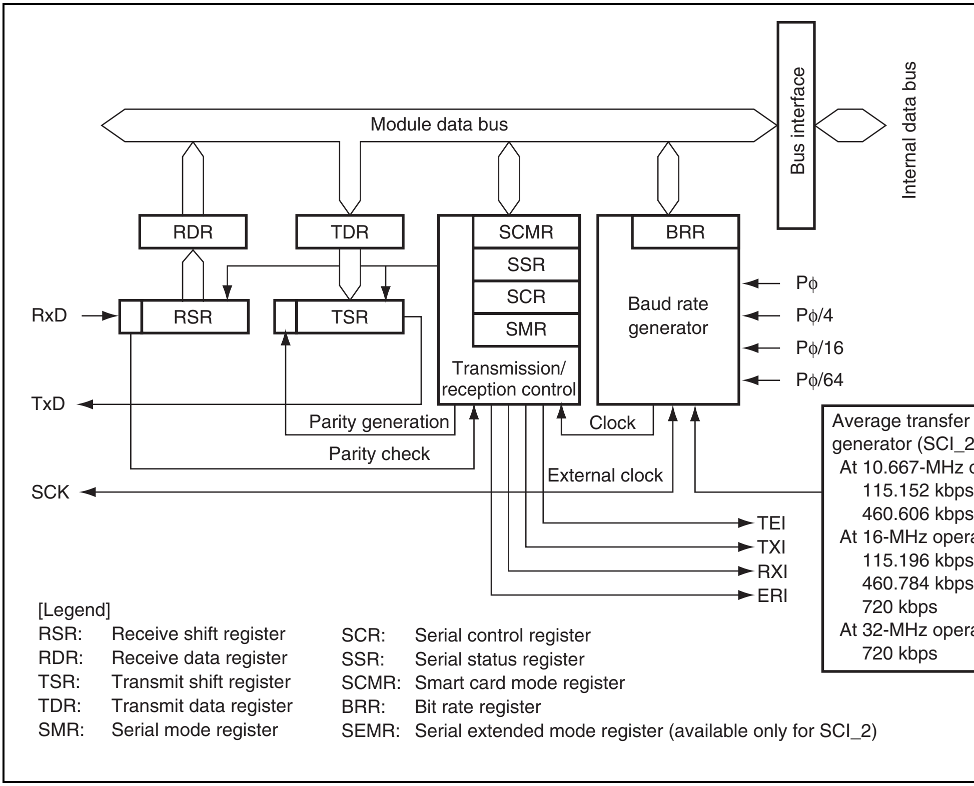


Figure 14.1 Block Diagram of SCI

	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
3	SCK3	I/O	Channel 3 clock input/output
	RxD3	Input	Channel 3 receive data input
	TxD3	Output	Channel 3 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 3:

- Receive shift register_3 (RSR_3)
- Transmit shift register_3 (TSR_3)
- Receive data register_3 (RDR_3)
- Transmit data register_3 (TDR_3)
- Serial mode register_3 (SMR_3)
- Serial control register_3 (SCR_3)
- Serial status register_3 (SSR_3)
- Smart card mode register_3 (SCMR_3)
- Bit rate register_3 (BRR_3)

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)

receive the next data. Since RSR and RDR function as a double buffer in this way, contin
receive operations can be performed. After confirming that the RDRF bit in SSR is set to
RDR only once. RDR cannot be written to by the CPU.

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

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-bu
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

- When SMIF in SCMR = 0

Bit	7	6	5	4	3	2	1
Bit Name	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	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

- 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 0: Selects 8 bits as the data length. 1: Selects 7 bits as the data length. LSB-first and the MSB (bit 7) in TDR is not transmitt transmission. In clocked synchronous mode, a fixed data le bits is used.

				1: Selects odd parity.
3	STOP	0	R/W	Stop Bit Length (valid only in asynchronous mo Selects the stop bit length in transmission. 0: 1 stop bit 1: 2 stop bits In reception, only the first stop bit is checked. I second stop bit is 0, it is treated as the start bit next transmit frame.
2	MP	0	R/W	Multiprocessor Mode (valid only in asynchrono When this bit is set to 1, the multiprocessor fur enabled. The PE bit and O/$\overline{E}$ bit settings are in 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 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 b the baud rate, see section 14.3.9, Bit Rate Reg (BRR). n is the decimal display of the value of (see section 14.3.9, Bit Rate Register (BRR)).

6	BLK	0	R/W	Setting this bit to 1 allows block transfer mode operation. For details, see section 14.7.3, Blo Transfer Mode.
5	PE	0	R/W	Parity Enable (valid only in asynchronous mo When this bit is set to 1, the parity bit is added transmit data before transmission, and the pa checked in reception. Set this bit to 1 in smart interface mode.
4	O/$\overline{\text{E}}$	0	R/W	Parity Mode (valid only when the PE bit is 1 in asynchronous mode) 0: Selects even parity 1: Selects odd parity For details on the usage of this bit in smart ca interface mode, see section 14.7.2, Data Forn (Except in Block Transfer Mode).
3	BCP1	0	R/W	Basic Clock Pulse 1,0
2	BCP0	0	R/W	These bits select the number of basic clock cy 1-bit data transfer time in smart card interface 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 Timing and Reception Margin. S is described 14.3.9, Bit Rate Register (BRR).

the baud rate, see section 14.3.9, Bit Rate Reg
(BRR). n is the decimal display of the value of
(see section 14.3.9, Bit Rate Register (BRR)).

Note: etu (Elementary Time Unit): 1-bit transfer time

14.3.6 Serial Control Register (SCR)

SCR is a register that enables/disables the following SCI transfer operations and interrupt
and selects the transfer clock source. For details on interrupt requests, see section 14.8, In
Sources. Some bits in SCR have different functions in normal mode and smart card interf
mode.

- When SMIF in SCMR = 0

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

- 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

				When this bit is set to 1, RXI and ERI interrup are enabled. RXI and ERI interrupt requests can be cancel reading 1 from the RDRF, FER, PER, or ORE then clearing the flag to 0, or by clearing the F 0.
5	TE	0	R/W	Transmit Enable When this bit is set to 1, transmission is enab this condition, serial transmission is started by transmit data to TDR, and clearing the TDRE SSR to 0. Note that SMR should be set prior t the TE bit to 1 in order to designate the transn format. If transmission is halted by clearing this bit to TDRE flag in SSR is fixed 1.
4	RE	0	R/W	Receive Enable When this bit is set to 1, reception is enabled. this condition, serial reception is started by de the start bit in asynchronous mode or the sync clock input in clocked synchronous mode. No SMR should be set prior to setting the RE bit order to designate the reception format. Even if reception is halted by clearing this bit RDRF, FER, PER, and ORER flags are not af and the previous value is retained.

				When receive data including MPB = 0 in SSR received, transfer of the received data from RS RDR, detection of reception errors, and the set RDRF, FER, and ORER flags in SSR are not performed. When receive data including MPB = received, the MPB bit in SSR is set to 1, the M automatically cleared to 0, and RXI and ERI int requests (in the case where the TIE and RIE b SCR are set to 1) and setting of the FER and O flags are enabled.
2	TEIE	0	R/W	Transmit End Interrupt Enable When this bit is set to 1, a TEI interrupt reques enabled. A TEI interrupt request can be cancel reading 1 from the TDRE flag and then clearing to 0 in order to clear the TEND flag to 0, or by the TEIE bit to 0.

rate from the SCK pin.)

1X: External clock

(Inputs a clock with a frequency 16 times 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

Bit Functions in Smart Card Interface Mode (When SMIF in SCMR = 1):

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 reques enabled. A TXI interrupt request can be cancelled by re from the TDRE flag and then clearing the flag clearing the TIE bit to 0.

				this condition, serial transmission is started by transmit data to TDR, and clearing the TDRE fl SSR to 0. Note that SMR should be set prior to the TE bit to 1 in order to designate the transm format. 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 this condition, serial reception is started by det the start bit in asynchronous mode or the sync clock input in clocked synchronous mode. Note SMR should be set prior to setting the RE bit to order to designate the reception format. Even if reception is halted by clearing this bit to RDRF, FER, PER, and ORER flags are not aff and 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 fun
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 allo DMAC or DTC to write data to TDR
6	RDRF	0	R/(W)*	Receive Data Register Full Indicates whether receive data is stored in RD [Setting condition] • When serial reception ends normally and re data 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 sure to read the flag after writing 0 to it.) • When an RXI interrupt request is issued all DMAC or DTC to read data from RDR The RDRF flag is not affected and retains its p 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.

				be performed. Note that, in clocked synch mode, serial transmission also cannot con [Clearing condition] • When 0 is written to ORER after reading O Even when the RE bit in SCR is cleared, t flag is not affected and retains its previous (When the CPU is used to clear this flag b while the corresponding interrupt is enable sure to read the flag after writing 0 to it.)
4	FER	0	R/(W)*	Framing Error Indicates that a framing error has occurred du reception in asynchronous mode and the rece ends abnormally. [Setting condition] • When the stop bit is 0 In 2-stop-bit mode, only the first stop bit is whether it is 1 but the second stop bit is no checked. Note that receive data when the error occurs is transferred to RDR, howev RDRF flag is not set. In addition, when the is being set to 1, the subsequent serial rec cannot be performed. In clocked synchron mode, serial transmission also cannot con [Clearing condition] • When 0 is written to FER after reading FE Even when the RE bit in SCR is cleared, t flag is not affected and retains its previous (When the CPU is used to clear this flag b while the corresponding interrupt is enable sure to read the flag after writing 0 to it.)

				the subsequent serial reception cannot be performed. In clocked synchronous mode, s transmission also cannot continue. [Clearing condition] • When 0 is written to PER after reading PEF Even when the RE bit in SCR is cleared, th bit is not affected and retains its previous v (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	TEND	1	R	Transmit End [Setting conditions] • When the TE bit in SCR is 0 • When TDRE = 1 at transmission of the last transmit character [Clearing conditions] • When 0 is written to TDRE after reading TD • When a TXI interrupt request is issued allov DMAC or DTC to write data to TDR
1	MPB	0	R	Multiprocessor Bit Stores the multiprocessor bit value in the recei When the RE bit in SCR is cleared to 0 its prev state is retained.
0	MPBT	0	R/W	Multiprocessor Bit Transfer Sets the multiprocessor bit value to be added t transmit frame.

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

				• When 0 is written to TDRE after reading T (When the CPU is used to clear this flag b while the corresponding interrupt is enable sure to read the flag after writing 0 to it.) • When a TXI interrupt request is issued all DMAC or DTC to write data to TDR
6	RDRF	0	R/(W)*	Receive Data Register Full Indicates whether receive data is stored in RD [Setting condition] • When serial reception ends normally and data is transferred from RSR to RDR [Clearing conditions] • When 0 is written to RDRF after reading R (When the CPU is used to clear this flag b while the corresponding interrupt is enable sure to read the flag after writing 0 to it.) • When an RXI interrupt request is issued a DMAC or DTC to read data from RDR The RDRF flag is not affected and retains its p value even when the RE bit in SCR is cleared Note that when the next reception is complete the RDRF flag is being set to 1, an overrun er and the received data is lost.

				ORER flag is set to 1, subsequent serial re cannot be performed. Note that, in clocked synchronous mode, serial transmission als continue. [Clearing condition] • When 0 is written to ORER after reading O Even when the RE bit in SCR is cleared, th flag is not affected and retains its previous (When the CPU is used to clear this flag by while the corresponding interrupt is enable sure to read the flag after writing 0 to it.)
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 ER

the subsequent serial reception cannot be
performed. In clocked synchronous mode
transmission also cannot continue.

[Clearing condition]

- When 0 is written to PER after reading PE

  Even when the RE bit in SCR is cleared, t
  flag is not affected and retains its previous

  (When the CPU is used to clear this flag b
  while the corresponding interrupt is enable
  sure to read the flag after writing 0 to it.)

				follows: When GM = 0 and BLK = 0, 2.5 etu after transmission start When GM = 0 and BLK = 1, 1.5 etu after transmission start When GM = 1 and BLK = 0, 1.0 etu after transmission start When GM = 1 and BLK = 1, 1.0 etu after transmission start [Clearing conditions] • When 0 is written to TDRE after reading TD • When a TXI interrupt request is issued allo DMAC or DTC to write the next data to TD
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.

7 to 4	—	All 1	R	Reserved These are read-only bits and cannot be modif
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 forma for transmission/reception; when the 7-bit dat used, data is always transmitted/received wit
2	SINV	0	R/W	Smart Card Data Invert Inverts the transmit/receive data logic level. T does not affect the logic level of the parity bit. the parity bit, invert the O/$\overline{E}$ bit in SMR. 0: TDR contents are transmitted as they are. data is stored as it is in RDR. 1: TDR contents are inverted before being tra Receive data is stored in inverted form in F
1	—	1	R	Reserved This is a read-only bit and cannot be modifiec
0	SMIF	0	R/W	Smart Card Interface Mode Select When this bit is set to 1, smart card interface selected. 0: Normal asynchronous or clocked synchron 1: Smart card interface mode

Asynchronous mode	$N = \frac{P\phi \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	$\text{Error (\%)} = \{\frac{P\phi \times 10^6}{B \times 64 \times 2^{2n-1} \times (N+1)} -$
Clocked synchronous mode	$N = \frac{P\phi \times 10^6}{8 \times 2^{2n-1} \times B} - 1$	
Smart card interface mode	$N = \frac{P\phi \times 10^6}{S \times 2^{2n+1} \times B} - 1$	$\text{Error (\%)} = \{\frac{P\phi \times 10^6}{B \times S \times 2^{2n+1} \times (N+1)} -$

[Legend]

B: Bit rate (bit/s)

N: BRR setting for baud rate generator (0 ≤ N ≤ 255)

Pϕ: Operating frequency (MHz)

n and S: Determined by the SMR settings shown in the following table.

SMR Setting		
CKS1	**CKS0**	**n**
0	0	0
0	1	1
1	0	2
1	1	3

SMR Setting		
BCP1	**BCP0**	**S**
0	0	32
0	1	64
1	0	37
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

16	500000	0	0

Table 14.5 Maximum Bit Rate with External Clock Input (Asynchronous Mode)

Pϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bit/s)
8	2.0000	125000
9.8304	2.4576	153600
10	2.5000	156250
12	3.0000	187500
12.288	3.0720	192000
14	3.5000	218750
14.7456	3.6864	230400
16	4.0000	250000

5k	1	99	1	124	1	199	1
10k	0	199	0	249	1	99	1
25k	0	79	0	99	0	159	0
50k	0	39	0	49	0	79	0
100k	0	19	0	24	0	39	0
250k	0	7	0	9	0	15	0
500k	0	3	0	4	0	7	0
1M	0	1			0	3	0
2.5M			0	0*			0
5M							0

10k	1	155	1	187	1	205	1
25k	0	249	1	74	1	82	1
50k	0	124	0	149	0	164	0
100k	0	62	0	74	0	82	0
250k	0	24	0	29	0	32	0
500k	—	—	0	14	—	—	—
1M	—	—	—	—	—	—	—
2.5M	—	—	0	2	—	—	—
5M	—	—	—	—	—	—	—

[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	3333333
10	1.6667	1666666.7	25	4.1667	4166666
12	2.0000	2000000.0	30	5.0000	5000000
14	2.3333	2333333.3	33	5.5000	5500000
16	2.6667	2666666.7	35	5.8336	5833625
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

	Operating Frequency Pϕ (MHz)											
Bit Rate	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 Interface 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	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 (valic 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 tr
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

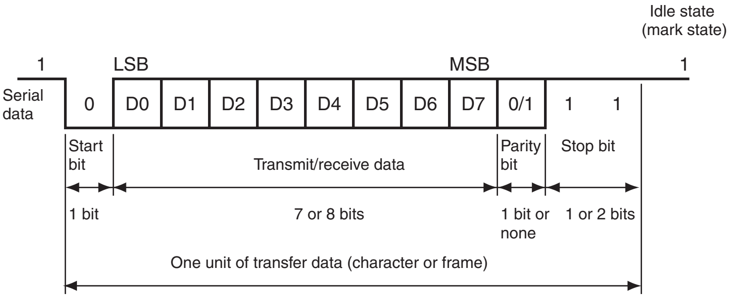


Figure 14.2 Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, Two Stop Bits)

0	0	0	0	S 8-bit data STOP
0	0	0	1	S 8-bit data STOP STOP
0	1	0	0	S 8-bit data P STOP
0	1	0	1	S 8-bit data P STOP STO
1	0	0	0	S 7-bit data STOP
1	0	0	1	S 7-bit data STOP STOP
1	1	0	0	S 7-bit data P STOP
1	1	0	1	S 7-bit data P STOP STOP
0	—	1	0	S 8-bit data MPB STOP
0	—	1	1	S 8-bit data MPB STOP STO
1	—	1	0	S 7-bit data MPB STOP
1	—	1	1	S 7-bit data MPB STOP STOP

[Legend]

S:	Start bit
STOP:	Stop bit
P:	Parity bit
MPB:	Multiprocessor bit

M: Reception margin
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.

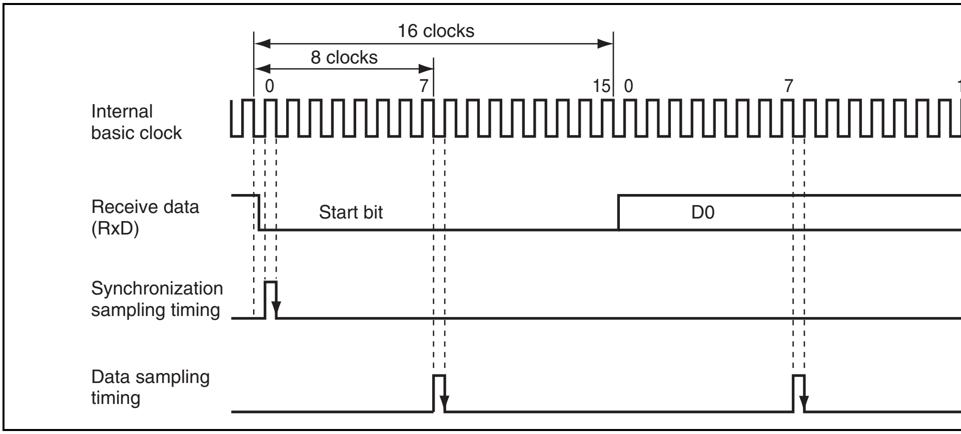


Figure 14.3 Receive Data Sampling Timing in Asynchronous Mode

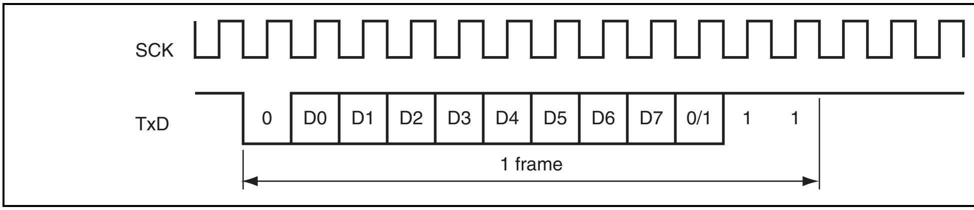
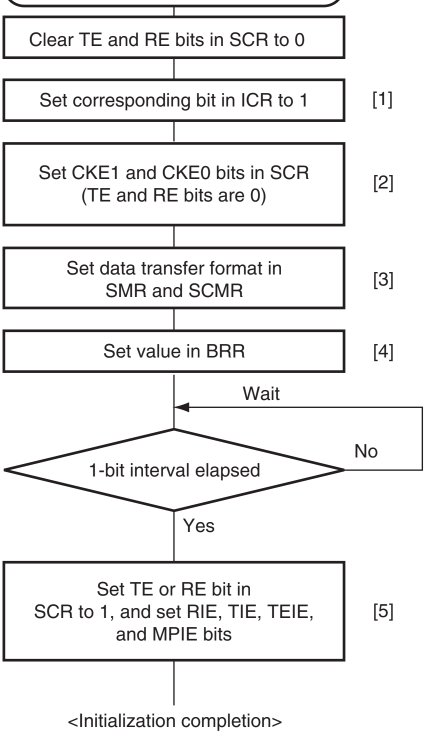


Figure 14.4 Phase Relation between Output Clock and Transmit Data (Asynchronous Mode)



Clear TE and RE bits in SCR to 0
Set corresponding bit in ICR to 1
[1]
Set CKE1 and CKE0 bits in SCR (TE and RE bits are 0)
[2]
Set data transfer format in SMR and SCMR
[3]
Set value in BRR
[4]
Wait
1-bit interval elapsed
No
Yes
Set TE or RE bit in SCR to 1, and set RIE, TIE, TEIE, and MPIE bits
[5]
<Initialization completion>

[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.

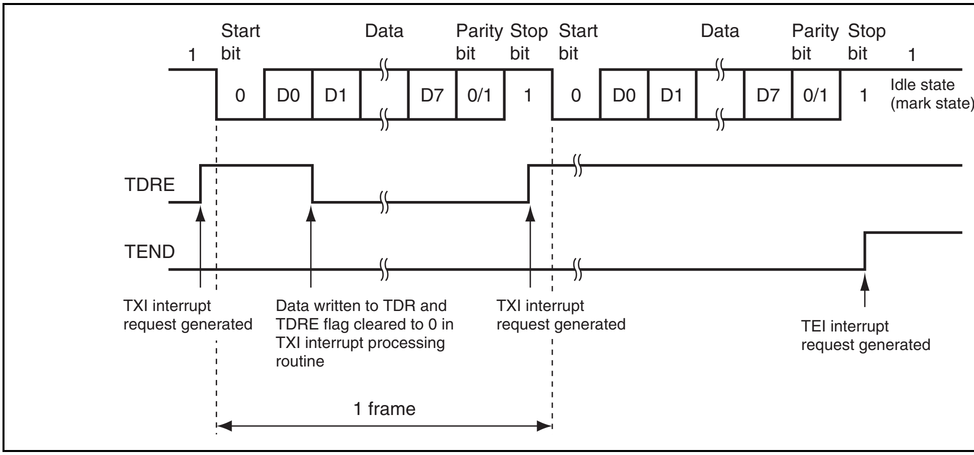
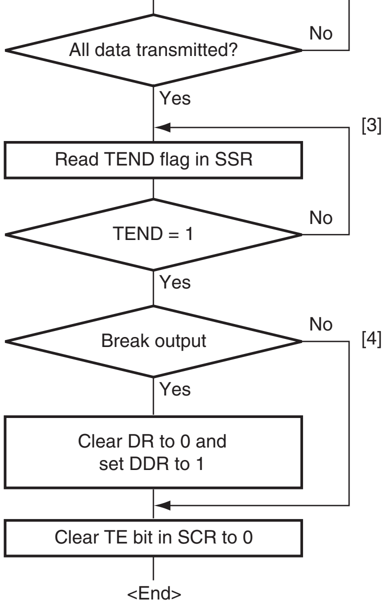

1
Start bit
Data
Parity bit
Stop bit
Start bit
Data
Parity bit
Stop bit
1
0
D0
D1
D7
0/1
1
0
D0
D1
D7
0/1
1
Idle state (mark state)
TDRE
TEND
TXI interrupt request generated
Data written to TDR and TDRE flag cleared to 0 in TXI interrupt processing routine
TXI interrupt request generated
TEI interrupt request generated
1 frame

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 DTC or DMAC 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.

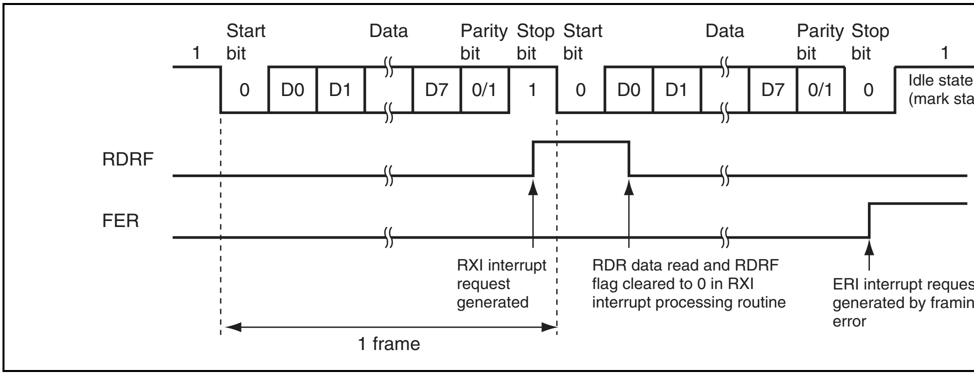
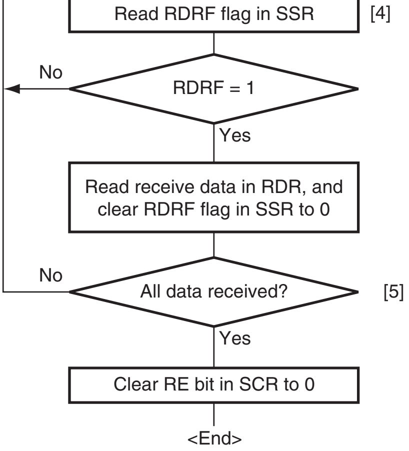

1
Start bit
0
D0
D1
Data
D7
Parity bit
0/1
Stop bit
1
Start bit
0
D0
D1
Data
D7
Parity bit
0/1
Stop bit
0
1
Idle state
(mark sta
RDRF
FER
RXI interrupt request generated
RDR data read and RDRF flag cleared to 0 in RXI interrupt processing routine
ERI interrupt reques generated by framin error
1 frame

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 DTC or DMAC is initiated by an RXI interrupt and reads data from RDR.

Figure 14.9 Sample Serial Reception Flowchart (1)

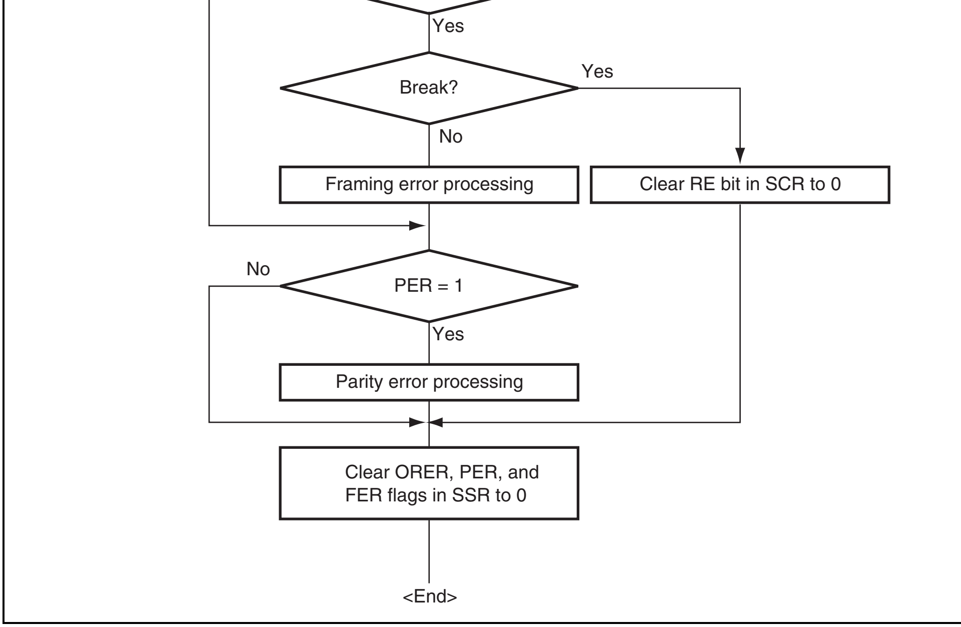


Figure 14.9 Sample Serial Reception Flowchart (2)

14.10 shows an example of inter-processor communication using the multiprocessor form 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 II 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.

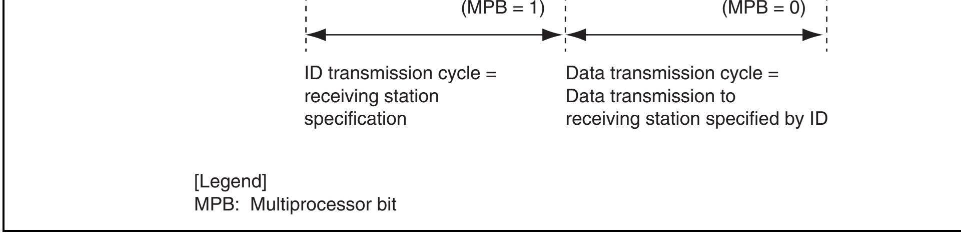
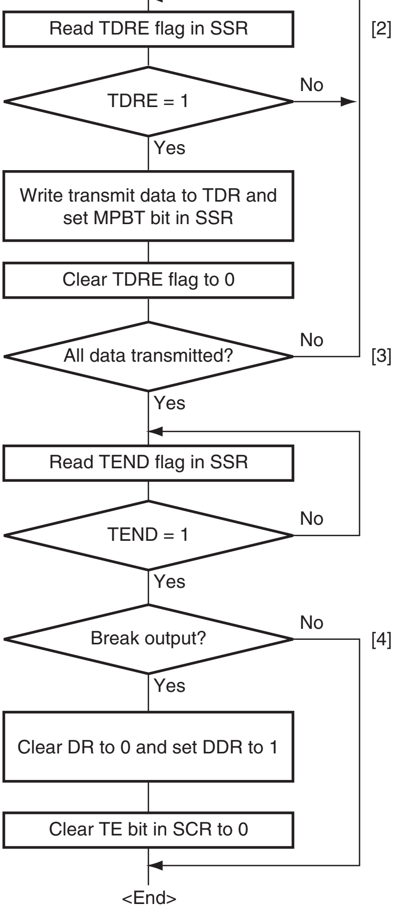


Figure 14.10 Example of Communication Using Multiprocessor Format
(Transmission of Data H'AA to Receiving Station A)



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 DTC or DMAC 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

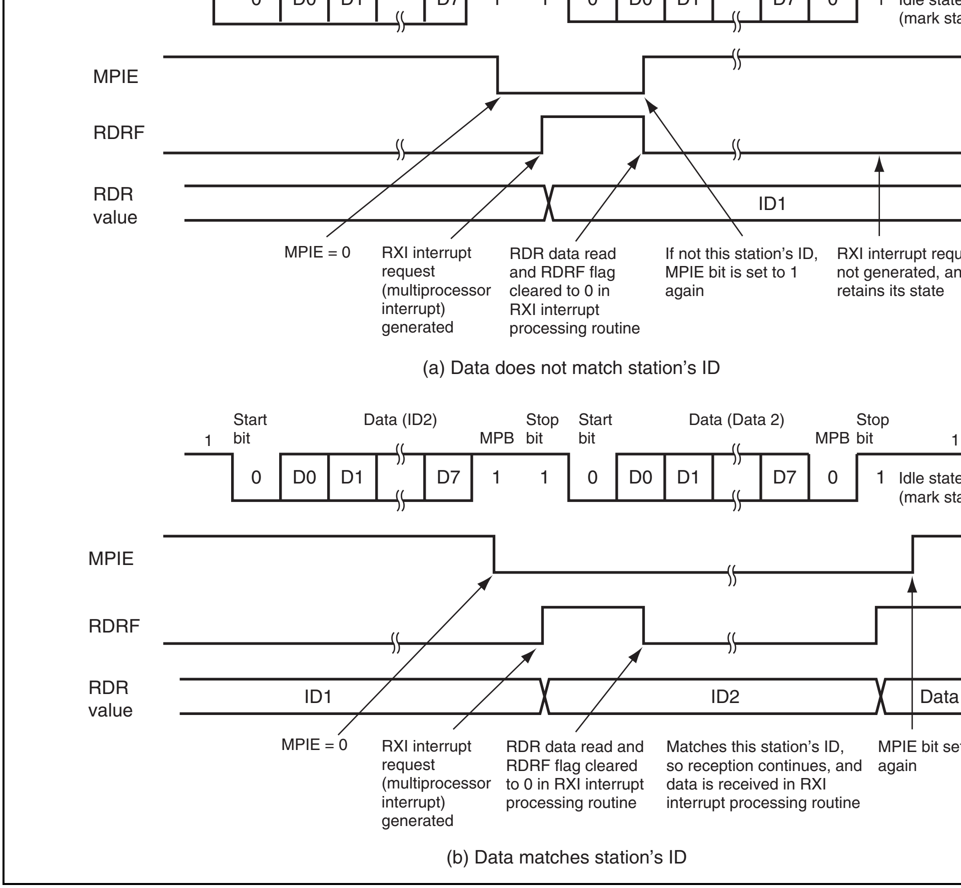
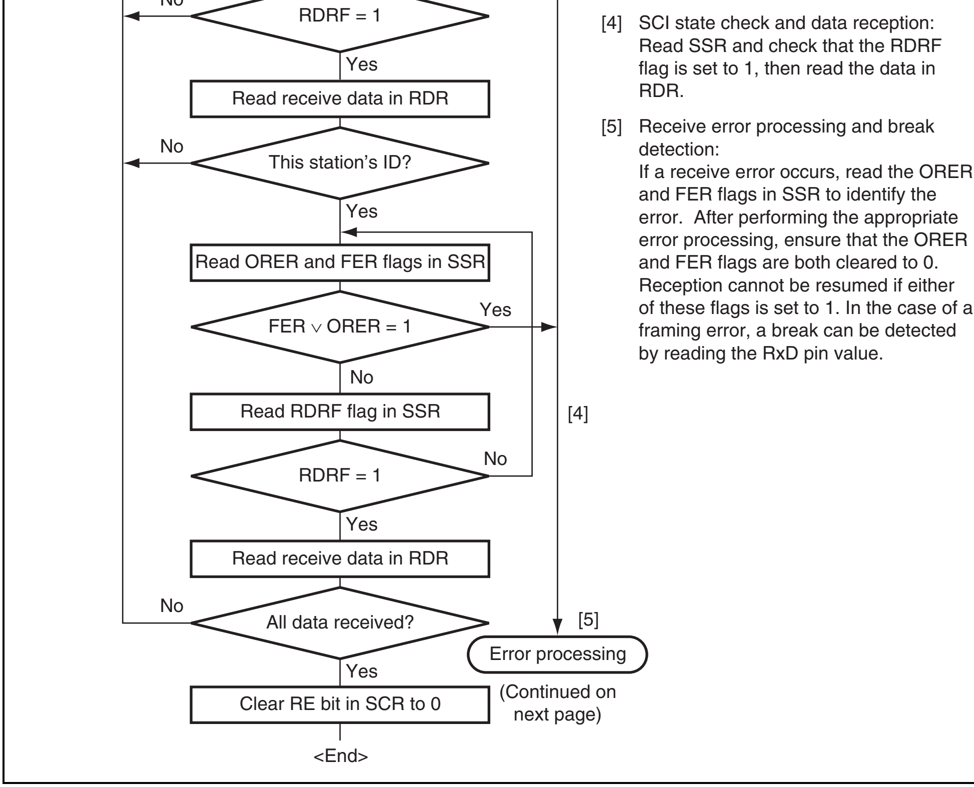


Figure 14.12 Example of SCI Operation for Reception (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)



(Continued on next page)

[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)

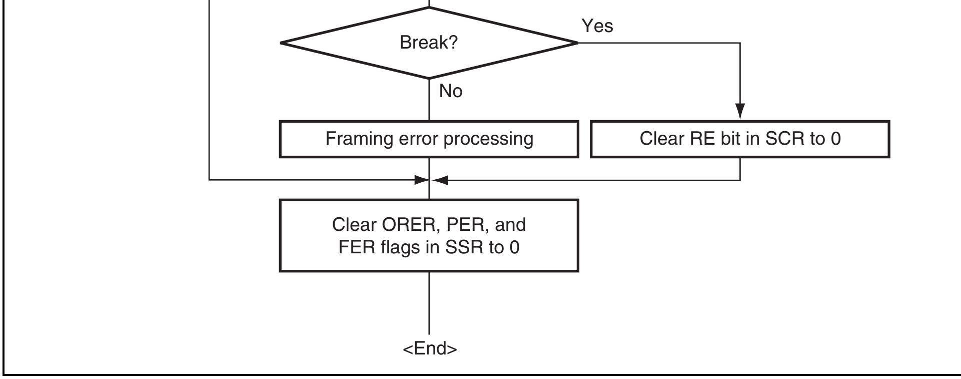
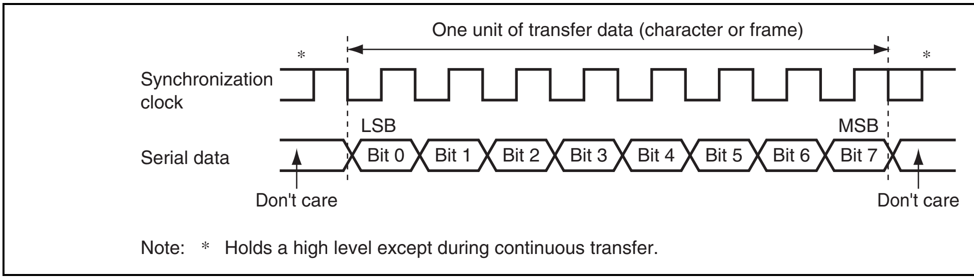


Figure 14.13 Sample Multiprocessor Serial Reception Flowchart (2)

receiver also have a double-buffered structure, so that the next transmit data can be writte transmission or the previous receive data can be read during reception, enabling continuo transfer.

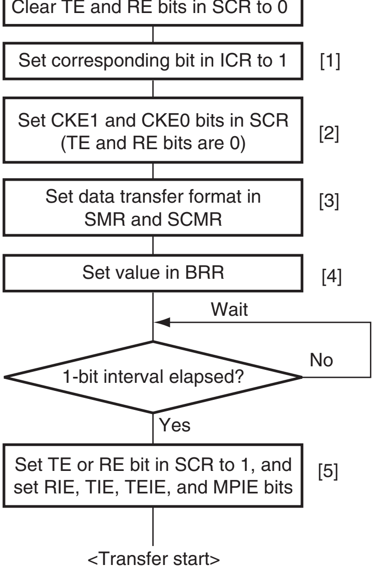


Note: * Holds a high level except during continuous transfer.

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.


Clear TE and RE bits in SCR to 0
Set corresponding bit in ICR to 1
[1]
Set CKE1 and CKE0 bits in SCR (TE and RE bits are 0)
[2]
Set data transfer format in SMR and SCMR
[3]
Set value in BRR
[4]
Wait
No
1-bit interval elapsed?
Yes
Set TE or RE bit in SCR to 1, and set RIE, TIE, TEIE, and MPIE bits
[5]
<Transfer start>

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.

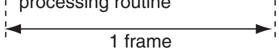
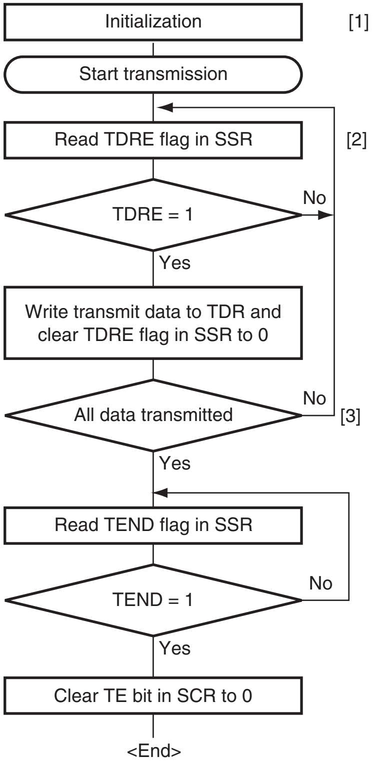


Figure 14.16 Example of Operation for Transmission in Clocked Synchronous



[1] SCI initialization:
The TxD pin is automatically designated as the transmit data output pin.

[2] SCI state check and transmit data write:
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR and clear the TDRE flag to 0.

[3] Serial transmission continuation procedure:
To continue serial transmission, be sure to read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and then clear the TDRE flag to 0. However, the TDRE flag is checked and cleared automatically when the DTC or DMAC 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.

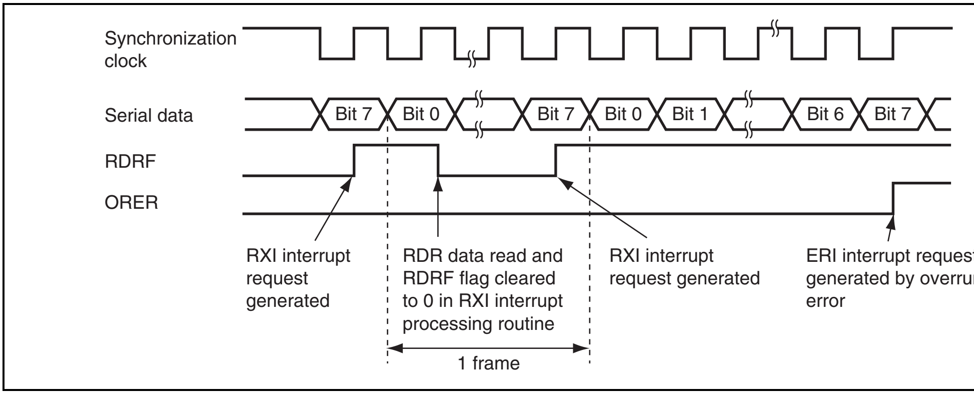
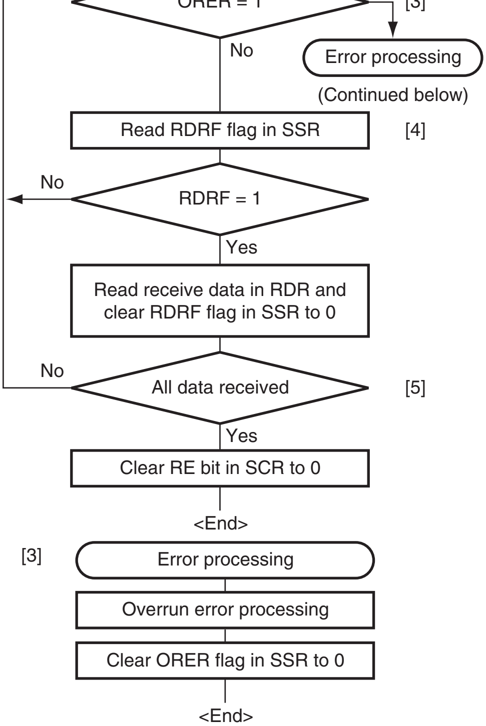

Synchronization clock
Serial data
Bit 7
Bit 0
Bit 7
Bit 0
Bit 1
Bit 6
Bit 7
RDRF
ORER
RXI interrupt request generated
RDR data read and RDRF flag cleared to 0 in RXI interrupt processing routine
RXI interrupt request generated
ERI interrupt reques generated by overru error
1 frame

Figure 14.18 Example of Operation for Reception in Clocked Synchronous M



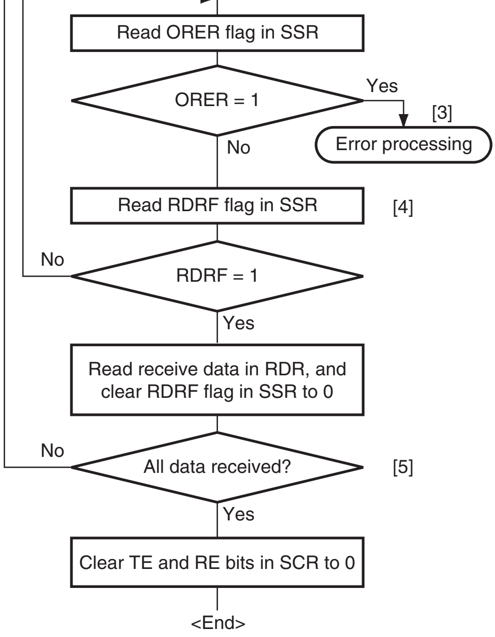
Receive cannot be resumed if the ORER flag is set to 1.

[4] SCI state check and receive data read:
Read SSR and check that the RDRF flag is set to 1, then read the receive data in RDR and clear the RDRF flag to 0. Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial reception continuation procedure:
To continue serial reception, before the MSB (bit 7) of the current frame is received, 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 DTC or DMAC is initiated by a receive data full interrupt (RXI) and reads data from RDR.

Figure 14.19 Sample Serial Reception Flowchart

the RDRF bit and receive error flags (ORER, FER, and PER) are cleared to 0, simultaneo
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, re
from the TDRE flag to confirm that writing i
possible. Then write data to TDR and clear
TDRE flag to 0.
However, the TDRE flag is checked and cle
automatically when the DTC or DMAC is in
by a transmit data empty interrupt (TXI) req
writes data to TDR. Similarly, the RDRF fla

Note: When switching from transmit or receive op
to simultaneous transmit and receive opera
first clear the TE bit and RE bit to 0, then se
these bits to 1 simultaneously.

Figure 14.20 Sample Flowchart of Simultaneous Serial Transmission and Rec

TxD and RxD pins and pull up the data transmission line to V_{CC} using a resistor. Setting t and TE bits to 1 with the IC card not connected enables closed transmission/reception all self diagnosis. To supply the IC card with the clock pulses generated by the SCI, input th pin output to the CLK pin of the IC card. A reset signal can be supplied via the output po LSI.

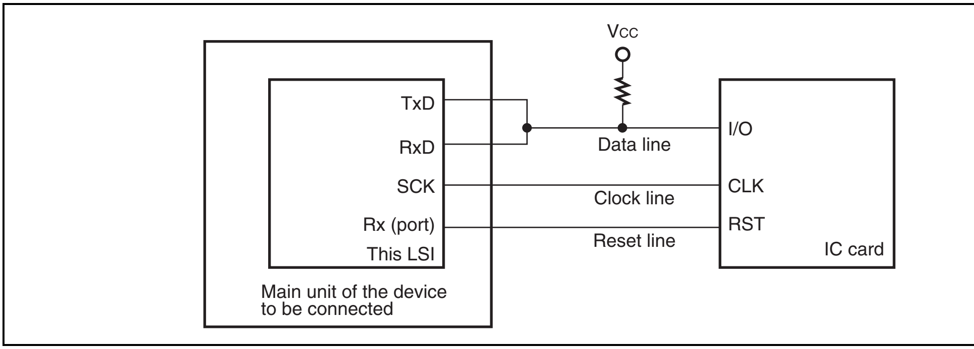


Figure 14.21 Pin Connection for Smart Card Interface

after at least 2 etu.

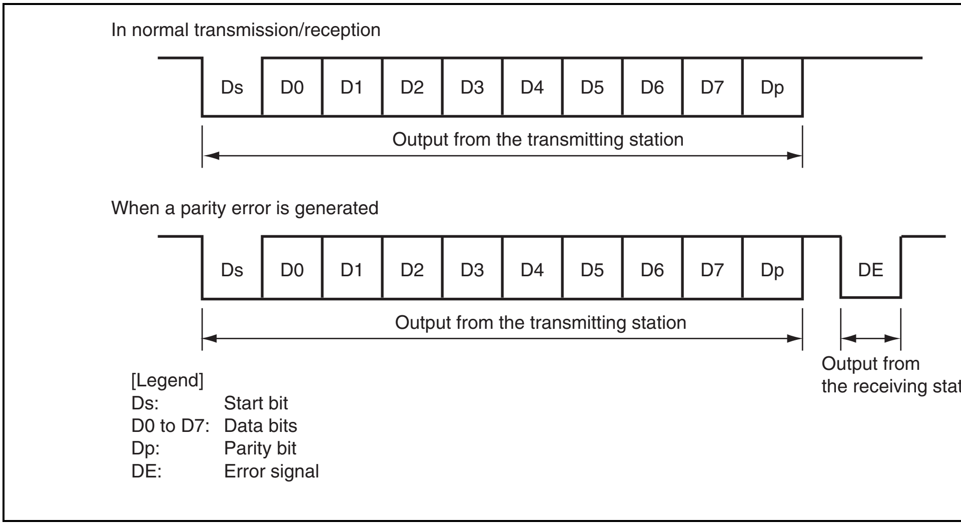


Figure 14.22 Data Formats in Normal Smart Card Interface Mode

For communication with the IC cards of the direct convention and inverse convention ty follow the procedure below.

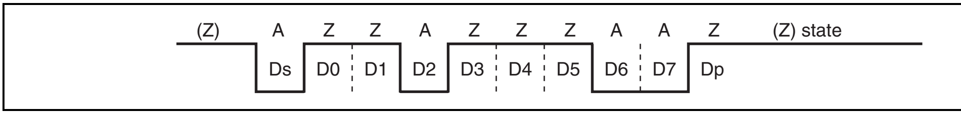


Figure 14.23 Direct Convention (SDIR = SINV = O/$\overline{\text{E}}$ = 0)

For the inverse convention type, logic levels 1 and 0 correspond to states A and Z, respec
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, w
both the SDIR and SINV bits in SCMR. The parity bit is logic level 0 to produce even pa
which is prescribed by the smart card standard, and corresponds to state Z. Since the SNI
this LSI only inverts data bits D7 to D0, write 1 to the O/$\overline{\text{E}}$ bit in SMR to invert the parity
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 the
  frame.
- During transmission, at least 1 etu is secured as a guard time after the end of the parity
  before the start of the next frame.
- Since the same data is not re-transmitted during transmission, the TEND flag is set 11
  after transmission start.
- Although the ERS flag in block transfer mode displays the error signal status as in nor
  smart card interface mode, the flag is always read as 0 because no error signal is trans

$$M = \left| \left(0.5 - \frac{1}{2N}\right) - (L - 0.5)\,F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\%$$

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 mar
determined by the formula below.

$$M = \left(0.5 - \frac{1}{2 \times 372}\right) \times 100\% = 49.866\%$$

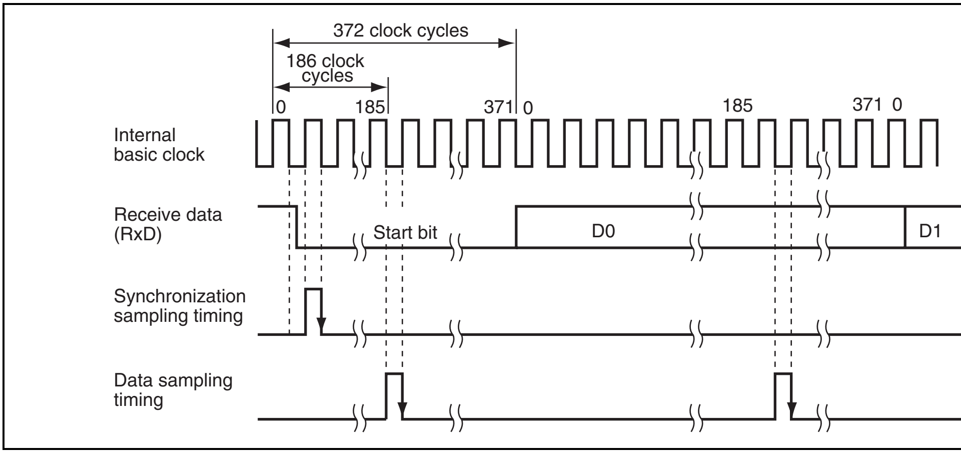


Figure 14.25 Receive Data Sampling Timing in Smart Card Interface Mo
(When Clock Frequency is 372 Times the Bit Rate)

5. Set the SMIF, SDIR, and SINV bits in SCMR appropriately. When the DDR correspo the TxD pin is cleared to 0, the TxD and RxD pins are changed from port pins to SCI 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, MPI 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-b interval. Setting the TE and RE bits to 1 simultaneously is prohibited except for self d

To switch from reception to transmission, first verify that reception has completed, then i the SCI. At the end of initialization, RE and TE should be set to 0 and 1, respectively. Re 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 S 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.

re-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
4. In this case, one frame of data is determined to have been transmitted including re-tr the TEND bit in SSR is set to 1. Here, a TXI interrupt request is generated if the TIE 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 DTC or DMAC. In transmission, the TEND and TDRE flags in SSR are simultaneously set to 1, thus genera TXI interrupt request if the TIE bit in SCR has been set to 1. This activates the DTC or I a TXI request thus allowing transfer of transmit data if the TXI interrupt request is speci source of DTC or DMAC activation beforehand. The TDRE and TEND flags are autom cleared to 0 at data transfer by the DTC or DMAC. If an error occurs, the SCI automatic transmits the same data. During re-transmission, TEND remains as 0, thus not activating or DMAC. Therefore, the SCI and DTC or DMAC automatically transmit the specified bytes, including re-transmission in the case of error occurrence. However, the ERS flag automatically cleared; the ERS flag must be cleared by previously setting the RIE bit to enable an ERI interrupt request to be generated at error occurrence.

When transmitting/receiving data using the DTC or DMAC, be sure to set and enable th DMAC prior to making SCI settings. For DTC or DMAC settings, see section 7, DMA (DMAC) and 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 SM
Figure 14.27 shows the TEND flag set timing.

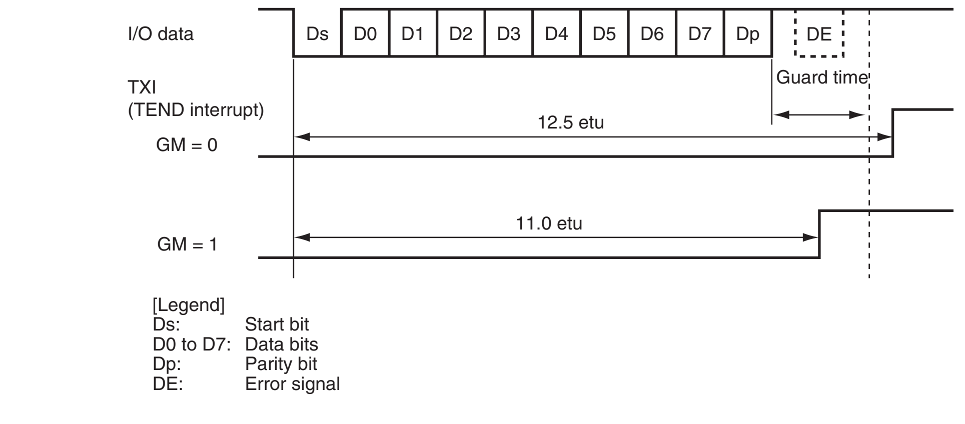


Figure 14.27 TEND Flag Set Timing during Transmission

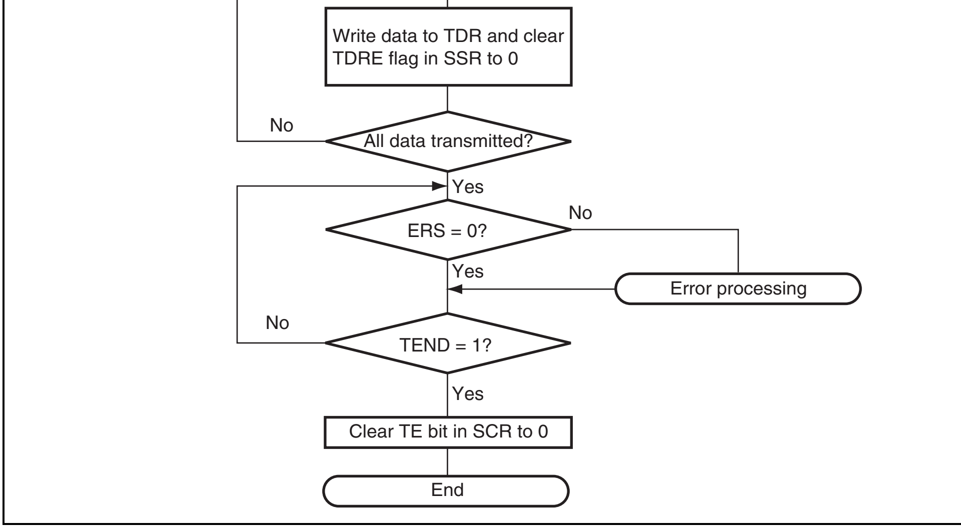


Figure 14.28 Sample Transmission Flowchart

4. In this case, data is determined to have been received successfully, and the RDRF bit 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 automa performed using an RXI interrupt request to activate the DTC or DMAC. In reception, se RIE bit to 1 allows an RXI interrupt request to be generated when the RDRF flag is set to activates the DTC or DMAC by an RXI request thus allowing transfer of receive data if t interrupt request is specified as a source of DTC or DMAC activation beforehand. The R is automatically cleared to 0 at data transfer by the DTC or DMAC. If an error occurs dur 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 DTC or DM not activated and receive data is skipped, therefore, the number of bytes of receive data sp in the DTC or DMAC is transferred. Even if a parity error occurs and the PER bit is set to 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 Asynchrono

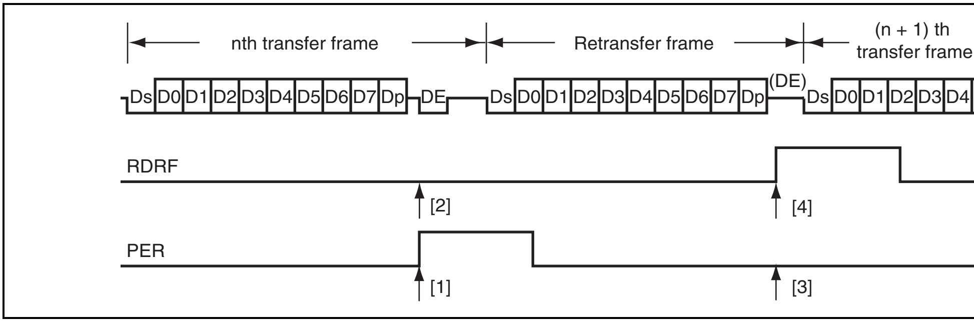


Figure 14.29 Data Re-Transfer Operation in SCI Reception Mode

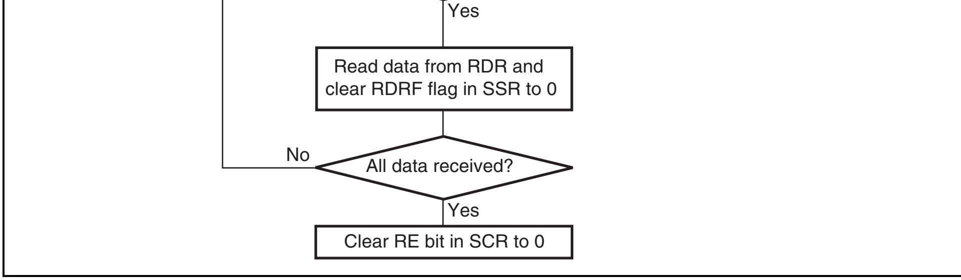


Figure 14.30 Sample Reception Flowchart

14.7.8 Clock Output Control

Clock output can be fixed using the CKE1 and CKE0 bits in SCR when the GM bit in S
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 cont
with GM = 1 and CKE1 = 0.

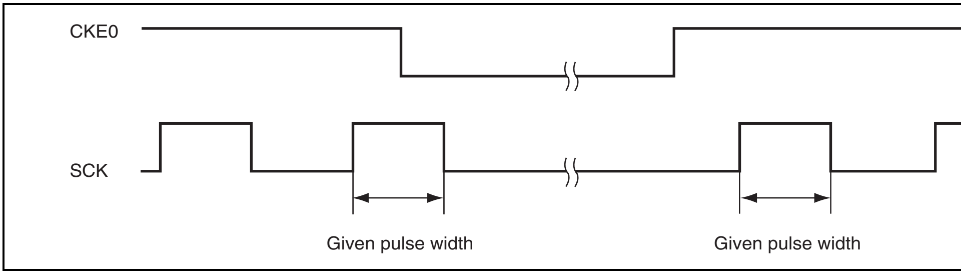


Figure 14.31 Clock Output Fixing Timing

Set the CKE0 bit in SCR to 1 to start clock output.

- At mode switching

— At transition from smart card interface mode to software standby mode

1. Set the data register (DR) and data direction register (DDR) corresponding to t 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. Simultan set the CKE1 bit to the value for the output fixed state in software standby mo
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 fix 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.

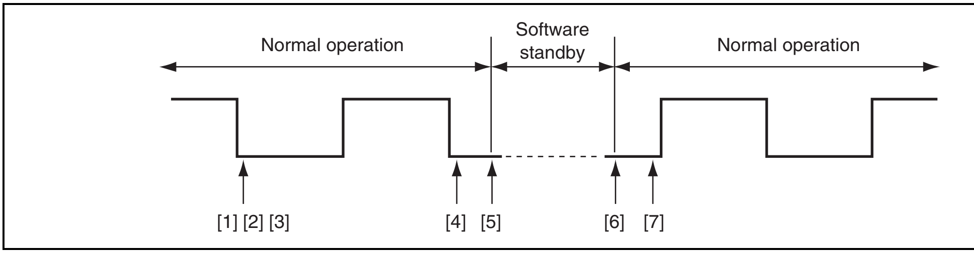


Figure 14.32 Clock Stop and Restart Procedure

DTC or DMAC to allow data transfer. The TDRE flag is automatically cleared to 0 at da
by the DTC or DMAC.

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 inter
activate the DTC or DMAC to allow data transfer. The RDRF flag is automatically clea
data transfer by the DTC or DMAC.

A TEI interrupt is requested when the TEND flag is set to 1 while the TEIE bit is set to
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 simultane
the TXI interrupt processing routine, the SCI cannot branch to the TEI interrupt processi
later.

Table 14.12 SCI Interrupt Sources

Name	Interrupt Source	Interrupt Flag	DMAC Activation	DTC 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	L

Data transmission/reception using the DTC or DMAC is also possible in smart card interf
mode, similar to in the normal SCI mode. In transmission, the TEND and TDRE flags in
simultaneously set to 1, thus generating a TXI interrupt. This activates the DTC or DMA
TXI request thus allowing transfer of transmit data if the TXI request is specified as a sou
DTC or DMAC activation beforehand. The TDRE and TEND flags are automatically cle
at data transfer by the DTC or DMAC. If an error occurs, the SCI automatically re-transm
same data. During re-transmission, the TEND flag remains as 0, thus not activating the D
DMAC. Therefore, the SCI and DTC or DMAC automatically transmit the specified num
bytes, including re-transmission in the case of error occurrence. However, the ERS flag i
which is set at error occurrence, is not automatically cleared; the ERS flag must be cleare
previously setting the RIE bit in SCR to 1 to enable an ERI interrupt request to be genera
error occurrence.

When transmitting/receiving data using the DTC or DMAC, be sure to set and enable the
DMAC prior to making SCI settings. For DTC or DMAC settings, see section 7, DMA C
(DMAC) and section 8, Data Transfer Controller (DTC).

In reception, an RXI interrupt request is generated when the RDRF flag in SSR is set to 1
activates the DTC or DMAC by an RXI request thus allowing transfer of receive data if t
request is specified as a source of DTC or DMAC activation beforehand. The RDRF flag
automatically cleared to 0 at data transfer by the DTC or DMAC. If an error occurs, the R
flag is not set but the error flag is set. Therefore, the DTC or DMAC 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
directly. In a break, the input from the RxD pin becomes all 0s, and so the FER flag is s
PER flag may also be set. Note that, since the SCI continues the receive operation even
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

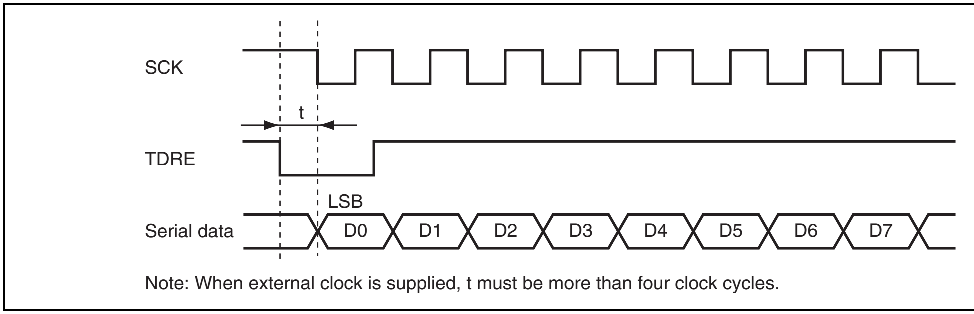
When the TE bit is 0, the TxD pin is used as an I/O port whose direction (input or outpu
level are determined by DR and DDR. This can be used to set the TxD pin to mark state
level) or send a break during serial data transmission. To maintain the communication li
state (the state of 1) until TE is set to 1, set both DDR and DR to 1. Since the TE bit is c
at this point, the TxD pin becomes an I/O port, and 1 is output from the TxD pin. To sen
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
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 Mo

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
cleared to 0.

14.9.6 Restrictions on Using DTC or DMAC

- When the external clock source is used as a synchronization clock, update TDR by th DMAC and wait for at least five Pϕ clock cycles before allowing the transmit clock to input. If the transmit clock is input within four clock cycles after TDR modification, t may malfunction (figure 14.33).
- When using the DTC or DMAC to read RDR, be sure to set the receive end interrupt the DTC or DMAC activation source.



Note: When external clock is supplied, t must be more than four clock cycles.

Figure 14.33 Sample Transmission using DTC in Clocked Synchronous Mo

SSR, write to TDR, clear TDRE in this order, and then start transmission. To transmit da different transmission mode, initialize the SCI first.

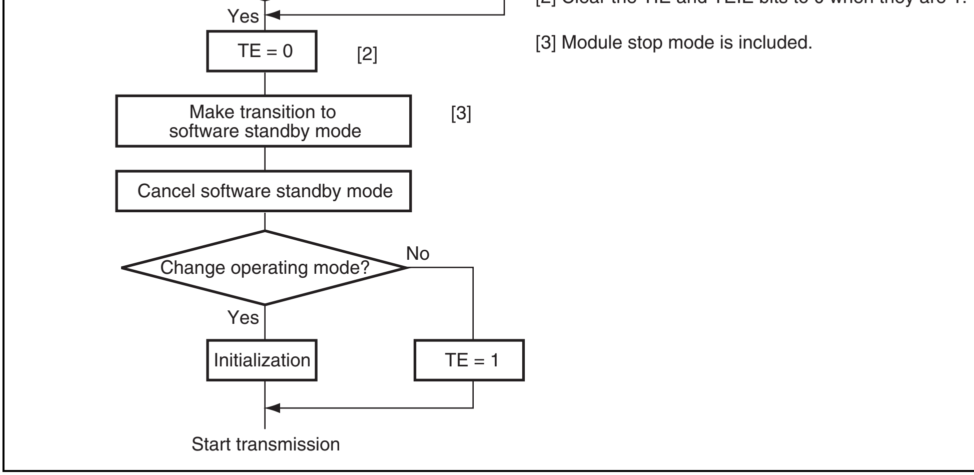
Figure 14.34 shows a sample flowchart for mode transition during transmission. Figures 14.36 show the port pin states during mode transition.

Before making the transition from the transmission mode using DTC transfer to module or software standby mode, stop all transmit operations (TE = TIE = TEIE = 0). Setting t TIE bits to 1 after mode cancellation sets the TXI flag to start transmission using the DT

(2) Reception

Before making the transition to module stop mode or software standby mode, stop the re operations (RE = 0). RSR, RDR, and SSR are reset. If transition is made during data rec data being received will be invalid.

To receive data in the same reception mode after mode cancellation, set the RE bit to 1, start reception. To receive data in a different reception mode, initialize the SCI first.



[2] Clear the TE and TEIE bits to 0 when they are 1.

[3] Module stop mode is included.

Figure 14.34 Sample Flowchart for Mode Transition during Transmission

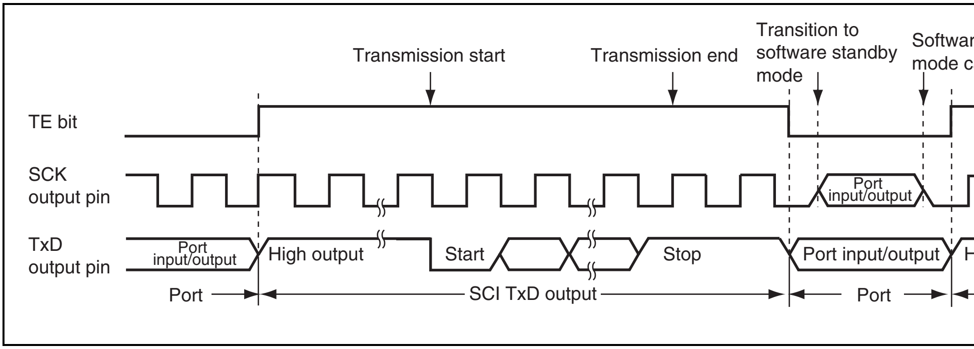
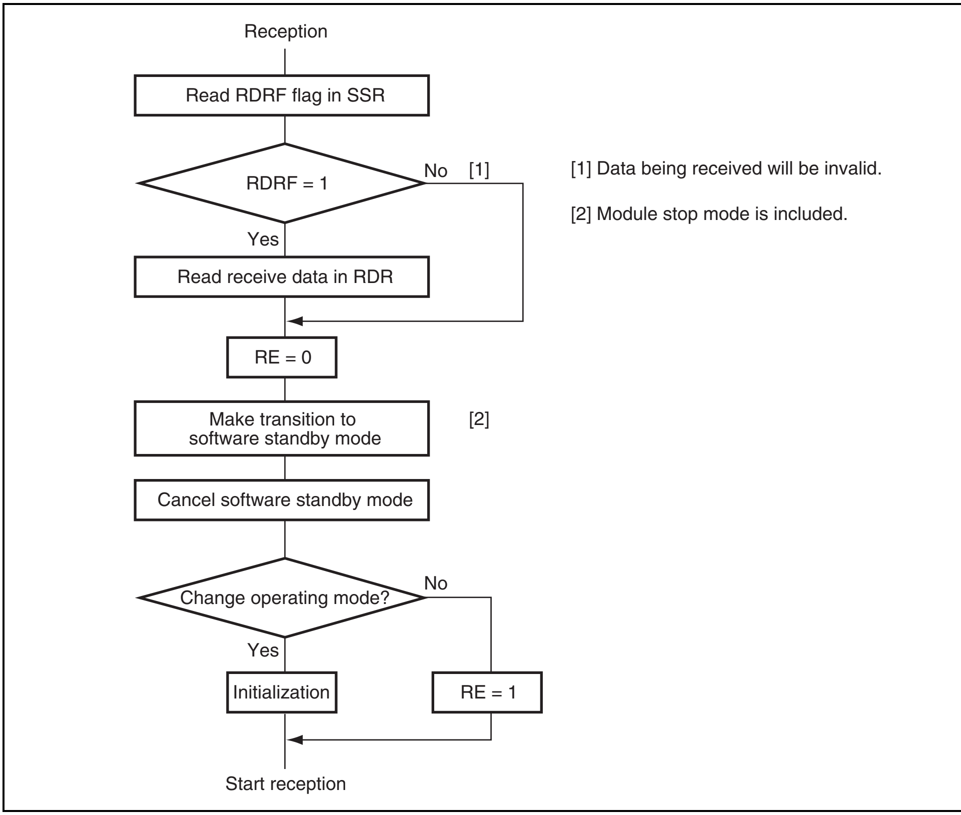


Figure 14.35 Port Pin States during Mode Transition (Internal Clock, Asynchronous Transmission)

Figure 14.36 Port Pin States during Mode Transition (Internal Clock, Clocked Synchronous Transmission)



[1] Data being received will be invalid.

[2] Module stop mode is included.

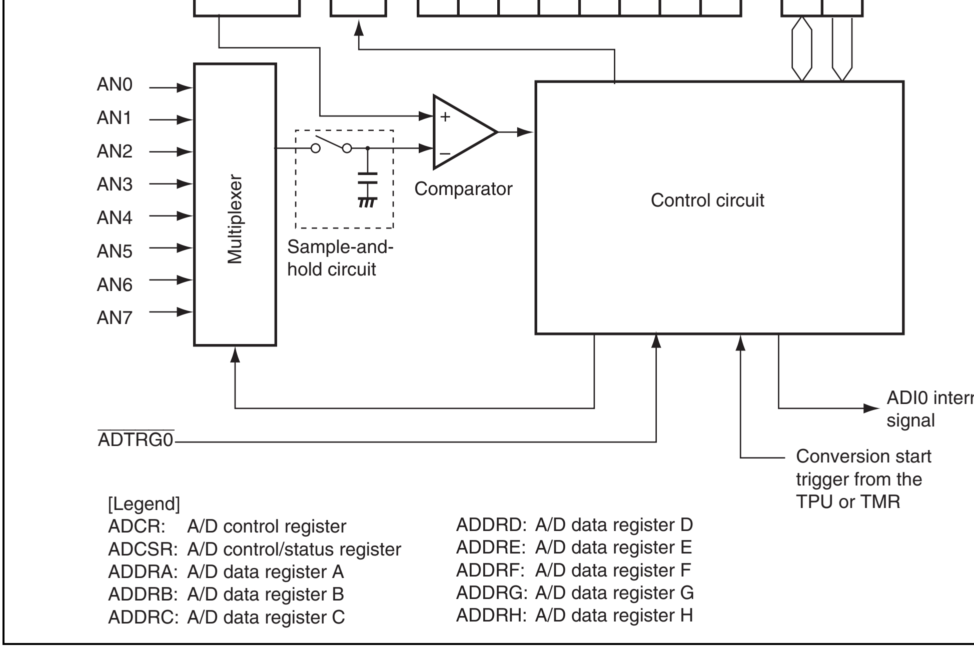
Figure 14.37 Sample Flowchart for Mode Transition during Reception

- 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 mode can be set



[Legend]

ADCR:	A/D control register	ADDRD:	A/D data register D
ADCSR:	A/D control/status register	ADDRE:	A/D data register E
ADDRA:	A/D data register A	ADDRF:	A/D data register F
ADDRB:	A/D data register B	ADDRG:	A/D data register G
ADDRC:	A/D data register C	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
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	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 Resul
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 DTC or DMAC 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 speci channels until this bit is cleared to 0 by softwar or hardware standby mode.

0011: AN3

0100: AN4

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.

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 stopped 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)

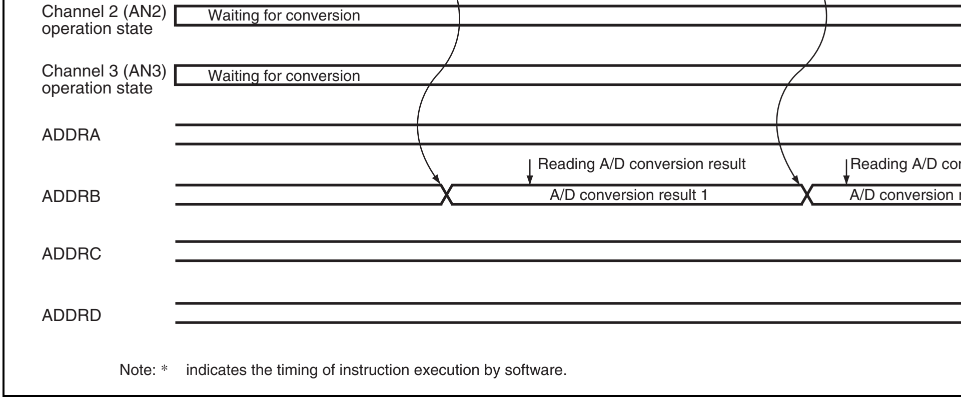
15.4 Operation

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



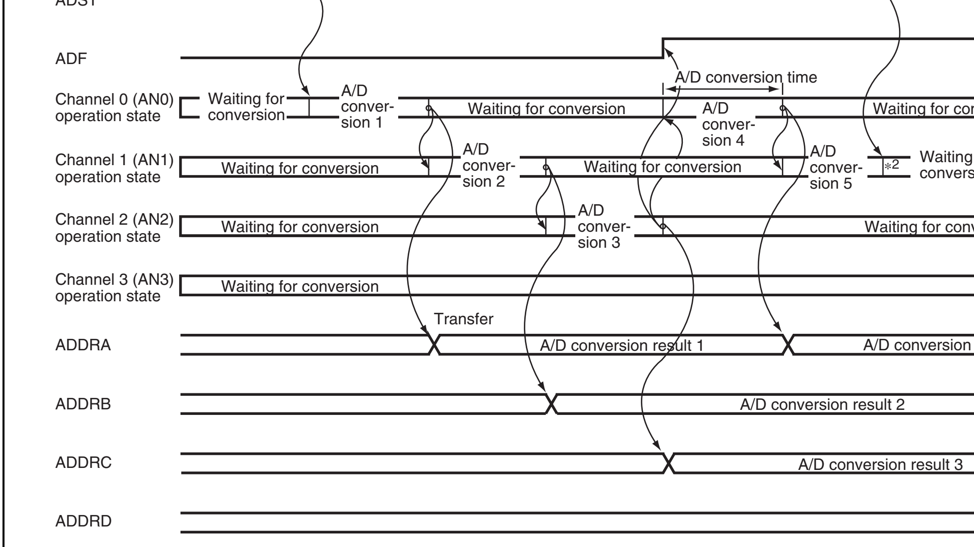
Note: * indicates the timing of instruction execution by software.

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.

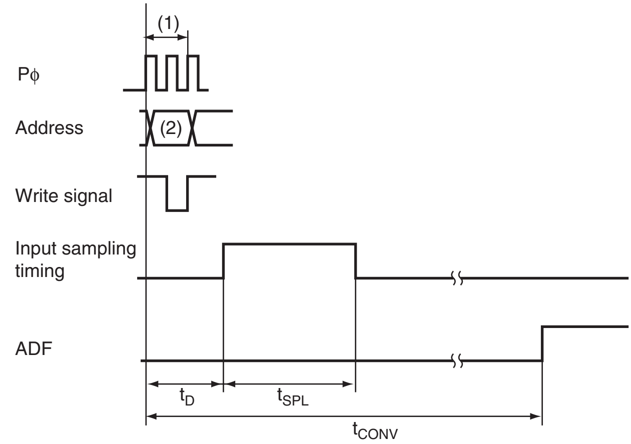
1. When the ADST bit in ADCSR is set to 1 by software, TPU, TMR, or an external tri input, A/D conversion starts on the first channel in the group. Consecutive A/D conv a maximum of four channels (SCANE and SCANS = B'10) or on a maximum of eigh channels (SCANE and SCANS = B'11) can be selected. When 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

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.

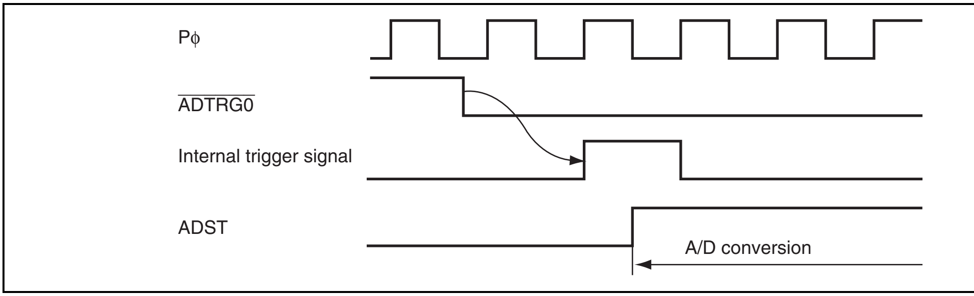


Figure 15.5 External Trigger Input Timing

Name	Interrupt Source	Interrupt Flag	DTC Activation	DMAC A
ADI	A/D conversion end	ADF	Possible	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'3I 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.

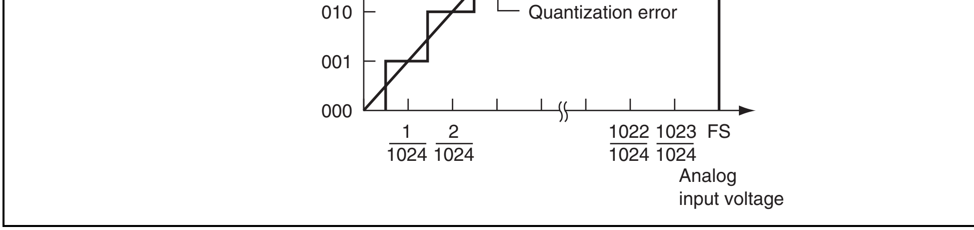


Figure 15.6 A/D Conversion Accuracy Definitions

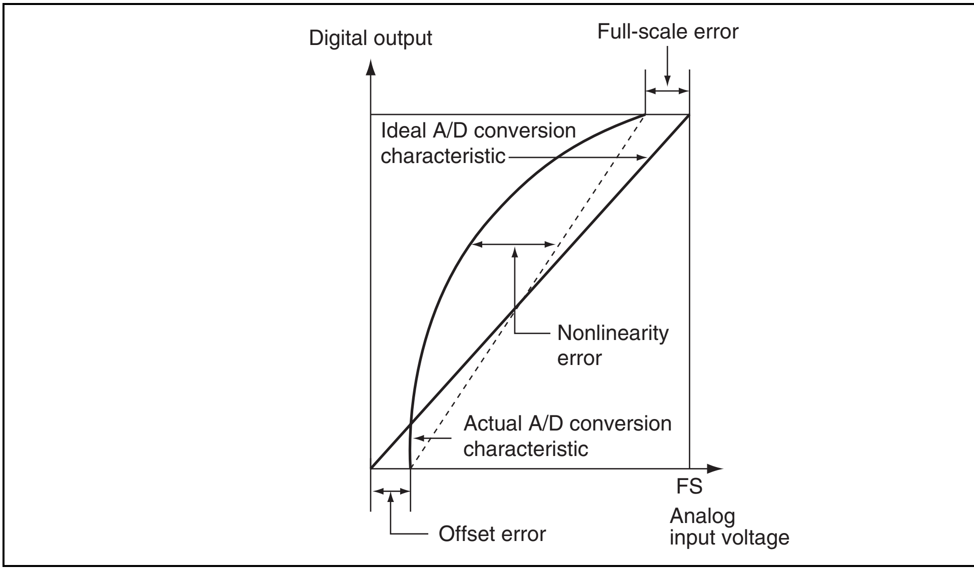


Figure 15.7 A/D Conversion Accuracy Definitions

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 analo 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

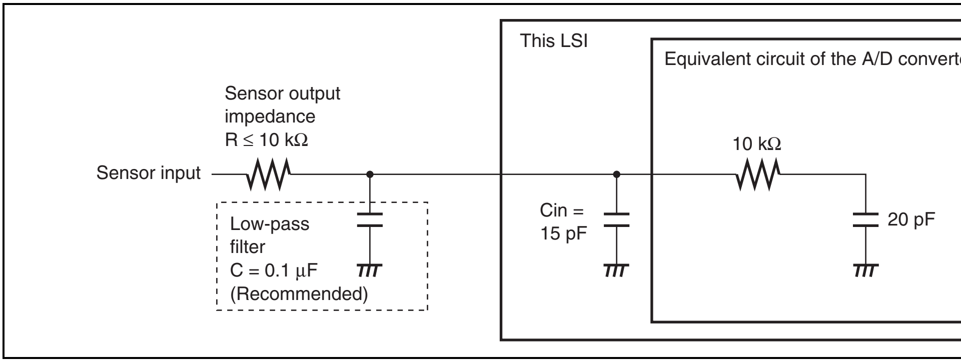


Figure 15.8 Example of Analog Input Circuit

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 ≤ VAN ≤ Vref.

- Relation between AVcc, AVss and Vcc, Vss

  As the relationship between AVcc, AVss and Vcc, Vss, set AVcc = Vcc ± 0.3 V and 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 bo input pin voltage. Careful consideration is therefore required when deciding the circuit c

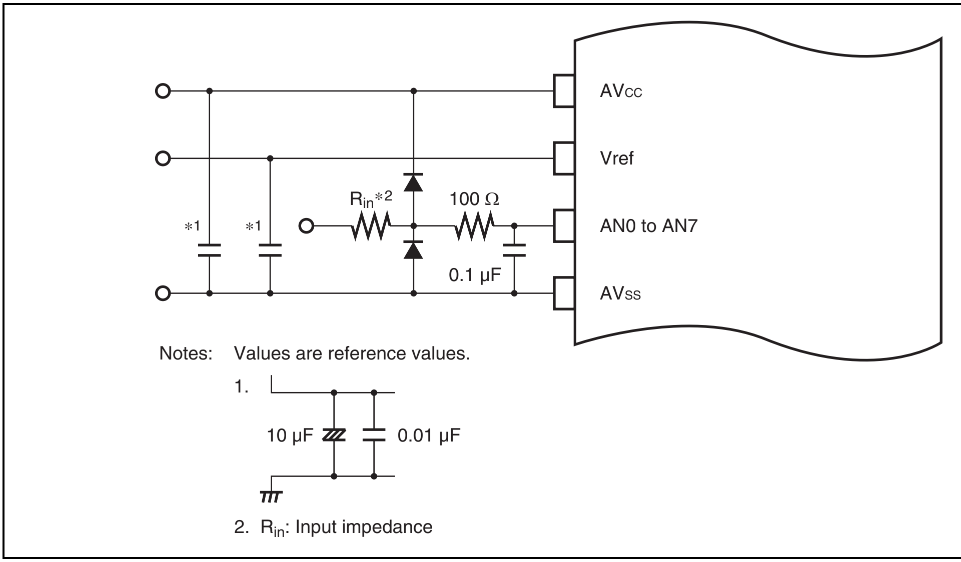


Figure 15.9 Example of Analog Input Protection Circuit

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 mode can be set

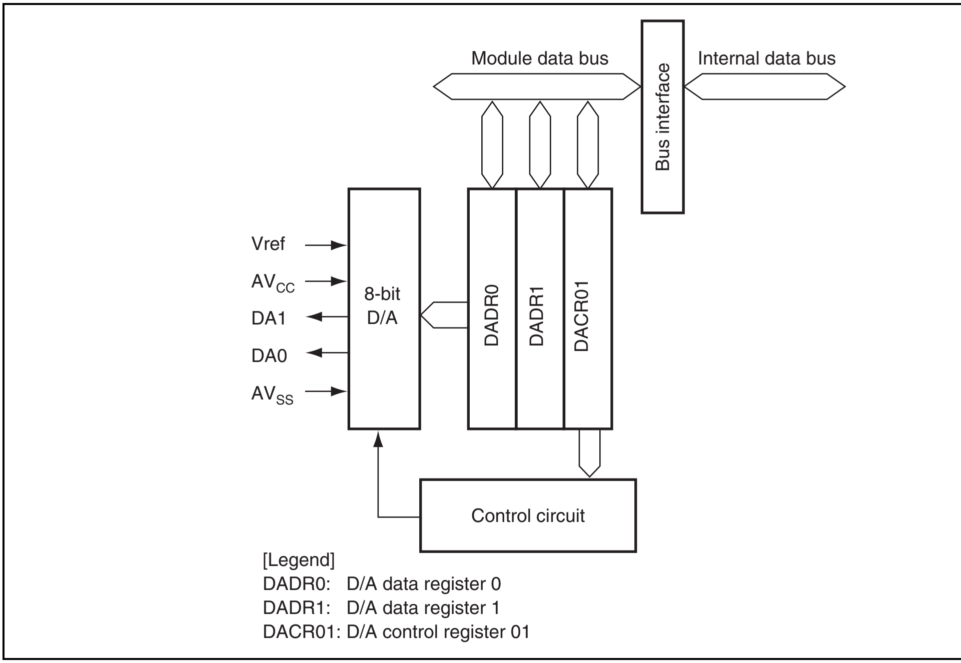


Figure 16.1 Block Diagram of D/A Converter

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)

DADR is an 8-bit readable/writable register that stores data to which D/A conversion is t performed. Whenever analog output is enabled, the values in DADR are converted and ou 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 cor 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.

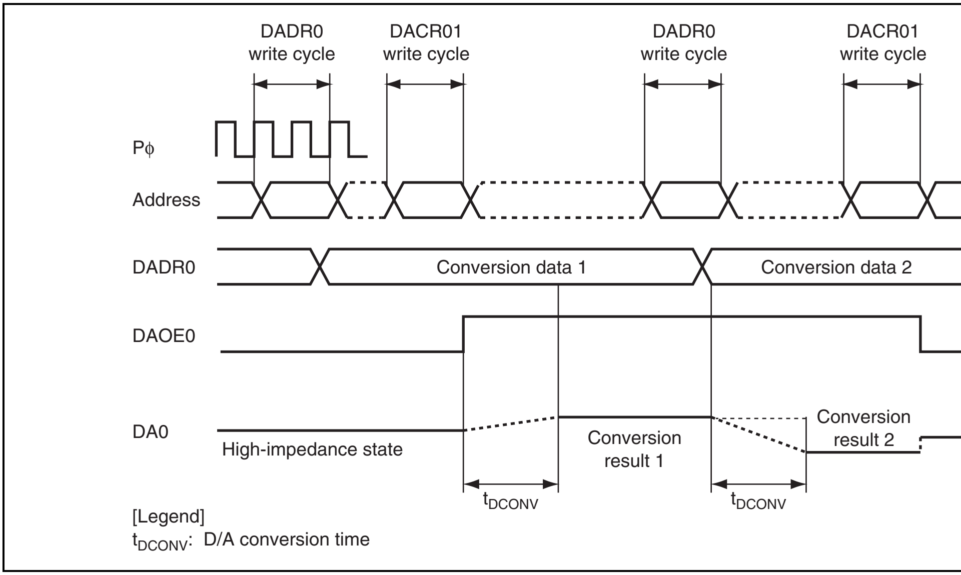


Figure 16.2 Example of D/A Converter Operation

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.

ROMless	H8SX/1651C	40 kbytes	H'FF2000 to H'

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.

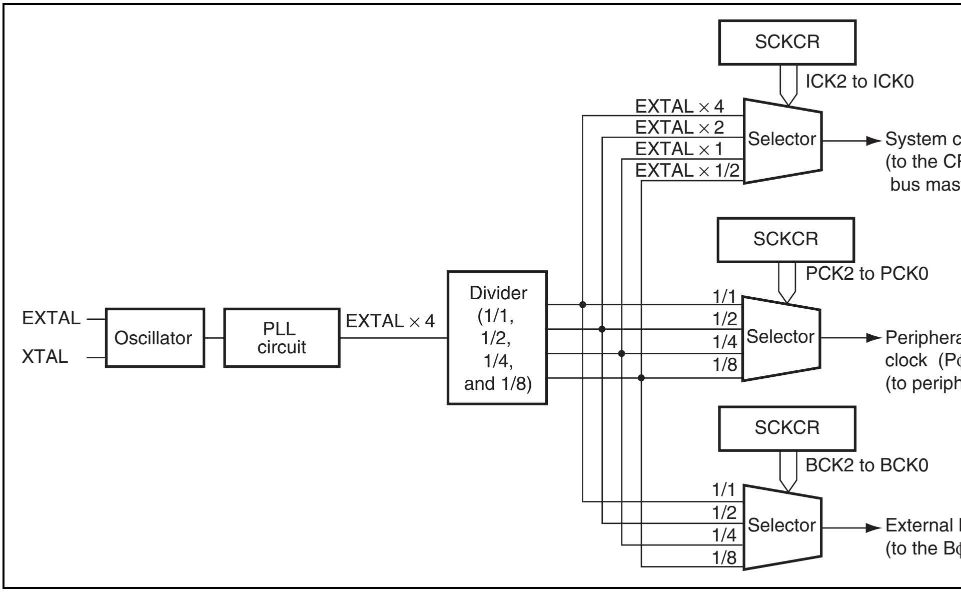


Figure 18.1 Block Diagram of Clock Pulse Generator

Bit	15	14	13	12	11	10	9
Bit Name	PSTOP1	—	POSEL1	—	—	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	Bφ Output Select Controls the φ output on PA7. • Normal operation 0: Bφ output 1: Fixed high
14	—	0	R/W	Reserved This bit is always read as 0. The write value shou always be 0.
13	POSEL1	0	R/W	φ Output Select 1 Controls the φ output on PA7. 0: External bus clock (Bφ) 1: Setting prohibited

				001: × 2 010: × 1 011: × 1/2 1XX: Setting prohibited 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 5 4	PCK2 PCK1 PCK0	0 1 0	R/W R/W R/W	Peripheral Module Clock (Pϕ) Select These bits select the frequency of the periphera 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.

[Legend]

X: Don't care

frequency of 8 to 18 MHz should be connected.

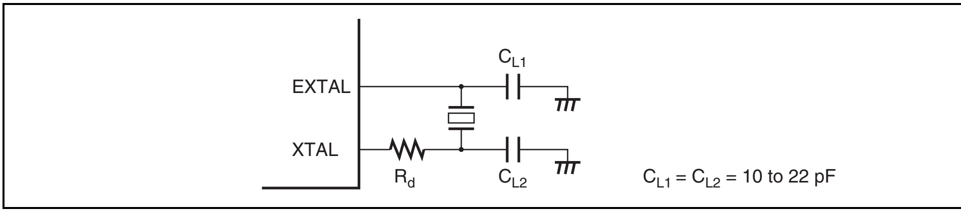


Figure 18.2 Connection of Crystal Resonator (Example)

Table 18.1 Damping Resistance Value

Frequency (MHz)	8	12	18
R_d (Ω)	200	0	0

Figure 18.3 shows an equivalent circuit of the crystal resonator. Use a crystal resonator t the characteristics shown in table 18.2.

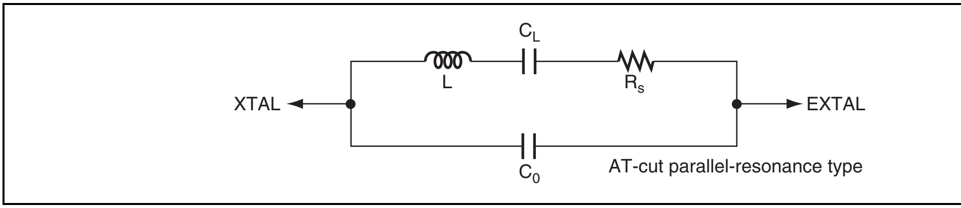


Figure 18.3 Crystal Resonator Equivalent Circuit

input to the XTAL pin, make sure that the external clock is held high in standby mode.

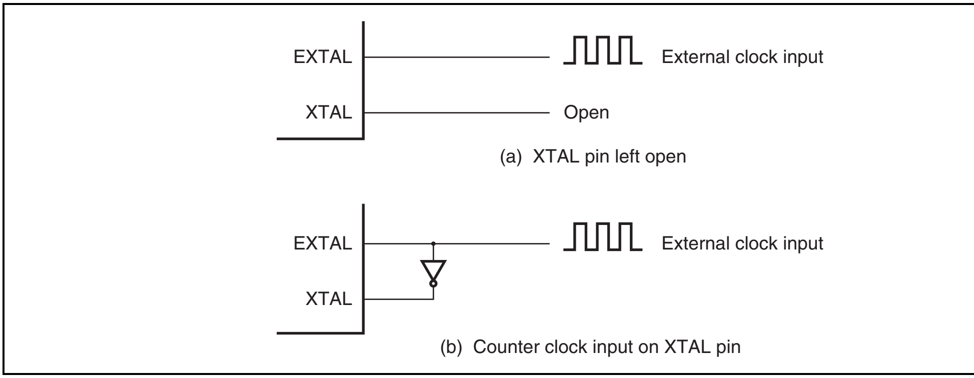


Figure 18.4 External Clock Input (Examples)

For the input conditions of the external clock, refer to tables 21.4 and 21.14 in section 21
Electrical Characteristics. The input external clock should be from 8 to 18 MHz.

18.3 PLL Circuit

The PLL circuit has the function of multiplying the frequency of the clock from the oscil 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

18.4 Frequency Divider

The frequency divider divides the PLL clock to generate a 1/2, 1/4, or 1/8 clock. After bi to ICK0, PCK 2 to PCK0, and BCK2 to BCK0 are modified, this LSI operates at the mod frequency.

≤ Pϕ ≤ 35 MHz, and 8 MHz ≤ Bϕ ≤ 50 MHz.

2. All the on-chip peripheral modules (except for the DTC) operate on the Pϕ. Therefor that the time processing of modules such as a timer and SCI differs before and after the clock division ratio.

   In addition, wait time for clearing software standby mode differs by changing the cl division ratio. For details, see section 19.5.3, 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. Acc 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 18.5 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.

Figure 18.5 Clock Modification Timing

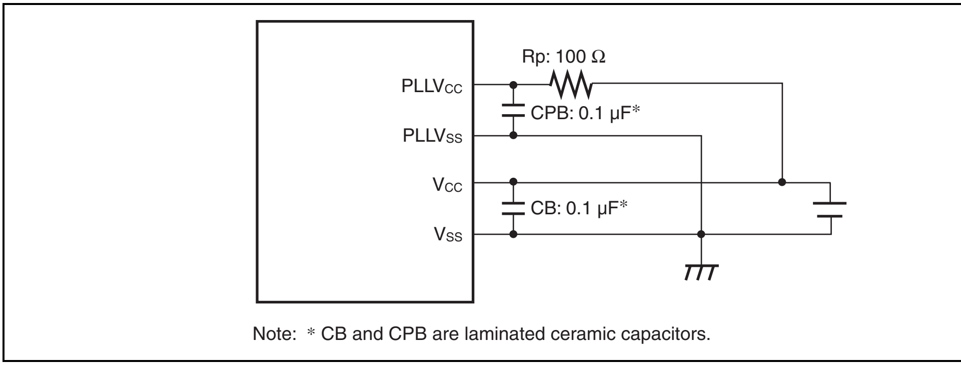
18.5.2 Notes on Resonator

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.

18.5.3 Notes on Board Design

When using the crystal resonator, place the crystal resonator and its load capacitors as clo possible to the XTAL and EXTAL pins. Other signal lines should be routed away from th oscillation circuit as shown in figure 18.6 to prevent induction from interfering with corre oscillation.

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 18.7 Recommended External Circuitry for PLL Circuit

- Module stop function

  The functions for each peripheral module can be stopped to make a transition to a po
  mode.

- 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

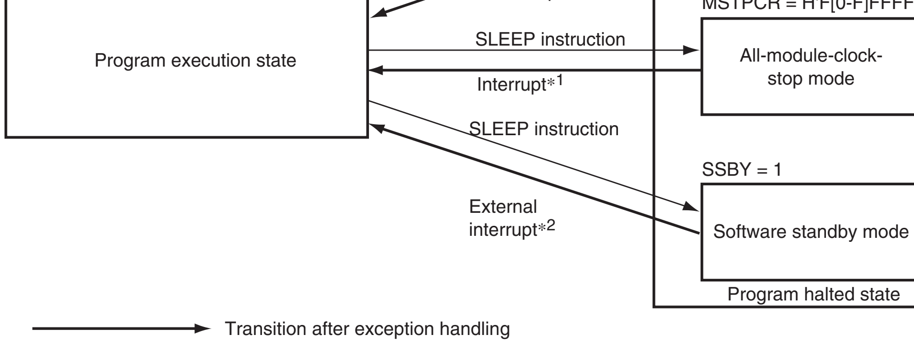
  Software standby mode

  Hardware standby mode

Cancellation method	Interrupt	Interrupt*	External interrupt	
Oscillator	Functioning	Functioning	Halted	Halted
CPU	Halted (retained)	Halted (retained)	Halted (retained)	Halted
Watchdog timer	Functioning	Functioning	Halted (retained)	Halted
8-bit timer	Functioning	Functioning*4	Halted (retained)	Halted
Other peripheral modules	Functioning	Halted*1	Halted*1	Halted*3
I/O port	Functioning	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. "Functioning" or "Halted" is selectable through the setting of bits MSTPA11 to in MSTPCRA. However, pin output is disabled even when "Functioning" is sele



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

Figure 19.1 Mode Transitions

19.2 Register Descriptions

The registers related to the power-down modes are shown below. For details on the syst control register (SCKCR), see section 18.1.1, System Clock Control Register (SCKCR).

- 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 does not change when clearing the softw standby mode by using external interrupts and s normal operation. For clearing, write 0 to this bit. the WDT is used as the watchdog timer, the sett bit is disabled. In this case, a transition is always sleep mode or all-module-clock-stop mode after SLEEP instruction is executed. When the SLPIE to 1, this bit should be cleared to 0.
14	OPE	1	R/W	Output Port Enable Specifies whether the output of the address bus control signals ($\overline{\text{CS0}}$ to $\overline{\text{CS7}}$, $\overline{\text{AS}}$, $\overline{\text{RD}}$, $\overline{\text{HWR}}$, 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

oscillation settling time. With an external clock,
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

				instruction exception handling and does not ca transition to power-down mode. After executio sleep instruction exception handling, this bit re set to 1. Clear the bit by writing 0 to this bit.
6 to 0	—	All 0	R/W	Reserved These bits are always read as 0. The write value always be 0.

Note: X: Don't care

19.2.2 Module Stop Control Registers A, B (MSTPCRA, MSTPCRB)

MSTPCRA and MSTPCRB control module stop mode. Setting a bit to 1 makes the corre module enter module stop mode, while clearing the bit to 0 clears module stop mode.

- MSTPCRA

Bit	15	14	13	12	11	10	9
Bit Name	ACSE	MSTPA14	MSTPA13	MSTPA12	MSTPA11	MSTPA10	MSTPA9
Initial Value	0	0	0	0	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	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

- MSTPCRA

Bit	Bit Name	Initial Value	R/W	Module
15	ACSE	0	R/W	All-Module-Clock-Stop Mode Enable Enables/disables all-module-clock-stop mode fo reducing current consumption by stopping the b controller and I/O ports operations when the CP executes the SLEEP instruction after module st has been set for all the on-chip peripheral modu controlled by MSTPCR. 0: All-module-clock-stop mode disabled 1: All-module-clock-stop mode enabled
14	MSTPA14	0	R/W	Reserved This bit is always read as 0. The write value sho always be 0.
13	MSTPA13	0	R/W	DMA controller (DMAC)
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 valu 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 valu always be 1.

0	MSTPA0	1	R/W	16-bit timer pulse unit (TPU channels 5 to 0)

- MSTPCRB

Bit	Bit Name	Initial Value	R/W	Module
15	MSTPB15	1	R/W	Programmable pulse generator (PPG)
14	MSTPB14	1	R/W	Reserved
13	MSTPB13	1	R/W	These bits are always read as 1. The write value always be 1.
12	MSTPB12	1	R/W	Serial communication interface_4 (SCI_4)
11	MSTPB11	1	R/W	Serial communication interface_3 (SCI_3)
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 value always be 1.
5	MSTPB5	1	R/W	
4	MSTPB4	1	R/W	
3	MSTPB3	1	R/W	
2	MSTPB2	1	R/W	
1	MSTPB1	1	R/W	
0	MSTPB0	1	R/W	

Bit Name	MSTPC7	MSTPC6	MSTPC5	MSTPC4	MSTPC3	MSTPC2	MSTPC1
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 These bits are always read as 1. The write valu always be 1.
14	MSTPC14	1	R/W	
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 These bits are always read as 0. The write valu always be 0.
6	MSTPC6	0	R/W	
5	MSTPC5	0	R/W	
4	MSTPC4	0	R/W	On-chip RAM_4 (H'FFF2000 to H'FFF3FFF)
3	MSTPC3	0	R/W	On-chip RAM_3 (H'FFF4000 to H'FFF5FFF)
2	MSTPC2	0	R/W	On-chip RAM_2 (H'FFF6000 to H'FFF7FFF)
1	MSTPC1	0	R/W	On-chip RAM_1 (H'FFF8000 to H'FFF9FFF)
0	MSTPC0	0	R/W	On-chip RAM_0 (H'FFFA000 to H'FFFBFFF)

the operating clock specified by bits ICK2 to ICK0.

Multi-clock mode is cleared by clearing all of bits ICK2 to ICK0, PCK2 to PCK0, and BC
BCK0 to 0. A transition is made to normal mode at the end of the bus cycle, and multi-cl
is cleared.

If a SLEEP instruction is executed while the SSBY bit in SBYCR is cleared to 0, this LS
sleep mode. When sleep mode is cleared by an interrupt, multi-clock mode is restored.

If a SLEEP instruction is executed while the SSBY bit in SBYCR is set to 1, this LSI ente
software standby mode. When software standby mode is cleared by an external interrupt,
clock mode is restored.

When the $\overline{\text{RES}}$ pin is driven low, the reset state is entered and multi-clock mode is cleare
same applies to a reset caused by watchdog timer overflow.

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

Sleep mode is exited by any interrupt, signals on the $\overline{\text{RES}}$ or $\overline{\text{STBY}}$ pin, and a reset caus
watchdog timer overflow.

1. Clearing by interrupt

   When an interrupt occurs, sleep mode is exited and interrupt exception processing st
   mode is not exited if the interrupt is disabled, or interrupts other than NMI are mask
   CPU.

2. Clearing by $\overline{\text{RES}}$ pin

   Setting the $\overline{\text{RES}}$ pin level low selects the reset state. After the stipulated reset input d
   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 m

4. Clearing by reset caused by watchdog timer overflow

   Sleep mode is exited by an internal reset caused by a watchdog timer overflow.

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.

19.5.2 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 19.2 Oscillation Settling Time Settings

					Standby	Pϕ* [MHz]		
STS4	STS3	STS2	STS1	STS0	Time	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.

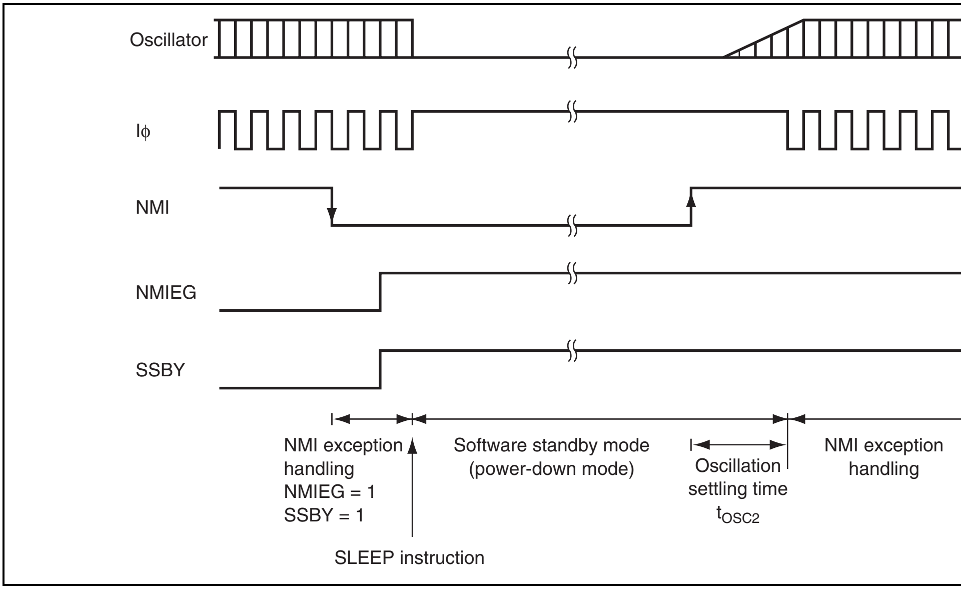


Figure 19.2 Software Standby Mode Application Example

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.

19.6.2 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 19.2). When the $\overline{\text{RES}}$ pin is subsequently driven hig
transition is made to the program execution state via the reset exception handling state.

19.6.3 Hardware Standby Mode Timing

Figure 19.3 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 hig
waiting for the oscillation settling time, then changing the $\overline{\text{RES}}$ pin from low to high.

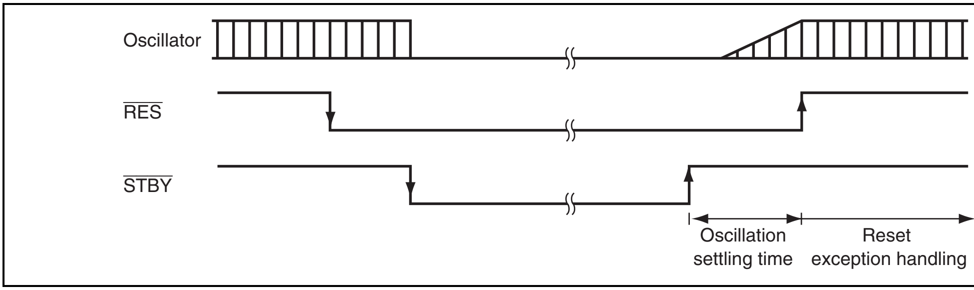


Figure 19.3 Hardware Standby Mode Timing

Timing.

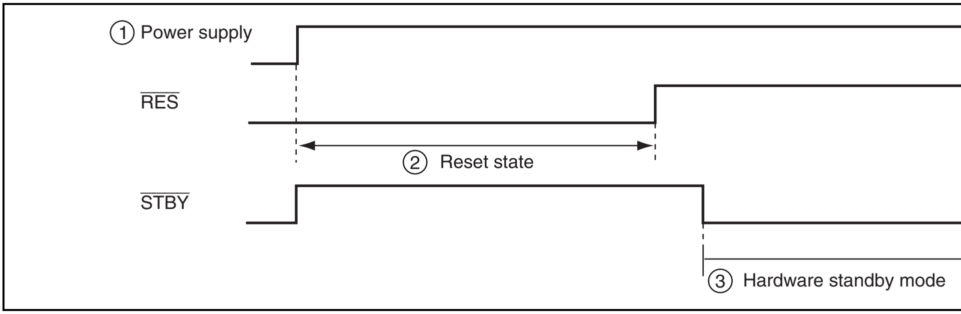


Figure 19.4 Timing Sequence at Power-On

When the corresponding MSTP bit is cleared to 0, module stop mode is cleared and the m starts operating at the end of the bus cycle. In module stop mode, the internal states of mo other than the SCI are retained.

After the reset state is cleared, all modules other than the DTC, DMAC, or on-chip RAM module stop mode.

The registers of the module for which module stop mode is selected cannot be read from to.

19.7.2 All-Module-Clock-Stop Mode

When the ACSE bit is set to 1 and all modules controlled by MSTPCR are stopped (MST MSTPCRB = H'FFFFFFFF), or all modules except for the 8-bit timer are stopped (MSTP MSTPCRB = H'F[0 to F]FFFFFF), executing a SLEEP instruction with the SSBY bit in S cleared to 0 will cause all modules (except for the 8-bit timer* and watchdog timer), the b controller, and the I/O ports to stop operating, and to make a transition to all-module-cloc mode at the end of the bus cycle.

All-module-clock-stop mode is cleared by an external interrupt (NMI or $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ11}}$ p $\overline{\text{RES}}$ pin input, or an internal interrupt (8-bit timer* or watchdog timer), and the CPU retu normal program execution state via the exception handling state. All-module-clock-stop n not cleared if interrupts are disabled, if interrupts other than NMI are masked on the CPU 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 MS MSTPCRA.

Transitions to the power-down state are inhibited when sleep instruction exception hand
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
19.5).

When a canceling factor interrupt is generated immediately before the execution of a SL
instruction, exception handling for the interrupt starts. When execution returns from the
service routine, the SLEEP instruction is executed to enter the power-down state. In this
power-down state is not canceled until the next canceling factor interrupt is generated (s
19.6).

When the SLPIE bit is set to 1 in the service routine for a canceling factor interrupt so th
execution of a SLEEP instruction will produce sleep instruction exception handling, the
of the system is as shown in figure 19.7. Even if a canceling factor interrupt is generated
immediately before the SLEEP instruction is executed, sleep instruction exception hand
initiated by execution of the SLEEP instruction. Therefore, the CPU executes the instruc
follows the SLEEP instruction after sleep instruction exception and exception service ro
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 S
to 0.

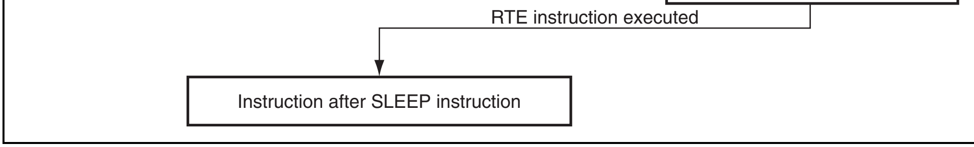


Figure 19.5 When Canceling Factor Interrupt is Generated after SLEEP Instruction Execution

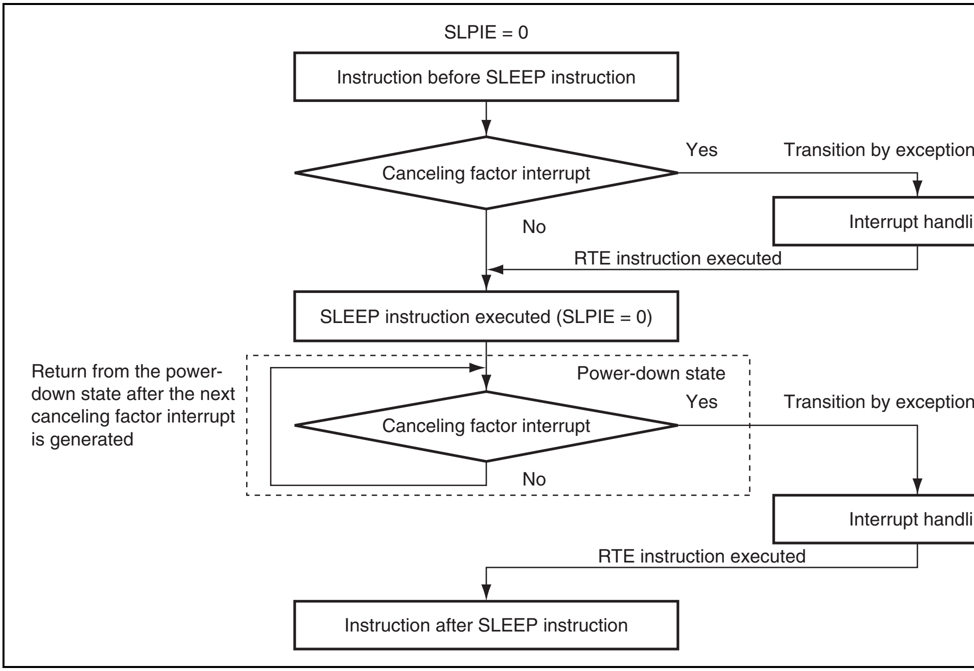

SLPIE = 0
Instruction before SLEEP instruction
Canceling factor interrupt
Yes
Transition by exception
Interrupt handli
No
RTE instruction executed
SLEEP instruction executed (SLPIE = 0)
Return from the power-down state after the next canceling factor interrupt is generated
Power-down state
Yes
Transition by exception
Canceling factor interrupt
No
Interrupt handli
RTE instruction executed
Instruction after SLEEP instruction

Figure 19.6 When Canceling Factor Interrupt is Generated before SLEEP Instruction Execution (Sleep Instruction Exception Handling Not In

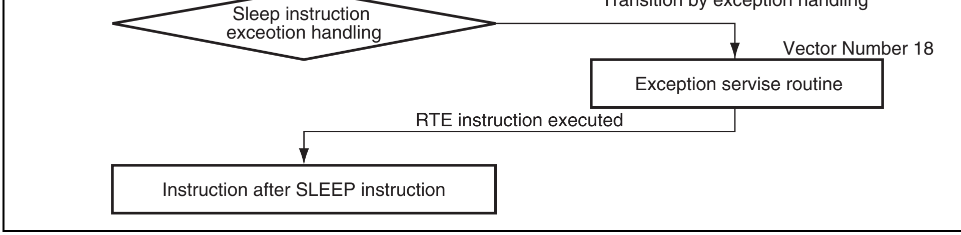

Transition by exception handling
Sleep instruction
exceotion handling
Vector Number 18
Exception servise routine
RTE instruction executed
Instruction after SLEEP instruction

Figure 19.7 When Canceling Factor Interrupt is Generated before SLEEP Instruction Execution (Sleep Instruction Exception Handling Ini

Table 19.3 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

19.10.3 Module Stop Mode of DMAC or DTC

Depending on the operating state of the DMAC and DTC, bits MSTPA13 and MSTPA1
be set to 1, respectively. The module stop mode setting for the DMAC or DTC should b
out only when the DMAC or DTC is not activated.

For details, refer to section 7, DMA Controller (DMAC), and section 8, Data Transfer C
(DTC).

19.10.4 On-Chip Peripheral Module Interrupts

Relevant interrupt operations cannot be performed in module stop mode. Consequently,
stop mode is entered when an interrupt has been requested, it will not be possible to clea
interrupt source or the DMAC or DTC activation source. Interrupts should therefore be
before entering module stop mode.

19.10.5 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 correspondi 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 B data direction register	PBDDR	8	H'FFB8A	I/O port	8	2F
Port D data direction register	PDDDR	8	H'FFB8C	I/O port	8	2F
Port E data direction register	PEDDR	8	H'FFB8D	I/O port	8	2F
Port F data direction register	PFDDR	8	H'FFB8E	I/O port	8	2F
Port 1 input buffer control register	P1ICR	8	H'FFB90	I/O port	8	2F
Port 2 input buffer control register	P2ICR	8	H'FFB91	I/O port	8	2F
Port 3 input buffer control register	P3ICR	8	H'FFB92	I/O port	8	2F
Port 5 input buffer control register	P5ICR	8	H'FFB94	I/O port	8	2F
Port 6 input buffer control register	P6ICR	8	H'FFB95	I/O port	8	2F
Port A input buffer control register	PAICR	8	H'FFB99	I/O port	8	2F
Port B input buffer control register	PBICR	8	H'FFB9A	I/O port	8	2F
Port D input buffer control register	PDICR	8	H'FFB9C	I/O port	8	2F
Port E input buffer control register	PEICR	8	H'FFB9D	I/O port	8	2F
Port F input buffer control register	PFICR	8	H'FFB9E	I/O port	8	2F
Port H register	PORTH	8	H'FFBA0	I/O port	8	2F
Port I register	PORTI	8	H'FFBA1	I/O port	8	2F
Port H data register	PHDR	8	H'FFBA4	I/O port	8	2F
Port I data register	PIDR	8	H'FFBA5	I/O port	8	2F
Port H data direction register	PHDDR	8	H'FFBA8	I/O port	8	2F
Port I data direction register	PIDDR	8	H'FFBA9	I/O port	8	2F

Port 2 open drain control register	P2ODR	8	H'FFBBC	I/O port	8	2
Port F open drain control register	PFODR	8	H'FFBBD	I/O port	8	2
Port function control register 0	PFCR0	8	H'FFBC0	I/O port	8	2
Port function control register 1	PFCR1	8	H'FFBC1	I/O port	8	2
Port function control register 2	PFCR2	8	H'FFBC2	I/O port	8	2
Port function control register 4	PFCR4	8	H'FFBC4	I/O port	8	2
Port function control register 6	PFCR6	8	H'FFBC6	I/O port	8	2
Port function control register 7	PFCR7	8	H'FFBC7	I/O port	8	2
Port function control register 9	PFCR9	8	H'FFBC9	I/O port	8	2
Port function control register B	PFCRB	8	H'FFBCB	I/O port	8	2
Port function control register C	PFCRC	8	H'FFBCC	I/O port	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_2	DSAR_2	32	H'FFC40	DMAC_2	16	2I
DMA destination address register_2	DDAR_2	32	H'FFC44	DMAC_2	16	2I
DMA offset register_2	DOFR_2	32	H'FFC48	DMAC_2	16	2I
DMA transfer count register_2	DTCR_2	32	H'FFC4C	DMAC_2	16	2I
DMA block size register_2	DBSR_2	32	H'FFC50	DMAC_2	16	2I
DMA mode control register_2	DMDR_2	32	H'FFC54	DMAC_2	16	2I
DMA address control register_2	DACR_2	32	H'FFC58	DMAC_2	16	2I
DMA source address register_3	DSAR_3	32	H'FFC60	DMAC_3	16	2I
DMA destination address register_3	DDAR_3	32	H'FFC64	DMAC_3	16	2I
DMA offset register_3	DOFR_3	32	H'FFC68	DMAC_3	16	2I
DMA transfer count register_3	DTCR_3	32	H'FFC6C	DMAC_3	16	2I
DMA block size register_3	DBSR_3	32	H'FFC70	DMAC_3	16	2I
DMA mode control register_3	DMDR_3	32	H'FFC74	DMAC_3	16	2I
DMA address control register_3	DACR_3	32	H'FFC78	DMAC_3	16	2I
DMA module request select register_0	DMRSR_0	8	H'FFD20	DMAC_0	16	2I
DMA module request select register_1	DMRSR_1	8	H'FFD21	DMAC_1	16	2I
DMA module request select register_2	DMRSR_2	8	H'FFD22	DMAC_2	16	2I
DMA module request select register_3	DMRSR_3	8	H'FFD23	DMAC_3	16	2I

Interrupt priority register I	IPRI	16	H'FFD50	INTC	16	2
Interrupt priority register K	IPRK	16	H'FFD54	INTC	16	2
Interrupt priority register L	IPRL	16	H'FFD56	INTC	16	2
IRQ sense control register H	ISCRH	16	H'FFD68	INTC	16	2
IRQ sense control register L	ISCRL	16	H'FFD6A	INTC	16	2
DTC vector base register	DTCVBR	32	H'FFD80	BSC	16	2
Bus width control register	ABWCR	16	H'FFD84	BSC	16	2
Access state control register	ASTCR	16	H'FFD86	BSC	16	2
Wait control register A	WTCRA	16	H'FFD88	BSC	16	2
Wait control register B	WTCRB	16	H'FFD8A	BSC	16	2
Read strobe timing control register	RDNCR	16	H'FFD8C	BSC	16	2
$\overline{\text{CS}}$ assert period control register	CSACR	16	H'FFD8E	BSC	16	2
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

Serial extended mode register_2	SEMR_2	8	H'FFE84	SCI_2	8	2F
Serial mode register_3	SMR_3	8	H'FFE88	SCI_3	8	2F
Bit rate register_3	BRR_3	8	H'FFE89	SCI_3	8	2F
Serial control register_3	SCR_3	8	H'FFE8A	SCI_3	8	2F
Transmit data register_3	TDR_3	8	H'FFE8B	SCI_3	8	2F
Serial status register_3	SSR_3	8	H'FFE8C	SCI_3	8	2F
Receive data register_3	RDR_3	8	H'FFE8D	SCI_3	8	2F
Smart card mode register_3	SCMR_3	8	H'FFE8E	SCI_3	8	2F
Serial mode register_4	SMR_4	8	H'FFE90	SCI_4	8	2F
Bit rate register_4	BRR_4	8	H'FFE91	SCI_4	8	2F
Serial control register_4	SCR_4	8	H'FFE92	SCI_4	8	2F
Transmit data register_4	TDR_4	8	H'FFE93	SCI_4	8	2F
Serial status register_4	SSR_4	8	H'FFE94	SCI_4	8	2F
Receive data register_4	RDR_4	8	H'FFE95	SCI_4	8	2F
Smart card mode register_4	SCMR_4	8	H'FFE96	SCI_4	8	2F
Timer control register_2	TCR_2	8	H'FFEC0	TMR_2	16	2F
Timer control register_3	TCR_3	8	H'FFEC1	TMR_3	16	2F
Timer control/status register_2	TCSR_2	8	H'FFEC2	TMR_2	16	2F
Timer control/status register_3	TCSR_3	8	H'FFEC3	TMR_3	16	2F

Timer counter control register_3	TCCR_3	8	H'FFECB	TMR_3	16	2
Timer control register_4	TCR_4	8	H'FFEE0	TPU_4	16	2
Timer mode register_4	TMDR_4	8	H'FFEE1	TPU_4	16	2
Timer I/O control register_4	TIOR_4	8	H'FFEE2	TPU_4	16	2
Timer interrupt enable register_4	TIER_4	8	H'FFEE4	TPU_4	16	2
Timer status register_4	TSR_4	8	H'FFEE5	TPU_4	16	2
Timer counter_4	TCNT_4	16	H'FFEE6	TPU_4	16	2
Timer general register A_4	TGRA_4	16	H'FFEE8	TPU_4	16	2
Timer general register B_4	TGRB_4	16	H'FFEEA	TPU_4	16	2
Timer control register_5	TCR_5	8	H'FFEF0	TPU_5	16	2
Timer mode register_5	TMDR_5	8	H'FFEF1	TPU_5	16	2
Timer I/O control register_5	TIOR_5	8	H'FFEF2	TPU_5	16	2
Timer interrupt enable register_5	TIER_5	8	H'FFEF4	TPU_5	16	2
Timer status register_5	TSR_5	8	H'FFEF5	TPU_5	16	2
Timer counter_5	TCNT_5	16	H'FFEF6	TPU_5	16	2
Timer general register A_5	TGRA_5	16	H'FFEF8	TPU_5	16	2
Timer general register B_5	TGRB_5	16	H'FFEFA	TPU_5	16	2
DTC enable register A	DTCERA	16	H'FFF20	INTC	16	2
DTC enable register B	DTCERB	16	H'FFF22	INTC	16	2
DTC enable register C	DTCERC	16	H'FFF24	INTC	16	2
DTC enable register D	DTCERD	16	H'FFF26	INTC	16	2

IRQ enable register	IER	16	H'FFF34	INTC	16	2I
IRQ status register	ISR	16	H'FFF36	INTC	16	2I
Port 1 register	PORT1	8	H'FFF40	I/O port	8	2P
Port 2 register	PORT2	8	H'FFF41	I/O port	8	2P
Port 3 register	PORT3	8	H'FFF42	I/O port	8	2P
Port 5 register	PORT5	8	H'FFF44	I/O port	8	2P
Port 6 register	PORT6	8	H'FFF45	I/O port	8	2P
Port A register	PORTA	8	H'FFF49	I/O port	8	2P
Port B register	PORTB	8	H'FFF4A	I/O port	8	2P
Port D register	PORTD	8	H'FFF4C	I/O port	8	2P
Port E register	PORTE	8	H'FFF4D	I/O port	8	2P
Port F register	PORTF	8	H'FFF4E	I/O port	8	2P
Port 1 data register	P1DR	8	H'FFF50	I/O port	8	2P
Port 2 data register	P2DR	8	H'FFF51	I/O port	8	2P
Port 3 data register	P3DR	8	H'FFF52	I/O port	8	2P
Port 6 data register	P6DR	8	H'FFF55	I/O port	8	2P
Port A data register	PADR	8	H'FFF59	I/O port	8	2P
Port B data register	PBDR	8	H'FFF5A	I/O port	8	2P
Port D data register	PDDR	8	H'FFF5C	I/O port	8	2P
Port E data register	PEDR	8	H'FFF5D	I/O port	8	2P
Port F data register	PFDR	8	H'FFF5E	I/O port	8	2P

D/A data register 0	DADR0	8	H'FFF68	D/A	8	2
D/A data register 1	DADR1	8	H'FFF69	D/A	8	2
D/A control register 01	DACR01	8	H'FFF6A	D/A	8	2
PPG output control register	PCR	8	H'FFF76	PPG	8	2
PPG output mode register	PMR	8	H'FFF77	PPG	8	2
Next data enable register H	NDERH	8	H'FFF78	PPG	8	2
Next data enable register L	NDERL	8	H'FFF79	PPG	8	2
Output data register H	PODRH	8	H'FFF7A	PPG	8	2
Output data register L	PODRL	8	H'FFF7B	PPG	8	2
Next data register H*	NDRH	8	H'FFF7C	PPG	8	2
Next data register L*	NDRL	8	H'FFF7D	PPG	8	2
Next data register H*	NDRH	8	H'FFF7E	PPG	8	2
Next data register L*	NDRL	8	H'FFF7F	PPG	8	2
Serial mode register_0	SMR_0	8	H'FFF80	SCI_0	8	2
Bit rate register_0	BRR_0	8	H'FFF81	SCI_0	8	2
Serial control register_0	SCR_0	8	H'FFF82	SCI_0	8	2
Transmit data register_0	TDR_0	8	H'FFF83	SCI_0	8	2
Serial status register_0	SSR_0	8	H'FFF84	SCI_0	8	2
Receive data register_0	RDR_0	8	H'FFF85	SCI_0	8	2
Smart card mode register_0	SCMR_0	8	H'FFF86	SCI_0	8	2

A/D data register A	ADDRA	16	H'FFF90	A/D	16	2F
A/D data register B	ADDRB	16	H'FFF92	A/D	16	2F
A/D data register C	ADDRC	16	H'FFF94	A/D	16	2F
A/D data register D	ADDRD	16	H'FFF96	A/D	16	2F
A/D data register E	ADDRE	16	H'FFF98	A/D	16	2F
A/D data register F	ADDRF	16	H'FFF9A	A/D	16	2F
A/D data register G	ADDRG	16	H'FFF9C	A/D	16	2F
A/D data register H	ADDRH	16	H'FFF9E	A/D	16	2F
A/D control/status register	ADCSR	8	H'FFFA0	A/D	16	2F
A/D control register	ADCR	8	H'FFFA1	A/D	16	2F
Timer control/status register	TCSR	8	H'FFFA4	WDT		2F
Timer counter	TCNT	8	H'FFFA5	WDT		2F
Reset control/status register	RSTCSR	8	H'FFFA7	WDT		2F
Timer control register_0	TCR_0	8	H'FFFB0	TMR_0	16	2F
Timer control register_1	TCR_1	8	H'FFFB1	TMR_1	16	2F
Timer control/status register_0	TCSR_0	8	H'FFFB2	TMR_0	16	2F
Timer control/status register_1	TCSR_1	8	H'FFFB3	TMR_1	16	2F
Time constant register A_0	TCORA_0	8	H'FFFB4	TMR_0	16	2F
Time constant register A_1	TCORA_1	8	H'FFFB5	TMR_1	16	2F
Time constant register B_0	TCORB_0	8	H'FFFB6	TMR_0	16	2F
Time constant register B_1	TCORB_1	8	H'FFFB7	TMR_1	16	2F

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 interrupt enable register_0	TIER_0	8	H'FFFC4	TPU_0	16	2
Timer status register_0	TSR_0	8	H'FFFC5	TPU_0	16	2
Timer counter_0	TCNT_0	16	H'FFFC6	TPU_0	16	2
Timer general register A_0	TGRA_0	16	H'FFFC8	TPU_0	16	2
Timer general register B_0	TGRB_0	16	H'FFFCA	TPU_0	16	2
Timer general register C_0	TGRC_0	16	H'FFFCC	TPU_0	16	2
Timer general register D_0	TGRD_0	16	H'FFFCE	TPU_0	16	2
Timer control register_1	TCR_1	8	H'FFFD0	TPU_1	16	2
Timer mode register_1	TMDR_1	8	H'FFFD1	TPU_1	16	2
Timer I/O control register_1	TIOR_1	8	H'FFFD2	TPU_1	16	2
Timer interrupt enable register_1	TIER_1	8	H'FFFD4	TPU_1	16	2
Timer status register_1	TSR_1	8	H'FFFD5	TPU_1	16	2
Timer counter_1	TCNT_1	16	H'FFFD6	TPU_1	16	2
Timer general register A_1	TGRA_1	16	H'FFFD8	TPU_1	16	2
Timer general register B_1	TGRB_1	16	H'FFFDA	TPU_1	16	2

Timer general register B_2	TGRB_2	16	H'FFFEA	TPU_2	16	2F
Timer control register_3	TCR_3	8	H'FFFF0	TPU_3	16	2F
Timer mode register_3	TMDR_3	8	H'FFFF1	TPU_3	16	2F
Timer I/O control register H_3	TIORH_3	8	H'FFFF2	TPU_3	16	2F
Timer I/O control register L_3	TIORL_3	8	H'FFFF3	TPU_3	16	2F
Timer interrupt enable register_3	TIER_3	8	H'FFFF4	TPU_3	16	2F
Timer status register_3	TSR_3	8	H'FFFF5	TPU_3	16	2F
Timer counter_3	TCNT_3	16	H'FFFF6	TPU_3	16	2F
Timer general register A_3	TGRA_3	16	H'FFFF8	TPU_3	16	2F
Timer general register B_3	TGRB_3	16	H'FFFFA	TPU_3	16	2F
Timer general register C_3	TGRC_3	16	H'FFFFC	TPU_3	16	2F
Timer general register D_3	TGRD_3	16	H'FFFFE	TPU_3	16	2F

Note: * When the same output trigger is specified for pulse output groups 2 and 3 by th
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 outp
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 ar
H'FFF7F and H'FFF7D, respectively.

P5DDR	P57DDR	P56DDR	P55DDR	P54DDR	P53DDR	P52DDR	P51DDR	P50DDR
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

PIICR	PI7ICR	PI6ICR	PI5ICR	PI4ICR	PI3ICR	PI2ICR	PI1ICR	PI0ICR
PDPCR	PD7PCR	PD6PCR	PD5PCR	PD4PCR	PD3PCR	PD2PCR	PD1PCR	PD0PCR
PEPCR	PE7PCR	PE6PCR	PE5PCR	PE4PCR	PE3PCR	PE2PCR	PE1PCR	PE0PCR
PFPCR	PF7PCR	PF6PCR	PF5PCR	PF4PCR	PF3PCR	PF2PCR	PF1PCR	PF0PCR
PHPCR	PH7PCR	PH6PCR	PH5PCR	PH4PCR	PH3PCR	PH2PCR	PH1PCR	PH0PCR
PIPCR	PI7PCR	PI6PCR	PI5PCR	PI4PCR	PI3PCR	PI2PCR	PI1PCR	PI0PCR
P2ODR	P27ODR	P26ODR	P25ODR	P24ODR	P23ODR	P22ODR	P21ODR	P20ODR
PFODR	PF7ODR	PF6ODR	PF5ODR	PF4ODR	PF3ODR	PF2ODR	PF1ODR	PF0ODR
PFCR0	CS7E	CS6E	CS5E	CS4E	CS3E	CS2E	CS1E	CS0E
PFCR1	CS7SA	CS7SB	CS6SA	CS6SB	CS5SA	CS5SB	CS4SA	CS4SB
PFCR2	—	CS2S	BSS	BSE	—	RDWRE	ASOE	—
PFCR4	A23E	A22E	A21E	—	—	—	—	—
PFCR6	—	LHWROE	—	—	TCLKS	—	—	—
PFCR7	DMAS3A	DMAS3B	DMAS2A	DMAS2B	DMAS1A	DMAS1B	DMAS0A	DMAS0B
PFCR9	TPUMS5	TPUMS4	TPUMS3A	TPUMS3B	TPUMS2	TPUMS1	TPUMS0A	TPUMS0B
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

DOFR_0								
DTCR_0								
DBSR_0	BKSZH31	BKSZH30	BKSZH29	BKSZH28	BKSZH27	BKSZH26	BKSZH25	BKSZH24
	BKSZH23	BKSZH22	BKSZH21	BKSZH20	BKSZH19	BKSZH18	BKSZH17	BKSZH16
	BKSZ15	BKSZ14	BKSZ13	BKSZ12	BKSZ11	BKSZ10	BKSZ9	BKSZ8
	BKSZ7	BKSZ6	BKSZ5	BKSZ4	BKSZ3	BKSZ2	BKSZ1	BKSZ0
DMDR_0	DTE	DACKE	TENDE	—	DREQS	NRD	—	—
	ACT	—	—	—	ERRF	—	ESIF	DTIF
	DTSZ1	DTSZ0	MDS1	MDS0	TSEIE	—	ESIE	DTIE
	DTF1	DTF0	DTA	—	—	DMAP2	DMAP1	DMAP0
DACR_0	AMS	DIRS	—	—	—	RPTIE	ARS1	ARS0
	—	—	SAT1	SAT0	—	—	DAT1	DAT0
	SARIE	—	—	SARA4	SARA3	SARA2	SARA1	SARA0
	DARIE	—	—	DARA4	DARA3	DARA2	DARA1	DARA0

DOFR_1								
DTCR_1								
DBSR_1	BKSZH31	BKSZH30	BKSZH29	BKSZH28	BKSZH27	BKSZH26	BKSZH25	BKSZH24
	BKSZH23	BKSZH22	BKSZH21	BKSZH20	BKSZH19	BKSZH18	BKSZH17	BKSZH16
	BKSZ15	BKSZ14	BKSZ13	BKSZ12	BKSZ11	BKSZ10	BKSZ9	BKSZ8
	BKSZ7	BKSZ6	BKSZ5	BKSZ4	BKSZ3	BKSZ2	BKSZ1	BKSZ0
DMDR_1	DTE	DACKE	TENDE	—	DREQS	NRD	—	—
	ACT	—	—	—	—	—	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

DOFR_2								
DTCR_2								
DBSR_2	BKSZH31	BKSZH30	BKSZH29	BKSZH28	BKSZH27	BKSZH26	BKSZH25	BKSZH24
	BKSZH23	BKSZH22	BKSZH21	BKSZH20	BKSZH19	BKSZH18	BKSZH17	BKSZH16
	BKSZ15	BKSZ14	BKSZ13	BKSZ12	BKSZ11	BKSZ10	BKSZ9	BKSZ8
	BKSZ7	BKSZ6	BKSZ5	BKSZ4	BKSZ3	BKSZ2	BKSZ1	BKSZ0
DMDR_2	DTE	DACKE	TENDE	—	DREQS	NRD	—	—
	ACT	—	—	—	—	—	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

DOFR_3								
DTCR_3								
DBSR_3	BKSZH31	BKSZH30	BKSZH29	BKSZH28	BKSZH27	BKSZH26	BKSZH25	BKSZH24
	BKSZH23	BKSZH22	BKSZH21	BKSZH20	BKSZH19	BKSZH18	BKSZH17	BKSZH16
	BKSZ15	BKSZ14	BKSZ13	BKSZ12	BKSZ11	BKSZ10	BKSZ9	BKSZ8
	BKSZ7	BKSZ6	BKSZ5	BKSZ4	BKSZ3	BKSZ2	BKSZ1	BKSZ0
DMDR_3	DTE	DACKE	TENDE	—	DREQS	NRD	—	—
	ACT	—	—	—	—	—	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

	—	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
IPRG	—	IPRG14	IPRG13	IPRG12	—	IPRG10	IPRG9	IPRG8
	—	IPRG6	IPRG5	IPRG4	—	IPRG2	IPRG1	IPRG0
IPRH	—	IPRH14	IPRH13	IPRH12	—	IPRH10	IPRH9	IPRH8
	—	IPRH6	IPRH5	IPRH4	—	IPRH2	IPRH1	IPRH0
IPRI	—	IPRI14	IPRI13	IPRI12	—	IPRI10	IPRI9	IPRI8
	—	IPRI6	IPRI5	IPRI4	—	IPRI2	IPRI1	IPRI0
IPRK	—	IPRK14	IPRK13	IPRK12	—	—	—	—
	—	IPRK6	IPRK5	IPRK4	—	IPRK2	IPRK1	IPRK0
IPRL	—	IPRL14	IPRL13	IPRL12	—	IPRL10	IPRL9	IPRL8
	—	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

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
ENDIANCR	LE7	LE6	LE5	LE4	LE3	LE2	—	—
SRAMCR	BCSEL7	BCSEL6	BCSEL5	BCSEL4	BCSEL3	BCSEL2	BCSEL1	BCSEL0
	—	—	—	—	—	—	—	—
BROMCR	BSRM0	BSTS02	BSTS01	BSTS00	—	—	BSWD01	BSWD00
	BSRM1	BSTS12	BSTS11	BSTS10	—	—	BSWD11	BSWD10
MPXCR	MPXE7	MPXE6	MPXE5	MPXE4	MPXE3	—	—	—
	—	—	—	—	—	—	—	ADDEX
MDCR	—	—	—	—	—	MDS2	MDS1	MDS0
	—	—	—	—	—	—	—	—

	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	MSTPC6	MSTPC5	MSTPC4	MSTPC3	MSTPC2	MSTPC1	MSTPC0
SEMR_2	—	—	—	—	ABCS	ACS2	ACS1	ACS0
SMR_3*[1]	C/$\overline{\text{A}}$ (GM)	CHR (BLK)	PE (PE)	O/$\overline{\text{E}}$ (O/$\overline{\text{E}}$)	STOP (BCP0)	MP (BCP0)	CKS1	CKS0
BRR_3								
SCR_3*[1]	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
TDR_3								
SSR_3*[1]	TDRE	RDRF	ORER	FER (ERS)	PER	TEND	MPB	MPBT
RDR_3								
SCMR_3	—				SDIR	SINV		SMIF
SMR_4*[1]	C/$\overline{\text{A}}$ (GM)	CHR (BLK)	PE (PE)	O/$\overline{\text{E}}$ (O/$\overline{\text{E}}$)	STOP (BCP1)	MP (BCP0)	CKS1	CKS0
BRR_4								
SCR_4*[1]	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
TDR_4								
SSR_4*[1]	TDRE	RDRF	ORER	FER (ERS)	PER	TEND	MPB	MPBT
RDR_4								
SCMR_4	—	—	—	—	SDIR	SINV	—	SMIF

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	—	—	—	—	—	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	—	—	—	—	—	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								

	DTCE7	DTCE6	DTCE5	DTCE4	DTCE3	DTCE2	DTCE1	DTCE0
DTCERC	DTCE15	DTCE14	DTCE13	DTCE12	DTCE11	DTCE10	DTCE9	DTCE8
	DTCE7	DTCE6	DTCE5	DTCE4	DTCE3	DTCE2	DTCE1	DTCE0
DTCERD	DTCE15	DTCE14	DTCE13	DTCE12	DTCE11	DTCE10	DTCE9	DTCE8
	DTCE7	DTCE6	DTCE5	DTCE4	DTCE3	DTCE2	DTCE1	DTCE0
DTCERE	DTCE15	DTCE14	DTCE13	DTCE12	DTCE11	DTCE10	DTCE9	DTCE8
	DTCE7	DTCE6	DTCE5	DTCE4	DTCE3	DTCE2	DTCE1	DTCE0
DTCERF	DTCE15	DTCE14	DTCE13	DTCE12	DTCE11	DTCE10	DTCE9	DTCE8
	DTCE7	DTCE6	DTCE5	DTCE4	DTCE3	DTCE2	DTCE1	DTCE0
DTCERG	DTCE15	DTCE14	DTCE13	DTCE12	DTCE11	DTCE10	DTCE9	DTCE8
	DTCE7	DTCE6	DTCE5	DTCE4	DTCE3	DTCE2	DTCE1	DTCE0
DTCERH	DTCE15	DTCE14	DTCE13	DTCE12	DTCE11	DTCE10	DTCE9	DTCE8
	DTCE7	DTCE6	DTCE5	DTCE4	DTCE3	DTCE2	DTCE1	DTCE0
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

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	PB0DR
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
SMR_2*[1]	C/$\overline{A}$ (GM)	CHR (BLK)	PE (PE)	O/$\overline{E}$ (O/$\overline{E}$)	STOP (BCP1)	MP (BCP0)	CKS1	CKS0
BRR_2								
SCR_2*[1]	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
TDR_2								
SSR_2*[1]	TDRE	RDRF	ORER	FER (ERS)	PER	TEND	MPB	MPBT
RDR_2								
SCMR_2	—	—	—	—	SDIR	SINV	—	SMIF

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

TCCR_1	—	—	—	—	TMRIS	—	ICKS1	ICKS0
TSTR	—	—	CST5	CST4	CST3	CST2	CST1	CST0
TSYR	—	—	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0
TCR_0	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
TMDR_0	—	—	BFB	BFA	—	MD2	MD1	MD0
TIORH_0	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
TIORL_0	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0
TIER_0	TTGE	—	TCIEU	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
TSR_0	TCFD	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
TCNT_0								
TGRA_0								
TGRB_0								
TGRC_0								
TGRD_0								

TGRA_1								
TGRB_1								
TCR_2	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
TMDR_2	—	—	—	—	—	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								
TGRA_2								
TGRB_2								
TCR_3	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
TMDR_3	—	—	BFB	BFA	—	MD2	MD1	MD0
TIORH_3	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
TIORL_3	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0
TIER_3	TTGE	—	TCIEU	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
TSR_3	TCFD	—	—	TCFV	TGFD	TGFC	TGFB	TGFA

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 setting, the NDRH address is H'FFF7C. When different output triggers are sp 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 ou groups 0 and 1 by the PCR setting, the NDRL address is H'FFF7D. When dif output triggers are specified, the NDRL addresses for pulse output groups 0 a H'FFF7F and H'FFF7D, respectively.

PBDDR	Initialized	—	—	—	—	Initialized
PDDDR	Initialized	—	—	—	—	Initialized
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

PZODR	Initialized	—	—	—	—	Initialized
PFODR	Initialized	—	—	—	—	Initialized
PFCR0	Initialized	—	—	—	—	Initialized
PFCR1	Initialized	—	—	—	—	Initialized
PFCR2	Initialized	—	—	—	—	Initialized
PFCR4	Initialized	—	—	—	—	Initialized
PFCR6	Initialized	—	—	—	—	Initialized
PFCR7	Initialized	—	—	—	—	Initialized
PFCR9	Initialized	—	—	—	—	Initialized
PFCRB	Initialized	—	—	—	—	Initialized
PFCRC	Initialized	—	—	—	—	Initialized
SSIER	Initialized	—	—	—	—	Initialized
DSAR_0	Initialized	—	—	—	—	Initialized
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_2	Initialized	—	—	—	—	Initialized	D
DDAR_2	Initialized	—	—	—	—	Initialized	
DOFR_2	Initialized	—	—	—	—	Initialized	
DTCR_2	Initialized	—	—	—	—	Initialized	
DBSR_2	Initialized	—	—	—	—	Initialized	
DMDR_2	Initialized	—	—	—	—	Initialized	
DACR_2	Initialized	—	—	—	—	Initialized	
DSAR_3	Initialized	—	—	—	—	Initialized	D
DDAR_3	Initialized	—	—	—	—	Initialized	
DOFR_3	Initialized	—	—	—	—	Initialized	
DTCR_3	Initialized	—	—	—	—	Initialized	
DBSR_3	Initialized	—	—	—	—	Initialized	
DMDR_3	Initialized	—	—	—	—	Initialized	
DACR_3	Initialized	—	—	—	—	Initialized	
DMRSR_0	Initialized	—	—	—	—	Initialized	D
DMRSR_1	Initialized	—	—	—	—	Initialized	D
DMRSR_2	Initialized	—	—	—	—	Initialized	D
DMRSR_3	Initialized	—	—	—	—	Initialized	D

IPRJ	Initialized	—	—	—	—	Initialized
IPRK	Initialized	—	—	—	—	Initialized
IPRL	Initialized	—	—	—	—	Initialized
ISCRH	Initialized	—	—	—	—	Initialized
ISCRL	Initialized	—	—	—	—	Initialized
DTCVBR	Initialized	—	—	—	—	Initialized
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
SRAMCR	Initialized	—	—	—	—	Initialized
BROMCR	Initialized	—	—	—	—	Initialized
MPXCR	Initialized	—	—	—	—	Initialized

SEMR_2	Initialized	—	—	—	—	Initialized	S
SMR_3	Initialized	—	—	—	—	Initialized	S
BRR_3	Initialized	—	—	—	—	Initialized	
SCR_3	Initialized	—	—	—	—	Initialized	
TDR_3	Initialized	—	Initialized	Initialized	Initialized	Initialized	
SSR_3	Initialized	—	Initialized	Initialized	Initialized	Initialized	
RDR_3	Initialized	—	Initialized	Initialized	Initialized	Initialized	
SCMR_3	Initialized	—	—	—	—	Initialized	
SMR_4	Initialized	—	—	—	—	Initialized	S
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	
TCR_2	Initialized	—	—	—	—	Initialized	T
TCR_3	Initialized	—	—	—	—	Initialized	T
TCSR_2	Initialized	—	—	—	—	Initialized	T
TCSR_3	Initialized	—	—	—	—	Initialized	T
TCORA_2	Initialized	—	—	—	—	Initialized	T
TCORA_3	Initialized	—	—	—	—	Initialized	T
TCORB_2	Initialized	—	—	—	—	Initialized	T
TCORB_3	Initialized	—	—	—	—	Initialized	T

TIER_4	Initialized	—	—	—	—	Initialized
TSR_4	Initialized	—	—	—	—	Initialized
TCNT_4	Initialized	—	—	—	—	Initialized
TGRA_4	Initialized	—	—	—	—	Initialized
TGRB_4	Initialized	—	—	—	—	Initialized
TCR_5	Initialized	—	—	—	—	Initialized
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
DTCERB	Initialized	—	—	—	—	Initialized
DTCERC	Initialized	—	—	—	—	Initialized
DTCERD	Initialized	—	—	—	—	Initialized
DTCERE	Initialized	—	—	—	—	Initialized
DTCERF	Initialized	—	—	—	—	Initialized
DTCERG	Initialized	—	—	—	—	Initialized
DTCERH	Initialized	—	—	—	—	Initialized
DTCCR	Initialized	—	—	—	—	Initialized
INTCR	Initialized	—	—	—	—	Initialized

PORT6	—	—	—	—	—	—	
PORTA	—	—	—	—	—	—	
PORTB	—	—	—	—	—	—	
PORTD	—	—	—	—	—	—	
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	S
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	

PODRH	Initialized	—	—	—	—	Initialized
PODRL	Initialized	—	—	—	—	Initialized
NDRH	Initialized	—	—	—	—	Initialized
NDRL	Initialized	—	—	—	—	Initialized
SMR_0	Initialized	—	—	—	—	Initialized
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
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

ADDRH	Initialized	—	—	—	—	Initialized	
ADCSR	Initialized	—	—	—	—	Initialized	
ADCR	Initialized	—	—	—	—	Initialized	
TCSR	Initialized	—	—	—	—	Initialized	W
TCNT	Initialized	—	—	—	—	Initialized	
RSTCSR	Initialized	—	—	—	—	Initialized	
TCR_0	Initialized	—	—	—	—	Initialized	T
TCR_1	Initialized	—	—	—	—	Initialized	T
TCSR_0	Initialized	—	—	—	—	Initialized	T
TCSR_1	Initialized	—	—	—	—	Initialized	T
TCORA_0	Initialized	—	—	—	—	Initialized	T
TCORA_1	Initialized	—	—	—	—	Initialized	T
TCORB_0	Initialized	—	—	—	—	Initialized	T
TCORB_1	Initialized	—	—	—	—	Initialized	T
TCNT_0	Initialized	—	—	—	—	Initialized	T
TCNT_1	Initialized	—	—	—	—	Initialized	T
TCCR_0	Initialized	—	—	—	—	Initialized	T
TCCR_1	Initialized	—	—	—	—	Initialized	T
TSTR	Initialized	—	—	—	—	Initialized	T
TSYR	Initialized	—	—	—	—	Initialized	

TGRA_0	Initialized	—	—	—	—	Initialized
TGRB_0	Initialized	—	—	—	—	Initialized
TGRC_0	Initialized	—	—	—	—	Initialized
TGRD_0	Initialized	—	—	—	—	Initialized
TCR_1	Initialized	—	—	—	—	Initialized
TMDR_1	Initialized	—	—	—	—	Initialized
TIOR_1	Initialized	—	—	—	—	Initialized
TIER_1	Initialized	—	—	—	—	Initialized
TSR_1	Initialized	—	—	—	—	Initialized
TCNT_1	Initialized	—	—	—	—	Initialized
TGRA_1	Initialized	—	—	—	—	Initialized
TGRB_1	Initialized	—	—	—	—	Initialized
TCR_2	Initialized	—	—	—	—	Initialized
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

TGRA_3	Initialized	—	—	—	—	Initialized
TGRB_3	Initialized	—	—	—	—	Initialized
TGRC_3	Initialized	—	—	—	—	Initialized
TGRD_3	Initialized	—	—	—	—	Initialized

Input voltage (except port 5)	V_{in}	–0.3 to V_{CC} +0.3
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

Schmitt trigger input voltage	$\overline{\text{IRQ}}$ 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, $\overline{\text{RES}}$, $\overline{\text{STBY}}$, EMLE, NMI	V_{IH}	$V_{CC} \times 0.9$	—	$V_{CC} + 0.3$	V	
	EXTAL 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, $\overline{\text{RES}}$, $\overline{\text{STBY}}$, 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} = 1
	Port 3		—	—	1.0		I_{OL} = 1
Input leakage current	$\overline{\text{RES}}$	$\|I_{in}\|$	—	—	10.0	μA	V_{in} = 0 V_{CC} –
	MD, $\overline{\text{STBY}}$, EMLE, NMI		—	—	1.0		
	Port 5		—	—	1.0		V_{in} = 0 AV_{CC} –

Input pull-up MOS current	Ports D to F, H, I	$-I_p$	10	—	300	µA	V_{CC} = V_{in} = 0
Input capacitance	All input pins	C_{in}	—	—	15	pF	V_{in} = 0 f = 1 M T_a = 2
Current consumption *3	Normal operation	I_{CC}*5	—	30 (3.3 V)	45	mA	f = 35
	Sleep mode		—	25 (3.3 V)	37		
	Standby mode*4		—	0.1	0.5		$T_a \leq 5$
			—	—	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		—	0.1	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}.

2. The case where port 5 is used as $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ7}}$.

3. Current consumption values are for V_{IH}min = V_{CC} – 0.5 V and V_{IL}max = 0.5 V w output pins unloaded and all input pull-up MOSs in the off state.

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}.

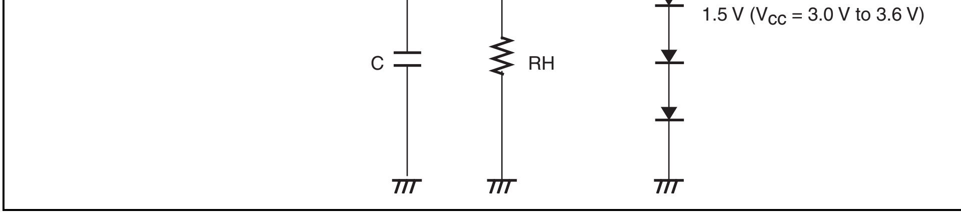


Figure 21.1 Output Load Circuit (1)

(1) Clock Timing

Table 21.4 Clock Timing

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, Iϕ = 8 MHz to 35 MHz, Bϕ = 8 MHz to 35 MHz, 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	Symbol	Min.	Max.	Unit.	Test Co
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	

width	t_{EXH}			
External clock rising time	t_{EXr}	—	5	ns
External clock falling time	t_{EXf}	—	5	ns

(2) Control Signal Timing

Table 21.5 Control Signal Timing

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC},
V_{SS} = AV_{SS} = 0 V, Iϕ = 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 Con
$\overline{\text{RES}}$ setup time	t_{RESS}	200	—	ns	Figure 21.
$\overline{\text{RES}}$ pulse width	t_{RESW}	20	—	t_{cyc}	
NMI setup time	t_{NMIS}	150	—	ns	Figure 21.
NMI hold time	t_{NMIH}	10	—	ns	
NMI pulse width (after leaving software standby mode)	t_{NMIW}	200	—	ns	
$\overline{\text{IRQ}}$ setup time	t_{IRQS}	150	—	ns	
$\overline{\text{IRQ}}$ hold time	t_{IRQH}	10	—	ns	
$\overline{\text{IRQ}}$ pulse width (after leaving software standby mode)	t_{IRQW}	200	—	ns	

Address setup time 1	t_{AS1}	$0.5 \times t_{cyc} - 8$	—	ns	21.20
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	
$\overline{RD}$ delay time 1	t_{RSD1}	—	15	ns	
$\overline{RD}$ delay time 2	t_{RSD2}	—	15	ns	
Read data setup time 1	t_{RDS1}	15	—	ns	
Read data setup time 2	t_{RDS2}	15	—	ns	
Read data hold time 1	t_{RDH1}	0	—	ns	
Read data hold time 2	t_{RDH2}	0	—	ns	
Read data access time 2	t_{AC2}	—	$1.5 \times t_{cyc} - 20$	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	

(from address) 5	$_{AA5}$		$_{cyc}$	
$\overline{WR}$ delay time 1	t_{WRD1}	—	15	ns
$\overline{WR}$ delay time 2	t_{WRD2}	—	15	ns
$\overline{WR}$ pulse width 1	t_{WSW1}	$1.0 \times t_{cyc} - 13$	—	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 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 21.18
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	
$\overline{\text{WAIT}}$ setup time	t_{WTS}	15	—	ns	Figure 21.18
$\overline{\text{WAIT}}$ hold time	t_{WTH}	5.0	—	ns	
$\overline{\text{BREQ}}$ setup time	t_{BREQS}	20	—	ns	Figure
$\overline{\text{BACK}}$ delay time	t_{BACD}	—	15	ns	
Bus floating time	t_{BZD}	—	30	ns	
$\overline{\text{BREQO}}$ delay time	t_{BRQOD}	—	15	ns	Figure
$\overline{\text{BS}}$ delay time	t_{BSD}	1.0	15	ns	Figure 21.9, 21.14
RD/$\overline{\text{WR}}$ delay time	t_{RWD}	—	15	ns	

	t_{DRQS}				
$\overline{\text{DREQ}}$ hold time	t_{DRQH}	5	—	ns	
$\overline{\text{TEND}}$ delay time	t_{TED}	—	15	ns	Figure
$\overline{\text{DACK}}$ delay time 1	t_{DACD1}	—	15	ns	Figures 21.24
$\overline{\text{DACK}}$ delay time 2	t_{DACD2}	—	15	ns	

(5) On-Chip Peripheral Modules

Table 21.8 Timing of On-Chip Peripheral Modules

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC},
V_{SS} = AV_{SS} = 0 V, 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			Symbol	Min.	Max.	Unit	Test Co
I/O ports	Output data delay time		t_{PWD}	—	40	ns	Figure 2
	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 2
	Timer input setup time		t_{TICS}	25	—	ns	
	Timer clock input setup time		t_{TCKS}	25	—	ns	Figure 2
	Timer clock pulse width	Single-edge setting	t_{TCKWH}	1.5	—	t_{cyc}	
		Both-edge setting	t_{TCKWL}	2.5	—	t_{cyc}	

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}	
	Transmit data delay time		t_{TXD}	—	40	ns	Figure
	Receive data setup time (clocked synchronous)		t_{RXS}	40	—	ns	
	Receive data hold time (clocked synchronous)		t_{RXH}	40	—	ns	
A/D converter	Trigger input setup time		t_{TRGS}	30	—	ns	Figure

Conversion time	7.4	—	—	μs
Analog input capacitance	—	—	20	pF
Permissible signal source impedance	—	—	10	kΩ
Nonlinearity error	—	—	±7.5	LSB
Offset error	—	—	±7.5	LSB
Full-scale error	—	—	±7.5	LSB
Quantization error	—	±0.5	—	LSB
Absolute accuracy	—	—	±8.0	LSB

21.1.5 D/A Conversion Characteristics

Table 21.10 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 Conditi
Resolution	8	8	8	Bit	
Conversion time	—	—	10	μs	20-pF capaci
Absolute accuracy	—	±2.0	±3.0	LSB	2-MΩ resistiv
	—	—	±2.0	LSB	4-MΩ resistiv

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

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, $\overline{\text{RES}}$, $\overline{\text{STBY}}$, EMLE, NMI	V_{IH}	$V_{CC} \times 0.9$	—	$V_{CC} + 0.3$	V	
	EXTAL Other input pins		$V_{CC} \times 0.7$	—	$V_{CC} + 0.3$		
	Port 5		$AV_{CC} \times 0.7$	—	$AV_{CC} + 0.3$		
Input low voltage (except Schmitt trigger input pin)	MD, $\overline{\text{RES}}$, $\overline{\text{STBY}}$, EMLE	V_{IL}	–0.3	—	$V_{CC} \times 0.1$	V	
	EXTAL, NMI		–0.3	—	$V_{CC} \times 0.2$		
	Other pins		–0.3	—	$V_{CC} \times 0.2$		
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} = 1
	Port 3		—	—	1.0		I_{OL} = 1
Input leakage current	$\overline{\text{RES}}$	$\lvert I_{in} \rvert$	—	—	10.0	μA	V_{in} = 0 V_{CC} –
	MD, $\overline{\text{STBY}}$, EMLE, NMI		—	—	1.0		
	Port 5		—	—	1.0		V_{in} = 0 – 0.5

(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 = 1 T_a =
Current consumption *3	Normal operation	I_{CC}*5	—	45 (3.3 V)	65	mA	f = 5
	Sleep mode		—	35 (3.3 V)	52		
	Standby mode*4		—	0.1	0.5		T_a ≤
			—	—	3.0		50°C
	All-module-clock-stop mode*6		—	22	36		
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		—	0.1	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}.

2. The case where port 5 is used as $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ7}}$.

3. Current consumption values are for V_{IH}min = V_{CC} – 0.5 V and V_{IL}max = 0.5 V w output pins unloaded and all input pull-up MOSs in the off state.

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}.

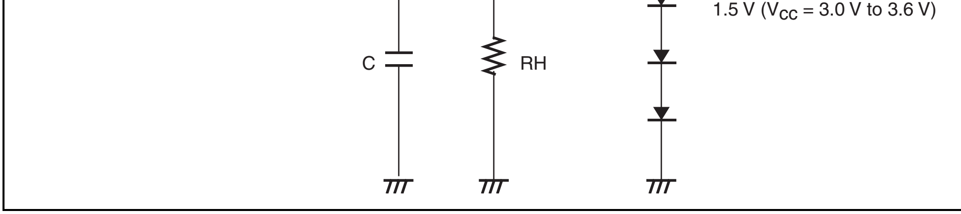


Figure 21.1 Output Load Circuit (2)

(1) Clock Timing

Table 21.14 Clock Timing

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC},
V_{SS} = AV_{SS} = 0 V, Iϕ = 8 MHz to 50 MHz, Bϕ = 8 MHz to 50 MHz,
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	Symbol	Min.	Max.	Unit.	Test Co
Clock cycle time	t_{cyc}	20.0	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	

width	t_{EXH}			
External clock rising time	t_{EXr}	—	5	ns
External clock falling time	t_{EXf}	—	5	ns

(2) Control Signal Timing

Table 21.15 Control Signal Timing

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC},
V_{SS} = AV_{SS} = 0 V, Iϕ = 8 MHz to 50 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 Con
$\overline{\text{RES}}$ setup time	t_{RESS}	200	—	ns	Figure 21.
$\overline{\text{RES}}$ pulse width	t_{RESW}	20	—	t_{cyc}	
NMI setup time	t_{NMIS}	150	—	ns	Figure 21.
NMI hold time	t_{NMIH}	10	—	ns	
NMI pulse width (after leaving software standby mode)	t_{NMIW}	200	—	ns	
$\overline{\text{IRQ}}$ setup time	t_{IRQS}	150	—	ns	
$\overline{\text{IRQ}}$ hold time	t_{IRQH}	10	—	ns	
$\overline{\text{IRQ}}$ pulse width (after leaving software standby mode)	t_{IRQW}	200	—	ns	

Address setup time 1	t_{AS1}	$0.5 \times t_{CYC} - 8$	—	ns	21.20
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	
$\overline{\text{CS}}$ delay time 1	t_{CSD1}	—	15	ns	
$\overline{\text{AS}}$ delay time	t_{ASD}	—	15	ns	
$\overline{\text{RD}}$ delay time 1	t_{RSD1}	—	15	ns	
$\overline{\text{RD}}$ delay time 2	t_{RSD2}	—	15	ns	
Read data setup time 1	t_{RDS1}	15	—	ns	
Read data setup time 2	t_{RDS2}	15	—	ns	
Read data hold time 1	t_{RDH1}	0	—	ns	
Read data hold time 2	t_{RDH2}	0	—	ns	
Read data access time 2	t_{AC2}	—	$1.5 \times t_{CYC} - 20$	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) 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
$\overline{\text{WR}}$ delay time 1	t_{WRD1}	—	15	ns
$\overline{\text{WR}}$ delay time 2	t_{WRD2}	—	15	ns
$\overline{\text{WR}}$ pulse width 1	t_{WSW1}	$1.0 \times t_{CYC} - 13$	—	ns
$\overline{\text{WR}}$ 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

Multiplexed address setup time 1	t_{MAS1}	0.5 × t_{CYC} – 15	—	ns	
Multiplexed address setup time 2	t_{MAS2}	1.5 × t_{CYC} – 15	—	ns	
Address hold delay time	t_{AHD}	—	15	ns	
Address hold pulse width 1	t_{AHW1}	1.0 × t_{CYC} – 15	—	ns	
Address hold pulse width 2	t_{AHW2}	2.0 × t_{CYC} – 15	—	ns	
$\overline{\text{WAIT}}$ setup time	t_{WTS}	15	—	ns	Figure 21.18
$\overline{\text{WAIT}}$ hold time	t_{WTH}	5.0	—	ns	
$\overline{\text{BREQ}}$ setup time	t_{BREQS}	20	—	ns	Figure
$\overline{\text{BACK}}$ delay time	t_{BACD}	—	15	ns	
Bus floating time	t_{BZD}	—	30	ns	
$\overline{\text{BREQO}}$ delay time	t_{BRQOD}	—	15	ns	Figure
$\overline{\text{BS}}$ delay time	t_{BSD}	1.0	15	ns	Figure 21.9, 21.14
RD/$\overline{\text{WR}}$ delay time	t_{RWD}	—	15	ns	

$\overline{\text{DREQ}}$ hold time	t_{DRQH}	5	—	ns	
$\overline{\text{TEND}}$ delay time	t_{TED}	—	15	ns	Figure
$\overline{\text{DACK}}$ delay time 1	t_{DACD1}	—	15	ns	Figures 21.24
$\overline{\text{DACK}}$ delay time 2	t_{DACD2}	—	15	ns	

(5) On-Chip Peripheral Modules

Table 21.18 Timing of On-Chip Peripheral Modules

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC},
V_{SS} = AV_{SS} = 0 V, 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			Symbol	Min.	Max.	Unit	Test Co
I/O ports	Output data delay time		t_{PWD}	—	40	ns	Figure 2
	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 2
	Timer input setup time		t_{TICS}	25	—	ns	
	Timer clock input setup time		t_{TCKS}	25	—	ns	Figure 2
	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 2

	cycle	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}	
	Transmit data delay time		t_{TXD}	—	40	ns	Figure
	Receive data setup time (clocked synchronous)		t_{RXS}	40	—	ns	
	Receive data hold time (clocked synchronous)		t_{RXH}	40	—	ns	
A/D converter	Trigger input setup time		t_{TRGS}	30	—	ns	Figure

Conversion time	7.4	—	—	μs
Analog input capacitance	—	—	20	pF
Permissible signal source impedance	—	—	10	kΩ
Nonlinearity error	—	—	±7.5	LSB
Offset error	—	—	±7.5	LSB
Full-scale error	—	—	±7.5	LSB
Quantization error	—	±0.5	—	LSB
Absolute accuracy	—	—	±8.0	LSB

21.2.5 D/A Conversion Characteristics

Table 21.20 D/A Conversion Characteristics

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC},
V_{SS} = AV_{SS} = 0 V, 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 Conditi
Resolution	8	8	8	Bit	
Conversion time	—	—	10	μs	20-pF capaci
Absolute accuracy	—	±2.0	±3.0	LSB	2-MΩ resistiv
	—	—	±2.0	LSB	4-MΩ resistiv

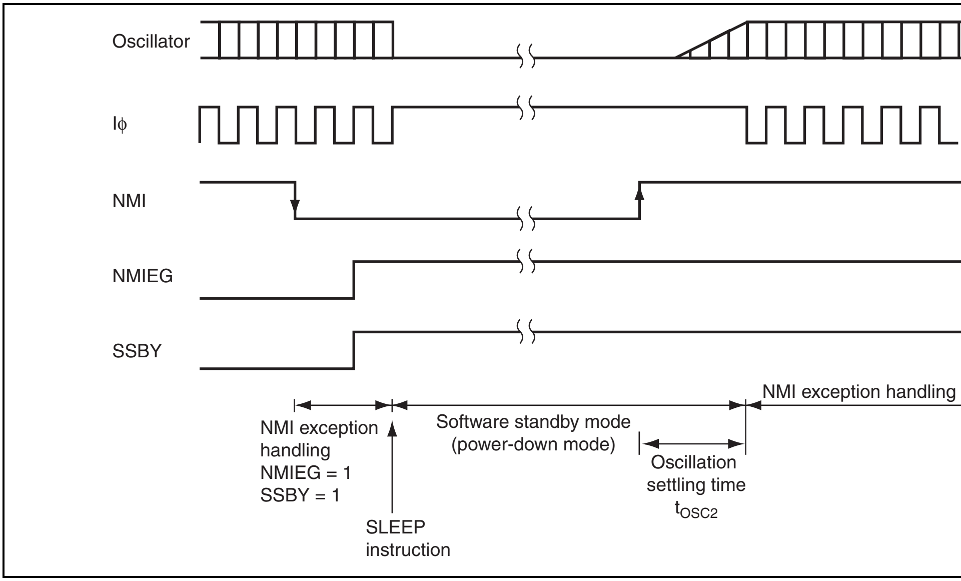


Figure 21.3 Oscillation Settling Timing after Software Standby Mode

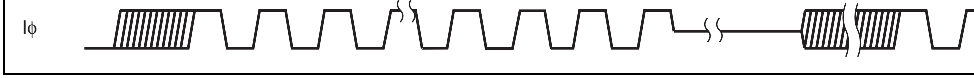


Figure 21.4   Oscillation Settling Timing

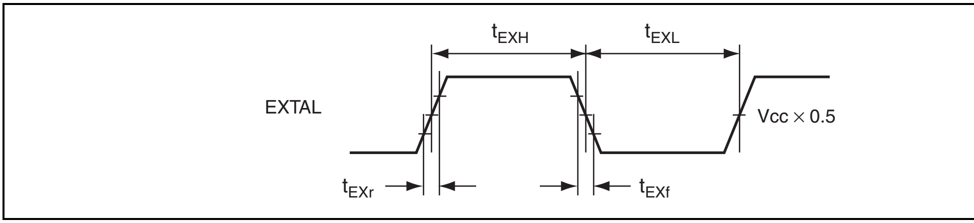


Figure 21.5   External Input Clock Timing

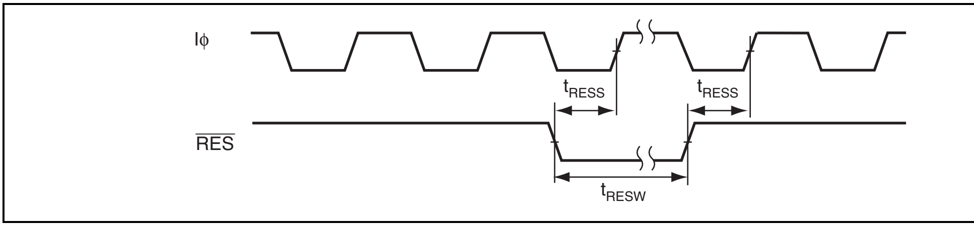
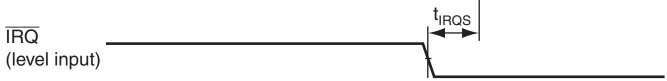


Figure 21.6   Reset Input Timing



Note: * SSIER must be set to cancel software standby mode.

Figure 21.7 Interrupt Input Timing

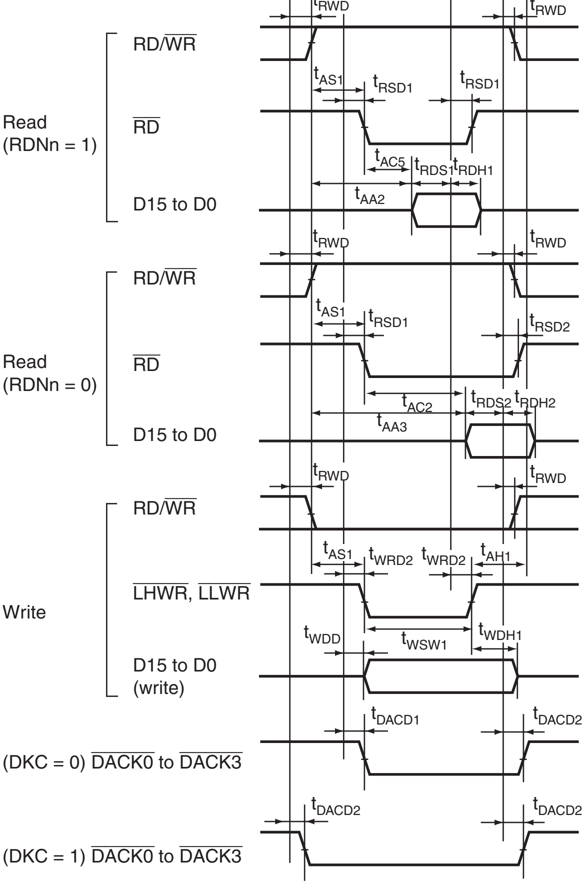


Figure 21.8 Basic Bus Timing: 2-State Access

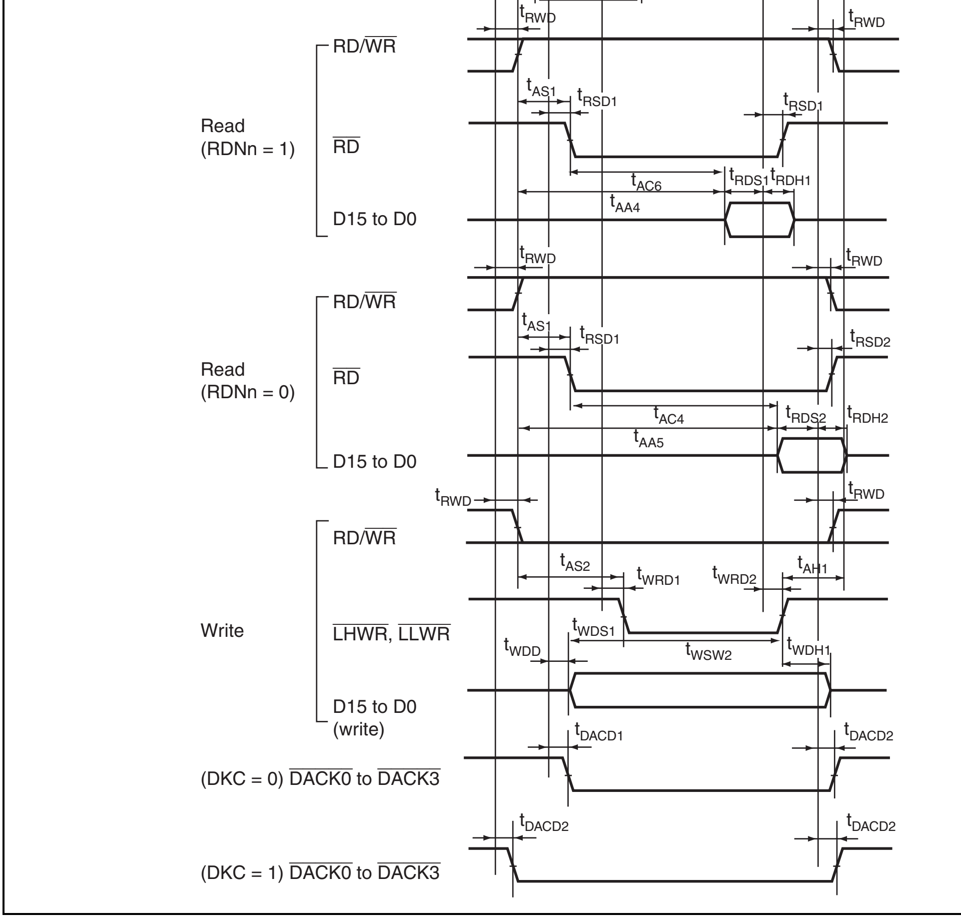


Figure 21.9 Basic Bus Timing: 3-State Access

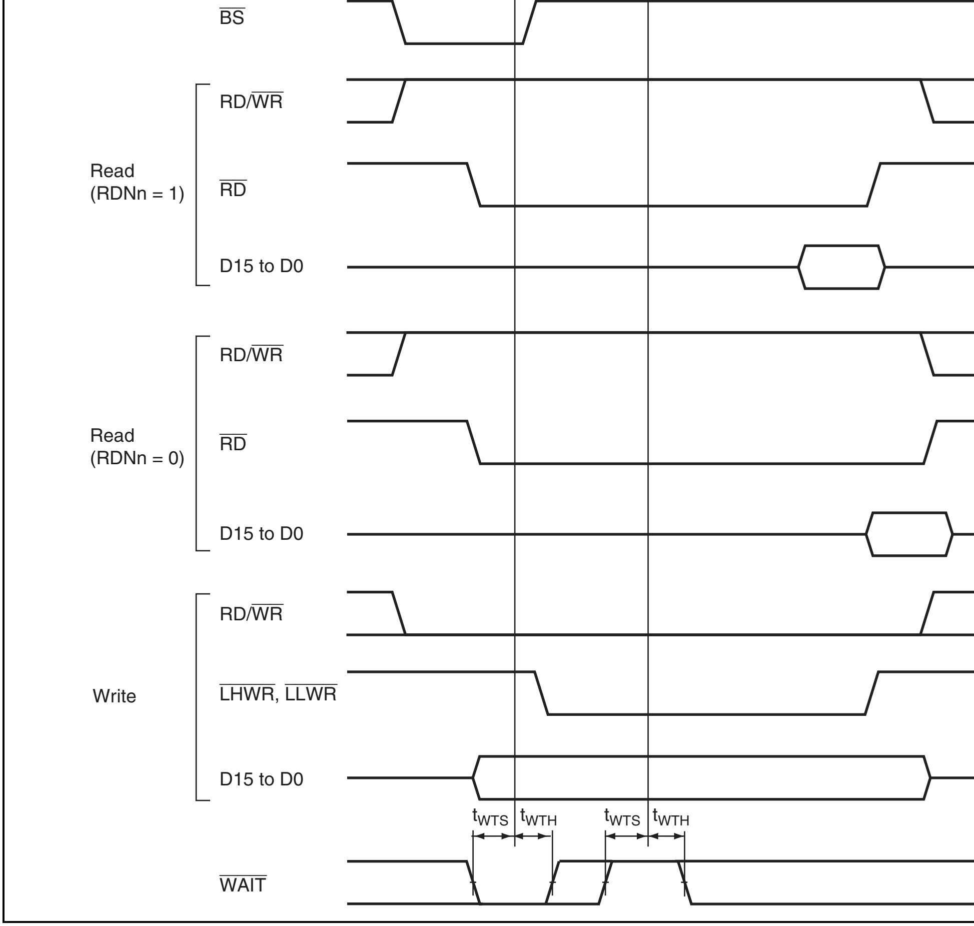


Figure 21.10 Basic Bus Timing: Three-State Access, One Wait

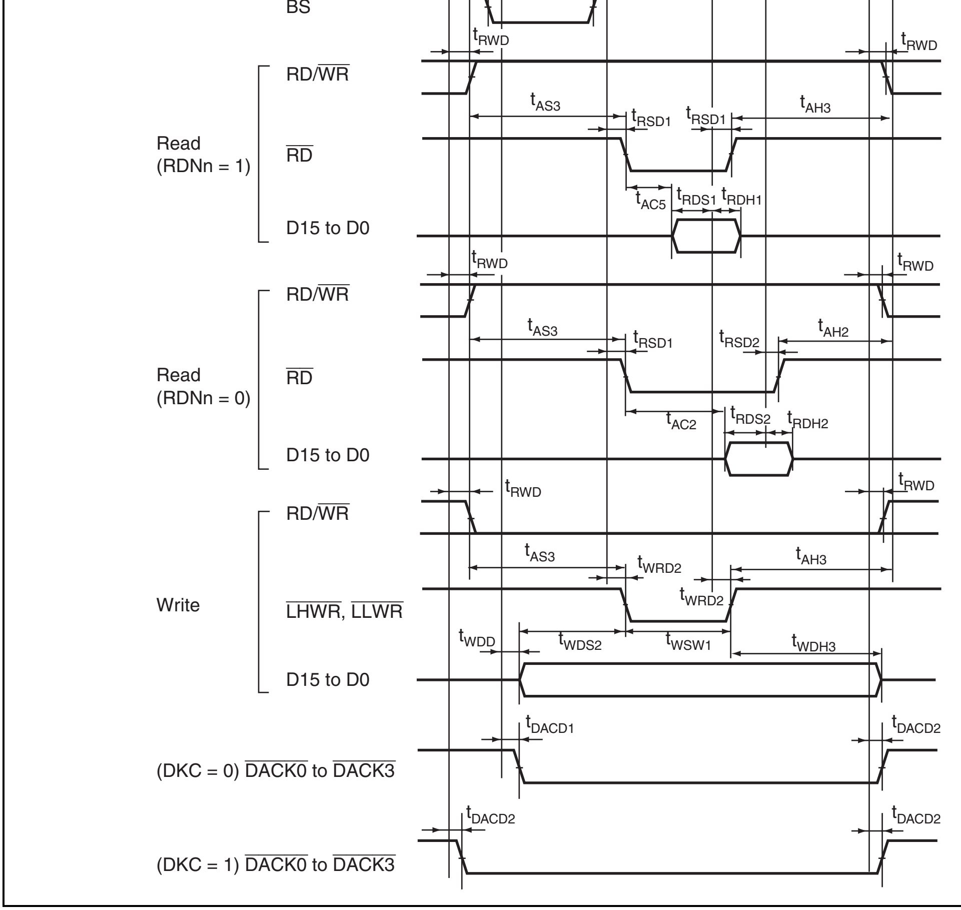


Figure 21.11 Basic Bus Timing: 2-State Access ($\overline{\text{CS}}$ Assertion Period Extend

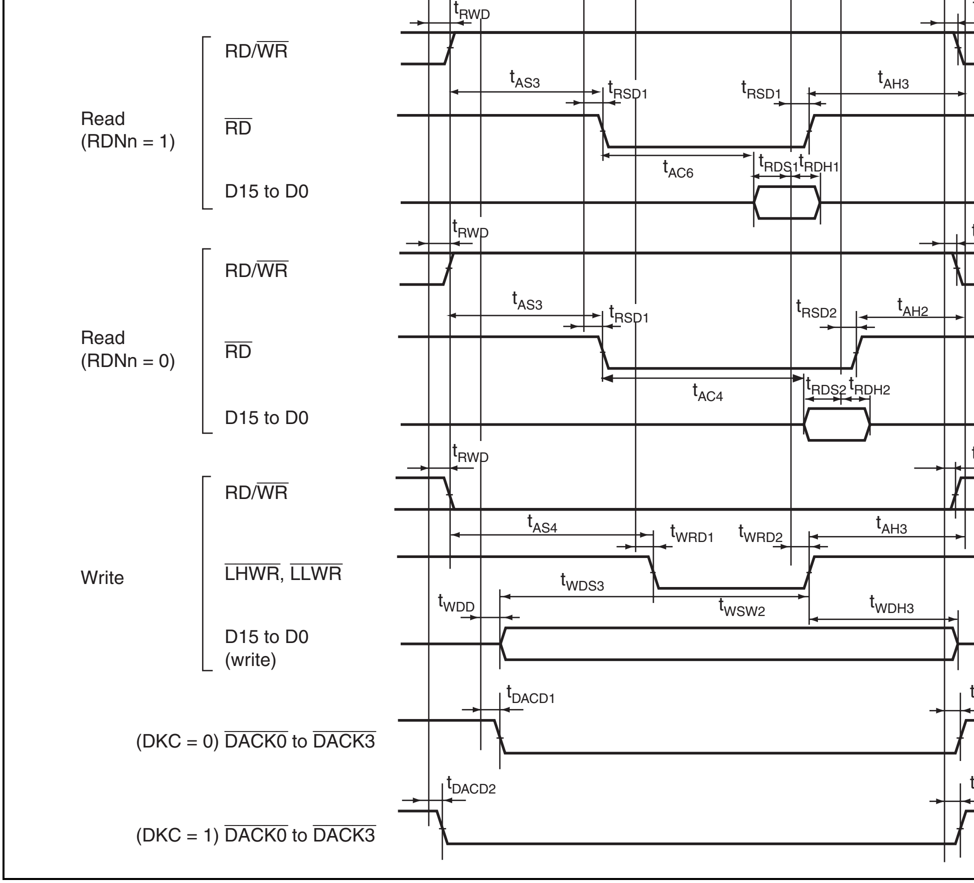


Figure 21.12 Basic Bus Timing: 3-State Access ($\overline{\text{CS}}$ Assertion Period Extende

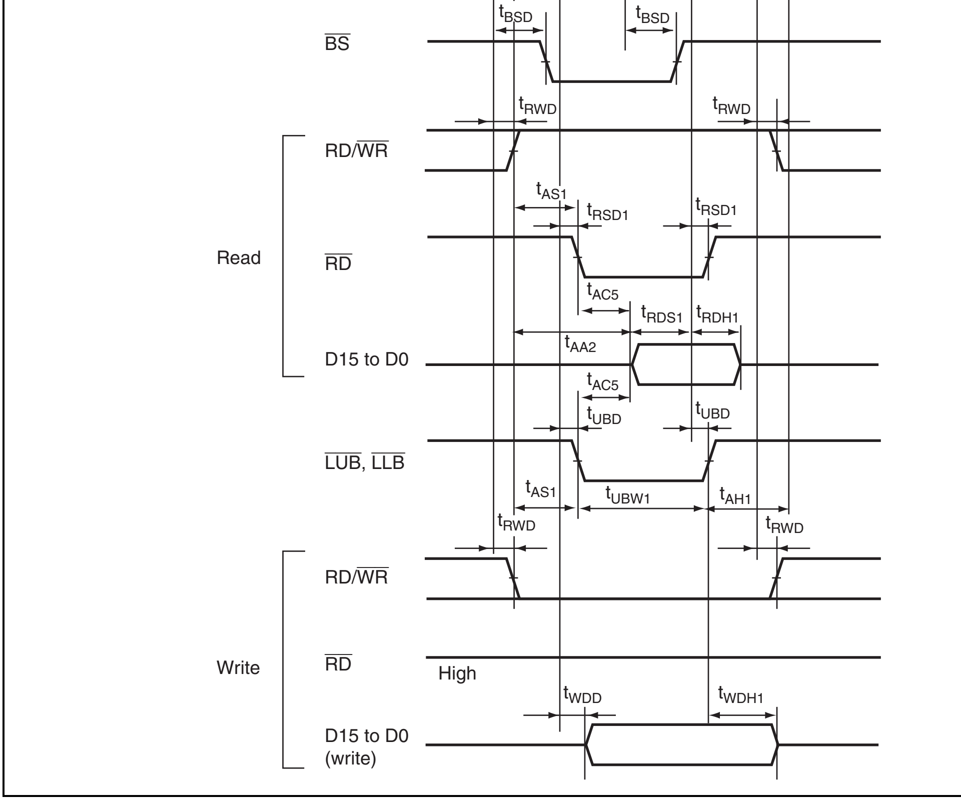


Figure 21.13 Byte Control SRAM: 2-State Read/Write Access

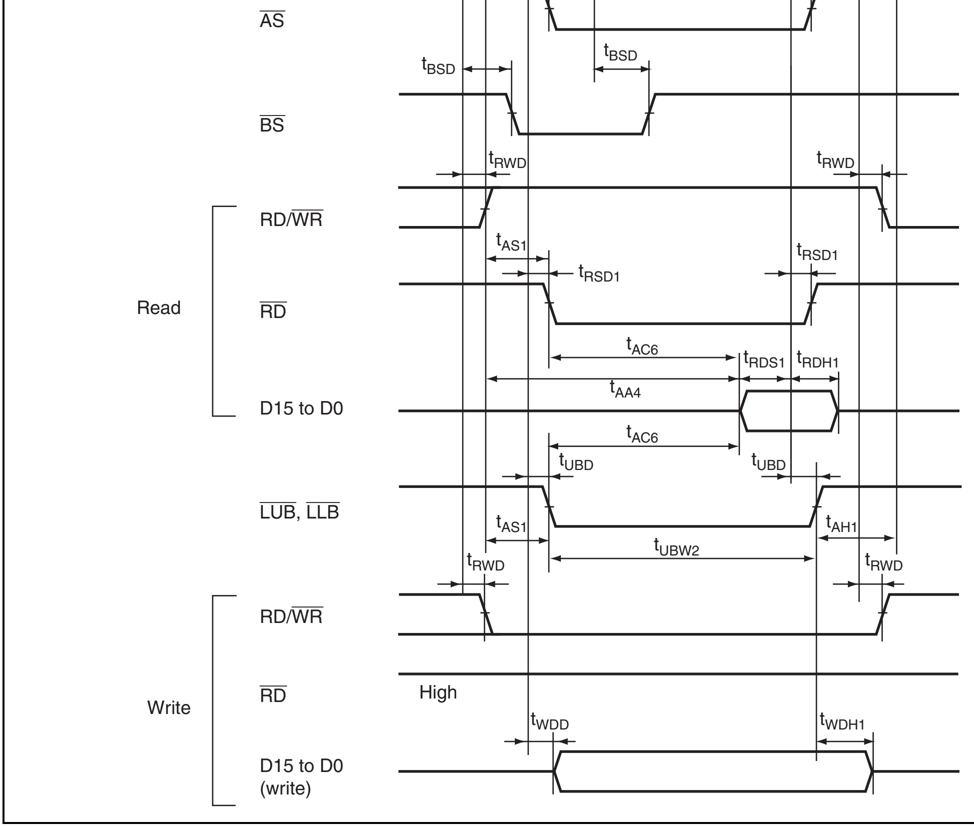


Figure 21.14 Byte Control SRAM: 3-State Read/Write Access

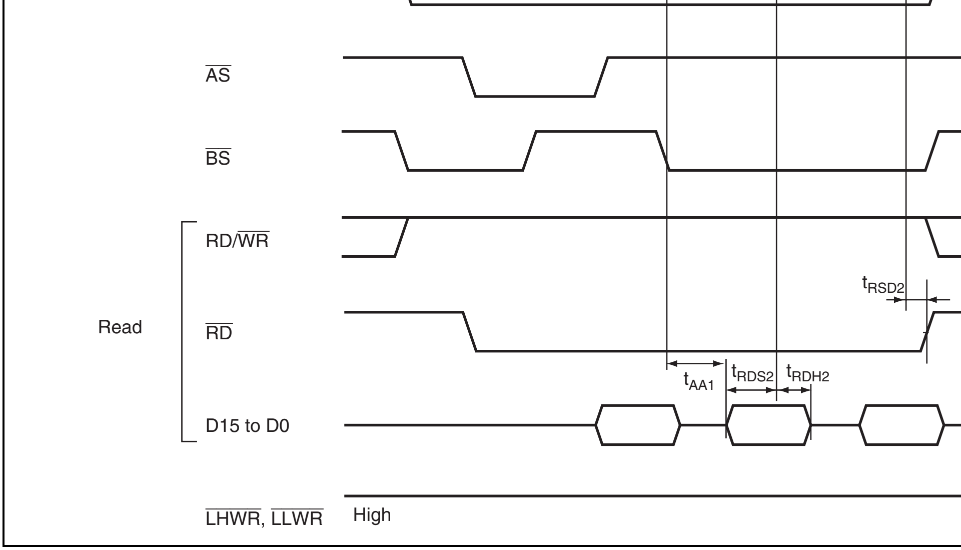


Figure 21.15 Burst ROM Access Timing: 1-State Burst Access

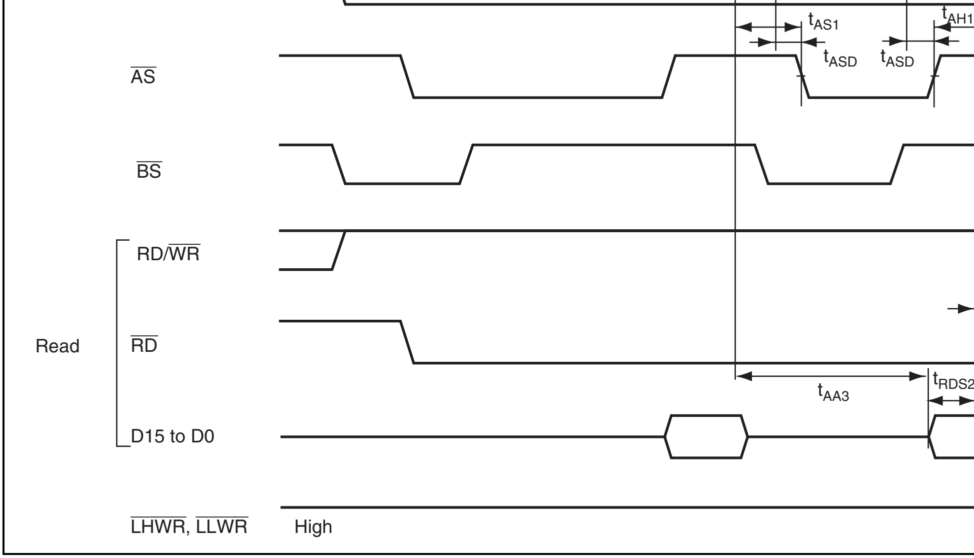


Figure 21.16 Burst ROM Access Timing: 2-State Burst Access

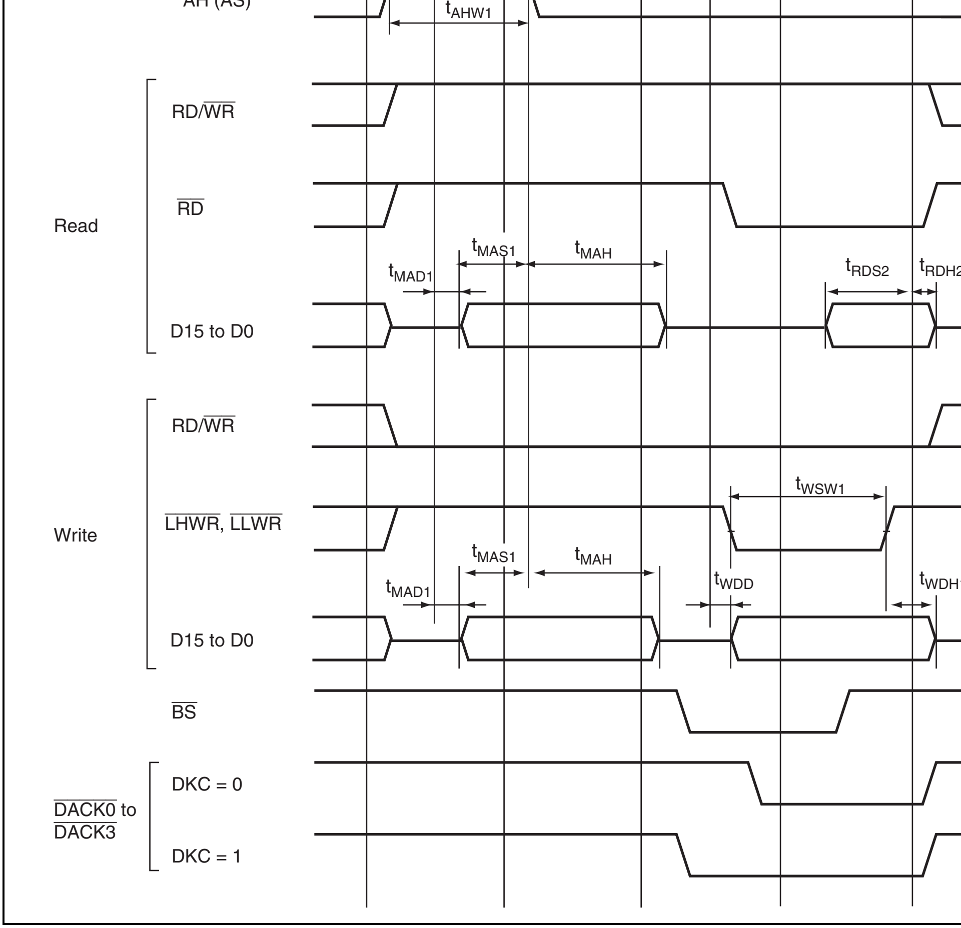


Figure 21.17 Address/Data Multiplexed Access Timing (No Wait) (Basic, 4-State

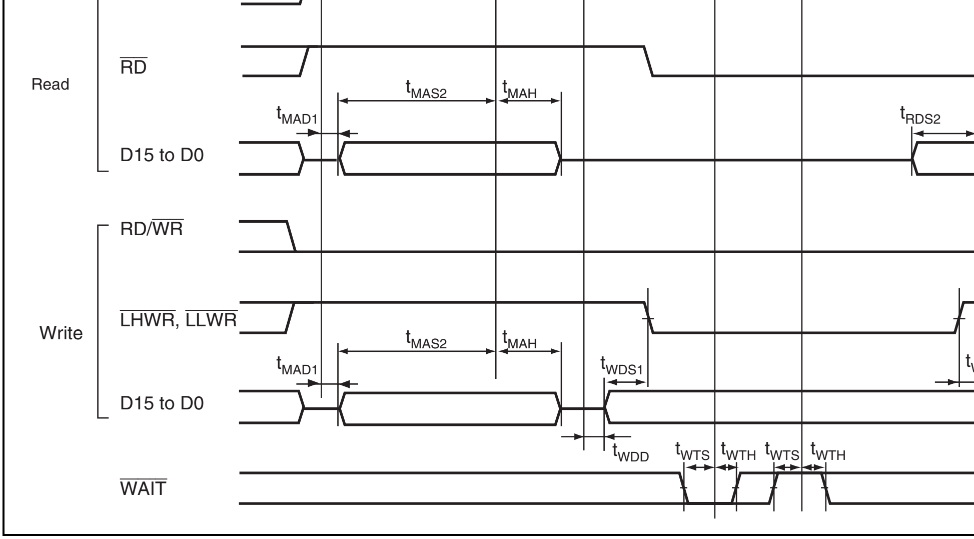


Figure 21.18 Address/Data Multiplexed Access Timing (Wait Control) (Address Cycle Program Wait × 1 + Data Cycle Program Wait × 1 + Data Cycle Pin Wait × 1)

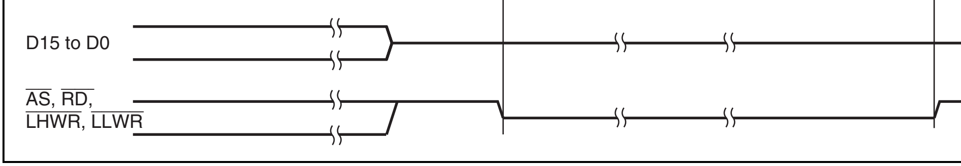


Figure 21.19 External Bus Release Timing

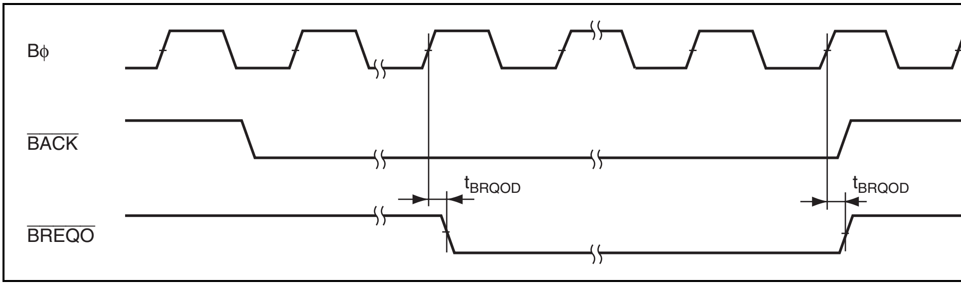


Figure 21.20 External Bus Request Output Timing

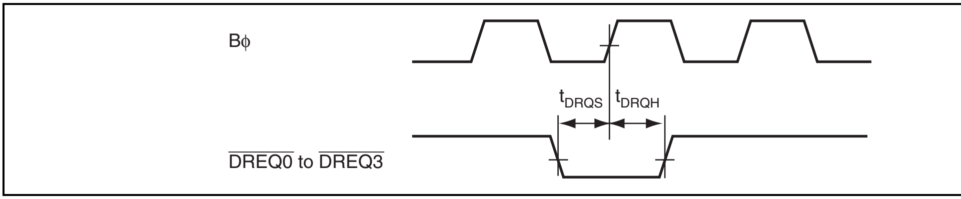


Figure 21.21 DMAC, $\overline{\text{DREQ}}$ Input Timing

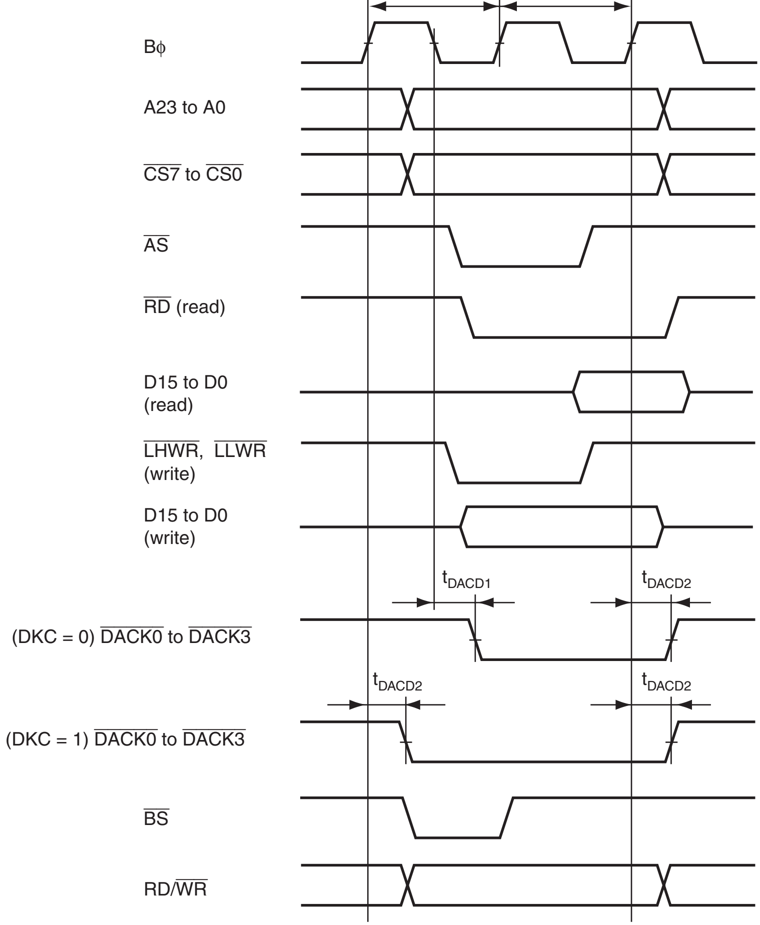


Figure 21.23 DMAC Single Address Transfer Timing: 2-State Access

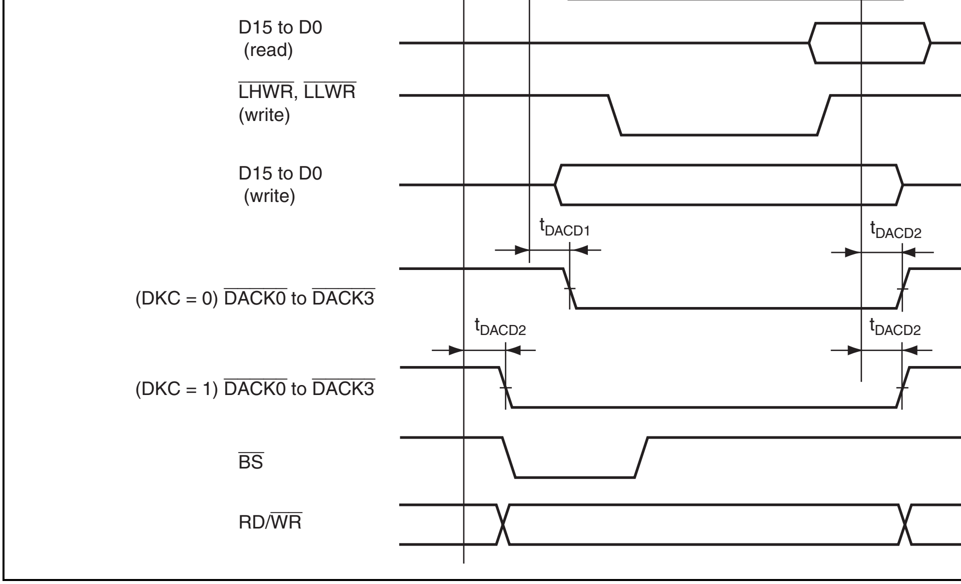

D15 to D0
(read)
$\overline{LHWR}$, $\overline{LLWR}$
(write)
D15 to D0
(write)
t_{DACD1}
t_{DACD2}
(DKC = 0) $\overline{DACK0}$ to $\overline{DACK3}$
t_{DACD2}
t_{DACD2}
(DKC = 1) $\overline{DACK0}$ to $\overline{DACK3}$
$\overline{BS}$
RD/$\overline{WR}$

Figure 21.24 DMAC Single Address Transfer Timing: 3-State Access

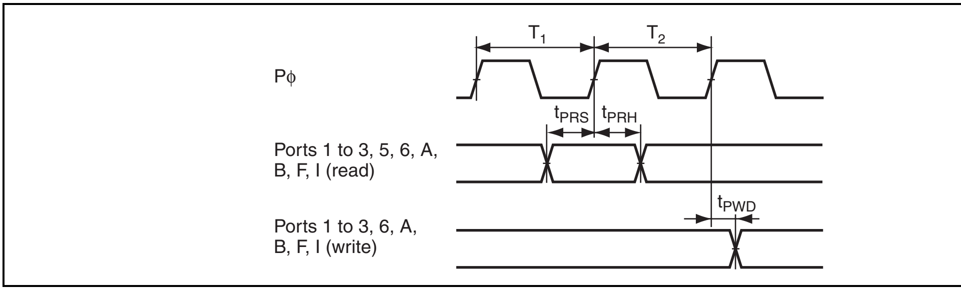


Figure 21.25 I/O Port Input/Output Timing

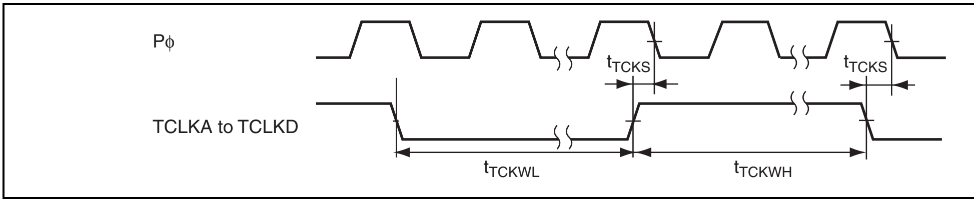


Figure 21.27 TPU Clock Input Timing

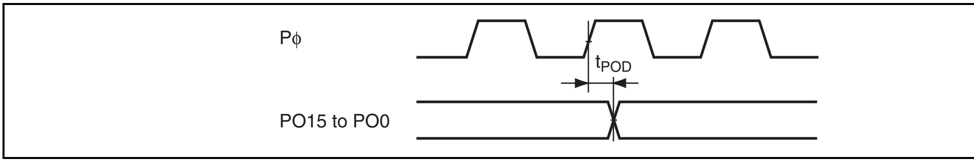


Figure 21.28 PPG Output Timing

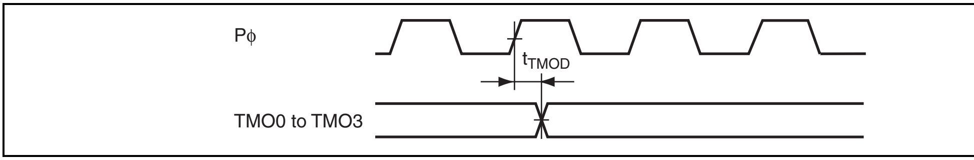


Figure 21.29 8-Bit Timer Output Timing

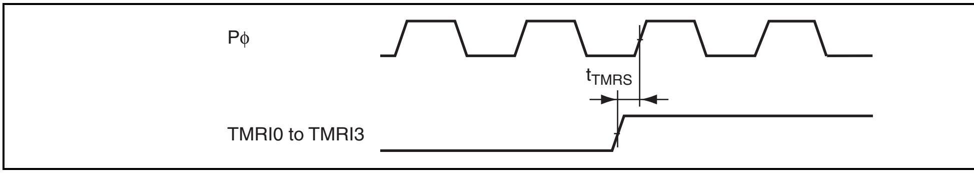


Figure 21.30 8-Bit Timer Reset Input Timing

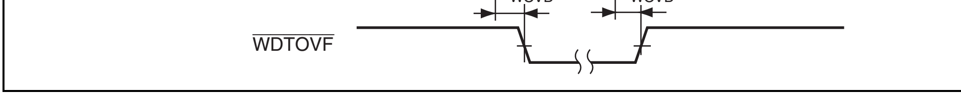


Figure 21.32 WDT Output Timing

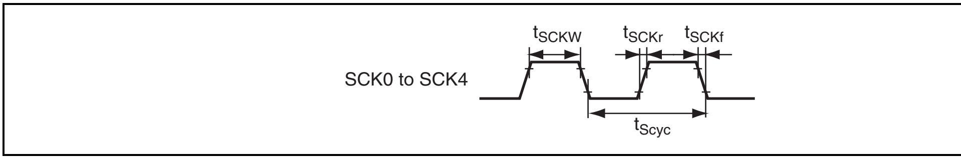


Figure 21.33 SCK Clock Input Timing

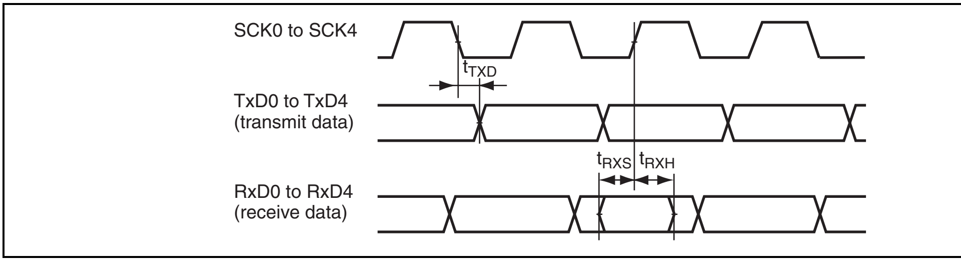


Figure 21.34 SCI Input/Output Timing: Clocked Synchronous Mode

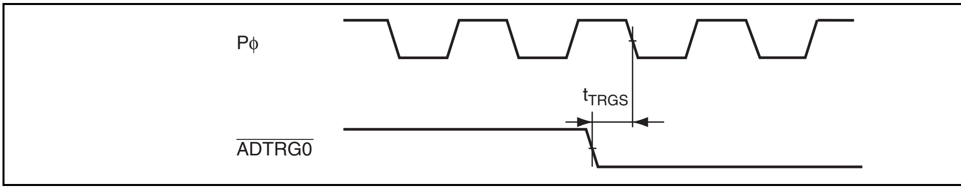


Figure 21.35 A/D Converter External Trigger Input Timing

Port 3	All	Hi-Z	Hi-Z	Keep	Keep	Keep
P50 to P55	All	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Keep
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	Keep
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	Keep
P60 to P65	All	Hi-Z	Hi-Z	Keep	Keep	Keep
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	[$\overline{\text{BREC}}$ $\overline{\text{BREQ}}$ [$\overline{\text{BS}}$ ou Hi-Z [Other Keep
PA1/ $\overline{\text{BACK}}$/ (RD/$\overline{\text{WR}}$)	All	Hi-Z	Hi-Z	[$\overline{\text{BACK}}$ output] Hi-Z [RD/$\overline{\text{WR}}$ output] Keep [Other than above] Keep	[$\overline{\text{BACK}}$ output] Hi-Z [RD/$\overline{\text{WR}}$ output] Hi-Z [Other than above] Keep	[$\overline{\text{BACK}}$ $\overline{\text{BACK}}$ [RD/$\overline{\text{W}}$ Hi-Z [Other Keep
PA2/ $\overline{\text{BREQ}}$/ $\overline{\text{WAIT}}$	All	Hi-Z	Hi-Z	[$\overline{\text{BREQ}}$ input] Hi-Z [$\overline{\text{WAIT}}$ input] Hi-Z [Other than above] Keep	[$\overline{\text{BREQ}}$ input] Hi-Z [$\overline{\text{WAIT}}$ input] Hi-Z [Other than above] Keep	[$\overline{\text{BREC}}$ Hi-Z (E [$\overline{\text{WAIT}}$ Hi-Z (V [Other Keep

PA6/ $\overline{AS}$/ $\overline{AH}$/ $\overline{BS}$-B	External extended mode	H	Hi-Z	[$\overline{AS}$, $\overline{BS}$ output] H [$\overline{AH}$ output] L [Other than above] Keep	[$\overline{AS}$, $\overline{AH}$, $\overline{BS}$ output] Hi-Z [Other than above] Keep	[$\overline{AS}$, $\overline{AH}$ output] Hi-Z [Other th Keep
PA7/Bϕ	External extended mode	Clock output	Hi-Z	[Clock output] H [Other than above] Keep	[Clock output] H [Other than above] Keep	[Clock o Clock ou [Other th Keep
PB0/ $\overline{CS0}$/ $\overline{CS4}$-A/ $\overline{CS5}$-B	External extended mode	H	Hi-Z	[$\overline{CS}$ output] H [Other than above] Keep	[$\overline{CS}$ output] Hi-Z [Other than above] Keep	[$\overline{CS}$ outp Hi-Z [Other th Keep
PB1/ $\overline{CS1}$/ $\overline{CS2}$-B/ $\overline{CS5}$-A/ $\overline{CS6}$-B/ $\overline{CS7}$-B	All	Hi-Z	Hi-Z	[$\overline{CS}$ output] H [Other than above] Keep	[$\overline{CS}$ output] Hi-Z [Other than above] Keep	[$\overline{CS}$ outp Hi-Z [Other th Keep
PB2/ $\overline{CS2}$-A/ $\overline{CS6}$-A	All	Hi-Z	Hi-Z	[$\overline{CS}$ output] H [Other than above] Keep	[$\overline{CS}$ output] Hi-Z [Other than above] Keep	[$\overline{CS}$ outp Hi-Z [Other th Keep
PB3/ $\overline{CS3}$/ $\overline{CS7}$-A	All	Hi-Z	Hi-Z	[$\overline{CS}$ output] H [Other than above] Keep	[$\overline{CS}$ output] Hi-Z [Other than above] Keep	[$\overline{CS}$ outp Hi-Z [Other th Keep

PF7/CS4-C/ $\overline{\text{CS5}}$-C/ $\overline{\text{CS6}}$-C/ $\overline{\text{CS7}}$-C					[$\overline{\text{CS}}$ output] H* [Other than above] Keep	[$\overline{\text{CS}}$ output] Hi-Z* [Other than above] Keep	[$\overline{\text{CS}}$ ou Hi-Z* [Other Keep
Port H	External extended mode		Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
Port I	External extended mode	8-bit bus mode	Hi-Z	Hi-Z	Keep	Keep	Keep
		16-bit bus mode	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z

[Legend]

H: High-level output

L: Low-level output

Keep: Input pins become high-impedance, output pins retain their state.

Hi-Z: High impedance

Note: * This is the state when PCR is cleared to 0. Since setting PCR to 1 turns on th pull-up MOS, do not set PCR to 1 if the pin is used as $\overline{\text{CS}}$ output.

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LQFP120-14x14-0.40	PLQP0120LA-A	120P6R-A / FP-120B / FP-120BV	0.7g

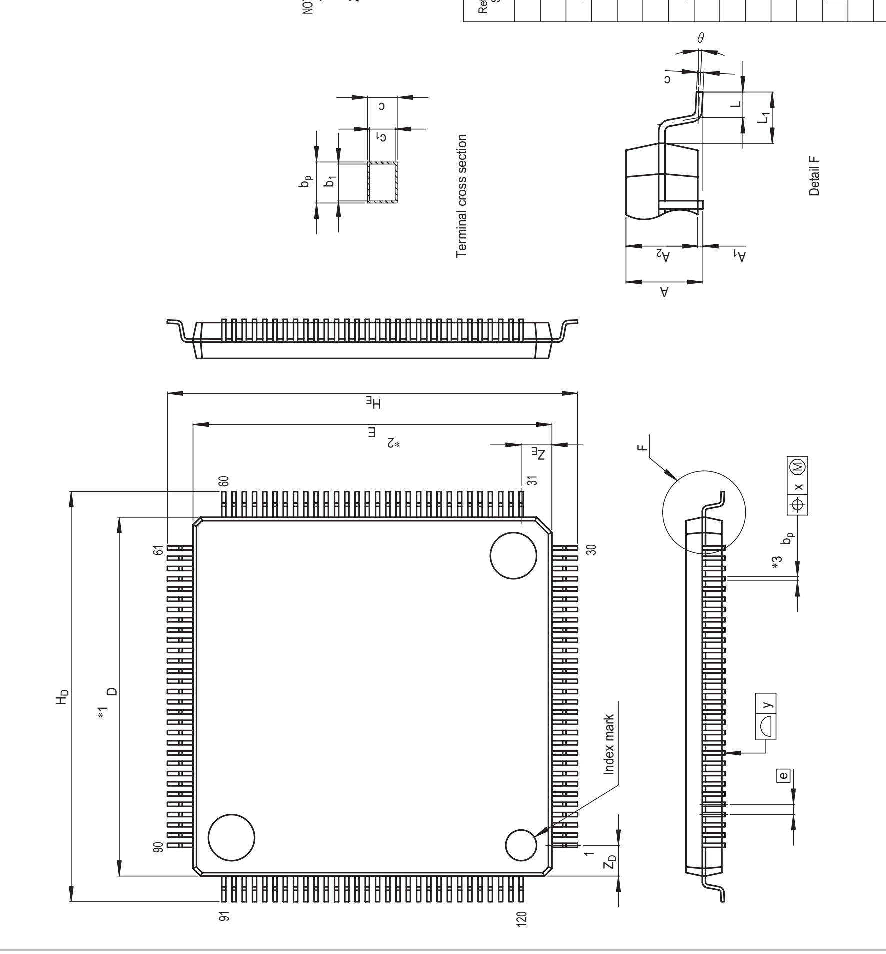


Figure C.1 Package Dimensions (FP-120BV)

MD2, MD1, MD0	(Always used as operating mode pins.)
NMI	• Connect to V_{CC} via a pull-up resistor.
EXTAL	(Always used as a clock pin.)
XTAL	• Leave this pin unconnected.
$\overline{\text{WDTOVF}}$	• Leave this pin unconnected.
Port 1 Port 2 Port 3 Port 6 PA2 to PA0 PB3 to PB0 PF7 to PF5	• Connect each pin to V_{CC} via a pull-up resistor or to V_{SS} via a pull- resistor.
Port 5	• Connect each pin to AV_{CC} via a pull-up resistor or to AV_{SS} via a pu resistor.
PA7	• Since this is the Bϕ output in its initial state, leave this pin unconn
PA6	• Since this is the AS output in its initial state, leave this pin uncon
PA5	• Since this is the RD output in its initial state, leave this pin uncon
PA4	• Since this is the LHWR output in its initial state, leave this pin unconnected.
PA3	• Since this is the LLWR output in its initial state, leave this pin unconnected.
PB0	• Since this is the CS0 output in its initial state, leave this pin unco

Vref	• Connect to AV_{CC}.

Notes: 1. Do not change the function of an unused pin from its initial state.

2. Do not change the initial value (input buffer disabled) of PnICR corresponding unused pin.

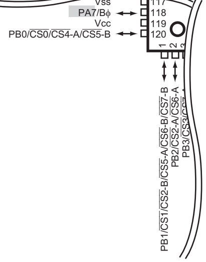

Vss
117
PA7/Bϕ
118
Vcc
119
PB0/$\overline{CS0}$/$\overline{CS4}$-A/$\overline{CS5}$-B
120
1
2
3
PB1/$\overline{CS1}$/$\overline{CS2}$-B/$\overline{CS5}$-A/$\overline{CS6}$-B/$\overline{CS7}$-B
PB2/$\overline{CS2}$-A/$\overline{CS6}$-A
PB3/$\overline{CS3}$/$\overline{CS7}$-A

Table 1.3 Pin Functions 14 Modified

Classification	Pin Name
I/O ports	PA7, PA6, PA4 PA2 to PA0

Table 3.3 Pin Functions in Each Operating Mode (Advanced Mode) 69 Modified

Port		Mode 4	Mo
Port A	PA7	P/C*	P/
	PA6, PA4	P/C*	P/
	PA2 to PA0	P*/C	P*
Port B	PB3 to 1	P*/C	P*
	PB0	P/C*	P/

10.4.5 PWM Modes 454 Modified

(b) PWM mode 2

PWM output is generated using one TGR as t
register and the others as duty cycle registers
output specified in TIOR is performed by mea
compare matches. Upon counter clearing by a
register compare match, the output value of e

Renesas 32-Bit CISC Microcomputer
Hardware Manual
H8SX/1651 Group

Publication Date: Rev.1.00, Jan. 16, 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.

RenesasTechnology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, J

RENESAS

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/1651 Group
Hardware Manual

RENESAS

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